Another Round of Breaking and Making Quantum Money: How Not to Do It, and More

Jiahui Liu
University of Texas, Austin

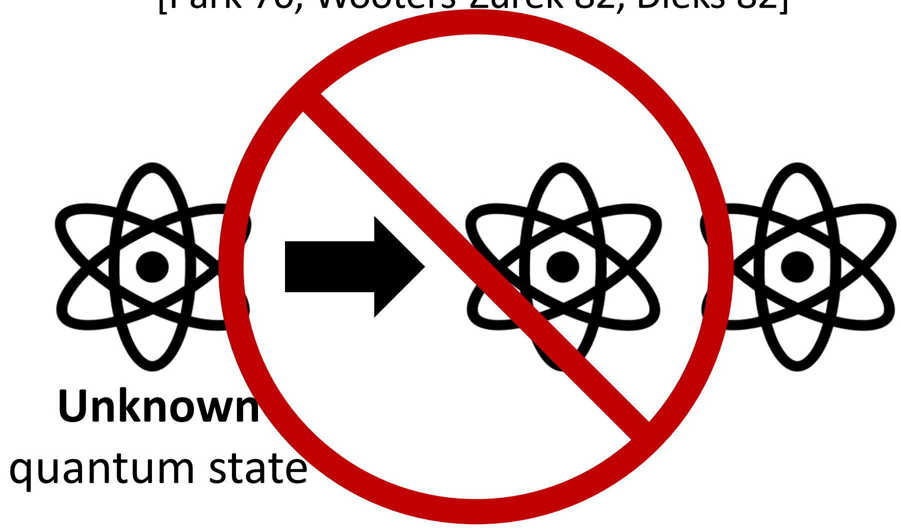
Hart Montgomery
Linux Foundation
(Formerly Fujitsu)

Mark Zhandry
NTT Research
(Formerly Princeton)

Background

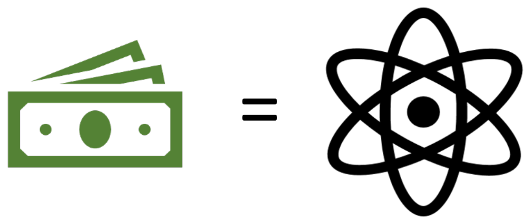
No-cloning Theorem

[Park'70, Wooters-Zurek'82, Dieks'82]



Secret key quantum money

[Wiesner'70]

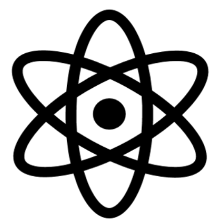


No-cloning ➔ banknotes unforgeable

Problem: only mint can verify

Public key quantum money

[Aaronson'09]

 =  + Public Verification

Challenge: state information-theoretically "known"

➔ breaks no-cloning theorem

➔ need **crypto** + quantum information

(Public Key) Quantum Money is Hard!

[Aaronson'09]: random stabilizer states	✗	[Lutomirski-Aaronson-Farhi-Gosset-Hassidim-Kelner-Shor'10]
[Farhi-Gosset-Hassidim-Lutomirski-Shor'10]: knots	?	little published cryptanalysis effort
[Aaronson-Christiano'12]: polynomials hiding subspaces	✗	[Pena-Faugère-Perret'14, Christiano-Sattath'16]
[Kane'18]: Modular forms	?	[Bilyk-Doliskani-Gong'22] some analysis
[Zhandry'19]: quadradic systems of equations	✗	[Roberts'21]
[Zhandry'19]: post-quantum iO	?	Post-quantum iO not well understood
[Kane-Sharif-Silverberg'21]: Quaternion Algebras	?	No published cryptanalysis effort
[Khesin-Lu-Shor'22]: lattices	?	No (prior) cryptanalysis effort

This Work: Breaking and making quantum money

Attack on general class of lattice-based schemes

↓

[Khesin-Lu-Shor'22] is insecure

"Walkable Invariant" framework + analysis

↓

Identify sufficient conditions for [FGHLS'12] to be secure

(unclear if conditions met)

New candidate walkable invariants

↓

Approach to building quantum money from isogenies

(one crucial missing piece)

How ***Not*** To Build Quantum Money

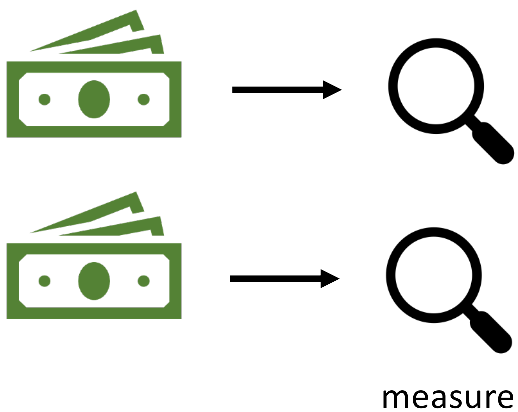
A lattice-based proposal

(folklore)

Verification key (aka serial number) = A , y

$$\propto \sum_{\substack{\text{“short” } x \text{ s.t.} \\ A.x \bmod q = y}} |x\rangle$$

Motivation



"short" x_1 s.t. $A.x_1 = y$

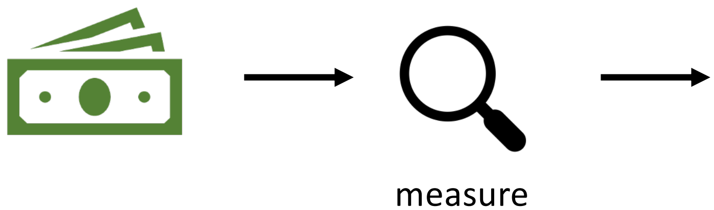
"short" x_2 s.t. $A.x_2 = y$

$x_1 \neq x_2$ whp

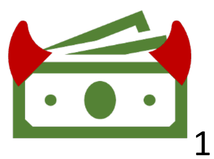
$A.(x_1 - x_2) = 0$ ➡ Short non-vector in kernel of A, aka SIS solution. Believed hard

Attack

(consequence of [Liu-Zhandry'19])



"short" x s.t. $A.x = y$



= $\left| x \right\rangle$

= $\left| x \right\rangle$

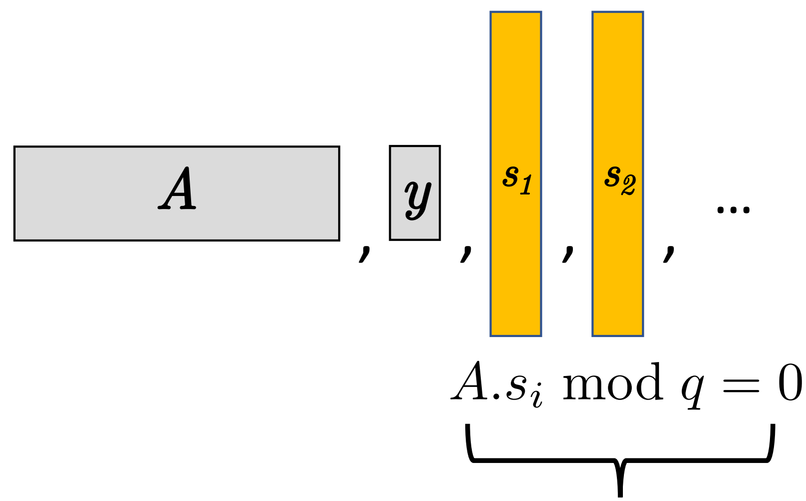
Thm [Liu-Zhandry'19]: LWE + super-poly q → SIS hash function is *collapsing*

Cor: Attack fools *any* efficient verification procedure

(note SIS → LWE [Regev'05])

A more general proposal

Verification key (aka serial number) =



Trapdoors for A, help with verification

Example: can re-interpret [Khesin-Lu-Shor’22] in this form

= “short”

Why Trapdoors are Useful

Assume  $\propto \sum_{x:A.x \bmod q=y} e^{-\pi|x|^2/\sigma^2}|x\rangle$

QFT  $\underset{\text{(approx.)}}{\propto} \sum_{r,e} \left(\omega_q^{r\cdot y}\right) e^{-\pi|e|^2/(q/\sigma)^2}|A^T \cdot r + e\rangle$

$$s^T \cdot (A^T \cdot r + e) = s^T \cdot e = \text{short}$$

Why Trapdoors are Useful

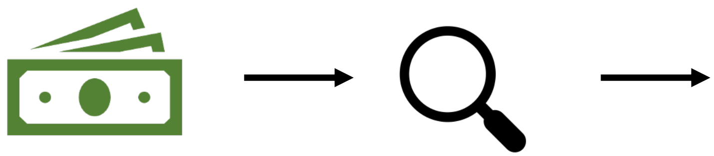
Meanwhile

$$QFT\,|x\rangle \propto \sum_z \left(\omega_q^{z\cdot x}\right)|z\rangle$$

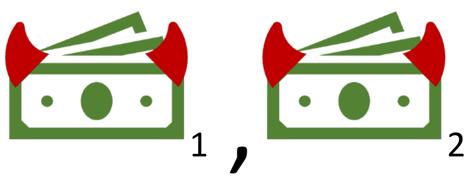
$$s^T \cdot z = \text{big (whp)}$$

Detects attack

Attack
(this work)



“short” x s.t. $A.x = y$



$$= \sum_{u_1, u_2, \ldots \text{ s.t. z is "short"}} |z = x + u_1 s_1 + u_2 s_2 + \ldots\rangle$$

Thm (this work):

1. LWE + ***any*** q → fools any efficient verification in many natural settings
3. Efficiently construct fake money state from x in many natural settings

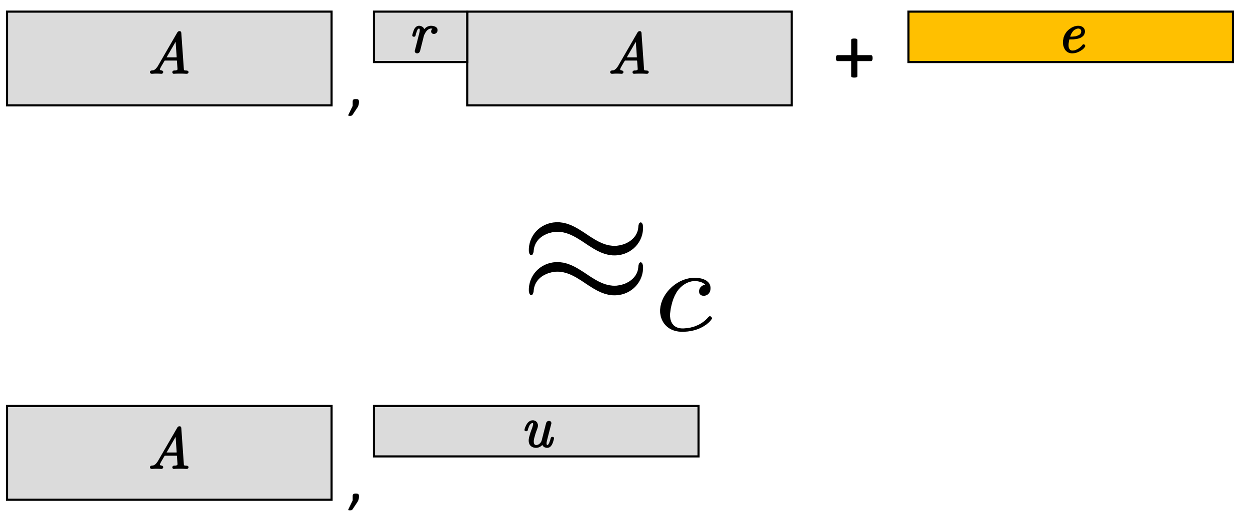
Cor: Scheme from [Khesin-Lu-Shor’22] is insecure

Along the way, improve known results about k-LWE problem

Proof Idea

Learning With Errors (LWE)

[Regev'05]

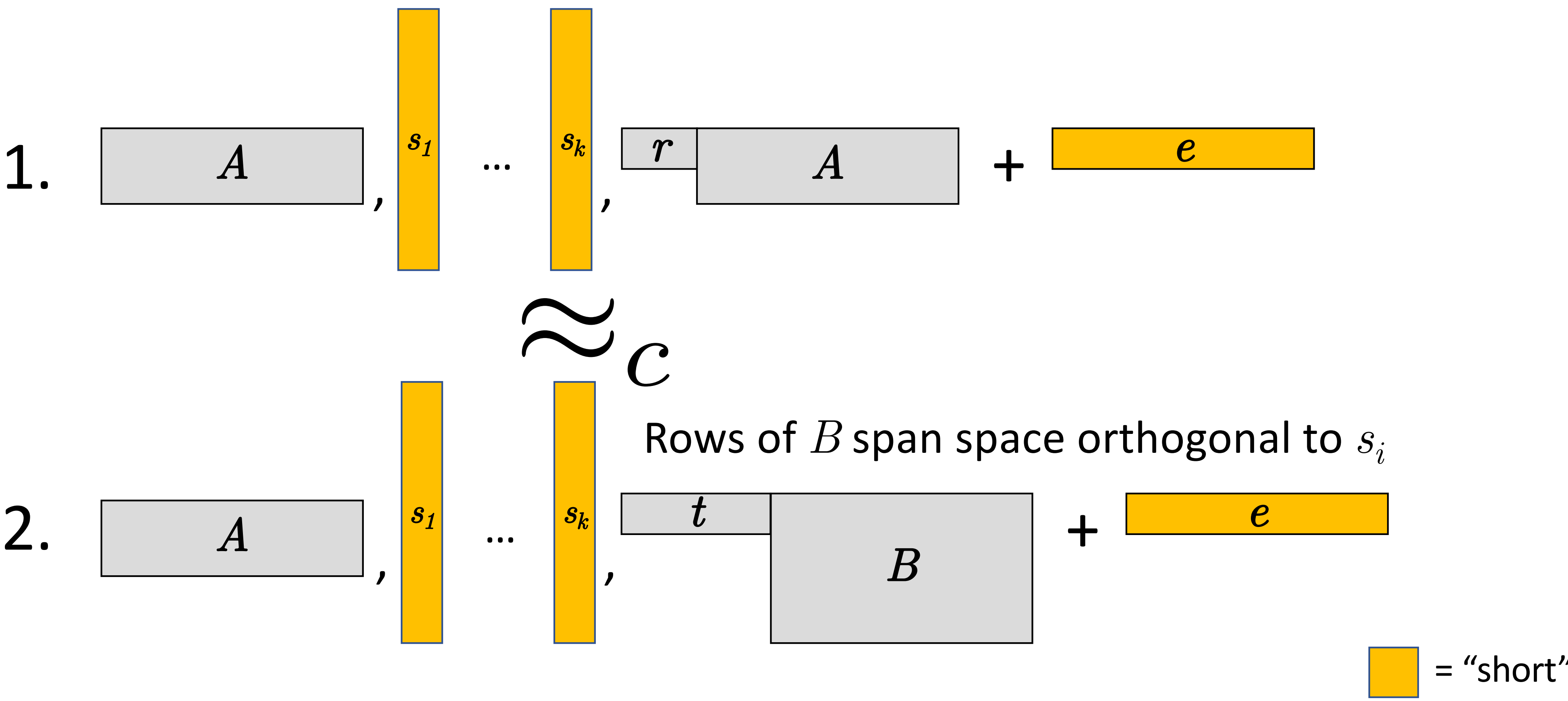


(everything defined mod q)


= "short"

k-LWE

[Ling-Phan-Stehlé-Steinfeld'14]



Thm [Ling-Phan-Stehlé-Steinfeld'14]:

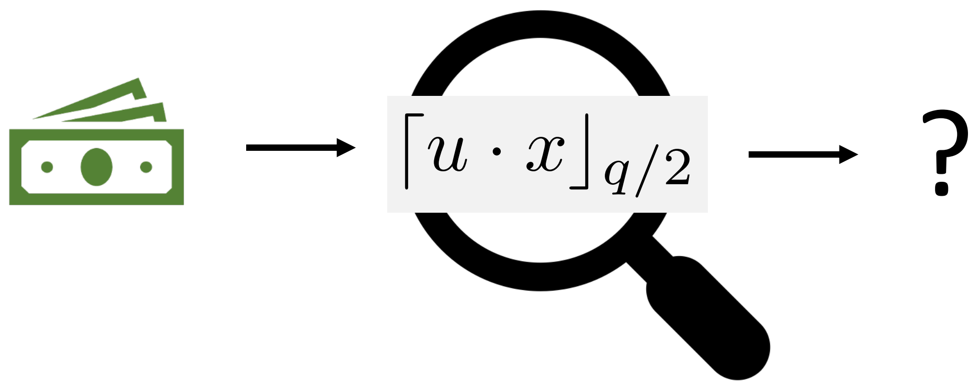
LWE ➔ k-LWE for polynomial k, if s_i are Gaussian

Thm (this work):

LWE ➔ k-LWE for constant k, for arbitrary short s_i

Proof Idea

Sample u as in either case 1. or 2. as in k-LWE



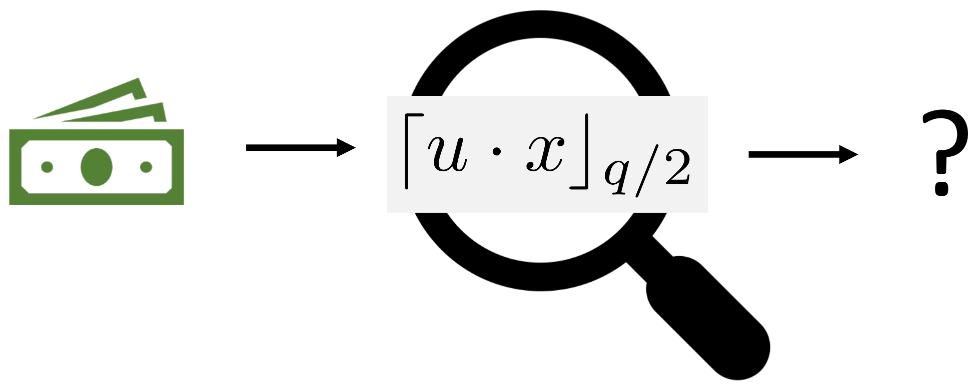
Case 1: $u \cdot x = (r \cdot A + e) \cdot x = r \cdot y + e \cdot x \approx r \cdot y$

➔ minimal collapse of 

$\lceil \cdot \rfloor_{q/2}$ = Round to 0 or $q/2$

Proof Idea

Sample u as in either case 1. or 2. as in k-LWE



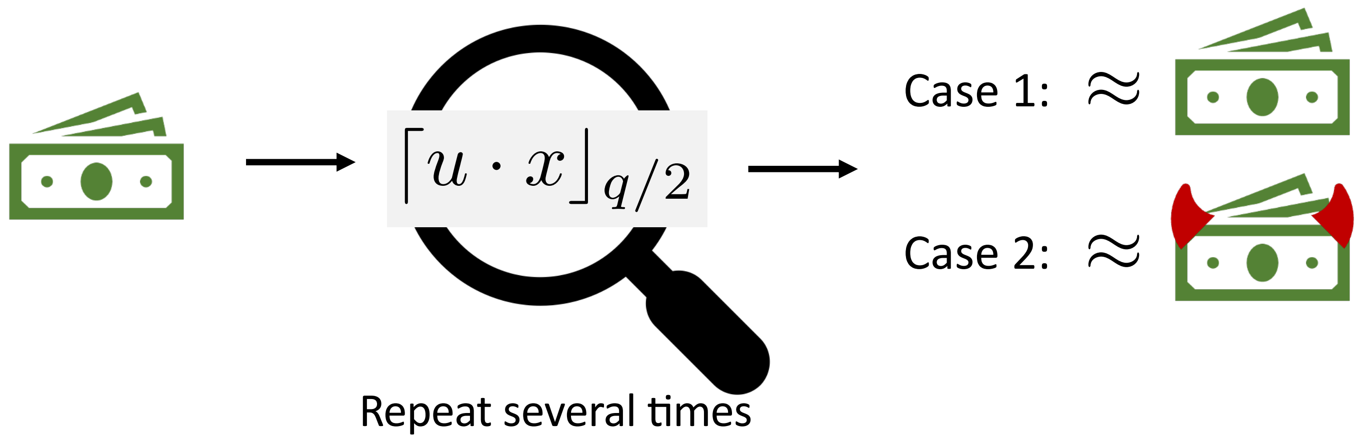
Case 2: $u \cdot x = (t \cdot B + e) \cdot x \approx t \cdot B \cdot x$

➔ collapse "toward" 

$\lceil \cdot \rfloor_{q/2}$ = Round to 0 or $q/2$

Proof Idea

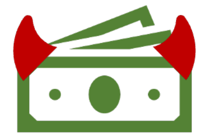
Sample u as in either case 1. or 2. as in k-LWE



Problem: error scales as $1/q$ → non-negligible for poly q

This work: More fine-grained analysis → handle poly q

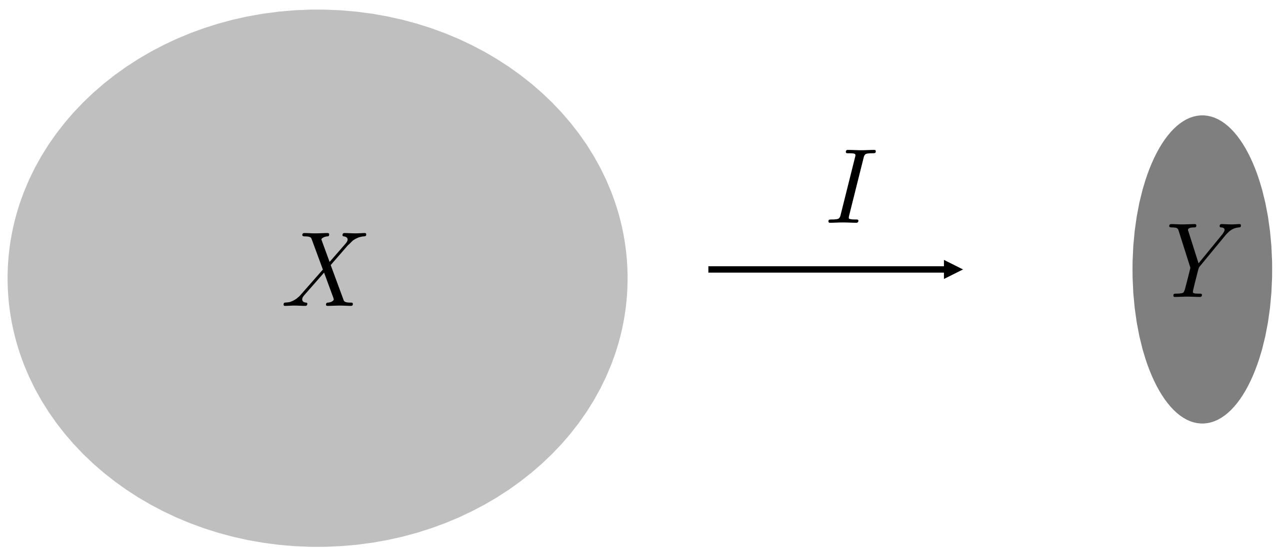
Proof Idea

Final missing piece: constructing  from x

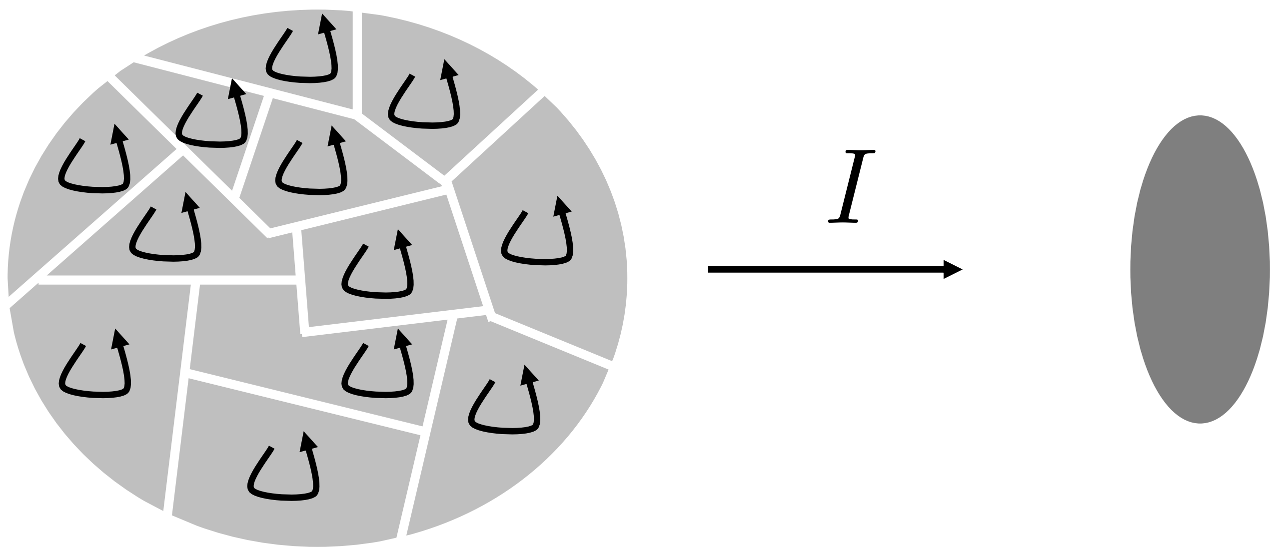
Solution: use classical techniques for sampling short vectors in lattices, but "in superposition"

Walkable Invariant Framework

(abstraction of [FGHLS'12])

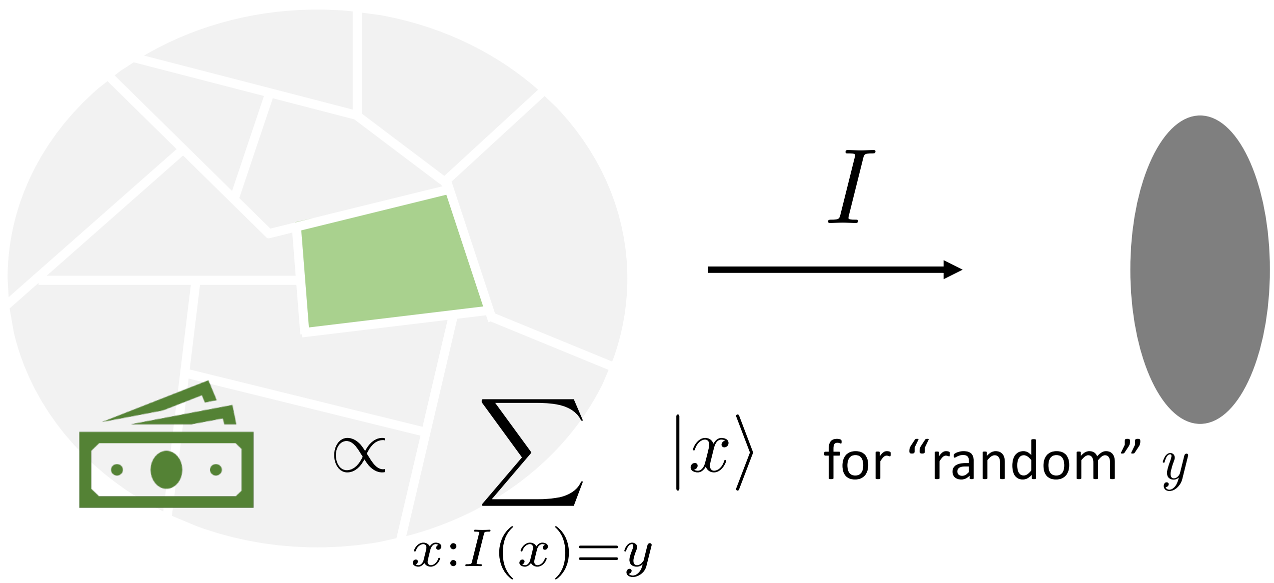


Permutations $\sigma_i : X \to X$



$$I(\ \sigma_i(x)\ ) = I(x)$$

Assume for purposes of talk that it is possible to go between any two elements in the same part via a sequence of σ_i. In the paper we handle the case where the parts are disconnected.



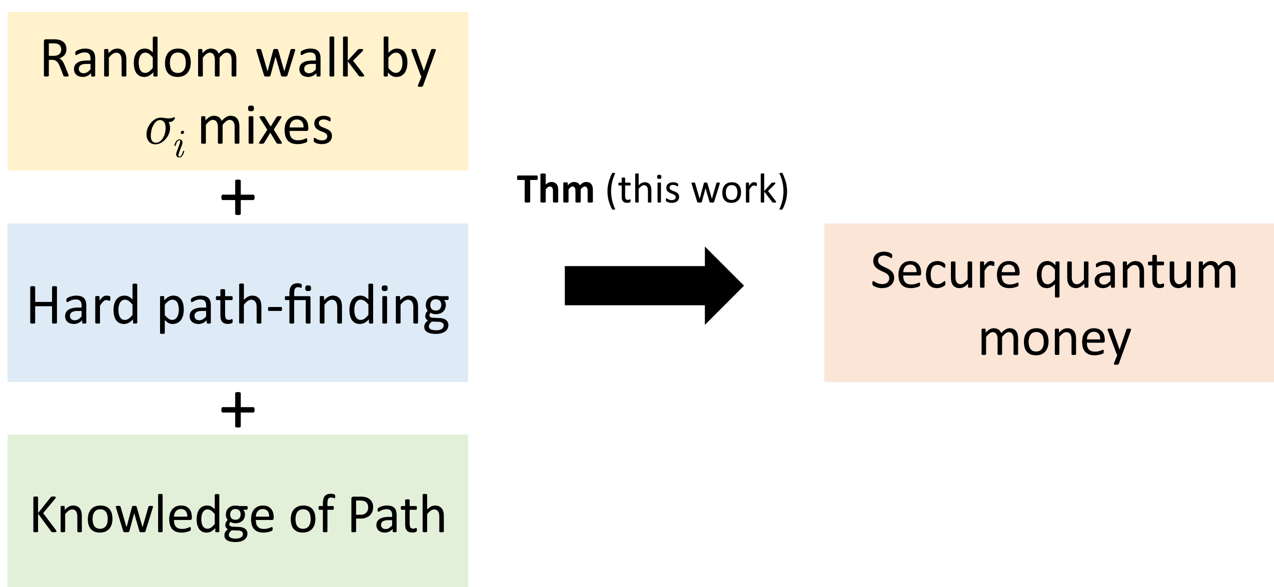
1. Creates uniform superposition over X
2. Measure $I(x)$

Verification:

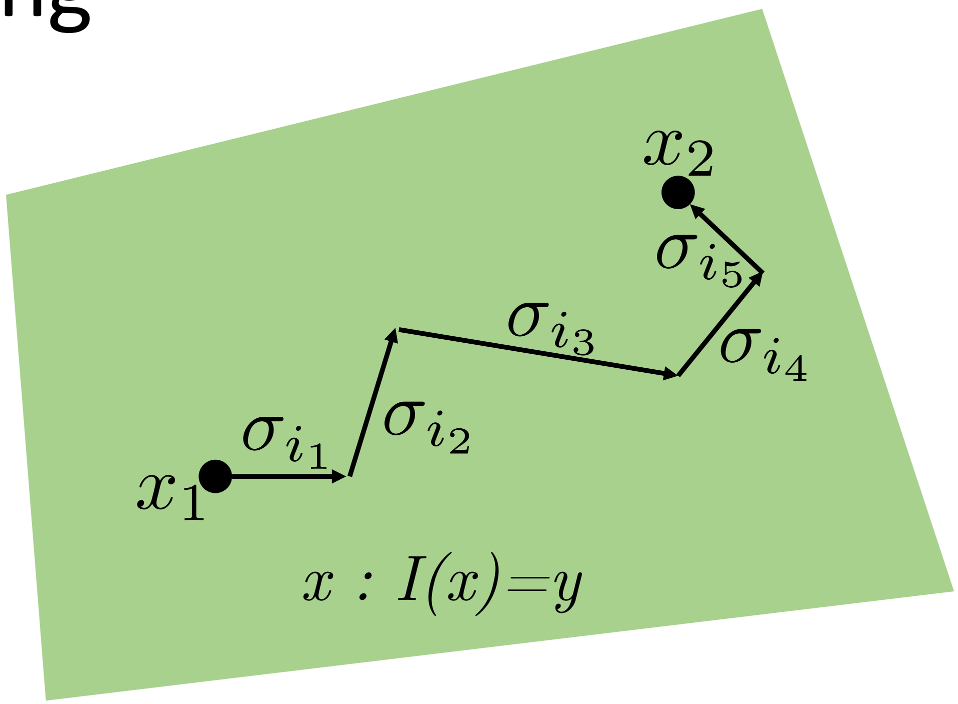
1. Test that support is on x s.t. $I(x)=y$
2. Test that state is unchanged under action by σ_i

Use version of swap test

Recipe for Quantum Money from Invariants

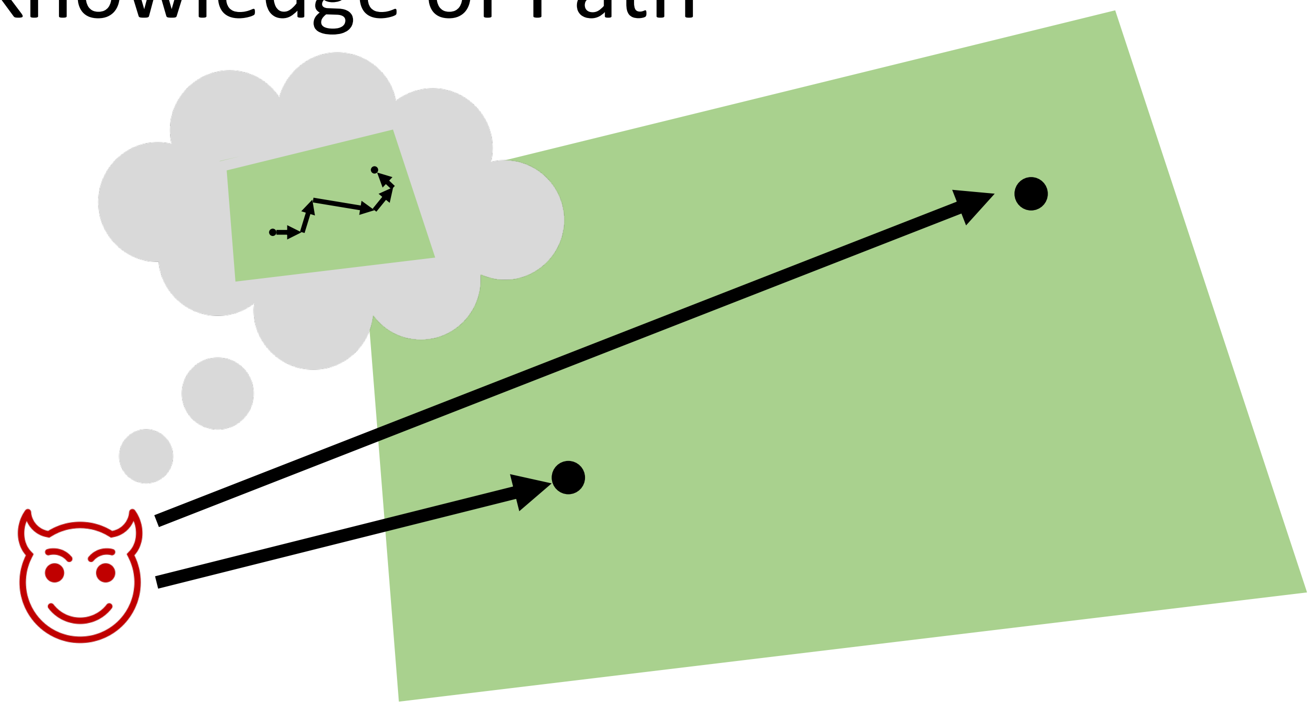


Path-finding



Given random x_1,x_2 with same invariant, compute a "path" = i_i, i_2, …

Knowledge of Path



Impossible to generate x_1, x_2 with same invariant without knowing path

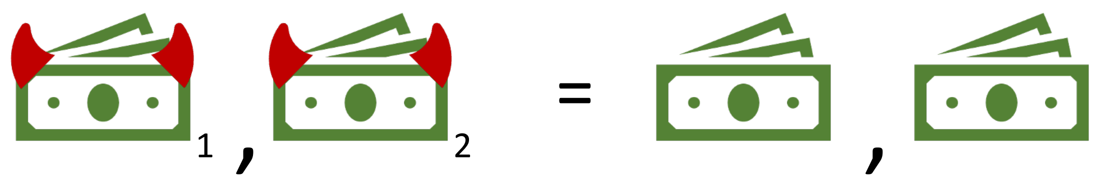
Proof Idea

Proof Idea

Assume toward contradiction:

with same $I(x)$

Mixing →



Proof Idea

Assume toward contradiction:

 with same $I(x)$

Measure each , get uniform independent x,y s.t. $I(x)=I(y)$

Knowledge of path → can construct path between x and y

→ contradicts hardness of path-finding

[FGHLS'12]

X = knot diagrams
$I(x)$ = Alexander polynomial
σ_i = Reidemeister moves

Security previously merely conjectured, with minimal analysis

Hardness of path-finding and knowledge of path
seem plausible, mixing unclear but possible

New Instantiations

Isogenies over (supersingular) elliptic curves

Path finding = computing isogenies, widely believe to be hard

Knowledge of Path = analog of knowledge of exponent from groups

Seems quite plausible, but need more cryptanalysis effort

Problem: unknown how to create uniform superposition over X for minting

Closely related to major open question of obliviously sampling super-singular elliptic curves

Other instantiations

Re-randomizeable Functional Encryption

Group actions + classical oracle

Thanks!