Unitary synthesis and unclonable cryptography

John Bostanci

In a world with quantum computers, we can imagine problems where our inputs and outputs are quantum states instead of classical strings.

In a world with quantum computers, we can imagine problems where our inputs and outputs are quantum states instead of classical strings.

Studying these kinds of problems had led to a number of new exciting open questions in complexity theory and cryptography.

In a world with quantum computers, we can imagine problems where our inputs and outputs are quantum states instead of classical strings.

Studying these kinds of problems had led to a number of new exciting open questions in complexity theory and cryptography.

I’m going to tell you about two big areas related to this: the unitary synthesis problem, and unclonable cryptography.

Quantum preliminaries

Quantum preliminaries

An n-qubit state is a norm 1, 2^n dimensional vector of complex numbers.

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$$

$$\| |\psi\rangle \| = \sqrt{\langle \psi | \psi \rangle} = 1$$

Quantum preliminaries

An n-qubit state is a norm 1, 2^n dimensional vector of complex numbers.

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$$

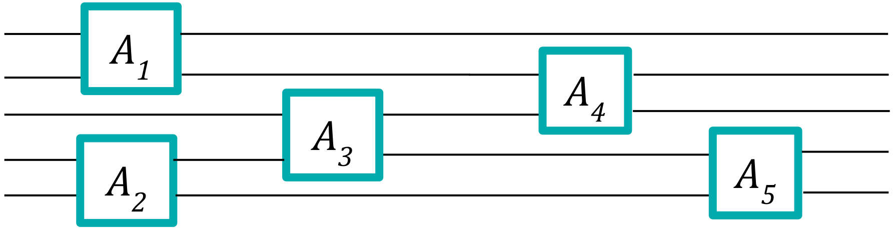
$$\||\psi\rangle\| = \sqrt{\langle\psi|\psi\rangle} = 1$$

An n-qubit unitary is a $2^n \times 2^n$ norm preserving matrix.

$$\|U|\psi\rangle\| = \||\psi\rangle\|$$

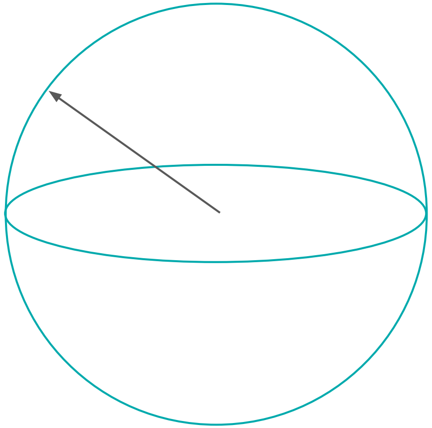
Quantum preliminaries

Efficient quantum computation is a poly(n) sized quantum circuit consisting of a sequence of two-qubit unitary gates.



Quantum preliminaries

A “random” state refers to a Haar random vector from the unit ball in $\mathbb{C}^{2^n}$.



The unitary synthesis problem

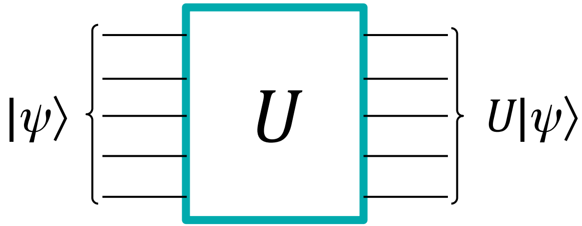
Motivating question:

Can we relate the complexity of “quantum problems” to the complexity of “classical problems”?

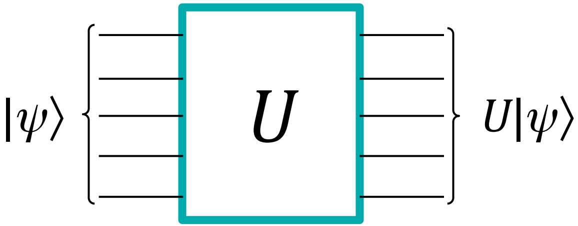
Classical problems can typically be reduced to the task of computing some function,

$$f : \{0, 1\}^* \rightarrow \{0, 1\}^*$$

The most general transformation that can be implemented on a quantum computer is called a unitary transformation.

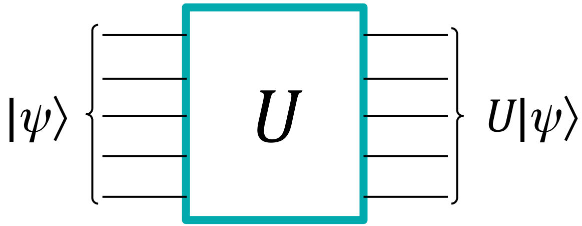


The most general transformation that can be implemented on a quantum computer is called a unitary transformation.



Question [AK'07]: Can we efficiently reduce every unitary to ***some*** function f?

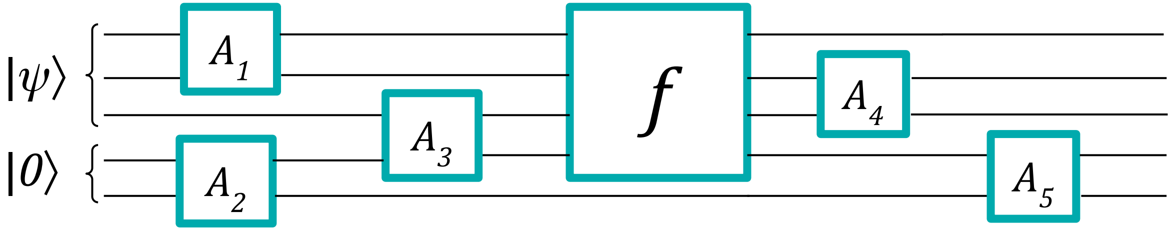
The most general transformation that can be implemented on a quantum computer is called a unitary transformation.



Question [AK'07]: Can we efficiently reduce every unitary to ***some*** function f?

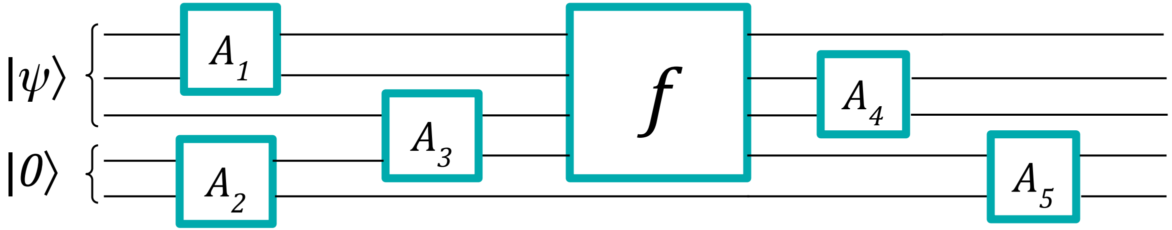
Query algorithms

A query algorithm A^f is a sequence of unitaries, with superposition queries to the function f in between.



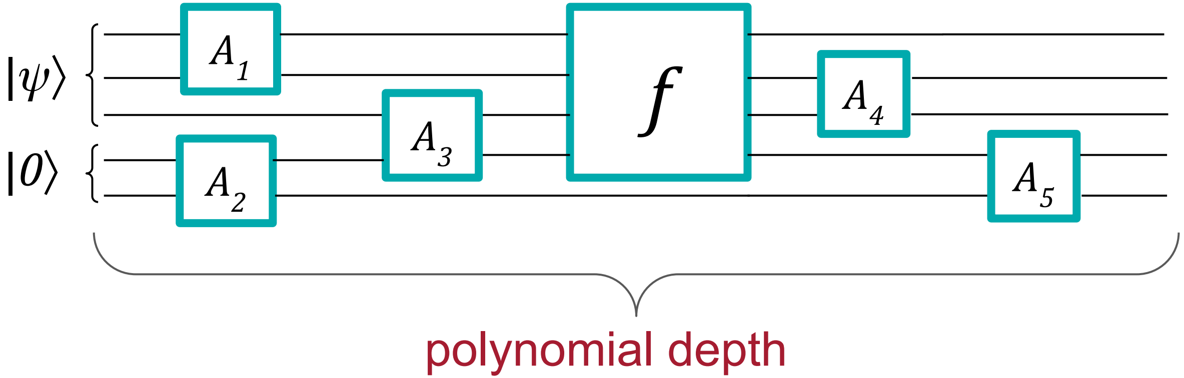
Unitary synthesis

Is there a universal, efficient, query algorithm A, such that for any unitary U, there exists a function f such that A^f implements U?



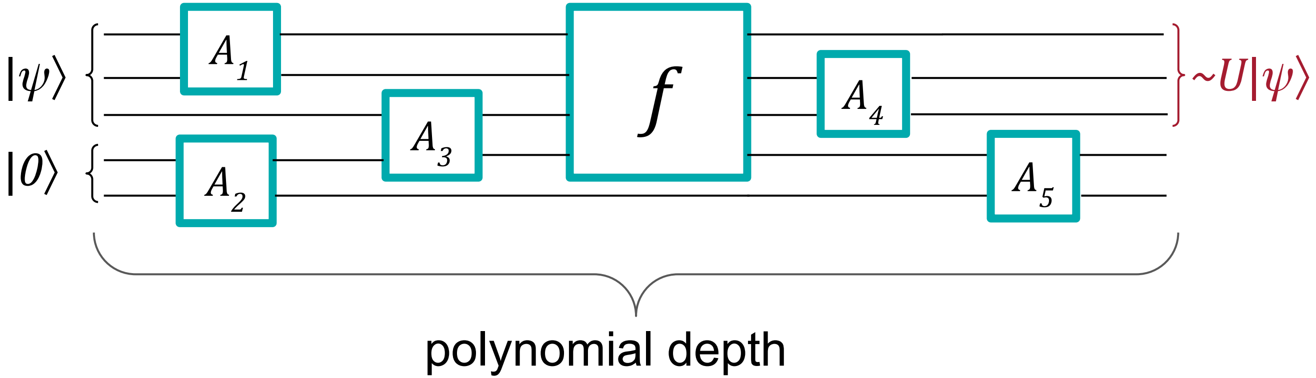
Unitary synthesis

Is there a universal, efficient, query algorithm A, such that for any unitary U, there exists a function f such that A^f implements U?



Unitary synthesis

Is there a universal, efficient, query algorithm A, such that for any unitary U, there exists a function f such that A^f implements U?



Unitary synthesis and cryptography

Both directions of the unitary synthesis problem seem to have connections to quantum cryptography.

Single-copy pseudo-random states

A single-copy pseudo-random state family is a keyed collection of states $\{|\psi_{n,k}\rangle\}_{n,k}$ such that,

Single-copy pseudo-random states

A single-copy pseudo-random state family is a keyed collection of states $\{|\psi_{n,k}\rangle\}_{n,k}$ such that,

- (Efficiency) There is an efficient algorithm that outputs $|\psi_{n,k}\rangle$ when run on input $(1^n, k)$ for n-bit key k.

Single-copy pseudo-random states

A single-copy pseudo-random state family is a keyed collection of states $\{|\psi_{n,k}\rangle\}_{n,k}$ such that,

- (Efficiency) There is an efficient algorithm that outputs $|\psi_{n,k}\rangle$ when run on input $(1^n, k)$ for n-bit key k.
- (Stretch) The number of qubits of $|\psi_{n,k}\rangle$ is greater than n.

Single-copy pseudo-random states

A single-copy pseudo-random state family is a keyed collection of states $\{|\psi_{n,k}\rangle\}_{n,k}$ such that,

- (Efficiency) There is an efficient algorithm that outputs $|\psi_{n,k}\rangle$ when run on input $(1^n, k)$ for n-bit key k.
- (Stretch) The number of qubits of $|\psi_{n,k}\rangle$ is greater than n.
- (Pseudo-randomness) For all efficient adversaries A, the following holds

$$|\mathrm{Pr}_{\mathrm{k}}[A(|\psi_{n,k}\rangle) \text{ accepts}] - \mathrm{Pr}_{\psi}[A(|\psi\rangle \text{ accepts})]| = \mathrm{negl(n)}$$

Unitary synthesis and cryptography

What computational power does the adversary A need in order to break the pseudo-randomness property of a single-copy pseudo-random state?

Unitary synthesis and cryptography

What computational power does the adversary A need in order to break the pseudo-randomness property of a single-copy pseudo-random state?

- An NP oracle?
- A PP oracle?
- A RE oracle?

Unitary synthesis and cryptography

What computational power does the adversary A need in order to break the pseudo-randomness property of a single-copy pseudo-random state?

Currently we do not know.

Unitary synthesis and cryptography

What computational power does the adversary A need in order to break the pseudo-randomness property of a single-copy pseudo-random state?

Currently we do not know.

If the unitary synthesis problem is resolved in the negative, then there exists constructions relative to oracles that can not be broken by an efficient adversary given oracle access to any classical problem.

Unitary synthesis: what is known?

Not much is known about the complexity of unitary synthesis.

Unitary synthesis: what is known?

Not much is known about the complexity of unitary synthesis.

Lower bound: You need at least 2 queries to the function [LMW'24].

Unitary synthesis: what is known?

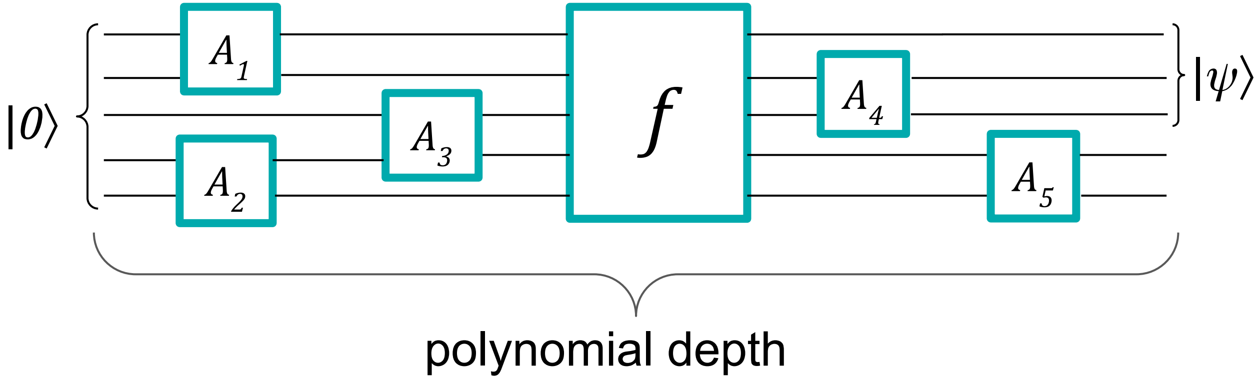
Not much is known about the complexity of unitary synthesis.

Lower bound: You need at least 2 queries to the function [LMW’24].

Upper bound: You only need $2^{n/2}$ queries to the function [Rosenthal’21].

Related problem: state synthesis

In state synthesis, we want to design a fixed query algorithm that, on input 0^n, queries a function to prepare a state $|\psi\rangle$.



Related problem: state synthesis

An efficient state synthesis algorithm (making O(n) queries) has been known since at least 2016.

Related problem: state synthesis

An efficient state synthesis algorithm (making O(n) queries) has been known since at least 2016.

We know how to do state synthesis efficiently only using a single query to a function [Rosenthal'23, INNRY'22].

Unitary synthesis: what’s next?

There is no real consensus about what the resolution to the unitary synthesis problem *should* be.

Unitary synthesis: what's next?

There is no real consensus about what the resolution to the unitary synthesis problem *should* be.

I personally think you shouldn't be able to synthesize unitaries, but proving this is challenging, so I can't say for sure how the proof would go.

Unitary synthesis: what's next?

There is no real consensus about what the resolution to the unitary synthesis problem *should* be.

I personally think you shouldn't be able to synthesize unitaries, but proving this is challenging, so I can't say for sure how the proof would go.

Either way will be exciting and tell us about the interplay between quantum and classical complexity theory!

Unclonable cryptography

No-cloning

The no-cloning theorem says that there is no algorithm that clones an unknown quantum state.

$$|\psi\rangle\,|0\rangle \not\to |\psi\rangle\,|\psi\rangle$$

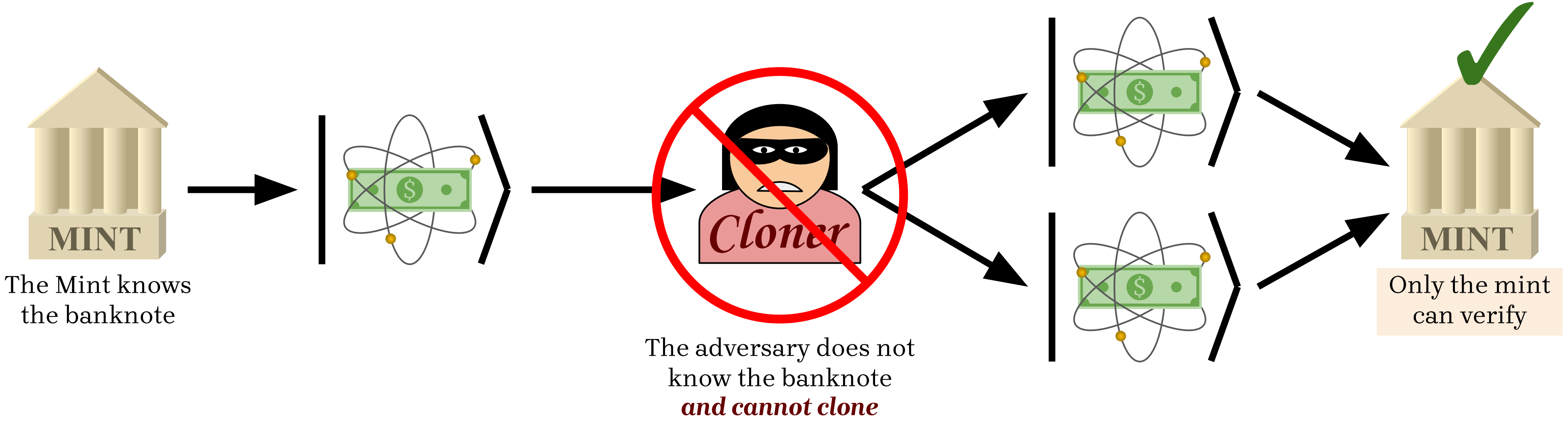
No-cloning

The no-cloning theorem says that there is no algorithm that clones an unknown quantum state.

$$|\psi\rangle\,|0\rangle \not\to |\psi\rangle\,|\psi\rangle$$

In the 1970’s, Stephen Wiesner imagined a use case: Money that no one can copy.

Private-key quantm money [Wiesner'83]



Credits: Barak Nehoran

Private-key quantum money [Wiesner'83]

A private-key quantum money scheme consists of the following algorithms:

- $\mathrm{Gen}(1^{\mathrm{n}}) \rightarrow (sk, s, |\psi_s\rangle)$
- $\mathrm{Ver}(sk, s, |\psi_s\rangle) \rightarrow \{\mathrm{accept}, \mathrm{reject}\}$

Private-key quantum money [Wiesner'83]

A private-key quantum money scheme consists of the following algorithms:

- $\mathrm{Gen}(1^{\mathrm{n}}) \rightarrow (sk, s, |\psi_s\rangle)$.
- $\mathrm{Ver}(sk, s, |\psi_s\rangle) \rightarrow \{\text{accept, reject}\}$

The scheme is secure if, for all adversaries A, the following holds.

$$\Pr\left[\mathrm{Ver}(\mathrm{sk}, \mathrm{s}, |\psi_1\rangle) \text{ and } \mathrm{Ver}(\mathrm{sk}, \mathrm{s}, |\psi_2\rangle) \;\middle|\; \begin{array}{c} (sk, s, |\psi_s\rangle) \leftarrow \mathrm{Gen}(1^{\mathrm{n}}), \\ (|\psi_1\rangle, |\psi_2\rangle) \leftarrow \mathrm{A}(s, |\psi_s\rangle) \end{array}\right] = \mathrm{negl}(\mathrm{n})$$

Private-key quantum money [Wiesner'83]

Wiesner proved that you could get private-key quantum money with information theoretic security (against all adversaries, no assumptions needed)!

Private-key quantum money [Wiesner’83]

Wiesner proved that you could get private-key quantum money with information theoretic security (against all adversaries, no assumptions needed)!

However it had a number of problems, the one we will focus on is the so-called “verifiability” problem.

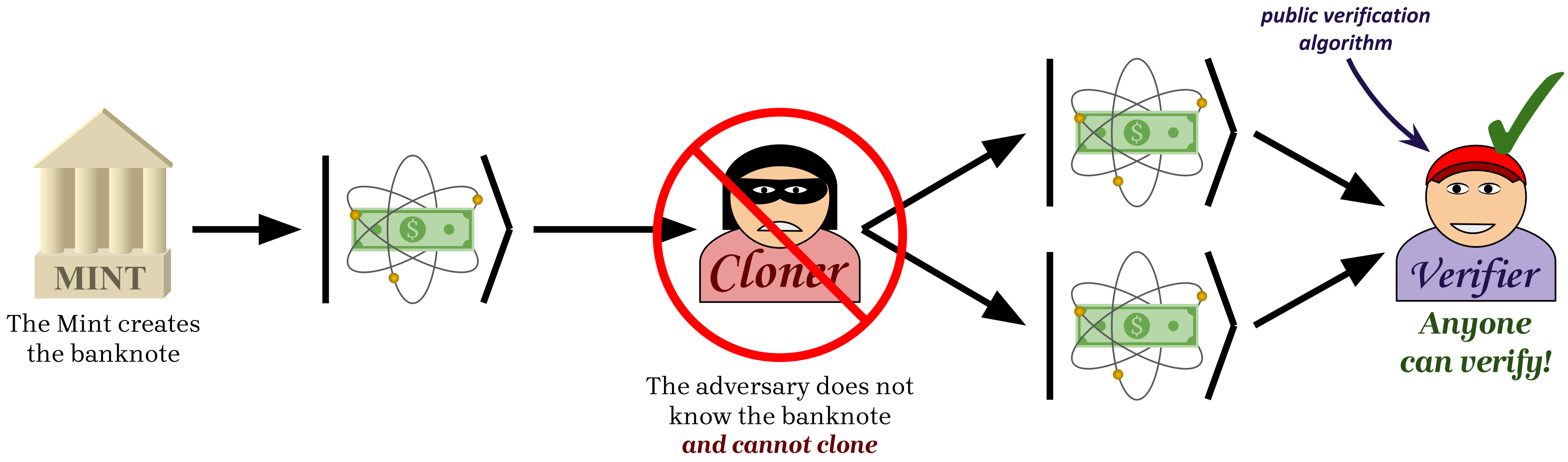
Private-key quantum money [Wiesner'83]

Wiesner proved that you could get private-key quantum money with information theoretic security (against all adversaries, no assumptions needed)!

However it had a number of problems, the one we will focus on is the so-called "verifiability" problem.

This inspired "public-key" quantum money.

Public-key quantum money [Aaronson'09]



Credits: Barak Nehoran

Public-key quantum money [Aaronson'09]

A public-key quantum money scheme consists of the following algorithms:

- $\mathrm{KeyGen}(1^{\mathrm{n}}) \rightarrow (sk, pk)$
- $\mathrm{Mint}(sk) \rightarrow (s, |\psi_s\rangle)$
- $\mathrm{Ver}(pk, s, |\psi_s\rangle) \rightarrow \{\mathrm{accept}, \mathrm{reject}\}$

Public-key quantum money [Aaronson'09]

A public-key quantum money scheme consists of the following algorithms:

- $\mathrm{KeyGen}(1^{\mathrm{n}}) \rightarrow (sk, pk)$
- $\mathrm{Mint}(sk) \rightarrow (s, |\psi_s\rangle)$
- $\mathrm{Ver}(pk, s, |\psi_s\rangle) \rightarrow \{\mathrm{accept}, \mathrm{reject}\}$

The scheme is secure if, for all adversaries A, the following holds.

$$\Pr\left[\mathrm{Ver}(\mathrm{sk}, \mathrm{s}, |\psi_1\rangle) \text{ and } \mathrm{Ver}(\mathrm{sk}, \mathrm{s}, |\psi_2\rangle) \;\middle|\; \begin{array}{c} (sk, pk, |\psi_s\rangle) \leftarrow \mathrm{KeyGen}(1^{\mathrm{n}}), \\ (s, |\psi_s\rangle) \leftarrow \mathrm{Mint}(sk) \\ (|\psi_1\rangle, |\psi_2\rangle) \leftarrow \mathrm{A}(pk, s, |\psi_s\rangle) \end{array} \right] = \mathrm{negl}(\mathrm{n})$$

Quantum lightning [Zhandry'17]

A quantum lightning scheme consists of the following algorithms:

- $\mathrm{KeyGen}(1^{\mathrm{n}}) \rightarrow (sk, pk)$
- $\mathrm{Mint}(sk) \rightarrow (s, |\psi_s\rangle)$
- $\mathrm{Ver}(pk, s, |\psi_s\rangle) \rightarrow \{\mathrm{accept}, \mathrm{reject}\}$

The scheme is secure if, for all adversaries A, the following holds.

$$\Pr\left[\mathrm{Ver}(\mathrm{sk}, \mathrm{s}, |\psi_1\rangle) \text{ and } \mathrm{Ver}(\mathrm{sk}, \mathrm{s}, |\psi_2\rangle) \,\middle|\, (\mathrm{s}, |\psi_1\rangle, |\psi_2\rangle) \leftarrow \mathrm{A}(1^n)\right] = \mathrm{negl}(\mathrm{n})$$

Public-key quantum money

Quantum money and lightning have been notoriously difficult to construct!

Merely conjectured

[Aaronson'09]: random stabilizer states

broken by [Lutomirski-Aaronson-Farhi-Gosset-Hassidim-Kelner-Shor'10]

[Farhi-Gosset-Hassidim-Lutomirski-Shor'10]: knots

[Aaronson-Christiano'12]: polynomials hiding subspaces

broken by [Pena-Faugère-Perret'14, Christiano-Sattath'16]

[Zhandry'19]: quadratic systems of equations

broken by [Roberts'21]

[Kane'18, Kane-Sharif-Silverberg'21]: quaternion algebras

[Khesin-Lu-Shor'22]: lattices

broken by [Liu-Montgomery-Zhandry'23]

Black box / oracle proof

[Aaronson'09]: quantum oracle

[Aaronson-Christiano'12]: classical hidden subspaces oracle

[Kane'18, Kane-Sharif-Silverberg'21]: Commuting unitaries

[Liu-Montgomery-Zhandry'23]: Walkable invariants

[Zhandry'23]: Abelian group actions

Black-box proofs may break in plain model

Reduction from a widely-studied assumption

[Zhandry'19]: indistinguishability obfuscation

Very strong assumption! quantum security unclear

No known security proof for quantum lightning in the plain model

Chart adapted from [Zhandry'23, Nehoran'24]

Beyond quantum money?

Public-key quantum money asks for unclonable states that are also publicly verifiable. But…

Beyond quantum money?

Public-key quantum money asks for unclonable states that are also publicly verifiable. But…

We might also want unclonable states that

- Are an encryption of some classical data (Unclonable encryption [BL'19, AK'21, AKLLZ'22, AKL'23, AB'24])

Beyond quantum money?

Public-key quantum money asks for unclonable states that are also publicly verifiable. But…

We might also want unclonable states that

- Are an encryption of some classical data (Unclonable encryption [BL’19, AK’21, AKLLZ’22, AKL’23, AB’24])
- Act as signature keys (Tokenized signatures [BS’16, Shmueli’22])

Beyond quantum money?

Public-key quantum money asks for unclonable states that are also publicly verifiable. But…

We might also want unclonable states that

- Are an encryption of some classical data (Unclonable encryption [BL'19, AK'21, AKLLZ'22, AKL'23, AB'24])
- Act as signature keys (Tokenized signatures [BS'16, Shmueli'22])
- Allow us to execute arbitrary functions (Copy-protected software [Aaronson'09, AL'21, ALLZZ'21, BJLPS'21, CMP'24])

Unclonable cryptography today

Despite lots of attention, we know little about unclonable cryptography!

Unclonable cryptography today

Despite lots of attention, we know little about unclonable cryptography!

The only successful instantiations have been in settings without public verification (private-key quantum money, unclonable encryption) where we can achieve information theoretic security using Wiesner's original construction.

Unclonable cryptography today

Despite lots of attention, we know little about unclonable cryptography!

The only successful instantiations have been in settings without public verification (private-key quantum money, unclonable encryption) where we can achieve information theoretic security using Wiesner's original construction.

Constructing public-key quantum money in the plain model and proving security from a "well-founded" assumption is a major open problem!