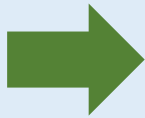


Revisiting Post-Quantum Fiat-Shamir

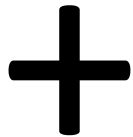
Qipeng Liu & **Mark Zhandry**
(Princeton & NTT Research)

Lattice Crypto \neq Post-Quantum Crypto

Typical Lattice Crypto Thm:

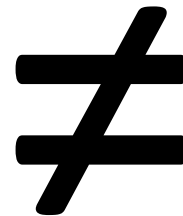


Alg for lattice problems



Assumption:

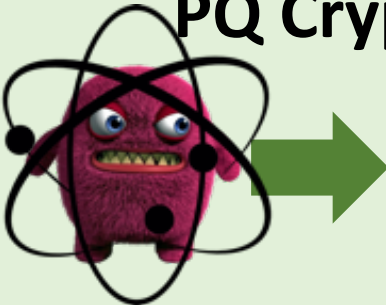
Lattice problems are quantum hard



Security Goal:



Post-Quantum Crypto

PQ Crypto Thm:
 Q alg for lattice problems

+

Assumption:
Lattice problems
are quantum hard

=

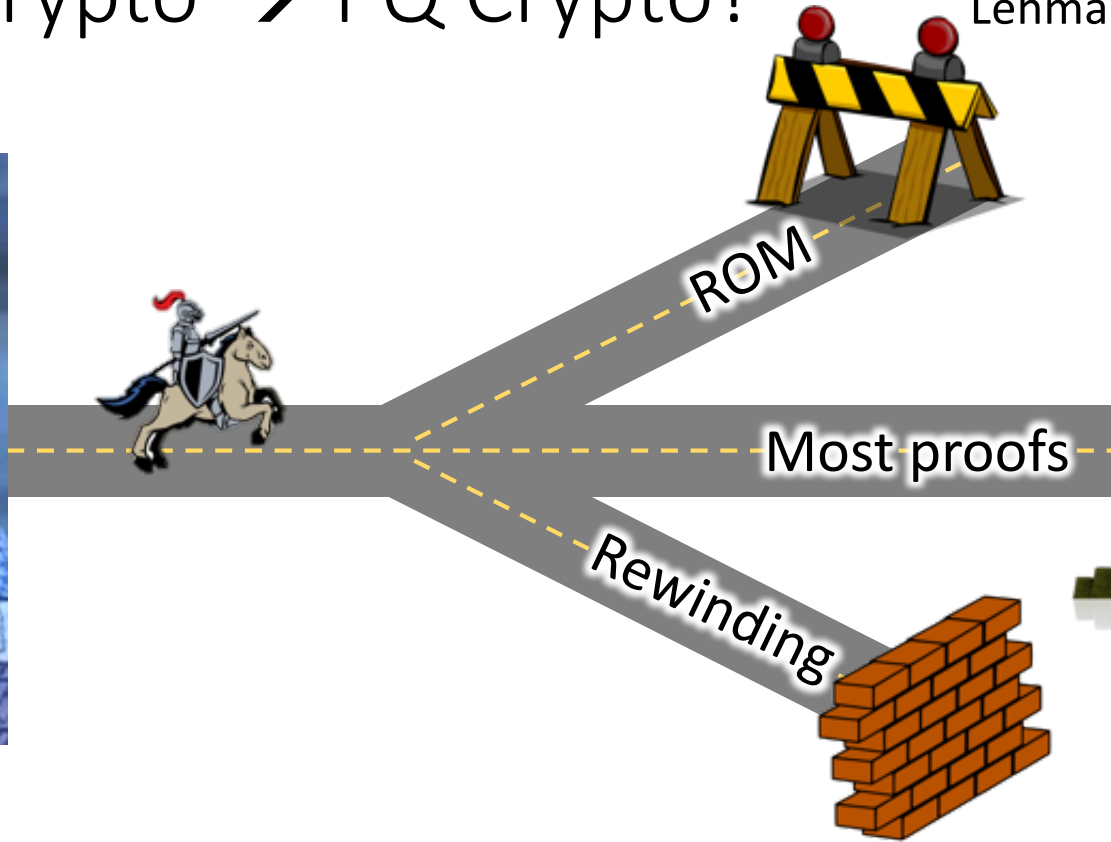
Security Goal:


Lattice Crypto \rightarrow PQ Crypto?

[Boneh-Dagdelen-Fischlin-Lehmann-Schaffner-Z'11]



Classical reduction



[van de Graaf'97, Ambainis-Rosmanis-Unruh'14]



Quantum reduction

PQ Signatures from Lattices?

Standard Model

[Cash-Hofheinz-Kiltz-Peikert'09,...]

Hash-and-sign

[Gentry et al.]

Vaikuntanathan

ROM

[BDFLSZ'11,...]



One-way Funcs

[Rompe'00] + [Ajtai'96]

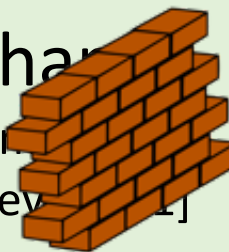
Partial Solutions

[Kiltz-Lyubashevsky-Schaffner'17,
Unruh'14,17,...]

FiFi-Shar

[FiFi-Shar]

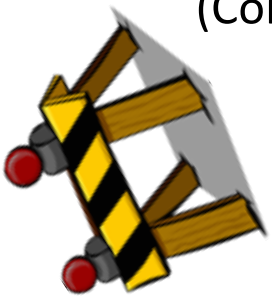
[Lyubashevsky'14]



This Work

Thm: Fiat-Shamir is
PQ secure in the ROM

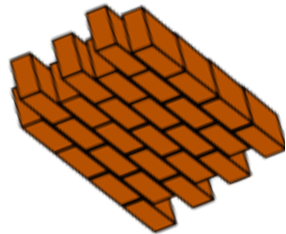
(Concurrently with [Don-Fehr-
Majenz-Schaffner'19])



New techniques for
quantum rewinding

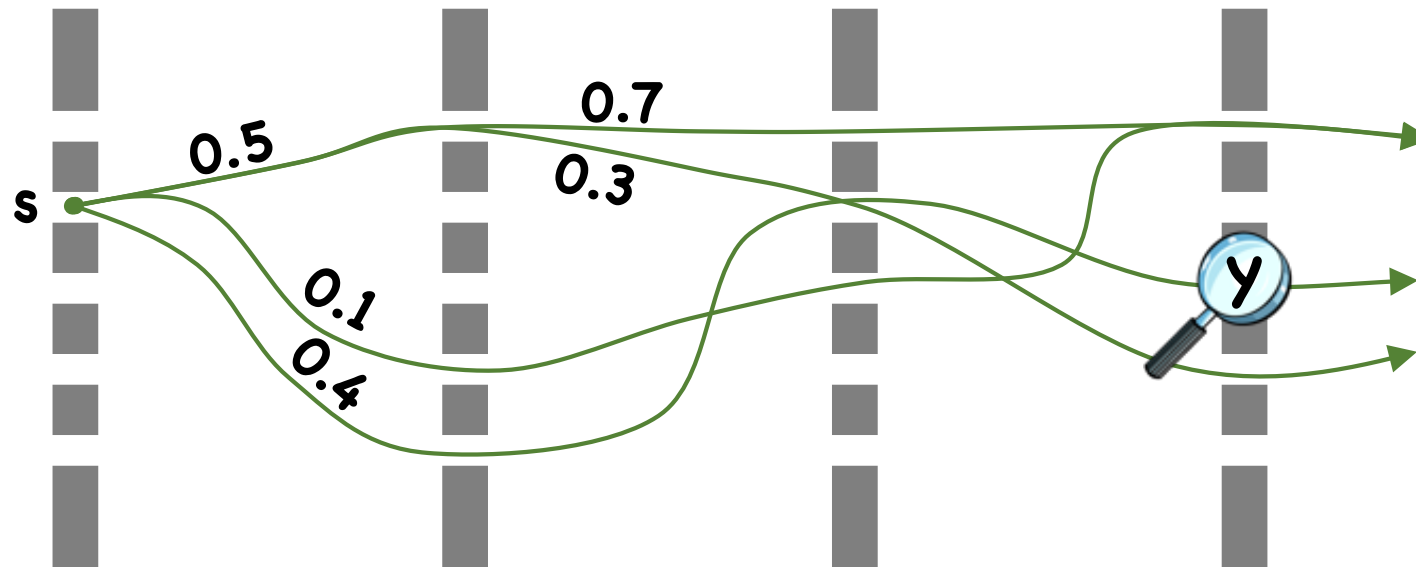


Cor: [Lyubashevsky'11] is
PQ secure assuming LWE



Quantum Background

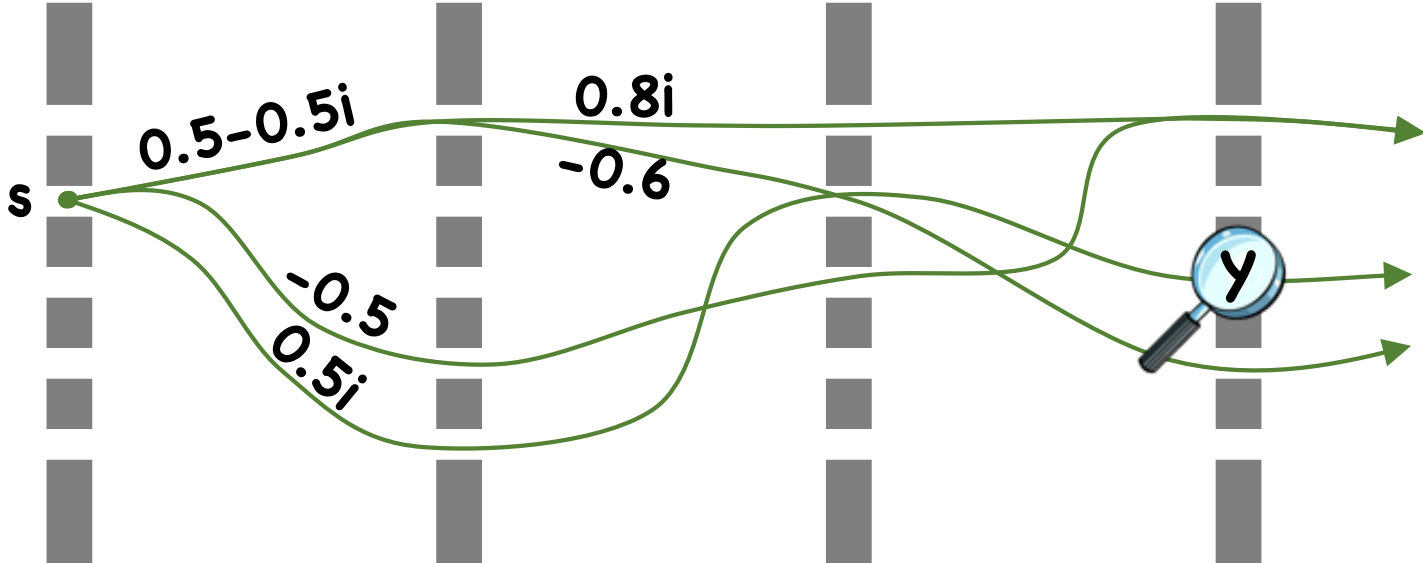
Classical Stochastic Process



$$W(\text{path } \mathbf{p}) := \prod(\text{probabilities along path}) = \Pr[\mathbf{p}]$$

$$\Pr[y] = \sum_{\mathbf{p}: s \rightarrow y} W(\mathbf{p})$$

Quantum Process

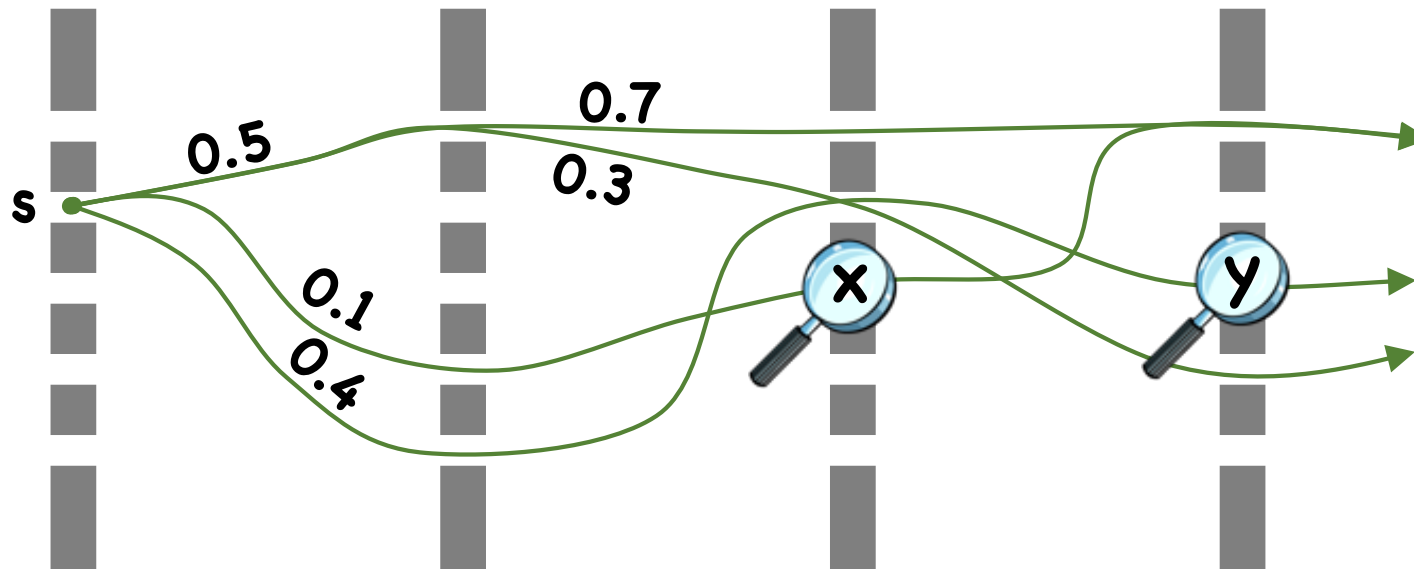


$$W(\text{path } p) := \prod(\text{weights along path})$$

$$\Pr[y] = \left| \sum_{p:s \rightarrow y} W(p) \right|^2$$

Main Diff between Quantum and Classical:
Paths can interfere constructively or destructively,
amplifying probabilities or eliminating them

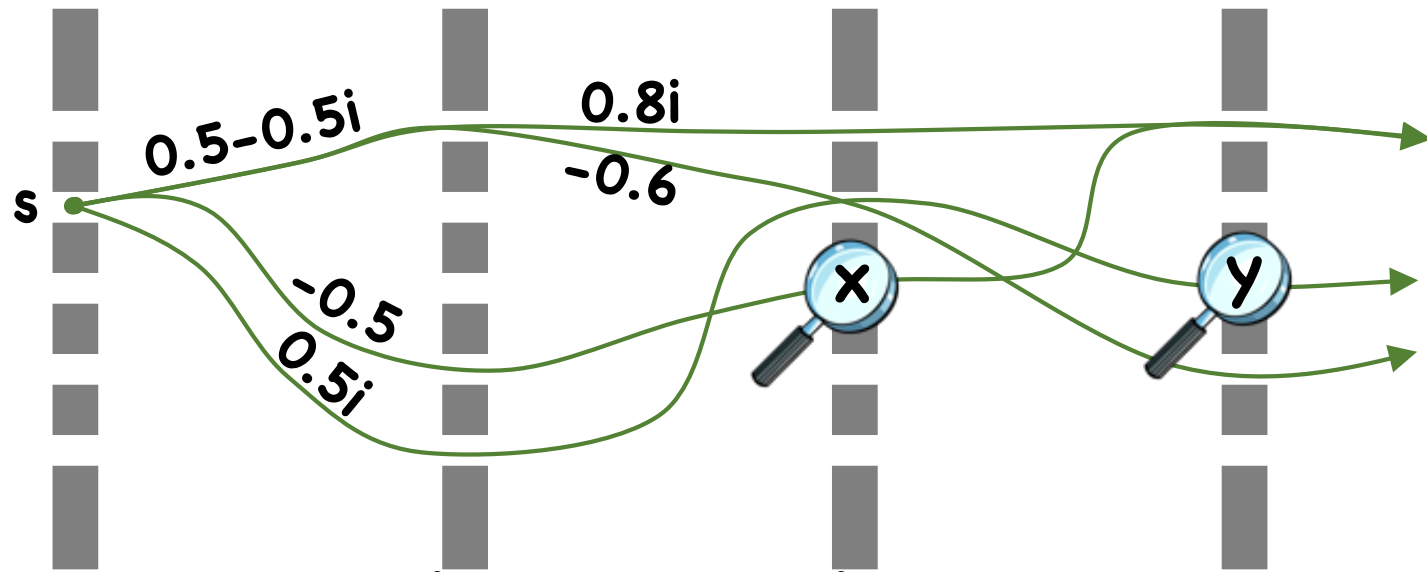
Intermediate Observation in Stochastic Process



$$\Pr[x \wedge y] = \sum_{p:s \rightarrow x \rightarrow y} W(p)$$

$$\sum_x \Pr[x \wedge y] = \sum_{x,p:s \rightarrow x \rightarrow y} W(p) = \sum_{p:s \rightarrow y} W(p) = \Pr[y]$$

Intermediate Observation in Quantum Process



$$\Pr[x \wedge y] = \left| \sum_{p:s \rightarrow x \rightarrow y} w(p) \right|^2$$

$$\sum_x \Pr[x \wedge y] = \sum_x \left| \sum_{p:s \rightarrow x \rightarrow y} w(p) \right|^2 \neq \Pr[y]$$



Paths for different x can
no longer interfere



Observer effect: Learning anything
about quantum system disturbs it

QM is Reversible?

Quantum Reversibility?

Transition matrices
preserve 2-norm \rightarrow Unitary \rightarrow Invertible

but...



Quantum Irreversibility:



\rightarrow Irreversibly alters state

Is CM Reversible?


Classical Irreversibility?


Transition matrices preserve 1-norm  Stochastic  May be singular

but...

Classical Reversibility:

Can always observe state
at any point in time

 Doesn't affect
output distribution

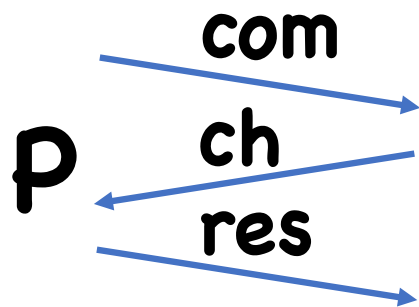
 Can "rewind" and
return to prior state

Part 1:
Fiat-Shamir In the Quantum
Random Oracle Model

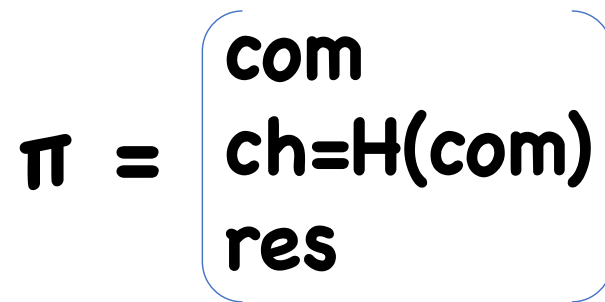
The Fiat-Shamir Transform [Fiat-Shamir'87]

(public coin, HV)

3-Round Proof (of Knowledge)



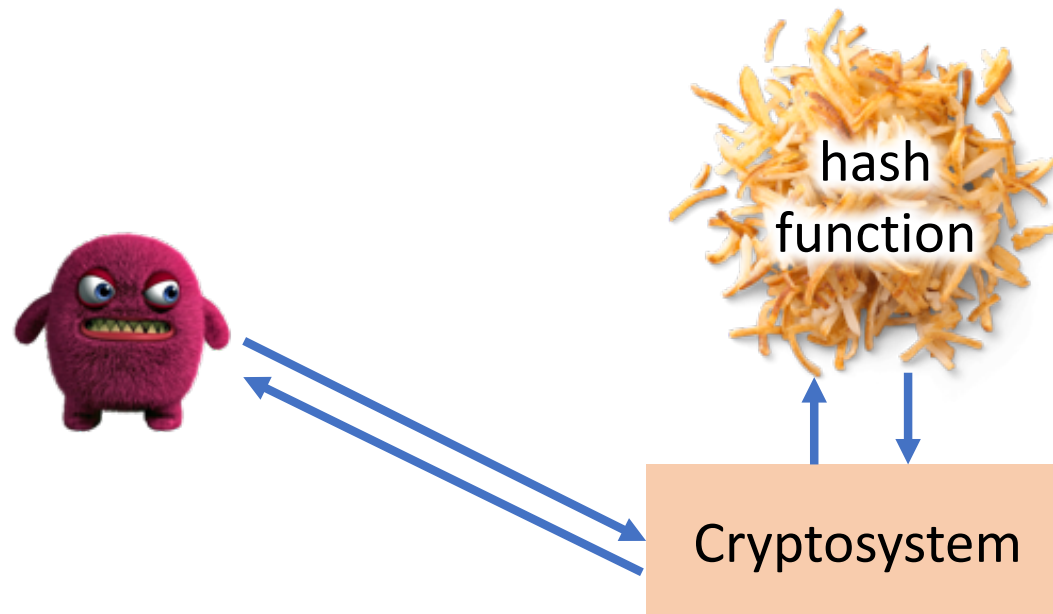
V



NI Proof (of Knowledge)

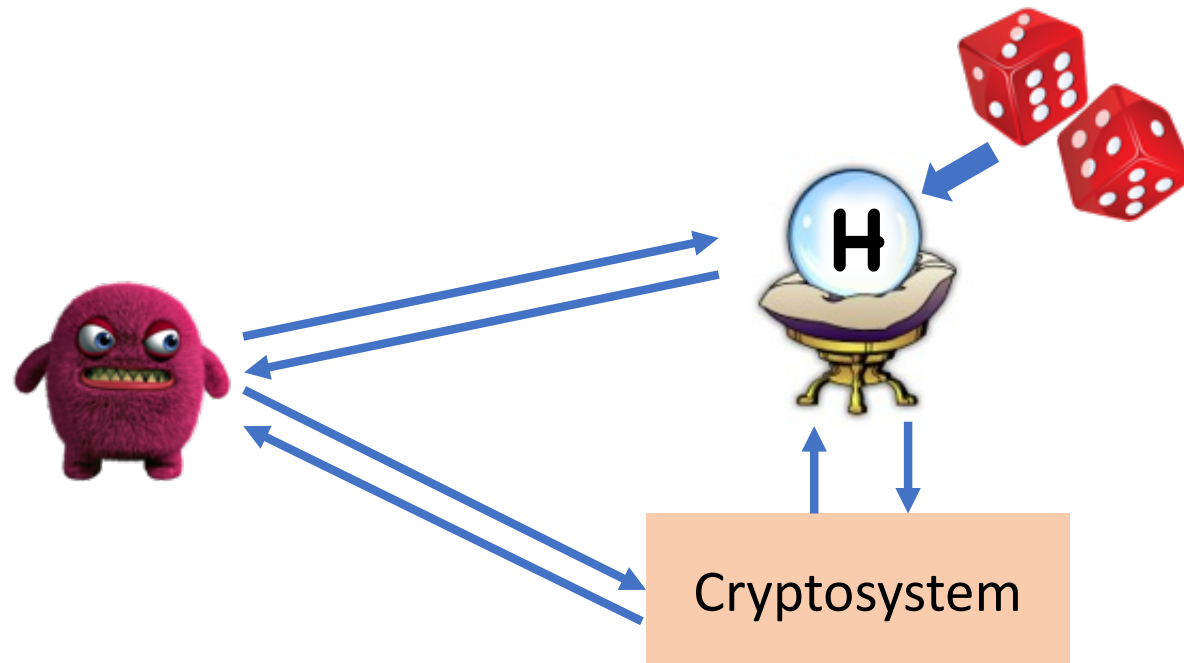
Also: Identification protocols \rightarrow signatures

PQ Fiat-Shamir Problem 1: ROM



For many schemes (including FS), can't base security on concrete hash function property

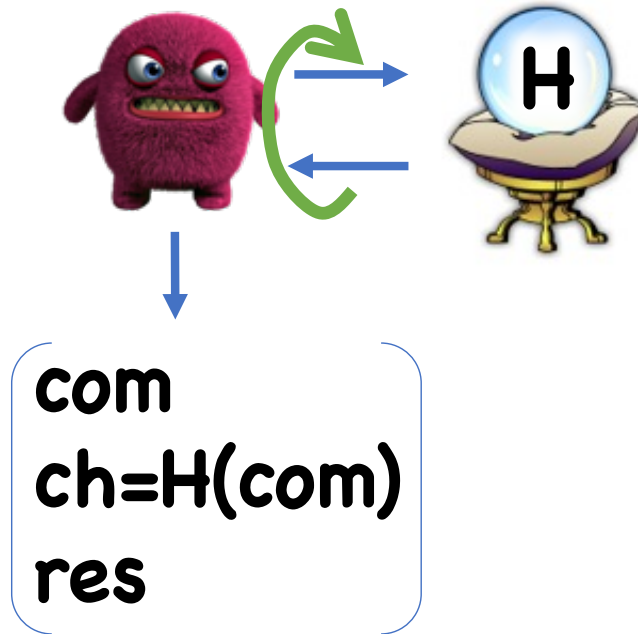
PQ Fiat-Shamir Problem 1: ROM



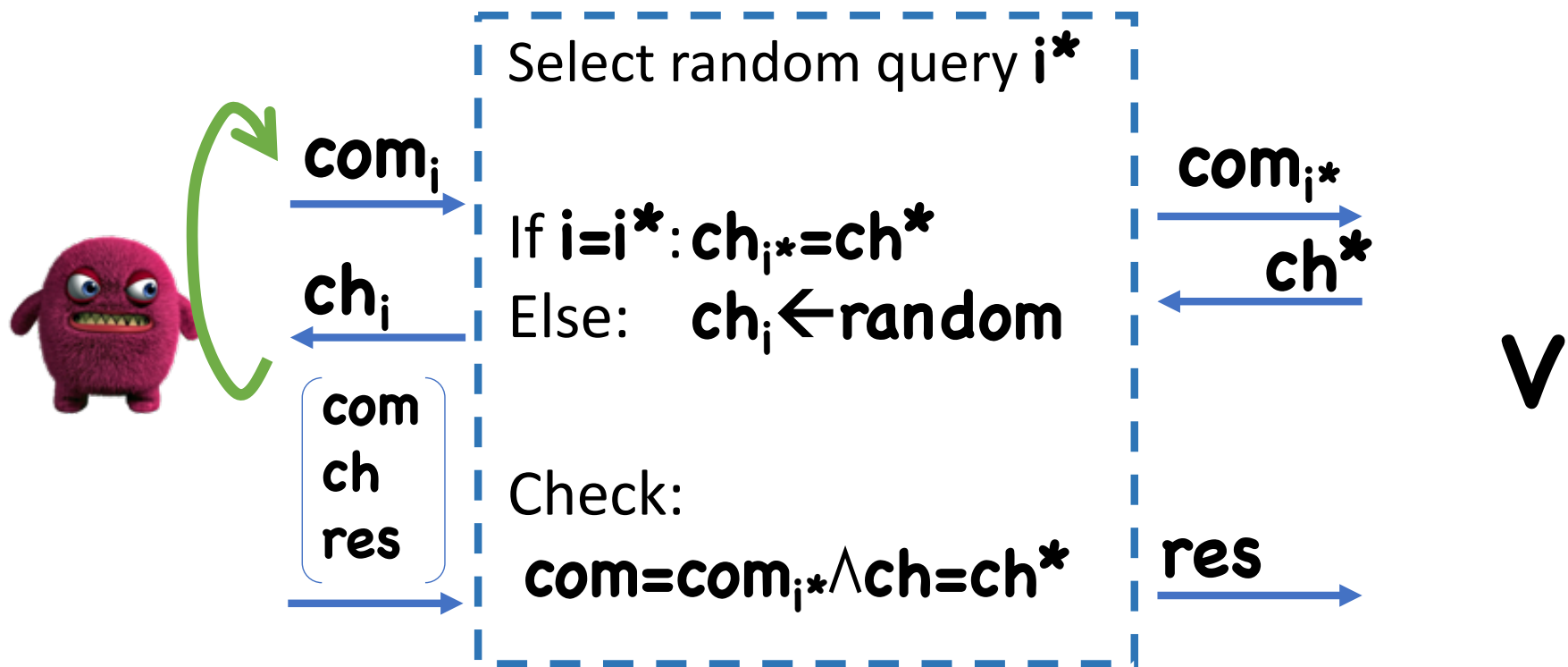
Solution ([Bellare-Rogaway'93]):
Model hash as random oracle

Classical Fiat-Shamir Proof

Assume:



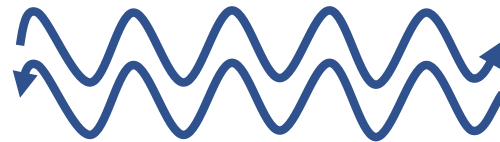
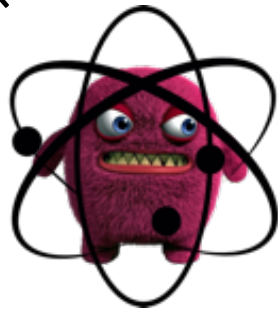
Classical Fiat-Shamir Proof



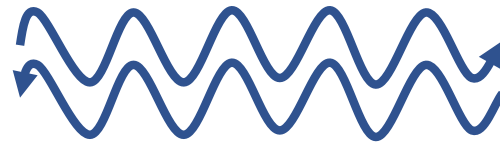
The Quantum Random Oracle Model (QROM)

[Boneh-Dagdelen-Fischlin-Lehmann-Schaffner-Z'11]

Real World

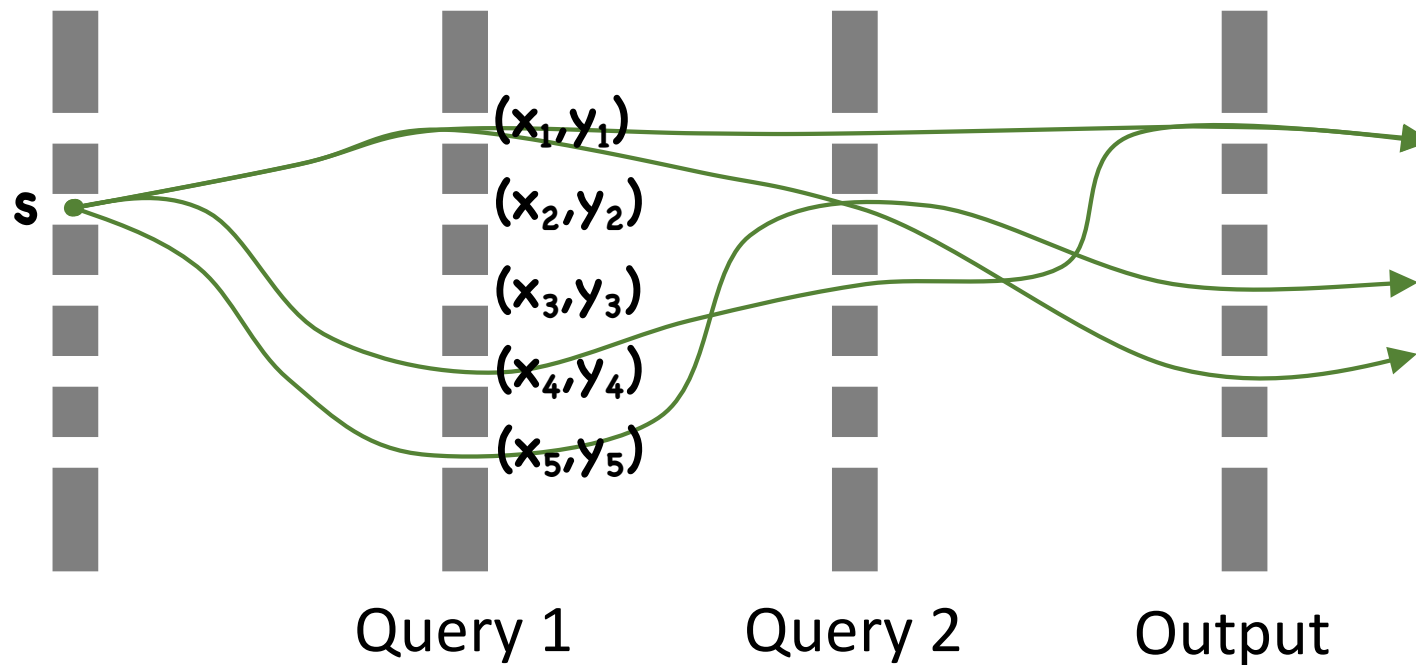


ROM



Now standard in post-quantum crypto

A Path View of Quantum Query Algs



Query: $(x, y) \rightarrow (x, y \oplus H(x))$

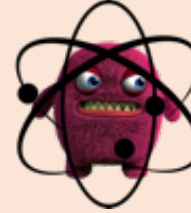
Problems with Fiat-Shamir in QRROM

Query extraction:



disturbed by
extracting com_{i^*}

On-the-fly simulation:



can “see” all of
 \mathbf{H} on first query

Adaptive Programming:

Can only set $\mathbf{H}(\text{com}_{i^*})$ *after*
queries already made



Typical solution:

Commit to entire
 \mathbf{H} at beginning

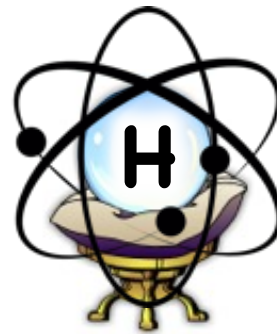
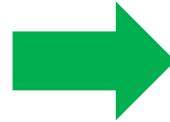


Main Theorem: Fiat-Shamir preserves knowledge soundness in the quantum random oracle model. Also signatures from ID protocols.

Tool: [Z'19b]



Equal prob.
on all oracles



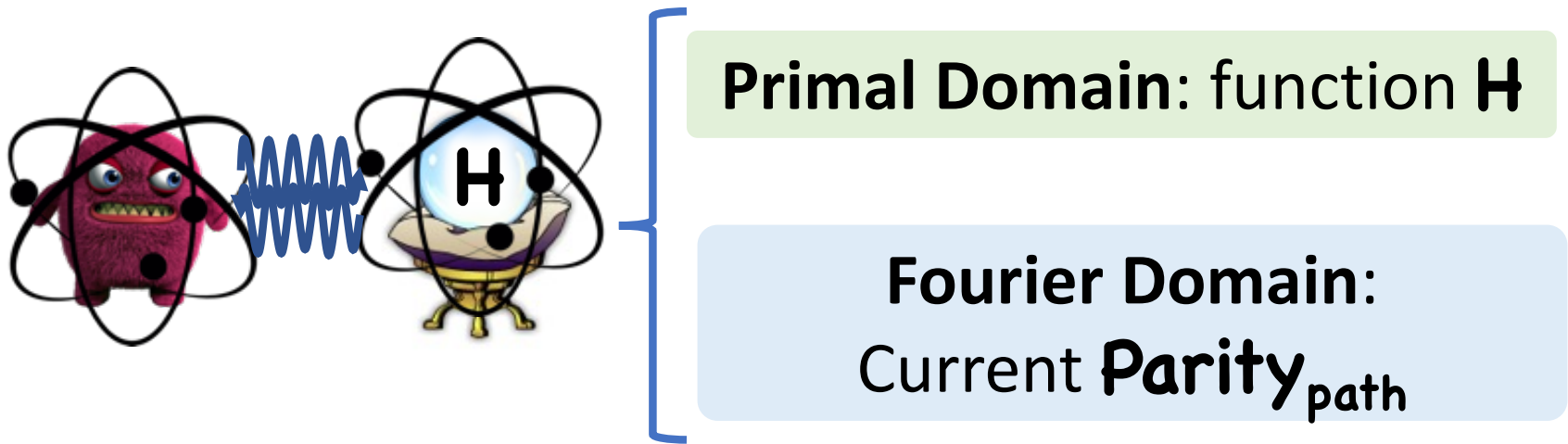
Equal weight
on all oracles

Paths for difference
H can't interfere



Quantum-ifying H has no
effect on output distribution

A Path View of [Z'19b]



$$\mathbf{Parity}_{\text{path}}(\mathbf{x}) := \bigoplus_{(x,y) \in \text{path}} \mathbf{y}$$

How to Extract from Quantum Queries

Lemma (informal): If $\text{Parity}_{\text{path}}(\mathbf{x}) = 0^n$,
path has no knowledge of $\mathbf{H}(\mathbf{x})$

Corollary: Any successful path must
have $\text{Parity}_{\text{path}}(\mathbf{com}) \neq 0^n$ at the end

(In particular must have queried **com**)

A Useful Tool

Observation Lemma ([Boneh-Z'13]): If observing \mathbf{x} gives t possible outcomes,

$$\Pr[y \mid \mathbf{x} \text{ observed}] \geq \Pr[y]/t$$

(simple consequence of Cauchy-Schwartz/Jensen)

Note: Doesn't work in other direction

Generalization

Lemma: Let $\mathcal{P} = \{\mathcal{P}_i\}_{i \in [t]}$ be a partition of possible paths.

$$\Pr[y \mid i \text{ observed}] \geq \Pr[y]/t$$

Our (First) Partition

$\mathcal{P}_i = \{\text{successful paths where}$
• $\text{Parity}_{\text{path}}(\text{com}) = 0^n$ just before query i
• $\text{Parity}_{\text{path}}(\text{com}) \neq 0^n$ after all queries $j \geq i\}$

Algorithm to sample \mathcal{P}_i (assuming i known)



- When making i -th query,
 - Observe **com**
 - Observe if $\text{Parity}_{\text{path}}(\text{com}) = 0^n$. If **not**, abort
- For j -th query, $j > i$, observe if $\text{Parity}_{\text{path}}(\text{com}) = 0^n$. If **so**, abort
- At end, if **adv** doesn't output **com**, abort

Must guess i  Loose extra factor of q

How to Adaptively Program

Adaptive Programming:

We now know **com**, but how do we embed **ch** into **H**?

Idea: Just before query i ,
 $\text{Parity}_{\text{path}}(\text{com}) = 0^n$  Adv knows nothing
Can replace  about **H(com)**
contents with **ch**

Problem: No more access to $\text{Parity}_{\text{path}}(\text{com})$

An Alternative Partition?

$P_i = \{\text{successful paths where}$
• $\text{Parity}_{\text{path}}(\text{com}) = 0^n$ after all queries $j < i$
• $\text{Parity}_{\text{path}}(\text{com}) \neq 0^n$ after query $i\}$

Problem:

Need to know
com at beginning

but

com isn't observed
until query i

How to Adaptively Program

Takeaway: Need partition that doesn't check **Parity_{path}(com)** once programmed

Takeaway: Need partition that doesn't check **Parity_{path}(com)** before **com** observed

Yet Another “Partition”?

$Q_i = \{$ successful paths where

• $\text{Parity}_{\text{path}}(\text{com}) = 0^n$ just before query i

• $\text{Parity}_{\text{path}}(\text{com}) \neq 0^n$ just after query $i\}$

Problem: some paths counted multiple times

$k = \left(\begin{array}{l} \text{number of times } \text{Parity}_{\text{path}}(\text{com}) \\ \text{switches from } 0^n \text{ to } \neq 0^n \end{array} \right)$

path will then be
in k of the Q_i

Yet Another “Partition”?

$Q_i = \{\text{successful paths where}$

- $\text{Parity}_{\text{path}}(\text{com}) = 0^n$ just before query i

- $\text{Parity}_{\text{path}}(\text{com}) \neq 0^n$ just after query $i\}$

R_i counts =

Q_i over-counts

$R_i = \{\text{successful paths where}$

- $\text{Parity}_{\text{path}}(\text{com}) \neq 0^n$ just before query i

- $\text{Parity}_{\text{path}}(\text{com}) = 0^n$ just after query $i\}$

Generalization of [Boneh-Z'13]


Thm: Let $\mathcal{P} = \{\mathcal{P}_i\}_{i \in [t]}$ be a *collection* of sets of paths. Suppose $\exists \{\alpha_i\}$ s.t. for all \mathbf{p} , $\sum_{i: \mathbf{p} \in \mathcal{P}_i} \alpha_i = 1$.

$$\Pr[y \mid \mathcal{P}_i, i \text{ uniformly random}] \geq \Pr[y] / \text{poly}(t)$$

Relation to [Don-Fehr-Majenz-Schaffner'19]

[Liu-Z'19]:

We actually use much larger set $\{R_i\}$

 worse reduction

[Don-Fehr-Majenz-Schaffner'19]:

Direct algorithm+analysis, essentially
same algorithm using the presented $\{R_i\}$

Takeaway

Most major ROM techniques/results
now ported to QRROM

Perhaps explains why known
counterexamples are so contrived

[Boneh-Dagdelen-Fischlin-
Lehmann-Schaffner-Z'11]:
Relies on timing

[Zhang-Yu-Feng-Fan-Zhang'19]:
Doesn't correspond to natural
crypto task

Part 2:
New Techniques for
Quantum Rewinding

PQ Fiat-Shamir Problem 2: Rewinding

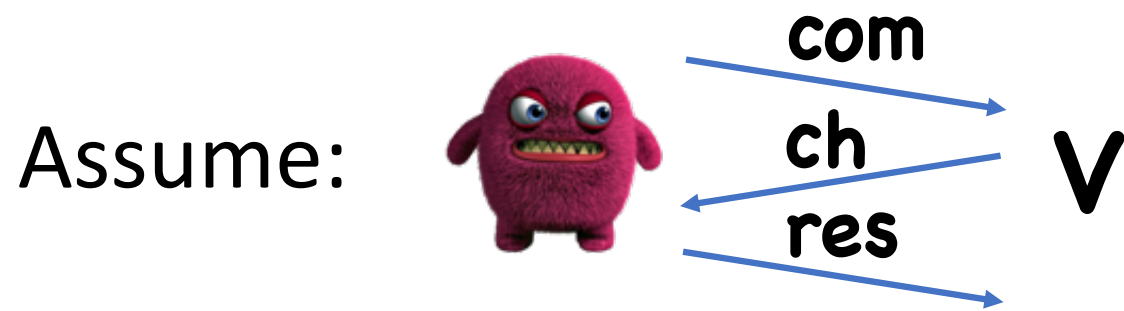
Special Soundness: Can extract witness from $(\mathbf{com}_0, \mathbf{ch}_0, \mathbf{res}_0)$, $(\mathbf{com}_1, \mathbf{ch}_1, \mathbf{res}_1)$ s.t. $\mathbf{com}_0 = \mathbf{com}_1$

Typically easy to prove

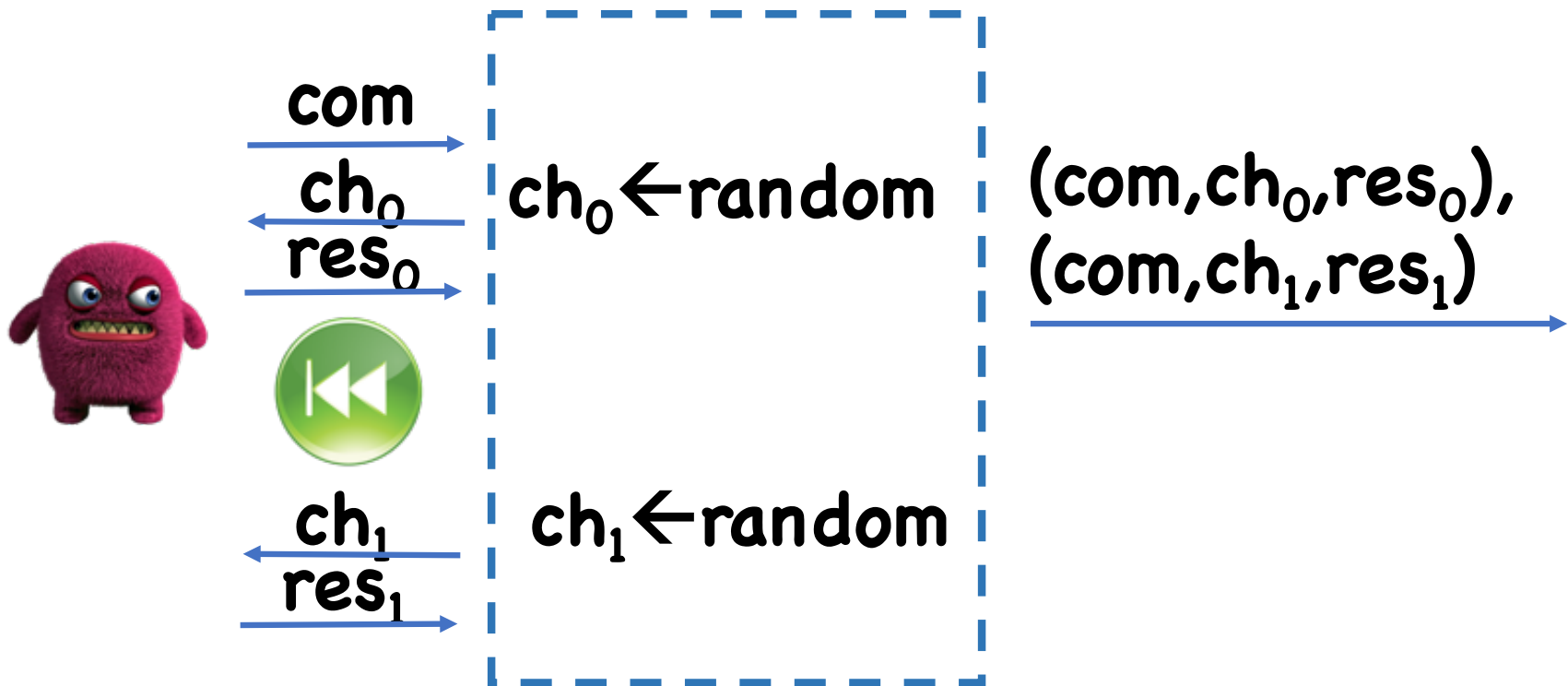


Knowledge Soundness

Classical Reduction



Classical Reduction



Quantum Rewinding?

Problem ([van de Graaf'97, Ambainis-Rosmanis-Unruh'14]):

Extracting **res**₀ alters
adversary's state



Adversary may no
longer work on **ch**₁

[Ambainis-Rosmanis-Unruh'14]:
Separation relative to quantum oracle

[Amos-Georgiou-Kiayias-Z'19]:
Relative to classical oracle

Solution?

Good news: No standard model separations known

But: Special soundness still not enough to prove anything

Solution: Add additional properties that allow proof

Prior Work

[Unruh'12]:

Special Soundness + Strict Soundness

[Unruh'17]:

Statistical Soundness

[Alkim-Bindell-Buchmann-Dagdelen-
Eaton-Gutoski-Krämer-Pawlega'17, Kiltz-
Lyubashevsky-Schaffner'17]:

Special Soundness + Lossy Keys

[Unruh'15]:

Alternative Construction

Limitation of Prior Work

Limitation: Ensuring extra properties or modifying scheme often makes protocols inefficient

In particular, does not apply to [Lyubashevsky'11] or the most efficient schemes based on it

Idea Behind [Unruh'12]

Assume Weaker Guarantee (for now):

If we only observe whether adversary succeeds (but not **res**), then rewinding works

+ **Strict Soundness:**
res unique,
given **(com,ch)**

+

Obs. Lemma with $\dagger=1$
→ Can observe **res** without
affecting success probability

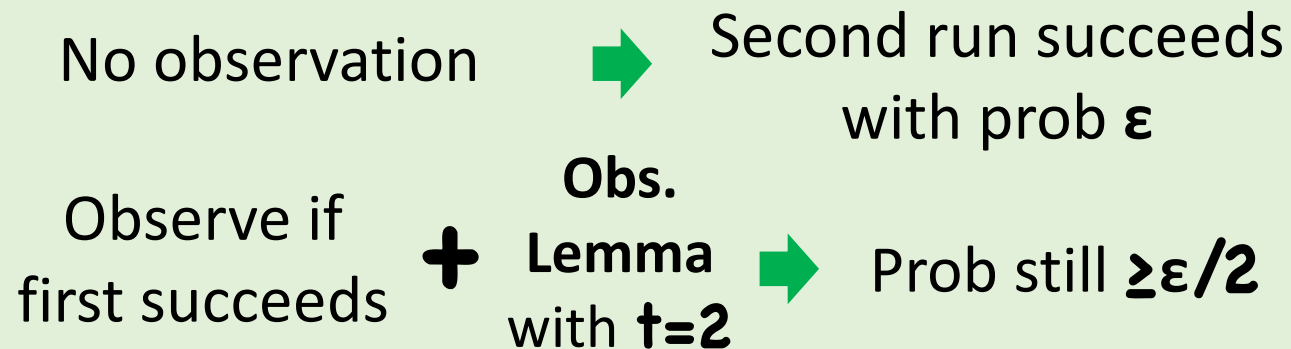
=

Knowledge Soundness

Idea Behind [Unruh'12]

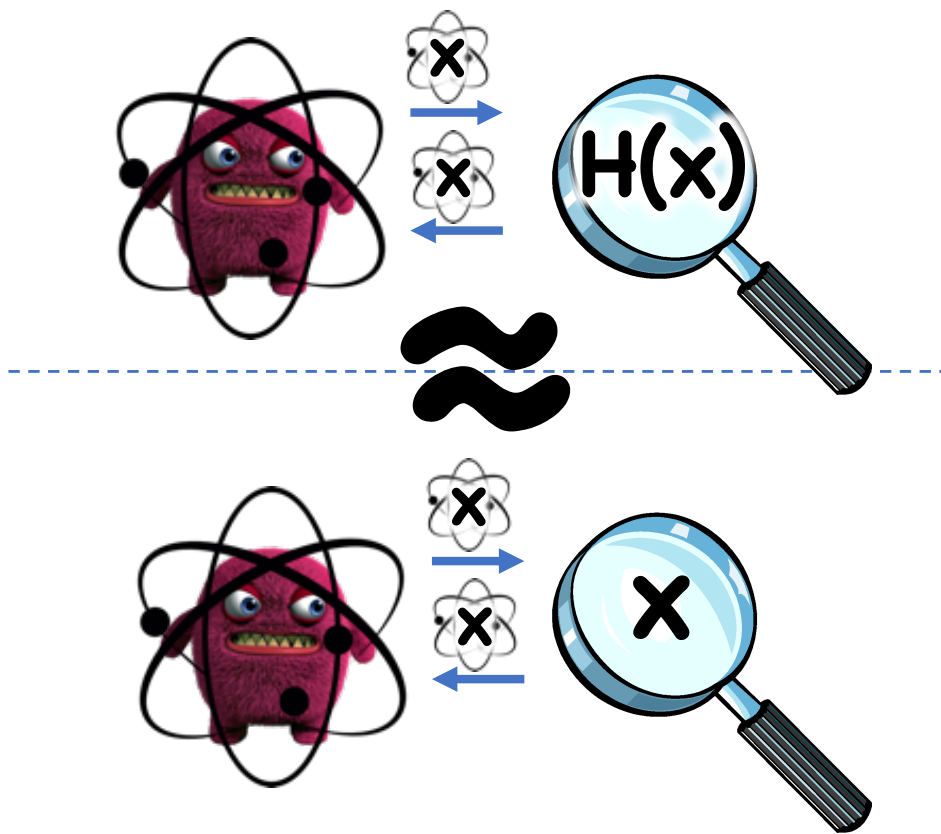
Thm [Unruh'12]: Weaker Guarantee holds

(My) Intuition:



Not Enough: Need both runs to succeed!

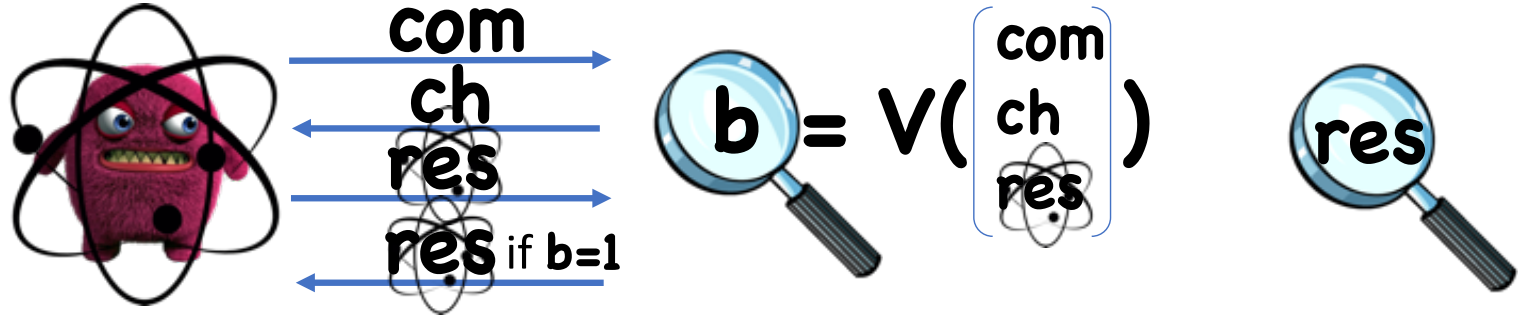
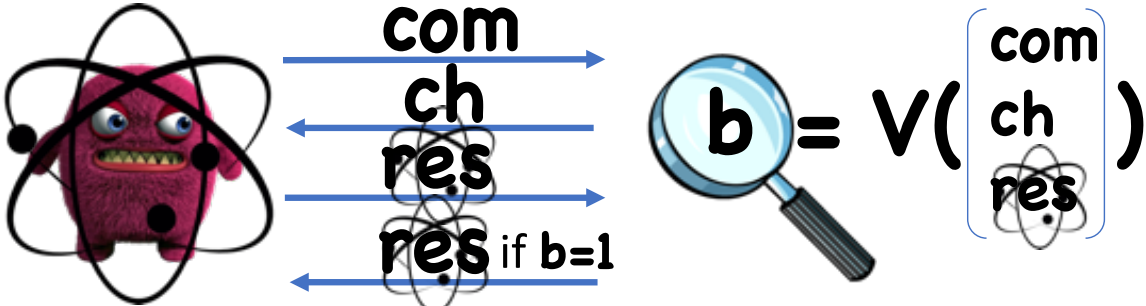
Segue: Collapsing Hash Functions [Unruh'16a]



By observer effect,
second message different
from first message

“right” generalization
of collision resistance
for post-quantum

Idea: Collapsing Sigma Protocols



Idea: Collapsing Sigma Protocols

Thm:

Collapsing +
Special Soundness  Knowledge
Soundness

Proof:

Essentially same as [Unruh'12], except
observing **res** now computational

(Also in [Don-Fehr-Majenz-Schaffner'19])

Final Piece: Collapsing Protocols

For this talk: focus on simpler problem of collapsing hash functions

Goal: Prove SIS is Collapsing

“short”

$$\begin{matrix} \text{“short”} \\ \swarrow \\ \mathbf{x} \end{matrix} \xrightarrow{\quad} \left[\mathbf{A} \right] \cdot \mathbf{x} \in \mathbb{Z}_q$$

Basically enough to prove [Lyu'12]

Existing Collapsing Hash Functions?

From Random Oracles

[Unruh'16a, Unruh'17b,
Czajkowski-Bruinderink-Hülsing-
Schaffner-Unruh'18]

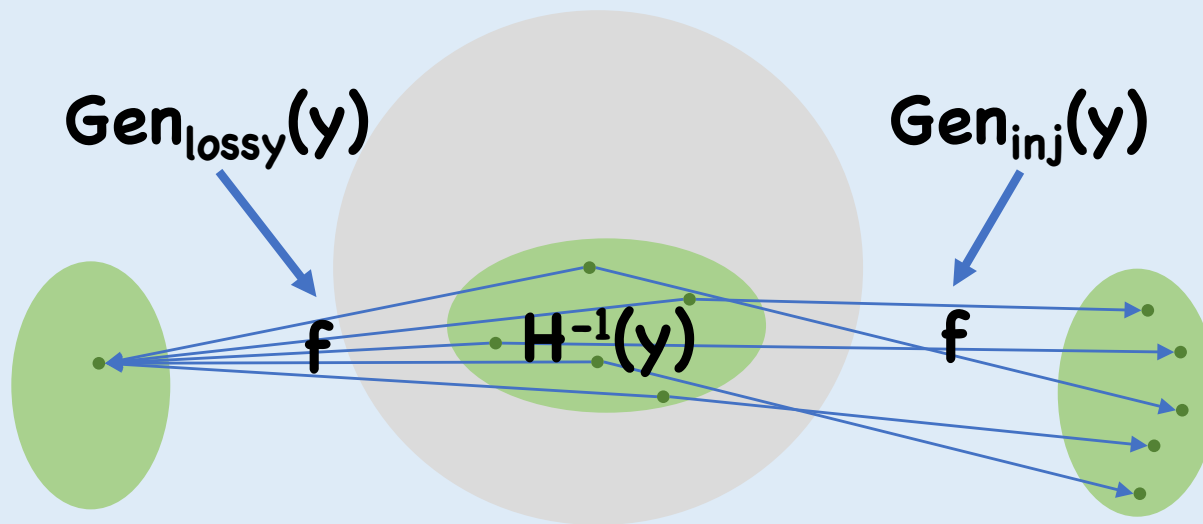
From Lossy Functions

[Unruh'16b]

SIS contains neither a random
oracle nor a lossy function!

Our Solution: Associated Lossy Functions

Def: Associated Lossy Function for \mathbf{H} :

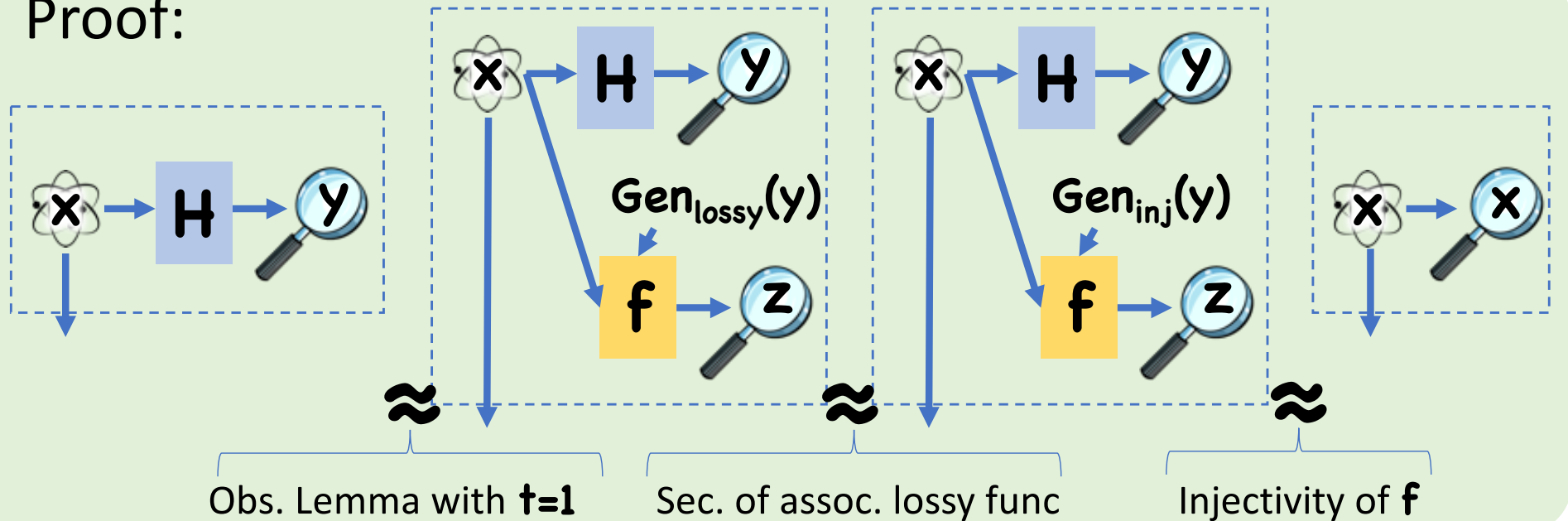


$$\text{Gen}_{\text{lossy}}(y) \approx_c \text{Gen}_{\text{inj}}(y)$$

Our Solution: Associated Lossy Functions

Thm:
 H has associated lossy func $\rightarrow H$ is collapsing

Proof:



Associated Lossy Functions for SIS

Thm (informal): Assuming LWE, SIS has associated lossy functions

Associated Lossy Functions for SIS

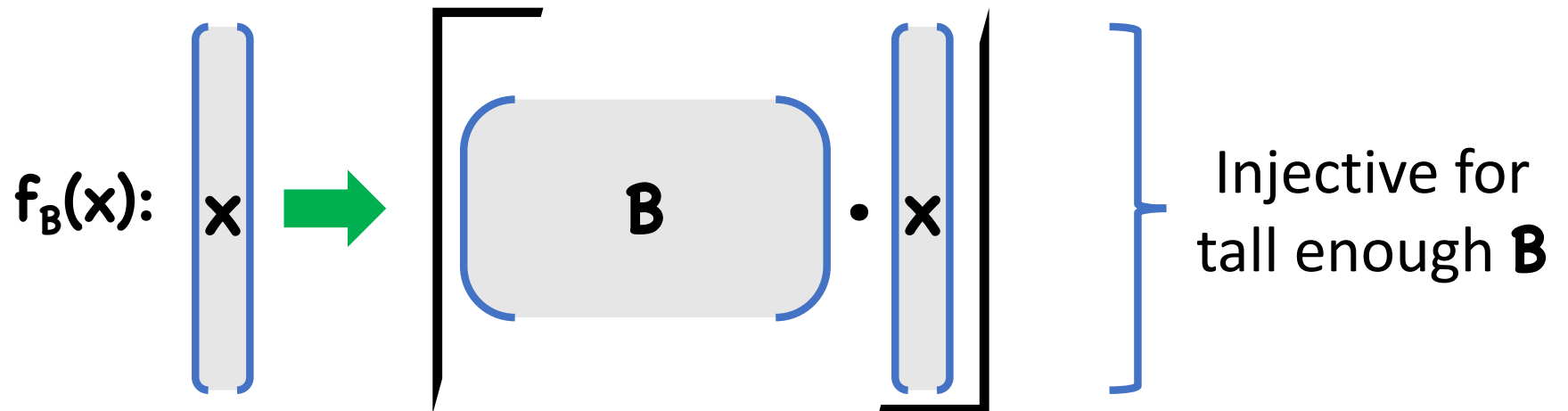
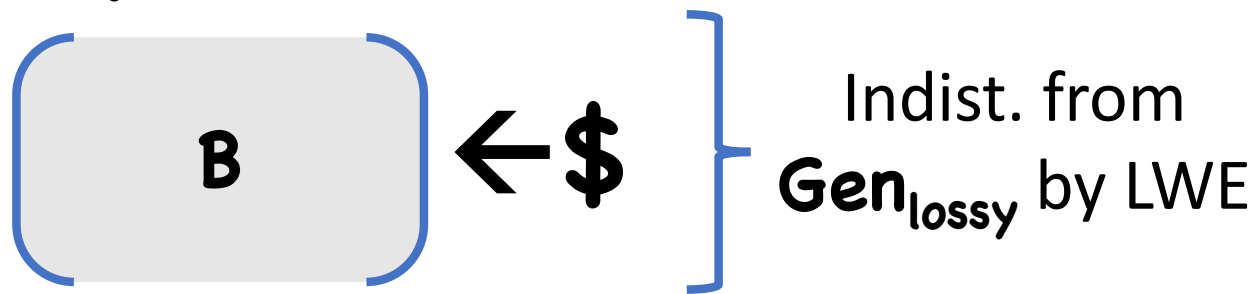
$\text{Gen}_{\text{lossy}}(y):$

$$\boxed{B} = \boxed{u} \cdot \boxed{A} + \boxed{\begin{matrix} e \\ \text{"short"} \end{matrix}}$$

$$f_B(x): \boxed{x} \xrightarrow{\text{green arrow}} \boxed{\boxed{B} \cdot \boxed{x}} = \boxed{\boxed{u} \cdot \boxed{y}}$$

Associated Lossy Functions for SIS

$\text{Gen}_{\text{inj}}(\mathbf{y})$:



Caveat

Correctness of **Gen**_{lossy}
needs super-poly **q**

But, most efficient
protocols have poly **q**

Solution:

Relax assoc.
lossy func



Relaxed notion
of collapsing



Good enough
for rewinding

Works for any polynomial **q**

Takeaway

any assoc. lossy function
implies collapsing

Collapsing probably much more
common than previously thought (can
potentially use crazy tools like iO)

Maybe unsurprising that collapsing
counterexamples are hard to find

[Z'19a]: Counterexamples
useful for quantum money

?