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 that quantum lightning prevents *cloning*—no adversary can produce two bolts with the same serial number. However, this security guarantee does not address a distinct and equally important question: *can we limit how many bolts are created in total?* As we now demonstrate, the answer is fundamentally negative for any public-key scheme.

4.1 The Core Theorem: Unbounded Generation

The central result of this section shows that unlimited generation is not a bug but an inherent feature of public-key quantum money.

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

Proof. The algorithm simply invokes $\mathsf{Gen}(1^\lambda)$ independently N times. By correctness, each state passes verification with probability $\geq 1 - \epsilon$.

For distinctness, we use the fact that uniqueness implies high min-entropy of serial numbers:

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

The collision probability among $N = \text{poly}(\lambda)$ serial numbers is therefore:

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

□

4.2 Why This Is Unavoidable: Public Verification Implies Public Generation

One might hope that some clever protocol design could restrict who can generate money. The following proposition shows this is impossible in any public-key setting.

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

Proof. The generation algorithm Gen must be publicly specified—otherwise, how could the original issuer produce valid notes? Since Gen runs in polynomial time using only public operations (Hadamard gates, controlled unitaries, measurement), any party with a quantum computer can execute it. This contrasts fundamentally with private-key schemes, where $\mathsf{Gen}_{\text{private}}(k, s)$ requires a secret key k held only by the bank. □

4.3 The Cloning-Generation Dichotomy

Combining the above results, we see a fundamental asymmetry in quantum money security:

- **Targeted generation (Cloning) is intractable.** To clone a bolt with serial number y, an adversary is forced to solve a specific instance of the multi-collision problem. Specifically, they must produce a fresh batch of $k + 1$ tensor copies of $|\psi_y\rangle$ to pass verification. Combined with the original bolt, this would yield $2(k+1)$ colliding inputs for y. The NAMCR assumption posits that finding such a large collision set for a *fixed* output y is computationally impossible.

- **Random generation (Minting) is trivial.** The generation algorithm Storm operates without a target constraint. It samples random inputs which map to a random output y'. Because the range of the hash function is exponentially large (2^n), the probability of hitting any previously generated serial number is negligible. Thus, generating *new* money requires no collision-finding effort; it merely requires running the forward circuit, which is efficient for anyone.

The uniqueness property guarantees that for any QPT adversary:

$$\Pr\left[\mathsf{Ver}(\rho_1) = \mathsf{Ver}(\rho_2) \neq \perp\right] = \mathrm{negl}(\lambda).$$

But this says nothing about generating N *distinct* valid notes, which succeeds with overwhelming probability by Theorem **??**.

In economic terms: quantum lightning perfectly prevents counterfeiting (copying existing money) but provides no mechanism to prevent inflation (creating new money). Any party can become a "mint" simply by running the public Storm algorithm.

5 Prospective Approaches to Supply Limitation

Several cryptographic approaches could potentially introduce supply constraints:

Hash-based difficulty. Require $H(y) < T$ for the serial number, converting generation into probabilistic search. Grover's algorithm provides $O(2^{d/2})$ speedup over classical $O(2^d)$ trials.

Verifiable Delay Functions. VDFs certify that time T has elapsed, preventing parallel mining. Quantum security of current VDF constructions remains open.

Quantum memory bounds. Exploit physical scarcity of coherent quantum storage. Verification of actual storage (vs. regeneration) is an open problem.

Entanglement-based certificates. Use monogamy of entanglement to bound supply, but this reintroduces trusted authorities.

Remark 5.1 (Open Problem)**.** Can we construct public-key quantum money where generation (not just cloning) is computationally hard? This requires making the search problem "Find ρ such that $\mathsf{Ver}(\rho) \neq \perp$" hard while keeping verification efficient.

6 Conclusion

We have analyzed the fundamental tension in public-key quantum money between *unclonability* and *unlimited generation*:

- **Cloning is hard:** The no-cloning theorem combined with NAMCR ensures duplicating valid bolts is infeasible.
- **Generation is easy:** Public verification implies public generation—any party can produce fresh bolts in polynomial time.
- **Inflation is inevitable:** Unlimited generation leads to unbounded supply, undermining scarcity-based currency applications.

This asymmetry implies that public-key quantum money, despite its strong anti-counterfeiting guarantees, cannot serve as a scarcity-based currency: any party can generate polynomially many valid coins, leading to unbounded inflation. While approaches such as hash-based difficulty or VDFs can slow generation, none fundamentally resolve this limitation.

This dichotomy is inherent to any public-key scheme. Quantum lightning shows that "lightning never strikes the same state twice," but also that lightning can strike *anywhere*. Constructing public-key quantum money where generation itself is hard remains a central open problem in quantum cryptography.

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`