Quantum Bitcoin: An Anonymous, Distributed, and Secure Currency Secured by the No-Cloning Theorem of Quantum Mechanics

Jonathan Jogenfors
Institutionen för Systemteknik, Linköpings Universitet, 581 83 Linköping, Sweden.
Email: jonathan.jogenfors@liu.se

Abstract—**The digital currency Bitcoin has had remarkable growth since it was first proposed in 2008. Its distributed nature allows currency transactions without a central authority by using cryptographic methods and a data structure called the blockchain. Imagine that you could run the Bitcoin protocol on a quantum computer. What advantages can be had over classical Bitcoin? This is the question we answer here by introducing Quantum Bitcoin which, among other features, has immediate local verification of transactions. This is a major improvement over classical Bitcoin since we no longer need the computationally-intensive and time-consuming method of recording all transactions in the blockchain. Quantum Bitcoin is the first distributed quantum currency, and this paper introduces the necessary tools including a novel two-stage quantum mining process. In addition, we have counterfeiting resistance, fully anonymous and free transactions, and a smaller footprint than classical Bitcoin.**

In this paper we construct Quantum Bitcoin, a novel currency that draws its inspiration from both Bitcoin and quantum mechanics. As we later show, our new currency has several interesting advantages over existing payment systems, although the drawback is that it requires a quantum computer to function.

I. Quantum Money

As early as around 1970, Wiesner [1] and Broadbent and Schaffner [2] proposed a scheme that uses the no-cloning theorem [3] to produce unforgeable quantum banknotes. According to Aaronson and Christiano [4] it took 13 years until the paper was finally published [1] in 1983. In the same year, BBBW [5] made improvements to Wiesner's scheme, such as an efficient way to keep track of every banknote in circulation. Another, more recent, extension by Pastawski et al. [6] increases the tolerance against noise. Even more recently, Brodutch et al. [7] presented an attack on the Wiesner and BBBW schemes.

After BBBW, quantum money received less attention due to the seminal 1984 paper by Bennett and Brassard [8] that created the separate field of quantum key distribution (QKD). Following two decades where virtually no work was done on quantum money, Mosca and Stebila [9] proposed *quantum coins* ten years ago. In contrast to quantum banknotes (where each banknote is unique), quantum coins are all identical.

We distinguish between *private key* and *public-key* quantum money systems. In a private-key system, only the bank that minted the quantum money can verify it as genuine, while a public-key system allows anyone to perform this verification. The advantage of a public-key system over a private-key are obvious as long as it is just as secure. Until recently, all quantum money proposals were private-key, however in 2009 Aaronson [10] proposed the first public-key quantum money system. While this first public-key system was broken in a short time by Lutomirski et al. [11], it inspired others to re-establish security. A novel proposal by Farhi et al. [12] produced a public-key system using knot theory and superpositions of link diagrams, and this idea was further developed by Lutomirski [13].

In 2012, public-key quantum money based on hiding subspaces was introduced by Aaronson and Christiano [4]. The contribution consists first of an implicit scheme based on random oracles and then an explicit version based on multivariate polynomials. Further work by Pena et al. [14] showed that this explicit scheme is insecure in the noiseless case, however the security of the noisy scheme remains an open question.

Common to all proposals discussed above is a centralized topology, with a number of users and one "bank" that issues (and possibly verifies) money. This requires all users to fully trust this bank, as a malevolent bank can perform fraud and revoke existing currency. This is true for both private-key and public-key schemes.

II. Cryptography and the Blockchain

Our scheme requires a digital signature scheme, which we model as follows:

Definition 1 *A classical public-key digital signature scheme $\mathcal{D}$ consists of three probabilistic polynomial-time classical algorithms [4]:*

1) $\mathsf{KeyGen}_{\mathcal{D}}$ *which takes as input a security parameter n and randomly generates a key pair $(k_{private}, k_{public})$.*
2) $\mathsf{Sign}_{\mathcal{D}}$ *which takes as input a private key $k_{private}$ and a message M and generates a (possibly randomized) signature $\mathsf{Sign}_{\mathcal{D}}(k_{private}, M)$.*
3) $\mathsf{Verify}_{\mathcal{D}}$*, which takes as input k_{public}, a message M, and a claimed signature ω, and either accepts or rejects.*

Next, we model the blockchain as an random access ordered array with timestamped dictionary entries. Blocks can be added to the end of the chain by solving a proof-of-work puzzle, and blocks in the chain can be read using a lookup function. In Quantum Bitcoin, blocks do not contain standard transactions, only classical descriptors of the newly minted quantum bitcoin (and shards, discussed later). Therefore, blocks in the quantum bitcoin blockchain should be seen as general-purpose dictionaries that match serial numbers s to public keys[1] k_{public}. We will use the following abstract definition:

Definition 2 *A classical distributed ledger scheme $\mathcal{L}$ consists of the following classical algorithms:*

- $\mathsf{Append}_{\mathcal{L}}$ *is an algorithm which takes (s, k_{public}) as input, where s is a classical serial number and k_{public} a classical public key. The algorithm fails if the serial number already exists in a block in the ledger. Otherwise, it begins to solve a proof-of-work puzzle by repeated trials of random nonce values. The algorithm succeeds if the puzzle is solved, at which time the ledger pair (s, k_{public}) is added as a new block.*
- $\mathsf{Lookup}_{\mathcal{L}}$ *is a polynomial-time algorithm that takes as input a serial number s and outputs the corresponding public key k_{public} if it is found in the ledger. Otherwise, the algorithm fails.*

Our formal definition is independent of the underlying block format and ruleset, so any secure blockchain implementation can be used. Note that $\mathsf{Append}_{\mathcal{L}}$ runs continuously until it succeeds — if another miner solves a proof-of-work puzzle it simply restarts the process transparently to the caller.

III. Quantum Bitcoin

Our contribution is a scheme that turns any centralized public-key quantum money scheme (such as Aaronson and Christiano [4] and Farhi et al. [12]) into a distributed one. In our trust model we do not have to assume every single miner to be trustworthy. Instead, the system works as long as a certain percentage of miners remains honest.

As with most quantum money schemes the central idea is the no-cloning theorem [3] which shows that it is impossible to copy an unknown quantum state $|\psi\rangle$. Quantum mechanics therefore provides an excellent basis on which to build a currency, as copy-protection is "built in". In section IV we quantify the level of security the no-cloning theorem gives, and show that the Quantum Bitcoin scheme is secure against counterfeiting.

We will use a quantum state as the unit of currency and endow it with classical information to facilitate verification. We then add a blockchain data structure which allows us to combine the quantum state with a distributed minting process, relinquishing trust in the central bank normally required by the security model in traditional quantum currencies.

In order to simplify the security analysis, we construct a small "mini-scheme" [4, 12] which is then generalized to a full distributed Quantum Bitcoin scheme. The mini-scheme $\mathcal{M}$ does not use a blockchain and can only mint and verify one single quantum bitcoin. We then combine $\mathcal{M}$ with a digital signature scheme $\mathcal{D}$ and a distributed ledger scheme $\mathcal{L}$ to get the full-fledged Quantum Bitcoin scheme. In section IV we will see that proving the security of $\mathcal{M}$ and $\mathcal{D}$ implies the security of the complete system $\mathcal{Q}$.

Intuitively we want the Quantum Bitcoin scheme $\mathcal{Q}$ to have the following properties:

1) a controlled method to feasibly generate an unlimited number of quantum bitcoin no faster than a given rate,
2) a verification procedure, and
3) protection against forgery, so that the only way to map a polynomial number of quantum bitcoin to a larger number of quantum bitcoin with non-negligible success probability is the aforementioned generation process.

Here, we call a function $f(n)$ *negligible* if $f(n) = o(1/p(n))$ for every polynomial $p(n)$.

A. The Hidden Subspace Mini-Scheme

We adopt the Hidden Subspace mini-scheme system introduced by Aaronson and Christiano [4]. Let $\mathbb{F}_2^n$ be the space of binary sequences of length n, equipped with the standard inner product

$$\langle u, v \rangle = \sum (u)_i (v)_i \pmod 2. \tag{1}$$

We define A as a (secretly chosen) $n/2$-dimensional subspace of $\mathbb{F}_2^n$. In addition, let $A^\perp$ be the $n/2$-dimensional orthogonal complement to A, that is, the set of $y \in \mathbb{F}_2^n$ such that $\langle x, y \rangle = 0$ for all $x \in A$.

In the Mini-Scheme we create *quantum bitcoin states* $|A\rangle$ in the following way:

$$|A\rangle = \frac{1}{\sqrt{|A|}} \sum_{x \in A} |x\rangle . \tag{2}$$

It is easy to prepare $\langle A \rangle$, a classical description of A which consists of $n/2$ generators. From this classical description one can create the quantum state in equation (2). Next, we define a public membership oracle U_A that is used to decide membership in A:

$$U_A |x\rangle = \begin{cases} -|x\rangle & \text{if } x \in A \\ |x\rangle & \text{otherwise,} \end{cases} \tag{3}$$

The membership oracle allows anybody to decide if a given, alleged quantum bitcoin state corresponds to the subspace A. Note the parallel to classical digital signatures.

The unitary gate U_A is assumed to be a random oracle, but explicit methods are available, such as the (noisy)

[1] Do not confuse the public key k_{public} with the key of the dictionary.

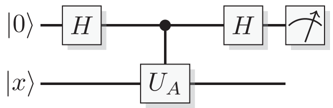

Figure 1: Quantum circuit $\mathbb{P}_A$.

multivariate polynomial scheme by Aaronson and Christiano [4]. Note that the noiseless version of the multivariate polynomial scheme was broken by Pena et al. [14] in 2015. Using U_A, we can build a quantum circuit $\mathbb{P}_A$ that projects onto the basis states of $|A\rangle$ (figure 1).

1) Initiate a control qubit $|0\rangle$
2) Apply the Hadamard transform H to the control qubit
3) Apply U_A to $|x\rangle$ conditioned on the control qubit being in state $|1\rangle$
4) Apply H to the control qubit
5) Measure the control qubit and postselect on the outcome $|1\rangle$.

Note that the above algorithm maps $|x\rangle$ to $|1\rangle|x\rangle$ if $x \in A$ and $|0\rangle|x\rangle$ otherwise. Therefore, when measurement and postselection is performed, the algorithm returns 0 if and only if $|x\rangle \notin A$ and $|1\rangle|x\rangle$ otherwise.

We define $U_{A^\perp}$ and $\mathbb{P}_{A^\perp}$ in a similar way as above, except we instead operate on $A^\perp$. Together with the projectors $\mathbb{P}_A$ and $\mathbb{P}_{A^\perp}$ we can create a unitary operator

$$V_A = H_2^{\otimes n}\mathbb{P}_{A^\perp} H_2^{\otimes n}\mathbb{P}_A, \tag{4}$$

where H again denotes the Hadamard transform. We will use V_A to verify quantum bitcoin states, where we interpret $V_A|\psi\rangle = |A\rangle$ as passing and $V_A|\psi\rangle = 0$ as failing. Aaronson and Christiano [4, p. 379] show that V_A is a projector onto A, and that V_A accepts an arbitrary state $|\psi\rangle$ with probability $|\langle\psi|A\rangle|^2$.

Recall that a mini-scheme only mints and verifies one single quantum bitcoin. We can now give the formal definition:

Definition 3 *The Hidden Subspace mini-scheme $\mathcal{M}$ consist of two polynomial-time algorithms $\mathsf{Mint}_\mathcal{M}$ and $\mathsf{Verify}_\mathcal{M}$.*

Next, we set the requirements for the algorithm that generates the secret subspace A_r of $\mathbb{F}_2^n$.

Definition 4 *A **Subspace Generator** $\mathcal{G}(r)$ takes a secret n-bit string r and returns $(s_r, \langle A_r\rangle)$, where s_r is a $3n$-bit string and $\langle A_r\rangle$ is a set of linearly independent secret generators $\{x_1, \ldots, x_{n/2}\}$ for a subspace $A_r \leq \mathbb{F}_2^n$. We require that the serial numbers are distinct for every r.*

The subspace generator is the first step in minting a quantum bitcoin as it generates the secret, random subspace necessary for state generation (A_r) as well as the public serial number s_r.

Definition 5 *A **Serial Number Verifier** $\mathcal{H}(s)$ takes a serial number s and passes if it is a valid serial number $s = s_r$ for some $\langle A_r\rangle$ and fails otherwise.*

Again, we remind the reader that the above definitions are abstract random oracles, however explicit constructions are available. We now complete the mini-scheme with minting and verification algorithms:

Definition 6 $\mathsf{Mint}_\mathcal{M}(n)$ *takes as input a security parameter n. It then randomly generates a secret n-bit key r which it passes to the Subspace Generator $\mathcal{G}(r)$ which returns the serial number s_r and a classical description $\langle A_r\rangle$ of the subspace A. Next, it prepares the quantum state $|A_r\rangle$ given in equation (2). The returned value is $(s_r, |A_r\rangle)$.*

Definition 7 $\mathsf{Verify}_\mathcal{M}(¢)$ *takes as input an alleged quantum bitcoin $¢$ and performs the following checks, in order:*

1) *Form check: Accept if and only if $¢$ has the form (s, ρ), where s is a classical serial number and ρ is a quantum state.*
2) *Serial number check: Accept if and only if the Serial Number Verifier $\mathcal{H}(s)$ accepts*
3) *Apply $V_{A_r} = H_2^{\otimes 2}\mathbb{P}_{A_r^\perp} H_2^{\otimes 2}\mathbb{P}_{A_r}$ to ρ and accept if and only if $V_{A_r}(\rho) \neq 0$*

Note that the verification procedure immediately terminates if any of the above steps fail.

B. The Standard Construction of Quantum Bitcoin

The mini-scheme $\mathcal{M}$ can only mint and verify one single quantum bitcoin, so to build a usable currency we need to extend the model with a mechanism for minting *any* amount of currency. For this purpose we will define the full Quantum Bitcoin scheme, $\mathcal{Q}$, and construct it as an extension of the mini-scheme $\mathcal{M}$. The connection between $\mathcal{M}$ and $\mathcal{Q}$ is derived from the "standard construction" by Aaronson and Christiano [4], Lutomirski et al. [11], and Farhi et al. [12].

Definition 8 *A public-key distributed Quantum Bitcoin scheme $\mathcal{Q}$ consists of the following algorithms:*

- $\mathsf{KeyGen}_\mathcal{Q}$*, a polynomial-time algorithm which takes as input a security parameter n and randomly generates a key pair $(k_{private}, k_{public})$.*
- $\mathsf{Mint}_\mathcal{Q}$ *which takes a security parameter n and produces a quantum bitcoin \$.*
- $\mathsf{Verify}_\mathcal{Q}$*, a polynomial-time algorithm which takes as input an alleged quantum bitcoin $¢$ and a corresponding public key k_{public} and either accepts or rejects.*

Given a mini-scheme $\mathcal{M} = (\mathsf{Mint}_\mathcal{M}, \mathsf{Verify}_\mathcal{M})$, a digital signature scheme $\mathcal{D} = (\mathsf{KeyGen}_\mathcal{D}, \mathsf{Sign}_\mathcal{D}, \mathsf{Verify}_\mathcal{D})$ and a distributed ledger scheme $\mathcal{D} = (\mathsf{KeyGen}_\mathcal{L}, \mathsf{Append}_\mathcal{L}, \mathsf{Lookup}_\mathcal{L})$, we will construct an initial, naive, version of the Quantum Bitcoin scheme $\mathcal{S} = (\mathsf{KeyGen}_\mathcal{Q}, \mathsf{Sign}_\mathcal{Q}, \mathsf{Verify}_\mathcal{Q})$. Later, we extend this standard construction to protect against a special type of attack.

First, $\mathsf{KeyGen}_\mathcal{Q}$ is $\mathsf{KeyGen}_\mathcal{D}$ from the digital signature scheme. Next, we define the algorithm for $\mathsf{Verify}_\mathcal{Q}$ for an alleged Quantum Bitcoin $¢$:

1) Check that $¢$ is on the form (s, ρ, σ), where the s is a classical serial number, ρ a quantum state, and σ a classical digital signature.
2) Call $\mathsf{Lookup}_{\mathcal{L}}(s)$ to retrieve the public key k_{public} associated with the serial number s.
3) Call $\mathsf{Verify}_{\mathcal{D}}(k_{public}, s, \sigma)$ to verify the digital signature of the quantum bitcoin.
4) Call $\mathsf{Verify}_{\mathcal{M}}(s, \rho)$ from the mini-scheme.

Here we identify the first major advantage of Quantum Bitcoin. Whereas classical Bitcoin requires each transaction to be recorded into the blockchain (a time-consuming process), Quantum Bitcoin transactions finalize *immediately*. Due to the no-cloning theorem of quantum mechanics, the underlying quantum state in a quantum bitcoin cannot be duplicated, thereby preventing counterfeiting in itself (see section IV). The only step of the protocol that uses $\mathsf{Append}_{\mathcal{L}}$ is minting, which end users never have to worry about.

The main challenge of constructing the Quantum Bitcoin scheme is that the miners are *untrusted*, in contrast to previous schemes where minting is done by a *trusted* bank. The strength of our scheme is that it allows *any* public-key quantum money scheme to adopt a Bitcoin-like trust model. That is, we take individually untrusted miners and still be able to trust them as a group [15]. Our first, naive, attempt at a $\mathsf{Mint}_{\mathcal{Q}}$ algorithm is the following:

1) Call $\mathsf{KeyGen}_{\mathsf{D}}$ to randomly generate a key pair $(k_{private}, k_{public})$.
2) Generate a quantum bitcoin candidate by calling $\mathsf{Mint}_{\mathcal{M}}$, which returns (s, ρ), where s is a classical serial number and ρ is a quantum state.
3) Sign the serial number: $\sigma = \mathsf{Sign}_{\mathcal{D}}(k_{private}, s)$
4) Call $\mathsf{Append}_{\mathcal{L}}(s, k_{public})$ to attempt to append the serial number s and the public key k_{public} to the ledger.
5) If $\mathsf{Append}_{\mathcal{L}}$ failed, start again from step 2
6) If the serial number was successfully appended, put the serial number, quantum state and signature together to create the quantum bitcoin $\$ = (s, \rho, \sigma)$.

Note that the "difficult step" in the minting algorithm is the call to $\mathsf{Append}_{\mathcal{L}}$.

C. Preventing Quantum Double-Mining

At first glance, the above algorithms appear to work, however there is a problem. As previously mentioned, there is no trust in the individual minters. This leads to a phenomena we call **quantum double-mining**. Ideally, $\mathsf{Mint}_{\mathcal{M}}$ should be an algorithm that generates unique quantum states every time it is called, similarly to a random oracle. However, a malicious miner could generate a quantum bitcoin, append it to the blockchain, and then covertly reuse $k_{private}$ to produce any number of identical quantum bitcoin. The severity of this cannot be understated, as it opens up the possibility for a single miner to undermine the entire payment system! Note that this effect is fundamentally different to counterfeiting resistance (see section IV) as double-mining only can be performed by an entity who has the private key.

Compare with classical Bitcoin. There, the blockchain records all transactions and a miner therefore relinquishes control over the mined bitcoin as soon as it is handed over to the next owner. In Quantum Bitcoin, however, there is no record of who owns what, so nobody would notice doubly-mined "Trojan horse" quantum bitcoin.

The incentives for the double-mining attack are huge. Reusing the private key allows a malicious to duplicate the quantum bitcoin they just created, leading to an easy double-spend. Another, more sinister, strategy is to mine a large number of quantum bitcoin, wait until these have circulated to other users, and then create a large amount of duplicate coins. Surely, such an attack would have detrimental effects on the currency as a whole.

The double-mining attack is prevented in a similar way as the double-spending problem is prevented in classical Bitcoin. However, while classical Bitcoin has to store *all* transactions in the blockchain, Quantum Bitcoin only records *minting*! This tremendously reduces the overhead of the protocol, and in section V we show that the countermeasure is effective.

The countermeasure is a secondary stage to the minting process, where data is appended to a new ledger $\mathcal{L}'$. For the secondary mining step, we introduce fixed protocol-level security parameters $m \geq 1$ and $T_{\max} > 0$ and the algorithm is as follows:

1) A miner (this time called a **quantum shard miner**) uses the above "naive" minting scheme, but the finished product (s, ρ, σ) is instead called a **quantum shard**, or simply **shard**[2].
2) Shard miners sell the shards on a marketplace.
3) A miner (called a **quantum bitcoin miner**) purchases m shards $\{(s, \rho_i, \sigma_i)\}_{1 \leq i \leq m}$ on the marketplace that, for all $1 \leq i \leq m$, meet the following conditions:
   - $\mathsf{Verify}_{\mathcal{Q}}((s, \rho_i, \sigma_i))$ accepts
   - The timestamp T of the shard in the shard ledger $\mathcal{L}$ fulfills $t - T \leq T_{\max}$, where t is the current time.
4) The quantum bitcoin miner calls $\mathsf{KeyGen}_{\mathcal{Q}}$ to randomly generate a key pair $(k_{private}, k_{public})$.
5) The quantum bitcoin miner takes the serial numbers of the m shards and compiles the **classical descriptor** $s = (s_1, \ldots, s_m)$ and signs it as $\sigma_0 = \mathsf{Sign}_{\mathcal{D}}(k_{private}, s)$.
6) The quantum bitcoin miner takes the m shards and, together with σ_0, produces a **quantum bitcoin candidate**: $(s_1, \rho_1, \sigma_1, \ldots, s_m, \rho_m, \sigma_m, \sigma_0)$.
7) The quantum bitcoin miner calls $\mathsf{Append}_{\mathcal{L}'}(s, k_{public})$ to attempt to pair the quantum bitcoin miner's public key k_{public} with the classical descriptor s in the ledger. Here, we require that Append fails if any of the m

[2] Do not confuse a quantum shard with the concept of "blockchain sharding".

shards already have been combined into a quantum bitcoin that exists in the ledger $\mathcal{L}'$.

This process is the complete quantum mining protocol, and it works because each participant is incentivized: shard miners invest computing power to produce shards, which quantum bitcoin miners want for quantum bitcoin production. As there is only a finite number of shards, they will have monetary value, thus rewarding the shard miners. In turn, quantum bitcoin miners invest computing power to mint quantum bitcoin from shards. The quantum bitcoin miners are rewarded with a valid quantum bitcoin, which, due to their limited supply, can be expected to have monetary value.

According to Nakamoto [15], such incentives "may help nodes to stay honest", and an attacker who has access to more computing power than the rest of the network combined finds that it is more rewarding to play by the rules than to commit fraud. Also, quantum double-mining is prevented because two-stage mining makes it overwhelmingly difficult for a single entity to first produce m shards, and secondly, combine them to a quantum bitcoin.

This construction assumes that a majority of miners are honest, i.e. discard private keys after mining (for details see section V). Note that shards, if not combined into a quantum bitcoin, expire after a finite time $T_{\max}$. This is necessary because otherwise the probability of successfully mining a single shard approaches 1 over time. Therefore, given enough time, a malicious miner can produce m valid shards which then can be combined into a valid quantum bitcoin. The expiry time $T_{\max}$ prevents this. An attacker must therefore compete against the rest of the network in manner similar to Bitcoin. Note that the selection of $T_{\max}$ is complicated by the possibility of blockchain forks.

What remains is to slightly modify $\mathsf{Verify}_{\mathcal{Q}}$ to take two-stage mining into account:

1) Check that $\not{c}$ is of the form $(s_1, \rho_1, \sigma_1, \ldots, s_m, \rho_m, \sigma_m, \sigma_0)$, where the s_i are classical serial numbers, ρ_i are quantum states, and σ_i (including σ_0) are digital signatures.
2) Call $\mathsf{Lookup}_{\mathcal{L}'}((s_1, \ldots, s_m))$ to retrieve the public key k_{public} of the quantum bitcoin miner associated with the classical descriptor $(s_1, \ldots, s_m)$.
3) Call $\mathsf{Verify}_{\mathcal{D}}(k_{public}, (s_1, \ldots, s_m), \sigma_0)$ to verify the digital signature of the quantum bitcoin.
4) For each $1 \leq i \leq m$, call $\mathsf{Lookup}_{\mathcal{L}}(s_i)$ in order to retrieve the corresponding public keys $k_{public,i}$ from the shard miners.
5) For each $1 \leq i \leq m$, call $\mathsf{Verify}_{\mathcal{D}}(k_{public,i}, s_i, \sigma_i)$ to verify the digital signature of each of the shards.
6) For $1 \leq i \leq m$, call $\mathsf{Verify}_{\mathcal{M}}(s_i, \rho_i)$.

The verification passes if and only if all of the above steps succeed. This method checks the digital signatures of both the quantum bitcoin and all its shards before calling the verification procedure of the mini-scheme $\mathcal{M}$.

IV. Counterfeiting resistance

We now show that our scheme resists counterfeiting by showing that (except for the minter itself), *no* quantum circuit can produce counterfeit quantum bitcoin with success probability greater than an arbitrarily small number.

Definition 9 *A* ***counterfeiter*** *C is a quantum circuit of polynomial size (in n) which maps a polynomial (in n) number of valid quantum bitcoin to a polynomial number (in n) of new, possibly entangled alleged quantum bitcoin.*

Next, we quantify the probability of a counterfeit quantum bitcoin to be accepted by the verification procedure. This is the probability of a false positive:

Definition 10 *A Quantum Bitcoin scheme $\mathcal{Q}$ has* ***soundness error*** *δ if, given any counterfeiter C and a collection of q valid quantum bitcoin $\$_1, \ldots, \$_q$, we have*

$$\Pr(\mathsf{Count}(C(\$_1, \ldots, \$_q)) > q) \leq \delta, \tag{5}$$

where Count *is a* ***counter*** *that takes as input a collection of (possibly-entangled) alleged quantum bitcoin $\$_1, \ldots, \$_r$ and outpts the number of indices $0 \leq i \leq r$ such that* $\mathsf{Verify}(\$_i)$ *accepts.*

Conversely, we quantify the probability of false negative, i.e. the probability that a valid quantum bitcoin is rejected by the verification procedure:

Definition 11 *A Quantum Bitcoin scheme $\mathcal{Q}$ has* ***completeness error*** *ε if* $\mathsf{Verify}(\$)$ *accepts with probability at least $1 - \varepsilon$ for all valid quantum bitcoin \$. If $\varepsilon = 0$ then $\mathcal{Q}$ has* ***perfect completeness****.*

We call a Quantum Bitcoin scheme $\mathcal{Q}$ **secure** if it has completeness error $\varepsilon \leq 1/3$ and negligible soundness error. Next, we continue with analyzing the mini-scheme. Recall that a mini-scheme only mints and verifies one single quantum bitcoin, so that a mini-scheme counterfeiter only takes the single valid quantum bitcoin as input. To perform this analysis, we need a technical tool, the double verifier:

Definition 12 *For a mini-scheme $\mathcal{M}$, we define the* ***double verifier*** Verify_2 *as a polynomial-time algorithm that takes as input a single serial number s and two (possibly-entangled) quantum states ρ_1 and ρ_2 and accepts if and only if* $\mathsf{Verify}_{\mathcal{M}}(s, \sigma_1)$ *and* $\mathsf{Verify}_{\mathcal{M}}(s, \sigma_2)$ *both accept.*

Now, we define the soundness and completeness error for the mini-scheme:

Definition 13 *A mini-scheme $\mathcal{M}$ has* ***soundness error*** *δ if, given any quantum circuit C (the* ***counterfeiter****),* $\mathsf{Verify}_2(s, C(\$))$ *accepts with probability at most δ. Here, the probability is over the quantum bitcoin \$ output by* $\mathsf{Mint}_{\mathcal{M}}$ *as well as the behavior of* Verify_2 *and C.*

Definition 14 *A mini-scheme $\mathcal{M}$ has* ***completeness error*** *ε if* $\mathsf{Verify}(\$)$ *accepts with probability at least $1 - \varepsilon$ for all valid quantum bitcoin or shards \$. If $\varepsilon = 0$ then $\mathcal{Q}$ has* ***perfect completeness****.*

As for the Quantum Bitcoin scheme $\mathcal{Q}$, we call a mini-scheme $\mathcal{M}$ **secure** if it has completeness error $\varepsilon \leq 1/3$ and negligible soundness error. While $1/3$ sounds like a high error probability, Aaronson and Christiano [4, pp. 42–43] show that the completeness error ε of a secure scheme can be made exponentially small in n at the cost of only a modest increase in the soundness error δ.

What remains is to show that the scheme $\mathcal{Q}$ is, in fact, secure. The mini-scheme construction allows this to be performed by a simple security reduction. The following theorem is adapted from Aaronson and Christiano [4, p. 20]:

Theorem 1 *If there exists a secure mini-scheme $\mathcal{M}$, then there also exists a secure Quantum Bitcoin scheme $\mathcal{Q}$.*

Proof *We use the Subspace Generator $\mathcal{G}(r)$ from definition 4 as a one-way function: given a n-bit string r, $\mathcal{G}(r)$ outputs (among others) an unique $3n$-bit serial number s_r. If there exists a polynomial-time quantum algorithm to recover r from s_r it would be possible for a counterfeiter to copy Quantum Bitcoin, which contradicts the security of the mini-scheme. Therefore, $\mathcal{G}(r)$ is a one-way function secure against quantum attack. Since such one-way functions are necessary and sufficient for secure digital signature schemes [16], we immediately get a digital signature scheme $\mathcal{D}$ secure against quantum chosen-plaintext attacks. The final step is to show that $\mathcal{M}$ and $\mathcal{D}$ together produce a secure Quantum Bitcoin scheme $\mathcal{Q}$, which is done in Aaronson and Christiano [4, p. 20].*

This is the elegance of the mini-scheme model: a secure mini-scheme immediately gives us the full, secure scheme. Therefore, a counterfeiter who wants to break $\mathcal{Q}$ is forced to break the security of $\mathcal{M}$. As such, if we show that $\mathcal{M}$ is secure we are done. We have the following:

Theorem 2 *The mini-scheme $\mathcal{M} = (\mathsf{Mint}_{\mathcal{M}}, \mathsf{Verify}_{\mathcal{M}})$, which is defined relative to the classical oracle U, has zero completeness and $1/\exp(n)$ soundness error.*

Proof *The Inner-Product Adversary Method by Aaronson and Christiano [4, p. 31] gives an upper bound to the information gained by a single oracle query. Theorem 1 then shows that the mini-scheme $\mathcal{M}$ is secure since a valid quantum bitcoin always passes verification (zero completeness error), and counterfeit quantum bitcoin pass with only an exponentially small probability (negligible soundness error).*

This ties together the security of the mini-scheme $\mathcal{M}$ and the Quantum Bitcoin scheme $\mathcal{Q}$. Theorem 2 shows that $\mathcal{M}$ is secure, and from theorem 1 it then follows that $\mathcal{Q}$ is secure. Explicitly, any counterfeiter must make $\Omega(2^{n/4})$ queries to successfully copy a quantum bitcoin. For large enough n, this is computationally infeasible. Specifically, $n = 512$ requires at least 2^{128} oracle queries which gives us complexity-theoretic security. Note that Farhi et al. [12] showed that public-key quantum money cannot be information-theoretically secure.

V. Double-Mining Resistance

Now we analyze the effect of the security parameters m and $T_{\max}$ on the probability of quantum double-mining. Recall that quantum double-mining is when the same entity covertly mines a number of quantum shards and then combines them into a quantum bitcoin. The security parameter m controls the number of quantum shards required per quantum bitcoin, and $T_{\max}$ is the maximum age of the quantum shards.

In order to perform quantum double-mining, the attacker must therefore mine $m-1$ shards within $T_{\max}$. In reality the attacker must both mint shards and combine them into quantum bitcoin before $T_{\max}$ runs out. We simplify the analysis, however, by making it easier for the attacker and allow $T_{\max}$ time to mine shards, and then again $T_{\max}$ time to mine quantum bitcoin.

We model our attack by assigning the probability p to the probability of an attacker mining the next block in either the shard or quantum bitcoin blockchain. p can be understood as the proportion of the world's computing power controlled by the attacker. We define $k := \lfloor T_{\max}/T_{block} \rfloor \geq 2$ as the average number of blocks mined before $T_{\max}$ runs out, where T_{block} is the average time between mined blocks. Bitcoin uses $T_{block} = 600\,\mathrm{s}$, although empirical research by Karame et al. [17] suggests that the distribution of mining times corresponds to a shifted geometric distribution with parameter 0.19. The probability of the attacker mining at least $m-1$ of these k shards is then

$$\eta_1 = \binom{k}{m-1} p^{m-1} (1-p)^{k-m+1}. \tag{6}$$

Next, the attacker must combine these shards into a quantum bitcoin before another $T_{\max}$ runs out. The probability for this is the probability of mining a single block:

$$\eta_2 = \binom{k}{1} p (1-p)^{k-1} = kp(1-p)^{k-1}. \tag{7}$$

The total probability of quantum double-mining η is then

$$\eta = \eta_1 \eta_2 = \binom{k}{m-1} k \left(\frac{p}{1-p}\right)^m (1-p)^{2k}. \tag{8}$$

We bound the binomial coefficient by above using the formula

$$\binom{n}{k} < \left(\frac{ne}{k}\right)^k \quad \text{for } 1 \leq k \leq n, \tag{9}$$

which gives

$$\eta < \left(\frac{ke}{m-1}\right)^{m-1} k \left(\frac{p}{1-p}\right)^m (1-p)^{2k} \tag{10}$$

for $2 \leq m \leq k+1$. We set $m-1 = \gamma k$ which for $1/k < \gamma < 1$ gives

$$\eta < k \left(\frac{e}{\gamma} \cdot \frac{p}{1-p}\right)^{\gamma k} \left(\frac{p}{1-p}\right) (1-p)^{2k}. \tag{11}$$

We note that

$$\frac{e}{\gamma} \cdot \frac{p}{1-p} < \frac{1}{2} \Leftrightarrow 0 \leq p < \frac{\gamma}{2e+\gamma}, \tag{12}$$

where the upper bound of p approaches $1/(2e+1) \approx 15.5\,\%$ as γ approaches 1. Under those constraints we get $\sup p/(1-p) = 1/2e$ and $(1-p)^{2k} \leq 1$. Plugging in all this in equation (11) we get the following worst-case upper limit for the double-mining probability:

$$\eta < \frac{k}{2e} 2^{-\gamma k}. \tag{13}$$

In other words, the probability of quantum double-mining is exponentially small in k as long as the attacker controls less than 15 % of the computing power. Note that equation (13) is the worst-case approximation and we should expect a much lower probability in a real scenario. What remains is to qualitatively determine the parameter γ. Too large, and it will be difficult for *any* quantum bitcoin to be mined as every single shard must be sold to a quantum bitcoin miner before $T_{\max}$ runs out. Too small, and the bound in equation (13) is weakened, making it easier for a malicious miner to perform double-mining. The smaller we make γ, the larger we must make k to achieve the required security.

VI. Quantum vs. Classical Bitcoin

We will now compare Quantum Bitcoin to the classical Bitcoin protocol by Nakamoto [15] and show that in comparison, the Quantum Bitcoin scheme has several advantages. Bitcoin transactions must be verified by third-party miners which, on average, takes 60 minutes[3] but has considerable variance [17]. Bitcoin transactions are therefore slow. In contrast, Quantum Bitcoin transactions are immediate and (in addition to a *quantum network*) only requires the receiver to have read-only access to a reasonably recent copy of the blockchain. In addition, transactions are local, so that no blockchain must be updated, nor does it require a third party to know of the transaction.

Local transactions are also independent of network access. Bitcoin requires two-way communication with the Internet, while Quantum Bitcoin transactions can be performed between two parties located in, for example, the same remote area (including deep space!). The read-only blockchain access requirement makes it possible to store a local blockchain copy in, for example, a book. A user only needs to consult this book to perform transactions, given that the quantum bitcoin in question were minted before the book was printed.

Another performance advantage is scalability. According to Garzik [18], Bitcoin as originally proposed by Nakamoto [15] has an estimated global limit of seven transactions per second. In comparison, the local transactions of Quantum Bitcoin implies that there is no upper limit to the transaction rate. It should be noted, however, that the minting rate is limited by the capacity of the two blockchains. By placing the performance restriction only in the minting procedure, the bottleneck should be much less noticeable than if it were in the transaction rate as well.

Local transactions also imply anonymity, since only the sender and receiver are aware of the transaction even occurring. There is no global paper trail of transactions. In this sense, a Quantum Bitcoin transaction is similar to handing over ordinary banknotes and coins, except that no central bank is required. Classical Bitcoin, on the other hand, records all transactions in the blockchain which allows anybody with a copy to trace transaction flows, even well after the fact. This has been used by several authors [19–23] to de-anonymize Bitcoin users.

Another advantage of Quantum Bitcoin is that transactions are free. Classical Bitcoin transactions usually require a small fee [15] to be paid to miners in order to prevent transaction spam and provide additional incentive. Nakamoto [15] also envisioned that fees will allow mining to continue past the year 2140, when the last new bitcoin is expected to be mined. In Bitcoin, mining is required for transactions to work. None of this is needed in Quantum Bitcoin as local transactions require no fees. Even a hypothetical inflation control scheme, similar to that of Bitcoin, will make mining stop completely as it no longer will be necessary at all.

Compared to Bitcoin, the blockchain of Quantum Bitcoin is smaller and grows at a more predictable rate. By design, data added to a blockchain can never be removed, and as of March 2019 the size of the Bitcoin blockchain exceeds 200 GB. Quantum Bitcoin also has a growing blockchain, however it only grows when minting currency, not due to transactions.

Per the discussion in the previous paragraph, Quantum Bitcoin mining could become superfluous so that the Quantum Bitcoin blockchain only grows to a given size. For example, if we limit the number of Quantum Bitcoin to 21 million (just like in Bitcoin) and choose 512-bit serial numbers and a 256-bit digital signature scheme $\mathcal{D}$, the Quantum Bitcoin blockchain will only ever grow to about 2 GB in size plus some overhead.

VII. Conclusion

Quantum Bitcoin is a tangible application of quantum mechanics where we construct the ideal distributed, publicly-verifiable payment system. The no-cloning theorem provides the foundation of an unforgeable item, and the addition of a blockchain allows us to produce currency without trusting a central entity. Quantum Bitcoin is the first example of a secure, distributed payment system with local transactions.

Two parties can transfer quantum bitcoin by transferring a quantum state over a quantum channel and reading off a publicly-available blockchain. Transactions are settled immediately without having to wait for confirmation from miners, yet the system can scale to allow an unlimited

[3] We assume that a transaction requires six confirmations.

rate of transactions even without transaction fees. Still, as sender and receiver must share a quantum channel, existing Internet infrastructure will not be sufficient to transact over a distance.

Note that while our protocol is secure against a counterfeiter with access to a quantum computer[4], it is not information-theoretically secure. The corresponding security proofs must therefore place the standard complexity assumptions on the attacker. Open questions that remain is if Quantum Bitcoin can be split into smaller denominations, and how the $T_{\max}$ parameter deals with blockchain forks.

References

[1] S. Wiesner, “Conjugate coding”, SIGACT News **15**, 78–88 (1983).

[2] A. Broadbent and C. Schaffner, “Quantum cryptography beyond quantum key distribution”, Designs, Codes and Cryptography **78**, 351–382 (2015).

[3] W. K. Wootters and W. H. Zurek, “A single quantum cannot be cloned”, Nature **299**, 802–803 (1982).

[4] S. Aaronson and P. Christiano, “Quantum Money from Hidden Subspaces”, Theory of Computing **9**, 349–401 (2013).

[5] C. H. Bennett, G. Brassard, S. Breidbart, and S. Wiesner, “Quantum cryptography, or unforgeable subway tokens”, in Advances in cryptology, edited by D. Chaum, R. L. Rivest, and A. T. Sherman (1983), pp. 267–275.

[6] F. Pastawski, N. Y. Yao, L. Jiang, M. D. Lukin, and J. I. Cirac, “Unforgeable Noise-Tolerant Quantum Tokens”, Proceedings of the National Academy of Sciences **109**, 16079–16082 (2012).

[7] A. Brodutch, D. Nagaj, O. Sattath, and D. Unruh, “An adaptive attack on Wiesner’s quantum money”, arXiv:1404.1507 [quant-ph] (2014).

[8] C. H. Bennett and G. Brassard, “Quantum cryptography: Public key distribution and coin tossing”, in Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing (1984), pp. 175–179.

[9] M. Mosca and D. Stebila, “Quantum Coins”, arXiv:0911.1295 [quant-ph] (2009).

[10] S. Aaronson, “Quantum Copy-Protection and Quantum Money”, in 24th Annual IEEE Conference on Computational Complexity, 2009. CCC ’09 (July 2009), pp. 229–242.

[11] A. Lutomirski, S. Aaronson, E. Farhi, D. Gosset, A. Hassidim, J. Kelner, and P. Shor, “Breaking and making quantum money: toward a new quantum cryptographic protocol”, arXiv:0912.3825 [quant-ph] (2009).

[12] E. Farhi, D. Gosset, A. Hassidim, A. Lutomirski, and P. Shor, “Quantum money from knots”, arXiv:1004.5127 [quant-ph] (2010).

[13] A. Lutomirski, “Component mixers and a hardness result for counterfeiting quantum money”, arXiv:1107.0321 [quant-ph] (2011).

[14] M. C. Pena, J.-C. Faugère, and L. Perret, “Algebraic Cryptanalysis of a Quantum Money Scheme: The Noise-Free Case”, in Public-Key Cryptography – PKC 2015, edited by J. Katz, Lecture Notes in Computer Science (2015), pp. 194–213.

[15] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system”, Consulted (2008).

[16] J. Rompel, “One-way Functions Are Necessary and Sufficient for Secure Signatures”, in Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing, STOC ’90 (1990), pp. 387–394.

[17] G. O. Karame, E. Androulaki, and S. Capkun, “Double-spending Fast Payments in Bitcoin”, in Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS ’12 (2012), pp. 906–917.

[18] J. Garzik, *Making Decentralized Economic Policy*, BIP 100 - Theory and Discussion, v0.8.1 (June 15, 2015).

[19] F. Reid and M. Harrigan, “An Analysis of Anonymity in the Bitcoin System”, in *Security and Privacy in Social Networks*, edited by Y. Altshuler, Y. Elovici, A. B. Cremers, N. Aharony, and A. Pentland (Springer New York, New York, NY, 2013), pp. 197–223.

[20] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, “A Fistful of Bitcoins: Characterizing Payments Among Men with No Names”, in Proceedings of the 2013 Conference on Internet Measurement Conference, IMC ’13 (2013), pp. 127–140.

[21] M. Möser, R. Böhme, and D. Breuker, “An inquiry into money laundering tools in the Bitcoin ecosystem”, in 2013 APWG eCrime Researchers Summit (Sept. 2013), pp. 1–14.

[22] M. Spagnuolo, F. Maggi, and S. Zanero, “BitIodine: Extracting Intelligence from the Bitcoin Network”, in Financial Cryptography and Data Security, edited by N. Christin and R. Safavi-Naini, Lecture Notes in Computer Science (2014), pp. 457–468.

[23] D. Kondor, M. Pósfai, I. Csabai, and G. Vattay, “Do the Rich Get Richer? An Empirical Analysis of the Bitcoin Transaction Network”, PLoS ONE **9**, edited by M. Perc, e86197 (2014).

[4] Full quantum resistance requires a quantum-safe digital signature scheme $\mathcal{D}$.