# Semi-Quantum Money

Roy Radian
Or Sattath
Computer Science Department, Ben-Gurion University of the Negev
Israel

## ABSTRACT

Private quantum money allows a bank to mint quantum money states that it can later verify, but that no one else can forge. In classically verifiable quantum money – introduced by Gavinsky (CCC 2012) – the verification is done via an interactive protocol between the bank and the user, where the communication is classical, and the computational resources required of the bank are classical. In this work, we consider memoryless interactive protocols in which the minting is likewise classical, and construct a private money scheme that achieves these two notions simultaneously (i.e., classical verification *and* classical minting). We call such a construction a private semi-quantum money scheme, since all the requirements from the bank in terms of computation and communication are classical.

In terms of techniques, our main contribution is a strong parallel repetition theorem for Noisy Trapdoor Claw Free Functions (NTCF), a notion introduced by Brakerski et al. (FOCS 2018).

## KEYWORDS

Quantum cryptography, Quantum Money, Trapdoor Claw Free Functions, Semi-Quantum Money

**ACM Reference format:**
Roy Radian and Or Sattath. 2019. Semi-Quantum Money . In *Proceedings of AFT '19: Conference on Advances in Financial Technologies, Zurich, Switzerland, October 21–23, 2019 (AFT '19),* 15 pages.
https://doi.org/10.1145/3318041.3355462

## 1 INTRODUCTION

Introduced by Wiesner circa 1969, quantum money was the precursor to what is now known as quantum cryptography [33]. The motivation behind quantum money is to design money that is physically impossible to counterfeit, by using a variant of the (quantum) no-cloning theorem [9, 25, 34]. This notion of quantum money is in sharp contrast to our current notions of bills and coins that, at least in principle,can be counterfeited.

All quantum money schemes consist of three parts: key-gen, which generates a key, mint which uses the key to issue a new quantum money state, and verify which tests whether an alleged money state is legitimate. There are two main categories of quantum money: private and public. In a private setting, the key is required to run the verification. On the other hand, in a public quantum money scheme, key-gen generates a secret/public key-pair, where the secret key is used in mint and the public key is used in verify. In this work we will focus mainly on private quantum money schemes.

A variant of quantum money called classically verifiable quantum money was introduced in [12] (see also [4, 13, 26]), which does not require quantum communication for verification: the money is verified via a protocol between the user and the bank that requires a quantum computer for the user, a classical computer for the bank, and classical communication between them.

In this work, we introduce a new variant of classically verifiable quantum money: semi-quantum money. In this setting, the minting also shares this property, i.e., it is a protocol that involves *both* the bank *and* the user, and requires only classical resources from the bank. In standard quantum money, in contrast, minting is a quantum algorithm run by the bank, which sends the output – the quantum money state – to the user, via a quantum channel.

In semi-quantum money, the money state is generated by the *user* – this concept seems somewhat counter intuitive with regard to the standard notion of banks; if banknotes are generated by the user, couldn't the user create as many notes as he or she pleases? The key point of the minting process is the protocol between the user and the bank: the user is supposed to generate a superposition over two registers using information provided by the bank, measure one of the registers, and report the result back to the bank. If the user will try to repeat the same procedure, the measurement outcome – as well as the post-measured state – will be different with overwhelming probability. As far as the authors are aware, no prior work considered classical minting.

The fact that semi-quantum money is also classically verifiable means that instead of sending the quantum state to the bank for verification, the user and the bank run a classical interactive verification protocol that tests the validity of the money. Semi-quantum money got its name from the fact that the minting and verification protocols require only classical resources (communication and computation) from the bank.

This introduction of a quantum money scheme where the banks are classical perhaps raises the question whether the concept could be improved, such that the bank would be quantum and the user classical. However, such a setting is inherently flawed; if the user is classical, they could not hold their own money, meaning the bank would have to hold the state of every note of every user[1]. This makes the "quantumness" of the money redundant, since it would be permanently kept within the bank in any case. Thus, it would seem that the setting where the bank and communication is

*AFT '19, October 21–23, 2019, Zurich, Switzerland*

ACM ISBN 978-1-4503-6732-5/19/10…$15.00
https://doi.org/10.1145/3318041.3355462

[1] We refer to such a scheme as "memory-dependent", and explore its consequences in the full version [28].

classical and the user is quantum is the "least quantum" a quantum money scheme could be.

In this work we present a construction for *private* semi-quantum money. However, the notion of semi-quantum money naturally lends itself to the public setting. In a future version we intend to present public semi-quantum money, based on the quantum lightning construction of [35] and Coladangelo's follow-up work [8] which introduced the notion of bolt-to-certificate. Public schemes have a major advantage over private schemes in terms of privacy: in a public scheme, any user can verify a banknote without aid from the bank. Thus, the bank cannot track all transactions of the note; only those made directly with it. It is much harder to construct a secure public scheme, however. Our classical verification based on [8] is memory-dependent, meaning the bank has to keep a database of spent notes. We leave it as an open question whether a *memoryless* public semi-quantum money exists (we compare memory-dependent vs. memoryless quantum money in the full version [28]).

Our main result is the following theorem:

**Theorem 1.1** (Main Theorem). *Assuming that the Learning With Errors (LWE) problem with certain parameters is hard for BQP and that a post-quantum existentially unforgeable under an adaptive chosen message attack MAC and an encryption scheme with post-quantum indistinguishability under adaptive chosen plaintext attack (see the full version for the definitions [28]) exist, then a secure semi-quantum private money scheme exists (Definition 3.2).*

The main technical tool through which to implement this scheme is the quantum secure trapdoor claw-free function recently introduced in [5] (see also [14, 20, 21]). Informally, a quantum secure TCF is a family of functions, where each function $f:\{0,1\}^w \rightarrow \{0,1\}^w$ in the family (a) is classically efficiently computable, (b) is 2-to-1, i.e., for every $x$ there exists a unique $x' \neq x$ such that $f(x)=f(x')$, and (c) has a trapdoor that, given $y$, can be used to find $x$ and $x'$ such that $f(x)=f(x')=y$ (when $y$ is in the image of $f$), but without the trapdoor a quantum polynomial adversary cannot find any pair $x,\ x'$ such that $f(x)=f(x')$.

In addition, we will require the adaptive hardcore bit property of a TCF that was introduced in [5], which is explained below. Using a quantum computer, the state $\frac{1}{\sqrt{2}}(|x\rangle+|x'\rangle)$, where $x$ and $x'$ are two pre-images of $y$, could be measured, and one pre-image of $y$ could be found. Moreover, by measuring the state in the Hadamard basis, a string $d$ that satisfies $d\cdot(x\oplus x')=0$ could be extracted:

$$\begin{aligned} H^{\otimes w}\frac{1}{\sqrt{2}}(|x\rangle+|x'\rangle) &= \frac{1}{\sqrt{2^{w+1}}}\sum_{d\in\{0,1\}^w}(-1)^{d\cdot x}+(-1)^{d\cdot x'}|d\rangle \\ &= \frac{1}{\sqrt{2^{w-1}}}\sum_{d\in\{0,1\}^w|d\cdot(x\oplus x')=0}(-1)^{d\cdot x}|d\rangle \end{aligned} \quad (1)$$

In our construction we use the following two tests: the pre-image test (providing a pre-image of $y$) and the equation test (providing a non-zero $d$ that satisfies the above condition). The adaptive hardcore bit property guarantees that, even though either test on its own can be easily passed, it is hard (for a quantum polynomial time (QPT) adversary) to successfully pass both tests with a probability that is noticeably higher than $\frac{1}{2}$. Brakerski et al. showed a construction of a *noisy* trapdoor claw-free function (NTCF) that holds this adaptive hardcore property, based on the hardness of the Learning With Errors (LWE) problem [5]. For the sake of clarity, we ignore the issues related to the noisy property in this introduction. Below is presented the outline and analysis of our semi-quantum private money scheme construction.

The security notion of our money scheme is rather straightforward: an adversary that receives $t$ banknotes, and can attempt to pass verification (polynomially) many times, cannot pass more than $t$ verifications. To show a construction that meets this notion, we work our way through several weaker security notions; this makes proving the security of our full scheme construction simpler. We first show how to construct a semi-quantum money scheme (Section 3) that provides a weaker level of security than a full scheme. Here, we wish to show that a counterfeiter that receives 1 quantum money state cannot create two states that will both pass verification with non-negligible probability. We call a scheme that satisfies this weaker notion of security a 2-of-2 mini-scheme – see Definition 3.4.

We now describe the construction of a 2-of-2 mini-scheme, starting with the (honest) minting protocol. The bank picks $n$ functions $f_1,\ldots,f_n$ uniformly at random from the TCF family and sends them to the user, while keeping the trapdoors $t_1,\ldots,t_n$ private. The user creates a superposition of the form $|\psi_1\rangle\otimes\ldots\otimes|\psi_n\rangle$, where $|\psi_i\rangle=\frac{1}{\sqrt{2^w}}\sum_{x\in\{0,1\}^w}|x\rangle\otimes|f_i(x)\rangle$. The user measures all the r.h.s. registers (i.e., $|f_i(x)\rangle\ \forall 1\leq i\leq n$) and sends the resulting $y_1,\ldots,y_n$ to the bank, who saves them to its database[2]. Note that due to the measurement, the $i^{th}$ state collapses to $|\psi_{y_i}\rangle=\frac{1}{\sqrt{2}}(|x_i\rangle+|x_i'\rangle)$, where $f_i(x_i)=f_i(x_i')=y_i$.

For verification, the bank chooses a random challenge $C_i\in_R\{0,1\}$ (which is either the pre-image or the equation challenge) for each of the $n$ registers. For the pre-image challenge, $C_i=0$, the user must provide a string $x_i$ such that $f_i(x_i)=y_i$. The honest user can measure $|\psi_{y_i}\rangle$ to find a pre-image of $y_i$ to pass this test with certainty. In the equation challenge, $C_i=1$, the user must provide a non-zero string $d_i\in\{0,1\}^w$ such that $d_i\cdot(x_i\oplus x_i')=0$. The bank can test whether the equation challenge holds by using the trapdoor $t_i$ to calculate both $x_i$ and $x_i'$. An honest user can generate such a string by measuring $|\psi_{y_i}\rangle$ in the Hadamard basis, as described in Eq. (1). The measured $d_i$ will be non-zero (except with probability exponentially small in $w$) which will allow the user to pass this test.

We emphasize that for both the minting and the verification protocols, the bank only needs a classical computer.

We now outline the security argument. Suppose the user tries to pass verification twice. Denote by $C\in\{0,1\}^n$ the challenge vector in the first attempt, denote by $C'$ the challenge vector in the second attempt, and denote by $S$ the set of coordinates in which they differ: $S=\{i\in[n]|C_i\neq C_i'\}$. We expect $|S|$ to be roughly $\frac{n}{2}$. To pass the tests in each round $i\in S$ in both attempts, the user must pass both the pre-image *and* the equation challenges. We know that the success probability of passing both tests for each $i\in S$ is $\frac{1}{2}+\mathsf{negl}(\lambda)$. To argue that the probability of passing all these tests becomes exponentially small with $n$, we need some sort of a parallel repetition theorem (see Section 2.3). Luckily, we can rephrase this

[2] We deviate here slightly from the formal definitions; Since the bank does not have a "database", verification should only use the key. This is handled by using a message authentication code (MAC) and by returning to the user a tag for these values, and then verifying that tag during the verification. For the sake of clarity, we omit this part in the discussion – refer to Algorithm 3 to see how we work around this issue.

setting using the framework of *weakly verifiable puzzles* for which a (perfect) parallel repetition theorem is known [7]. This parallel repetition guarantees that answering these $\frac{n}{2}$ puzzles correctly is as hard as trying to answer them independently, i.e., at most $\left(\frac{1}{2}\right)^{n/2}$ (up to negligible corrections), which is exactly our goal.

The construction above is a semi-quantum 2-of-2 mini-scheme (rather than a full blown scheme). There is a slightly stronger notion of security (that is still weaker than a full blown scheme) called a mini-scheme (adapted from Aaronson and Christiano [2]). In a mini-scheme, the counterfeiter is given a single quantum money state and can attempt to pass verification polynomially many times. The counterfeiter succeeds if at least two of these verifications are accepted. We show in Section 3.2 that the scheme above also achieves this stronger notion.

In a full quantum money scheme, the adversary can ask for $t$ money states and must pass at least $t+1$ verifications. Aaronson and Christiano [2] defined the notion of a *public* money mini-scheme and showed how such a mini-scheme can be lifted to a full-blown scheme. Ben-David and Sattath [4] showed a similar result that lifts a *private* money mini-scheme to a full-blown scheme. In this work, we show how to lift an *interactive* private money mini-scheme to a full-blown scheme. The goal of such a mapping is to ensure that the scheme can support the issuance of multiple money states without increasing the key-size. This is done by using an authenticated encryption scheme for the mini-scheme keys and including that authenticated ciphertext as part of the money. As part of the verification, the bank can later decrypt the mini-scheme key, and use it to run the original mini-scheme verification. It is important that the encryption scheme be authenticated to prevent the adversary from altering that information (which would be possible if, for example, the encryption scheme was malleable).

*Related works.* The security of private quantum money schemes is generally solid, [12, 13, 16, 22, 23, 26, 31, 33], while secure public quantum money is much harder to construct. The constructions of [1] and [2] were broken in Refs. [19, 27], respectively. The only two constructions that are not known to be broken are in [10, 35] (see also [18]) and are based either on non-standard assumptions or on generic primitives for which no candidate constructions exist.

*Our contribution.* Our contribution is twofold: the first, naturally, is private semi-quantum money: a new model of private quantum money that requires no quantum communication, and only a classical bank. The main advantage of the new scheme compared to previous quantum money schemes is the following: the new scheme could be used without quantum communication infrastructure. Classical communication has several interesting benefits over quantum communication. The most obvious one is that a classical communication infrastructure already exists; a semi-quantum money scheme – unlike previous money schemes – will not require quantum communication infrastructure. Implementing such an infrastructure on a global scale will be expensive and challenging, and might be realized years after efficient quantum computers are commonly used. There are other benefits to classical communication, even if quantum communication infrastructure was readily available. First, due to the no-cloning theorem, quantum information cannot be re-sent. In the context of quantum money, data-loss is extremely problematic – data loss means lost money. Quantum communication will naturally suffer more data-loss, at least initially. Second, for classical communication we can keep a record (and even a signed record) which helps with matters of dispute resolution, auditing and error-handling, whereas quantum communication cannot be logged. The same argument can be made for the banks themselves; classical banks could more easily keep records and be audited.

The second contribution is the parallel repetition theorem for 1-of-2 puzzles (described earlier in the introduction). Parallel repetition (the idea of repeating a protocol polynomially many times to gain an exponential increase in soundness) seems deceptively simple, while in truth it sometimes behaves in unexpected ways, and such proofs are usually challenging (see [29] and references therein for the non-cryptographic setting); [3] present several cases where parallel repetition surprisingly does not grant an exponential reduction in error rate in cryptographic-settings. The parallel repetition theorem for 1-of-2 puzzles could be useful in other cryptographic settings, as it builds on the TCF primitive to introduce a tool with an exponentially small error rate (rather than the constant error rate which is guaranteed in the original work).

*Prior Knowledge.* Before we go any further, we discuss the accessibility of this work. The reader is assumed to have a basic understanding of classical cryptography, and we follow the definitions and conventions of [15] and [17]. Familiarity with quantum computing and quantum cryptography is not necessary. Some technical concepts are used, but they are not critical to the understanding of the paper as a whole, as they are used solely in Section 2.2. For further reading, consult [24] for general quantum computing, and [6] for quantum cryptography. The only two "quantum" facts that are crucial to understand this paper are the following: (i) A qubit is the quantum analog of a bit. Unlike bits, qubits cannot be copied due to the no-cloning theorem. (ii) To extract classical information from qubits, a measurement has to be preformed. The measurement changes the quantum state. Crucially, this process is not reversible. This is in contrast to classical systems, where rewinding is possible.

*Structure.* A structural overview of our paper is shown in Fig. 1.

In Section 2, we deal with NTCF and 1-of-2 puzzles. In Section 2.1, we define a 1-of-2 puzzle. In Section 2.2, we show a construction of a $\frac{1}{2}$-hard 1-of-2 puzzle based on an NTCF. We conclude Section 2 by showing, in Section 2.3, a method for constructing a strong 1-of-2 puzzle using repetition of weak 1-of-2 puzzles.

In Section 3, we deal with our proposed semi-quantum money. Section 3.1 contains the relevant definitions. In Section 3.2, we construct a semi-quantum money mini-scheme and prove its security. In Section 3.3, we present a full semi-quantum money scheme construction based on any semi-quantum mini-scheme, and prove its security.

In Section 4 we combine the results of the preceding sections to prove our main result, namely, Theorem 1.1.

Some of the proofs, and preliminary standard definitions are deferred to the full version [28].

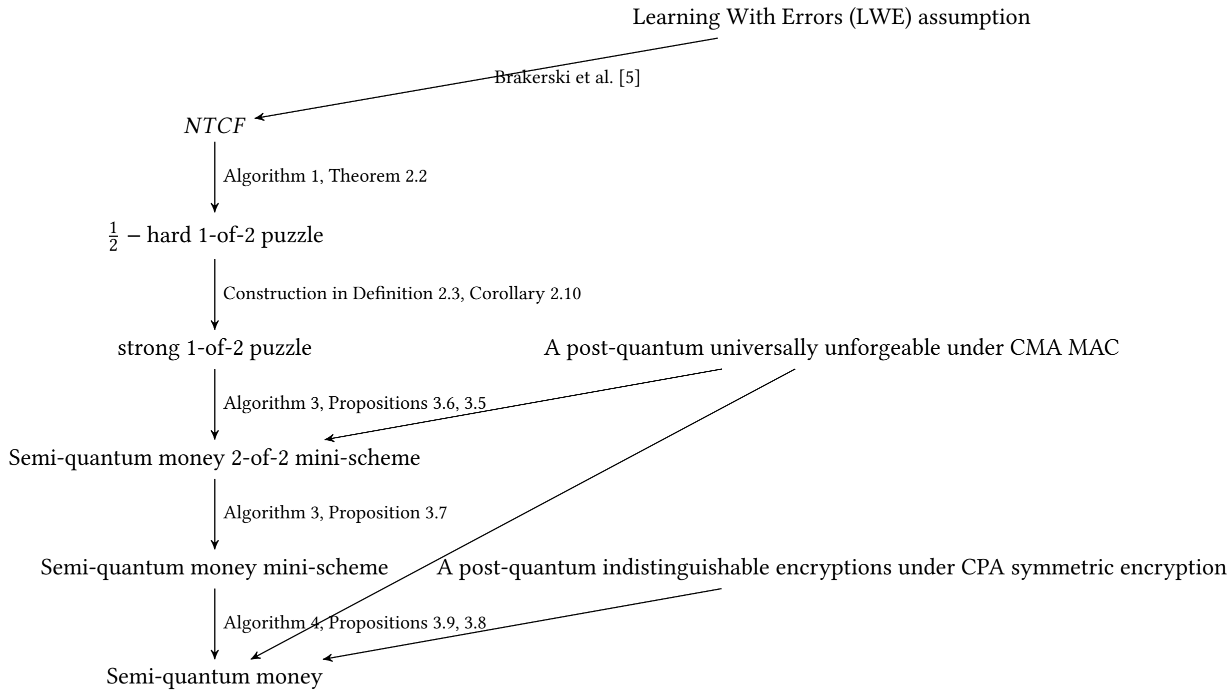
Learning With Errors (LWE) assumption
Brakerski et al. [5]
NTCF
Algorithm 1, Theorem 2.2
½ – hard 1-of-2 puzzle
Construction in Definition 2.3, Corollary 2.10
strong 1-of-2 puzzle
A post-quantum universally unforgeable under CMA MAC
Algorithm 3, Propositions 3.6, 3.5
Semi-quantum money 2-of-2 mini-scheme
Algorithm 3, Proposition 3.7
Semi-quantum money mini-scheme
A post-quantum indistinguishable encryptions under CPA symmetric encryption
Algorithm 4, Propositions 3.9, 3.8
Semi-quantum money

**Figure 1: Structure of our construction. The right-hand side of the figure shows our *assumptions*. The arrows point to constructions that make use of these assumptions.**

## 2 TRAPDOOR CLAW-FREE FAMILIES AND 1-OF-2 PUZZLES

In this section, as the name suggests, we discuss the concepts of NTCF and 1-of-2 puzzles. In Section 2.1, we introduce 1-of-2 puzzles. In Section 2.2 we show how to construct a 1-of-2 puzzle using an NTCF, and in Section 2.3 we show a parallel repetition theorem for 1-of-2 puzzles that is subsequently used to construct strong 1-of-2 puzzles.

### 2.1 1-of-2 Puzzles

*Definition 2.1 (1-of-2 puzzle).* A 1-of-2 puzzle consists of four efficient algorithms: the puzzle generator $G$, an obligation algorithm $O$, a 1-of-2 solver $S$, and a verification algorithm $V$. $G$ is a classical algorithm, $V$ is a classical deterministic algorithm, and $O$ and $S$ are quantum algorithms.

(1) $G$ outputs, on security parameter $1^\lambda$, a random puzzle $p$ and some verification key $v$: $(p, v) \leftarrow_{\$} G(1^\lambda)$.
(2) $O$ receives a puzzle $p$ as input and outputs a classical string $o$ called the obligation and a quantum state $\rho$: $(o, \rho) \leftarrow_{\$} O(p)$.
(3) $S$ receives $p, o, \rho$ and a bit $b \in \{0, 1\}$ as input and outputs an answer string $a$: $a \leftarrow_{\$} S(p, o, \rho, b)$.
(4) $V$ receives $p, v, o, b, a$ as input and outputs 0 or 1: $V(p, v, o, b, a) \in \{0, 1\}$.

Completeness: Let $\eta$ be some arbitrary function $\eta : \mathbb{N} \mapsto [0, 1]$. We say that the 1-of-2 puzzle has completeness $\eta$ if there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$\begin{aligned}\Pr[(p, v) \leftarrow_{\$} G(1^\lambda), (o, \rho) \leftarrow_{\$} O(p), b \leftarrow_{\$} \{0, 1\}, a \leftarrow_{\$} S(p, o, \rho, b) :&\\ V(p, v, o, b, a) = 1]&\\ \geq \eta(\lambda) - \mathsf{negl}(\lambda)\,.&\end{aligned} \tag{2}$$

We define the $V_2$ algorithm as:

$$V_2(p, v, o, a_0, a_1) = V(p, v, o, 0, a_0) \cdot V(p, v, o, 1, a_1) \tag{3}$$

Hardness: Let $h : \mathbb{N} \mapsto [0, 1]$ be an arbitrary function. We say that the 1-of-2 puzzle $\mathcal{Z}$ is $1 - h$ hard if for any QPT 2-of-2 solver $\mathcal{T}$ there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$\Pr[\mathsf{SOLVE} - 2_{\mathcal{T},\mathcal{Z}}(\lambda) = 1] \leq h(\lambda) + \mathsf{negl}(\lambda) \tag{4}$$

The 2-of-2 solving game $\mathsf{SOLVE} - 2_{\mathcal{T},\mathcal{Z}}(\lambda)$:

(1) The puzzle giver runs $(p, v) \leftarrow_{\$} G(1^\lambda)$
(2) The 2-of-2 solver $\mathcal{T}$ receives input $p$ and outputs a triple $(o, a_o, a_1)$
(3) The puzzle giver runs $r \leftarrow V_2(p, v, o, a_0, a_1)$ and outputs $r$
(4) $\mathcal{T}$ wins the game if and only if $r = 1$, in which case the output of the game is defined to be 1.

We say that the 1-of-2 puzzle is strong if $\eta = 1$ and $h = 0$ (i.e., the puzzle is 1-hard). We say that the 1-of-2 puzzle is weak if $\eta = 1$ and $1 - h$ is noticeable.

### 2.2 An NTCF Implies a 1-of-2 Puzzle

This section presents how an NTCF can be used to construct a 1-of-2 puzzle. The formal definition of an NTCF and its properties used in this section can be found in the full version [28], as well as in Ref. [5].

Theorem 2.2. *An NTCF implies a 1-of-2 puzzle with completeness $\eta = 1$ and hardness $h = \frac{1}{2}$.*

Note that the 1-of-2 puzzle above is a weak 1-of-2 puzzle.

Proof. The proof contains arguments similar to those used by Brakerski et al. [5].

Given an NTCF family $\mathcal{F}$ that consists of the algorithms

$$\text{key-gen}_{\mathcal{F}}, \text{Inv}_{\mathcal{F}}, \text{CHK}_{\mathcal{F}}, \text{SAMP}_{\mathcal{F}}, J_{\mathcal{F}}$$

we construct the 1-of-2 puzzle $\mathcal{Z} = (\text{key-gen}_{\mathcal{Z}}, O_{\mathcal{Z}}, S_{\mathcal{Z}}, V_{\mathcal{Z}})$ as specified in Algorithm 1.

Completeness: we need to show that Eq. (2) holds for $\mathcal{Z}$ defined above. By the efficient range superposition property of NTCF [28], the state $|\psi'\rangle$ in line 2 of the algorithm $O_{\mathcal{Z}}$ is negligibly close in trace distance to:

$$|\tilde{\psi}\rangle = \frac{1}{\sqrt{2|\mathcal{X}|}} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}, b \in \{0,1\}} \sqrt{(f'_{k,b}(x))(y)}|b, x\rangle|y\rangle$$

For the sake of the analysis, therefore, we can replace $|\psi\rangle$ with $|\tilde{\psi}\rangle$, and the algorithm will behave the same, up to a negligible probability. By the injective pair property of NTCF, the post-measurement state $|\psi\rangle$ generated by $O_{\mathcal{Z}}$ is $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle|x_0\rangle + |1\rangle|x_1\rangle)$, where $(x_0, x_1) \in \mathcal{R}_p$. Since $o$ was the outcome of the measurement in line 3, we know that $o \in \textsc{Supp} f_{p,i}(x_i)$. By the trapdoor property of NTCF, for $i \in \{0, 1\}$:

$$x_i = \text{INV}_{\mathcal{F}}(v, i, o) \tag{5}$$

Consider the case $b = 0$. In this case, the output of $S_{\mathcal{Z}}$ is $a \equiv (i, x_i)$, where, by Eq. (5), $x_i = \text{INV}_{\mathcal{F}}(v, i, o)$. Therefore, $V_{\mathcal{Z}}$ will return 1 in line 6. In the case of $b = 1$, before line 6 in $S_{\mathcal{Z}}$ the state is $\frac{1}{\sqrt{2}}(|0\rangle|x_0\rangle + |1\rangle|x_1\rangle)$, after the evaluation of $J$ on the second register the state is $\frac{1}{\sqrt{2}} \sum_{j \in \{0,1\}} |j\rangle|J(x_j)\rangle$, and after the Hadamard on both registers (which consist of $w + 1$ qubits), the state is

$$\frac{1}{\sqrt{2^{w+2}}} \sum_{i \in \{0,1\}, d \in \{0,1\}^w} \left( \sum_{j \in \{0,1\}} (-1)^{ij + d \cdot J(x_j)} \right) |i\rangle|d\rangle$$
$$= \frac{1}{\sqrt{2^w}} \sum_{d \in \{0,1\}^w} (-1)^{d \cdot J(x_0)} |d \cdot (J(x_0) \oplus J(x_1))\rangle|d\rangle$$

Therefore, the outcome of the measurement in line 7 will provide a random $d \in \{0, 1\}^w$ and an $i \in \{0, 1\}$ that satisfy $i = d \cdot (J(x_0) \oplus J(x_1))$. Since $d$ is random, the adaptive hardcore bit property guarantees that the first condition in line 12 of $V_{\mathcal{Z}}$ will be met (up to a negligible probability), and the analysis in the previous sentence guarantees that the second condition will be met. Overall, the probability that $V_{\mathcal{Z}}$ outputs 1 is $1 - \text{negl}(\lambda)$, for some negligible function negl, as required.

Soundness: We need to show that Eq. (4) holds for every QPT $\mathcal{T}$. In Algorithm 2, we show a reduction that maps a 2-of-2 solver $\mathcal{T}$ for the 1-of-2 puzzle as in Eq. (4) to an NTCF adversary $\mathcal{A}$.

If $\mathcal{T}$ succeeds with probability $\frac{1}{2} + \epsilon(\lambda)$ (where $\epsilon(\lambda)$ is not necessarily negligible), then from the adaptive hardcore bit property, plugging the definition of $V_2$ (see Eq. (3)) and the acceptance criteria of $V_{\mathcal{Z}}$ into lines 6 and 12, we see that the 2-of-2 solver $\mathcal{T}$ needs to find $o, i, x, d, i'$ such that $d \in G_{p,0,x_0} \cap G_{p,1,x_1}$ and $x = x_i$, where $x_0 = \text{INV}_{\mathcal{F}}(v, 0, o)$, $x_1 = \text{INV}_{\mathcal{F}}(v, 1, o)$ and $i' = d \cdot (J(x_0) \oplus J(x_1))$. This implies the membership of $(i, x, d, i')$ in $H_k$ ). Therefore, $\Pr_{(k, t_k) \leftarrow\$ \text{key-gen}_{\mathcal{F}}(1^\lambda)}[\mathcal{A}(k) \in H_k] \geq \frac{1}{2} + \epsilon(\lambda)$. Since $H_k$ and $\overline{H}_k$ are disjoint, $\Pr_{(k, t_k) \leftarrow\$ \text{key-gen}_{\mathcal{F}}(1^\lambda)}[\mathcal{A}(k) \in \overline{H}_k] \leq \frac{1}{2} - \epsilon(\lambda)$, and

$$\left| \Pr_{(k,t_k) \leftarrow \text{key-gen}_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) \in H_k] - \Pr_{(k,t_k) \leftarrow \text{key-gen}_{\mathcal{F}}(1^\lambda)} [\mathcal{A}(k) \in \overline{H}_k] \right|$$
$$\geq 2\epsilon(\lambda) .$$

By the injective pair property we conclude that $\epsilon(\lambda)$ must be negligible, as required for $h = \frac{1}{2}$.

□

## 2.3 A Parallel Repetition Theorem for 1-of-2 Puzzles

*Definition 2.3 (Repetition of 1-of-2 puzzle).* Let $\mathcal{Z}$ be a 1-of-2 puzzle system, and let $n \in \mathbb{N}$. We denote by $G^n$ the algorithm that, on security parameter $\lambda$, runs $G(1^\lambda)$ for $n(\lambda)$ times and outputs all the $n$ puzzles with their verification keys:

$$((p_1, \ldots, p_n)), (v_1, \ldots, v_n)) \leftarrow\$ G^n(1^\lambda) \tag{6}$$

where $(p_i, v_i) \leftarrow\$ G(1^\lambda)$. A similar approach is used for all other algorithms in $\mathcal{Z}$:

$$((o_1, \ldots, o_n)), (\rho_1 \otimes \cdots \otimes \rho_n)) \leftarrow\$ O^n(p_1, \ldots, p_n) \tag{7}$$

where $(o_i, \rho_i) \leftarrow\$ O(p_i)$.

$$(a_1, \ldots, a_n) \leftarrow\$ S^n((p_1, \ldots, p_n), (o_1, \ldots, o_n), \rho_1 \otimes \cdots \otimes \rho_n, b)$$

where $a_i \leftarrow\$ S(p_i, o_i, \rho_i, b)$. The algorithm

$$V^n((p_1, \ldots, p_n), (v_1, \ldots, v_n), (o_1, \ldots, o_n), b, (a_1, \ldots, a_n))$$

outputs 1 iff for all $i \in [n]$, $V(p_i, v_i, o_i, b, a_i) = 1$.

The $n$-fold repetition of $\mathcal{Z}$ is the 1-of-2 puzzle

$$\mathcal{Z}^n = (G^n, O^n, S^n, V^n)$$

We emphasize that $\mathcal{Z}^n$ is a 1-of-2 puzzle (and not a 1-of-$2^n$ puzzle), which explains why the algorithm contains a single challenge bit $b$ rather than $n$ bits. The reason for that should be made clear later – see Fact 2.8.

Theorem 2.4 (Parallel repetition of 1-of-2 puzzles). *Let $\mathcal{Z}$ be a 1-of-2 puzzle with completeness $\eta$ and hardness parameter $h$. For a function $n(\lambda)$ that satisfies $n(\lambda) = \text{poly}(\lambda)$, the 1-of-2 puzzle $\mathcal{Z}^n$ has completeness $\eta^n$ and hardness parameter $1 - h^n$.*

Proof. First we prove the completeness property (see Eq. (2)). For ease of notation, we write $n$, negl, $\eta$ ,etc., instead of $n(\lambda)$, $\text{negl}(\lambda)$, $\eta(\lambda)$. Suppose that the success probability of $\mathcal{Z}$ is $\eta - \text{negl}$ for some negligible function negl. Since the repeated game $\mathcal{Z}^n$ is an independent repetition of $\mathcal{Z}$, its success probability is $(\eta - \text{negl})^n$. We show that for the negligible function $\text{negl}' = n^2\text{negl}(\lambda)$, indeed

**Algorithm 1** The 1-of-2 Puzzle $\mathcal{Z}$

```
procedure key-gen$_{\mathcal{Z}}(\lambda)$
    $(k, t_k) \leftarrow_\$ \text{key-gen}_{\mathcal{F}}(\lambda)$
    Set $p \equiv k$, $v \equiv t_k$
    return $(p, v)$
end procedure
procedure $O_{\mathcal{Z}}(p)$
    $|\psi'\rangle \leftarrow_\$ \text{SAMP}_{\mathcal{F}}(p, |+\rangle)$
    Measure the last register to obtain an $o \in \mathcal{Y}$. Denote the post-measurement state $|\psi\rangle$    ▷ In the completeness we show that $|\psi\rangle \approx \frac{1}{\sqrt{2}}(|0\rangle|x_0\rangle + |1\rangle|x_1\rangle)$.
    return $(o, |\psi\rangle)$
end procedure
procedure $S_{\mathcal{Z}}(p, o, |\psi\rangle, b)$    ▷ $p$ and $o$ are not used in this construction.
    if $b = 0$ then
        Measure both registers of $|\psi\rangle$ to obtain a result $i \in \{0,1\}$ and $x \in \mathcal{X}$.
        Set $a \equiv (i, x)$
    else if $b = 1$ then
        Evaluate the function $J$ on the second register of $|\psi\rangle$, and apply Hadamard transform on both registers.
        Measure both registers to obtain the result $i \in \{0,1\}$ and $d$.
        Set $a \equiv (i, d)$
    end if
    return $a$
end procedure
procedure $V_{\mathcal{Z}}(p, v, o, b, a)$
    Set $x_0 \equiv \text{INV}_{\mathcal{F}}(v, 0, o)$ and $x_1 \equiv \text{INV}_{\mathcal{F}}(v, 1, o)$    ▷ Recall that $v$ is the trapdoor, and $o$ is an image of the NTCF.
    if $b = 0$ then
        Interpret $a$ as $i, x$
        if $x = x_i$ then
            return 1
        else
            return 0
        end if
    else if $b = 1$ then
        Interpret $a$ as $i, d$.
        if $d \in G_{p,0,x_0} \cap G_{p,1,x_1}$ and $d \cdot (J(x_0) \oplus J(x_1)) = i$ then    ▷ This membership test uses CHK$_{\mathcal{F}}$
            return 1
        else
            return 0
        end if
    end if
end procedure
```

**Algorithm 2** The Adversary $\mathcal{A}$

```
procedure $\mathcal{A}_{\mathcal{F}}(k)$
    $(o, a_0, a_1) \leftarrow_\$ \mathcal{T}(k)$
    Interpret $a_0$ as $i, x$ and $a_1$ as $i', d$.
    return $(i, x, d, i')$
end procedure
```

$(\eta - \text{negl})^n \geq \eta^n - \text{negl}'$:

$$\begin{aligned}(\eta - \text{negl})^n &= \eta^n + \sum_{k=1}^{n} (-1)^k \binom{n}{k} \eta^{n-k} \text{negl}^k \\ &\geq \eta^n - \sum_{k=1}^{n} n^k \text{negl}^k \\ &\geq \eta^n - \sum_{k=1}^{n} n \cdot \text{negl} = \eta^n - \text{negl}',\end{aligned}$$

where the last inequality holds for all $\lambda \geq \lambda_0$ (where $n \cdot \text{negl} \leq 1$).

We are now ready to prove the soundness. Our main tool is the notion of a weakly verifiable puzzle system defined by Canetti, Halevi and Steiner:

*Definition 2.5 (A weakly verifiable puzzle, adapted from [7]).* A system for weakly verifiable puzzles consists of a pair of efficient classical algorithms $\hat{\mathcal{Z}} = (G, V)$ such that

(1) The puzzle generator $G$ outputs, on security parameter $\lambda$, a random puzzle $p$ along with some verification information $v$: $(p, v) \leftarrow_\$ G(1^\lambda)$.

(2) The puzzle verifier $V$ is a deterministic efficient classical algorithm that, on input of a puzzle $p$, verification key $v$, and answer $a$, outputs either zero or one: $V(p, v, a) \in \{0, 1\}$.

The hardness of a weakly verifiable puzzle is defined as follows:

*Definition 2.6 (Hardness of a weakly verifiable puzzle, adapted from [7]).* Let $h : \mathbb{N} \mapsto [0, 1]$ be an arbitrary function. A weakly verifiable puzzle $\hat{\mathcal{Z}}$ is said to be $1-h$ hard if, for any QPT[3] algorithm $S$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that:

$$\Pr[\mathsf{SOLVE}_{\mathcal{S},\hat{\mathcal{Z}}}(\lambda)] \leq h(\lambda) + \mathsf{negl}(\lambda)$$

The event $\mathsf{SOLVE}_{\mathcal{S},\hat{\mathcal{Z}}}(\lambda)$ is defined by the following security game:

(1) The puzzle giver runs $(p, v) \leftarrow_{\$} G(1^{\lambda})$
(2) The solver $\mathcal{S}$ is given input $p$ and outputs an answer $a$
(3) The puzzle giver runs $r \leftarrow V(p, v, a)$. The event $\mathsf{SOLVE}_{\mathcal{S},\hat{\mathcal{Z}}}(\lambda)$ is when $r = 1$.

To avoid confusion, we always use $\mathcal{Z}$ to denote a 1-of-2 puzzle and $\hat{\mathcal{Z}}$ to denote a weakly verifiable puzzle.

*Definition 2.7 (Repetition of weakly verifiable puzzles, from [7]).* Let $\hat{\mathcal{Z}} = (G, V)$ be a weakly verifiable puzzle system, and let $n : \mathbb{N} \mapsto \mathbb{N}$ be some function. We denote by $G^n$ the algorithm that, on security parameter $\lambda$, runs $G(1^{\lambda})$ for $n(\lambda)$ times and outputs all the $n$ puzzles with their respective verification keys:

$$((p_1, \ldots, p_n)), (v_1, \ldots, v_n)) \leftarrow_{\$} G^n(1^{\lambda})$$

where $(p_i, v_i) \leftarrow_{\$} G^n(1^{\lambda})$. $V^n$ receives $n$ inputs and accepts if and only if all $n$ runs of $V$ accept:

$$V^n((p_1, \ldots, p_n), (v_1, \ldots, v_n), (a_1, \ldots, a_n)) \equiv \prod_{i=1}^{n(\lambda)} V(p_i, v_i, a_i)$$

There is a tight relation between the hardness of a 1-of-2 puzzle and the hardness of a weakly verifiable puzzle. Given a 1-of-2 puzzle $\mathcal{Z} = (G, O, S, V)$, we define the weakly verifiable puzzle

$$\hat{\mathcal{Z}} = (G, V_2)$$

(where $V_2$ is defined in Eq. (3)).

**Fact 2.8.** *For every polynomially bounded function $n : \mathbb{N} \mapsto \mathbb{N}$, the 1-of-2 puzzle $\mathcal{Z}^n$ is $1 - h$-hard if and only if the weakly verifiable puzzle $\hat{\mathcal{Z}}^n$ is $1 - h$-hard.*

This fact follows from the observation that the hardness property of the 1-of-2 puzzle $\mathcal{Z}$ is equivalent to the hardness of the weakly verifiable puzzle $\hat{\mathcal{Z}}$ (see Definitions 2.5 and 2.1). Furthermore, the hardness of $\mathcal{Z}^n$ is equivalent to the hardness of $\hat{\mathcal{Z}}^n$ (see Definitions 2.3 and 2.7).

Canetti, Halevi and Steiner proved a parallel repetition theorem for weakly verifiable puzzles.

Theorem 2.9 ([7]). *Let $\epsilon : \mathbb{N} \mapsto [0, 1]$ be an efficiently computable function, let $n : \mathbb{N} \mapsto \mathbb{N}$ be efficiently computable and polynomially bounded, and let $\hat{\mathcal{Z}} = (G, V)$ be a weakly verifiable puzzle system. If $\hat{\mathcal{Z}}$ is $1 - h$-hard, then $\hat{\mathcal{Z}}^n$, the n-fold repetition of $\hat{\mathcal{Z}}$, is $1 - h^n$-hard.*

Although the original proof of Canetti, Halevi and Steiner assumed that the hardness is with respect to a classical solver, the proof uses a black-box reduction that also holds with respect to quantum solvers, as defined in this work.

We use Theorem 2.9 to prove the soundness of the 1-of-2 puzzle $\mathcal{Z}^n$. We assume $\mathcal{Z} = (G, O, S, V)$ is $1-h$ hard. We define the weakly verifiable puzzle $\hat{\mathcal{Z}} = (G, V_2)$. By the equivalence in Fact 2.8, we know that $\hat{\mathcal{Z}}$ is also $1 - h$ hard. By Theorem 2.9, we know that $\hat{\mathcal{Z}}^n$ is $1 - h^n$-hard. Using the equivalence in Fact 2.8 again, we conclude that $\mathcal{Z}^n$ is $1 - h^n$-hard, which completes the proof. □

Corollary 2.10. *A weak 1-of-2 puzzle implies a strong 1-of-2 puzzle.*

Note that we define a weak 1-of-2 puzzle to have completeness $\eta = 1$. We refrain from answering the question whether any puzzle in which $\eta(\lambda) - h(\lambda)$ is noticeable, implies a strong puzzle.

Proof. By using Theorem 2.4 with $n(\lambda) = \frac{\log^2(\lambda)}{\log(\frac{1}{h})}$ repetitions[4] of the weak $h$-hard 1-of-2 puzzle, we construct a 1-complete [5], $1 - h^n = 1 - \frac{1}{\lambda^{\log(\lambda)}} = 1 - \mathsf{negl}(\lambda)$-hard 1-of-2 puzzle. Note that a $1 - \mathsf{negl}(\lambda)$-hard 1-of-2 puzzle is equivalent to a 1-hard 1-of-2 puzzle, which completes the proof. □

# 3 STRONG 1-OF-2 PUZZLES IMPLY SEMI-QUANTUM MONEY

In this section, we show a construction of a semi-quantum money scheme using strong 1-of-2 puzzles.

In Section 3.1, we define *interactive* quantum money. We define three degrees of security. Full scheme security means that every QPT counterfeiter cannot pass $t + 1$ verifications given $t$ quantum money states. We define mini-scheme security as a weaker variant of full security, which is secure only when the adversary is given a single banknote. Finally, we define 2-of-2 mini-scheme security as an even weaker variant wherein the adversary does not have a banknote verification oracle. We also formally define semi-quantum money.

In Section 3.2, we show the construction of a 2-of-2 mini-scheme, and show that our 2-of-2 mini-scheme is in fact a mini scheme (see Definition 3.4).

In Section 3.3, we show that any (interactive private quantum money) mini scheme can be elevated to a full (interactive private quantum money) scheme – see Definition 3.3.

## 3.1 Definitions of Semi-Quantum Money

*Definition 3.1 (Interactive memoryless private quantum money).* An interactive memoryless private quantum money scheme consists of a classical PPT key generation algorithm key-gen and two-party interactive memoryless QPT protocols mint and verify. key-gen$(1^{\lambda})$ outputs a key $k$. Both the minting protocol and the verification protocol are two-party quantum protocols involving the Acquirer (a user), denoted $A$, and a Bank, denoted $B$. During both protocols, the bank receives the key $k$ as input, and the user does not. At the

[3] In [7] this notion is defined for any PPT algorithm.

[4] Note that $n(\lambda)$ is indeed polynomial in $\lambda$ - since a weak 1-of-2 puzzle holds that $1 - h$ is noticeable (see Definition 2.1), by using the inequality $\ln(1 - \varepsilon) \leq -\varepsilon$ we get that $\log(1/h)$ is noticeable.

[5] Recall that a weak 1-of-2 puzzle has completeness $\eta = 1$ (see Definition 2.1).

end of the honest run of mint, the user holds a quantum money state that, in general, could be a mixed state. In this work, the protocols will end with a pure state, usually denoted $|\$\rangle$. In the following sections, for the sake of clarity, we work with the pure-state formalism. The banknote the user chooses to verify is denoted in this work as the input of the verify protocol. At the end of the verification protocol, the bank outputs a bit $b$ that states whether the money is valid or not.

*Correctness.* The scheme is *correct* if there exists a negligible function $\mathsf{negl}(\lambda)$ such that:

$$\Pr(k \leftarrow_\$ \mathsf{key\text{-}gen}(1^\lambda); |\$\rangle \leftarrow_\$ \mathsf{mint}_k(1^\lambda); b \leftarrow_\$ \mathsf{verify}_k(|\$\rangle) : b = 1) = 1 - \mathsf{negl}(\lambda)$$

*Definition 3.2.* We say that the protocol has classical minting (verification) if $B$ is classical in mint (verify). To emphasize that the verification is classical, we use cverify to denote the (calssical) verifiaction algorithm. We define private semi-quantum money as any secure memoryless interactive private quantum money protocol that has classical minting *and* classical verification.

In the quantum setting, there are a number of possible verifications with different qualities; a notable quality is whether the verification "destroys" the banknote (i.e., whether the banknote can be used again after verification). This distinction can be thought of as the difference between verifying – proving that a legal money state exists – and spending – proving a legal money state doesn't exist – and it becomes more interesting when considering the public setting; there, a banknote can be spent with the bank in the same manner as in the private setting, but it can also be verified with other users – in such a case it is important that the banknote is preserved, so it could be transferred. Another distinction is added by the introduction of classically verified money: whether the verification is a classical or quantum protocol. Moreover, a classical verification must be a challenge-response protocol – otherwise the same proof can be passed twice, effectively spending the same banknote twice. In our scheme, verification is classical and does not preserve the banknote, proving both that it existed and that it does not exist anymore.

In this definition, we emphasize that the protocols mint and verify are *memoryless*: i.e., all outgoing messages depend solely on the key and the input from the user. In other words, the bank does not maintain a variable state that changes between different runs of the protocols – each run is independent. Constructing a stateful scheme is trivial even in the classical setting, as discussed in the full version [28]. In addition, it is interesting to note that our protocols are composed of a fixed number of messages, independent of the security parameter: verify has 2 messages (a single round) and mint has 3 messages.

*Definition 3.3.* We say that an interactive private quantum money scheme \$ is secure if for every QPT counterfeiter $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\lambda)$ such that:

$$\Pr[\mathsf{COUNTERFEIT}^{full}_{\mathcal{A},\$}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$$

The money counterfeiting game $\mathsf{COUNTERFEIT}^{full}_{\mathcal{A},\$}(\lambda)$:

(1) The bank generates a key $k \leftarrow_\$ \mathsf{key\text{-}gen}(1^\lambda)$.
(2) The bank and the counterfeiter interact. The counterfeiter can ask the bank to run $\mathsf{mint}_k(\cdot)$ and $\mathsf{verify}_k(\cdot)$ polynomially many times, in any order the counterfeiter wishes. The counterfeiter is not bound to following his side of the protocols honestly. The counterfeiter can keep ancillary registers from earlier runs of these protocols and use them in later steps. Let $w$ be the number of successful verifications, $\ell$ the number of times that mint was called by the counterfeiter and $v$ the number of times that verify was called by the counterfeiter.
(3) The bank outputs $(w, \ell, v)$.

The value of the game is 1 iff $w > \ell$. In this case we sometimes simply say that the counterfeiter wins.

Following previous works [2, 13], we define a private quantum money mini-scheme, with a slight deviation. Additionally, we define a 2-of-2 mini-scheme, which is a weaker variant of the mini-scheme.

*Definition 3.4 (quantum money mini-scheme and 2-of-2 mini-scheme).* We define mini-scheme security as we defined full scheme security but with regard to $\mathsf{COUNTERFEIT}^{mini}_{\mathcal{B},\$}(\lambda)$, wherein the counterfeiter $\mathcal{B}$ wins iff $w > \ell \wedge \ell = 1$.

We define 2-of-2 mini-scheme security as we did above but with regard to $\mathsf{COUNTERFEIT}^{2-of-2}_{C,\$}(\lambda)$, where the counterfeiter $C$ wins iff $w > \ell \wedge \ell = 1 \wedge v = 2$.

Note that the definitions in this sections could be naturally extended to the public settings.

## 3.2 Construction of a Mini-Scheme

In this section, we show the construction of a scheme that we then prove to be a 2-of-2 semi-quantum mini-scheme. Later we prove that our construction in fact achieves a stronger security notion – a semi-quantum mini-scheme.

We now give an informal description of our construction, which is defined formally in Algorithm 3. The construction uses a strong 1-of-2 puzzle (see Definition 2.1) and a post-quantum existentially unforgeable under an adaptive chosen-message attack (PQ-EU-CMA) MAC . In key-gen, the bank generates a MAC signing key and $n$ pairs of strong 1-of-2 puzzles and their respective verification keys. The minting process is done as follows. The bank sends these $n$ puzzles to the user, who then runs the obligation protocol $\mathcal{Z}.O$ on all the $n$ puzzles. The user keeps the quantum output of $O$ and sends the classical outputs (called the obligations) to the bank. The bank signs these obligations using the classical MAC scheme and sends these tags back to the user. The verification starts with the bank sending random challenges to the user. The user then has to present a set of signed obligations (which the user should have from the mint protocol) together with a set of solutions to the challenges of these puzzles. The bank verifies the solution to each puzzle with its respective verification key (the set of verification keys is part of the key). Due to the fact that this verification is classical, it is denoted cverify. We show that a counterfeiter cannot double-spend a banknote without breaking the soundness of a strong 1-of-2 puzzle (or the security of the MAC).

Intuitively, an adversary could try to double-spend the banknote using the solutions he received from the first verification, while hoping to be given the same challenges. However, assuming a sufficiently large number of puzzles (say, $n = \log^2(\lambda)$), the probability of encountering the exact same set of challenges more than once is negligible. Passing two verifications of any banknote in which the challenges were not the same both times essentially requires one to pass the SOLVE − 2 security game for the 1-of-2 puzzle. Insofar as this is considered a strong 1-of-2 puzzle, the probability that it can occur is therefore negligible.

For ease of notation, we write:

- $\overline{p}^n := (p_1, \dots, p_n)$
- $\overline{v}^n := (v_1, \dots, v_n)$
- $\overline{o}^n := (o_1, \dots, o_n)$
- $\overline{\psi}^n := |\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle$
- $\overline{b}^n := (b_1, \dots, b_n)$
- $\overline{a}^n := (a_1, \dots, a_n)$

Proposition 3.5 (Correctness of $\$_{\mathcal{Z}}$). *Assuming MAC has perfect completeness and $\mathcal{Z}$ is a 1-of-2 puzzle with completeness $\eta = 1$, $\$_{\mathcal{Z}}$ (Algorithm 3) is a semi-quantum money scheme that satisfies the correctness property (see Definition 3.1).*

Proof. Clearly, the communication and the bank's operation in mint and cverify are classical – therefore, the scheme is semi-quantum.

From the *perfect completeness* property of MAC (see the full version [28])we get:

$$\Pr[MAC.\mathsf{verify}_k(\overline{o}^n, MAC.\mathsf{mac}_k(\overline{o}^n)) = 1] = 1$$

meaning $\Pr[r_{MAC} = 1] = 1$.

From the completeness $\eta = 1$ of $\mathcal{Z}$ we get:

$$\begin{aligned}\Pr[&(p, v) \leftarrow_\$ \mathcal{Z}.G(\lambda); (o, |\psi\rangle) \leftarrow_\$ \mathcal{Z}.O(p); b \leftarrow_\$ \{0, 1\};\\ &a \leftarrow_\$ \mathcal{Z}.S(p, o, |\psi\rangle, b) :\\ &\mathcal{Z}.v(p, v, o, b, a) = 1]\\ &\geq 1 - \mathsf{negl}(\lambda)\end{aligned}$$

Let $b_i$ be the event of failing verification on the $i^{th}$ puzzle. From the previous equation, $\Pr[b_i] \leq \mathsf{negl}(\lambda)$ for some negligible function $\mathsf{negl}(\lambda)$. Let $\mathsf{negl}'(\lambda) := n \cdot \mathsf{negl}(\lambda) = \log^2(\lambda) \cdot \mathsf{negl}(\lambda)$. Using the union bound:

$$\Pr[\cup_{i=1}^n b_i] \leq \sum_{i=1}^{n} \Pr[b_i] = \log^2(\lambda) \cdot \mathsf{negl}(\lambda) = \mathsf{negl}'(\lambda)$$

meaning $\Pr[\left(\prod_{i=1}^n r_i\right) = 1] \geq 1 - \mathsf{negl}'(\lambda)$. Thus:

$$\begin{aligned}\Pr[&k_\$ \leftarrow_\$ \$_{\mathcal{Z}}.\mathsf{key\text{-}gen}(1^\lambda); (\overline{p}^n, \overline{o}^n, t_o, \overline{\psi}^n) \leftarrow_\$ \$_{\mathcal{Z}}.\mathsf{mint}_{k_\$}();\\ &\$_{\mathcal{Z}}.\mathsf{cverify}_{k_\$}(\overline{p}^n, \overline{o}^n, t_o, \overline{\psi}^n) = 1]\\ =& \Pr[r_{MAC} = 1 \bigcap \left(\prod_{i=1}^{n} r_i\right) = 1]\\ \geq& 1 - \mathsf{negl}'(\lambda)\end{aligned}$$

□

Proposition 3.6 ($\$_{\mathcal{Z}}$ is a 2-of-2 mini-scheme). *Assuming $\mathcal{Z}$ is a strong 1-of-2 puzzle and MAC is a PQ-EU-CMA MAC, the scheme $\$_{\mathcal{Z}}$ (Algorithm 3) is a 2-of-2 mini-scheme (see Definition 3.4).*

Proof. We show that the probability of a QPT counterfeiter to win the 2-of-2 mini-scheme security game against $\$_{\mathcal{Z}}$ (Algorithm 3) is bound by the negligible probability to solve both challenges of the strong 1-of-2 puzzle $\mathcal{Z}$. Intuitively, double-spending a banknote entails solving both challenges for at least one of its $n$ puzzles, which is intractable. For this proof, as well as the following security proofs of our money scheme (Proposition 3.7 and Theorem 3.9), we use a sequence-of-games based technique adapted from [30]. The following sequence of games binds the success probability of any QPT 2-of-2 mini-scheme counterfeiter to that of a QPT 2-of-2 puzzle solver (see Eq. (4)):

*Game 0.* Let $C$ be a QPT 2-of-2 mini-scheme counterfeiter. We assume w.l.o.g. that $C$ performs exactly two verifications and one mint (i.e., $\ell = 1$ and $v = 2$) – an adversary which does not comply with this assumption will necessarily fail (see Definition 3.4). We define Game 0 to be $\mathsf{COUNTERFEIT}^{2-of-2}_{C,\$_{\mathcal{Z}}}(\lambda)$.

Let $S_0$ be the event where $w > 1$ (see Definition 3.4) in Game 0 (this is the original win condition for $C$, since we assume $\ell = 1 \wedge v = 2$).

*Game 1.* We now transform Game 0 into Game 1, simply by changing the win condition: game 1 is identical to game 0, but we define the following event: let $\overline{b^1}^n, \overline{b^2}^n$ be the random bit strings that were generated in line 3 of $\$_{\mathcal{Z}}$.cverify the first and second times $C$ asked for verification, respectively. Let $S_1$ be the event where $w > 1 \wedge \overline{b^1}^n \neq \overline{b^2}^n$ in Game 1.

Let $F$ be the event where $\overline{b^1}^n = \overline{b^2}^n$ in Game 1, and $F'$ the event where $w > 1 \wedge \overline{b^1} = \overline{b^2}$ in Game 1. Since $\overline{b^1}^n$ and $\overline{b^2}^n$ are generated uniformly and independently, $\Pr[F] = \frac{1}{2^n} \leq \mathsf{negl}(\lambda)$ for some negligible function $\mathsf{negl}(\lambda)$. Therefore: $\Pr[S_0] = \Pr[S_1 \cup F'] \leq \Pr[S_1 \cup F] \leq \Pr[S_1] + \Pr[F] \leq \Pr[S_1] + \mathsf{negl}(\lambda)$. So $\Pr[S_0] \leq \Pr[S_1] + \mathsf{negl}(\lambda)$.

*Game 2.* We now add a small change to the game above: at the start of the game, a uniform $i' \in_R [n]$ is chosen by the bank. Let $j$ be the first index such that $b^1_j \neq b^2_j$ ($j = \infty$ if $b^1 = b^2$).

Let $S_2$ be the event where $w > 1 \wedge b^1 \neq b^2 \wedge i' = j$ in Game 2.

$S_1 \Rightarrow b^1 \neq b^2$, so since $i'$ was chosen uniformly and independently of $w$, $b^1$, $b^2$ and $j$, we get that $\Pr[S_2|S_1] = \frac{1}{n}$. Moreover, it is easy to see that $\Pr[S_2|\neg S_1] = 0$. So $\Pr[S_2] = \frac{1}{n} \cdot \Pr[S_1]$, meaning $\Pr[S_1]$ is a polynomial multiplicative factor of $\Pr[S_2]$.

*Game 3.* Game 3 is identical to Game 2, but we now add an additional constraint to the win condition. Let $\overline{o}^n$ be the set of obligations $C$ sent in line 7 of $\$_{\mathcal{Z}}$.mint, and let $\overline{o^1}^n, \overline{o^2}^n$ be the sets of obligations sent by $C$ during line 6 of $\$_{\mathcal{Z}}$.cverify the first and second times $C$ asks for verification, respectively.

Let $S_3$ be the event where $w > 1 \wedge b^1 \neq b^2 \wedge i' = j \wedge \overline{o^1}^n = \overline{o^2}^n = \overline{o}^n$ in Game 3.

Let $F$ be the event where $C$ passes one or more verifications such that $\overline{o^1}^n \neq \overline{o}^n$ or $\overline{o^2}^n \neq \overline{o}^n$. It is easy to see that $S_2 \wedge \neg F \iff S_3 \wedge \neg F$. Therefore, from the Difference Lemma ([30], see also the full version [28]) we get

$$|\Pr[S_3] - \Pr[S_2]| \leq \Pr[F]$$

**Algorithm 3** The Interactive Private Money Scheme $\$_{\mathcal{Z}}$

```
$\$_{\mathcal{Z}}$.key-gen($1^\lambda$)
1 :  $n \leftarrow \log^2(\lambda)$
2 :  foreach $i \in [n]$ :
3 :      $(p_i, v_i) \leftarrow \mathcal{Z}.G(1^\lambda)$
4 :  $\overline{p}^n \leftarrow (p_1, \ldots, p_n)$, $\overline{v}^n \leftarrow (v_1, \ldots, v_n)$
5 :  $k \leftarrow MAC$.key-gen($1^\lambda$)
6 :  $k_\$ \leftarrow (\overline{p}^n, \overline{v}^n, k)$
7 :  return $k_\$$
```

```
$\$_{\mathcal{Z}}$.mint$_{k_\$}$
1 :  Acquirer                                              Bank
2 :
3 :                          $\overline{p}^n$
                        <--------------------
4 :  foreach $i \in [n]$ :
5 :      $(o_i, \psi_i) \leftarrow \mathcal{Z}.O(p_i)$
6 :  $\overline{o}^n \leftarrow (o_1, \ldots, o_n)$, $\overline{\psi}^n \leftarrow (\psi_1, \ldots, \psi_n)$
7 :                          $\overline{o}^n$
                        -------------------->
8 :                                                        $t_o \leftarrow MAC.\mathsf{mac}_k(\overline{o}^n)$
9 :                          $t_o$
                        <--------------------
```

```
$\$_{\mathcal{Z}}$.cverify$_{k_\$}(\overline{o}^n, t_o, \overline{\psi}^n)$
1 :  Acquirer                                    Bank
2 :
3 :                $\overline{b}^n \in_R \{0, 1\}^n$
              <--------------------
4 :  foreach $i \in [n]$ :
5 :      $a_i \leftarrow \mathcal{Z}.S(p_i, o_i, |\psi_i\rangle, b_i)$
6 :                $\overline{a}^n, \overline{o}^n, t_o$
              -------------------->
7 :                                              $r_{MAC} \leftarrow MAC.\mathsf{verify}_k(\overline{o}^n, t_o)$
8 :                                              foreach $i \in [n]$ :
9 :                                                  $r_i \leftarrow \mathcal{Z}.V(p_i, v_i, o_i, b_i, a_i)$
10 :                                             $r \leftarrow r_{MAC} \cdot \prod_{i=1}^{n} r_i$
11 :                                             return $r$
```

From the unforgeability of MAC , $\Pr[F]$ is negligible[6]. Therefore, $\Pr[S_2] \leq \Pr[S_3] + \mathrm{negl}(\lambda)$.

*Game 4.* We now change the behavior of verifications. Let $\overline{a^1}^n$, $\overline{a^2}^n$ be the sets of answers sent by $C$ in line 6 of $\$_{\mathcal{Z}}$.cverify the first and second times $C$ asks for verification, respectively[7]. Instead of performing verifications both times, the bank now performs both verifications only on the second time: the first time $\$_{\mathcal{Z}}$.cverify is called, after line 6 the bank returns 1 and stops. The second time $\$_{\mathcal{Z}}$.cverify is called, the bank performs both verifications: i.e., on the second verification we replace everything from line 9 with:

[6] Otherwise, we could construct a MAC forger $\mathcal{F}$ with non-negligible success probability. Assume towards a contradiction that with non-negligible probability, $C$ passes verification by sending in line 6 $\overline{o'}^n$, $t'_o$ such that $\overline{o'}^n \neq \overline{o}^n$. That means that the MAC verification in line 7 passed. So $\mathcal{F}$ could simulate a bank, but instead of signing and verifying with $k$ generated in $\$_{\mathcal{Z}}$.key-gen, $\mathcal{F}$ uses the signing and verification oracles. $\mathcal{F}$ runs $C$ against the simulated bank, and present $\overline{o'}^n$, $\tilde{o'}^n$. With non-negligible probability, MAC verification passes, and since $\overline{o'}^n \neq \overline{o}^n$, and no other signings are requested (mint was run only once), $\mathcal{F}$ wins MAC $-$ FORGE$_{\mathcal{F},MAC}(\lambda)$.

[7] $C$ can, of course, run both verification protocols simultaneously. We number the verifications according to the one that got to line 6 of the protocol first.

```
8 :  foreach $i \in [n]$ :
9 :      $r_i \leftarrow \mathcal{Z}.V(p_i, v_i, o_i, \boxed{b_i^1, a_i^1})$
10 :     $\boxed{r'_i \leftarrow \mathcal{Z}.V(p_i, v_i, o_i, b_i^2, a_i^2)}$
11 : endforeach
12 : $r \leftarrow r_{MAC} \cdot \prod_{i=1}^{n} r_i \boxed{\cdot r'_i}$
13 : return $r$
```

Let $S_4$ be the event where $w > 1 \wedge b^1 \neq b^2 \wedge i' = j \wedge \overline{o^1}^n = \overline{o^2}^n = \overline{o}^n$ in Game 4.

Verifying both inputs on the second request is equivalent to verifying them individually: $S_3 \Rightarrow S_4$ since if both verifications pass in Game 3, then both pass in Game 4 (the first one always passes, the second one runs both verifications that passed in $S_3$), and $\neg S_3 \Rightarrow \neg S_4$ since that means one of the verifications in Game 3 fail, which means the second verification in Game 4 fails. So $\Pr[S_3] = \Pr[S_4]$.

*Game 5.* We now change the second verification: on the $i'^{th}$ pair of puzzles, if $b_{i'}^1 \neq b_{i'}^2$ (we note that this always holds when $i' = j$), we perform $V_2$ instead of normal verification – i.e., we replace everything from line 9 forward in $\$_{\mathcal{Z}}$.cverify in the second verification with:

```
8 :  foreach $i \in [n]$ :
9 :      $\boxed{\textbf{if } i = i' \wedge b_{i'}^1 \neq b_{i'}^2 :}$
10 :         $\boxed{\textbf{if } b_i^1 = 0 : \hat{a}_0 \leftarrow a_i^1, \hat{a}_1 \leftarrow a_i^2}$
11 :         $\boxed{\textbf{else } : \hat{a}_0 \leftarrow a_i^2, \hat{a}_1 \leftarrow a_i^1}$
12 :         $\boxed{r_i, r'_i \leftarrow V_2(p_i, v_i, o_i, \hat{a}_0, \hat{a}_1)}$
13 :     $\boxed{\textbf{else } :}$
14 :         $r_i \leftarrow \mathcal{Z}.V(p_i, v_i, o_i, b_i^1, a_i^1)$
15 :         $r'_i \leftarrow \mathcal{Z}.V(p_i, v_i, o_i, b_i^2, a_i^2)$
16 :     $\boxed{\textbf{endif}}$
17 : endforeach
18 : $r \leftarrow r_{MAC} \cdot \prod_{i=1}^{n} r_i \cdot r'_i$
19 : return $r$
```

Let $S_5$ be the event where $w > 1 \wedge b^1 \neq b^2 \wedge i' = j \wedge \overline{o^1}^n = \overline{o^2}^n = \overline{o}^n \wedge V_2(p_i, v_i, o_i, \hat{a}_0, \hat{a}_1) = 1$ in Game 5.

In the case where $i = i' \wedge b_i^1 = b_i^2$, running $V_2(p_i, v_i, o_i, \hat{a}_0, \hat{a}_1)$ is equivalent to running $\mathcal{Z}.v$ twice, since we assign $\hat{a}_0$ and $\hat{a}_1$ respective to $b_i^1$ and $b_i^2$. So $S_4 \iff S_5$, meaning $\Pr[S_4] = \Pr[S_5]$.

*Game 6.* We now simply relax the win condition: Game 6 goes exactly the same as Game 5, but we define the following event: let $S_6$ be the event where $V_2(p_i, v_i, o_i, \hat{a}_0, \hat{a}_1) = 1$ in Game 6. Since this is a relaxation of the conditions of $S_5$, we get $\Pr[S_5] \leq \Pr[S_6]$.

*Bound on success probability.* We show a reduction mapping a 2-of-2 mini-scheme counterfeiter to a 2-of-2 solver (see Definition 2.6):

Let $C$ be a QPT 2-of-2 mini-scheme counterfeiter. We construct a QPT 2-of-2 solver $\mathcal{T}$ in the following manner:

Let $(p, v)$ be the output of $G(1^\lambda)$ at step 1 of the solving game. On step 2, $\mathcal{T}$ simulates a Game 6 bank (by honestly running mints and verifications as defined in game 5, as well as choosing $i'$ uniformly) with two changes:

(1) The $i'^{th}$ puzzle is replaced with $p$.
(2) In line 12 of the second verification, $\mathcal{T}$ outputs $(o_i, \hat{a}_0, \hat{a}_1)$ to the puzzle giver instead of running $V_2$. The honest puzzle giver runs $V_2(p, v, o_i, \hat{a}_0, \hat{a}_1)$ and returns the result, which $\mathcal{T}$ uses as $r_i$ and $r_{i'}$.

We can see that for any $C$, $\Pr[S_6]$ is not affected by the above changes: in the first change we replace a random puzzle with another random puzzle, which has no affect on $\Pr[S_6]$. In the second change, the honest puzzle giver runs $V_2$ with exactly the same input as the bank in the original Game 6 should, and returns the result – this also does not affect $\Pr[S_6]$.

$\mathcal{T}$ runs $C$ against Game 6. $S_6$ is exactly the win condition of the 2-of-2 solving game, which means $\mathcal{T}$ wins the 2-of-2 solving game with probability $\Pr[S_6]$. Since $\mathcal{Z}$ is a strong 1-of-2 puzzle, the success probability of any QPT 2-of-2 solver is negligible – meaning $\Pr[S_6]$ is negligible for any QPT counterfeiter.

For each pair of consecutive games $i$ and $i+1$, we have shown that $\Pr[S_i] \leq \mathsf{poly}(\lambda) \cdot \Pr[S_{i+1}] + \mathsf{negl}(\lambda)$ for some $\mathsf{poly}(\lambda), \mathsf{negl}(\lambda)$. Finally, we have shown that $\Pr[S_6]$ is negligible in $\lambda$, so we can conclude that $\Pr[S_0]$ is negligible in $\lambda$. Since Game 0 is defined as the 2-of-2 mini-scheme counterfeiting game, and $S_0$ is defined as its win condition, no QPT 2-of-2 mini-scheme counterfeiter can win the game with more than negligible probability. □

We now prove that $\$_{\mathcal{Z}}$ (Algorithm 3) is, in fact, a mini-scheme (see Definition 3.4). Unlike the others, this proof is not modular – not every 2-of-2 mini-scheme is a mini-scheme. For example, consider a scheme wherein the bank shares with the counterfeiter a single bit of the key on each verification. This scheme could have 2-of-2 mini-scheme security, but obviously, it would not be secure for a counterfeiter with a verification oracle, which could easily discern the key.

Proposition 3.7 ($\$_{\mathcal{Z}}$ is a mini-scheme). *Assuming $\$_{\mathcal{Z}}$ is a 2-of-2 mini-scheme (where $\$_{\mathcal{Z}}$ is given in Algorithm 3, and a 2-of-2 mini-scheme is defined in Definition 3.4), $\$_{\mathcal{Z}}$ is a mini-scheme (see Definition 3.4).*

Proof. We use an idea very similar to that used in [26, Theorm 5] (a slightly different variation also appeared in [4, Appendix C]); we show that if a counterfeiter with access to a verification oracle can ask for $v$ verifications and have two of them succeed, a 2-of-2 counterfeiter could guess the two success indices randomly and apply the same strategy, thus breaking the security of the 2-of-2 mini-scheme. The following sequence of games binds the success probability of any QPT mini-scheme counterfeiter to that of a QPT 2-of-2 mini-scheme counterfeiter against $\$_{\mathcal{Z}}$:

*Game 0.* Let $\mathcal{B}$ be a QPT mini-scheme counterfeiter. We assume w.l.o.g. that $\mathcal{B}$ asks for mint only once (i.e., $\ell = 1$), and for verification $v$ times such that $v$ is polynomial in $\lambda$ – an adversary which does not comply with this assumption necessarily fails (see Definition 3.4). We define the first game to be $\mathsf{COUNTERFEIT}^{mini}_{\mathcal{B},\$_{\mathcal{Z}}}(\lambda)$ (see Definition 3.4).

Let $S_0$ be the event where $w > 1$ (see Definition 3.4) in Game 0 (this is the original win condition for $\mathcal{B}$ since we assume $\ell = 1$ and $v$ is polynomial in $\lambda$).

*Game 1.* We now make one small change to Game 0, namely, that the game stops after $\mathcal{B}$ receives two successful verifications (i.e., the counterfeiter is not allowed to make additional verifications after receiving two successful ones. We model this by defining additional verification attempts as failures).

Let $S_1$ be the event where $w = 2$ in Game 1.

It is obvious why $S_1 \Rightarrow S_0$. In addition, $S_0 \Rightarrow S_1$, since any run of Game 0 with more than two successful verifications is equivalent to a run of Game 1 in which all verifications beyond the second successful one are ignored. So $\Pr[S_0] = \Pr[S_1]$.

*Game 2.* We model a run of $v$ verifications using a string $r \in \{0,1\}^v$, such that $r_i = 1$ if and only if the $i^{th}$ time $\mathcal{B}$ asked for verification was successful[8]. At the beginning of Game 2, a uniform binary string $r' \in_R \{0,1\}^v$ is generated such that $\sum_{i=1}^{v} r'_i = 2$.

Let $S_2$ be the event where $w = 2 \wedge r' = r$ in Game 2.

Given $S_1$, we know that the string $r$ representing the verifications in Game 1, like $r'$, also holds $\sum_{i=1}^{v} r_i = 2$. There are $\binom{v}{2}$ such strings, so since $r'$ was chosen uniformly and independently of $r$, there is a $\frac{1}{\binom{v}{2}}$ probability that $r' = r$. So $\Pr[S_2] = \frac{1}{\binom{v}{2}} \cdot \Pr[S_1]$, meaning $\Pr[S_1] = \binom{v}{2} \cdot \Pr[S_2]$.

*Game 3.* We transform Game 2 into Game 3 by changing the following: for each $i \in [v]$, for the $i^{th}$ time $\mathcal{B}$ runs a verification protocol with the bank, instead of receiving the actual result of the MAC and puzzle verifications ($r$), it receives $r'_i$; i.e., we change line 11 with **return** $r'_i$.

Let $S_3$ be the event where $w = 2 \wedge r' = r$ in Game 3.

Given $S_2$, since $r' = r$ in both Game 2 and Game 3, the fact that $\mathcal{B}$ receives $r'_i$ instead of $r_i$ changes nothing. So $\Pr[S_3|S_2] = 1$. Trivially, $\Pr[S_3|\neg S_2] = 0$. So $\Pr[S_2] = \Pr[S_3]$.

*Game 4.* Let $k, h$ be the two indices such that $r'_k = r'_h = 1, k \neq h$ (by construction there are exactly two such indices). In Game 4, for every verification other than the $k^{th}$ and the $h^{th}$, the MAC verification and puzzle verifications are not called at all – $b_i$ is generated and $r'_i$ is returned; i.e., lines 7 to are removed.

Let $S_4$ be the event where $w = 2 \wedge r' = r$ in Game 4.

It is easy to see that $\Pr[S_3] = \Pr[S_4]$, since for every verification but the $k^{th}$ and the $h^{th}$, the bank did nothing with the result of the MAC or puzzle verifications, so whether we run them at all changes nothing.

*Bound on success probability.* We show a reduction mapping a mini-scheme counterfeiter to a 2-of-2 mini-scheme counterfeiter (see Definition 3.4):

Let $\mathcal{B}$ be a QPT mini-scheme counterfeiter. We construct a a QPT 2-of-2 mini-scheme counterfeiter $C$ in the following manner: $C$ simulates a Game 4 bank with the following difference: when asked to run $\$_{\mathcal{Z}}.\mathsf{mint}$, it, in turn, asks the real bank to run $\$_{\mathcal{Z}}.\mathsf{mint}$ and returns the result, and on the $k^{th}$ an $h^{th}$ verifications, it asks the real bank to run $\$_{\mathcal{Z}}.\mathsf{cverify}$ and returns the result. We note that for any other verification, $C$ can simulate the bank since MAC and puzzle verifications are not performed; all it needs to do is choose a uniform $b$ and return $r'_i$. $C$ runs $\mathcal{B}$ against the simulated Game 4 bank.

So $\Pr[S_4] = \Pr[\mathsf{COUNTERFEIT}^{2-of-2}_{C,\$_{\mathcal{Z}}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$ for some negligible function $\mathsf{negl}(\lambda)$. Therefore, by construction, we get that $\Pr[S_0] \leq \mathsf{poly}(\lambda) \cdot \Pr[S_4]$ for some $\mathsf{poly}(\lambda)$ and therefore is also negligible for any QPT counterfeiter. Game 0 is defined as the original mini-scheme security game, and $S_0$ is defined as its original win condition; therefore, $\$_{\mathcal{Z}}$ (Algorithm 3) is a mini-scheme. □

## 3.3 A Mini-Scheme Implies a Full Blown Scheme

We show how a mini-scheme \$ can be used to construct a full blown scheme $\hat{\$}$. The construction is based on a very similar idea to those in [4, Appendix C] and [2, Section 3.3].

Here we provide an informal description of our full scheme $\hat{\$}$. The construction is defined formally in Algorithm 4. Our full scheme is constructed by minting mini-scheme banknotes, and including the key of the mini-scheme in each one. To that end, a MAC and a private-key encryption scheme are used: on minting, the bank mints a mini-scheme banknote, encrypts the mini-scheme key that was generated in the process, signs it in its encrypted form, and hands it to the user together with the mini-scheme banknote. The secure nature of the encryption scheme prevents the user from exploiting the mini-scheme key to break the mini-scheme's underlying security. On verification, the bank uses the MAC scheme to verify that the note was indeed minted by a bank, after which it decrypts the mini-scheme key to verify the mini-scheme banknote itself.

In both [4] and [2], the core idea of the construction is the same, with minor differences: in [4] algorithms are used instead of interactive protocols, and [2] is in the public setting, so a digital signature scheme is used instead of MAC, and an encryption scheme is not nescessary.

We prove the security of the full-blown scheme by showing a reduction mapping a full-blown scheme counterfeiter to a mini-scheme counterfeiter, such that the mini-scheme counterfeiter generates fake bank notes for the full-blown counterfeiter.

Proposition 3.8 (Correctness of $\hat{\$}$). *Assuming \$ is a* correct *mini-scheme (see Definition 3.1) and that both MAC and ENC have* perfect completeness, *$\hat{\$}$ (Algorithm 4) is* correct *(see Definition 3.1).*

Proof. From the *perfect completeness* of MAC (see the full version [28]), we get that $\Pr[S_m] = 1$, where

$$S_m := MAC.\mathsf{verify}_{k_m}(c, MAC.\mathsf{mac}_{k_m}(c)) = 1$$

Therefore, when the acquirer is honest, we know that he will send $t = MAC.sign_{k_m}(c)$ (that he received during the run of $\hat{\$}.\mathsf{mint}$) to the bank on line 3 of $\hat{\$}.\mathsf{cverify}$. Thus, the MAC verification on line 4 will succeed.

[8] $\mathcal{B}$ can, of course, run several verification protocols simultaneously. We number the verifications according to the order in which they were initiated.

**Algorithm 4** The Interactive Private Money Scheme $\hat{\$}$

```
$\hat{\$}$.key-gen$(1^\lambda)$
1 :  $k_m \leftarrow MAC$.key-gen
2 :  $k_e \leftarrow ENC$.key-gen
3 :  return $(k_m, k_e)$

$\hat{\$}$.mint$_{(k_m,k_e)}$
1 :  Acquirer                                   Bank
2 :
3 :                                             $k_\$ \leftarrow \$$.key-gen$(1^\lambda)$
4 :                                             $c \leftarrow ENC$.encrypt$_{k_e}(k_\$)$
5 :                                             $t \leftarrow MAC$.mac$_{k_m}(c)$
         $|\$\rangle \leftarrow \$$.mint$_{k_\$}()$
6 :
7 :                  $\xleftarrow{c,\ t}$

$\hat{\$}$.cverify$_{(k_m,k_e)}(c, t, |\$\rangle)$
1 :  Acquirer                                   Bank
2 :
3 :                  $\xrightarrow{c,\ t}$
4 :                                             $r_m \leftarrow MAC$.verify$_{k_m}(c, t)$
   ........................ if $r_m = 1$ ........................
5 :
6 :                                             $k_\$ \leftarrow ENC$.decrypt$_{k_e}(c)$
         $r_v \leftarrow \$$.cverify$_{k_\$}(|\$\rangle)$
7 :
8 :                                             return $r_v$
   ........................ if $r_m = 0$ ........................
9 :                                             return 0
```

From the *perfect completeness* of ENC (see the full version [28]), we get that $\Pr[S_e] = 1$, where

$$S_e := ENC.\mathsf{decrypt}_{k_e}(ENC.\mathsf{encrypt}_{k_e}(k_\$)) = k_\$$$

Therefore, when the acquirer is honest, we know that he will send $c = ENC.\mathsf{encrypt}_{k_e}(k_\$)$ (that he received during the run of $\hat{\$}$.mint) to the bank on line 3 of $\hat{\$}$.cverify. Thus, the decryption in line 6 will succeed.

From the above, we conclude that for an honest acquirer both the decryption and MAC verification in $\hat{\$}$.cverify always succeeds. As such, the verification can only fail in $\$.\mathsf{cverify}_{k_\$}$. We know that the result of the decryption is $k_\$$ as it was generated in $\hat{\$}$.mint, and that this $k_\$$ was generated by running $.key-gen. Thus, from the *correctness* of the mini-scheme $ (see Definition 3.1), we get that $\Pr[S_\$] \geq 1 - \mathsf{negl}(\lambda)$ for some negligible function $\mathsf{negl}(\lambda)$, where

$$S_\$ := k_\$ \leftarrow_\$ \$.\mathsf{key\text{-}gen}(1^\lambda); |\$\rangle \leftarrow_\$ \$.\mathsf{mint}_{k_\$}(); \quad \$.\mathsf{cverify}_{k_\$}(|\$\rangle) = 1$$

$\hat{\$}$.cverify passes when *MAC*.verify, *ENC*.decrypt and $.cverify all pass, so for an honest acquirer:

$$\begin{aligned} &\Pr[(k_m, k_e) \leftarrow_\$ \hat{\$}.\mathsf{key\text{-}gen}(1^\lambda); (c, t, |\hat{\$}\rangle) \leftarrow_\$ \hat{\$}.\mathsf{mint}_{(k_m,k_e)}();\\ &\qquad \hat{\$}.\mathsf{cverify}_{(k_m,k_e)}(c, t, |\hat{\$}\rangle) = 1]\\ &\qquad = 1 - \Pr[\neg S_m \cup \neg S_e \cup \neg S_\$]\\ &\qquad \geq 1 - \mathsf{negl}(\lambda) \end{aligned}$$

□

Theorem 3.9 ($\hat{\$}$ is a secure interactive private quantum money scheme). *Assuming \$ is an interactive private quantum money mini-scheme, MAC is a PQ-EU-CMA MAC and ENC has PQ-IND-CPA (see the full version [28]), $\hat{\$}$ (Algorithm 4) is a secure interactive private quantum money scheme (see Definition 3.3). Moreover, if \$ is semi-quantum, $\hat{\$}$ is also semi-quantum.*

The proof of Theorem 3.9 is given in the full version [28].

## 4 PUTTING IT ALL TOGETHER

For convenience, we restate the main theorem:

**Theorem 1.1** (Main Theorem)**.** *Assuming that the Learning With Errors (LWE) problem with certain parameters is hard for BQP and that a post-quantum existentially unforgeable under an adaptive chosen message attack MAC and an encryption scheme with post-quantum indistinguishability under adaptive chosen plaintext attack (see the full version for the definitions [28]) exist, then a secure semi-quantum private money scheme exists (Definition 3.2).*

Proof. From [5, Theorem 26] (see also the full version [28])we get that the hardness of LWE implies that an NTCF family exists. From Theorem 2.2 we get that an NTCF implies $\frac{1}{2}$-hard 1-of-2 puzzles, and from Corollary 2.10 we get that weak 1-of-2 puzzles (and in particular, $\frac{1}{2}$-hard 1-of-2 puzzles) imply strong 1-of-2 puzzles.

By combining Propositions 3.5, 3.6, 3.7 and 3.8 and Theorem 3.9 (based on the constructions of Algorithm 3 and Algorithm 4), we get that — assuming a PQ-EU-CMA MAC and a PQ-IND-CPA private-key encryption scheme exist — strong 1-of-2 puzzles imply secure semi-quantum private money. □

## 5 DISCUSSION

The main question that is raised in this work is the following. There are many multi-party quantum cryptographic protocols which require that both parties have quantum resources. This work elicits an important question: is there a way (preferably, as general as possible) to convert some of these protocols to ones in which at least one of the parties does not need a quantum computer? A weaker open question can be posed from the perspective of device-independent cryptography: can at least one party use an *untrusted* quantum computer in unison with a trusted classical computer? We emphasize that device independent protocols (see [11, 32] and references therein), such as DI quantum key distribution, DI randomness expansion[9] and randomness amplification, use unconditional (information theoretic) security notions, while our protocols are only computationally secure.

It is known that public quantum money schemes cannot be secure against computationally unbounded adversaries [2], and hence, computational assumptions are necessary for any public scheme. Are computational assumptions also necessary for semi-quantum private money? To the best of our knowledge, a similar question holds for the classical verification of quantum computation, where the only known way to tackle this problem while using a single server uses computational assumptions [21], but it is not clear whether a computational assumption is necessary.

Lastly, can we have *public* semi-quantum money? That is a semi-quantum money scheme, together with a public verification algorithm (which is quantum).

## Acknowledgments

We wish to thank Zvika Brakerski, Urmila Mahadev and Thomas Vidick for fruitful discussions. O.S. and R.R. are supported by the Israel Science Foundation (Grant 682/18), and by a Ben-Gurion University Cyber Security Research Center grant.

## REFERENCES

[1] Scott Aaronson. 2009. Quantum Copy-Protection and Quantum Money. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009.* IEEE Computer Society, 229–242. https://doi.org/10.1109/CCC.2009.42 arXiv:1110.5353

[2] Scott Aaronson and Paul Christiano. 2013. Quantum Money from Hidden Subspaces. *Theory of Computing* 9 (2013), 349–401. https://doi.org/10.4086/toc.2013.v009a009 arXiv:1203.4740

[3] Mihir Bellare, Russell Impagliazzo, and Moni Naor. 1997. Does Parallel Repetition Lower the Error in Computationally Sound Protocols?. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997.* IEEE Computer Society, 374–383. https://doi.org/10.1109/SFCS.1997.646126

[4] Shalev Ben-David and Or Sattath. 2016. Quantum Tokens for Digital Signatures. (2016). arXiv:1609.09047

[5] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. 2018. A Cryptographic Test of Quantumness and Certifiable Randomness from a Single Quantum Device. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, Mikkel Thorup (Ed.). IEEE Computer Society, 320–331. https://doi.org/10.1109/FOCS.2018.00038 arXiv:1804.00640

[6] Anne Broadbent and Christian Schaffner. 2016. Quantum cryptography beyond quantum key distribution. *Des. Codes Cryptography* 78, 1 (2016), 351–382. https://doi.org/10.1007/s10623-015-0157-4 arXiv:1510.06120

[7] Ran Canetti, Shai Halevi, and Michael Steiner. 2005. Hardness Amplification of Weakly Verifiable Puzzles. In *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings (Lecture Notes in Computer Science)*, Joe Kilian (Ed.), Vol. 3378. Springer, 17–33. https://doi.org/10.1007/978-3-540-30576-7_2

[8] Andrea Coladangelo. 2019. Smart Contracts Meet Quantum Cryptography. (2019). arXiv:1902.05214

[9] D. Dieks. 1982. Communication by EPR Devices. *Physics Letters A* 92, 6 (1982), 271 – 272. https://doi.org/10.1016/0375-9601(82)90084-6

[10] Edward Farhi, David Gosset, Avinatan Hassidim, Andrew Lutomirski, and Peter Shor. 2012. Quantum Money from Knots. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference.* ACM, ACM, 276–289. https://doi.org/10.1145/2090236.2090260 arXiv:1004.5127

[11] Rotem Arnon Friedman, Renato Renner, and Thomas Vidick. 2019. Simple and Tight Device-Independent Security Proofs. *SIAM J. Comput.* 48, 1 (2019), 181–225. https://doi.org/10.1137/18M1174726 arXiv:1607.01797

[12] Dmitry Gavinsky. 2012. Quantum Money with Classical Verification. In *Proceedings of the 27th Conference on Computational Complexity, CCC 2012, Porto, Portugal, June 26-29, 2012.* IEEE Computer Society, 42–52. https://doi.org/10.1109/CCC.2012.10 arXiv:1109.0372

[13] Marios Georgiou and Iordanis Kerenidis. 2015. New Constructions for Quantum Money. In *10th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2015, May 20-22, 2015, Brussels, Belgium.* Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 92–110. https://doi.org/10.4230/LIPIcs.TQC.2015.92

[14] Alexandru Gheorghiu and Thomas Vidick. 2019. Computationally-secure and composable remote state preparation. *CoRR* abs/1904.06320 (2019). arXiv:1904.06320 http://arxiv.org/abs/1904.06320

[15] Oded Goldreich. 2004. *The Foundations of Cryptography - Vol. 2, Basic Applications.* Cambridge University Press.

[16] Zhengfeng Ji, Yi-Kai Liu, and Fang Song. 2018. Pseudorandom Quantum States. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III (Lecture Notes in Computer Science)*, Hovav Shacham and Alexandra Boldyreva (Eds.), Vol. 10993. Springer, 126–152. https://doi.org/10.1007/978-3-319-96878-0_5 arXiv:1711.00385

[17] Jonathan Katz and Yehuda Lindell. 2014. *Introduction to Modern Cryptography, Second Edition.* CRC Press.

[9] Also known as certified randomness

[18] Andrew Lutomirski. 2011. Component Mixers and a Hardness Result for Counterfeiting Quantum Money. (2011). arXiv:1107.0321

[19] Andrew Lutomirski, Scott Aaronson, Edward Farhi, David Gosset, Jonathan A. Kelner, Avinatan Hassidim, and Peter W. Shor. 2010. Breaking and Making Quantum Money: Toward a New Quantum Cryptographic Protocol. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, Andrew Chi-Chih Yao (Ed.). Tsinghua University Press, 20–31. arXiv:0912.3825

[20] Urmila Mahadev. 2018. Classical Homomorphic Encryption for Quantum Circuits. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, Mikkel Thorup (Ed.). IEEE Computer Society, 332–338. https://doi.org/10.1109/FOCS.2018.00039 arXiv:1708.02130

[21] Urmila Mahadev. 2018. Classical Verification of Quantum Computations. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, Mikkel Thorup (Ed.). IEEE Computer Society, 259–267. https://doi.org/10.1109/FOCS.2018.00033 arXiv:1804.01082

[22] Abel Molina, Thomas Vidick, and John Watrous. 2013. Optimal Counterfeiting Attacks and Generalizations for Wiesner's Quantum Money. In *Theory of Quantum Computation, Communication, and Cryptography*. Springer, 45–64. arXiv:1202.4010

[23] Michele Mosca and Douglas Stebila. 2010. *Quantum Coins*. Contemp. Math., Vol. 523. Amer. Math. Soc., 35–47. https://doi.org/10.1090/conm/523/10311 arXiv:0911.1295

[24] Michael A. Nielsen and Isaac L. Chuang. 2011. *Quantum Computation and Quantum Information: 10th Anniversary Edition* (10th ed.). Cambridge University Press, New York, NY, USA.

[25] James L. Park. 1970. The Concept of Transition in Quantum Mechanics. *Foundations of Physics* 1, 1 (01 Mar 1970), 23–33. https://doi.org/10.1007/BF00708652

[26] Fernando Pastawski, Norman Y. Yao, Liang Jiang, Mikhail D. Lukin, and J. Ignacio Cirac. 2012. Unforgeable Noise-Tolerant Quantum Tokens. *Proceedings of the National Academy of Sciences* 109, 40 (2012), 16079–16082. https://doi.org/10.1073/pnas.1203552109 arXiv:1112.5456

[27] Marta Conde Pena, Raul Durán Díaz, Jean-Charles Faugère, Luis Hernández Encinas, and Ludovic Perret. 2018. Non-Quantum Cryptanalysis of the Noisy Version of Aaronson–Christiano's Quantum Money Scheme. (December 2018). https://doi.org/10.1049/iet-ifs.2018.5307

[28] Roy Radian and Or Sattath. 2019. Semi-Quantum Money. (2019). arXiv:1908.08889

[29] Ran Raz. 2011. A Counterexample to Strong Parallel Repetition. *SIAM J. Comput.* 40, 3 (2011), 771–777. https://doi.org/10.1137/090747270

[30] Victor Shoup. 2004. Sequences of Games: a Tool for Taming Complexity in Security Proofs. Cryptology ePrint Archive, Report 2004/332. (2004). https://eprint.iacr.org/2004/332.

[31] Yuuki Tokunaga, Taisuaki Okamoto, and Nobuyuki Imoto. 2003. Anonymous Quantum Cash. (2003). http://qci.is.s.u-tokyo.ac.jp/qci/eqis03/program/papers/O09-Tokunaga.ps.gz

[32] Umesh V. Vazirani and Thomas Vidick. 2019. Fully Device Independent Quantum Key Distribution. *Commun. ACM* 62, 4 (2019), 133. https://doi.org/10.1145/3310974 arXiv:1210.1810

[33] Stephen Wiesner. 1983. Conjugate Coding. *ACM Sigact News* 15, 1 (1983), 78–88. https://doi.org/10.1145/1008908.1008920

[34] William K Wootters and Wojciech H Zurek. 1982. A Single Quantum Cannot be Cloned. *Nature* 299, 5886 (1982), 802–803.

[35] Mark Zhandry. 2019. Quantum Lightning Never Strikes the Same State Twice. In *Advances in Cryptology - EUROCRYPT 2019 - Germany, May 19-23, 2019, Proceedings, Part III (Lecture Notes in Computer Science)*, Yuval Ishai and Vincent Rijmen (Eds.), Vol. 11478. Springer, 408–438. https://doi.org/10.1007/978-3-030-17659-4_14 arXiv:1711.02276