Intuitions Behind Quantum Lightning

Author:

xxx

UC Berkeley

date:

Abstract

Quantum money leverages the no-cloning theorem to create unforgeable digital currency. While private-key schemes (Wiesner, BBBW) require trusted banks, public-key quantum money enables decentralized verification. Zhandry's *quantum lightning* framework strengthens security by demanding that no adversary can produce two valid states with the same serial number—a property strictly stronger than traditional unforgeability. This report presents the intuitions behind quantum lightning, focusing on the degree-2 polynomial construction based on the Non-Affine Multi-Collision Resistance (NAMCR) assumption. We then analyze a critical limitation: public-key quantum money inherently lacks supply constraints, leading to potential hyperinflation in any cryptocurrency application. We prove that unbounded generation is a necessary consequence of public verification, and survey prospective solutions including verifiable delay functions, hybrid blockchain systems, and economic mechanisms. Our analysis reveals a fundamental tension between decentralization, scarcity, and simplicity that remains an open problem in quantum cryptography.

Contents

1 Motivation for Quantum Money

The motivation for quantum money originates from the fact that quantum states cannot, in general, be cloned. Formally, the no-cloning theorem states that there is no completely positive trace-preserving (CPTP) map $\mathcal{C}$ satisfying

$$\mathcal{C}(|\psi\rangle\langle\psi|) = |\psi\rangle\langle\psi| \otimes |\psi\rangle\langle\psi| \quad \text{for all } |\psi\rangle.$$

This physical constraint suggests that a quantum state may serve as an unclonable certificate of validity—an idea first captured in Wiesner's original conception of quantum money.

1.1 Wiesner's Private-Key Quantum Money

In Wiesner's scheme, each banknote consists of a classical serial number s and a quantum state

$$|\$_s\rangle = \bigotimes_{i=1}^{n} |\psi_i\rangle, \quad |\psi_i\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}.$$

The bank privately stores a classical database mapping

$$s \mapsto (\text{basis choices } b_i),$$

and verification consists of measuring each qubit in its designated basis. The verification procedure is therefore a private function:

$$\mathsf{Ver}_{\text{bank}}(\rho, s) = \begin{cases} 1 & \text{if measurements match } b_i, \\ 0 & \text{otherwise.} \end{cases}$$

This construction achieves information-theoretic security but is fundamentally private-key: verification requires secret information.

1.2 Public-Key vs. Private-Key Quantum Money

Public-key quantum money aims to make the verification map

$$\mathsf{Ver} : \mathcal{D}(\mathcal{H}) \to \{0, 1\}$$

publicly computable, meaning anyone can check whether a state is valid. A public-key scheme provides:

- a public verification circuit V,

- such that $V(\rho) = 1$ for valid notes, and
- it is computationally infeasible to prepare any ρ' satisfying $V(\rho') = 1$.

Formally, if $\mathcal{G}$ is the public generation procedure, a sound scheme requires that no QPT adversary $\mathcal{A}$ can produce

$$\rho_1, \rho_2 \quad \text{such that} \quad \mathsf{Ver}(\rho_1) = \mathsf{Ver}(\rho_2) = 1$$

except with negligible probability. This definition mirrors unforgeability in classical signature schemes but with quantum states as certificates.

1.3 Why Classical Public-Key Verification is Desirable

A classical verification algorithm is particularly desirable: instead of requiring a quantum device, verification becomes evaluating a classical predicate

$$V(s, y) \in \{0, 1\},$$

where s is the serial number and y may be classical information derived from the quantum state. The goal is:

$$\text{Quantum banknote } \rho \quad \xrightarrow{\text{measure}} \quad y \quad \xrightarrow{V(\cdot)} \quad \text{valid/invalid.}$$

This enables circulation without trusted authorities and aligns quantum money with public-key cryptographic primitives.

1.4 Problems with Earlier Approaches

Earlier approaches attempted to achieve classical public-key verification using additional structure, often leading to vulnerabilities:

Oracle-based constructions. Many early schemes were proven secure only relative to a black-box oracle $\mathcal{O}$, making them unsuitable for concrete instantiation.

The Aaronson–Christiano subspace scheme. Their candidate relied on obfuscating membership in a hidden subspace $S \subseteq \mathbb{F}_2^n$. A banknote was a uniform superposition

$$|\$\rangle = \frac{1}{\sqrt{|S|}} \sum_{x \in S} |x\rangle,$$

and verification tested that

$$x \in S \quad \text{and} \quad Hx \in S^{\perp},$$

where H is the Hadamard transform. However, subsequent works showed that the "subspace-hiding obfuscation" used to publish the verification procedure leaked information about S itself, enabling forgery.

Structural leakage in general. Making verification public often reveals algebraic information that adversaries can exploit. This highlights the need for public-key quantum money schemes based on assumptions that resist such leakage—one of the main motivations behind Zhandry's quantum lightning framework.

2 Quantum Lightning: Zhandry's Contribution

Zhandry formalizes *quantum lightning* as a public procedure for generating quantum states that satisfy a strong uniqueness property: it should be computationally infeasible for any efficient adversary to produce two valid states — called *bolts* — that verify to the same classical serial number. This goes beyond the ordinary no-cloning theorem, which prohibits duplicating a *given* unknown quantum state but does not preclude an adversary from generating two different states that nonetheless pass verification.

2.1 The Win-Win Framework

Before describing the construction, it is essential to understand Zhandry's "win-win" framework that motivates the entire approach. Consider a collision-resistant hash function H secure against quantum adversaries. Zhandry shows that such a function must fall into one of two categories:

1. H is *collapsing* (a strong quantum security notion defined by Unruh), meaning it is computationally infeasible to distinguish between measuring just the output register versus measuring both input and output registers, or
2. H is *not collapsing*, in which case it can be used to construct quantum lightning without additional assumptions.

This dichotomy is significant: if most natural hash functions turn out to be collapsing, this validates stronger quantum security definitions. If any are not collapsing, we

immediately obtain quantum lightning. The degree-2 polynomial construction is specifically chosen because it is believed to be non-collapsing — that is, one can efficiently distinguish the superposition $|\psi_y\rangle$ of all preimages from a single random input $|x\rangle$.

2.2 Definition: Bolts and Strong Unclonability

A quantum lightning scheme consists of two public algorithms:

$$\mathsf{Storm}(1^\lambda) \to |\psi\rangle, \qquad \mathsf{Ver}(\rho) \to s \in \{0,1\}^* \cup \{\perp\}.$$

Here, Storm generates a candidate bolt and Ver either outputs a classical *serial number* s or rejects with $\perp$.

A state ρ is a valid bolt if

$$\Pr\left[\mathsf{Ver}(\rho) \neq \perp\right] \geq 1 - \mathrm{negl}(\lambda).$$

The central security requirement, called *uniqueness*, is expressed via the following experiment. An adversary outputs two (possibly entangled) states (ρ_1, ρ_2) and succeeds if

$$\mathsf{Ver}(\rho_1) = \mathsf{Ver}(\rho_2) = s \neq \perp.$$

The scheme is secure if for all QPT adversaries $\mathcal{A}$,

$$\Pr\left[\begin{array}{c}(\rho_1,\rho_2) \leftarrow \mathcal{A} \\ \mathsf{Ver}(\rho_1) = \mathsf{Ver}(\rho_2) \neq \perp\end{array}\right] = \mathrm{negl}(\lambda).$$

2.3 Why Quantum Lightning is Stronger Than Traditional Quantum Money

In public-key quantum money, unforgeability typically means that no adversary can produce a new valid banknote:

$$\mathsf{Ver}(\rho') = 1.$$

However, this does *not* prevent an adversary from producing two distinct states ρ_1, ρ_2 such that

$$\mathsf{Ver}(\rho_1) = \mathsf{Ver}(\rho_2),$$

i.e., two notes with the same classical identity.

Zhandry emphasizes that this distinction parallels classical hash-function security:

- Quantum money corresponds to *second-preimage resistance*:

$$\rho \text{ valid } \Rightarrow \text{ hard to find } \rho' \neq \rho \text{ with } \mathsf{Ver}(\rho') = \mathsf{Ver}(\rho).$$

- Quantum lightning corresponds to full *collision resistance*:

$$\text{hard to find any } \rho_1, \rho_2 \text{ with } \mathsf{Ver}(\rho_1) = \mathsf{Ver}(\rho_2) \neq \perp.$$

This stronger guarantee is essential for applications such as verifiable randomness or decentralized ledgers, where even a single duplicated serial number constitutes a complete break.

2.4 Intuition: Why "Lightning Never Strikes Twice"

The phrase captures the fundamental intuition of Zhandry's definition. Because both the generation algorithm Storm and the verification algorithm Ver are public, an adversary is free to construct arbitrary quantum states in an effort to engineer a specific serial number. Nevertheless, the scheme demands that:

No efficient adversary can ever produce two bolts with the same serial number.

When a bolt ρ is generated, the verifier extracts a classical value

$$s = \mathsf{Ver}(\rho),$$

and this s acts as a unique "fingerprint" for the underlying quantum state. The same bolt will always verify to the same s, but producing *another* state ρ' with the same verified fingerprint is assumed to be computationally infeasible.

This requirement is strictly stronger than the no-cloning theorem. No-cloning prevents a map of the form

$$|\psi\rangle \mapsto |\psi\rangle \otimes |\psi\rangle,$$

but does not prevent an adversary from preparing *two different* states ρ_1, ρ_2 such that

$$\mathsf{Ver}(\rho_1) = \mathsf{Ver}(\rho_2).$$

Quantum lightning rules this out entirely. The intuition is that each bolt contains hidden structure that, although publicly verifiable, cannot be reproduced without solving an underlying computationally hard problem. Thus, "lightning"—the successful generation of a valid bolt—never "strikes the same serial number twice."

3 The Degree-2 Polynomial Construction

Zhandry's concrete quantum lightning construction uses degree-2 polynomial hash functions over $\mathbb{F}_2$. Crucially, these hash functions are *not* collision-resistant in the standard sense. Instead, security relies on a weaker but still plausible assumption about the hardness of finding *non-affine multi-collisions* (NAMCR).

3.1 The Hash Function Family

The hash function is defined by n random upper-triangular matrices $A_i \in \{0,1\}^{m\times m}$ for $i = 1, \ldots, n$, where $m > n$. Given $\mathcal{A} = \{A_i\}_i$, the function $f_{\mathcal{A}} : \{0,1\}^m \to \{0,1\}^n$ is defined as:

$$f_{\mathcal{A}}(x) = (x^\top A_1 x, \ldots, x^\top A_n x),$$

where arithmetic is performed over $\mathbb{F}_2$.

Why degree-2 polynomials are NOT collision-resistant. As shown by Ding–Yang and Applebaum et al., these functions admit efficient collision-finding attacks. Given a random offset Δ, one can find a collision pair $(x, x-\Delta)$ by solving a *linear* system of n equations in m unknowns, which has a solution when $m \geq n$. More generally:

- For $m \approx kn$, one can efficiently find $k+1$ *affine* colliding inputs (lying on a k-dimensional affine subspace).
- One can also find $k+1$ *non-affine* colliding inputs (not lying in any $(k-1)$-dimensional affine subspace).

However, finding $2(k+1)$ non-affine colliding inputs appears to be hard. This distinction is central to Zhandry's security assumption.

3.2 The NAMCR Assumption

Zhandry introduces the *Non-Affine Multi-Collision Resistance* (NAMCR) assumption:

> **Assumption (NAMCR):** Let $k = \text{poly}(n)$ and $m < (k+\frac{1}{2})n$. For random upper-triangular matrices A_i, the function $f_{\mathcal{A}}$ is $2(k+1)$-NAMCR. That is, for any quantum polynomial-time adversary $\mathcal{A}$,
>
> $$\Pr\left[(x_1, \ldots, x_{2k+2}) \text{ collide in } f_{\mathcal{A}} \text{ and are non-affine}\right] = \text{negl}(\lambda).$$

The key insight is that while affine collisions are easy to find (via linear algebra), finding collisions that have no affine relationships is conjectured to be hard. Known attacks can produce $k+1$ non-affine collisions but fail to produce $2(k+1)$.

3.3 The Bolt Structure: Why Multiple Copies Are Necessary

A single superposition state

$$|\psi_y\rangle = \frac{1}{\sqrt{|S_y|}} \sum_{x:f_{\mathcal{A}}(x)=y} |x\rangle$$

is **not** a secure bolt. This is because the collision-finding attack can generate $k+1$ copies of the *same* $|\psi_y\rangle$: one simply uses the attack to produce $(x_0, x_1, \ldots, x_k)$ with $f_{\mathcal{A}}(x_i) = y$ for all i, and then constructs

$$|x_0, x_1, \ldots, x_k\rangle \approx |\psi_y\rangle^{\otimes(k+1)}.$$

Therefore, a secure bolt must consist of *multiple* tensor copies:

$$\mathbf{B}_y := |\psi_y\rangle^{\otimes(r+1)},$$

where $r \approx k$ is chosen so that honest generation can produce $r+1$ copies but an adversary cannot produce $2(r+1)$ copies with the same serial number.

The bolt generation procedure works as follows:

1. Generate a uniform superposition over random offsets $\Delta_1, \ldots, \Delta_k$.
2. In superposition, compute the affine subspace S_Δ of colliding inputs and sample uniformly from it.
3. Apply $f_{\mathcal{A}}$ and measure to obtain y.
4. Transform the resulting state to obtain $|x_0, x_1, \ldots, x_k\rangle$ where all x_i hash to y.

The output is (negligibly close to) $|\psi_y\rangle^{\otimes(k+1)}$.

3.4 Verification: Mini-Verification and Span Membership

Verification is non-trivial and proceeds in two stages:

Mini-verification on each component. For each of the $(k+1)$ tensor components, the verifier checks whether the state lies in the span of $\{|\psi_z\rangle : z \in \{0,1\}^n\}$. This is equivalent to checking membership in the span of states

$$|\phi_r\rangle = \frac{1}{2^{m/2}} \sum_x (-1)^{r \cdot f_{\mathcal{A}}(x)} |x\rangle,$$

for $r \in \{0,1\}^n$. The verification uses a carefully designed procedure that:

- Applies Hadamard gates to extract linear constraints from the degree-2 phase polynomial.
- Iteratively solves for the "hidden" vector r by measuring and solving linear systems.
- Projects the state onto the correct span, rejecting if the projection fails.

Consistency check. After mini-verification passes on all components, the verifier measures $f_{\mathcal{A}}(x)$ on each component to obtain serial numbers $y_1, \ldots, y_{k+1}$. Verification accepts only if all y_i are equal, outputting this common value y as the serial number.

Why this prevents forgery. If verification accepts on two bolts with the same serial number y, the post-verification state is exactly $|\psi_y\rangle^{\otimes 2(k+1)}$. Measuring this state yields $2(k+1)$ random preimages of y. With overwhelming probability, these preimages have no affine relationships, violating the NAMCR assumption.

3.5 Summary: The Security Argument

The security of Zhandry's quantum lightning construction can be summarized as follows:

1. The hash function $f_{\mathcal{A}}$ (degree-2 polynomials) is *not* collision-resistant: affine collisions are easy.
2. However, $f_{\mathcal{A}}$ is conjectured to be NAMCR: finding $2(k+1)$ non-affine colliding inputs is hard.
3. A single copy $|\psi_y\rangle$ is insecure because the attack produces $k+1$ copies.
4. A bolt must be $|\psi_y\rangle^{\otimes(k+1)}$, requiring any adversary producing two bolts to find $2(k+1)$ non-affine collisions.
5. Verification projects onto the correct span, ensuring that the only accepted states are honest superpositions.

This construction represents the first quantum lightning scheme based on a plausible classical computational assumption, providing a foundation for public-key quantum money and verifiable randomness.

3.6 Zhandry's Instantiation Using Multi-Collision-Resistant Hash Functions

The candidate hash family considered by Zhandry consists of random degree-2 polynomials over $\mathbb{F}_2$:

$$H_A(x) = x^\top Ax \ \in \ \mathbb{F}_2,$$

where A is a random symmetric matrix over $\mathbb{F}_2^{n\times n}$. Such functions admit many collisions—indeed, solving $H_A(x) = 0$ is equivalent to finding vectors in a quadratic variety. However, Zhandry's construction does *not* require traditional collision resistance. Instead, it relies on the hardness of producing *large, non-affine* collision sets.

Formally, an adversary succeeds in a multi-collision attack if it outputs a set

$$S = \{x_1, \ldots, x_k\} \subseteq \mathbb{F}_2^n \quad \text{such that} \quad H_A(x_1) = H_A(x_2) = \cdots = H_A(x_k),$$

and such that S satisfies additional independence properties—for example, S must not be contained in any affine subspace of dimension significantly smaller than k. Zhandry argues that producing such a structured collision set would require solving problems believed to be computationally infeasible even for quantum adversaries.

A bolt in the scheme corresponds to a state whose amplitudes are distributed uniformly over a large affine subspace of preimages. Replicating this structure would require generating a corresponding structured collision set, which is assumed to be hard. This is the essence of Zhandry's *multi-collision-resistance* assumption.

3.7 The Idea of Incompressibility

A valid bolt has the form

$$|\psi_y\rangle = \frac{1}{\sqrt{|S_y|}} \sum_{x\in S_y} |x\rangle, \qquad S_y := \{x : H_A(x) = y\},$$

where S_y is typically an affine subspace of dimension $\Theta(n)$. The crucial observation is that S_y is *too large and structured* to be described by a short classical string.

Zhandry formalizes incompressibility using the notion that no QPT algorithm $\mathcal{A}$ can, given oracle access to H_A, output a string d such that a second algorithm can regenerate

a large subset of S_y:

$$d \xrightarrow{\mathcal{R}} S'_y \subseteq S_y, \qquad |S'_y| \gg \text{poly}(n).$$

If such a compression were possible, then an adversary could create a second bolt:

$$|\psi'_y\rangle = \frac{1}{\sqrt{|S'_y|}} \sum_{x \in S'_y} |x\rangle,$$

that verifies to the same serial number y, thereby violating the fundamental uniqueness requirement of quantum lightning.

Thus, incompressibility asserts that no efficient adversary can replace the exponentially large structure of S_y with a polynomial-size classical description. This property prevents an attacker from reproducing the combinatorial structure encoded in a bolt, making it computationally infeasible to produce two bolts with the same serial number.

In summary, the collision geometry induced by random degree-2 hash functions is believed to be too "spread out" and too high-dimensional to be succinctly encoded or reconstructed. This is the hash-based foundation that makes quantum lightning plausible: the internal structure of a bolt cannot be duplicated, ensuring that "lightning never strikes the same serial number twice."

4 The Inflation Problem: Unlimited Generation in Public-Key Quantum Money

The previous sections established the cryptographic security of quantum lightning—specifically its resistance to cloning (uniqueness). However, a fundamental economic limitation arises when attempting to use this primitive as a decentralized currency: *there is no inherent mechanism to limit the rate or total amount of money generation.* This section provides a rigorous analysis of why public-key quantum money schemes, including quantum lightning, suffer from unbounded generation capabilities.

4.1 The Fundamental Asymmetry: Cloning vs. Generation

The security of quantum money rests on the *no-cloning theorem* and its computational extensions. However, there is a crucial asymmetry that must be understood:

> **No-cloning prevents duplication of existing money, but does not prevent creation of new money.**

Let us formalize this distinction. A quantum money scheme provides two algorithms:

$$\mathsf{Gen}(1^\lambda) \to |\$\rangle, \tag{1}$$

$$\mathsf{Ver}(\rho) \to s \in \{0,1\}^* \cup \{\bot\}. \tag{2}$$

The security property (uniqueness/unforgeability) states that for any QPT adversary $\mathcal{A}$:

$$\Pr\left[\mathsf{Ver}(\rho_1) = \mathsf{Ver}(\rho_2) = s \neq \bot \ : \ (\rho_1, \rho_2) \leftarrow \mathcal{A}(|\$\rangle)\right] = \mathrm{negl}(\lambda).$$

Critically, this definition says nothing about the following scenario:

$$\Pr\left[\mathsf{Ver}(\rho_i) \neq \bot \text{ for } i = 1, \ldots, N \ : \ (\rho_1, \ldots, \rho_N) \leftarrow \mathcal{A}^{\mathsf{Gen}}\right],$$

where $\mathcal{A}$ is allowed to invoke Gen polynomially many times. In fact, this probability is *overwhelming* by the correctness of the scheme.

Theorem 4.1 (Unbounded Generation)**.** *Let* $(\mathsf{Gen}, \mathsf{Ver})$ *be any public-key quantum money scheme with correctness error* ϵ*. Then for any polynomial* $N = N(\lambda)$*, there exists a QPT algorithm that produces* N *valid, pairwise-distinct banknotes with probability at least* $(1-\epsilon)^N - \mathrm{negl}(\lambda)$*.*

Proof. Consider the algorithm that simply invokes $\mathsf{Gen}(1^\lambda)$ independently N times, obtaining states $|\$_1\rangle, \ldots, |\$_N\rangle$. By correctness, each state passes verification with probability at least $1 - \epsilon$.

For distinctness, observe that each invocation of Gen samples a serial number from a distribution with high min-entropy. More precisely, the min-entropy security property of quantum lightning (which follows from uniqueness) guarantees that:

$$H_\infty(\mathsf{Ver}(\mathsf{Gen}(1^\lambda))) \geq n(\lambda) - O(\log \lambda),$$

where $n(\lambda)$ is the bit-length of serial numbers. Therefore, the probability of collision among $N = \mathrm{poly}(\lambda)$ serial numbers is at most:

$$\binom{N}{2} \cdot 2^{-n+O(\log\lambda)} = \mathrm{negl}(\lambda).$$

By a union bound, all N states are valid and pairwise distinct with probability at least $(1-\epsilon)^N - \mathrm{negl}(\lambda)$. □

4.2 Why Public Verification Implies Public Generation

A key insight is that in any public-key quantum money scheme, the generation algorithm must be publicly executable. This is not merely a design choice but a consequence of the security model.

Proposition 4.2 (Public Generation is Inherent)**.** *In any public-key quantum money scheme where* Ver *is public, any party with access to* Ver *can efficiently generate valid banknotes.*

Proof. Consider the following strategy. The generation algorithm Gen consists of:

1. Prepare some initial quantum state $|\phi_0\rangle$.
2. Apply a sequence of operations (possibly including measurements).
3. Output a state $|\$\rangle$ that passes Ver.

Since Ver is public, any party can simulate this procedure. The only constraint is computational efficiency. In Zhandry's construction, Gen runs in polynomial time, and all operations are publicly specified. Therefore, generation is inherently a public capability. □

This stands in stark contrast to *private-key* schemes (e.g., Wiesner, BBBW), where generation requires secret information:

$$\mathsf{Gen}_{\text{private}}(k, s) \to |\$_{k,s}\rangle,$$

and only the bank possessing the secret key k can produce valid notes.

4.3 Formal Analysis: The Economics of Unlimited Supply

We now provide a formal economic analysis of the inflation problem. Let $M(t)$ denote the total money supply at time t, and let $C(t)$ denote the computational capacity (in operations per unit time) available to all participants.

Definition 4.3 (Marginal Cost of Generation)**.** The marginal cost of generating one unit of currency is:

$$\mathcal{C}_{\text{marginal}} = \frac{(\text{computational cost per } \mathsf{Gen} \text{ invocation})}{(\text{expected value of generated bolt})}.$$

For quantum lightning, each invocation of Gen requires $T_{\mathsf{Gen}} = \text{poly}(\lambda)$ quantum operations. As quantum computing technology advances:

1. The cost per quantum operation decreases (Moore's law analogue for quantum).
2. The total available computational capacity $C(t)$ increases exponentially.

This leads to the following dynamics:

$$\frac{dM}{dt} = \frac{C(t)}{T_{\mathsf{Gen}}},$$

where we assume rational actors generate money whenever $\mathcal{C}_{\text{marginal}}$ is below the market value.

Theorem 4.4 (Inflation Dynamics)**.** *In a quantum lightning-based currency without external supply constraints, if computational capacity grows as $C(t) = C_0 e^{\gamma t}$ for some growth rate $\gamma > 0$, then:*

$$M(t) = M_0 + \frac{C_0}{T_{\mathsf{Gen}} \cdot \gamma} \left(e^{\gamma t} - 1\right).$$

In particular, $M(t) \to \infty$ exponentially fast.

This exponential growth in money supply, without corresponding growth in economic value, constitutes *hyperinflation*.

4.4 The Mechanism: Why Each Generation Succeeds

To understand why generation is unrestricted, we examine the concrete mechanism in Zhandry's degree-2 polynomial construction.

Step 1: Superposition over preimage space. The Storm algorithm begins by preparing a superposition:

$$|\phi_0\rangle = \frac{1}{2^{m/2}} \sum_{x \in \{0,1\}^m} |x\rangle.$$

Step 2: Computation and measurement. The algorithm computes $f_{\mathcal{A}}(x)$ in superposition and measures the output register:

$$|\phi_0\rangle \xrightarrow{U_{f_{\mathcal{A}}}} \frac{1}{2^{m/2}} \sum_x |x\rangle|f_{\mathcal{A}}(x)\rangle \xrightarrow{\text{measure}} |\psi_y\rangle = \frac{1}{\sqrt{|S_y|}} \sum_{x: f_{\mathcal{A}}(x)=y} |x\rangle,$$

where the measurement outcome y becomes the serial number.

Step 3: Tensor product structure. A valid bolt consists of $k+1$ copies:

$$\mathbf{B}_y = |\psi_y\rangle^{\otimes(k+1)}.$$

The crucial observation is that each invocation of this procedure:

- Succeeds with probability 1 (up to negligible error).
- Produces a uniformly random serial number $y \in \{0,1\}^n$.
- Requires only $\text{poly}(\lambda)$ quantum operations.
- Is independent of all previous generations.

Proposition 4.5 (Independence of Generations)**.** *Let $y_1, y_2, \ldots, y_N$ be the serial numbers obtained from N independent invocations of* Storm*. Then:*

$$\Pr[y_i = y_j \text{ for some } i \neq j] \leq \frac{N^2}{2^n} = \text{negl}(\lambda)$$

for $N = \text{poly}(\lambda)$.

This independence is precisely what prevents any form of "congestion" or "saturation" in the generation process. Unlike Bitcoin, where miners compete for the same block reward and difficulty adjusts to maintain a target block time, quantum lightning generators operate in complete isolation.

4.5 Comparison with Classical Cryptocurrencies

To highlight the severity of the inflation problem, we compare quantum lightning with Bitcoin's difficulty adjustment mechanism.

Property	**Bitcoin**	**Quantum Lightning**
Generation rate	Difficulty-adjusted	Unbounded
Global state	Blockchain	None
Difficulty function	$H(\text{block}) < T_{\text{target}}$	None
Supply cap	21 million BTC	∞
Consensus required	Yes (proof-of-work)	No

In Bitcoin, the difficulty target T_{target} is recalculated every 2016 blocks based on the time taken to mine the previous 2016 blocks:

$$T_{\text{new}} = T_{\text{old}} \times \frac{\text{actual time}}{\text{target time (2 weeks)}}.$$

This feedback mechanism ensures that regardless of total network hashrate, the average block time remains approximately 10 minutes. Quantum lightning lacks any analogous mechanism because:

1. There is no global state to track total generation rate.
2. Verification is local and instantaneous.
3. There is no "competition" between generators.

5 The Inflation Problem in Decentralized Quantum Lightning

While the previous sections established the cryptographic security of the Degree-2 Polynomial construction—specifically its resistance to cloning (uniqueness)—a major limitation arises when applying this primitive to decentralized currency. Unlike classical cryptocurrencies which utilize dynamic difficulty adjustments to regulate supply, Quantum Lightning, in its raw form, suffers from unbounded generation capabilities.

5.1 Theoretical Absence of Supply Caps

In standard public-key quantum money schemes (e.g., Aaronson-Christiano), the supply of currency is centralized; the bank holds the secret parameters required to generate valid states $|\$_s\rangle$. However, Zhandry's definition of Quantum Lightning inherently requires the generation procedure, denoted as $\mathcal{S}$ (Storm), to be public.

Formally, the scheme consists of public algorithms:

$$\mathcal{S}(1^\lambda) \to |\xi\rangle, \quad \mathcal{V}(|\xi\rangle) \to s \in \{0,1\}^* \cup \{\perp\} \tag{3}$$

The security definition (Uniqueness) ensures that it is computationally infeasible to produce two bolts $|\xi_1\rangle, |\xi_2\rangle$ such that $\mathcal{V}(|\xi_1\rangle) = \mathcal{V}(|\xi_2\rangle) = s$. However, this definition imposes no limit on the generation of distinct bolts $|\xi_1\rangle, \ldots, |\xi_N\rangle$ with distinct serial numbers $s_1, \ldots, s_N$.

In a decentralized setting without a global ledger (blockchain), there is no mechanism to throttle the execution of $\mathcal{S}$. Consequently, the rate of money supply R is strictly a function of available computational power P:

$$R \propto P \cdot \frac{1}{T_{\mathcal{S}}} \tag{4}$$

where $T_{\mathcal{S}}$ is the time complexity of the bolt generation algorithm. As hardware technology improves, the marginal cost of producing a unit of currency approaches zero, leading to inevitable inflation.

5.2 The Mechanism of Infinite Generation

To understand the mechanics of this infinite generation, we examine the instantiation of the bolt generation procedure using Degree-2 polynomials. The procedure generates a serial number effectively at random from an exponentially large space.

Recalling the generation steps from Section 3:

1. The Storm algorithm creates a superposition over inputs and random offsets Δ:

$$|\phi_0\rangle = \sum_{\Delta} \sum_{x \in S_\Delta} |x, \Delta\rangle \tag{5}$$

2. The algorithm computes the hash function $f_{\mathcal{A}}(x)$ in superposition and measures the result to obtain a serial number y:

$$|\phi_0\rangle \xrightarrow{\text{Measure } f_{\mathcal{A}}} |\psi_y\rangle \propto \sum_{x : f_{\mathcal{A}}(x) = y} |x\rangle \tag{6}$$

The measurement outcome y serves as the serial number. Because $f_{\mathcal{A}} : \{0,1\}^m \to \{0,1\}^n$ maps to a space of size 2^n, and the measurement outcome is probabilistic, repeated invocations of the algorithm yield distinct serial numbers with overwhelming probability.

Unlike Bitcoin, where finding a valid block requires iterating through nonces to satisfy a difficulty target that adjusts globally, the Quantum Lightning generation is constant-time (polynomial in λ). An adversary can simply run $\mathcal{S}$ in parallel loops, generating valid, unique bolts $|\psi_{y_1}\rangle, |\psi_{y_2}\rangle, \ldots$ ad infinitum. Since the validity of a bolt is self-contained (verified solely by $\mathcal{V}$ without a public ledger), there is no mechanism to declare that the "space" of serial numbers is saturated.

6 Prospective Solutions: Toward Scarcity in Quantum Money

The inflation problem identified in Section 4 presents a fundamental challenge for deploying quantum lightning as a practical currency. This section surveys potential solutions, analyzing their theoretical soundness and practical feasibility. We emphasize that all proposed solutions involve trade-offs, and achieving true scarcity without centralization remains an open problem.

6.1 Approach 1: Static Difficulty via Hash Constraints

The most straightforward approach, suggested by Zhandry himself, is to impose a proof-of-work constraint on the serial number.

Definition 6.1 (Difficulty-Constrained Coin)**.** A valid coin is a bolt $|\xi\rangle$ such that:

$$\mathsf{ValidCoin}(|\xi\rangle) \iff \mathsf{Ver}(|\xi\rangle) = y \neq \perp \ \wedge \ H(y) < T_{\text{target}},$$

where $H : \{0,1\}^* \to \{0,1\}^k$ is a cryptographic hash function and $T_{\text{target}} = 2^{k-d}$ for difficulty parameter d.

To generate a valid coin, one must repeatedly invoke Storm until obtaining a serial number y with $H(y) < T_{\text{target}}$. The expected number of attempts is:

$$\mathbb{E}[\text{attempts}] = \frac{2^k}{T_{\text{target}}} = 2^d.$$

Limitations. As Zhandry explicitly notes in the original paper:

> "This cryptocurrency is unlikely to be useful in practice due to a very important limitation. Namely, as technology gets better, it will be easier and easier to create new coins. Without any modifications, this will lead to an exponentially increasing supply of coins, and hence rampant inflation."

The fundamental flaw is that d must be fixed at system genesis. Let $C(t)$ denote the computational capacity at time t. The coin generation rate is:

$$R(t) = \frac{C(t)}{2^d \cdot T_{\mathsf{Storm}}}.$$

If $C(t)$ grows exponentially (as expected for quantum computing), then $R(t)$ also grows exponentially, regardless of the initial choice of d.

Proposition 6.2 (Impossibility of Static Difficulty)**.** *For any fixed difficulty d and any desired constant inflation rate $r > 0$, there exists a time t^* such that for all $t > t^*$, the actual inflation rate exceeds r.*

Proof. Let $C(t) = C_0 e^{\gamma t}$ for growth rate $\gamma > 0$. The inflation rate at time t is proportional to $R(t) = C_0 e^{\gamma t}/(2^d T_{\mathsf{Storm}})$. For any target rate r, solving $R(t) > r$ gives:

$$t > t^* := \frac{1}{\gamma} \ln\left(\frac{r \cdot 2^d \cdot T_{\mathsf{Storm}}}{C_0}\right).$$

Since t^* is finite, the claim follows. □

The Fungibility Dilemma. One might suggest increasing d over time. However, this creates a *fungibility* problem: coins minted under old (easier) difficulty would need to be invalidated or exchanged, which:

1. Requires a central authority to manage the transition.
2. Cannot distinguish "old legitimate coins" from "recently minted easy coins."
3. Violates the trustless nature of the system.

6.2 Approach 2: Verifiable Delay Functions (VDFs)

A more sophisticated approach uses *verifiable delay functions* to bind coin validity to physical time.

Definition 6.3 (Verifiable Delay Function)**.** A VDF is a function $f : \mathcal{X} \to \mathcal{Y}$ with the following properties:

1. **Sequential:** Computing $f(x)$ requires T sequential operations, even with polynomial parallelism.
2. **Efficiently verifiable:** Given (x, y), one can verify $y = f(x)$ in time $\ll T$.
3. **Unique output:** For each x, there is a unique valid y.

The key property is that VDFs cannot be parallelized. This means that regardless of an adversary's computational power, computing $f(x)$ takes at least time T in wall-clock time.

Time-Locked Coins. We can define a time-locked coin as:

$$\mathsf{ValidCoin}(|\xi\rangle, \pi, t) \iff \begin{cases} \mathsf{Ver}(|\xi\rangle) = y \neq \perp, \\ \pi \text{ is a valid VDF proof for } (y, t), \\ H(y\|t) < T_{\text{target}}(t). \end{cases}$$

Here, $T_{\text{target}}(t)$ is a difficulty schedule that increases over time. The VDF proof π certifies that the coin was generated at time t, allowing the verifier to apply the appropriate difficulty level.

Challenges.

1. **Trusted time source:** Who determines the current time t? Without consensus, different parties may disagree.
2. **VDF security:** Current VDF constructions rely on assumptions (e.g., RSA group structure) that may not hold against quantum adversaries.
3. **Bootstrapping:** The system requires an initial trusted setup or reference point.

6.3 Approach 3: Hybrid Blockchain-Quantum Systems

A pragmatic approach is to combine quantum lightning with a classical blockchain, using each component's strengths.

Architecture.

1. **Quantum layer:** Provides unforgeable physical tokens (bolts).
2. **Classical layer:** Provides global consensus on which tokens are valid and prevents double-spending.

Definition 6.4 (Blockchain-Registered Quantum Coin)**.** A coin consists of a bolt $|\xi\rangle$ and a blockchain entry (y, block_i) where:

- $y = \mathsf{Ver}(|\xi\rangle)$ is the serial number.
- block_i is the block in which y was first registered.

A coin is valid if and only if (y, block_i) appears in the blockchain and no later block contains a transfer or invalidation of y.

This hybrid approach allows:

- **Dynamic difficulty:** The blockchain can implement difficulty adjustment based on global generation rate.
- **Supply cap:** The blockchain can enforce a maximum number of registrable coins.
- **Timestamping:** Each coin has a verifiable creation time.

Trade-offs. The hybrid approach reintroduces some centralization concerns:

- Requires miners/validators for the blockchain.
- Subject to 51% attacks on the classical layer.
- Loses the "blockchain-less" property that motivated quantum lightning.

However, it retains the key quantum advantage: the physical token cannot be copied, even if the blockchain is compromised. This provides defense-in-depth that purely classical systems lack.

6.4 Approach 4: Proof-of-Space and Storage Costs

Another approach ties coin generation to scarce physical resources rather than computational time.

Definition 6.5 (Storage-Bound Generation)**.** To generate a coin, one must:

1. Commit to a large random string R of size S bits.
2. Store R for a duration T.
3. Prove storage via a *proof of space-time* protocol.
4. Only after successful proof, invoke Storm with seed derived from R.

The scarcity arises because storage capacity is physically limited and cannot grow as rapidly as computational power.

Quantum Considerations. An important observation is that quantum memory is significantly more expensive and fragile than classical storage. A system requiring quantum memory commitment would naturally limit generation rate due to hardware constraints.

6.5 Approach 5: Economic Mechanisms (Burning and Demurrage)

Rather than preventing unlimited generation, one can design economic mechanisms that make generation costly or discourage hoarding.

Coin Burning for Registration. Require that to register a new coin, one must "burn" (provably destroy) existing coins of comparable value. This creates a zero-sum dynamic where new coin creation reduces existing supply.

Demurrage (Negative Interest). Coins could lose value over time unless "renewed" through some costly process. This:

- Discourages hoarding and encourages circulation.
- Creates a natural "decay" that offsets unlimited generation.
- Has historical precedent (Gesell's "stamp scrip" currency).

6.6 Summary: The State of the Art

Approach	Decentralized?	Quantum-Safe?	Practical?
Static difficulty	Yes	Yes	No (inflation)
VDF-based	Partially	Unknown	Limited
Hybrid blockchain	No	Partially	Yes
Proof-of-space	Partially	Yes	Limited
Economic mechanisms	Varies	Yes	Experimental

Remark 6.6 (Fundamental Tension)**.** There appears to be a fundamental tension between three desirable properties:

1. **Decentralization:** No trusted parties or central authorities.
2. **Scarcity:** Bounded or controlled money supply.
3. **Simplicity:** No complex consensus mechanisms.

Classical cryptocurrencies achieve (1) and (2) by sacrificing (3)—they require sophisticated consensus protocols. Quantum lightning achieves (1) and (3) but sacrifices (2). Whether all three can be achieved simultaneously remains a major open problem.

6.7 Open Problems

We conclude with several open problems suggested by this analysis:

1. **Quantum VDFs:** Can we construct verifiable delay functions secure against quantum adversaries?
2. **Decentralized Time:** Is there a way to achieve consensus on time without a blockchain?
3. **Physical Scarcity:** Can quantum money be tied to some inherently scarce quantum resource (e.g., entanglement)?
4. **Alternative Security Definitions:** Could a modified security definition for quantum money naturally incorporate supply constraints?
5. **Hybrid Optimality:** For hybrid systems, what is the minimal amount of classical infrastructure needed to achieve scarcity?

Until these problems are resolved, public-key quantum money serves primarily as a cryptographic primitive demonstrating "no-cloning with public verification" rather than a practical monetary system. Nevertheless, the theoretical insights from quantum lightning—particularly the win-win framework and the degree-2 polynomial construction—represent significant advances in our understanding of quantum cryptography.

7 Conclusion

This report has explored the theoretical foundations of quantum money, from Wiesner's original private-key construction to Zhandry's quantum lightning framework. We have examined both the remarkable security properties enabled by quantum mechanics and the fundamental limitations that arise when attempting to use these constructions as practical currencies.

Key Contributions of Quantum Lightning. Zhandry's quantum lightning represents a significant conceptual advance in several ways:

1. **Stronger security:** The uniqueness property—that no adversary can produce two bolts with the same serial number—is strictly stronger than traditional quantum money security, paralleling the distinction between collision resistance and second-preimage resistance in classical cryptography.
2. **The win-win framework:** By connecting quantum lightning to the collapsing property of hash functions, Zhandry establishes a powerful dichotomy: either most

natural hash functions satisfy a strong quantum security notion, or quantum lightning can be constructed unconditionally.

3. **Concrete instantiation:** The degree-2 polynomial construction provides the first candidate quantum lightning scheme based on a plausible classical hardness assumption (NAMCR), moving beyond oracle-based security proofs.

The Inflation Problem. Despite these achievements, we have demonstrated that public-key quantum money inherently lacks supply constraints. The core issue is fundamental: security definitions focus on preventing *duplication* of existing money, not *creation* of new money. Since the generation procedure must be public (to enable decentralization), anyone can mint new currency at will, subject only to computational resources.

This stands in stark contrast to private-key schemes, where the bank's secret key acts as a natural rate limiter. The tension between public verifiability (desirable for decentralization) and controlled supply (desirable for monetary stability) appears to be fundamental.

Toward Practical Solutions. We surveyed several approaches to introducing scarcity:

- Static difficulty constraints, while simple, fail to adapt to technological progress.
- Verifiable delay functions offer theoretical promise but face quantum security challenges.
- Hybrid blockchain systems sacrifice full decentralization but may offer the best practical trade-off.
- Economic mechanisms (burning, demurrage) remain largely unexplored.

Broader Implications. The study of quantum money illuminates fundamental questions at the intersection of physics, computer science, and economics:

- What physical resources can serve as the basis for unforgeable tokens?
- How do cryptographic security definitions interact with economic desiderata?
- What is the minimal infrastructure needed to achieve both security and scarcity?

While quantum lightning may not immediately yield a practical blockchain-less cryptocurrency, its theoretical contributions—demonstrating that public verification is compatible with unclonability, and providing concrete constructions based on plausible assumptions—advance our understanding of what is possible at the quantum-classical boundary.

Future Directions. The most pressing open problems include: (1) constructing quantum-secure VDFs, (2) finding mechanisms for decentralized time consensus, (3) exploring alternative security models that naturally incorporate supply constraints, and (4) investigating whether quantum resources themselves (such as entanglement or quantum memory) can serve as the basis for scarcity.

As quantum computing technology matures, these questions will transition from purely theoretical concerns to practical engineering challenges. The foundations laid by Wiesner, Aaronson, Christiano, Zhandry, and others provide the conceptual tools needed to navigate this transition.

References

[1] S. Wiesner, "Conjugate coding," *SIGACT News*, vol. 15, no. 1, pp. 78–88, 1983. (Original manuscript circa 1970.)

[2] C. H. Bennett, G. Brassard, S. Breidbart, and S. Wiesner, "Quantum cryptography, or unforgeable subway tokens," in *Advances in Cryptology: Proceedings of Crypto '82*, pp. 267–275, 1982.

[3] S. Aaronson, "Quantum copy-protection and quantum money," in *Proceedings of the 24th Annual IEEE Conference on Computational Complexity (CCC)*, pp. 229–242, 2009.

[4] S. Aaronson and P. Christiano, "Quantum money from hidden subspaces," in *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 41–60, 2012.

[5] M. Zhandry, "Quantum lightning never strikes the same state twice. Or: quantum money from cryptographic assumptions," *Journal of Cryptology*, vol. 34, no. 1, article 8, 2021. (arXiv:1711.02276)

[6] A. Molina, T. Vidick, and J. Watrous, "Optimal counterfeiting attacks and generalizations for Wiesner's quantum money," in *Proceedings of the 7th Conference on*

Theory of Quantum Computation, Communication, and Cryptography (TQC), pp. 45–64, 2012.

[7] D. Unruh, "Computationally binding quantum commitments," in *Advances in Cryptology – EUROCRYPT 2016*, pp. 497–527, 2016.

[8] D. Boneh and V. Shoup, *A Graduate Course in Applied Cryptography*, Version 0.5, 2020. Available at: `https://toc.cryptobook.us/`

[9] S. Aaronson, "Introduction to Quantum Information Science," Lecture Notes, 2023.

[10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. Available at: `https://bitcoin.org/bitcoin.pdf`