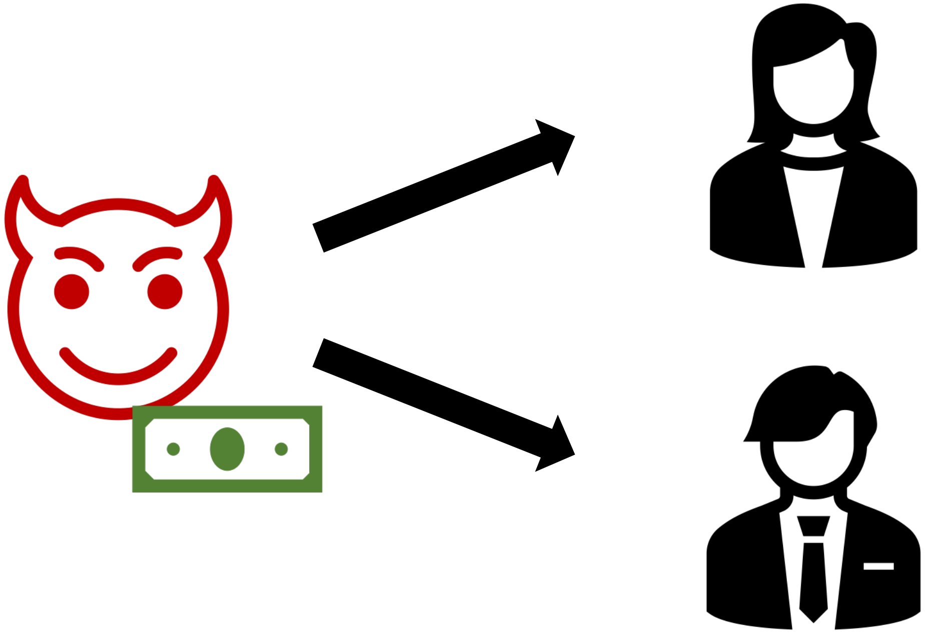
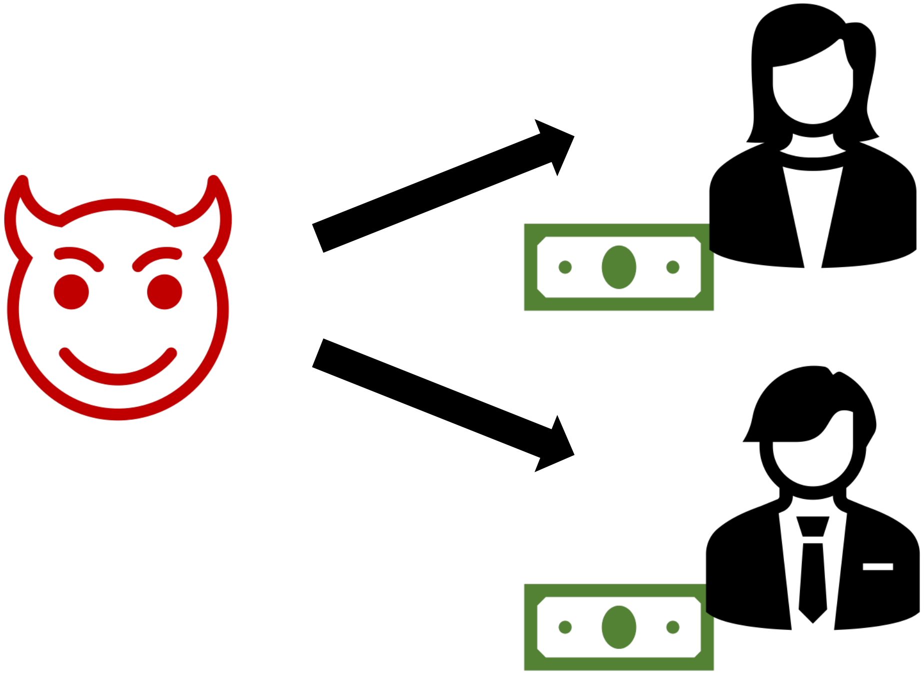
On Quantum Money and Evasive Obfuscation

Mark Zhandry (NTT Research)

The Double-Spend Problem with Digital Currency



The Double-Spend Problem with Digital Currency



Any classical solution needs some coordination between Alice and Bob (possibly involving third party)

Enter quantum…

Quantum no-cloning

[Park’70, Wooters-Zurek’82, Dieks’82]

$$|\psi\rangle \cancel{\rightarrow |\psi\rangle} |\psi\rangle$$

Secret key Quantum Money

[Wiesner'70]

$$💵 = |\psi\rangle$$

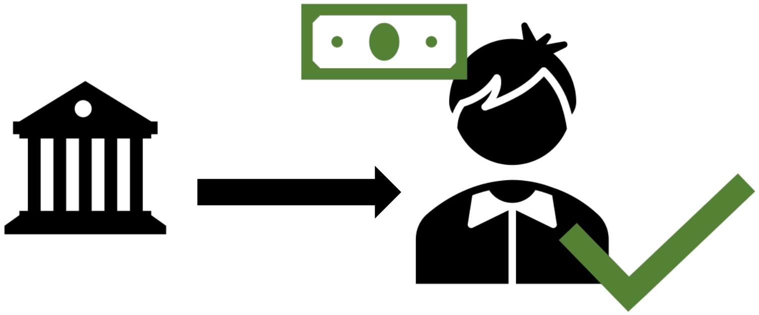
Unfortunately, mint required to verify money, so still need coordination

Public Key Quantum Money

[Aaronson'09]

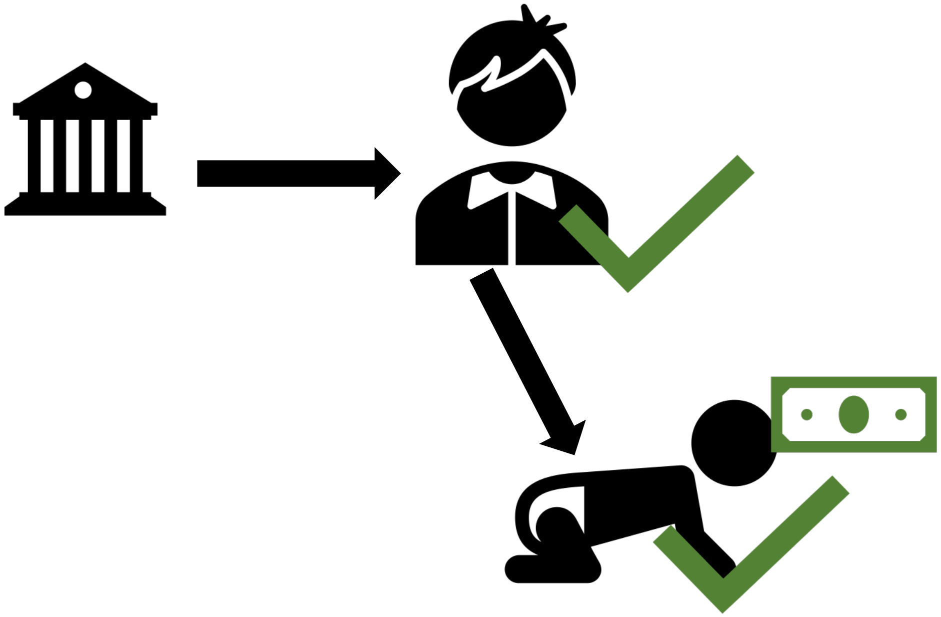
Public Key Quantum Money

[Aaronson'09]



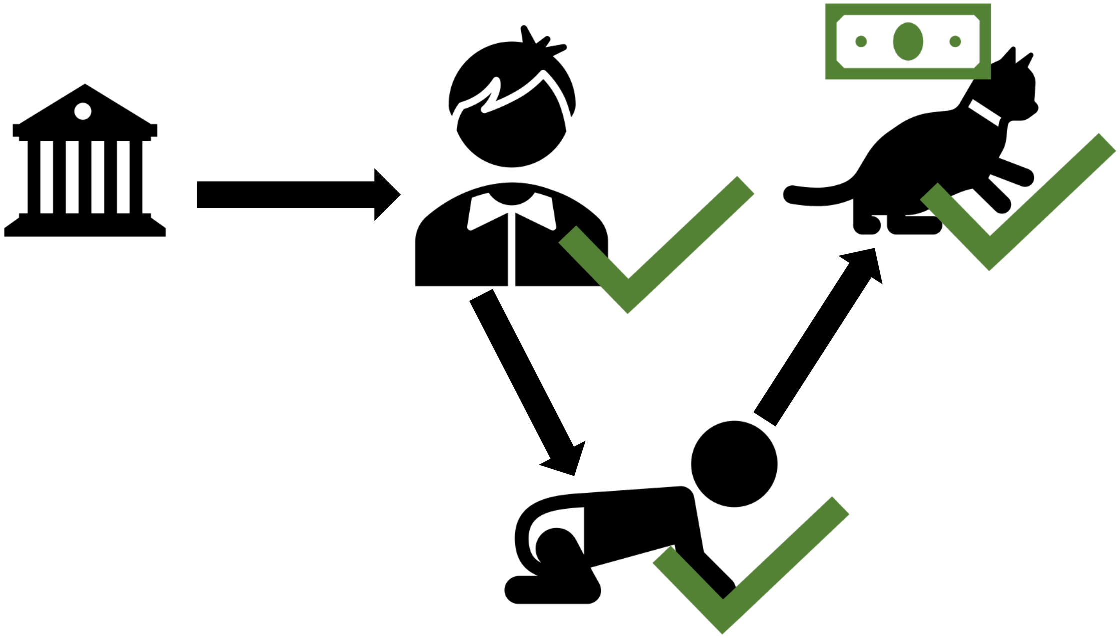
Public Key Quantum Money

[Aaronson'09]



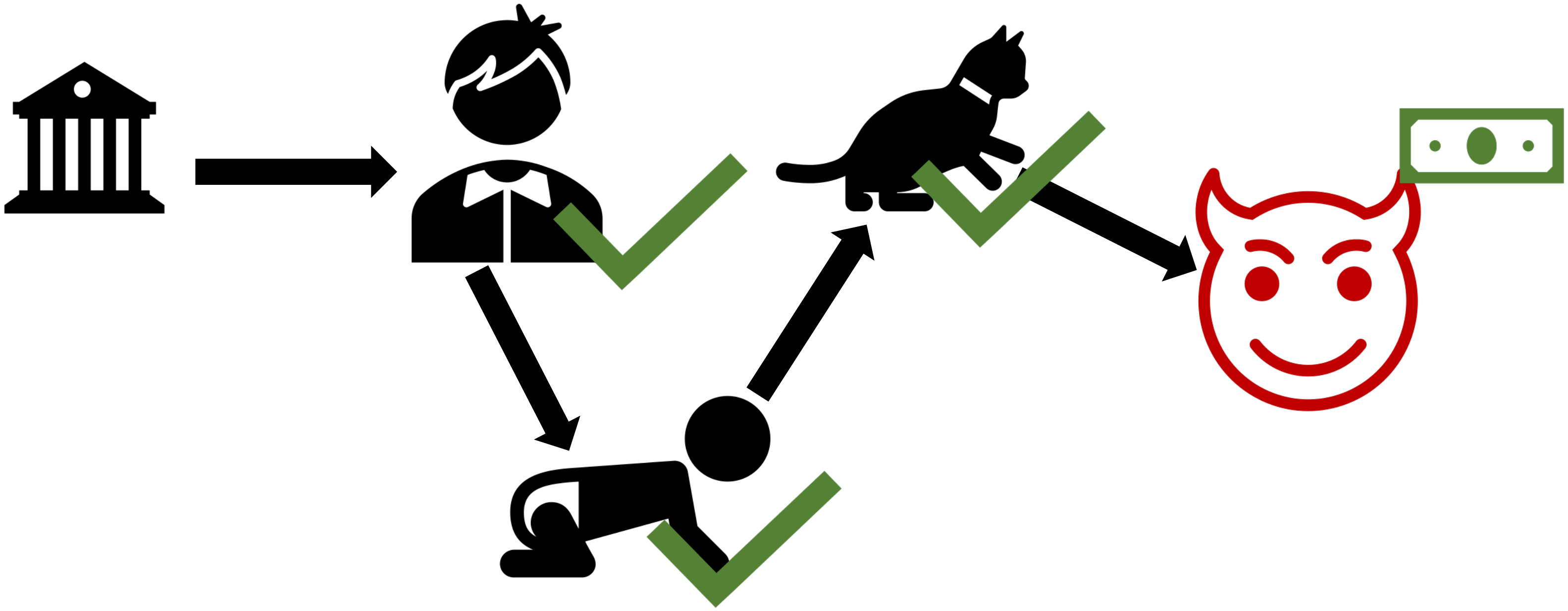
Public Key Quantum Money

[Aaronson'09]



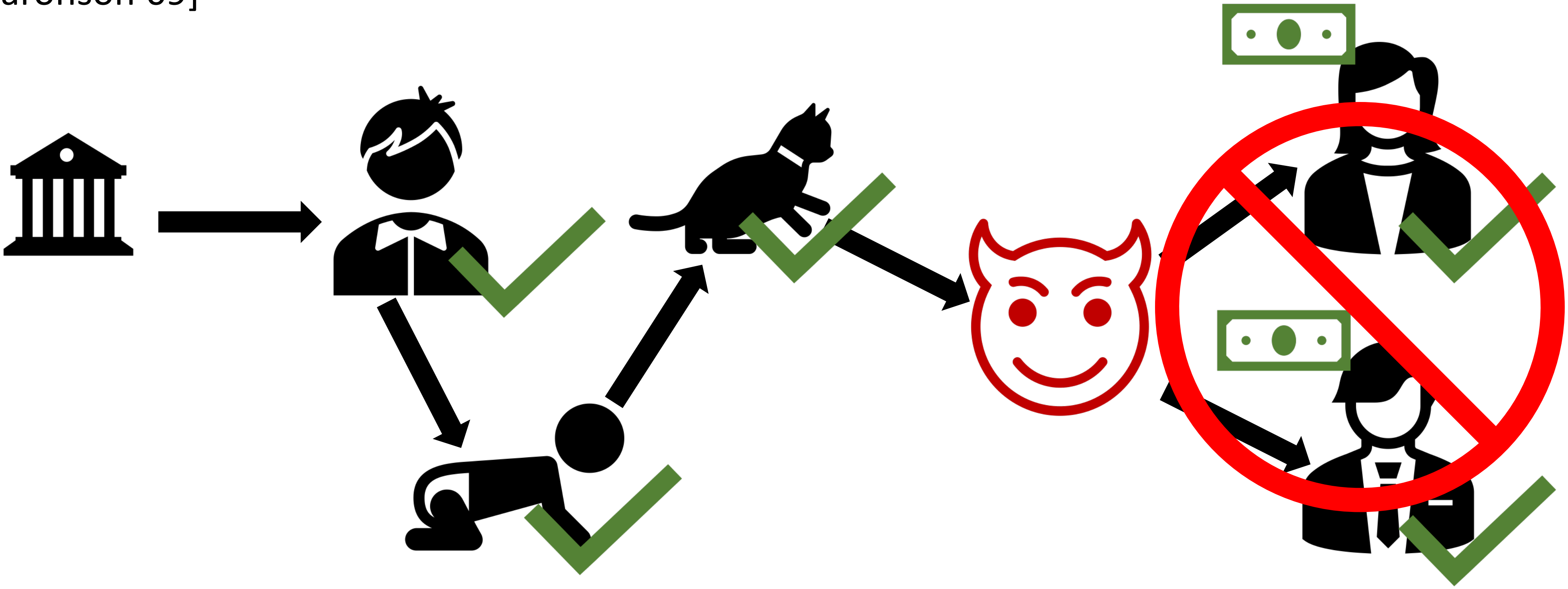
Public Key Quantum Money

[Aaronson'09]



Public Key Quantum Money

[Aaronson'09]



PK Quantum money is a central object in the study of quantum protocols

PK Quantum Money is Notoriously Difficult!

[Aaronson’09]: random stabilizer states

✗ [Lutomirski-Aaronson-Farhi-Gosset-Hassidim-Kelner-Shor’10]

[Aaronson-Christiano’12]: polynomials hiding subspaces

✗ [Pena-Faugère-Perret’14, Christiano-Sattath’16]

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

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

[Z’19]: quadratic systems of equations

✗ [Roberts’21]

[Z’19]: indistinguishability obfuscation

[Khesin-Lu-Shor’22]: lattices

✗ [Liu-Montgomery-Z’23]

[Liu-Montgomery-Z’23]: Walkable invariants

[Z’24]: abelian group actions

[Bostanci-Nehoran-Z’24]: non-abelian group actions

PK Quantum Money is Notoriously Difficult!

Only scheme with provable security under assumptions studied by wider crypto community. But use of iO is undesirable

’12]: polynomials hiding subspaces

ere-Perret’14, Christiano-Sattath’16]

[Kane’1 Sharif-Silverberg’21]: quaternion algebras

[Z’19]: quadratic systems of equations

✗ [Roberts’21]

[Z’19]: indistinguishability obfuscation

[Khesin-Lu-Shor’22]: lattices

✗ [Liu-Montgomery-Z’23]

[Liu-Montgomery-Z’23]: Walkable invariants

[Z’24]: abelian group actions

[Bostanci-Nehoran-Z’24]: non-abelian group actions

Can Evasive Obfuscation Suffice?

Evasive obfuscation = Secure as long as adversary can't find accepting input

Thm [Goyal-Koppula-Waters'17, Wichs-Zirdelis'17]: LWE ➔ obfuscation for certain evasive functions

In classical world, a number of results showing how to base iO applications on milder tools, especially LWE. Often (perhaps implicitly) go through route of obfuscating evasive functions

Can Evasive Obfuscation Suffice?

[Z’19] is *almost* evasive

(building on [Aaronson-Christiano’12, Ben-David-Sattath’16])

Obfuscate random subspace $S, S^{\perp}$

On their own, evasive except for un-interesting point at origin

But…

💵 allows adversary to find one input in either S or $S^{\perp}$

Our Result

Thm [this work]: PK Qua
box based on evasive obfuscation, posing:
al obfuscation queries
cation queries (but possibly
es quantum evaluation queries)

Very natural restrictions that capture essentially all known applications of obfuscation to quantum protocols

Rough dual to [Ananth-Hu-Yuen’23], who prove impossibility when the *verifier* makes classical queries

Cor [this work] (informal): PK Quantum Money cannot be black-box based on one-way functions, supposing the mint only makes classical queries to the OWF and the verifier is “natural”

Step 1: Oracles for evasive obfuscation

O maps circuits
represents obfuscat

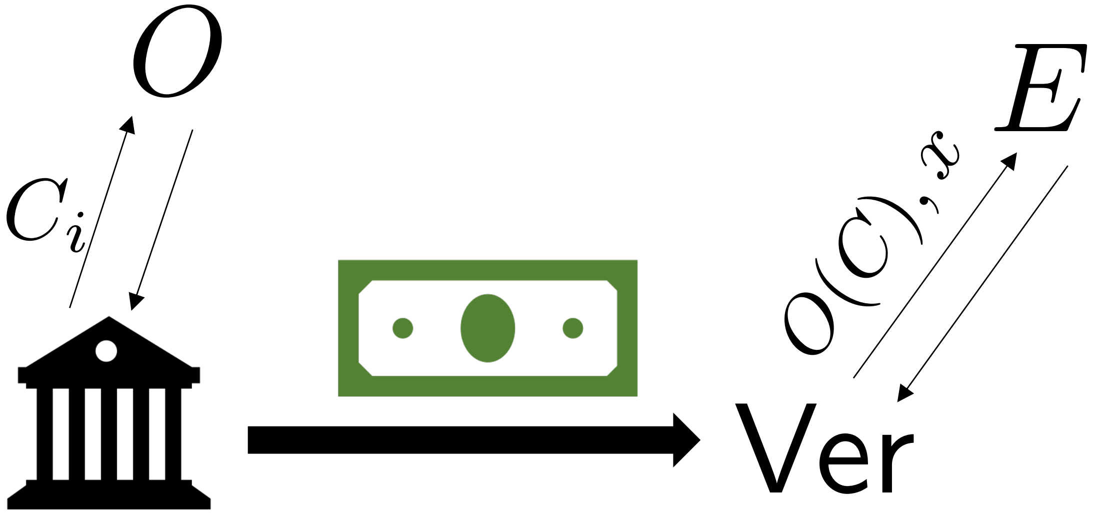
Ensures obfuscation is totally broken if any accepting input is known

$$E(\ O(C)\ , x) = \begin{cases} C & \text{if } C(x) = 1 \\ \bot & \text{if } C(x) = 0 \end{cases}$$

Ignore computation, only count queries

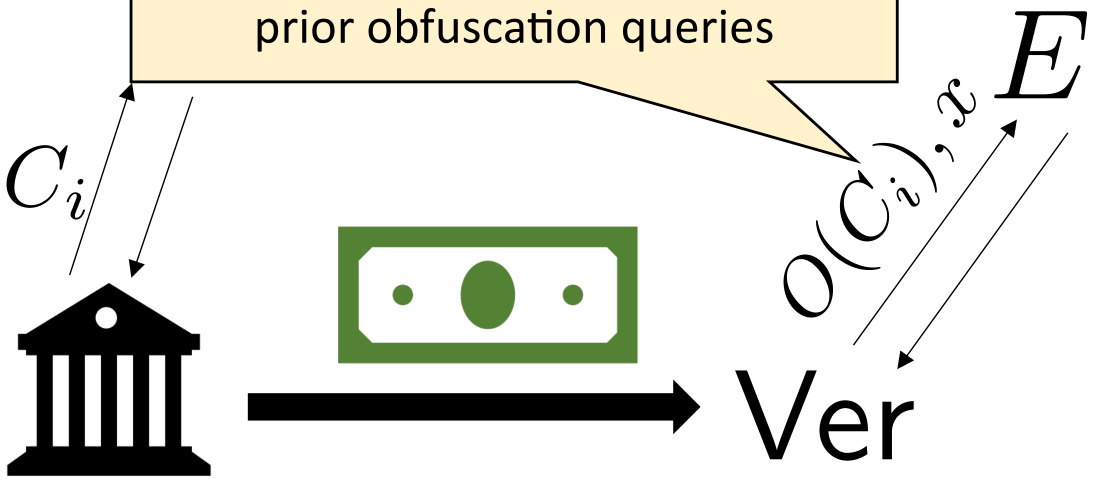
Lem [this work] (informal): Any reasonable notion of evasive obfuscation is captured by this oracle

Step 2: Use Oracle To Break Quantum Money

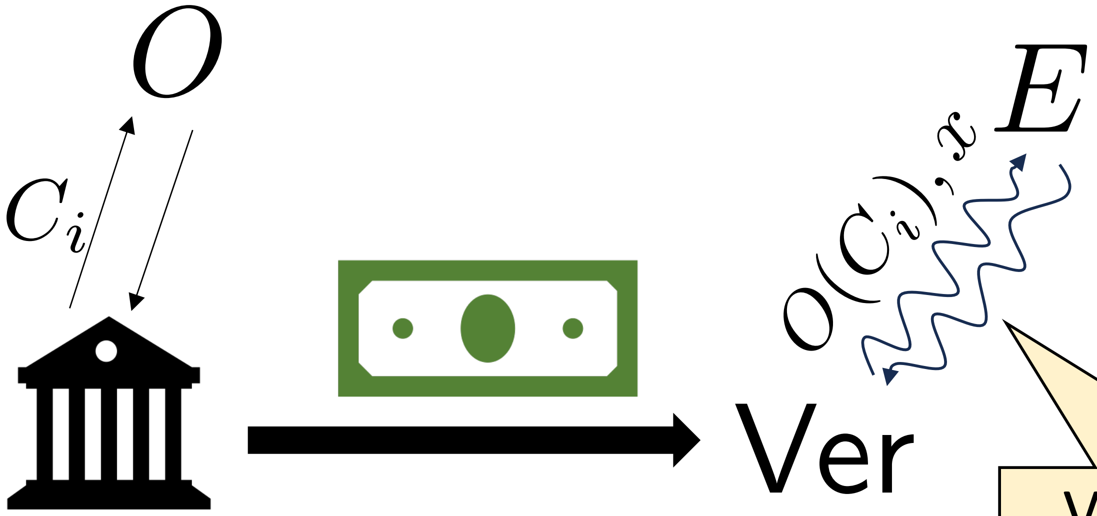


Step 2: Use Oracle To Break Quantum Money

Can assume wlog that only queries to verifier were responses from prior obfuscation queries



Step 2: Use Oracle To Break Quantum Money

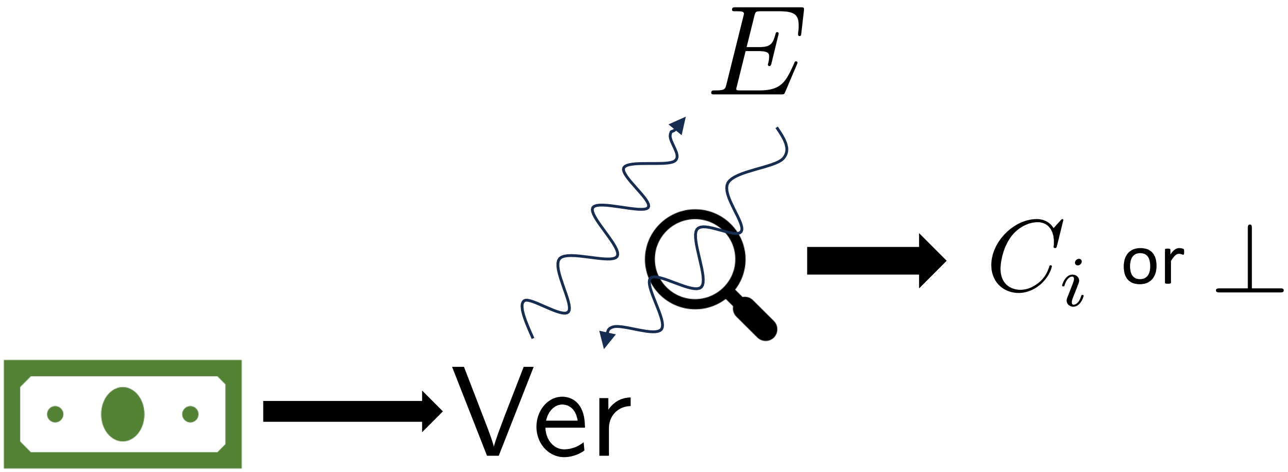


Verifiers queries can be quantum

Observation: If adversary can compute all C_i, scheme broken

Step 2: Use Oracle To Break Quantum Money

The attack idea:



Step 2: Use Oracle To Break Quantum Money

Assume for now a single C_i

Case 1: Measuring query gives C_i with non-negl prob.

➡ scheme broken

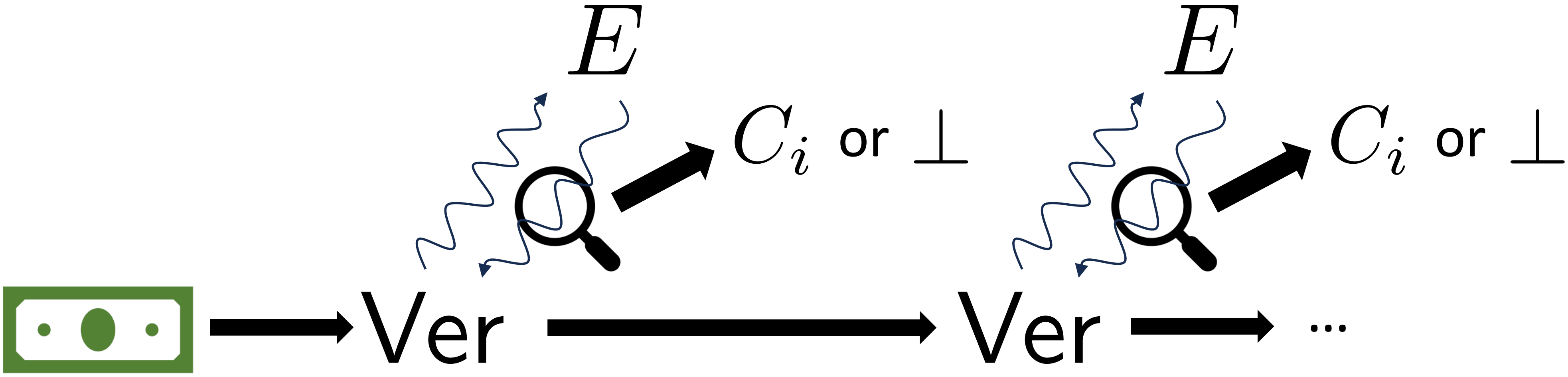
Case 2: Measuring query gives $\perp$ with overwhelming prob.

➡ Can answer E queries for ourselves (just output $\perp$)

➡ Oracle useless, so scheme broken

Step 2: Use Oracle To Break Quantum Money

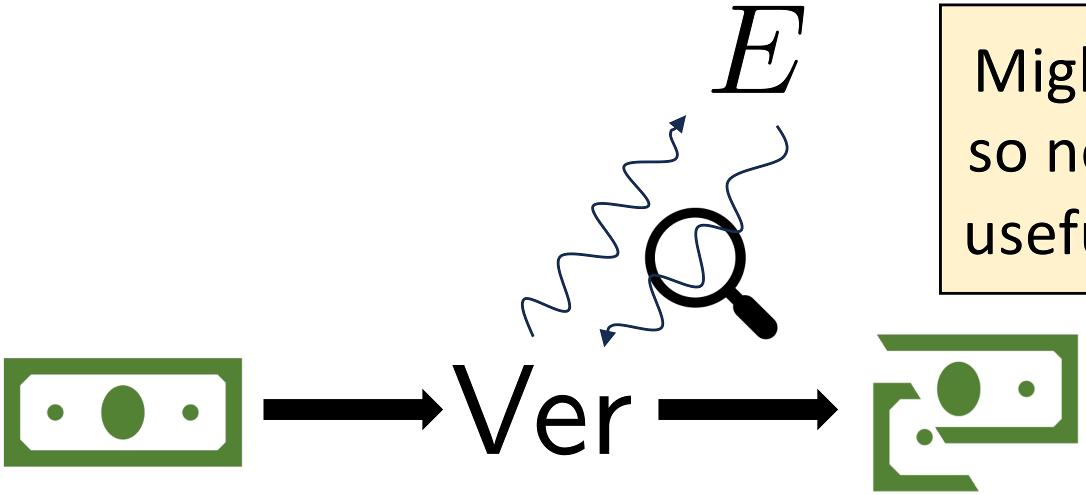
The attack idea (many C_i):



Hope: eventually pick up all C_i or queries useless

Step 2: Use Oracle To Break Quantum Money

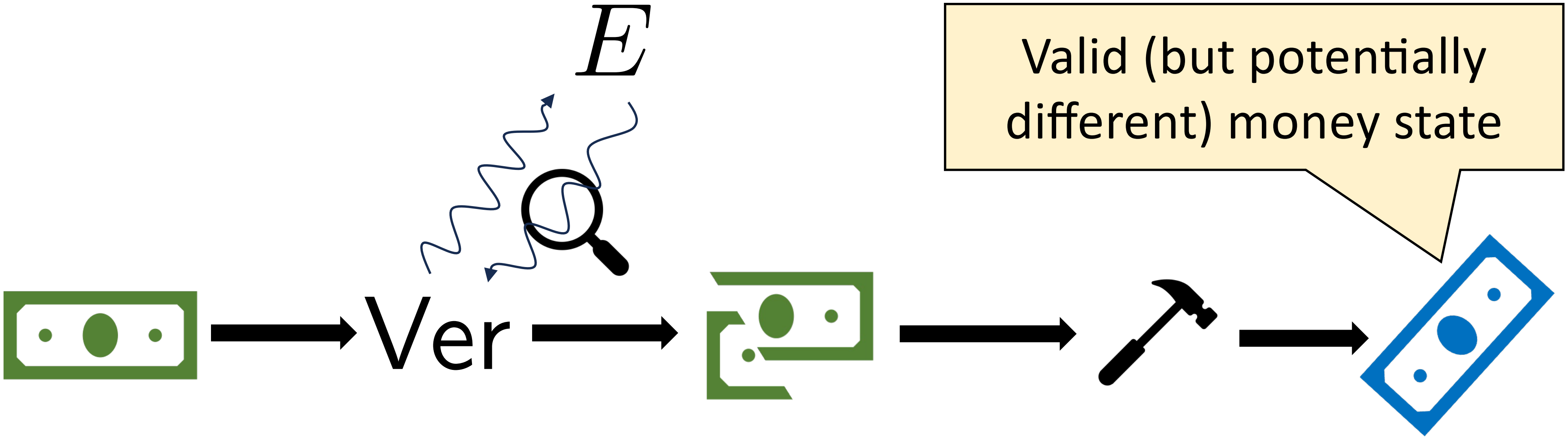
Measurement Principle: measuring a quantum state changes it



Might not be a valid money state, so not guaranteed to get anything useful on subsequent verifications

Step 2: Use Oracle To Break Quantum Money

State Repair Theorem [Chiesa-Ma-Spooner-Z’21]: Under some conditions, can “repair” post-measurement quantum states



Valid (but potentially different) money state

Main open question: separate PK quantum money from OWFs without any restrictions

We need classical mint queries for two reasons:

1. If learn all queries, can clone money
2. Poly-many obfuscated programs → poly-many measurement outcomes
   → employ state repair

[Ananth-Hu-Yuen'23] need classical verifier queries so that they can look at the queries without perturbing the state