Security Reductions in A Quantum World

Mark Zhandry (Princeton & NTT Research)

Security Proofs

Crypto security "proof" = Computational Assumption ₱ + Reduction from ₱

Should be well-studied and widely believed

Concrete assumptions: Hardness of FACTORING, DLOG, LWE

Generic assumptions: ∃ OWF, ∃ PKE

In other words, if you can break scheme, you can solve ₱

Enter Quantum

Thm [Shor'94]: $\exists$ Quantum polynomial time (QPT) algorithms solving FACTORING, DLOG

Post-Quantum Crypto = developing crypto secure against quantum attacks

Post-Quantum Security Proofs

Post-quantum security "proof" = *Post-quantum* Assumption ₱ + *Post-quantum* Reduction

Should be well-studied and widely believed

Concrete assumptions: (Quantum) hardness of LWE, …

Generic assumptions: ∃ (quantum immune) OWF, PKE

If you can break scheme *with a quantum computer*, then you can solve ₱ *with a quantum computer*

Main Takeaway

Post-quantum security "proof" $\neq$ *Post-quantum* Assumption P + **Classical Reduction**

BAD NEWS:	**GOOD NEWS:**	**BUT:**
Most crypto literature = classical reduction	Most results translate to quantum trivially	$\exists$ notable exceptions

Even those working with post-quantum tools

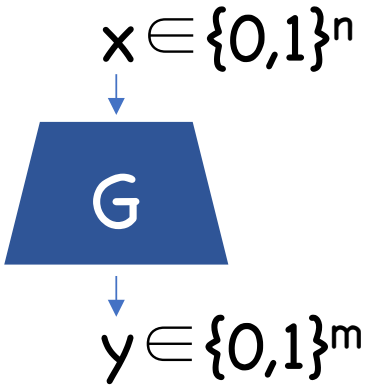
Outline for Today

1st hour: 4 illustrative examples

- Increasing PRG stretch – black box reductions
- PRFs – interaction
- Coin tossing – rewinding
- Goldreich-Levin – running adversary many times

2nd hour: Begin seeing new post-quantum techniques

Example 1: PRG Length Extension



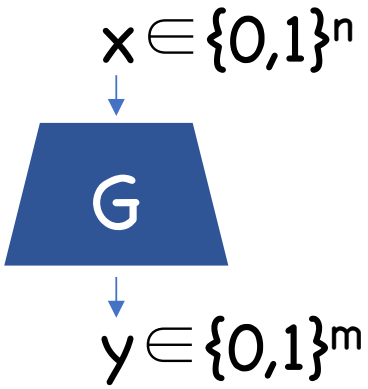
$(m>n)$

Def: G is a secure pseudorandom generator (PRG) if, $\forall$ PPT A, $\exists$ negligible ε such that

$$| \Pr[A(y)=1] - \Pr[A(G(x))=1] | < \varepsilon$$

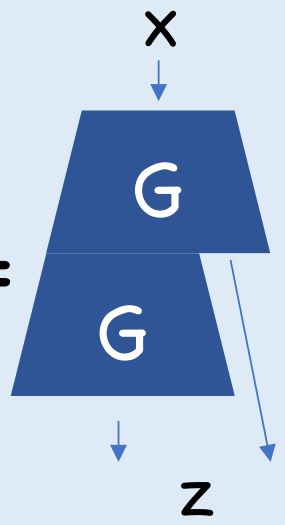
ε called "advantage" of A

Example 1: PRG Length Extension



Suppose **m=n+1**. How to get larger stretch?

Solution: G_2 =

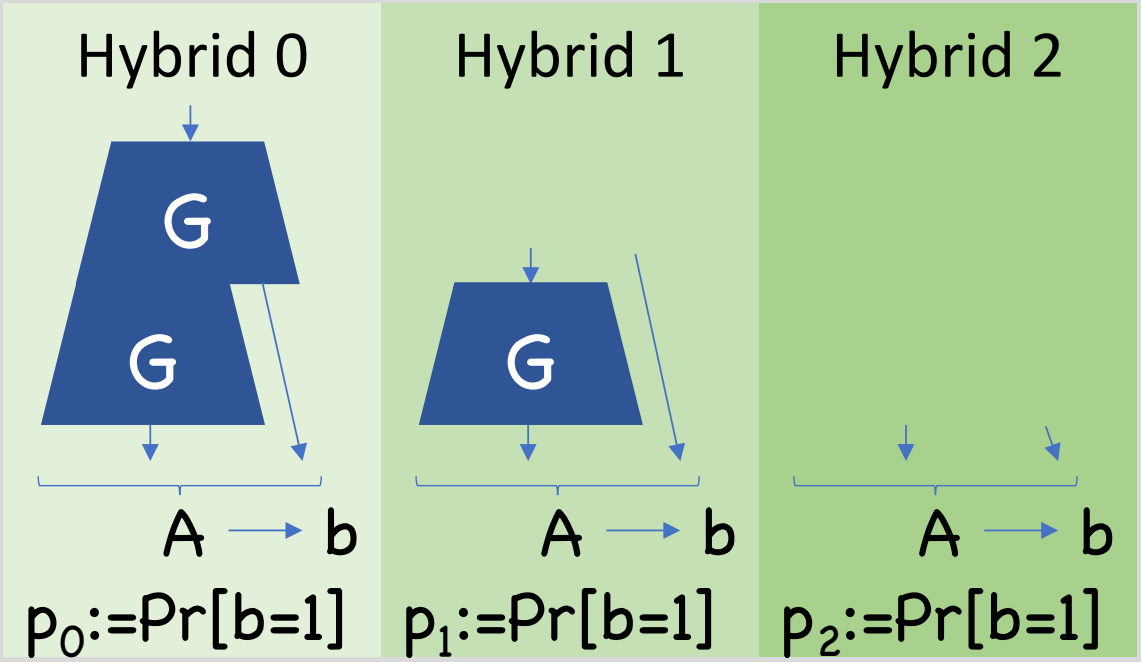


Thm: If G is secure, then so is G_2

Example 1: PRG Length Extension

Proof: Suppose G_2 insecure. Then $\exists$ PPT A, non-negl ε such that

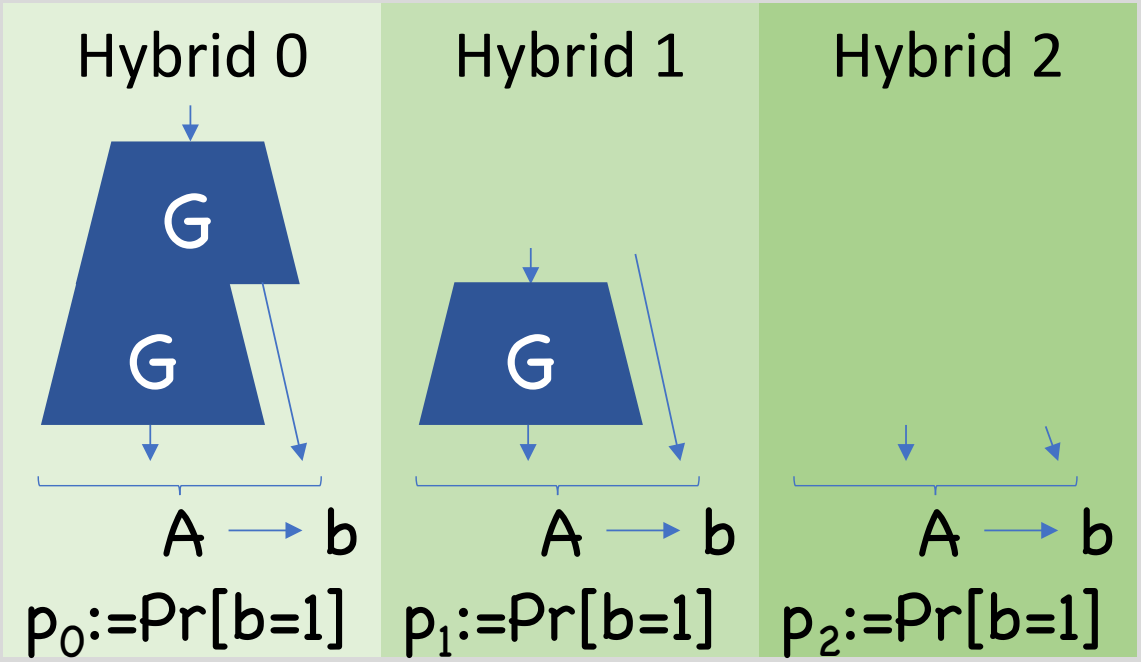
$$| \Pr[A(y)=1] - \Pr[A(G_2(x))=1] | \geq \varepsilon$$



Example 1: PRG Length Extension

Proof: Suppose G_2 insecure. Then $\exists$ PPT A, non-negl ε such that

$$| p_2 - p_0 | \geq \varepsilon$$



Either: $|p_1-p_0| \geq \varepsilon/2$

$B(y_0,y_1) = A(G(y_0),y_1)$

Or: $|p_2-p_1| \geq \varepsilon/2$

$B(y_0,y_1) = A(y_0,y_1,\$)$

In either case, B has advantage $\varepsilon/2$ against security of G

Example 1: PRG Length Extension

What about quantum?

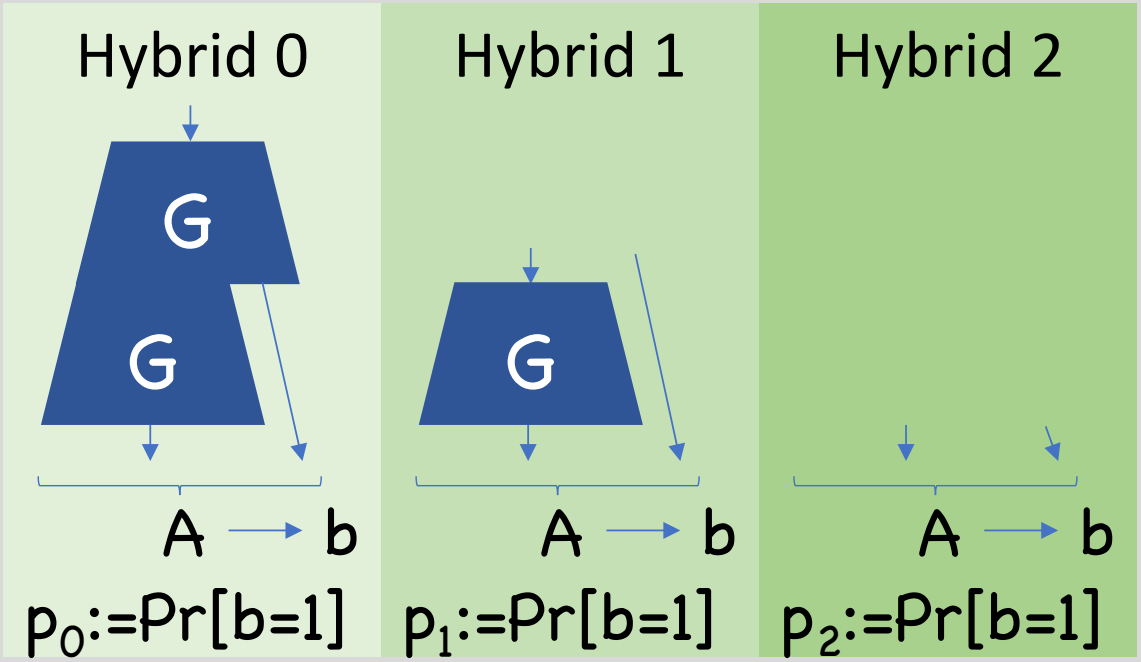
Def: G is a **post-quantum** secure PRG if,
$\forall$ **Q**PT A, $\exists$ negligible ε such that
$$| \Pr[A(y)=1] - \Pr[A(G(x))=1] | < \varepsilon$$

Thm: If G is post-quantum secure, then so is G_2

Example 1: PRG Length Extension

Proof: Suppose G_2 **PQ** insecure. Then $\exists$ **Q**PT A, non-negl ε s.t.

$$| p_2 - p_0 | \geq \varepsilon$$



Either:
$|p_1-p_0|\geq\varepsilon/2$

$B(y_0,y_1)=$
$A(G(y_0),y_1)$

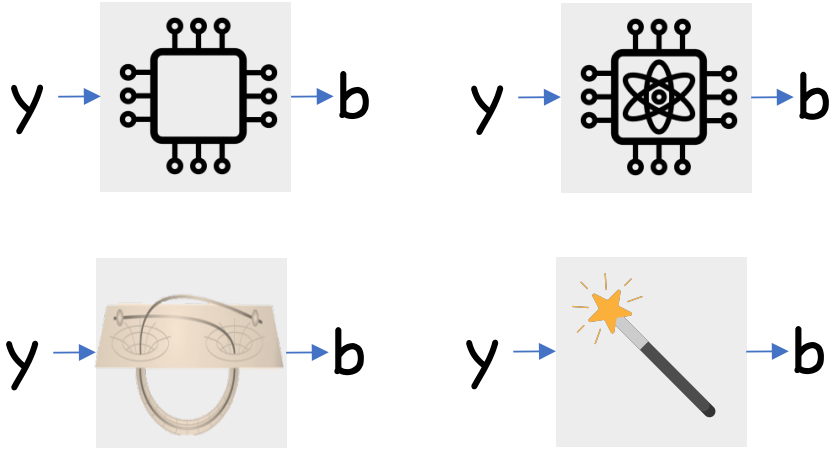
Or:
$|p_2-p_1|\geq\varepsilon/2$

$B(y_0,y_1)=$
$A(y_0,y_1,\$)$

In either case, B has advantage $\varepsilon/2$ against **PQ** security of G

Example 1: PRG Length Extension

Proof for G_2 doesn't care how A works internally, as long as it has non-negligible advantage

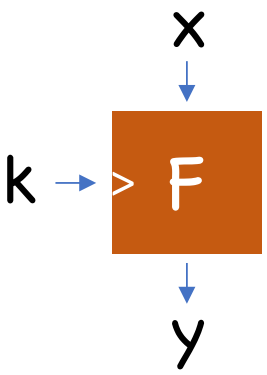


That is, proof treats A as "black box"

Example 1: PRG Length Extension

Key Takeaway: As long as reduction treats $\mathcal{A}$ as a *non-interactive single-run* black box, reduction likely works in quantum setting

Example 2: PRFs



Def: F is a secure pseudorandom function (PRF) if, $\forall$ PPT A, $\exists$ negligible ε such that

$$| \Pr[A^{F(k, \cdot)}()=1] - \Pr[A^{R(\cdot)}()=1] | < \varepsilon$$

Notes:

- k random
- R uniformly random function
- $A^{O(\cdot)}$ means A makes queries on x, receives O(x)

Example 2: PRFs

What is a post-quantum PRF?

$A^{|O(\cdot)\rangle}$ means quantum queries:

$\Sigma \alpha_{x,y} |x,y\rangle$

$\Sigma \alpha_{x,y} |x, y \oplus O(x)\rangle$

Def: F is a **PQ** secure PRF if, $\forall$ **Q**PT A,
$\exists$ negligible ε such that
$| \Pr[A^{F(k,\cdot)}()=1] - \Pr[A^{R(\cdot)}()=1] | < \varepsilon$

Def: F is a **Fully Quantum** secure PRF if,
$\forall$ **Q**PT A, $\exists$ negligible ε such that
$| \Pr[A^{|F(k,\cdot)\rangle}()=1] - \Pr[A^{|R(\cdot)\rangle}()=1] | < \varepsilon$

Example 2: PRFs

Is there a difference? YES!

Proof: Embed Simon's oracle/period finding

$$PRF'((k,z) , x) = PRF(k, \{x, x \oplus z\})$$

Example 2: PRFs

Ok. Which definition do we want?

It depends

Example 2a: PRFs → CPA-secure encryption

$$\mathrm{Enc}(k,m) = \begin{array}{l} r \leftarrow \$ \\ c = (r,\ F(k,r) \oplus m) \end{array}$$

Encrypter (honest) chooses r → always classical

PQ security suffices

Example 2: PRFs

Ok. Which definition do we want?

It depends

Example 2b: PRFs → MAC

$$MAC(k,m) = F(k,m)$$

Security model lets attacker choose m, but signer (honest) actually computes MAC

Can attacker force signer to MAC superpositions?

Example 2: PRFs

Ok. Which definition do we want?

It depends

Example 2c: PRFs → Pseudorandom quantum states

[Ji-Liu-Song'18,Brakerski-Shmueli'19]

$$\Sigma_x (-1)^{F(k,x)} |x\rangle$$

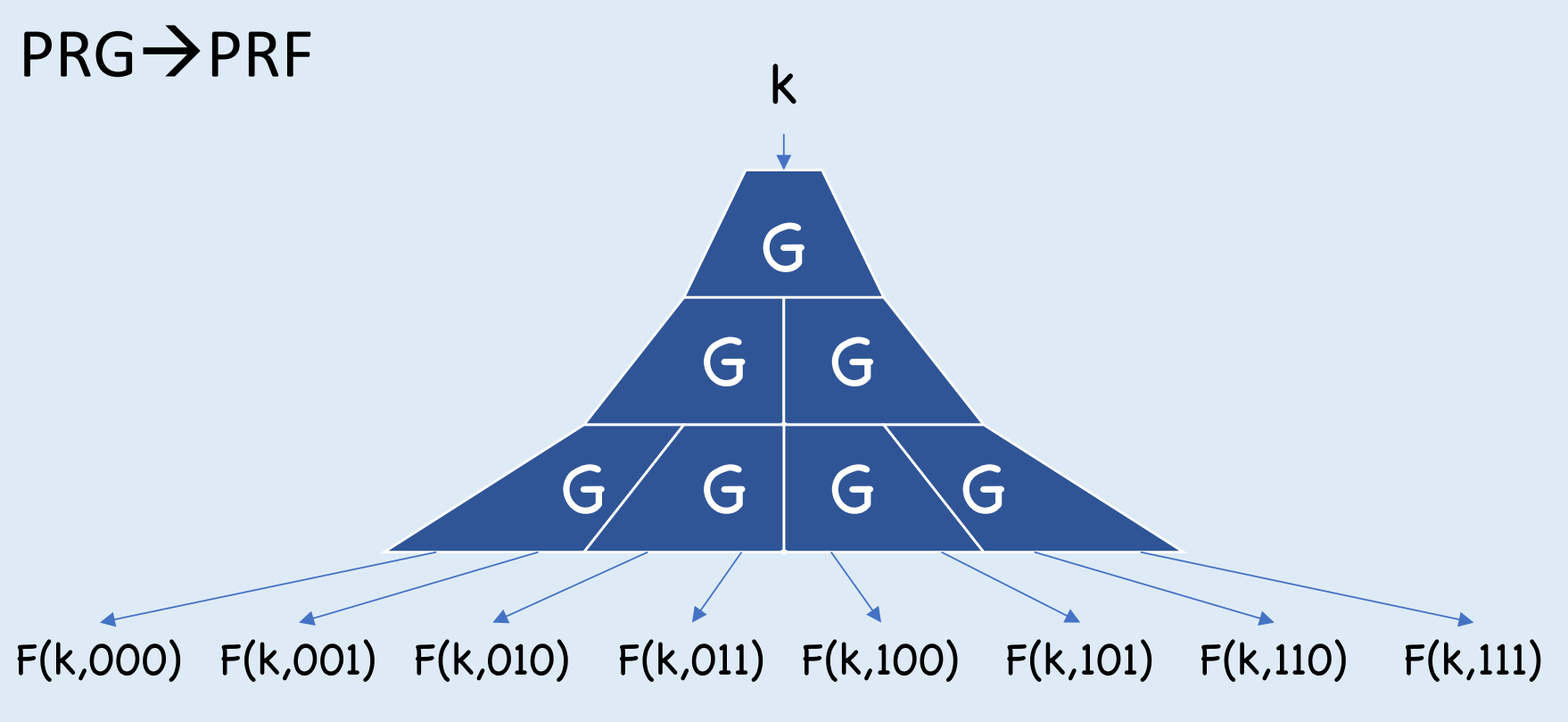
Generation of state makes superposition query to F

Need full quantum security

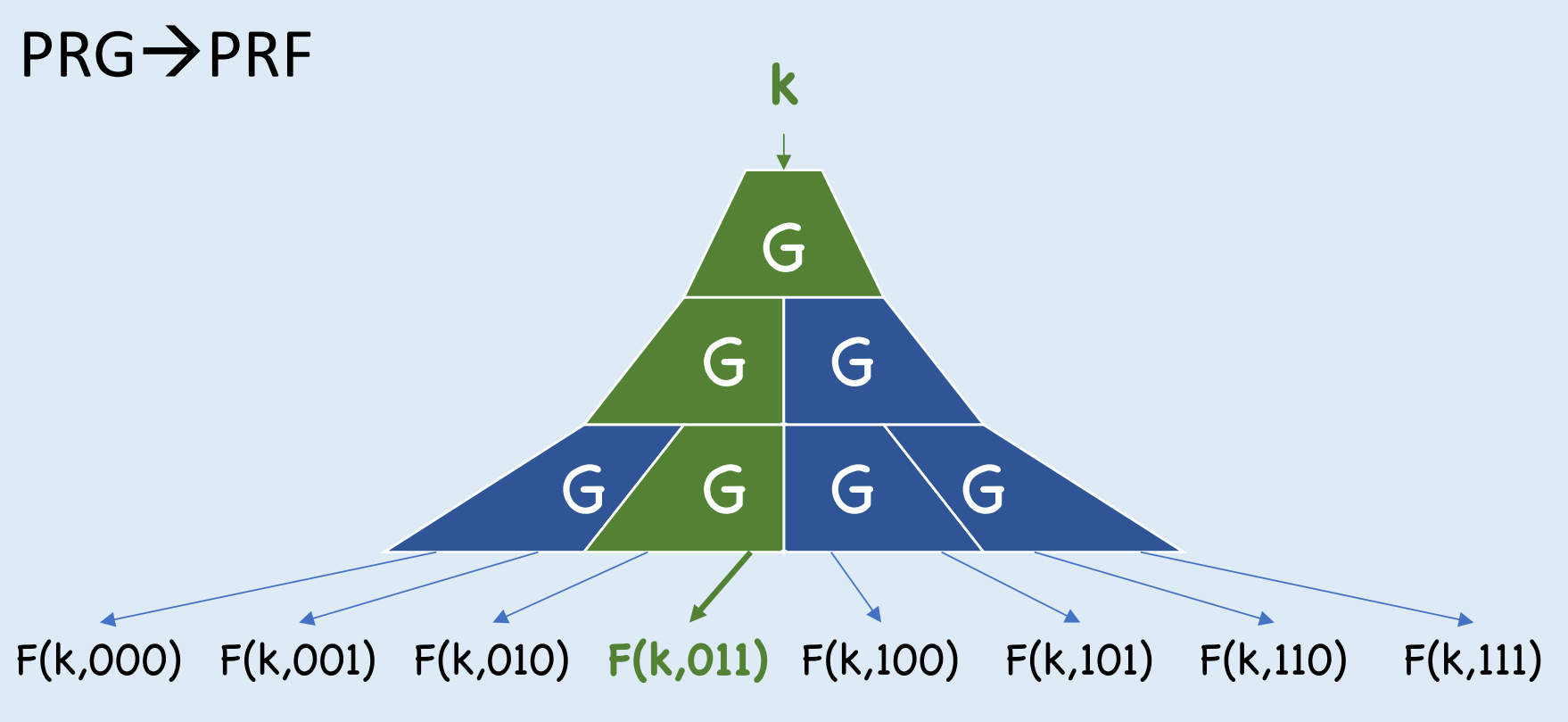
Example 2: PRFs

So then, what does a classical proof give us?

Example 2: PRFs



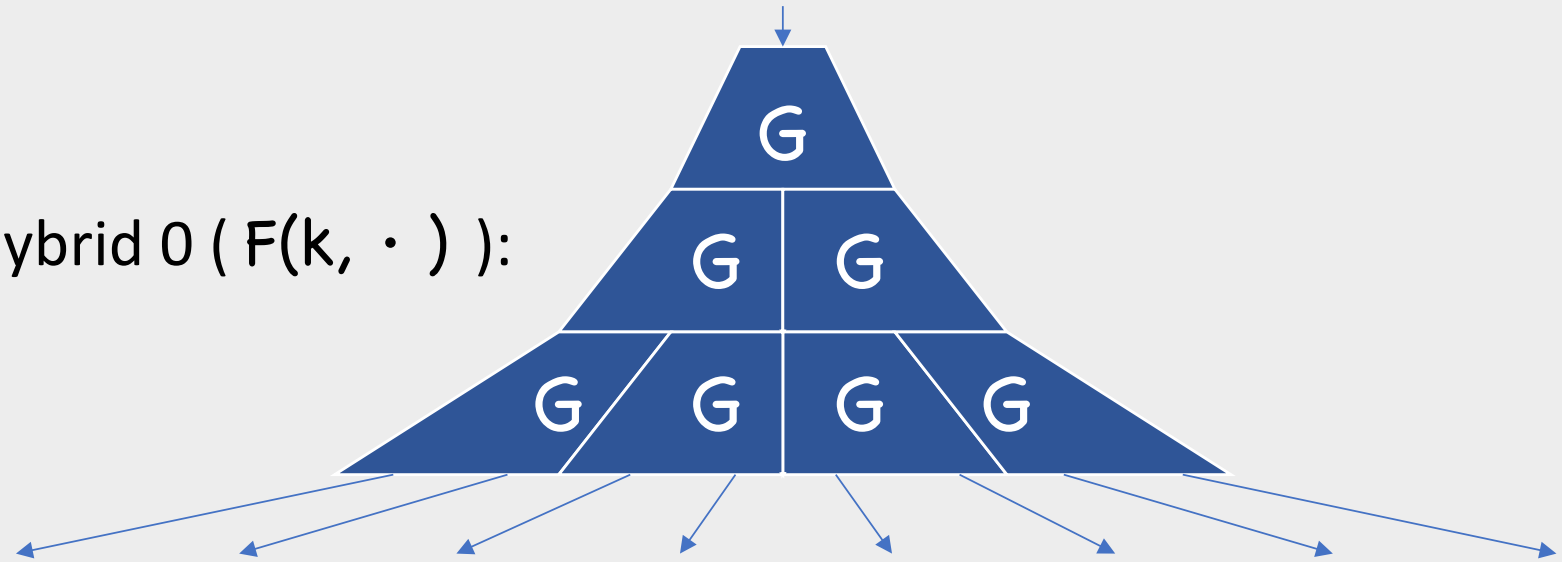
Example 2: PRFs



Example 2: PRFs

Classical proof, step 1: Hybrid

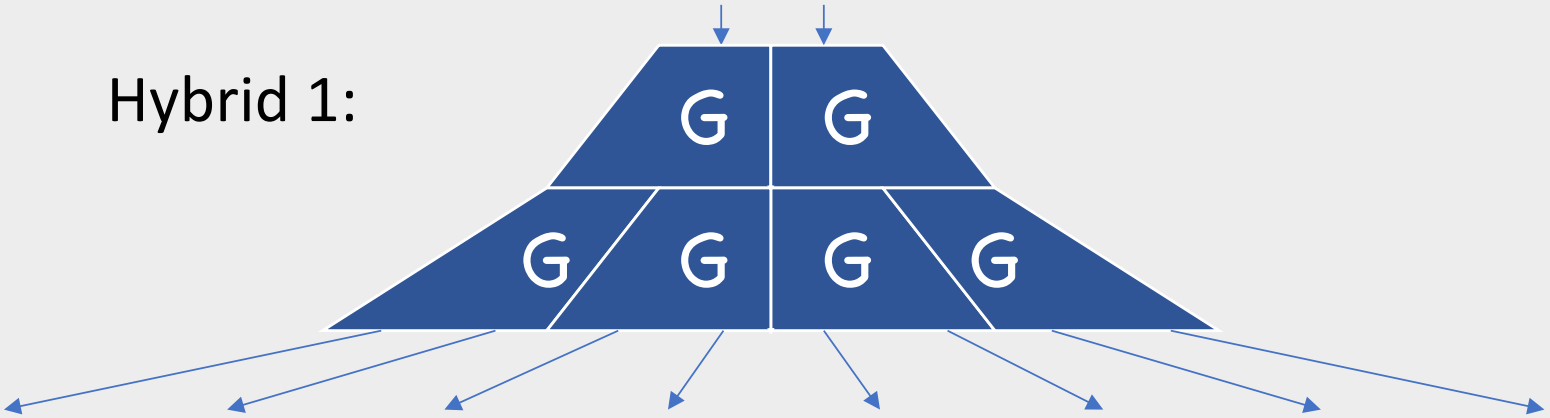
Hybrid 0 (F(k, ·)):



Example 2: PRFs

Classical proof, step 1: Hybrid

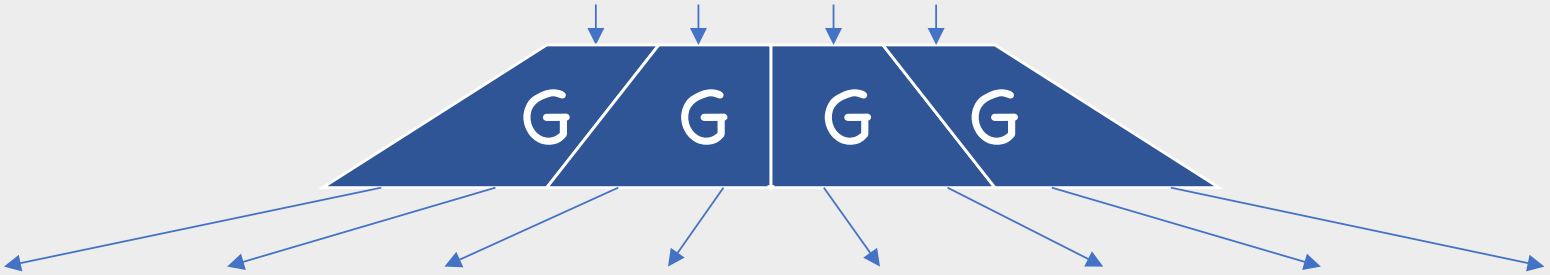
Hybrid 1:



Example 2: PRFs

Classical proof, step 1: Hybrid

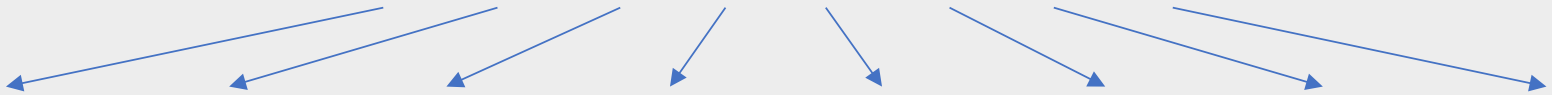
Hybrid 2:



Example 2: PRFs

Classical proof, step 1: Hybrid

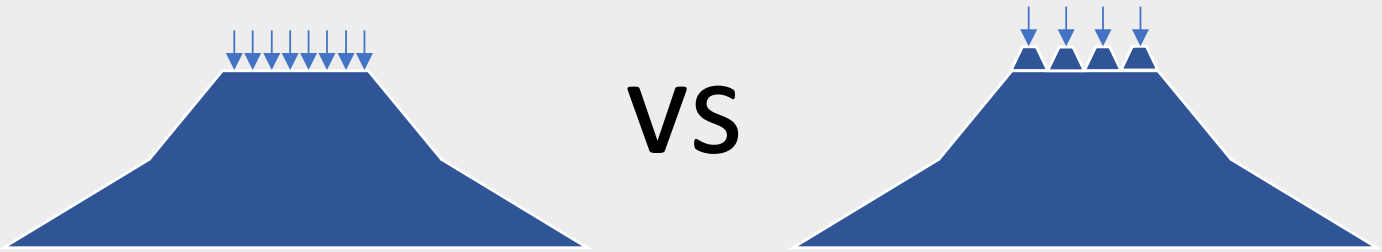
Hybrid n ($R(\cdot)$):



Example 2: PRFs

Classical proof, step 1: Hybrid

$$\exists i \text{ s.t. } | \Pr[A^{\text{Hybrid } i+1}() = 1] - \Pr[A^{\text{Hybrid } i}() = 1] | \geq \varepsilon/n$$

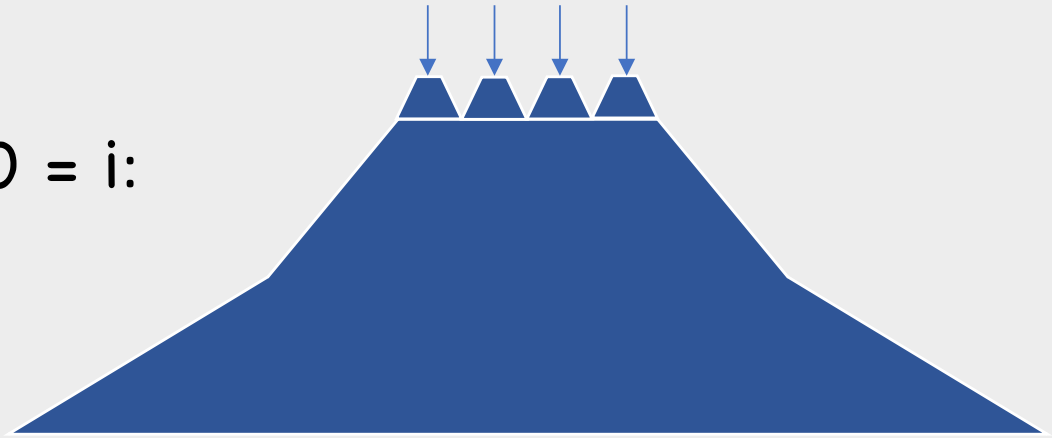


Step 1 makes sense if A classical, post-quantum, or fully quantum

Example 2: PRFs

Classical proof, step 2: Another hybrid

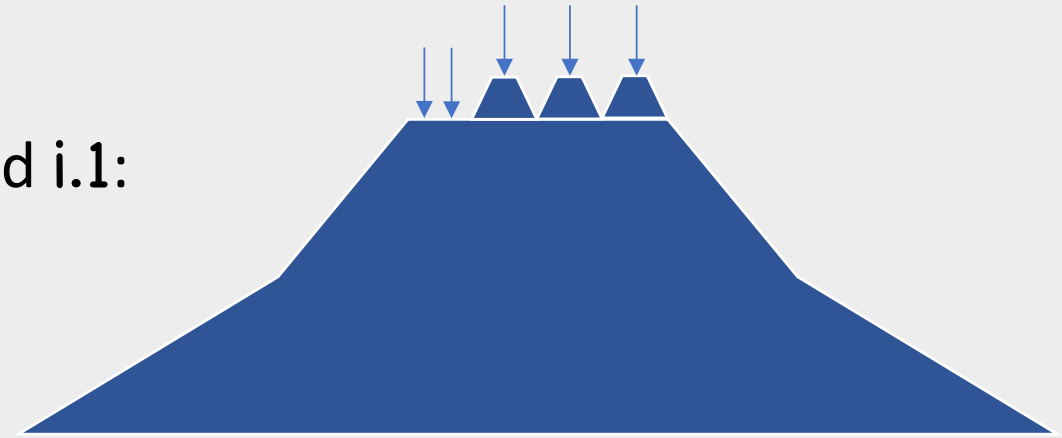
Hybrid $i.0 = i$:



Example 2: PRFs

Classical proof, step 2: Another hybrid

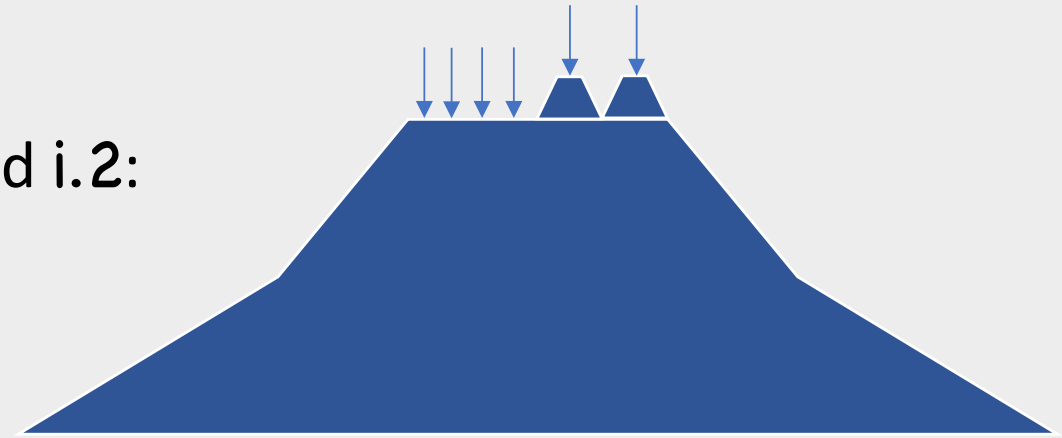
Hybrid i.1:



Example 2: PRFs

Classical proof, step 2: Another hybrid

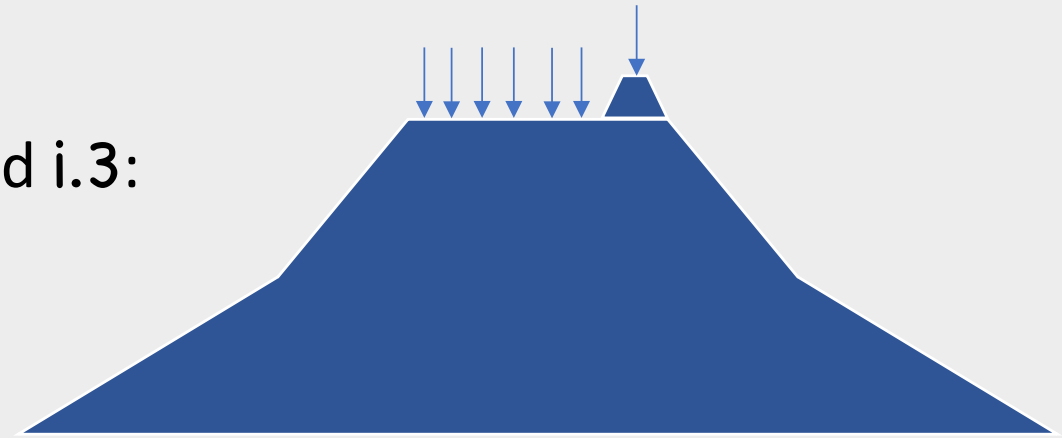
Hybrid i.2:



Example 2: PRFs

Classical proof, step 2: Another hybrid

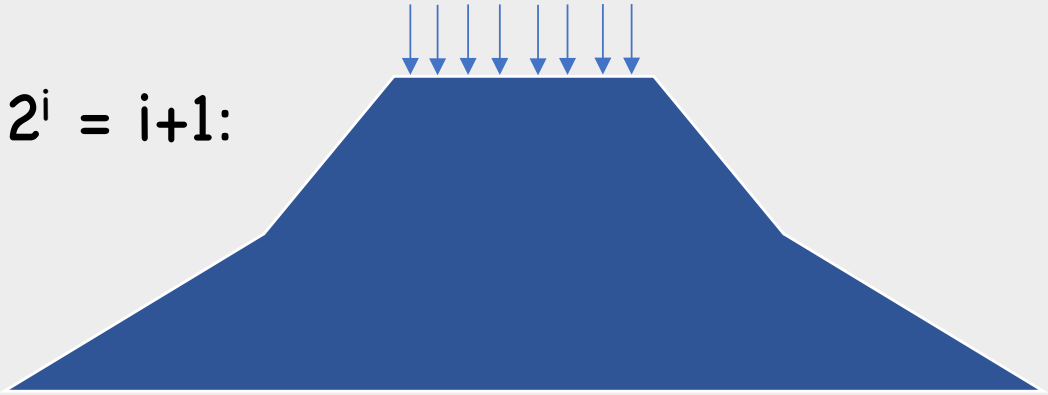
Hybrid i.3:



Example 2: PRFs

Classical proof, step 2: Another hybrid

Hybrid $i.2^i = i+1$:



Problem: 2^i loss potentially exponential

Example 2: PRFs

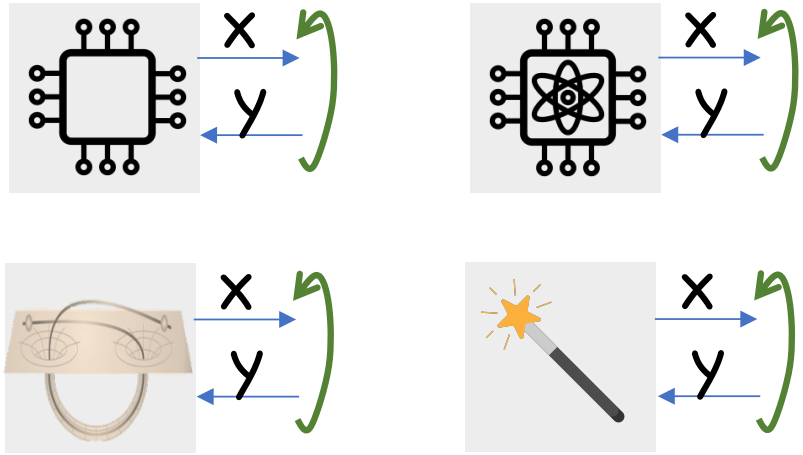
Classical proof, step 2: Another hybrid

Solution: lazy/on-the-fly sampling

q queries → Only hybrid over q “active” positions

Example 2: PRFs

Proof doesn't care how A works internally, as long as it has non-negligible advantage

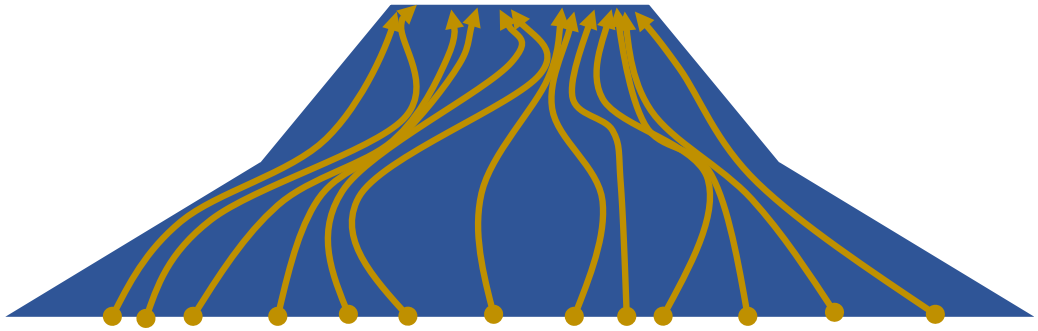


➔ Also post-quantum reduction

Example 2: PRFs

What about full quantum security?

Even single query touches **everything**



Lazy sampling?

Embedding challenges?

Example 2: PRFs

What about full quantum security?

Classical proof is black box, but requires classical queries

Can the proof be fixed for full quantum security?

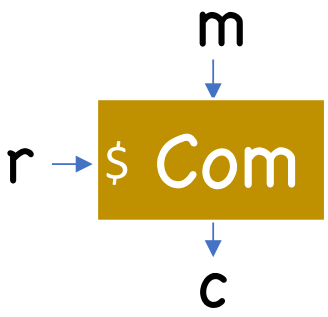
Topic for 2nd hour…

Example 2: PRFs

Key Takeaway: As long as reduction treats ~~A~~ as a *single-run* black box (potentially w/ *classical* interaction), reduction likely works in quantum setting

! But if interaction is quantum, all bets are off

Example 3: Coin Tossing



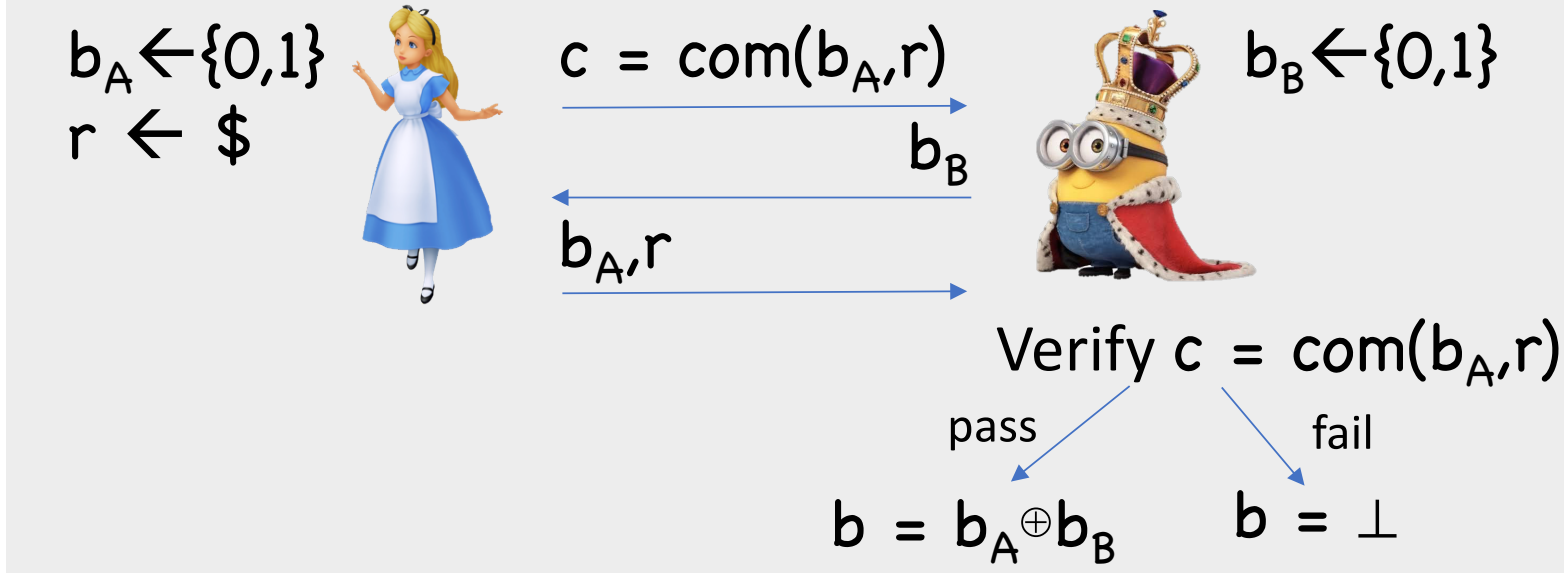
Def: Com is (computationally) binding if, $\forall$ PPT A, $\exists$ negligible ε such that

$$\Pr\left[\begin{array}{c} m_0 \neq m_1 \wedge \\ Com(m_0,r_0)=Com(m_1,r_1) \end{array} : (m_0,r_0,m_1,r_1) \leftarrow A() \right] < \varepsilon$$

Also want hiding, but we will ignore

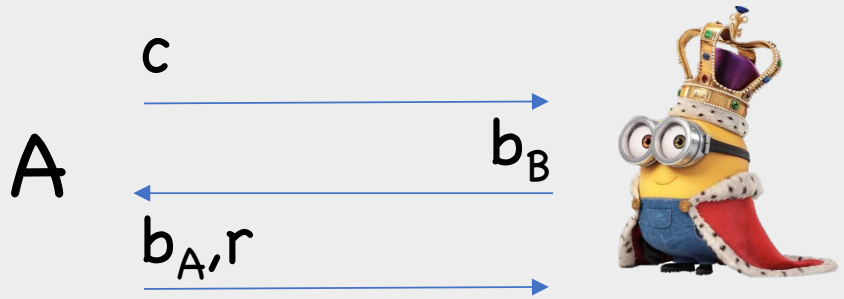
Example 3: Coin Tossing

Simple protocol:



Example 3: Coin Tossing

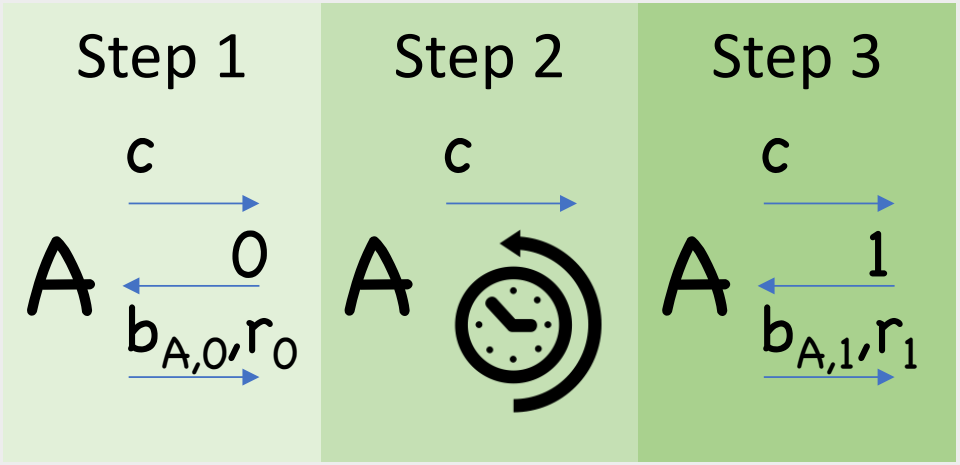
Proof that Alice can't bias b:
Let A be supposed adversary



$Pr[b=0] > ½+\varepsilon$ ➡ For both $b_B=0$ and $b_B=1$, good chance $b_A=b_B$ and $Com(b_A,r)=c$

Example 3: Coin Tossing

Proof that Alice can't bias b:



$$\Pr\left[\begin{array}{l} b_{A,0} = 0 \wedge b_{A,1} = 1 \wedge \\ Com(b_{A,0}, r_0) = Com(b_{A,1}, r_1) = c \end{array}\right] \geq poly(\varepsilon)$$

Example 3: Coin Tossing

What if $\mathcal{A}$ is quantum?

Def: Com is **post-quantum** (computationally) binding if, $\forall$ **Q**PT $\mathcal{A}$, $\exists$ negligible ε such that

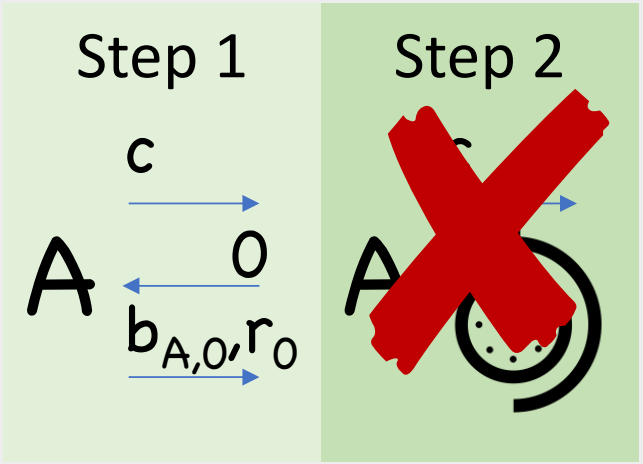
$$\Pr\left[\begin{matrix} m_0 \neq m_1 \wedge \\ Com(m_0,r_0)=Com(m_1,r_1) \end{matrix} : (m_0,r_0,m_1,r_1) \leftarrow \mathcal{A}() \right] < \varepsilon$$

Define coin-tossing goal similarly

Note: adversary's interaction unchanged (unlike Ex 2)

Example 3: Coin Tossing

Proof that **quantum** Alice can't bias **b**?



Measurement principle: extracting $b_{A,0}, r_0$ irreversibly altered A's state

Example 3: Coin Tossing

Thm (Ambainis-Rosmanis-Unruh'14,Unruh'16):
$\exists$ PQ binding ***Com*** s.t. Alice has a near-perfect strategy

I.e., quantumly, ability to produce either of two values isn't the same as ability to produce both simultaneously

Example + how to overcome topic for tomorrow

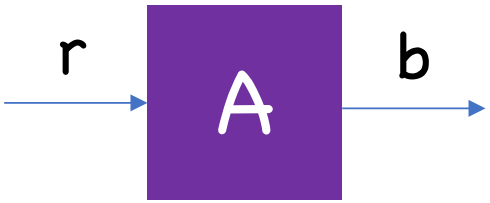
Example 3: Coin Tossing

Key Takeaway: As long as reduction treats A as a *single-run* black box (potentially w/ *classical* interaction), reduction likely works in quantum setting

! But if interaction is quantum, all bets are off

! But if rewinding A, all bets are off

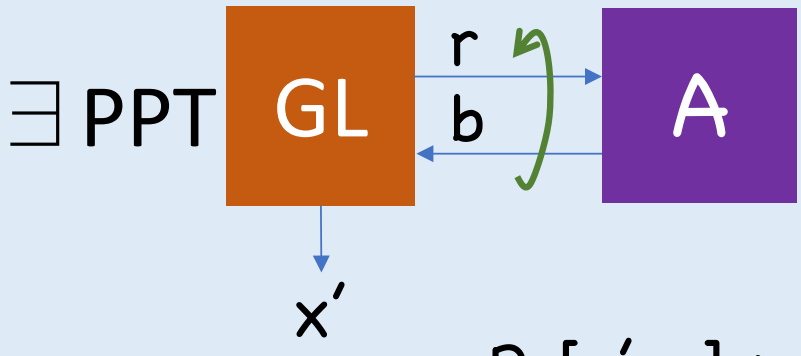
Example 4: Goldreich-Levin



Stateless/rewindable

"GL assumption": A is PPT, $\exists x$: $Pr[A(r) = \langle r,x \rangle] \geq \frac{1}{2} + \varepsilon$

Thm:



$Pr[x'=x] \geq \delta = poly(\varepsilon)$

Example 4: Goldreich-Levin

What happens in quantum setting?

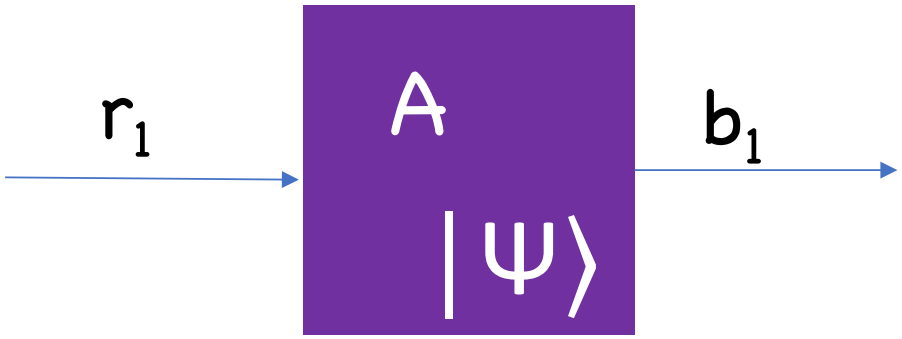
Proof of GL doesn't care how $\mathcal{A}$ works internally, as long as "GL Assumption" holds for **all** queries

$\mathcal{A}$ has classical description (even if quantum alg.) ✓

Good enough for most applications,
e.g. OWF → PRG [HILL'99]

But what if $\mathcal{A}$ contains quantum state?

Example 4: Goldreich-Levin



Measurement principle: extracting b_1 irreversibly altered $|\Psi\rangle$

GL assumption may not hold for 2nd query

Example 4: Goldreich-Levin

Thm (Adcock-Cleve'01): $\exists$ single-query quantum GL algorithm

Proof:

$$\sum_{r\in\{0,1\}^n} |r\rangle \rightarrow \boxed{A} \rightarrow \sum_{r\in\{0,1\}^n} (-1)^{A(r)} |r\rangle \rightarrow \boxed{\text{QFT}/H^{\otimes n}} \rightarrow$$

Results in tighter security reductions!

Example 4: Goldreich-Levin

Key Takeaway: As long as reduction treats ~~A~~ as a black box, potentially w/ *classical* interaction or w/ rewinding to *classical* value, reduction likely works in quantum setting

! But if interaction is quantum, all bets are off

! If rewinding to *quantum* state, all bets are off

Roadmap

New Quantum Attack Models

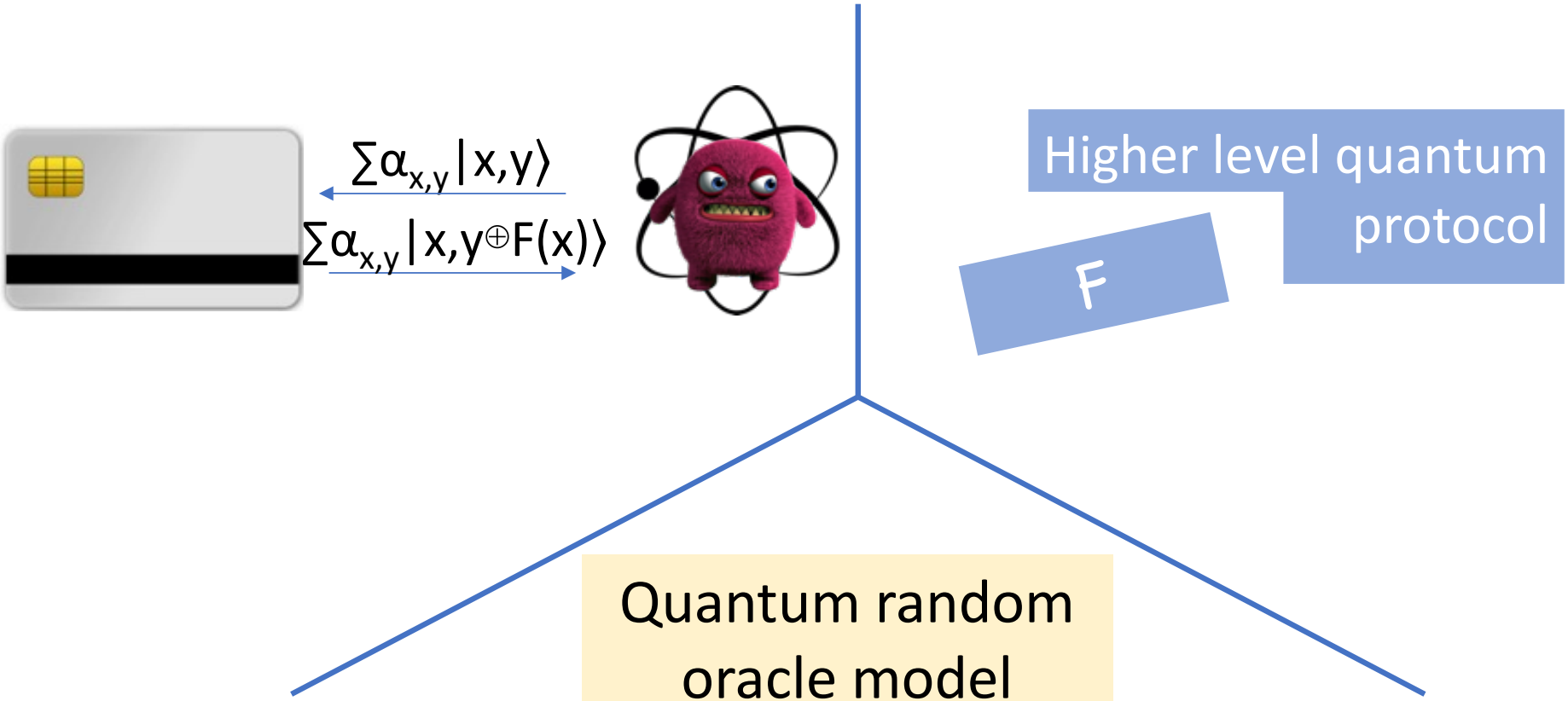
Quantum rewinding

Quantum Random Oracle Model

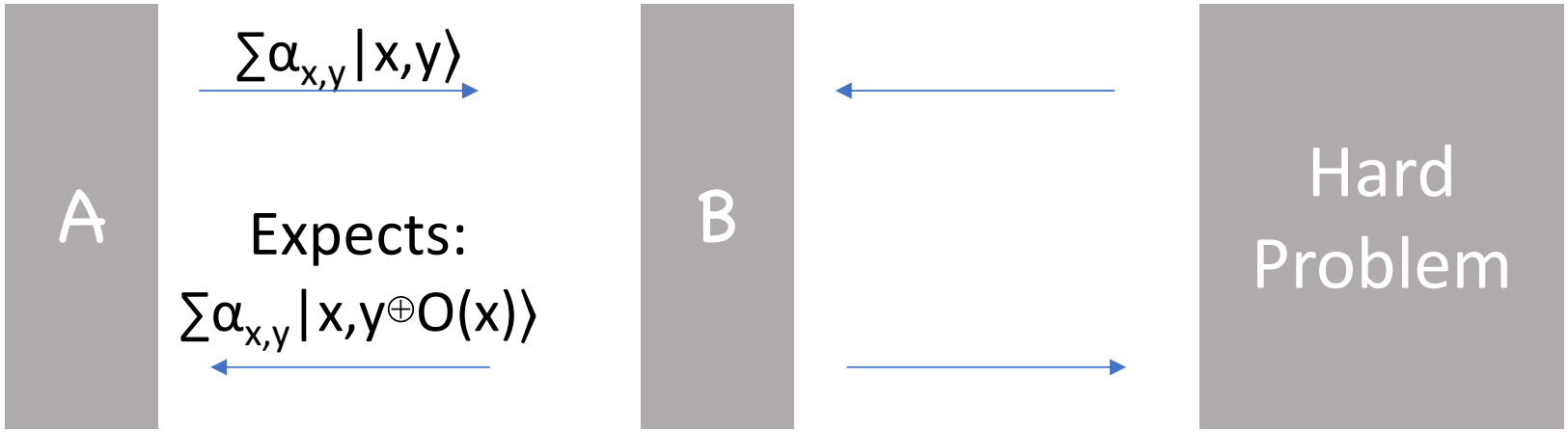
New Quantum Security Models

Mark Zhandry (Princeton & NTT Research)

Motivation


$\sum\alpha_{x,y}|x,y\rangle$
$\sum\alpha_{x,y}|x,y\oplus F(x)\rangle$
Higher level quantum protocol
F
Quantum random oracle model
(starting tomorrow)

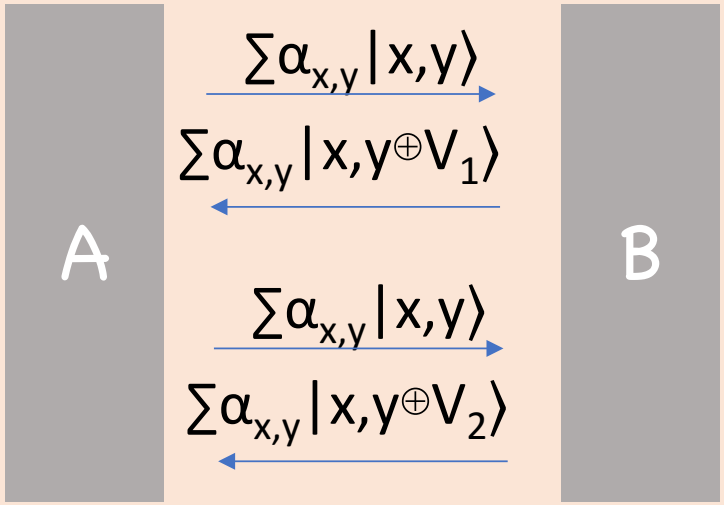
Security Proof Challenges



What does hybrid over queries look like?

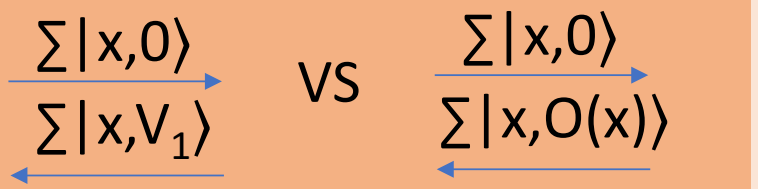
Security Proof Challenges

Take 1: Per QUERY



Problem: repeated queries?

Problem: distinguishing attack



Security Proof Challenges

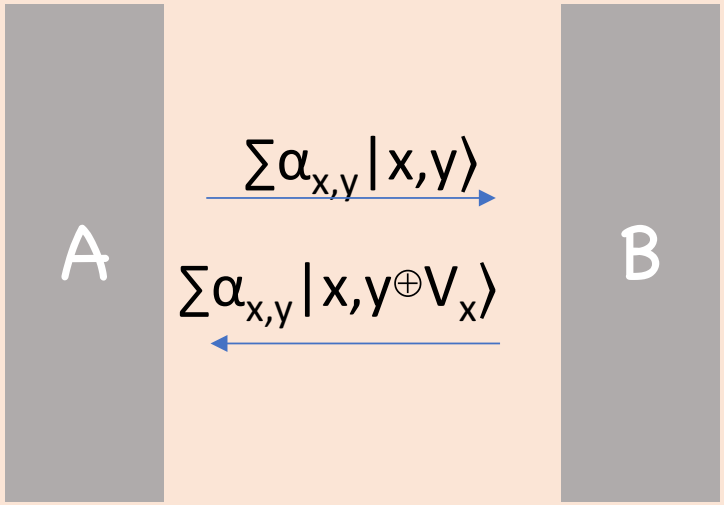
Typical reductions are commit to entire function O at beginning, remain consistent throughout

[Zhang-Yu-Feng-Fan-Zhang'19]: "Committed programming reductions"

Non-committing reductions: topic for later class

Security Proof Challenges

Take 2: Per VALUE



Problem: exp-many values

- Exponential loss in hybrid
- How to simulate efficiently?

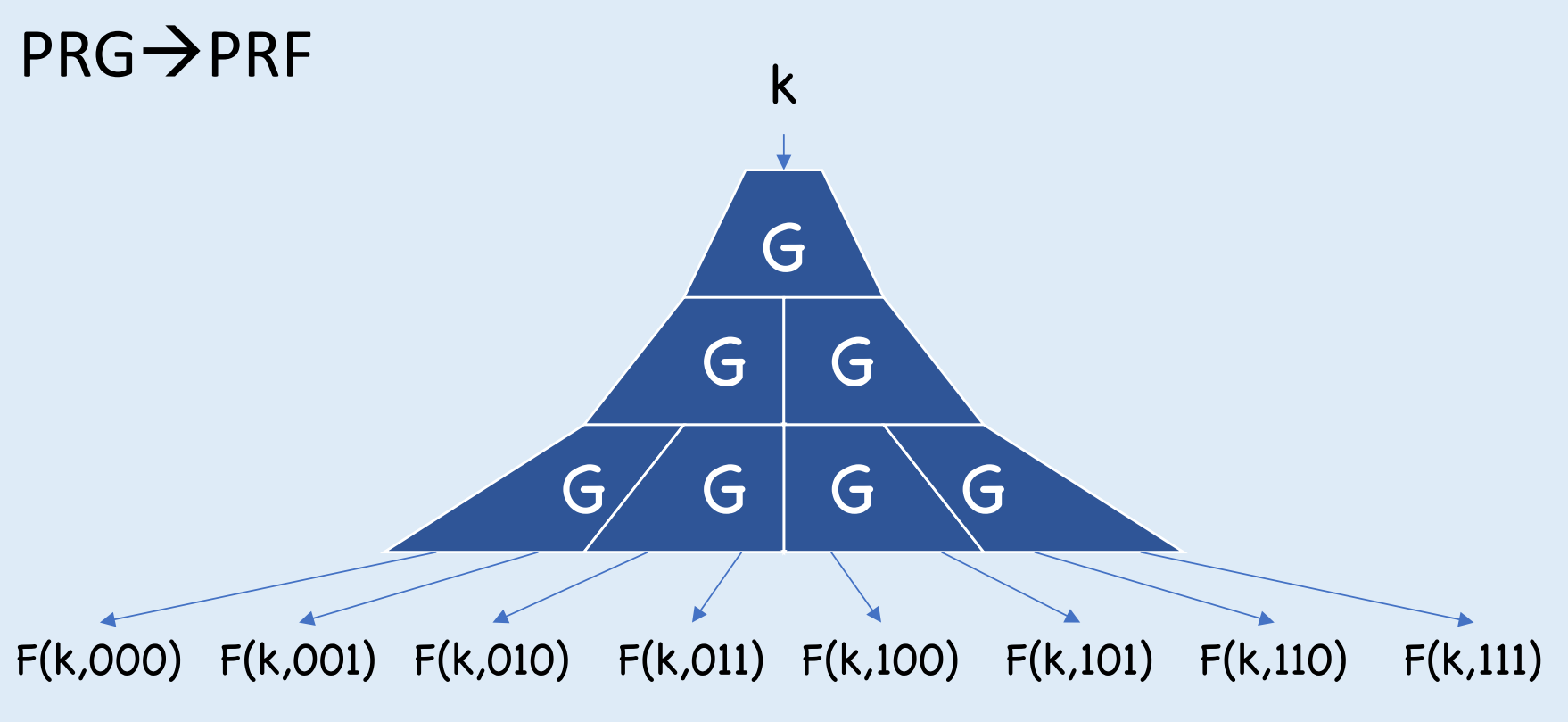
PRF Recap

Def: F is a **Fully Quantum** secure PRF if,
$\forall$ **Q**PT A, $\exists$ negligible ε such that
$| \Pr[A^{|F(k,\cdot)\rangle}()=1] - \Pr[A^{|R(\cdot)\rangle}()=1] | < \varepsilon$

$A^{|O(\cdot)\rangle}$ means quantum queries:

$\Sigma \alpha_{x,y} |x,y\rangle \Rightarrow \Sigma \alpha_{x,y} |x, y \oplus O(x)\rangle$

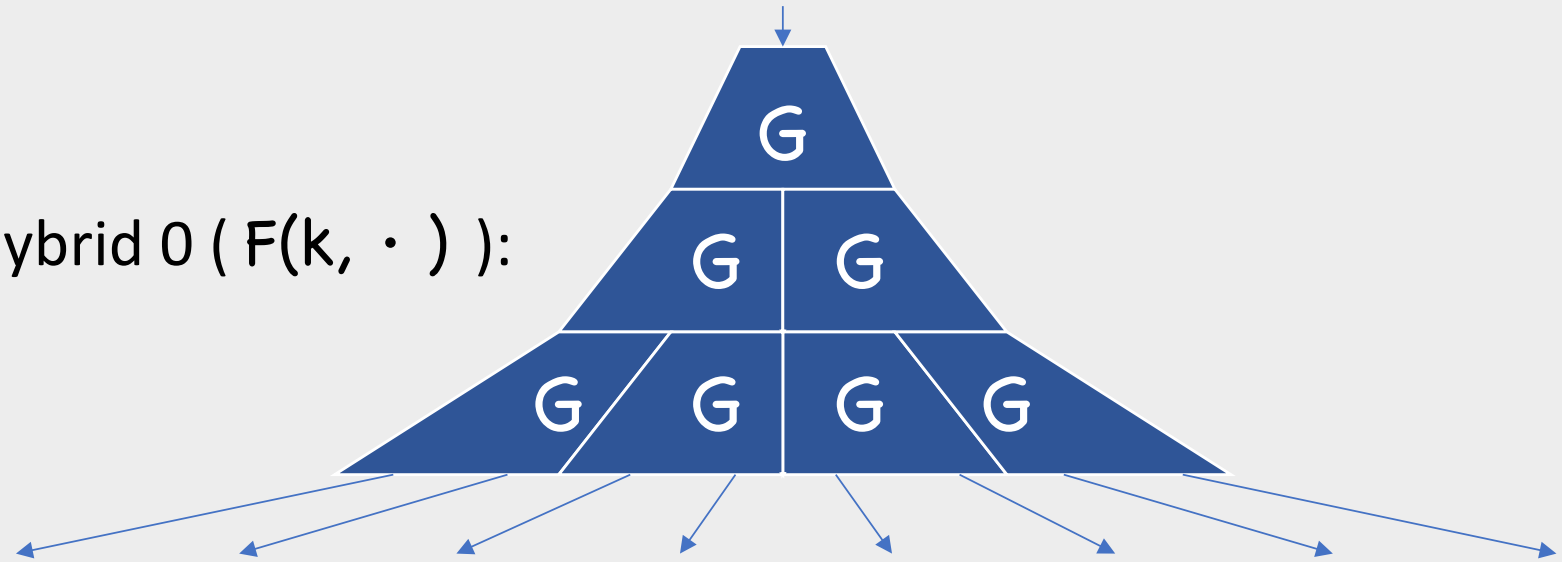
PRF Recap


PRG→PRF
k
G
G
G
G
G
G
G
F(k,000)
F(k,001)
F(k,010)
F(k,011)
F(k,100)
F(k,101)
F(k,110)
F(k,111)

PRF Recap

Proof, step 1: Hybrid

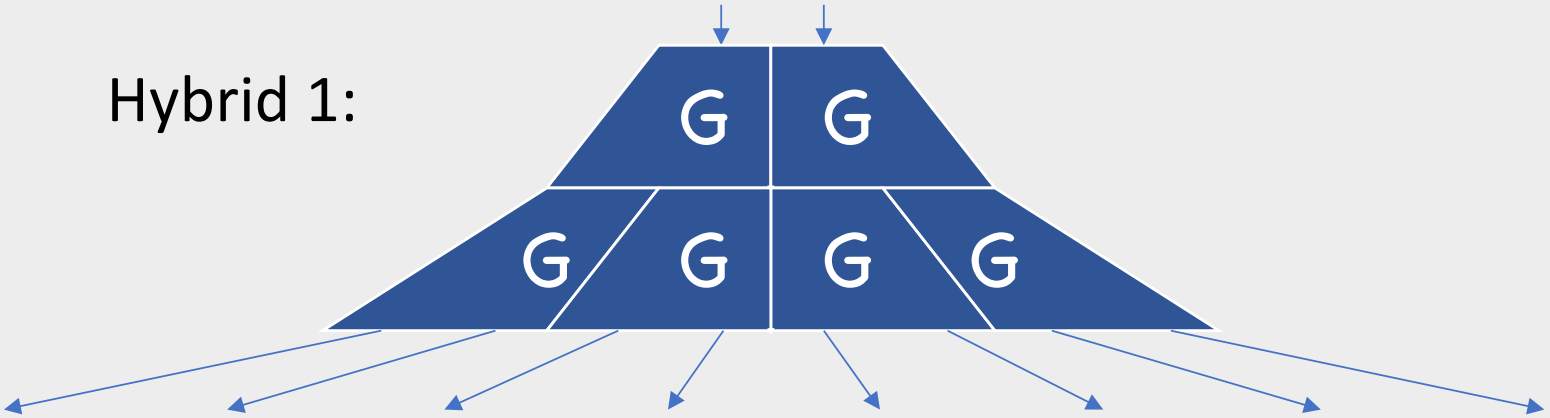
Hybrid 0 (F(k, ·)):



PRF Recap

Proof, step 1: Hybrid

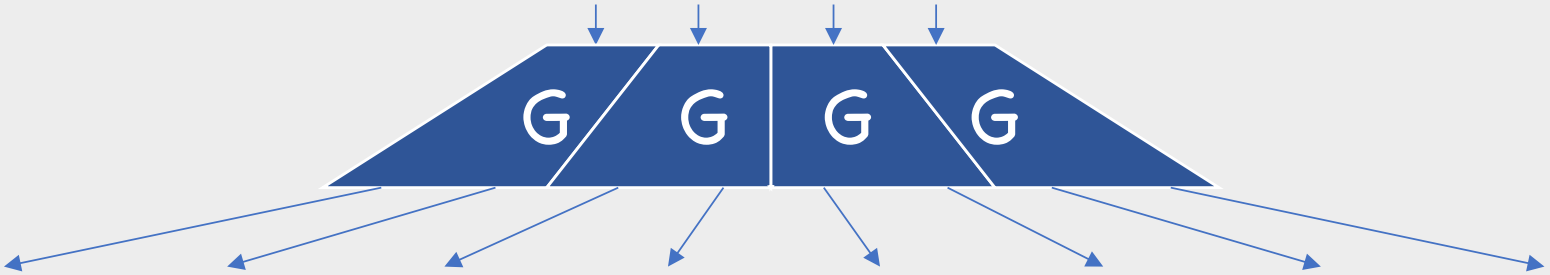
Hybrid 1:



PRF Recap

Proof, step 1: Hybrid

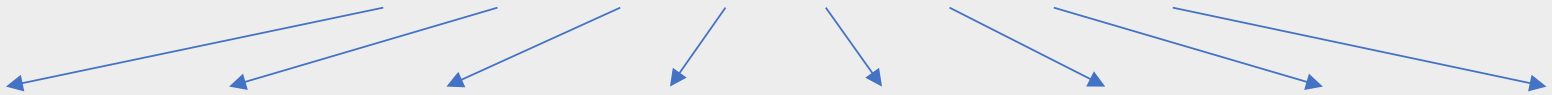
Hybrid 2:



PRF Recap

Proof, step 1: Hybrid

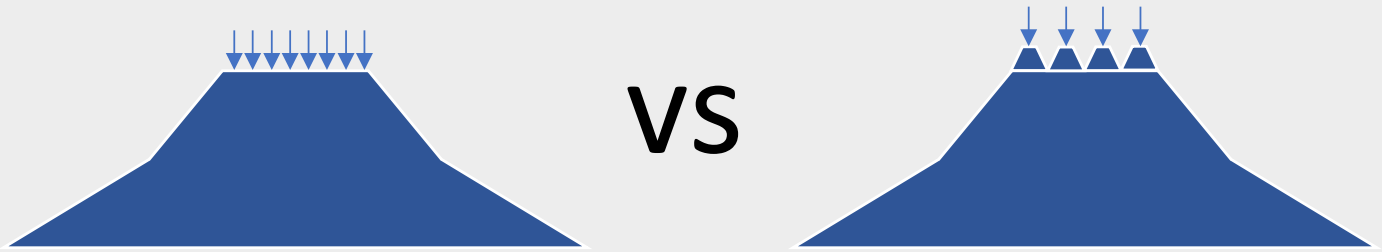
Hybrid n ($R(\cdot)$):



PRF Recap

Proof, step 1: Hybrid

$$\exists\, i \text{ s.t. } |\, Pr[A^{Hybrid\ i+1}() = 1] - Pr[A^{Hybrid\ i}() = 1]\, | \geq \varepsilon/n$$



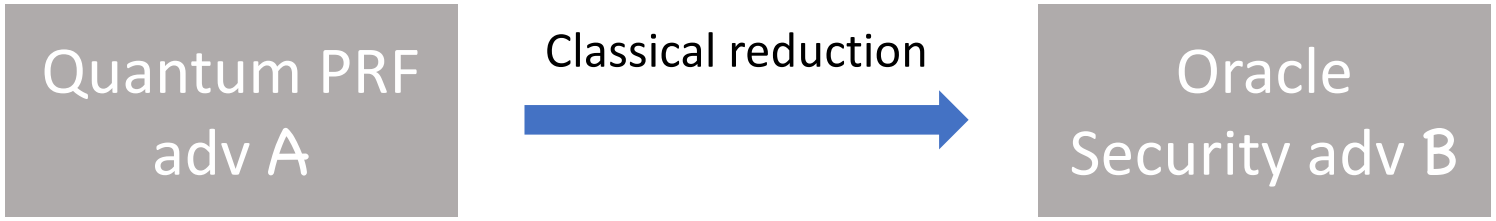
Step 1 makes sense if A classical, post-quantum, or fully quantum

Another View

Def: G is **Quantum Oracle Secure** if, $\forall$ QPT A, $\exists$ negligible ε such that

$$| \Pr[A^{|R\rangle} = 1] - \Pr[A^{|G \circ O\rangle} = 1] | < \varepsilon$$

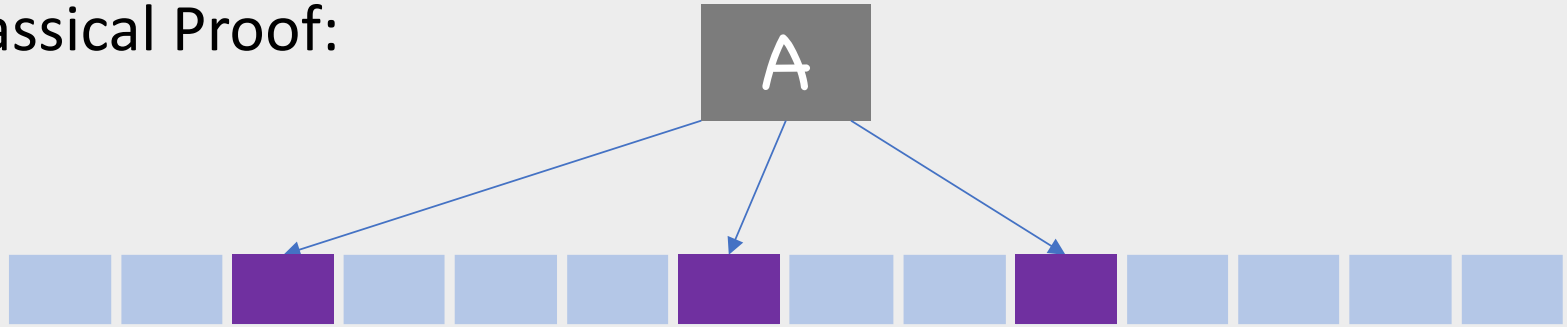
R,O random oracles



Another View

How to complete reduction from plain (post-quantum) PRGs?

Classical Proof:

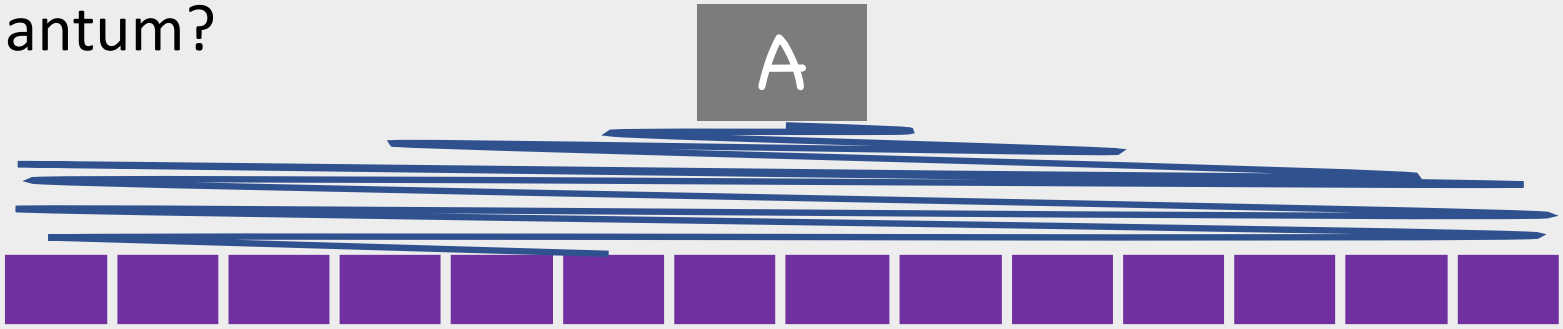


Only **q** queries → Can simulate with **q** samples
Hybrid over **q** values

Another View

How to complete reduction from plain (post-quantum) PRGs?

Quantum?



Need exponentially-many samples for perfect simulation

Reducing # of Hybrids

Goal: Simulate query responses using only poly-many samples

Simulating with Few Samples

Extreme 1: Same sample in all positions

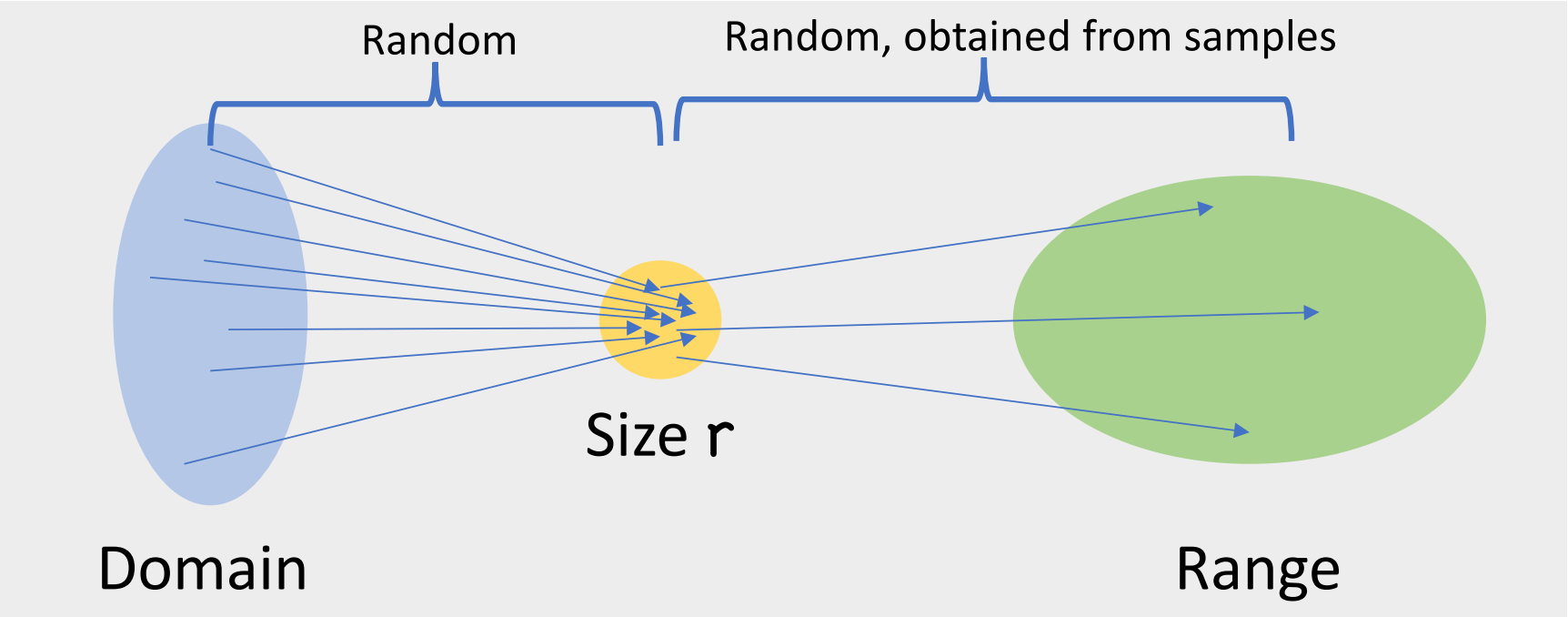
Distinguishable!

Middle ground: Several samples in random positions

Extreme 2: Independent sample in each position

Exponential loss!

Small Range Distributions



How big of r to be indistinguishable from truly random?

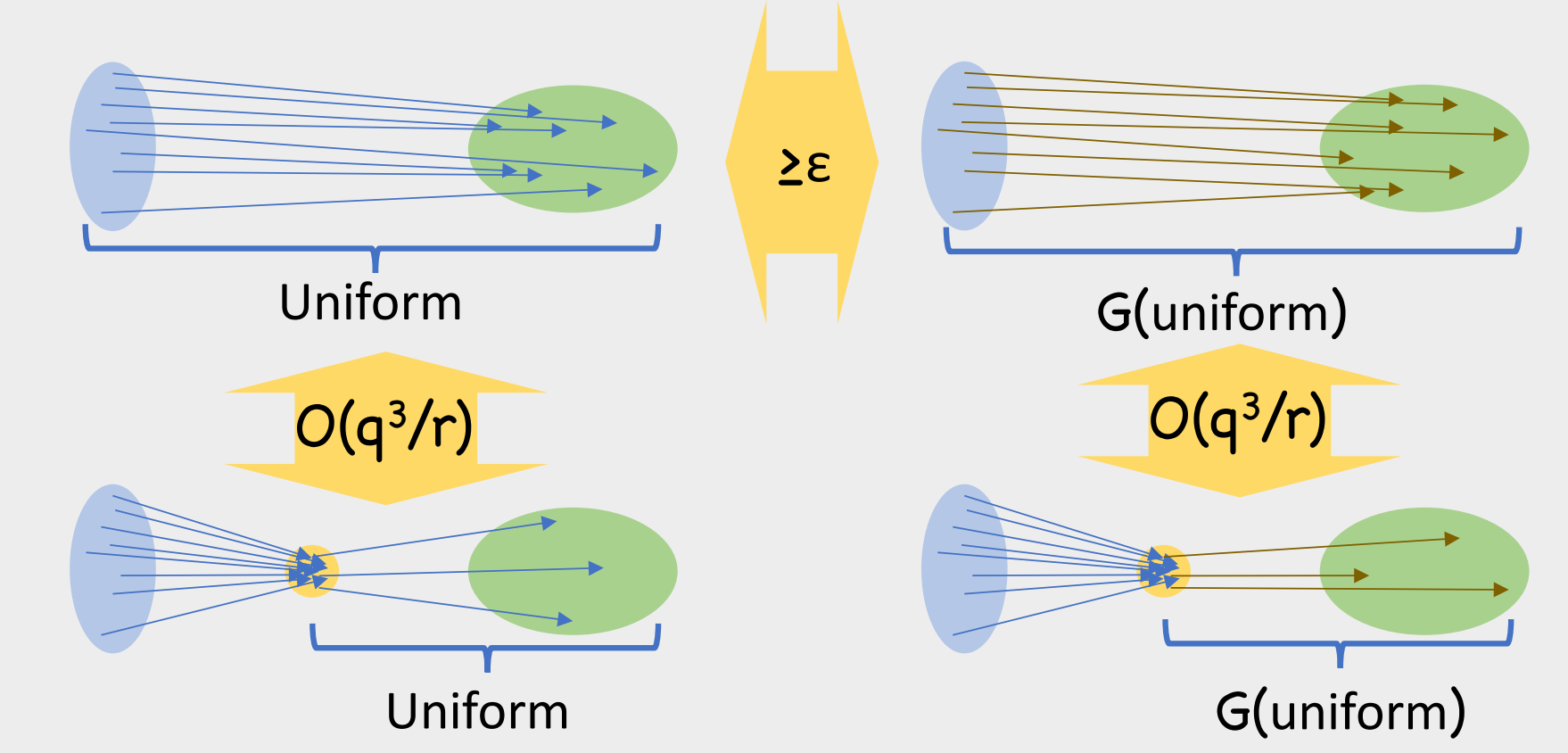
Small Range Distributions

Thm [Z'12b]: No q quantum query alg can distinguish SR_r from random, except with probability $O(q^3/r)$. Holds for any output distribution.

Quantum collision finding ➡ bound tight

$r=q^3$? $r=q^4$? $r=q^{20}$? $r=1.01^q$?

Quantum Proof



Quantum Proof

$$| \Pr[A^{|R\rangle} = 1] - \Pr[A^{|G \circ O\rangle} = 1] | \geq \varepsilon$$

$$| \Pr[B(y_1,\ldots,y_r) = 1] - \Pr[B(G(x_1),\ldots,G(x_r)) = 1] | \geq \varepsilon - O(q^3/r)$$

$$| \Pr[C(y) = 1] - \Pr[C(G(x)) = 1] | \geq \varepsilon/r - O(q^3/r^2)$$

Optimize by setting $r = O(q^3/\varepsilon)$ ➡ Final advantage $O(\varepsilon^2/q^3)$

Notes

Requires knowing ε

Can fix by guessing $\varepsilon=2^{-i}$ for random i

ε^2 means much bigger security loss

Proving SR Theorem

Thm [Z'12a]: If A makes q quantum queries to $O \leftarrow D$, then

$$\Pr[A^D()=1] = \sum_{\substack{x_1,\ldots,x_{2q} \\ y_1,\ldots,y_{2q}}} \Pr[D(x_i)=y_i \ \forall i \in [2q]]$$

(Restatement of polynomial method [Beals-Buhrman-Cleve-Mosca-de Wolf'01])

Thm [Z'12b]: For SR_r, the $\Pr[D(x_i)=y_i \ \forall i \in [k]]$ are degree k polynomials in $1/r$

$\Pr[A^{SR_r}()=1]$ = degree $2q$ polynomial in $1/r$

Proving SR Theorem

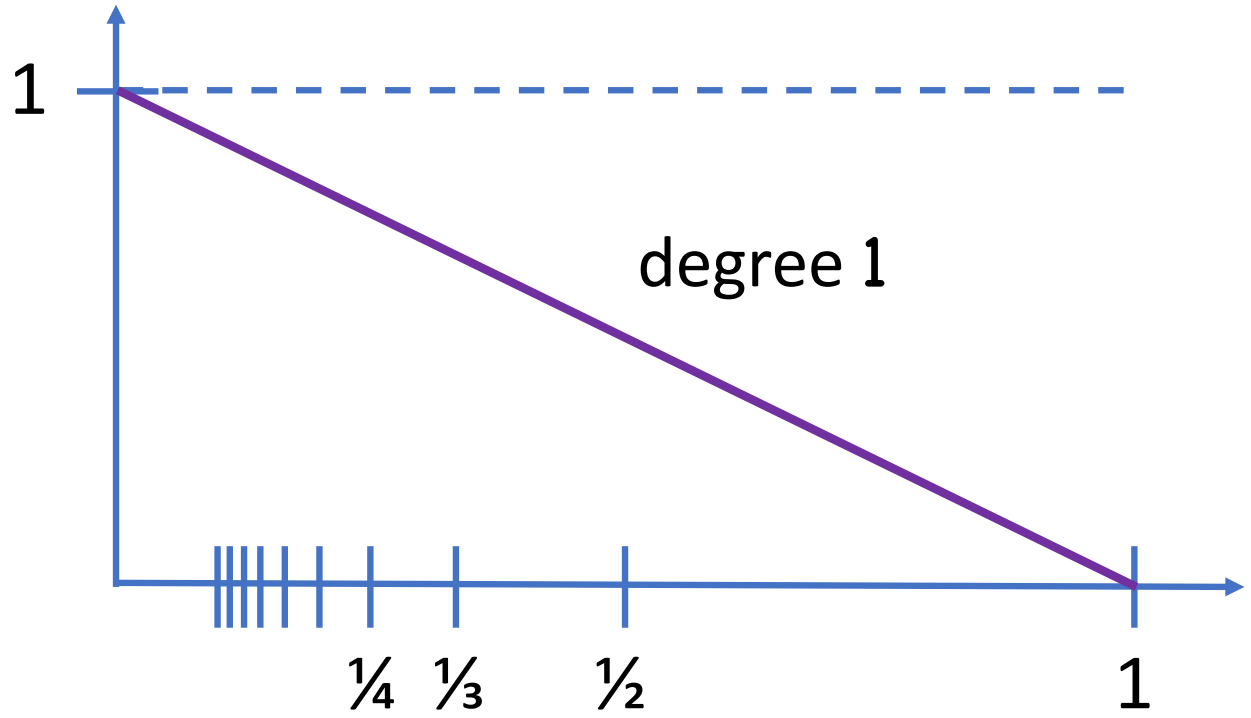
$\Pr[A^{SR_r}()=1] = P(1/r) =$ degree $2q$ polynomial

Additional observations:

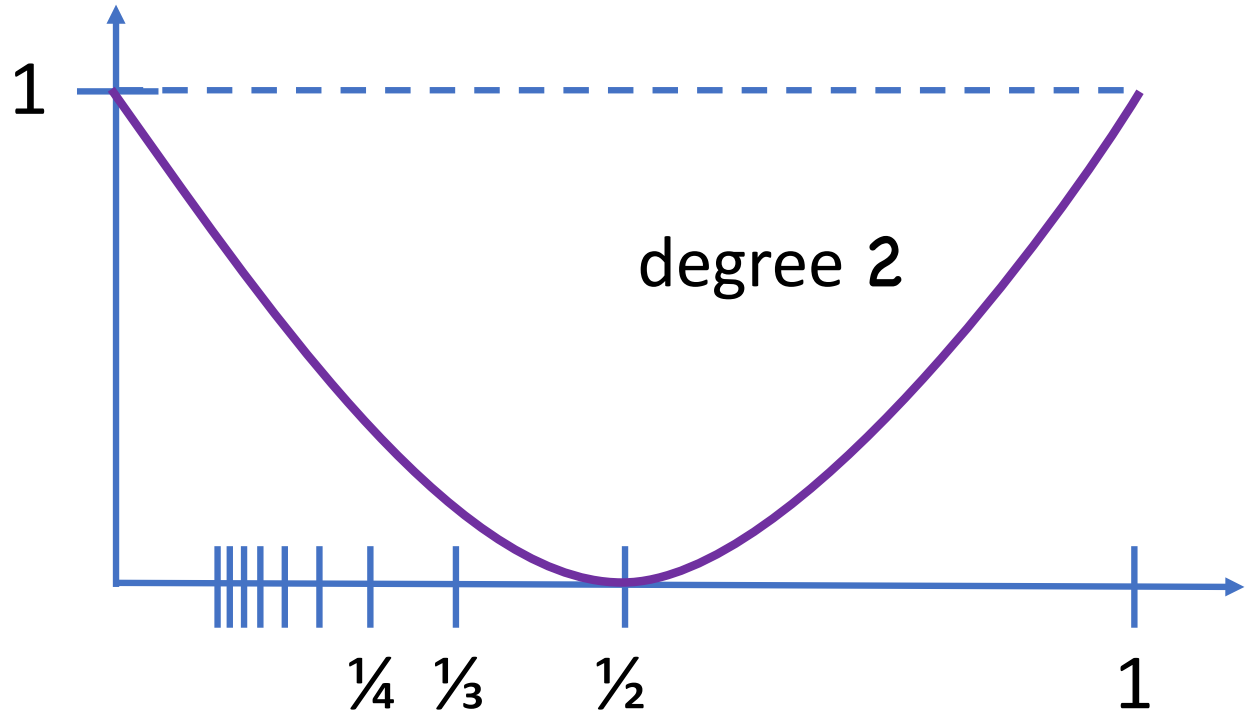
- SR_∞ = Truly random function
- $0 \leq P(1/r) \leq 1$ $\forall$ positive integers r

Goal: bound $|P(1/r) - P(0)|$

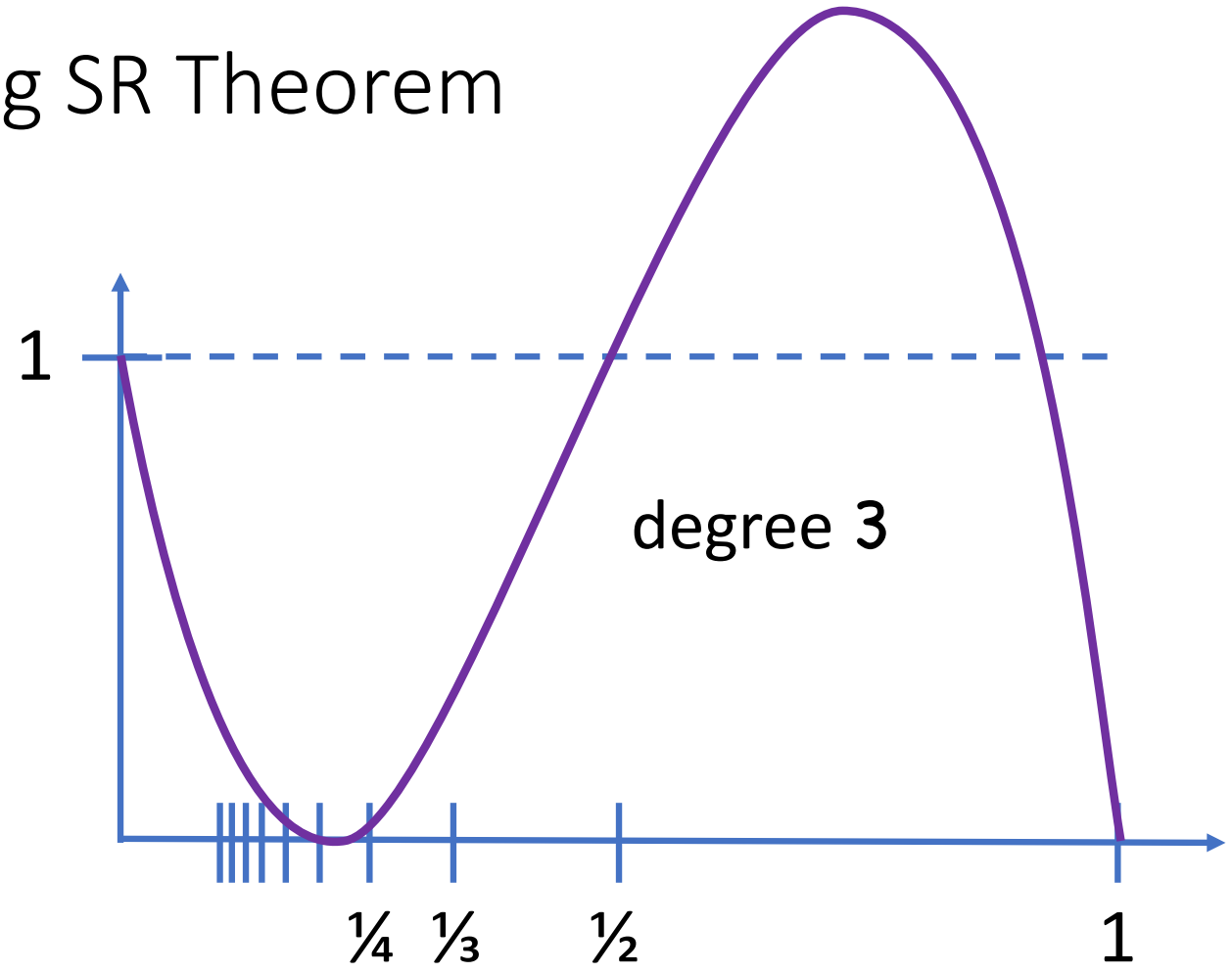
Proving SR Theorem



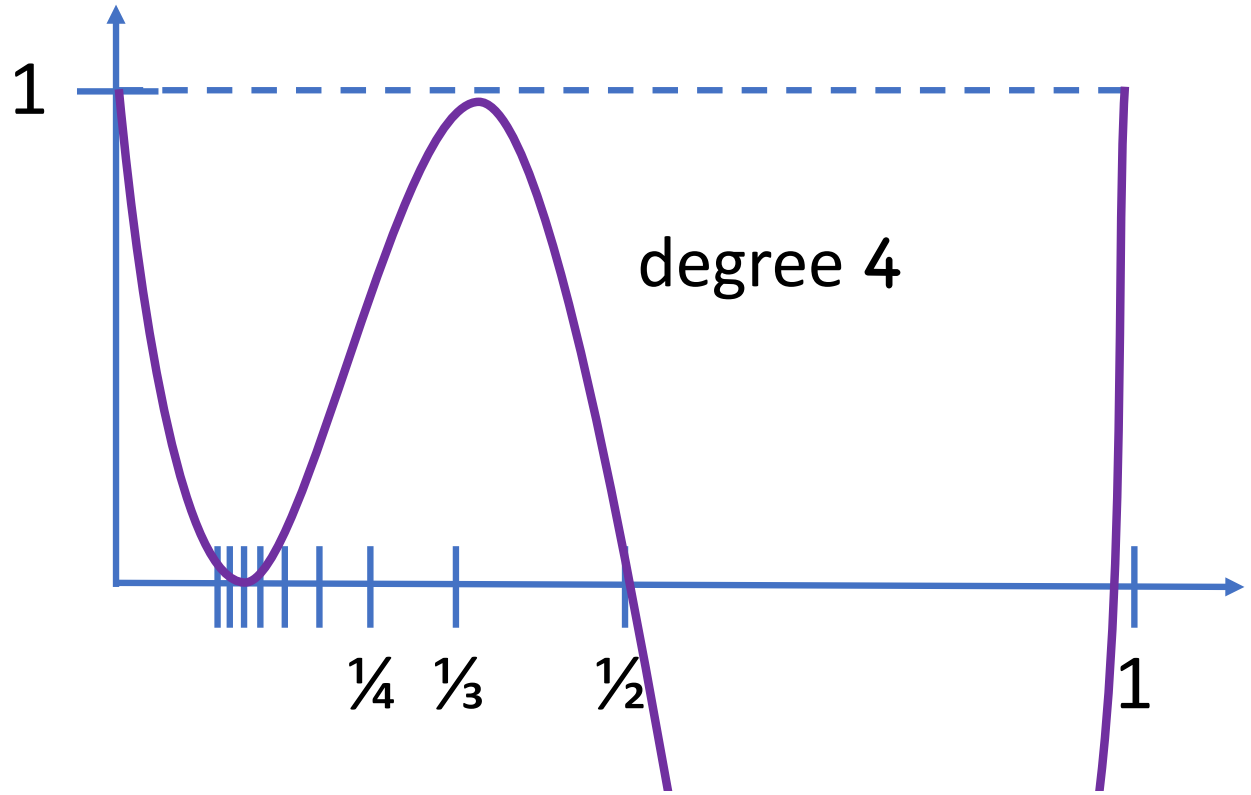
Proving SR Theorem



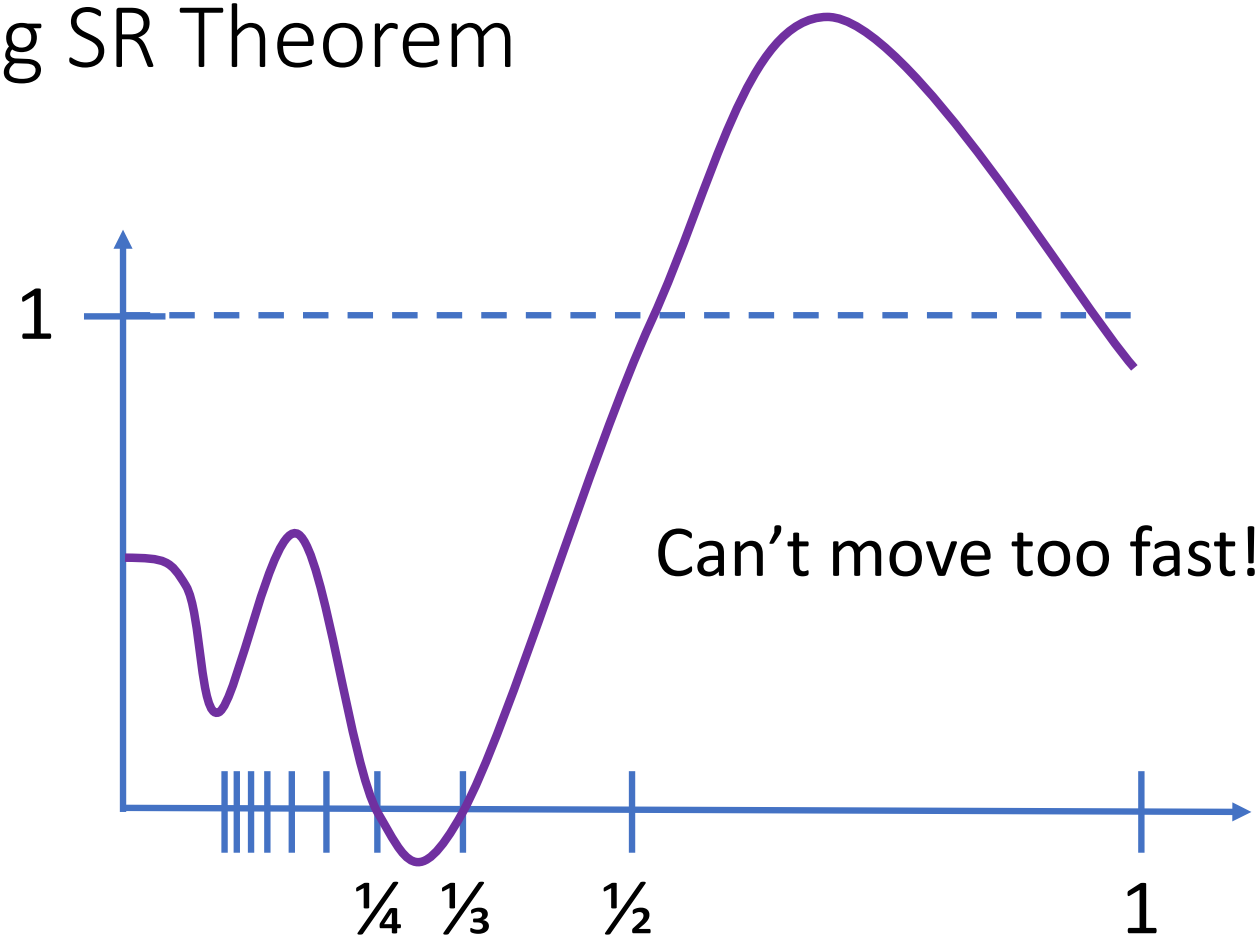
Proving SR Theorem



Proving SR Theorem



Proving SR Theorem



Proving SR Theorem

Thm [Z'12b]: If $P(1/r)$ satisfies:

- Degree $\leq k$
- $0 \leq p(1/r) \leq 1$ $\forall$ positive integers r

Then $|P(1/r) - P(0)| \leq 27k^3/r$

(Asymptotically tight)

Remaining Step

SR_r requires random functions; how to simulate?

Only **2q-wise** marginals matter
→ 2q-wise independent functions "look" random

What else is out there?

Secret sharing

Encryption

Authentication

IBE

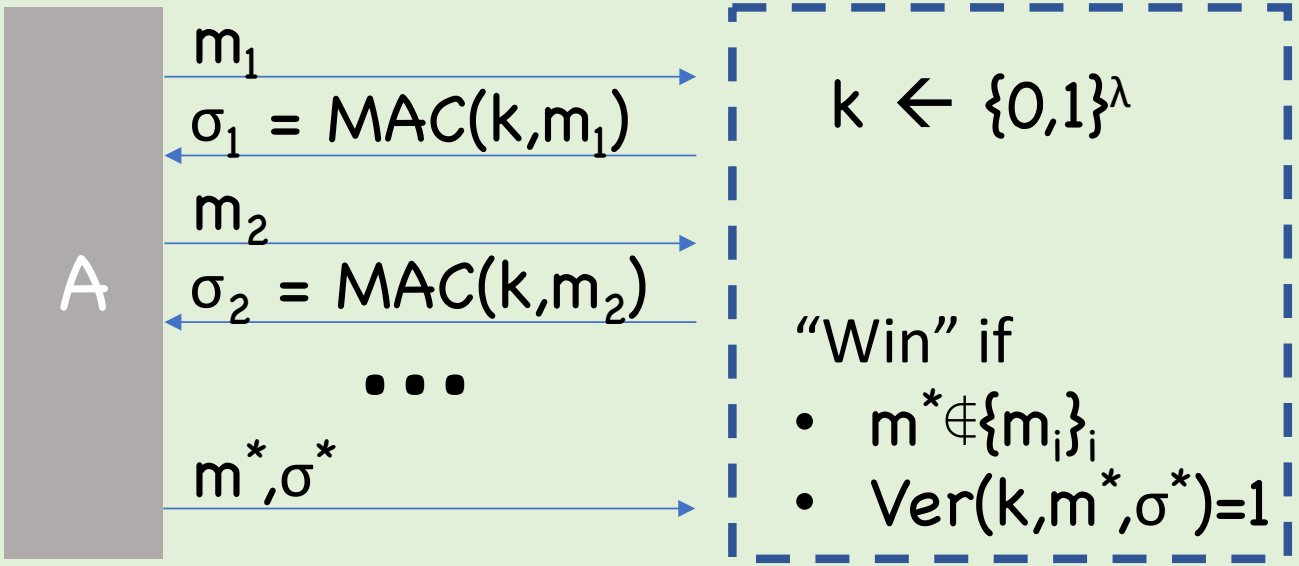
PRPs

MPC

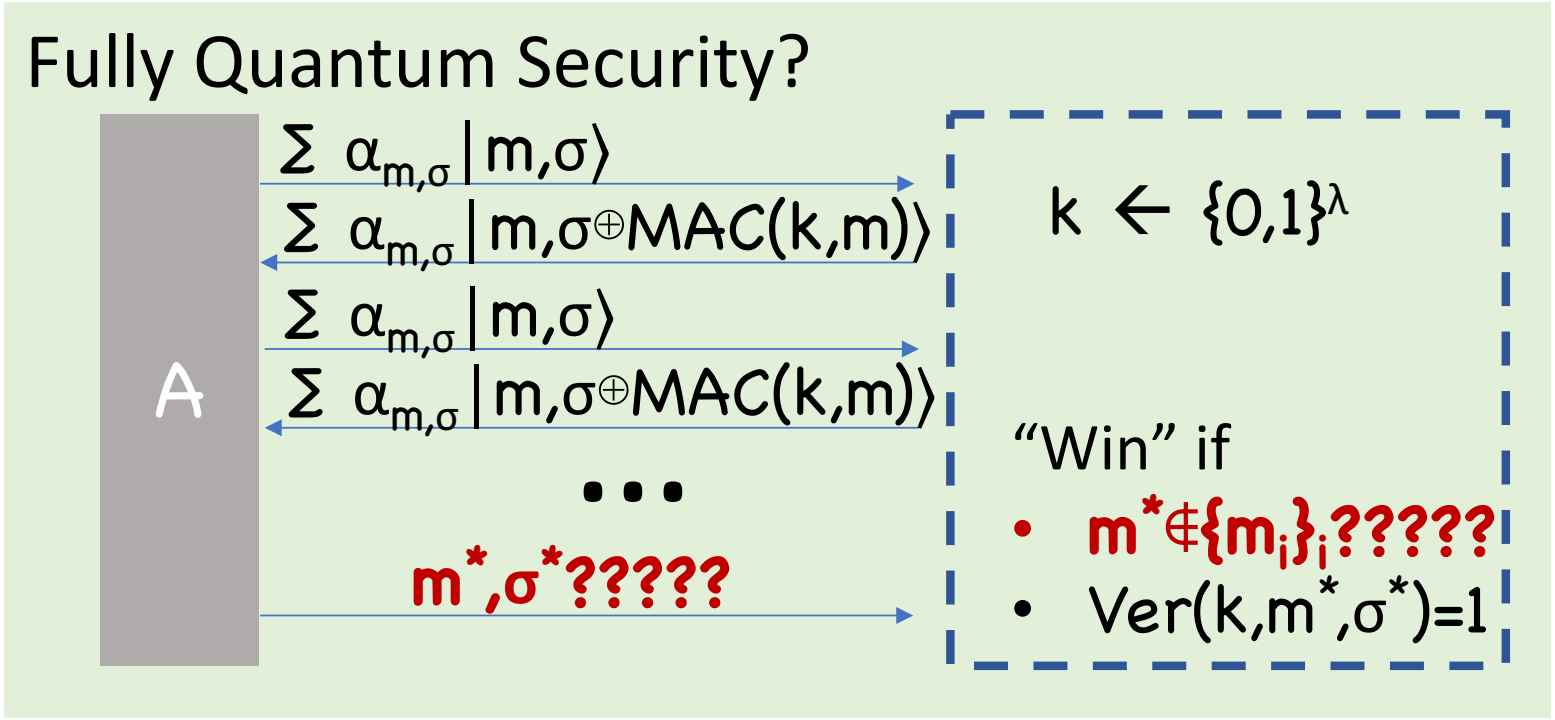
Remainder of lecture: definitional issues

Defining MACs/Signatures

Classical Security:



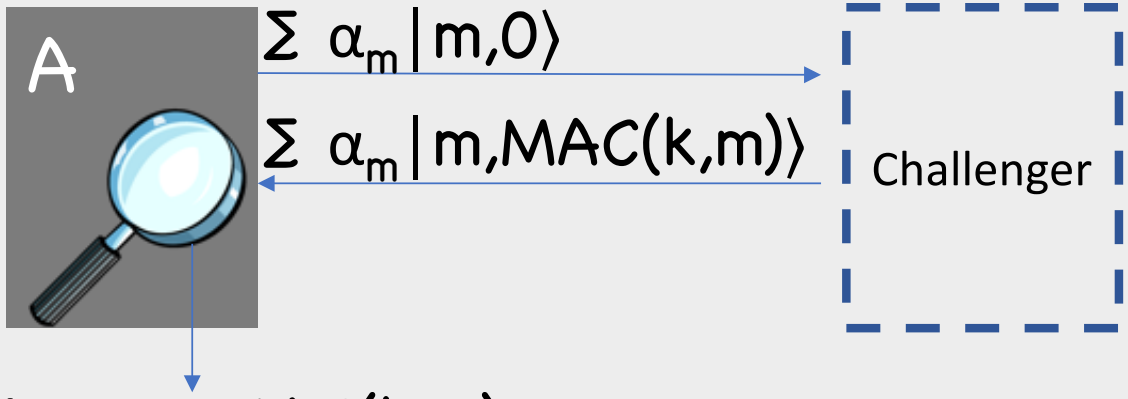
Defining MACs/Signatures



Defining MACs/Signatures

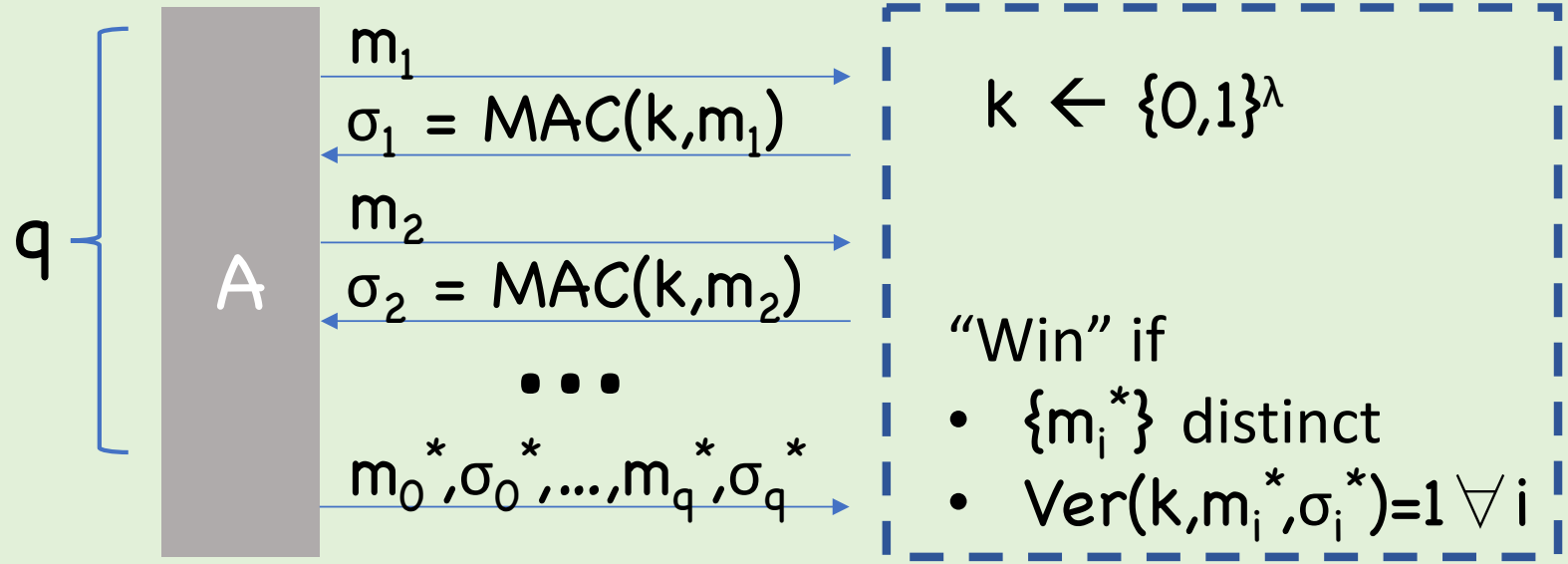
What does it mean to be “new”?

Example:



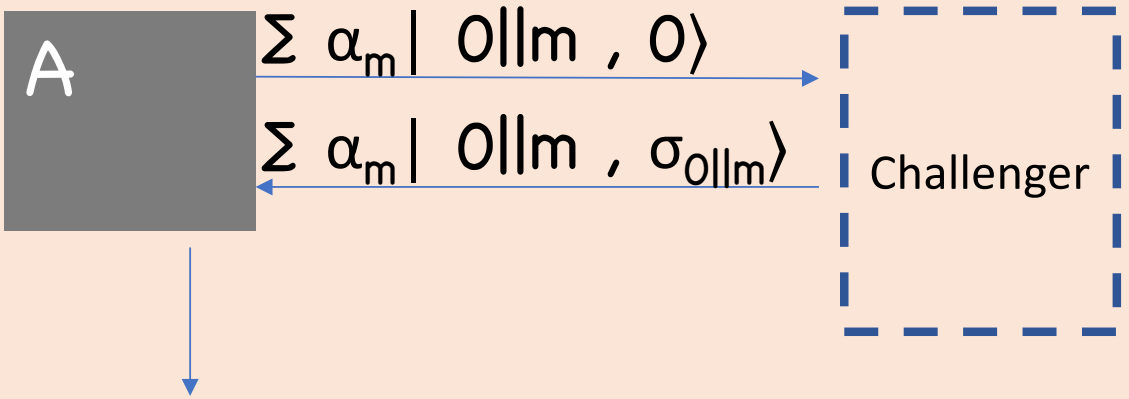
Defining MACs/Signatures

Partial Answer: One More Security [Boneh-Z'13a]



Defining MACs/Signatures

Limitation: Suppose:

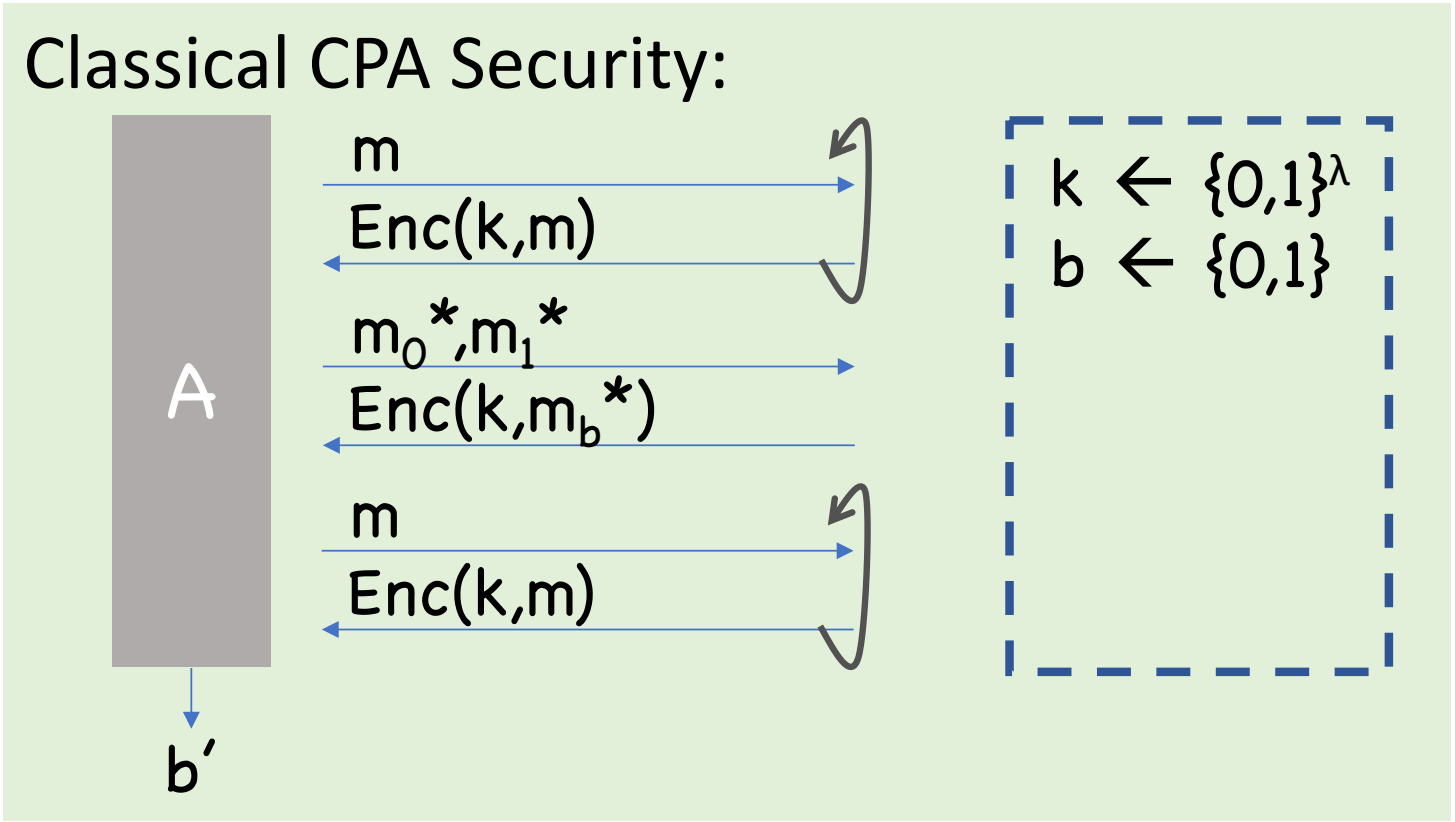


Doesn't violate one-more security!

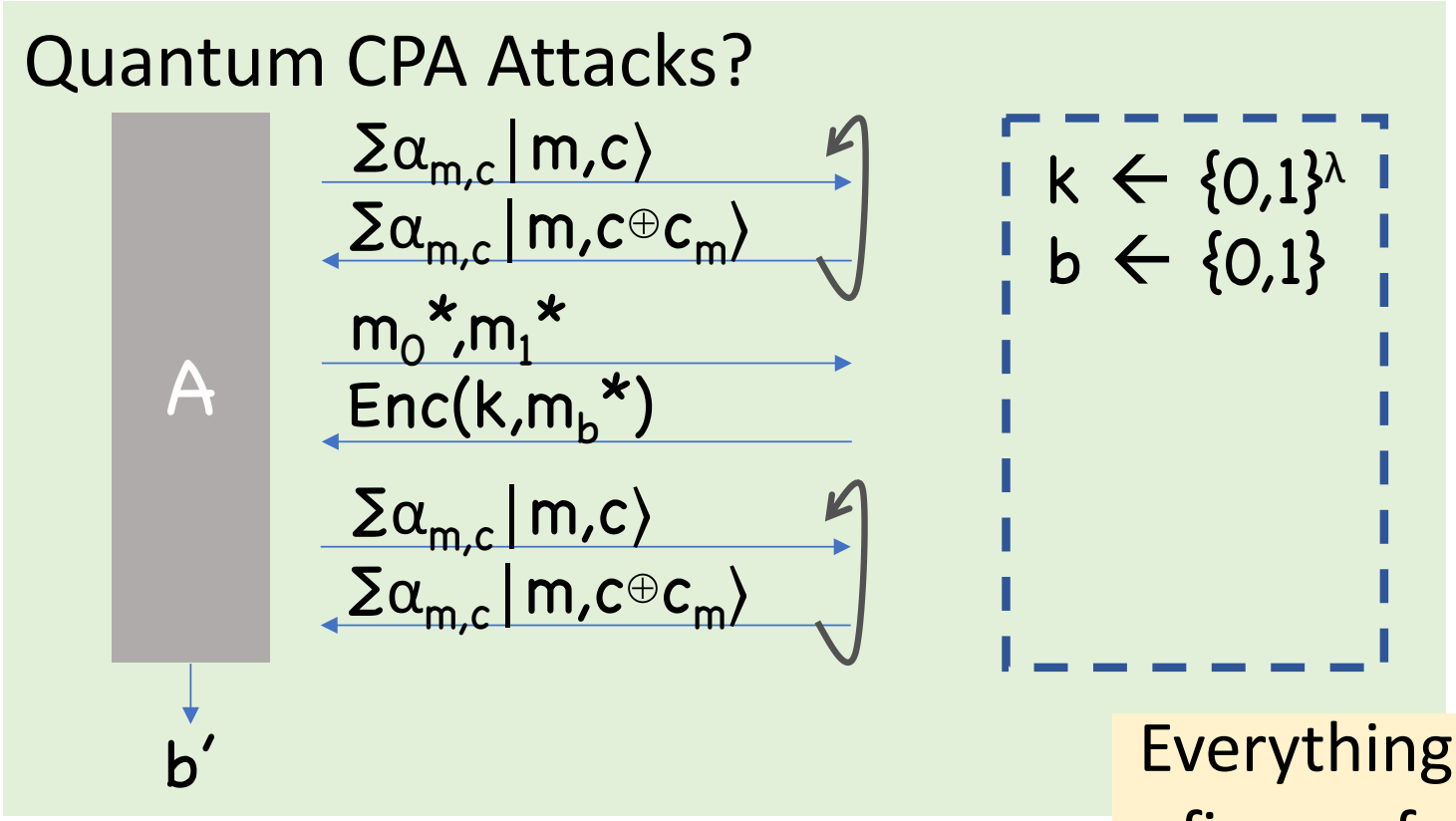
Defining MACs/Signatures

Other defs exist which fix this problem [Garg-Yuen-Z'17, Alagic-Majenz-Russell-Song'18], but IMO even satisfactory definition not yet solved

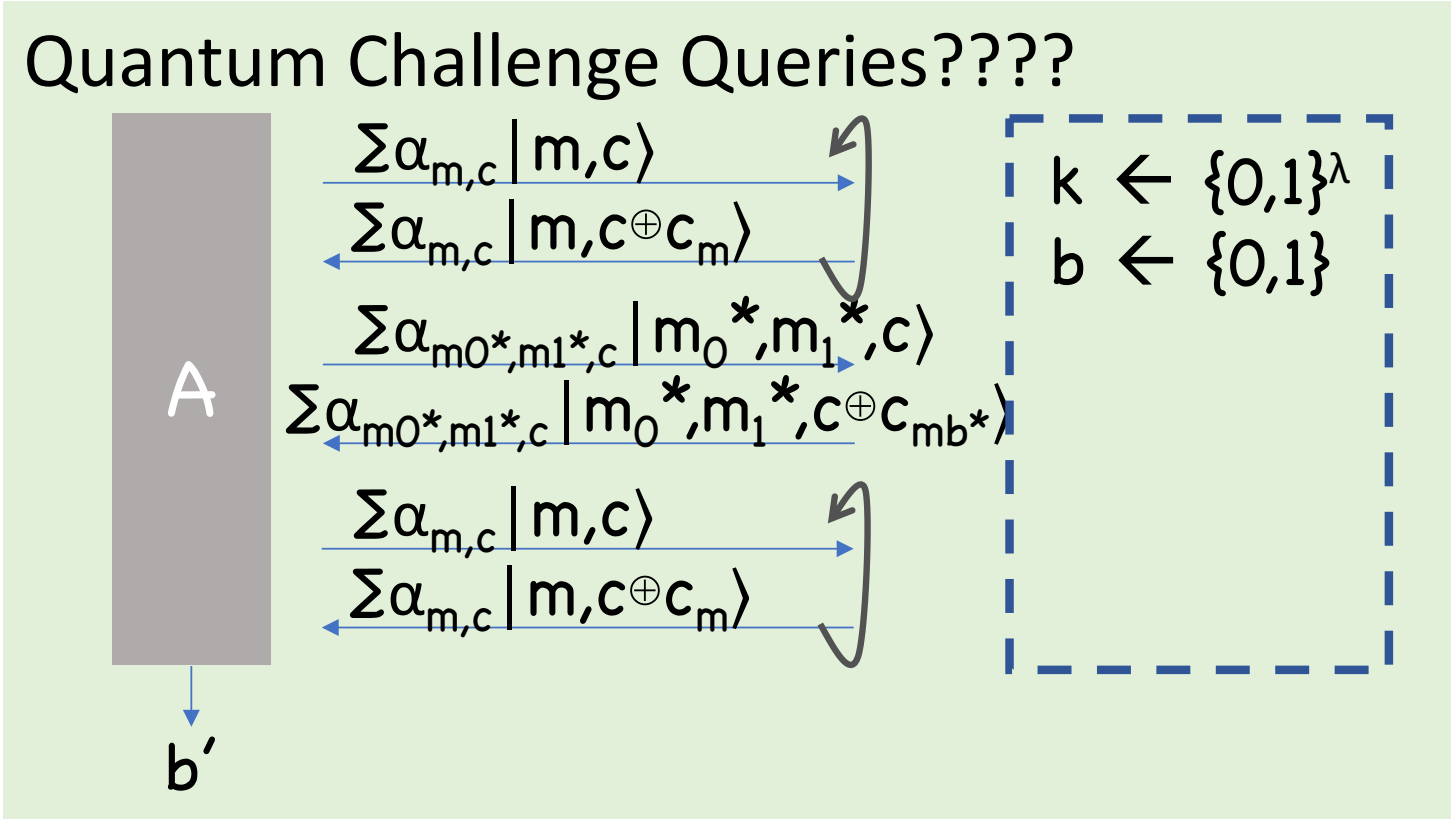
Defining Encryption



Defining Encryption

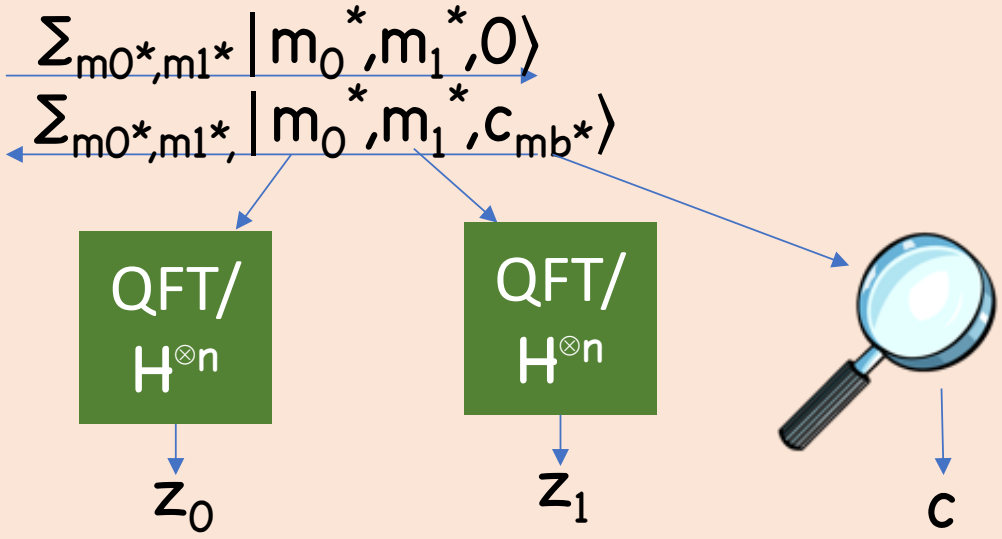


Defining Encryption



Defining Encryption

Attack:



$z_{1-b}=0^n$ and whp $z_b \neq 0^n$

Defining Encryption

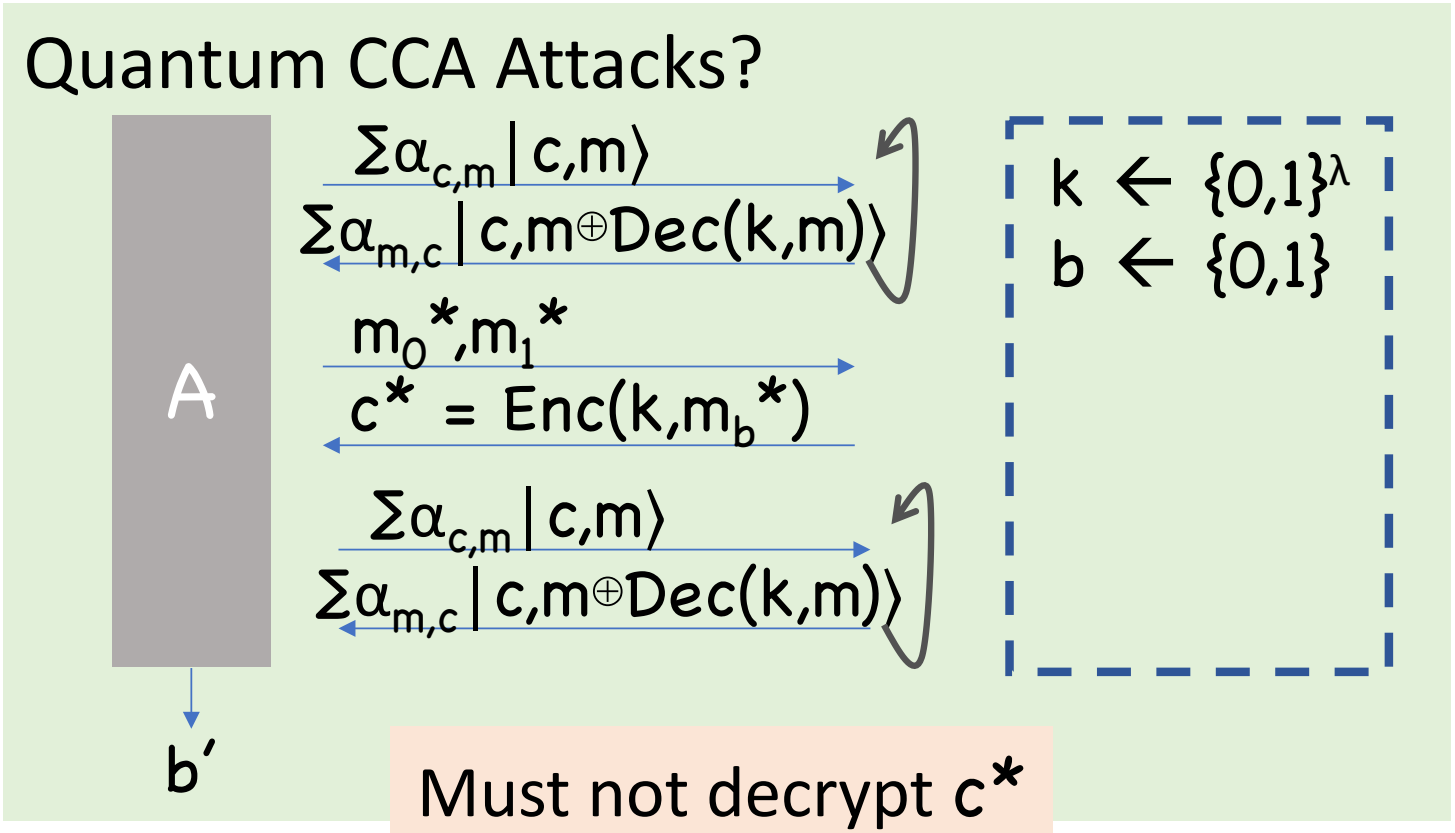
Classical encryption schemes are not secure for encrypting quantum messages, *if the attacker gets to see the original message registers*

[Boneh-Z'13b]: don't allow quantum challenge queries

[Gagliardoni-Hülsing-Schaffner'16]: make sure quantum challenge query never returned

More subtle than it sounds

Defining Encryption

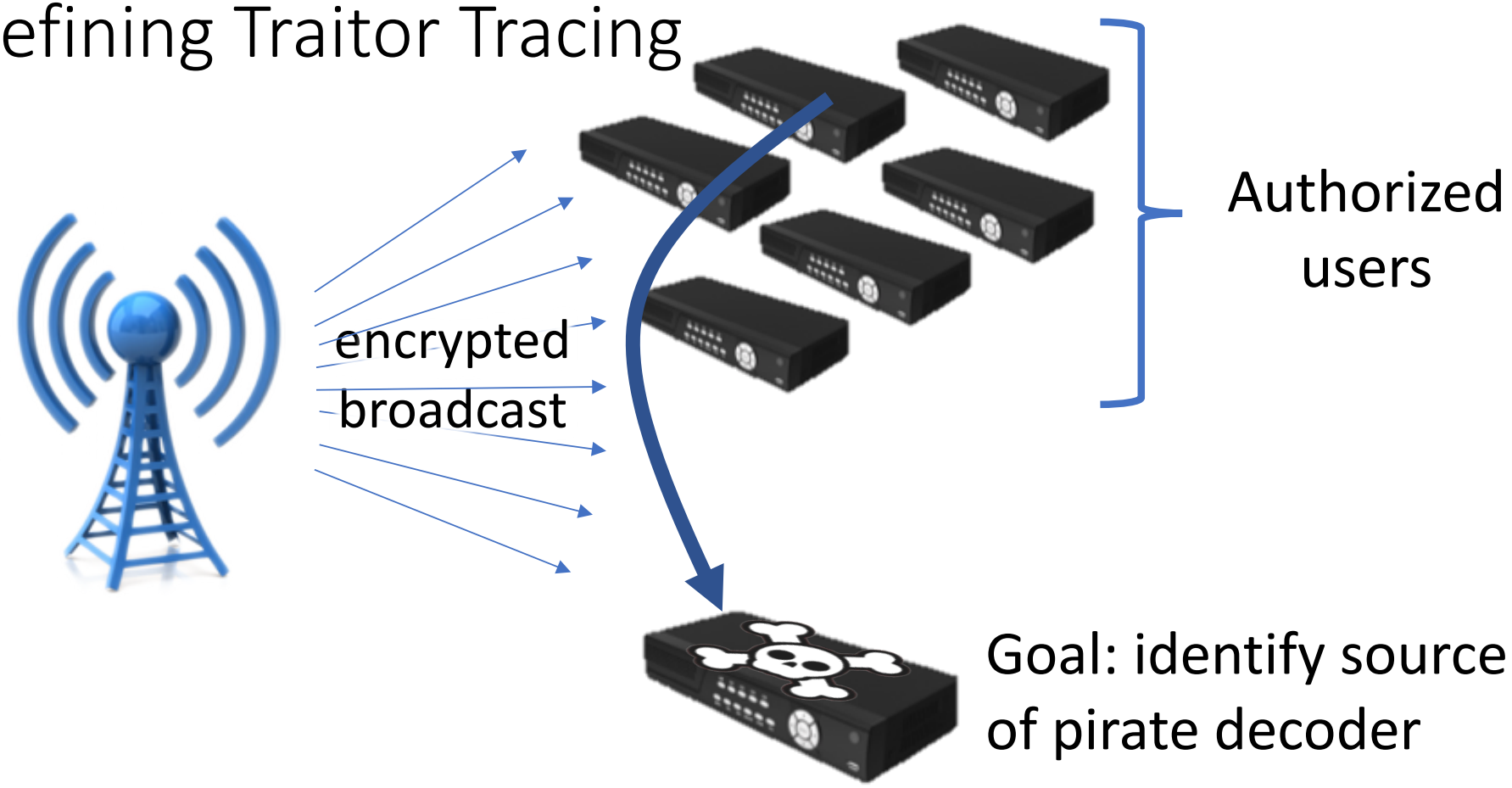


Defining Encryption

"Not decrypting c^*" problematic for quantum challenges

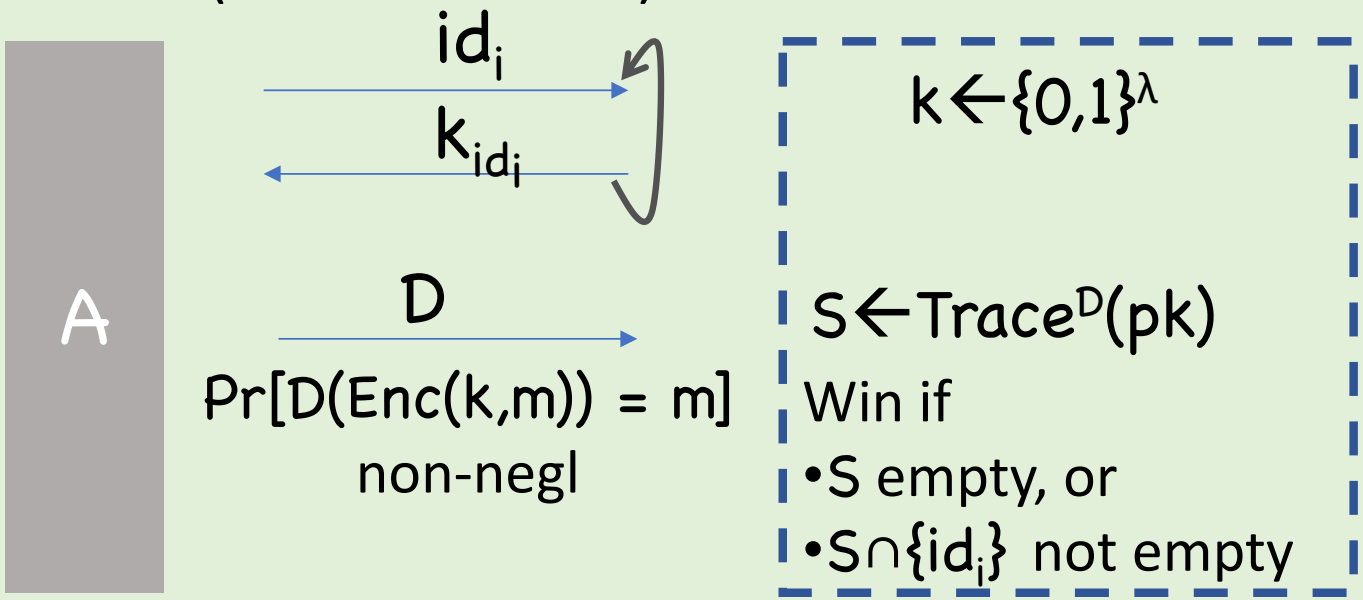
[Chevalier-Ebrahimi-Vu'20]: Formalize quantum CCA+Challenge

Defining Traitor Tracing



Defining Traitor Tracing

Classical Def (modulo details):



Defining Traitor Tracing

Problem: most prior work assumes D is stateless/can be rewound

Somewhat inherent: single query to D usually not enough to accuse

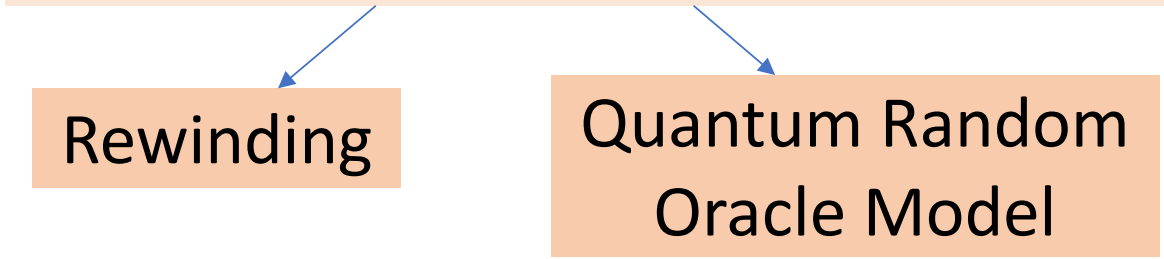
But if decoder has quantum state, single query may alter decoder

[Z'20]: Formalize quantum analog of "stateless"

Tomorrow: Unavoidable Quantum Attacks

So far, issues concern new quantum attack models

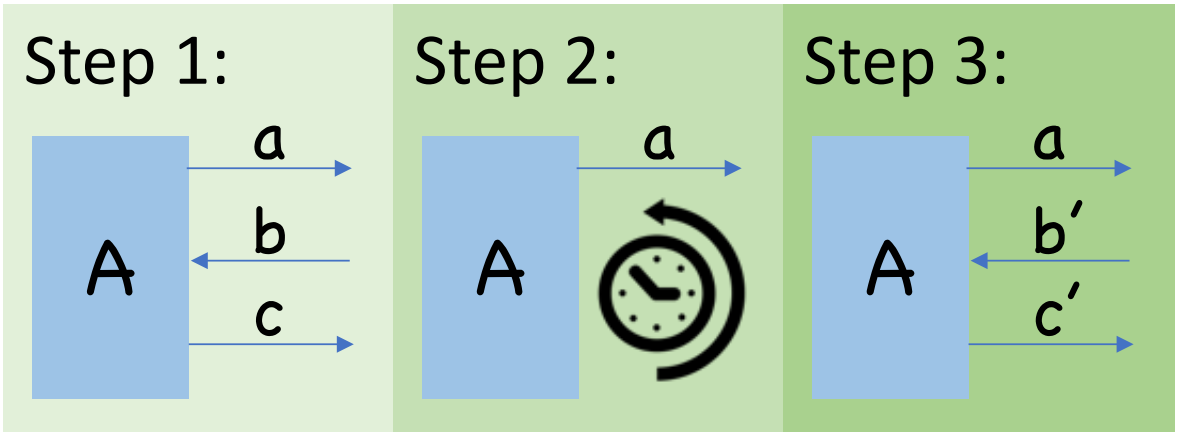
My remaining lectures: attacks/issues even under existing attack model



Quantum Rewinding

Mark Zhandry (Princeton & NTT Research)

Classical Rewinding

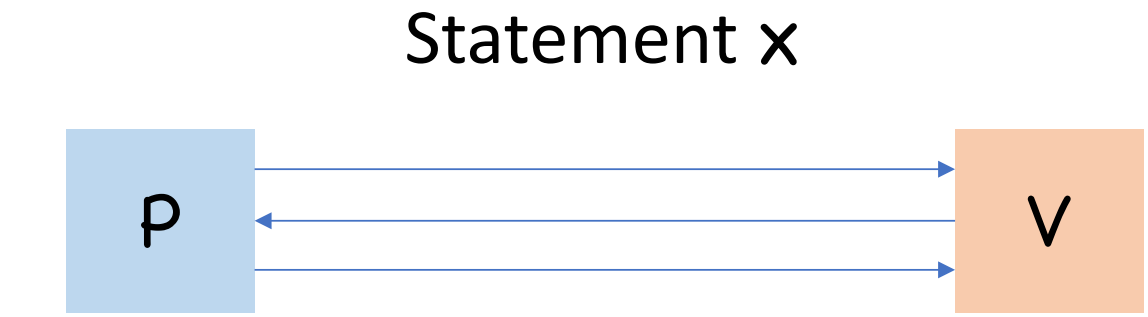


Zero knowledge

Commitments

Proofs of knowledge

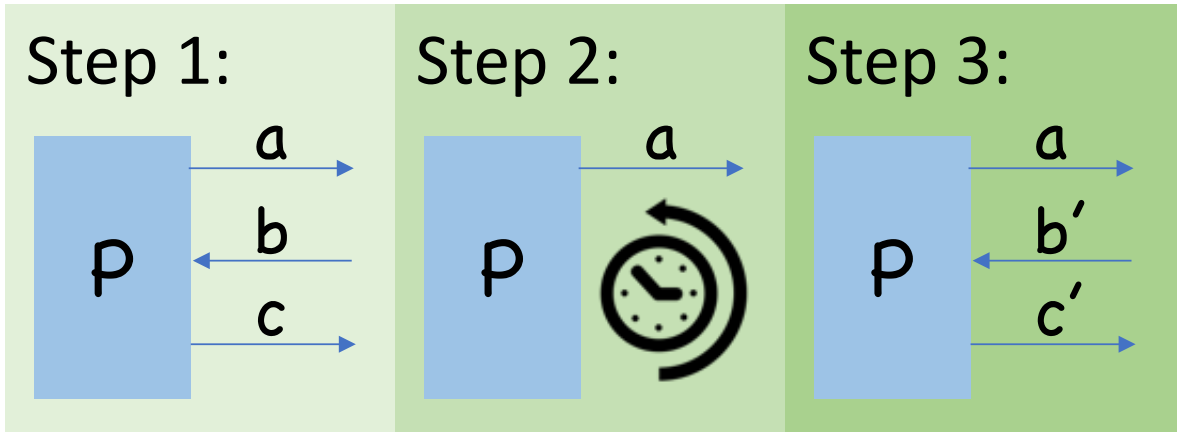
Proof of Knowledge (PoK)



If accepts, V should be convinced not only of x, but also that P "knows" witness

Usually combine with over properties like zero knowledge

Rewinding for PoK



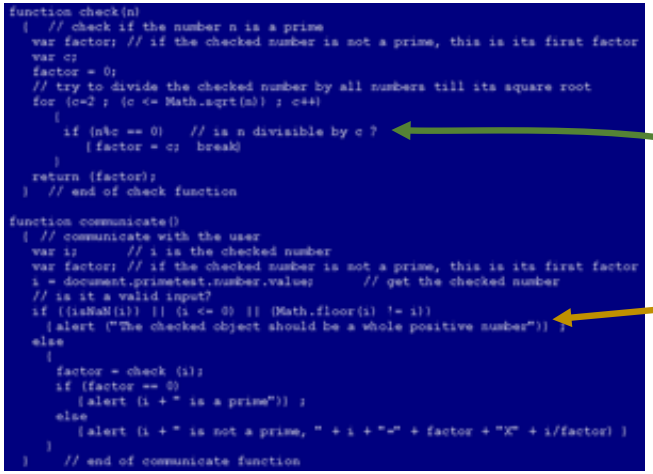
$(a,b,c,b',c'),\ b \neq b'$

$\downarrow$

w

"special soundness"

What Does Rewinding *Really* Mean



```
function check(n)
  {  // check if the number n is a prime
   var factor; // if the checked number is not a prime, this is its first factor
   var c;
   factor = 0;
   // try to divide the checked number by all numbers till its square root
   for (c=2 ; (c <= Math.sqrt(n)) ; c++)
     {
      if (n%c == 0)    // is n divisible by c ?
        {factor = c;  break}
     }
   return (factor);
  }  // end of check function

function communicate()
  {  // communicate with the user
   var i;       // i is the checked number
   var factor; // if the checked number is not a prime, this is its first factor
   i = document.primetest.number.value;       // get the checked number
   // is it a valid input?
   if ((isNaN(i)) || (i <= 0) || (Math.floor(i) != i))
     {alert ("The checked object should be a whole positive number")}
   else
     {
      factor = check (i);
      if (factor == 0)
        {alert (i + " is a prime")} ;
      else
        {alert (i + " is not a prime, " + i + "=" + factor + "X" + i/factor) }
     }
  }    // end of communicate function
```

Given state here,

can we remember state here?

Classical programs not necessarily “reversible”

But can be *made* reversible by recording program trace

What Does Rewinding *Really* Mean

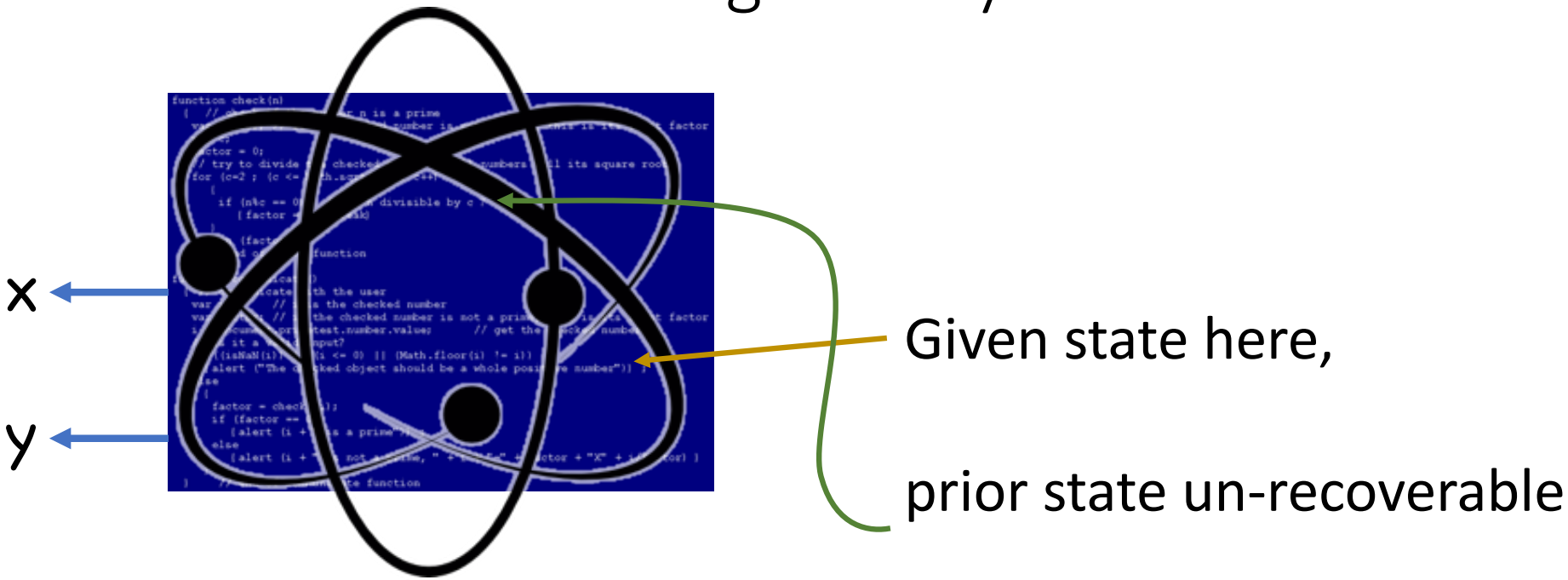
But isn't quantum computing alrady reversible?

Only until a measurement…

Uncertainty Principle: once measurement is performed, quantum state irreversibly altered

No Cloning: can't record program trace for later

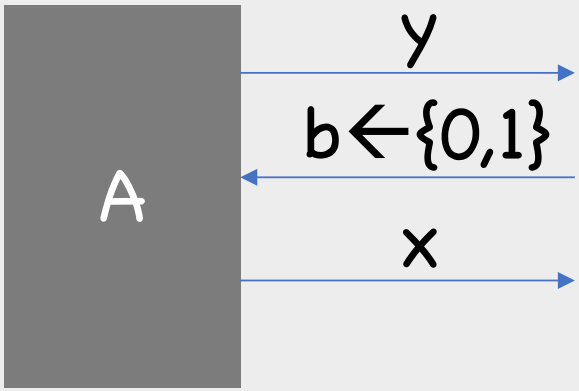
What Does Rewinding *Really* Mean



Interactive quantum programs *cannot* in general be made reversible

Impossibility of Quantum Rewinding
[Ambainis-Rosmanis-Unruh'14]

Coin flipping/commitment game



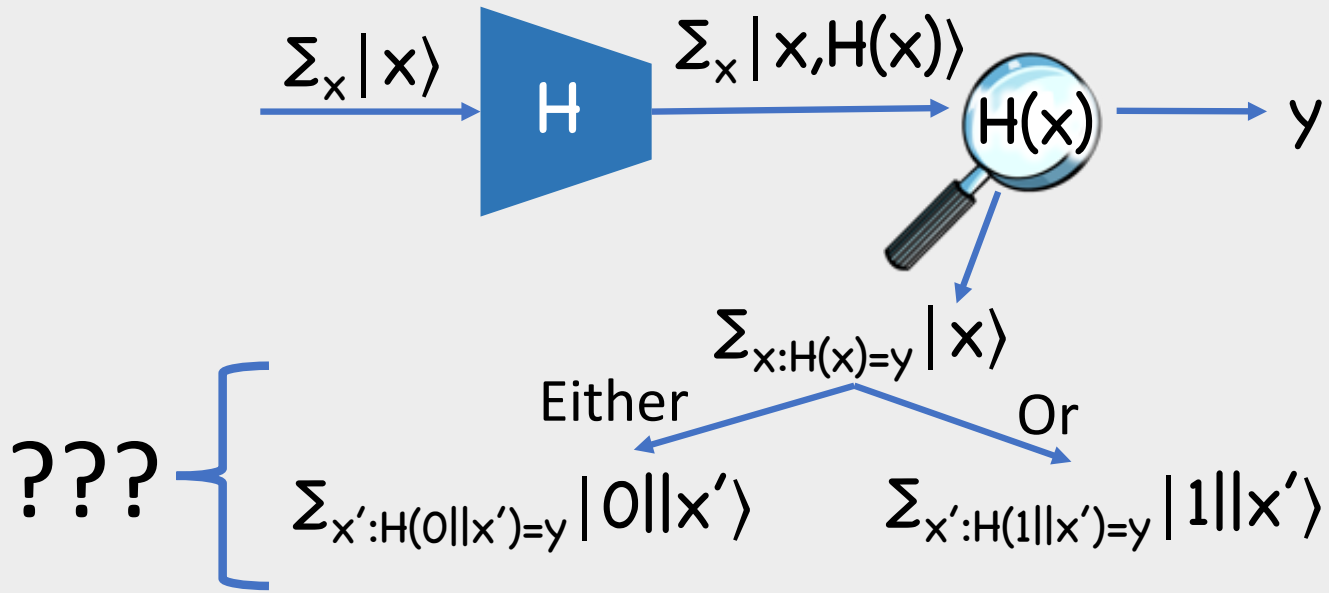
Win if
- $H(x)=y$
- $x_1 = b$

Classically:
$Pr[A \text{ wins}] \geq \frac{1}{2}+\varepsilon$
+ Rewinding
$= Pr[\text{collision}] \geq \text{poly}(\varepsilon)$

Goal: devise *quantum* A and col. res. H where $Pr[A \text{ wins}] \approx 1$

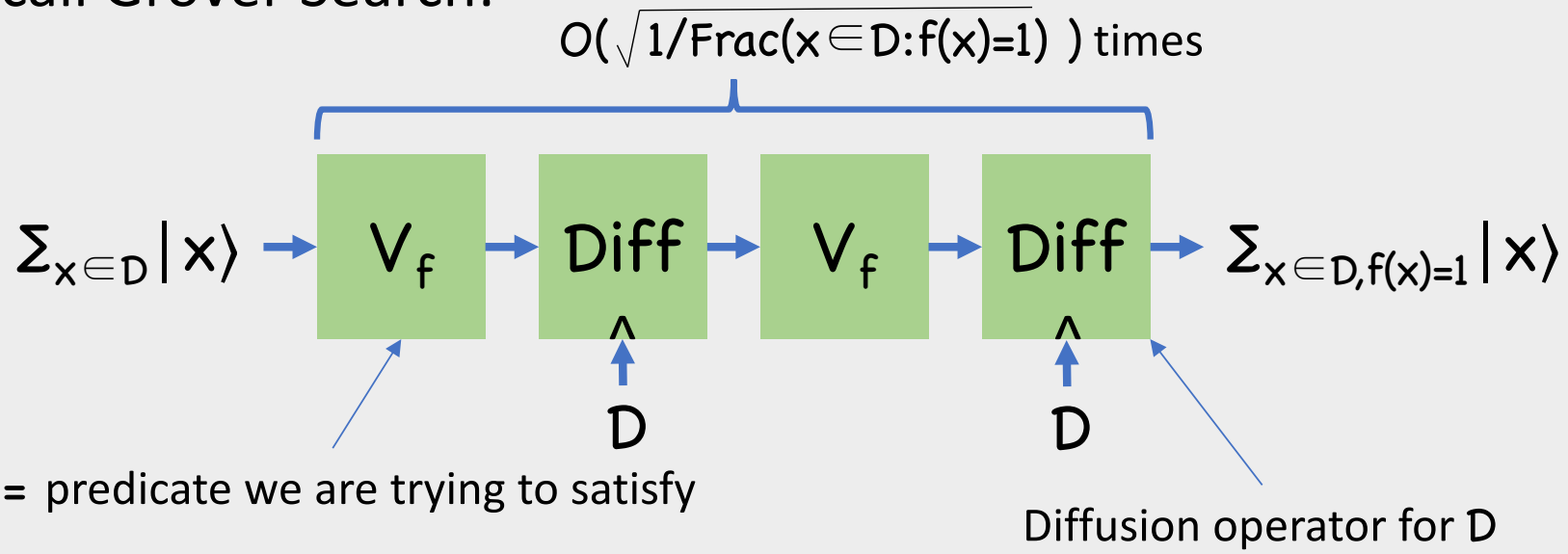
Impossibility of Quantum Rewinding
[Ambainis-Rosmanis-Unruh'14]

Idea:



Impossibility of Quantum Rewinding
[Ambainis-Rosmanis-Unruh'14]

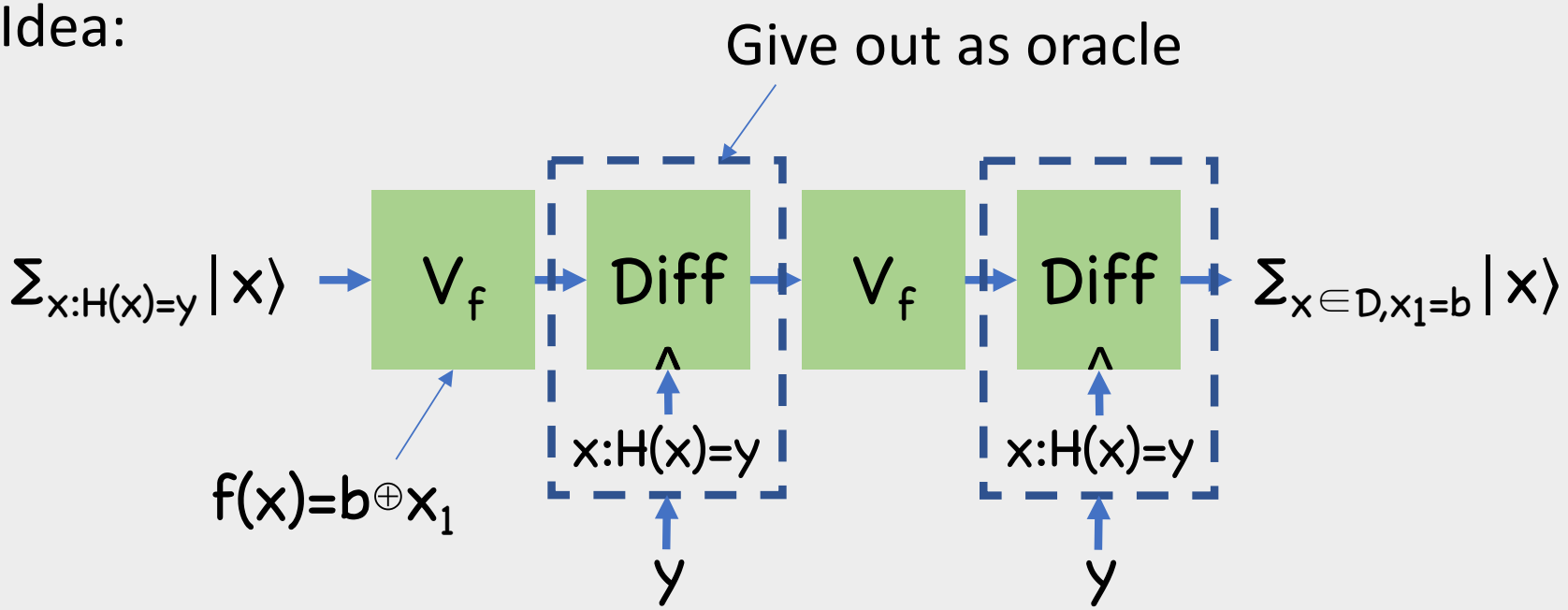
Recall Grover Search:



Impossibility of Quantum Rewinding

[Ambainis-Rosmanis-Unruh'14]

Idea:



Impossibility of Quantum Rewinding
[Ambainis-Rosmanis-Unruh'14]

Thm: A random function H (given as oracle) is collision resistant, even if additionally given Diff oracle

H is not a good commitment, despite being collision resistant

PoK cannot quantumly be justified based on special soundness alone

Ingredient 1: Rewinding Lemma

Lemma [Unruh'10]:

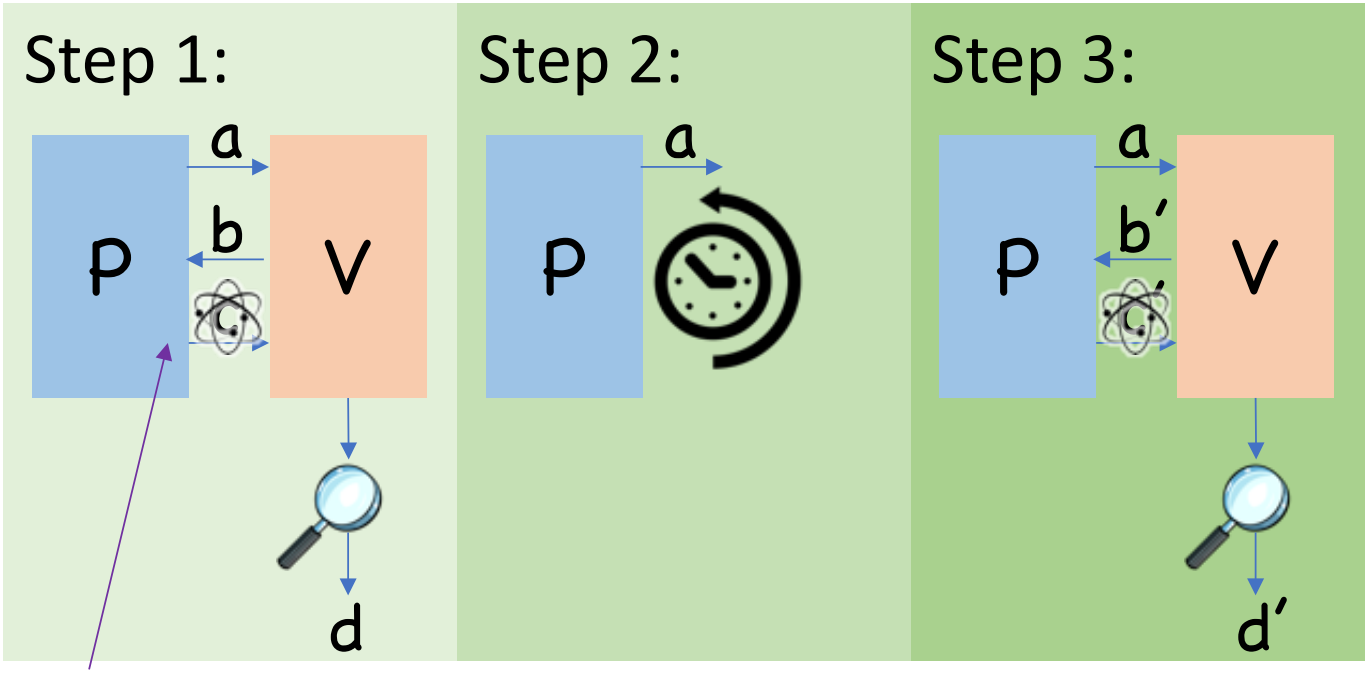
Suppose: (1) c is a single bit
(2) Defer all measurements except c
(3) $\Pr[c=1 \mid a]=\varepsilon$

Then: $\Pr[c=c'=1 \mid a] \geq \varepsilon^3$

Compare to ε^2 classically

Really need $\Pr[c=c'=1 \mid (b \neq b'), a]$, Unruh gives better bound

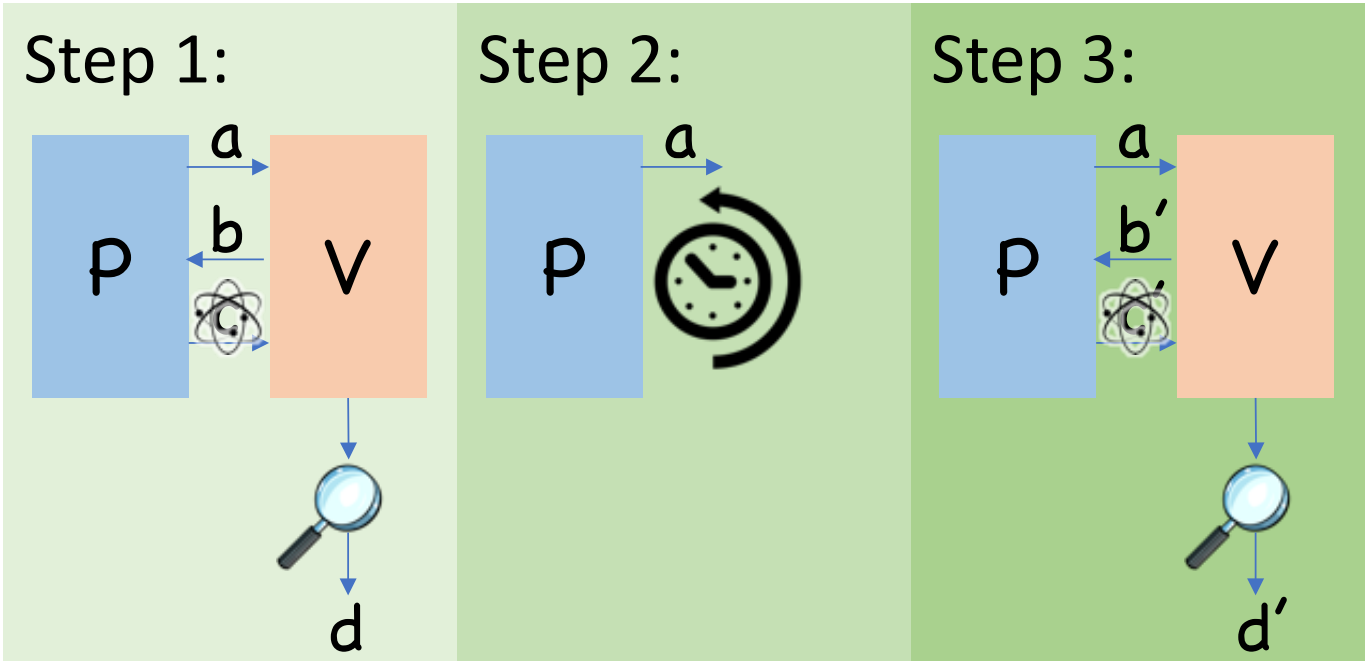
Applying Rewinding Lemma



No measurement after b!

Rewinding Lemma: $Pr[d=d'=1] \geq \varepsilon^3$

Applying Rewinding Lemma



Problem: Can't extract c, c' without changing d, d'

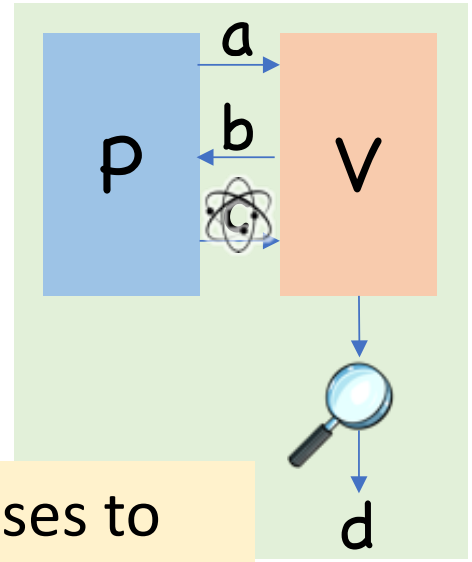
Ingredient 2: Additional Security Promises

Option 1: **Injective H** ➡ Unique "opening" x, can measure without any collapse

Ingredient 2: Additional Security Promises

Option 1 [Unruh'10]: **Strict Soundness:**

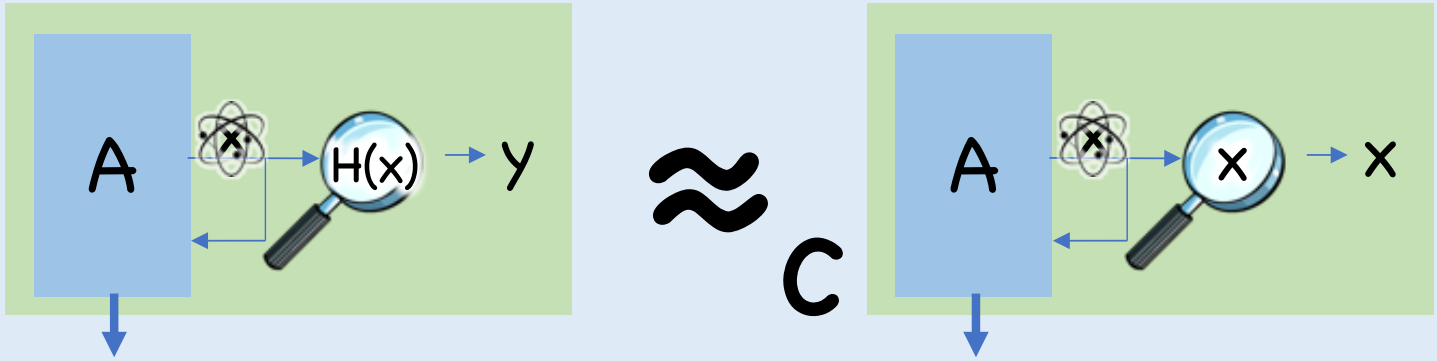
$\forall$ a,b, $\exists$ unique c s.t. V(a,b,c)=1



If d=1, c collapses to classical value anyway

Ingredient 2: Additional Security Promises

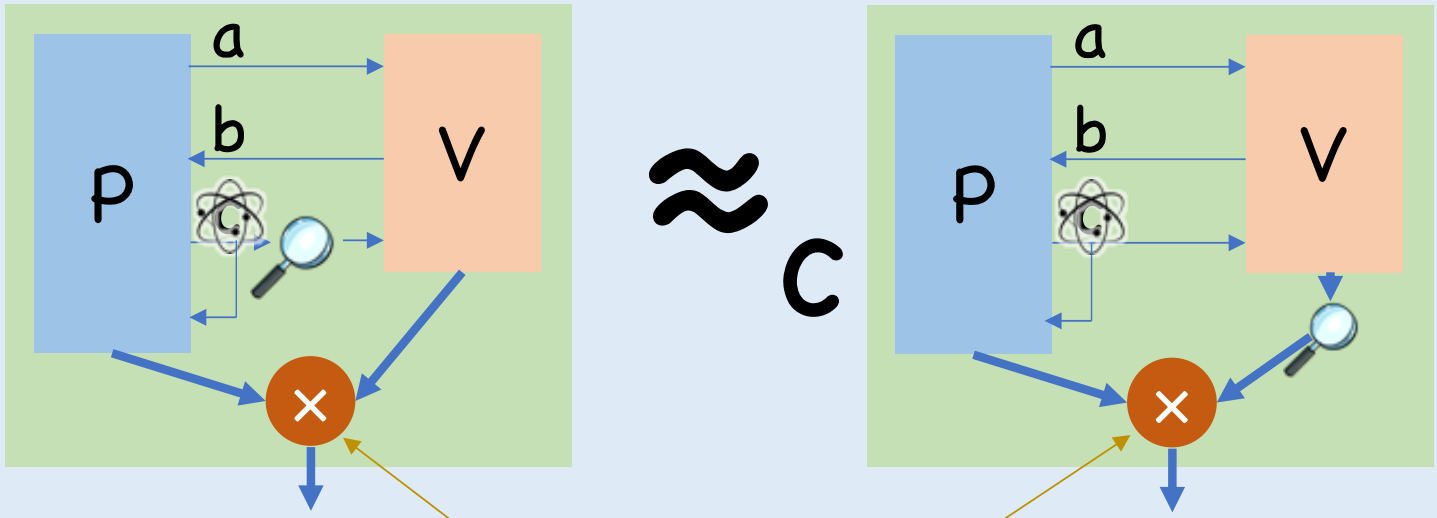
Option 2 [Unruh'16]: **Collapsing Hashes:**



Ingredient 2: Additional Security Promises

Option 2 [Liu-Z'19,Don-Fehr-Majenz-Schaffner'19]:

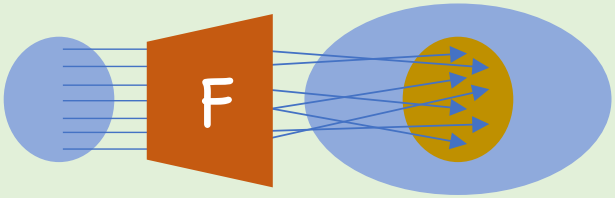
Collapsing:



Justify Collapsing: Lossy Functions
[Unruh'16]

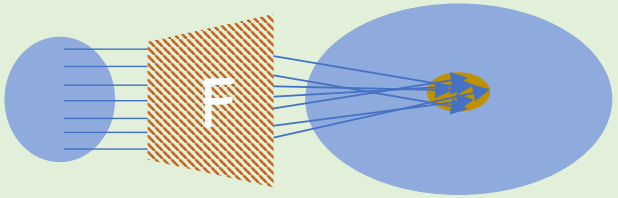
Lossy functions:

Injective Mode:



$\approx_c$

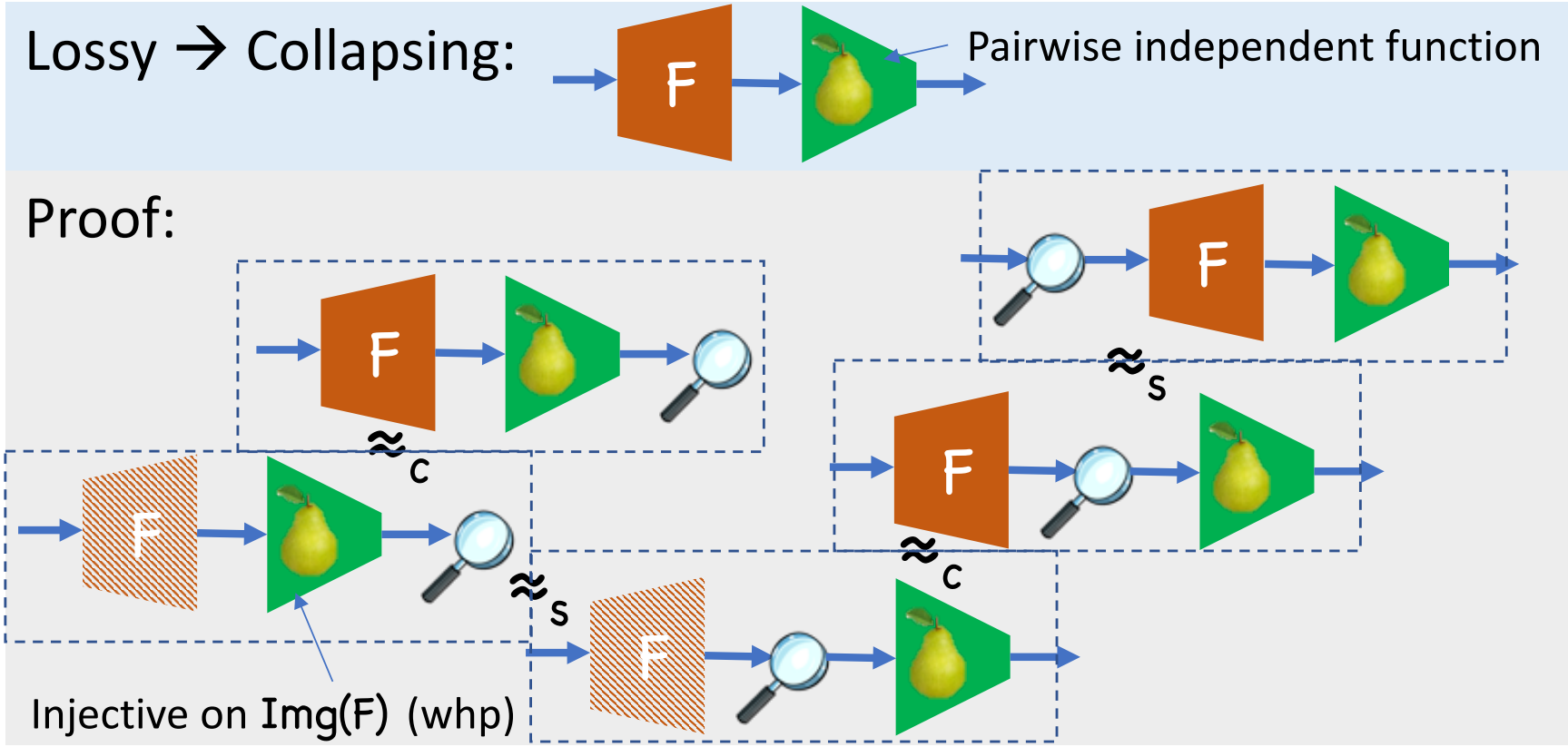
Lossy Mode:



Can construct from LWE

Justify Collapsing: Lossy Functions

[Unruh'16]



Limitations

For PoK's, applying 🍐 destroys structure, makes verification impossible

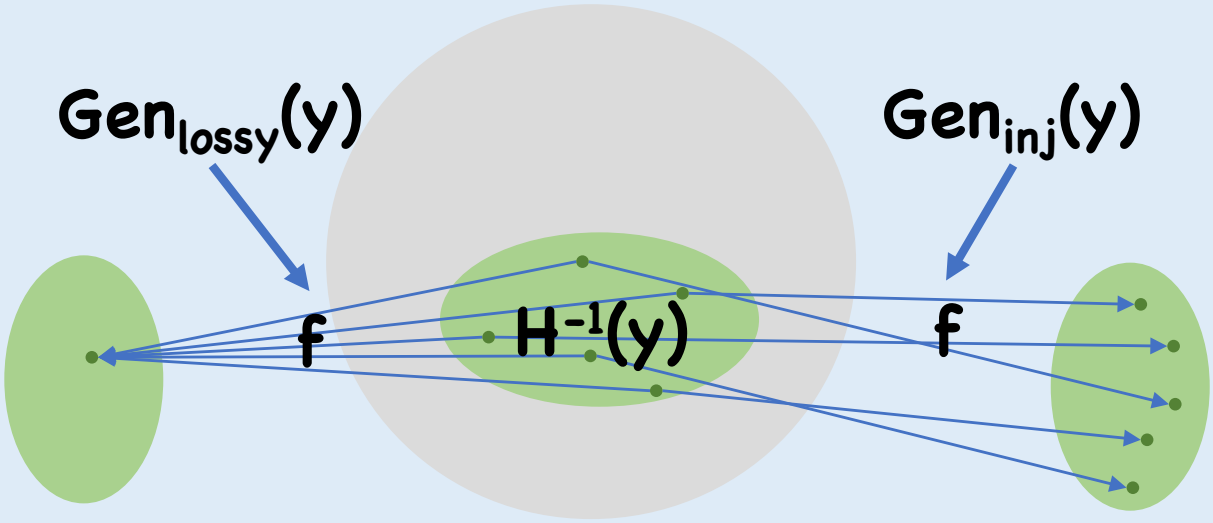
Can remove 🍐 , but then **c** is large; bad for some application (e.g. signatures)

May be inefficient (large intermediate computation)

Improvement: Associated Lossy Funcs

[Liu-Z'19]

Def:



$$Gen_{lossy}(y) \approx_c Gen_{inj}(y)$$

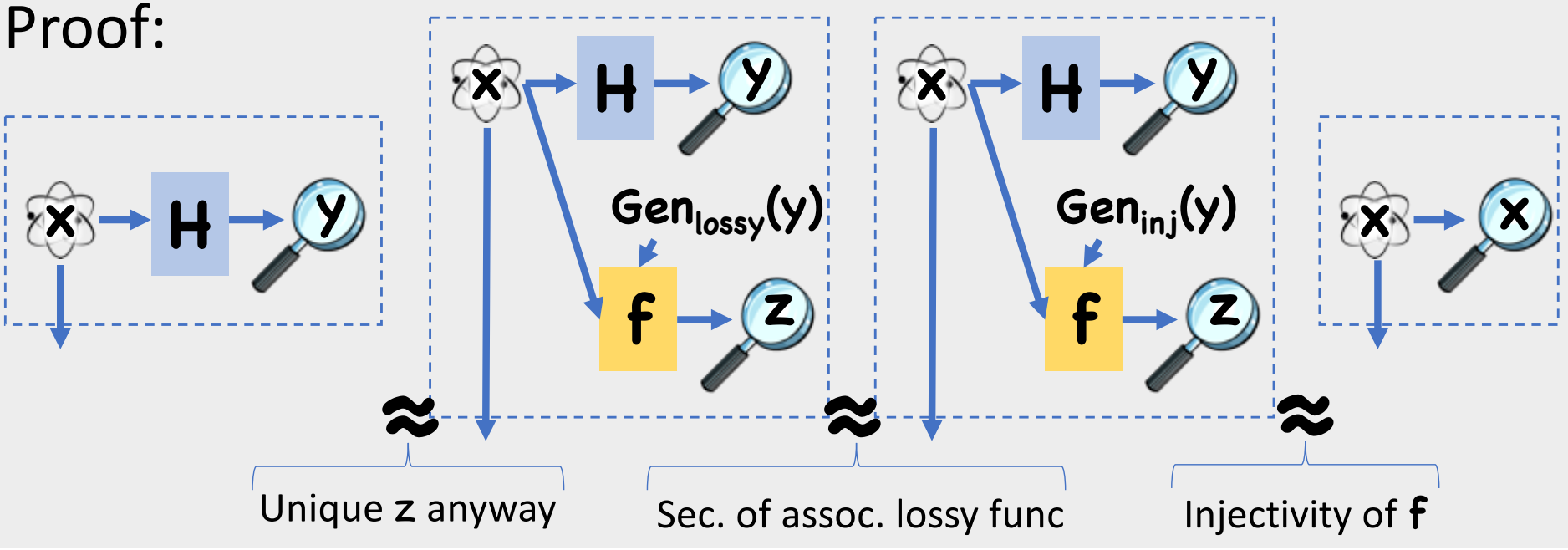
Improvement: Associated Lossy Funcs

[Liu-Z'19]

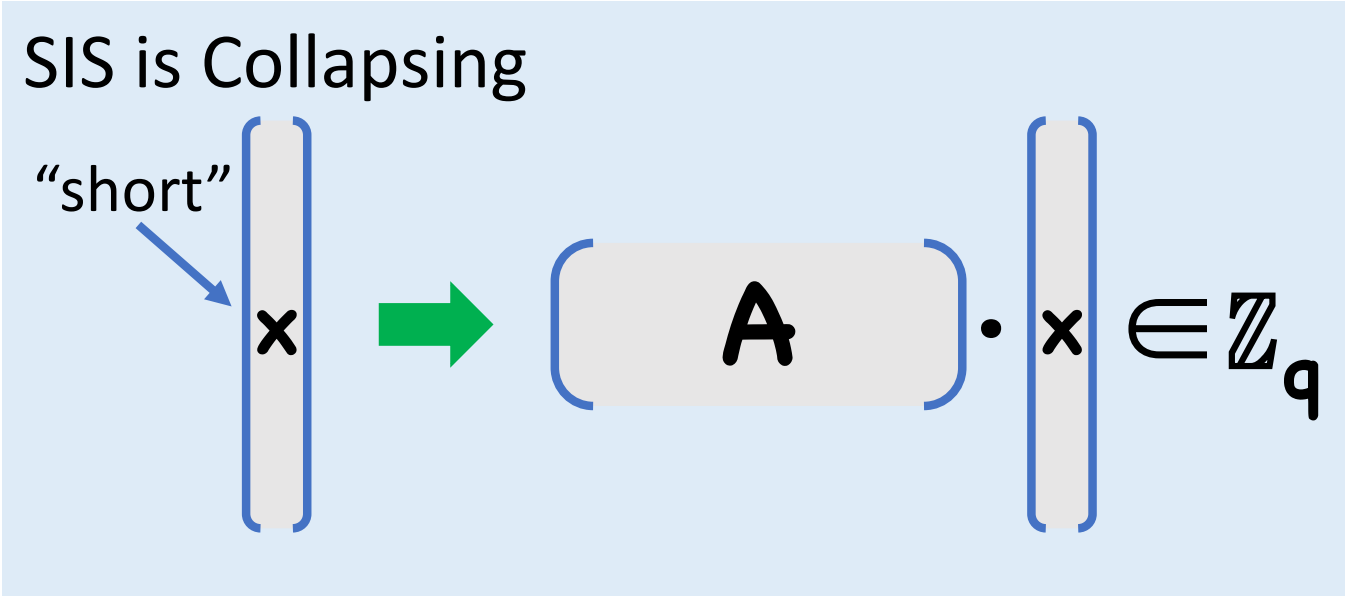
Thm:

H has associated lossy func ➡ **H** is collapsing

Proof:



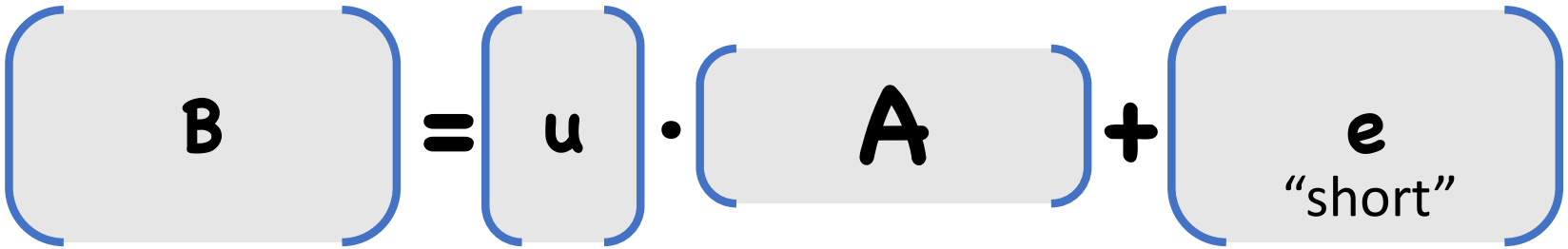
Consequences



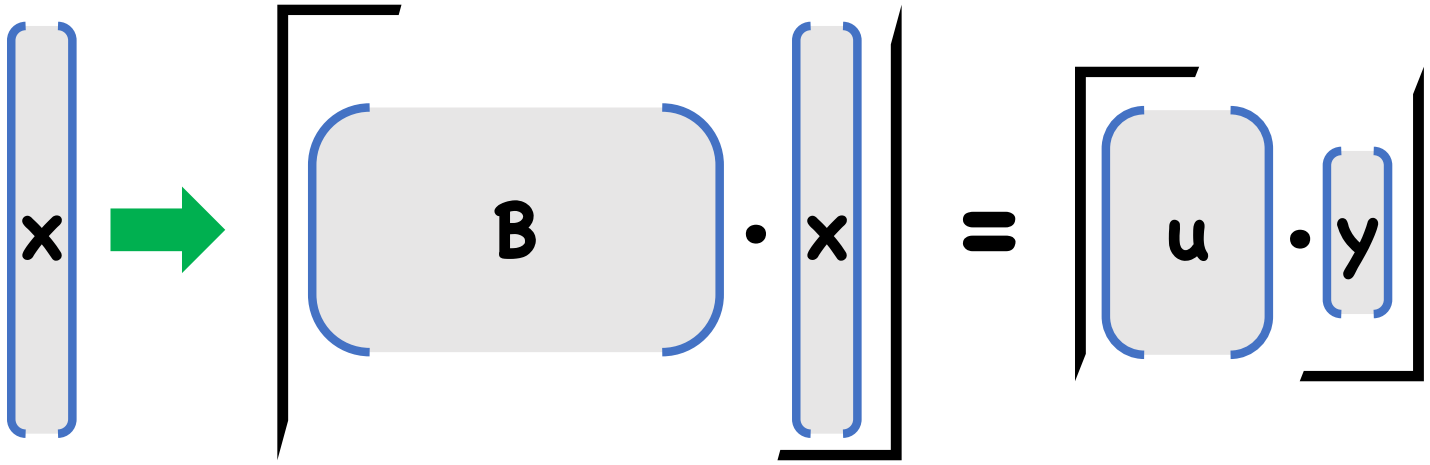
[Lyubashevsky’11] Is a PoK for SIS

Associated Lossy Functions for SIS

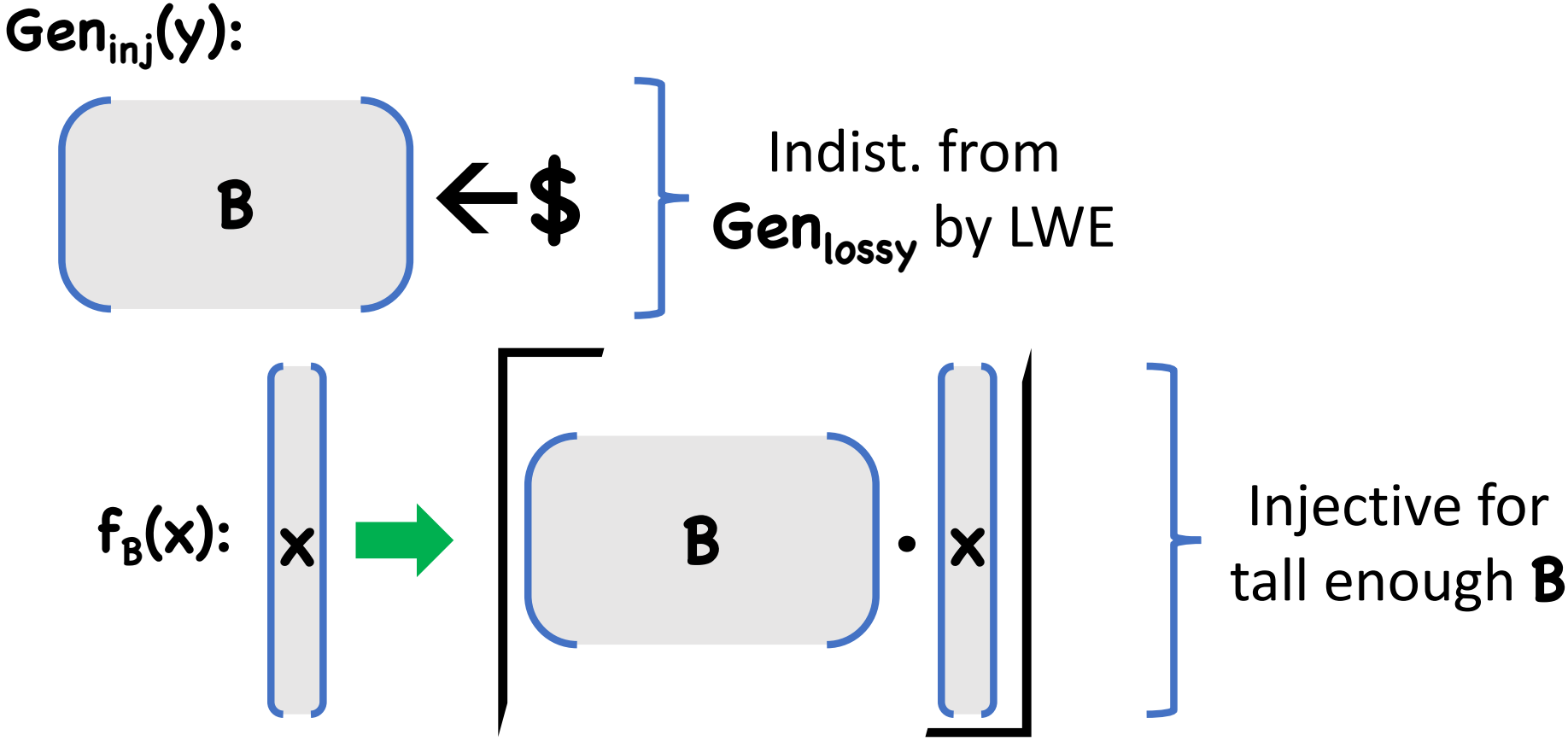
$\mathbf{Gen_{lossy}(y)}$:



$\mathbf{f_B(x)}$:



Associated Lossy Functions for SIS



The Silver Lining…

Proofs of Quantumness

But, can't be verified by others

Thm [Brakerski-Christiano-Mahadev-Vazirani-Vidick'18]:

LWE → Designated verifier (privately verifiable) proof of quantumness

Doesn't require quantum-easy assumptions

Proofs of Quantumness

Suppose $\cancel{A}$ wins coin-flipping game

Proof that $\cancel{A}$ is quantum, relying on collision resistance of $\cancel{H}$

Assuming honest verifier, anyone can tell that $\cancel{A}$ won

Proofs of Unclonable State

PQ collision resistance of H

+

A wins coin-flipping game

State after commitment can't be copied

And, it can be verified

No-Cloning = Quantum Money

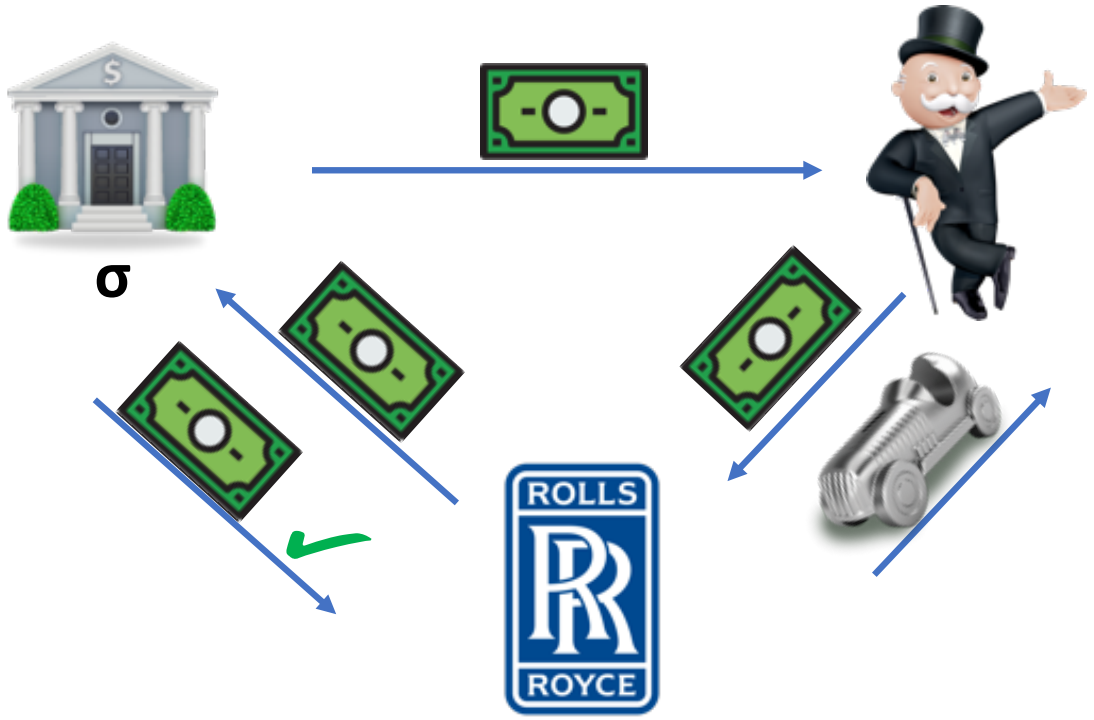
[Wiesner'70]

$= |\Psi\rangle$

Serial # = classical description

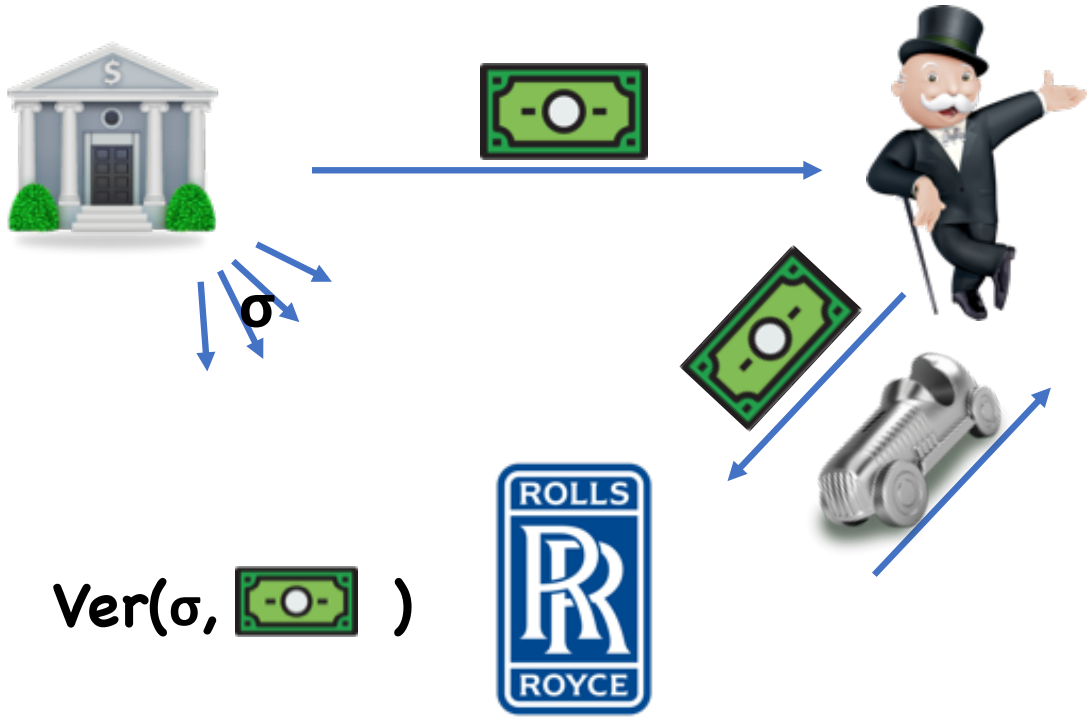
Kept secret

Limits of (Plain) Quantum Money



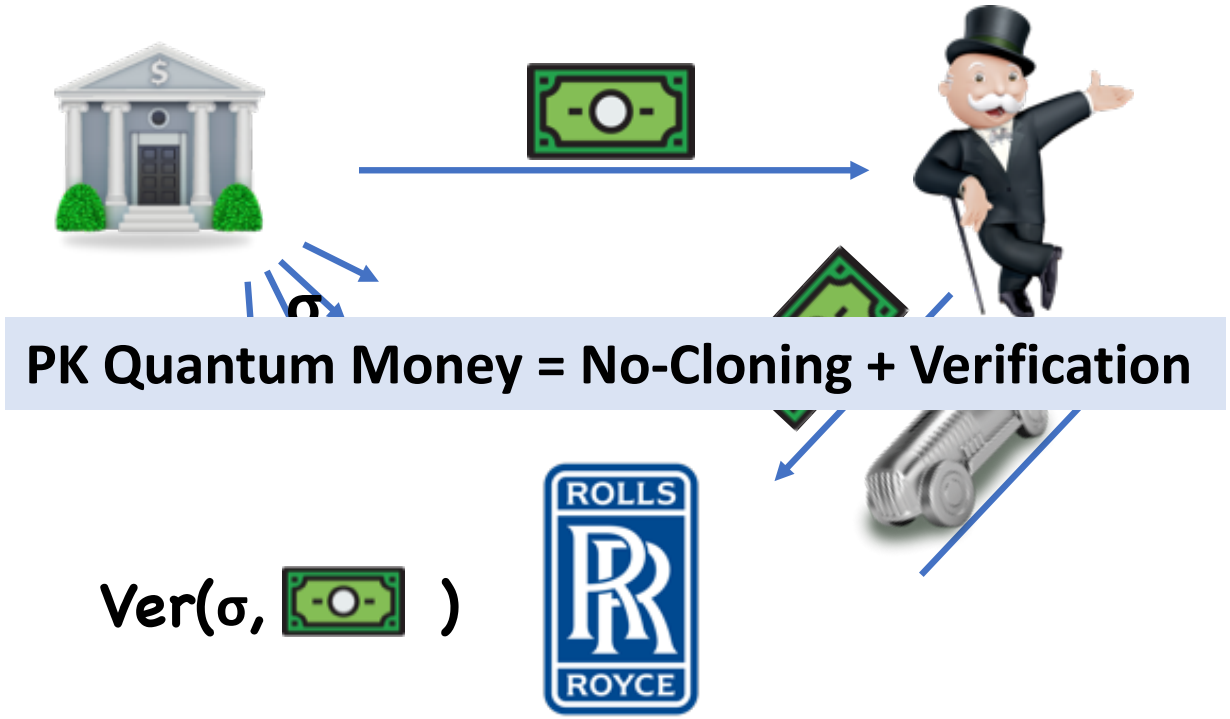
Public Key Quantum Money

[Aaronson'09]



Public Key Quantum Money

[Aaronson'09]



Constructing PK quantum money is a major goal in quantum cryptography

Public Key Quantum Money

PQ collision resistance of H

+

A wins coin-flipping game

[Z'19]

PK Quantum Money

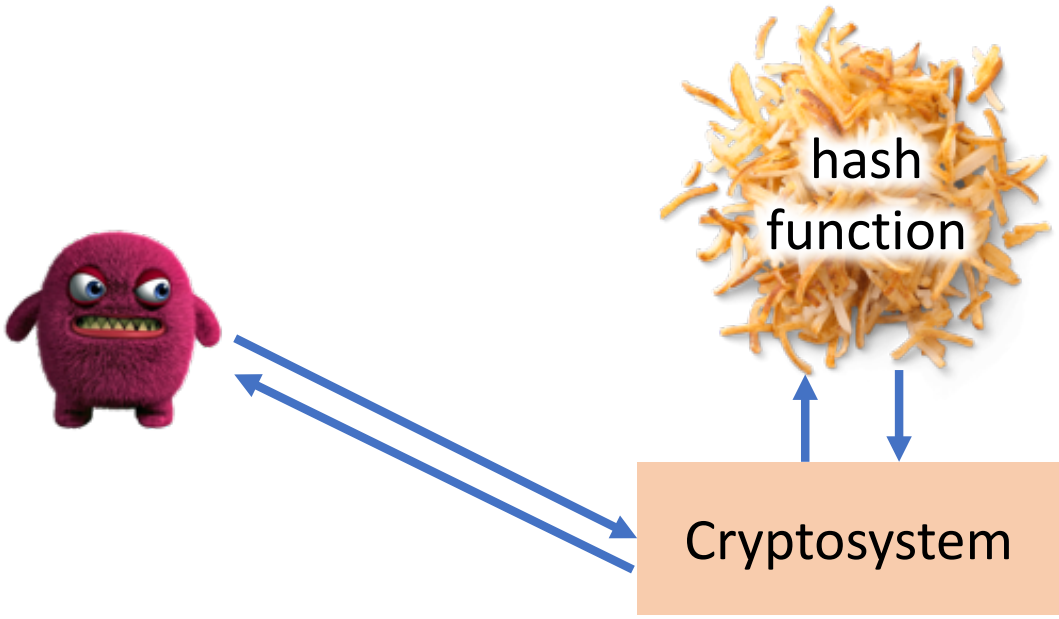
Or more generally, H not collapsing

Takeaway: whenever post-quantum proofs fail, look for interesting quantum crypto applications

Quantum Random Oracle Model, Part 1

Mark Zhandry (Princeton & NTT Research)

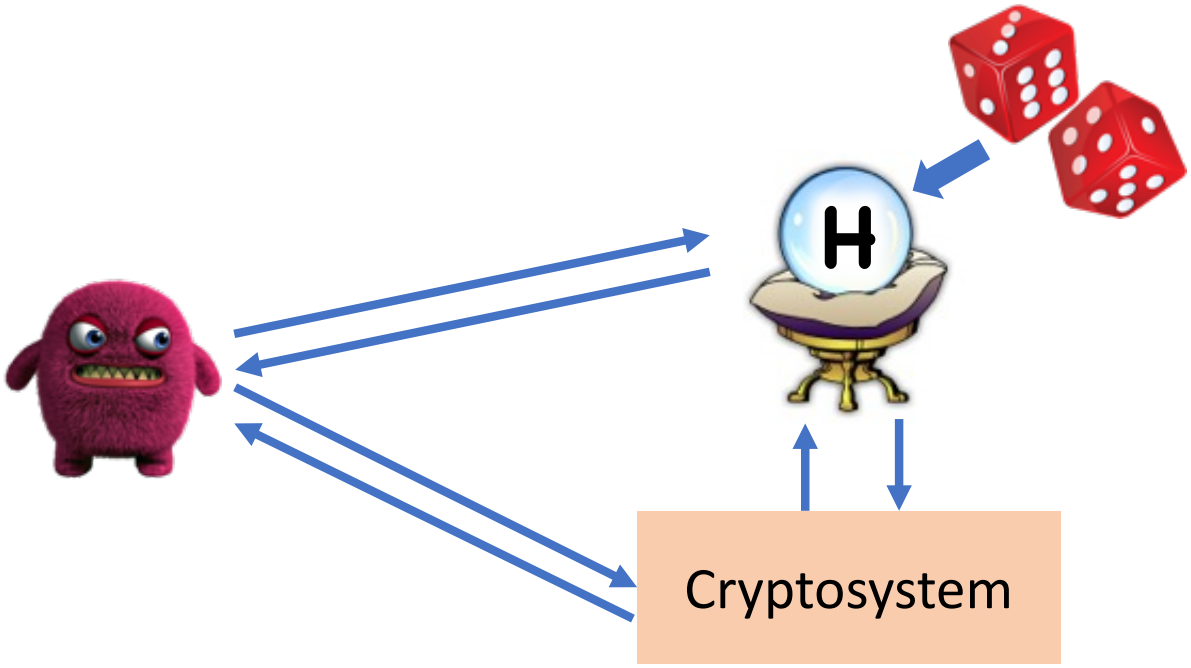
(Classical) Random Oracle Model (ROM)
[Bellare-Rogaway'93]



Examples: OAEP, Fujisaki-Okamoto, Full-Domain Hash, …

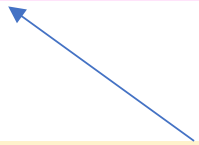
(Classical) Random Oracle Model (ROM)

[Bellare-Rogaway'93]



(Classical) Random Oracle Model (ROM)
[Bellare-Rogaway'93]

Idea: If $\exists$ ROM security proof, any attack must exploit structure of hash function

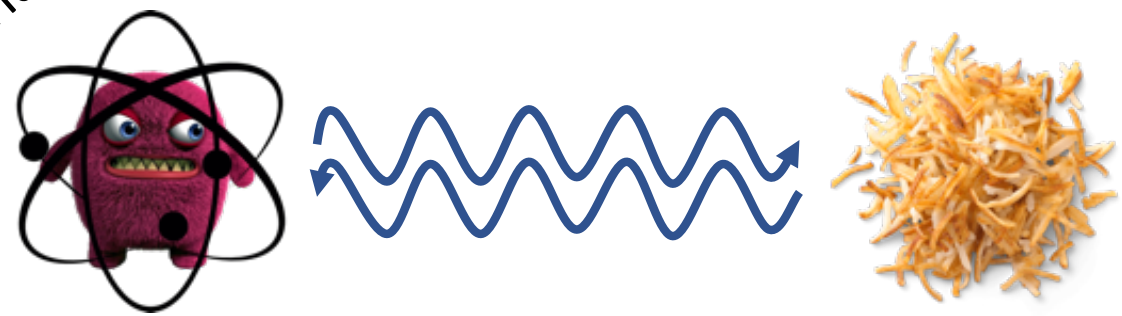


Hopefully not possible for well-designed hash

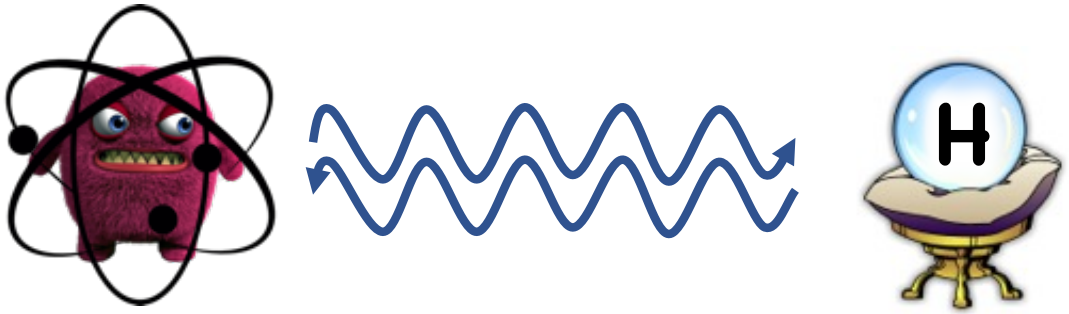
The Quantum Random Oracle Model (QROM)

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

Real World



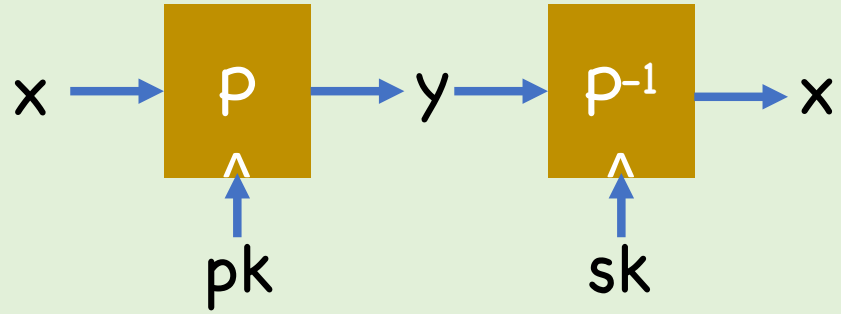
ROM



Now standard in post-quantum crypto

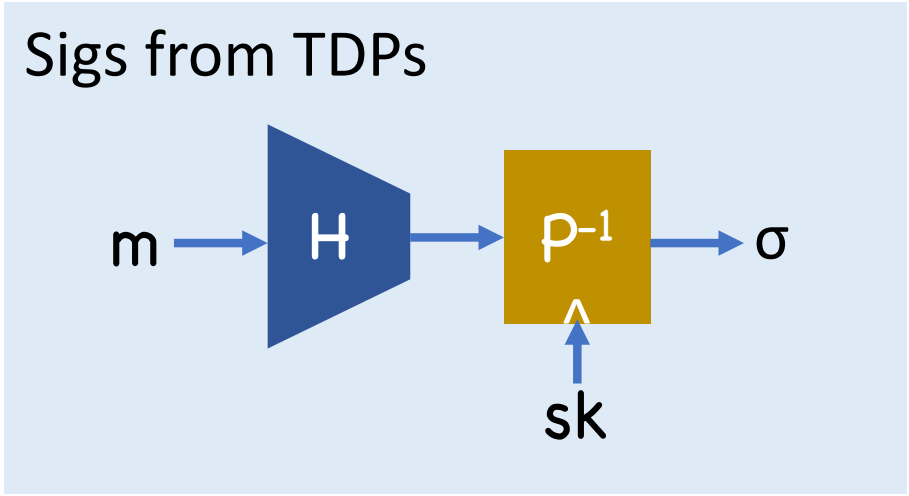
Example: Full Domain Hash

Building Block: Trapdoor Permutations



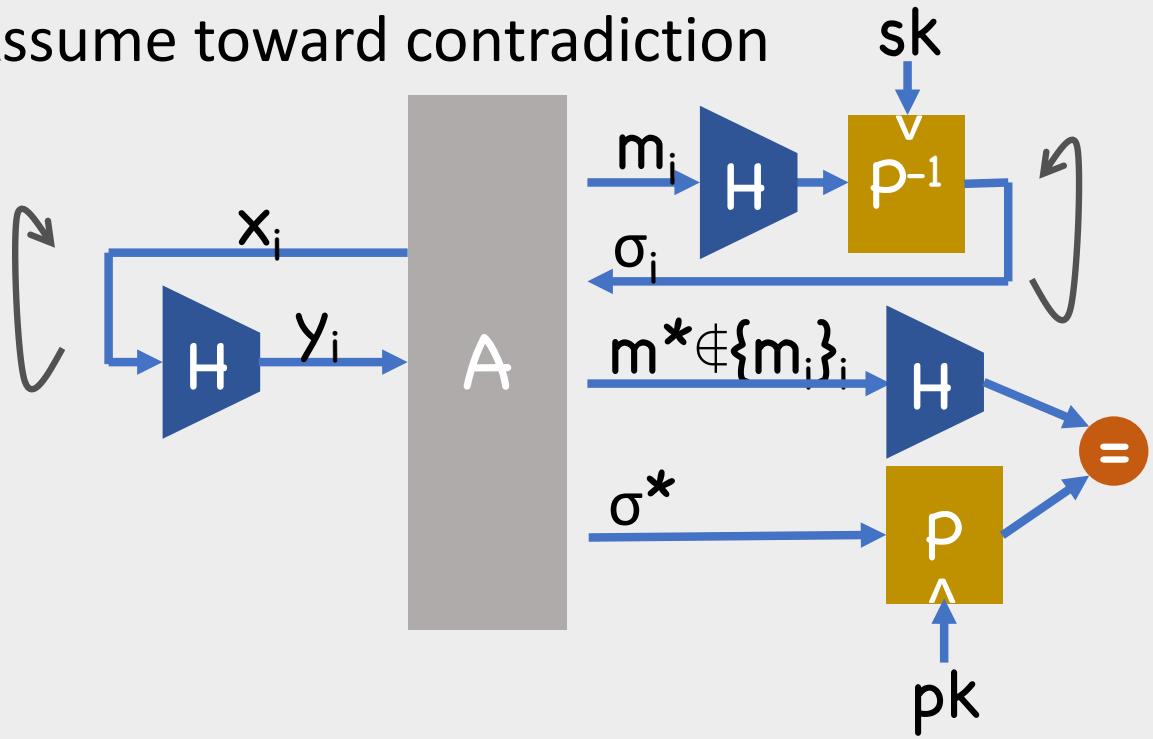
Security: $\forall$ PPT A, Pr[A(pk,y)=x] < negl

Example: Full Domain Hash



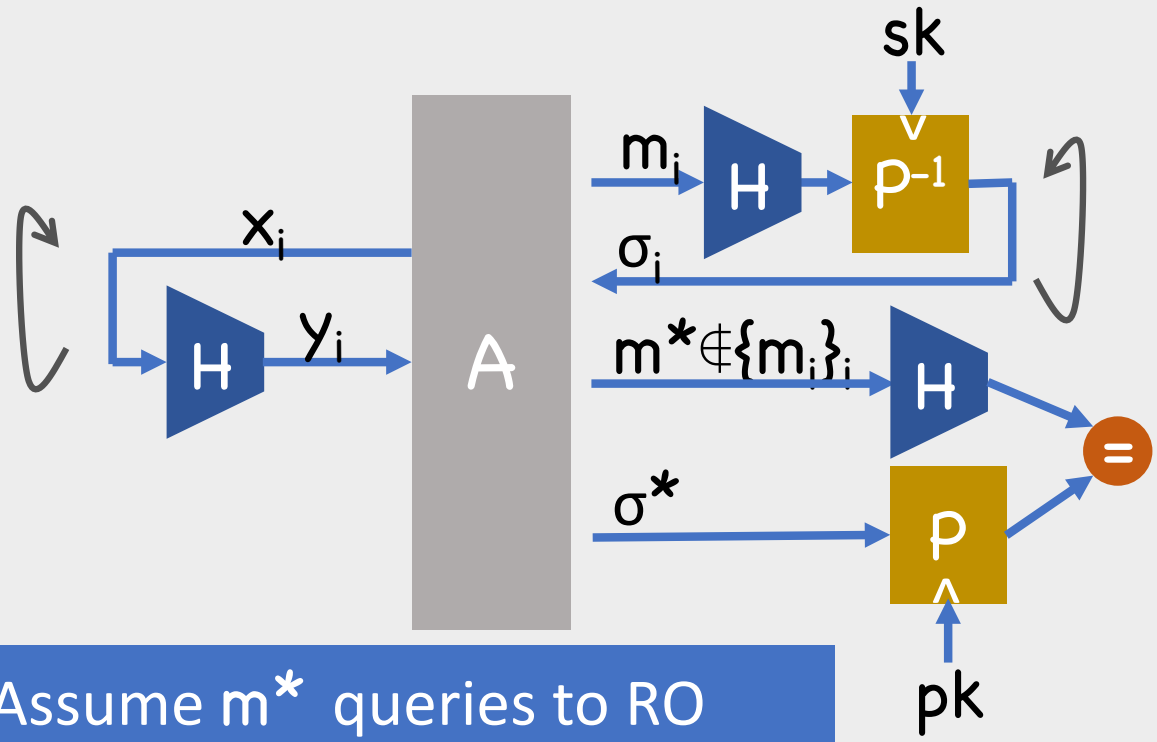
Example: Full Domain Hash

Proof: Assume toward contradiction



Example: Full Domain Hash

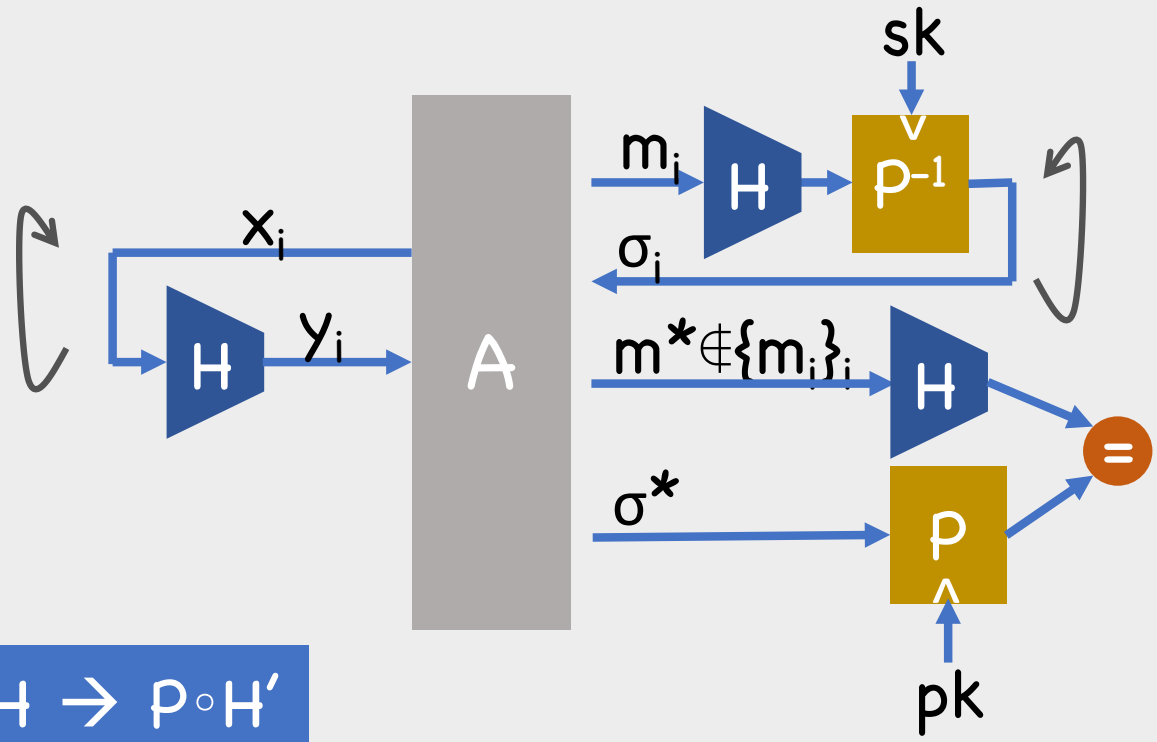
Proof:



Step 0: Assume m^* queries to RO

Example: Full Domain Hash

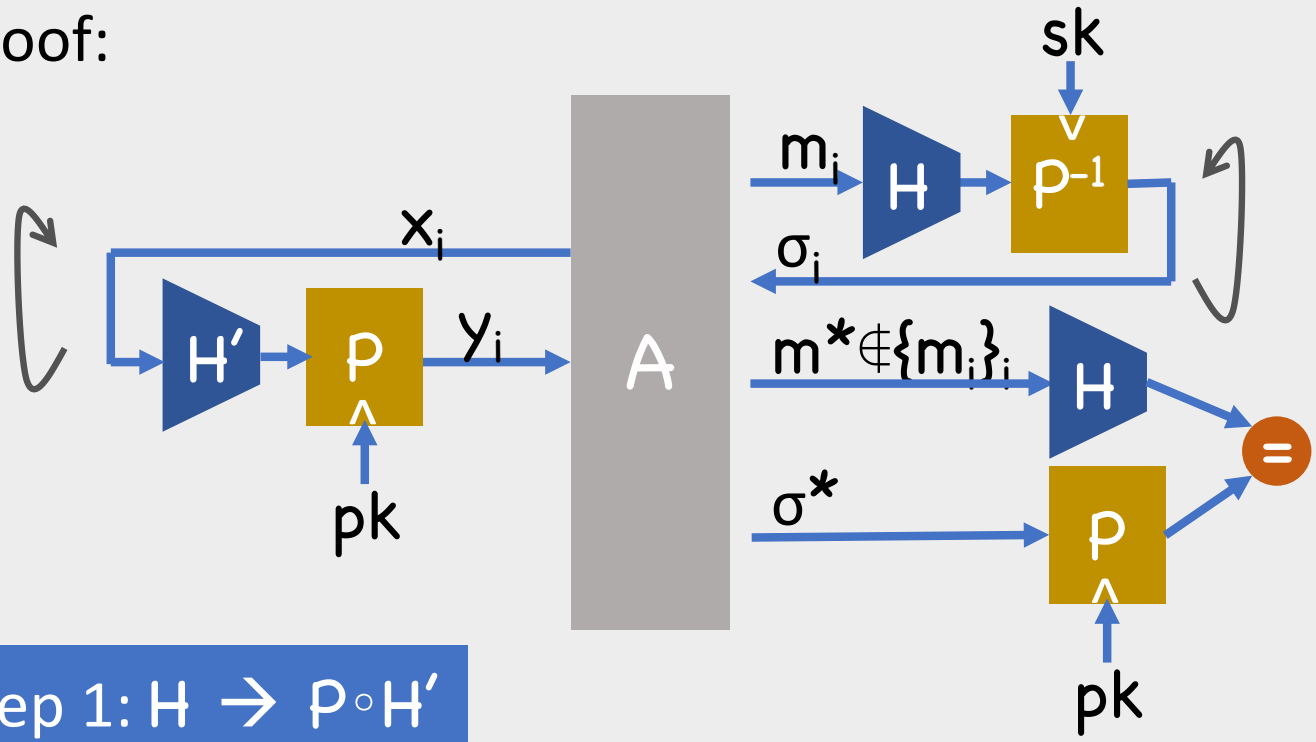
Proof:



Step 1: H → P∘H'

Example: Full Domain Hash

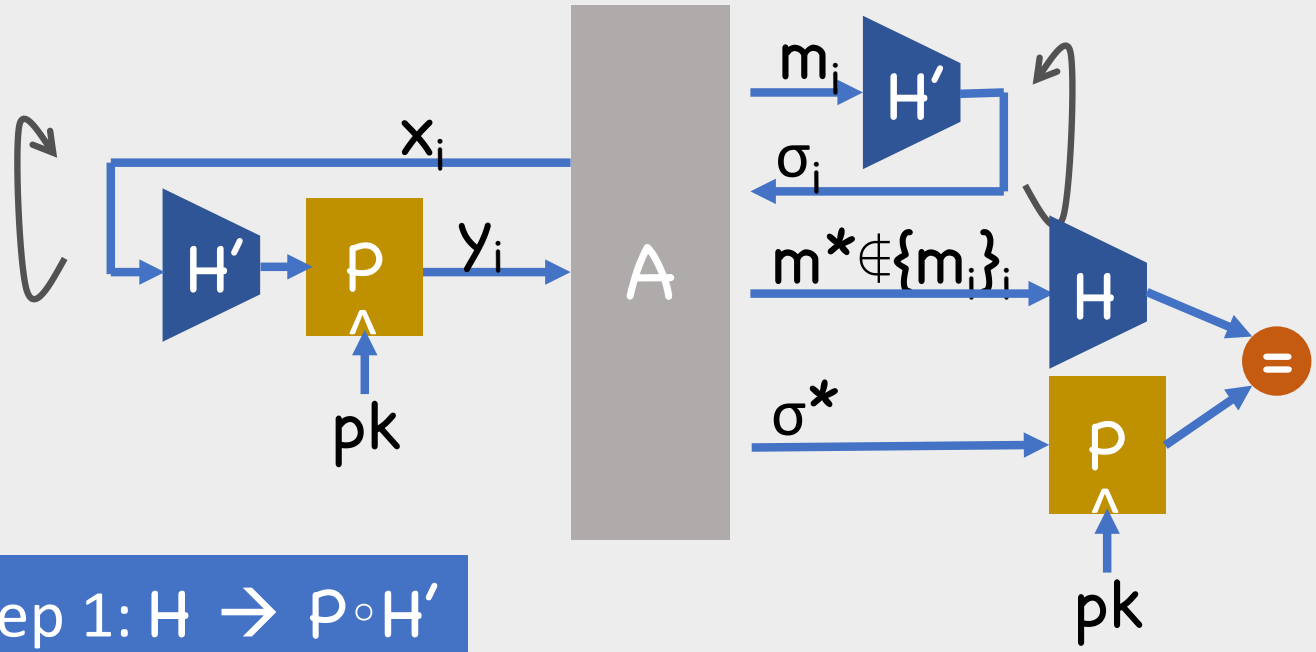
Proof:



Step 1: H → P∘H'

Example: Full Domain Hash

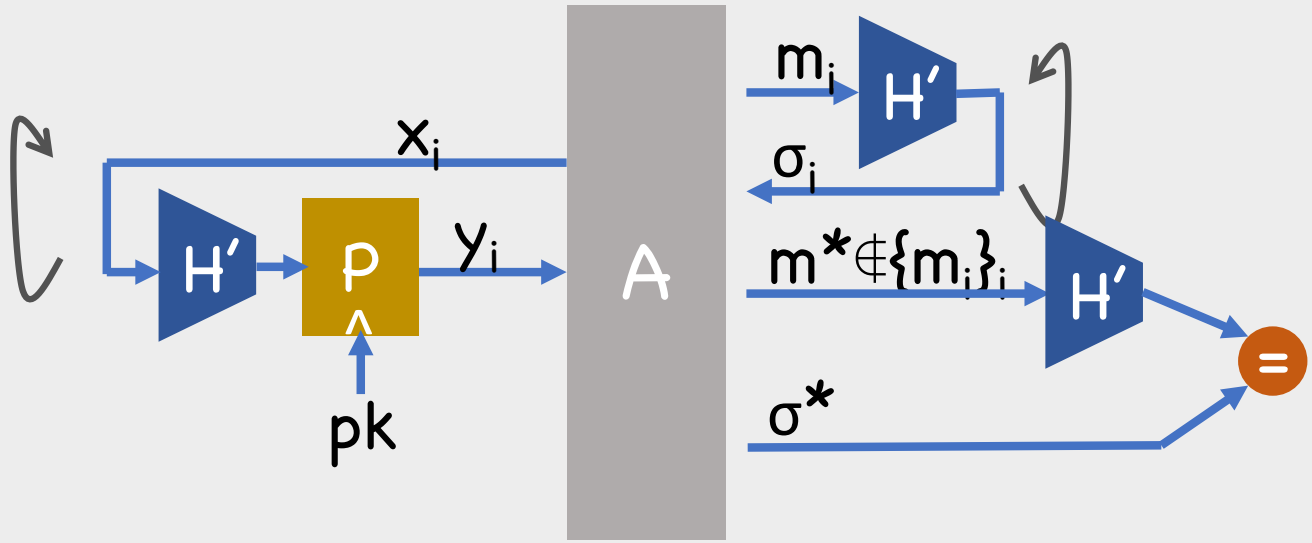
Proof:



Step 1: H → P∘H′

Example: Full Domain Hash

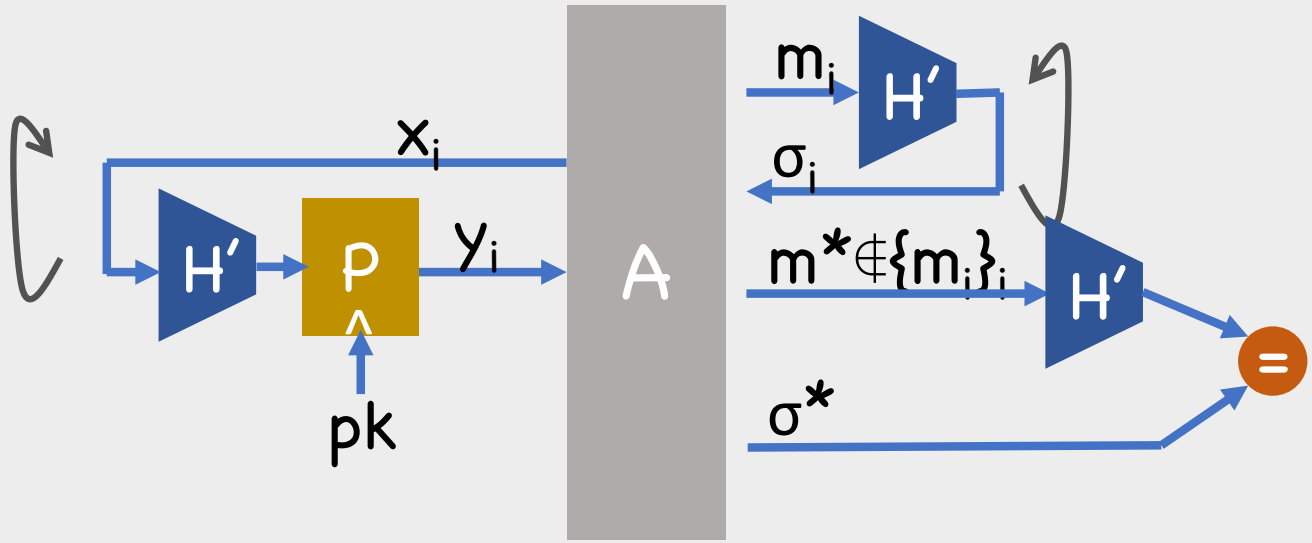
Proof:



Step 1: H → P∘H′

Example: Full Domain Hash

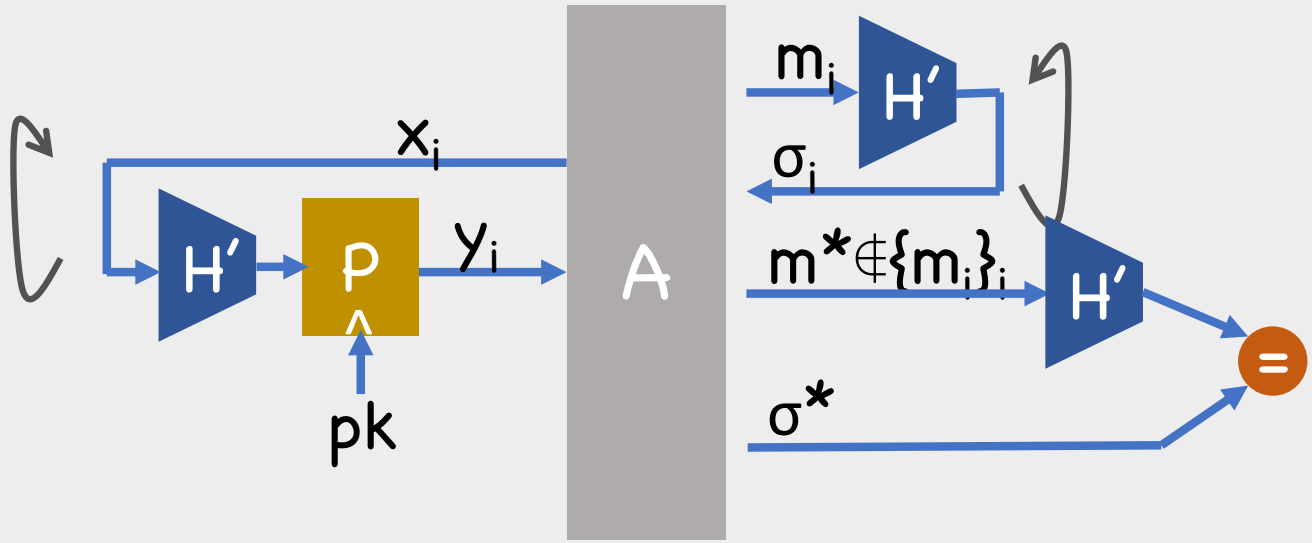
Proof:



Notice: A computes H′(m*), given only P(pk,H′(m*))

Example: Full Domain Hash

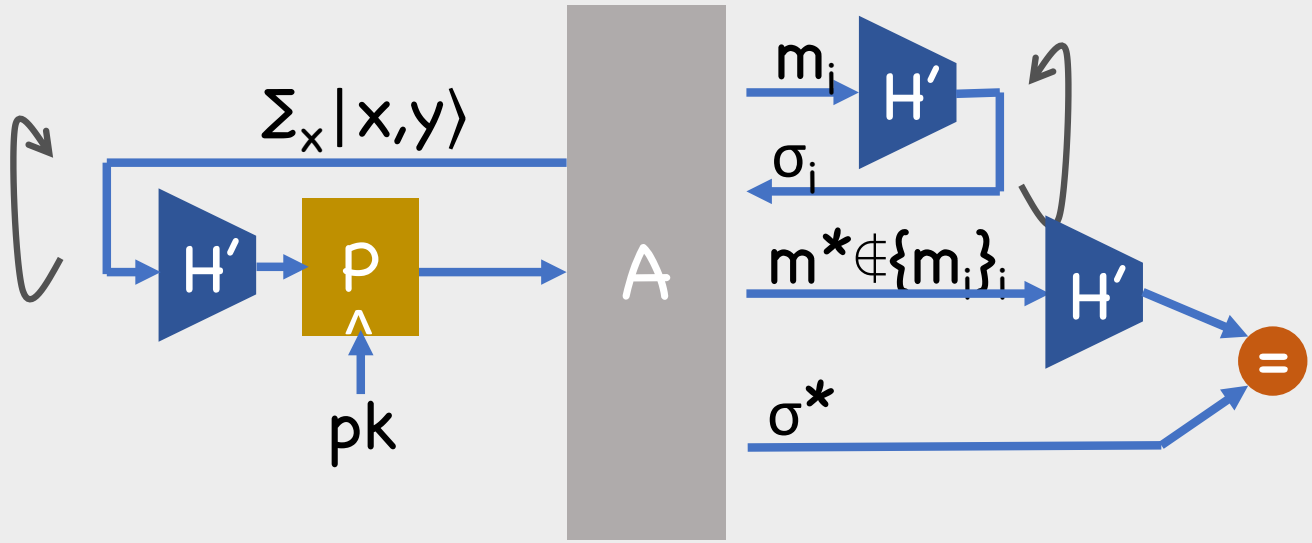
Proof:



B(y): set $H'(x_i)=y$ for random query → advantage ε/q

Example: Full Domain Hash

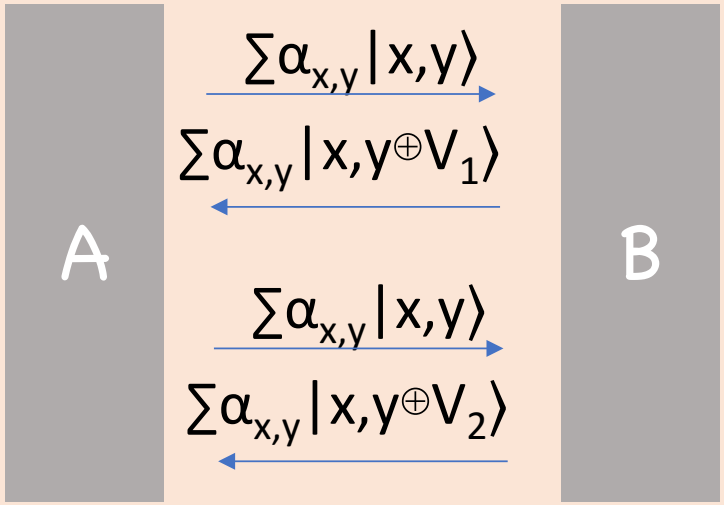
QROM Proof?



How does B insert challenge?

Challenges

Take 1: Per QUERY



Problem: repeated queries?

Problem: distinguishing attack

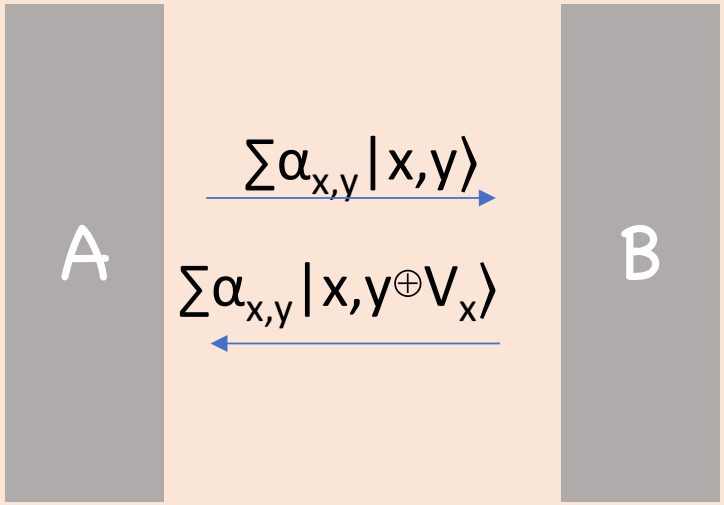
Security Proof Challenges

Typical QROM reductions commit to entire function H at beginning, remain consistent throughout

[Zhang-Yu-Feng-Fan-Zhang'19]: "Committed programming reductions"

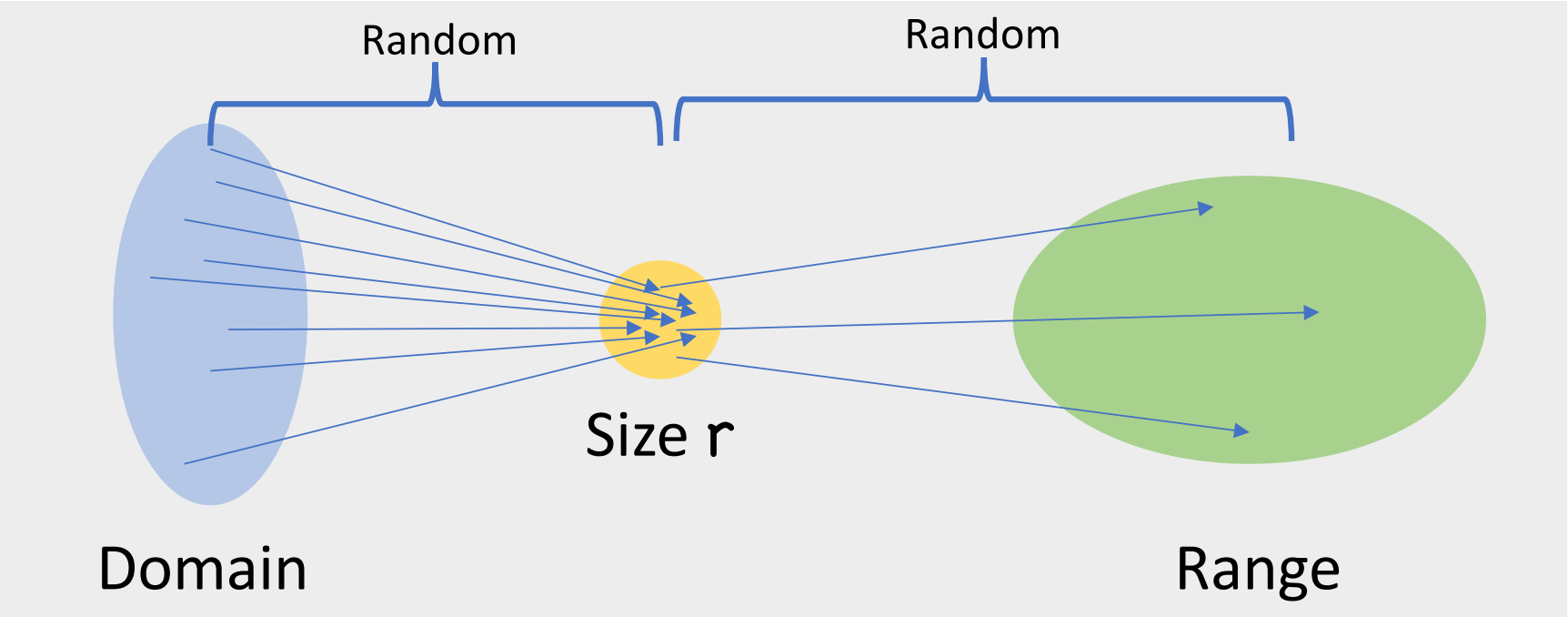
Security Proof Challenges

Take 2: Per VALUE



Problem: exp-many values
→ Pr[correctly guess m*] =negl

Small Range Distributions



Small Range Distributions

Thm [Z'12b]: No q quantum query alg can distinguish SR_r from random, except with probability $O(q^3/r)$.

Quantum collision finding ➡ bound tight

Finishing The Proof

$$\Pr[A \text{ wins} \mid H' \text{ random}] \geq \varepsilon$$

$$\Pr[A \text{ wins} \mid H' = SR_r] \geq \varepsilon - O(q^3/r)$$

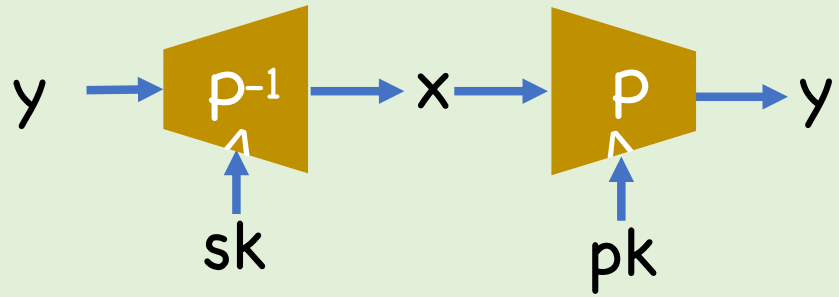
$B(y)$ inserts y into random output

$$\Rightarrow \Pr[B \text{ inverts } y] \geq \varepsilon/r - O(q^3/r^2) = O(\varepsilon^2/q^3)$$

$$r = O(q^3/\varepsilon)$$

Example: Full Domain Hash, Take 2

Building Block: Pre-image Sampleable Funcs

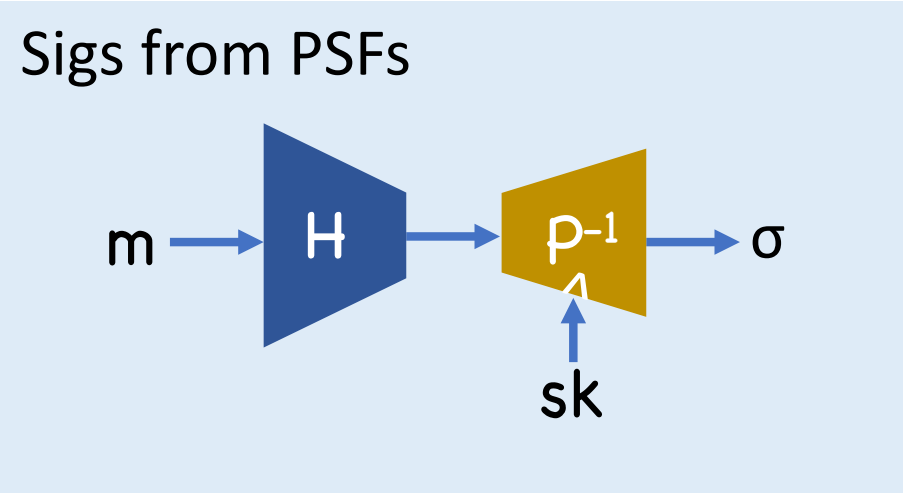


Security: (1) Collision resistant

(2) random y → ≈random x

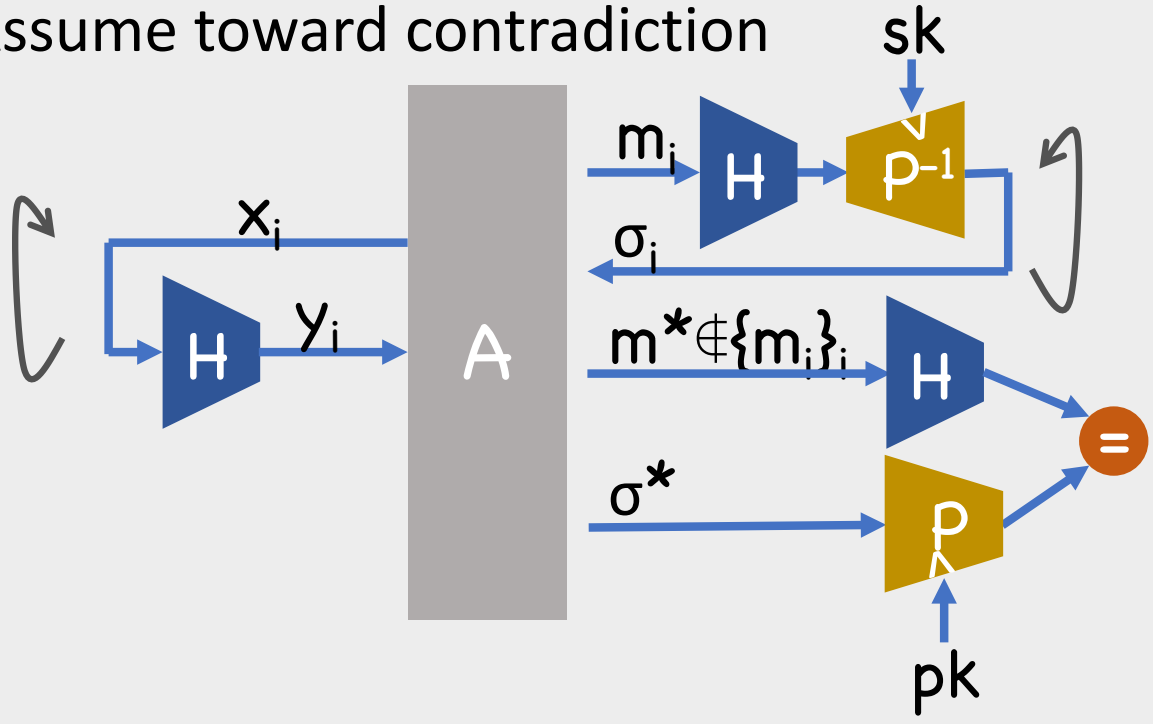
[Gentry-Peikert-Vaikuntanathan'08]: construction from LWE

Example: Full Domain Hash, Take 2



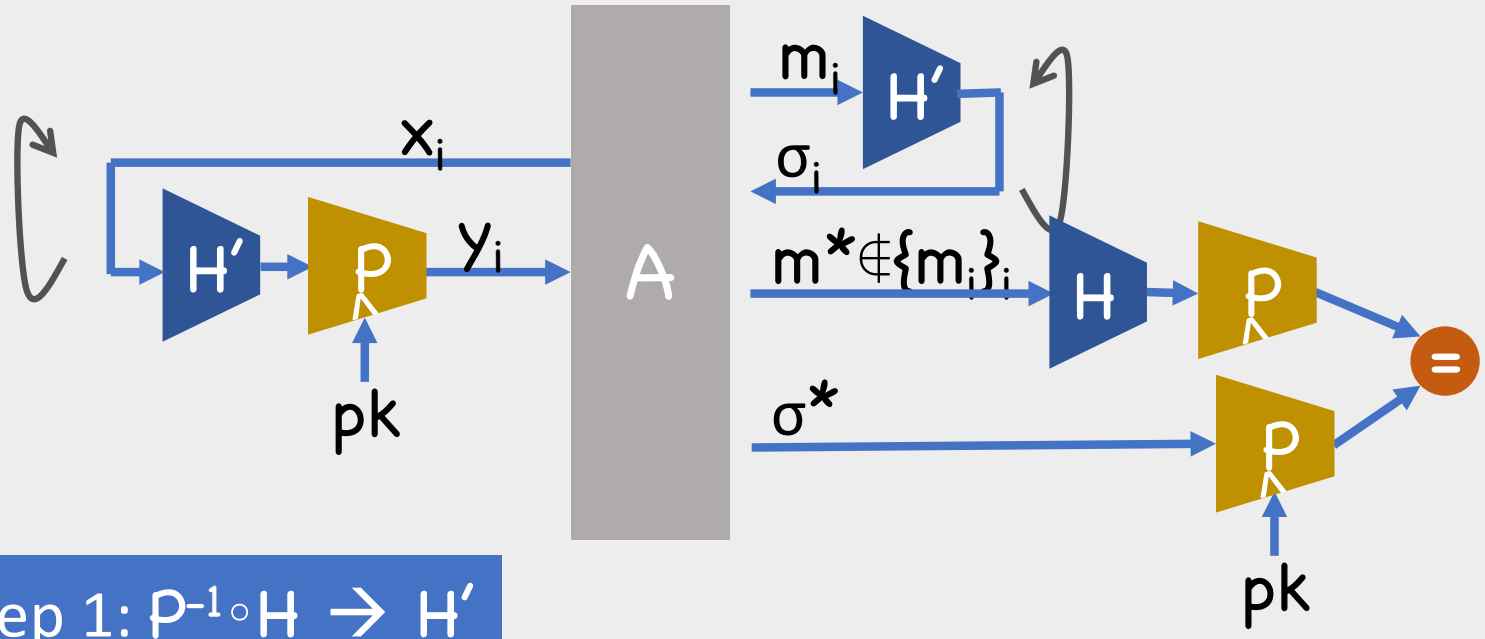
Example: Full Domain Hash, Take 2

Proof: Assume toward contradiction



Example: Full Domain Hash, Take 2

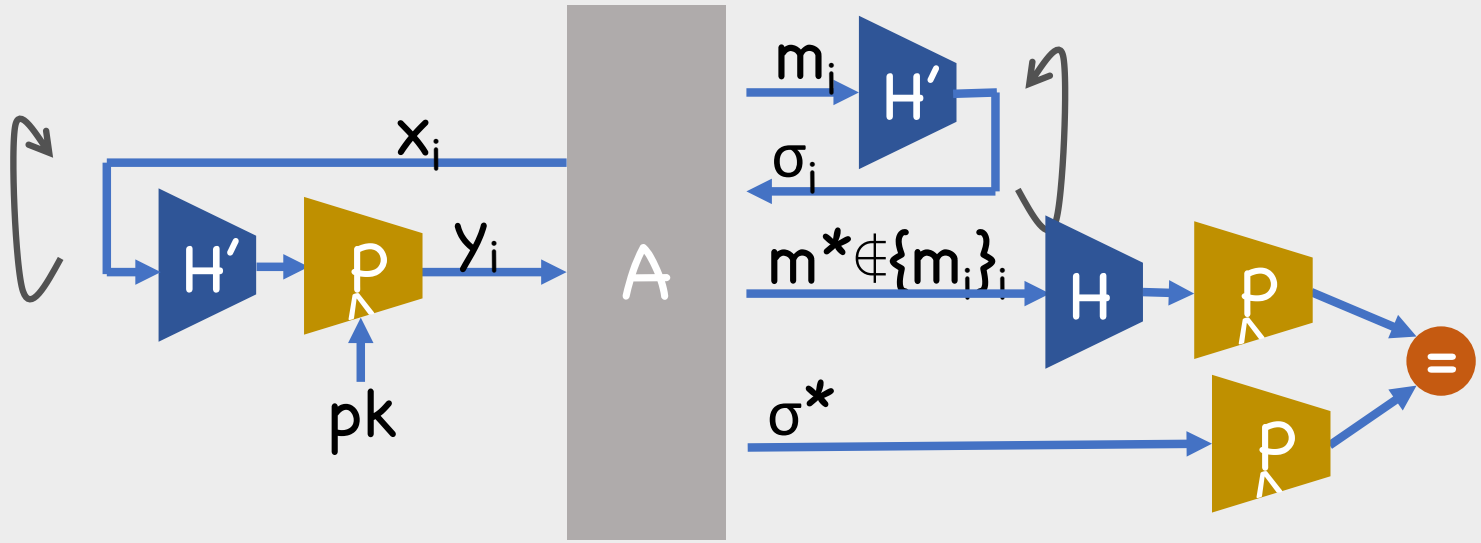
Proof:



Step 1: $P^{-1} \circ H \rightarrow H'$

Example: Full Domain Hash, Take 2

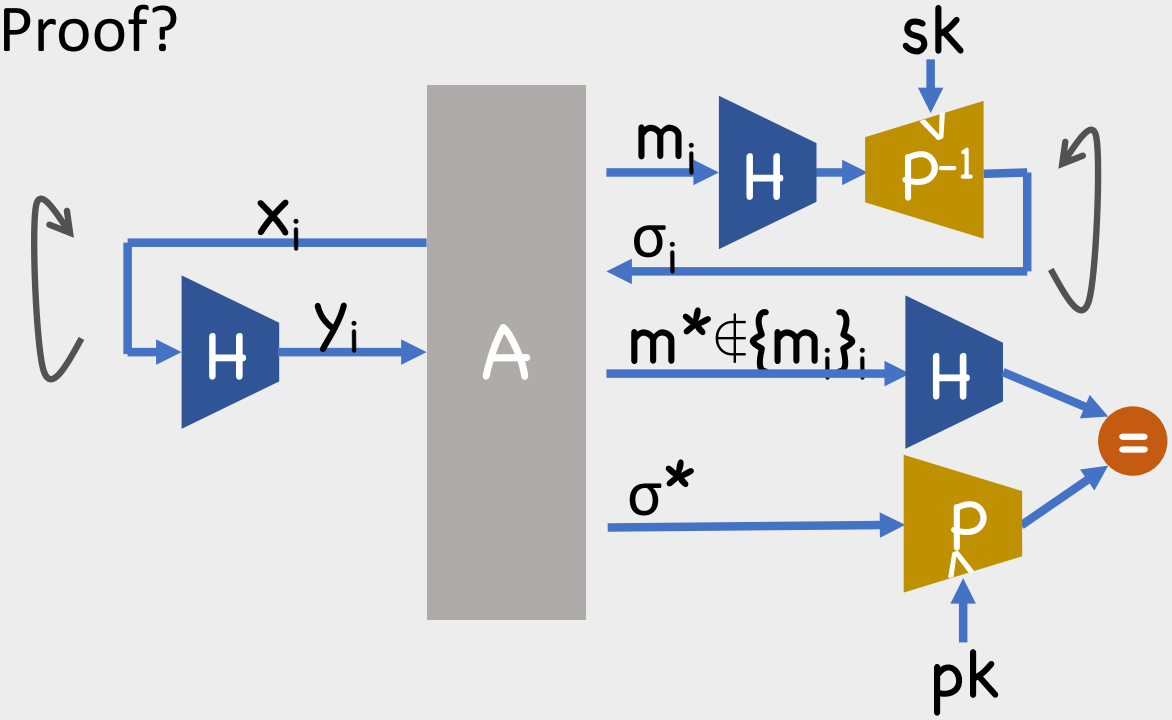
Proof:



Notice: H(m*), σ* form collision → advantage ε

Example: Full Domain Hash, Take 2

QROM Proof?



Example: Full Domain Hash, Take 2

Main* QROM issue: simulating H' efficiently

As before, can do using $2q$-wise independence

*some issues having to do with $P^{-1}(y)$ being only approximately uniform

Rule of Thumb

Rule of Thumb: If loss of classical reduction is independent of **q**, good chance we can upgrade to quantum security

No per query hybrid

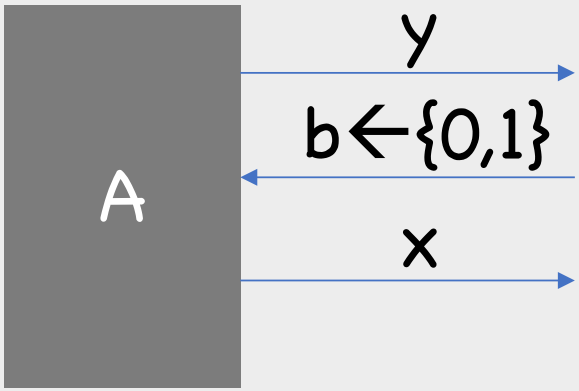
If loss in reduction depends on **q**, new reduction likely needed, maybe impossible

Can All ROM Proofs be Upgraded?

Thm [Yamakawa-Z'20]: No, assuming LWE or relative to an oracle

Recall: Impossibility of Quantum Rewinding
[Ambainis-Rosmanis-Unruh'14]

Coin flipping/commitment game

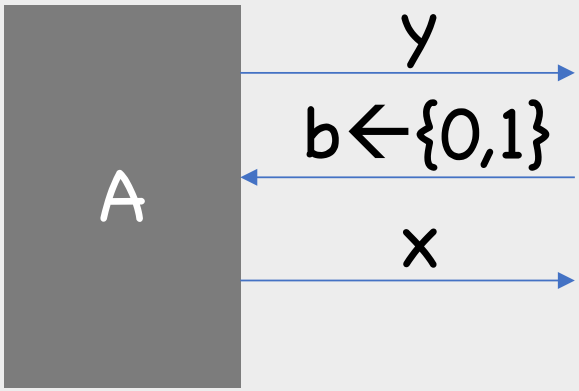


Win if
- Hash(x)=y
- $x_1 = b$

Devised *quantum* A and col. res. Hash where Pr[A wins] ≈ 1

New Game

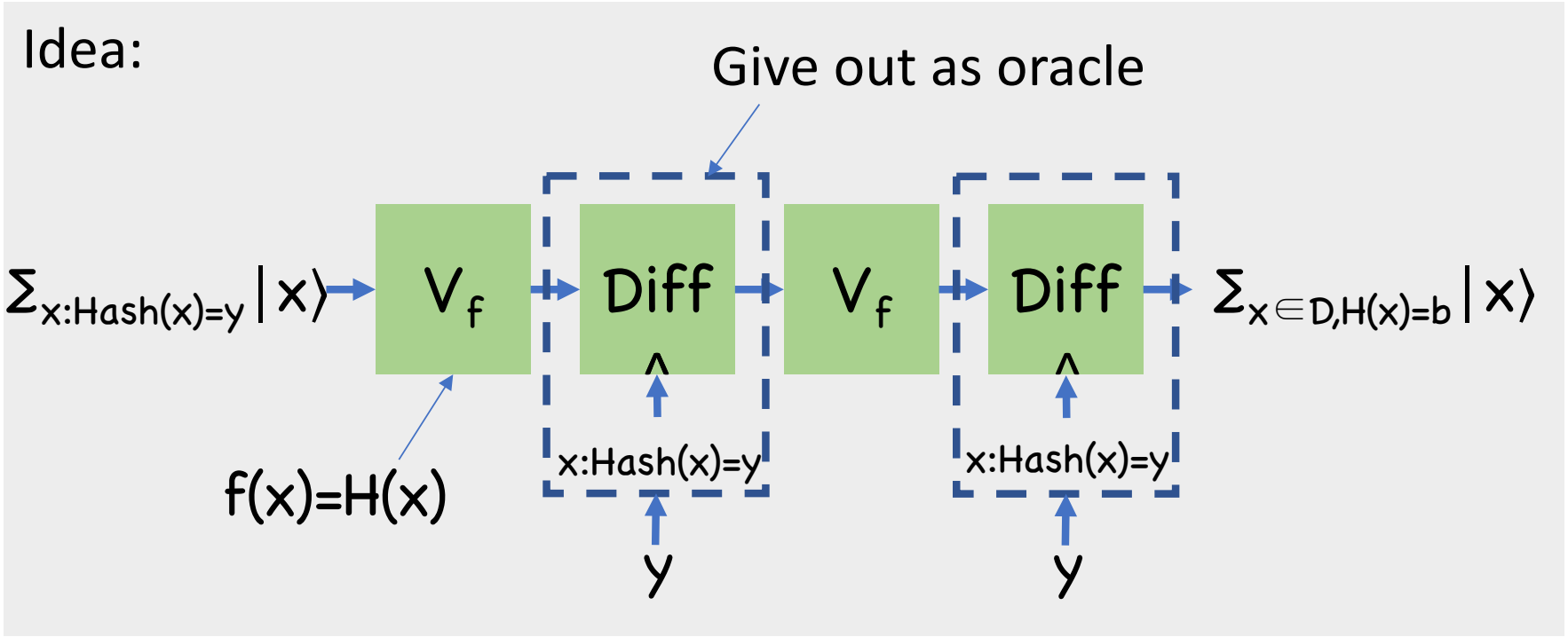
Coin flipping/commitment game



Win if
- Hash(x)=y
- H(x) = b (1-bit RO)

Essentially same A,Hash work here

Quantum Alg


Idea:
Give out as oracle
$\Sigma_{x:Hash(x)=y} |x\rangle$
V_f
Diff
V_f
Diff
$\Sigma_{x \in D, H(x)=b} |x\rangle$
f(x)=H(x)
x:Hash(x)=y
x:Hash(x)=y
y
y

No Classical-Query Alg

Suppose $\exists$ classical query quantum A s.t. $Pr[A \text{ wins}] \geq \frac{1}{2}+\varepsilon$

- Consider H queries on x s.t. $Hash(x)=y$
- First such query x_0 has prob $\frac{1}{2}$ of $H(x_0)=b$
- If A only ever outputs x_0, $Pr[A \text{ wins}] \leq \frac{1}{2}$
- Therefore, A must sometimes output $x_1 \neq x_0$
- But then x_0, x_1 form collision for $Hash$

QROM Impossibility

[Yamakawa-Z'20]: More generally, upgrade proofs of quantumness to proofs of quantum access to RO

Up Next

Tomorrow, will look at further examples

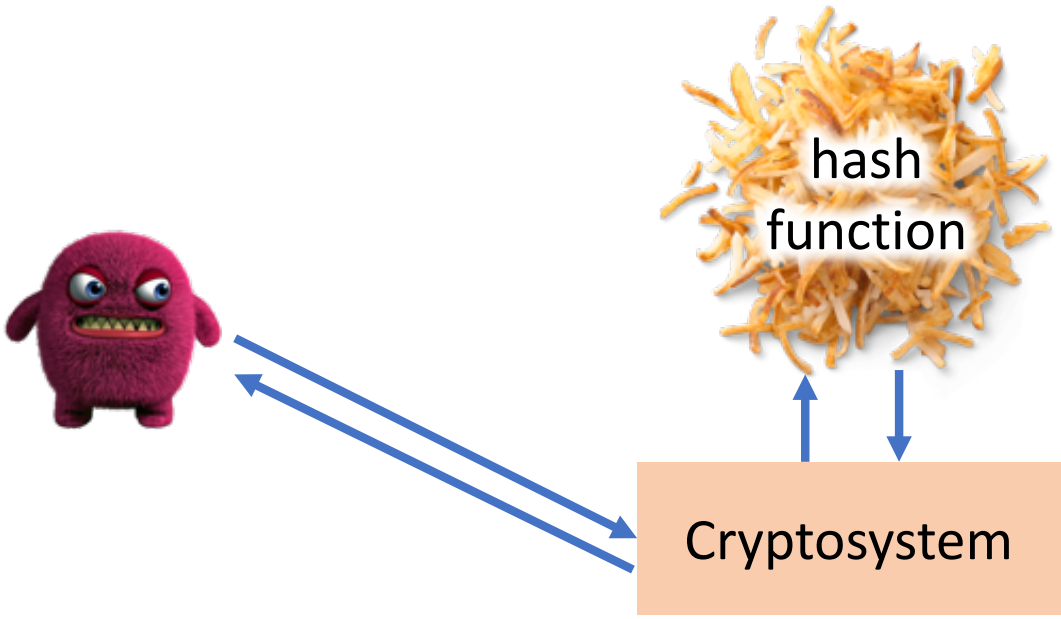
In particular, we will see barriers/impossibilities for committed programming reductions, and how to overcome them

Quantum Random Oracle Model, Part 2

Mark Zhandry (Princeton & NTT Research)

Recap: Classical ROM

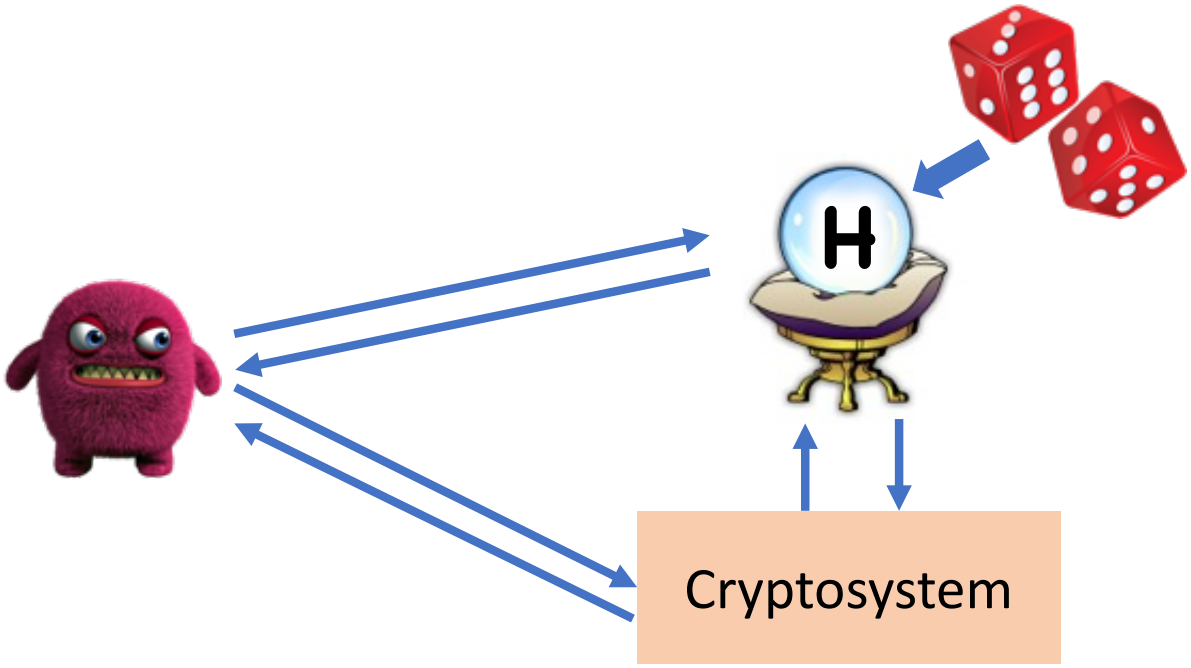
[Bellare-Rogaway'93]



Examples: OAEP, Fujisaki-Okamoto, Full-Domain Hash, …

Recap: Classical ROM

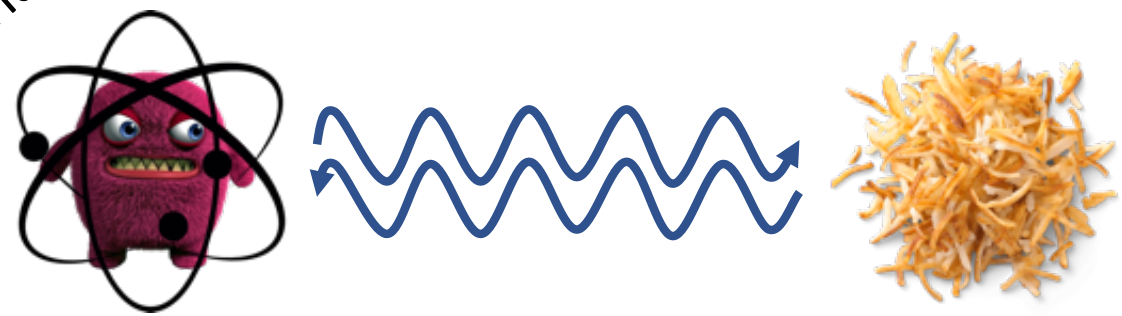
[Bellare-Rogaway'93]



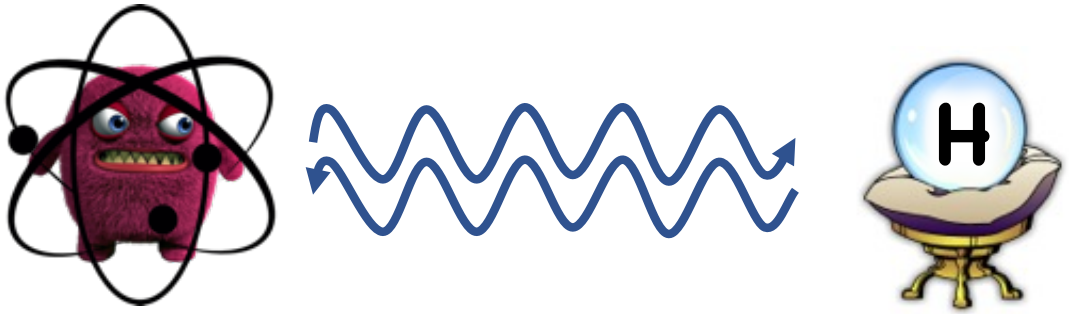
The Quantum Random Oracle Model (QROM)

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

Real World



ROM



Now standard in post-quantum crypto

Security Proof Challenges

Typical QROM reductions commit to entire function H at beginning, remain consistent throughout

[Zhang-Yu-Feng-Fan-Zhang'19]: "Committed programming reductions"

Limits of Committed Programming Reductions

What classical ROM proofs admit CPReds, and which don’t?

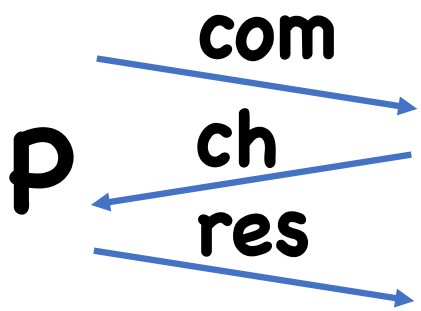
What to do if no CPRed?

Example: The Fiat-Shamir Transform

[Fiat-Shamir'87]

(public coin, HV)

3-Round Proof (of Knowledge)



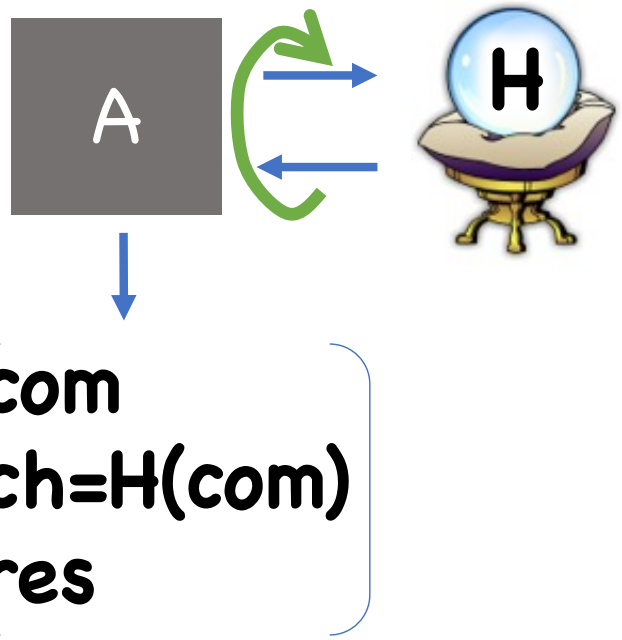
NI Proof (of Knowledge)

$$\pi = \begin{pmatrix} \text{com} \\ \text{ch=H(com)} \\ \text{res} \end{pmatrix}$$

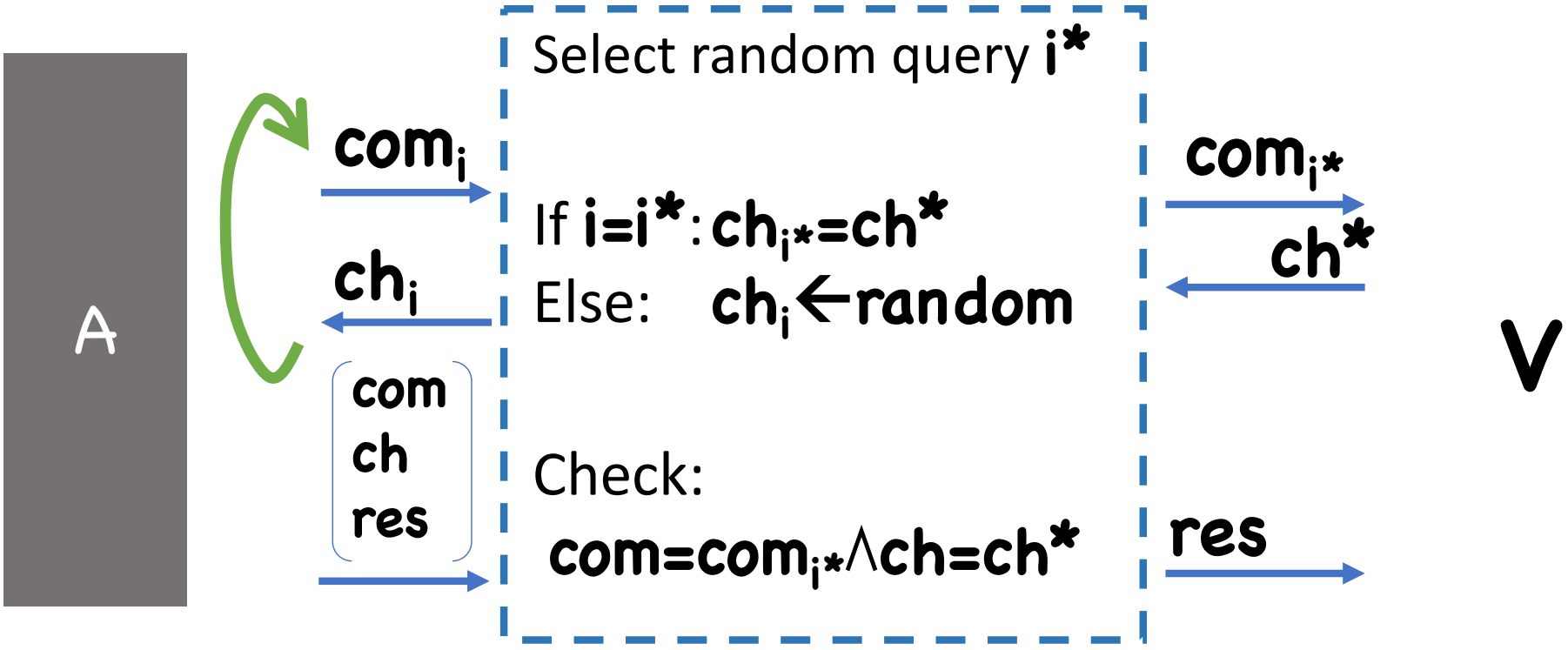
Also: Identification protocols → signatures

Classical Fiat-Shamir Proof

Assume:



Classical Fiat-Shamir Proof



Problems with Fiat-Shamir in QROM

Quantum analog of selecting random query?

Use small range distributions!?

Query extraction:
A's state disturbed by extracting $\mathbf{com}_{i^*}$

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

Problems with Fiat-Shamir in QROM

Thm [Dagdelen-Fischlin-Gagliardoni'13]:
There is no CPRed for Fiat-Shamir

Intuition: two cases:
(1) H committed before sending com to V
→ V's ch independent of A's ch
(2) H committed after sending com to V
→ A's com independent of reduction's com

Solutions?

[Unruh'15]: Use different conversion
Idea: ~~A~~ commits to all possible responses → can open using knowledge of RO
Problem: Less efficient

[Dagdelen-Fischlin-Gagliardoni'13,Unruh'17, Kiltz-Lyubashevsky-Schaffner'18]: Assume extra properties (e.g. statistical soundness) of proof system
Problem: Less efficient, maybe only proof (not PoK)

A Different Conversion

[Unruh'15]

Rough idea: $\pi = \begin{pmatrix} com \\ \{ H(res(ch)) \}_{ch} \end{pmatrix}$

Proof sketch:

- Simulate RO s.t. reduction can efficiently invert
- Invert π on verifier's ch
- Lots of details to make sure A doesn't cheat

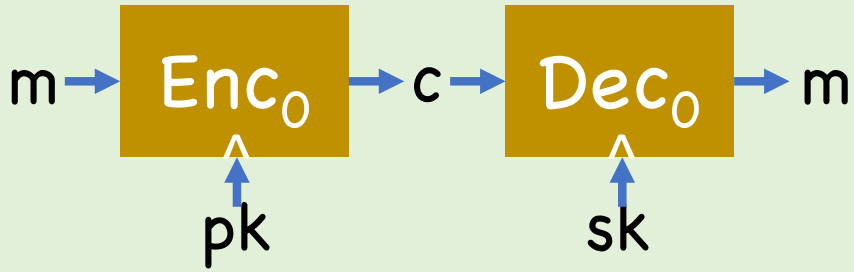
Simulating Invertible Random Oracles

How to simulate H so that reduction can invert?

Recall: already simulating as 2q-wise independent function
→ Can use degree $2q$ polynomial over finite field
→ Invertible by solving polynomial equations

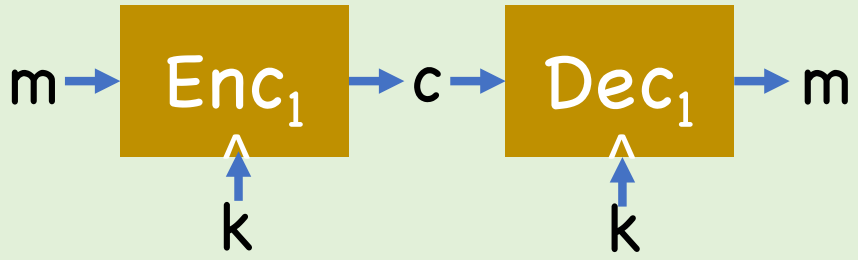
Example: Fujisaki-Okamoto

Building Block: One-*way* PKE



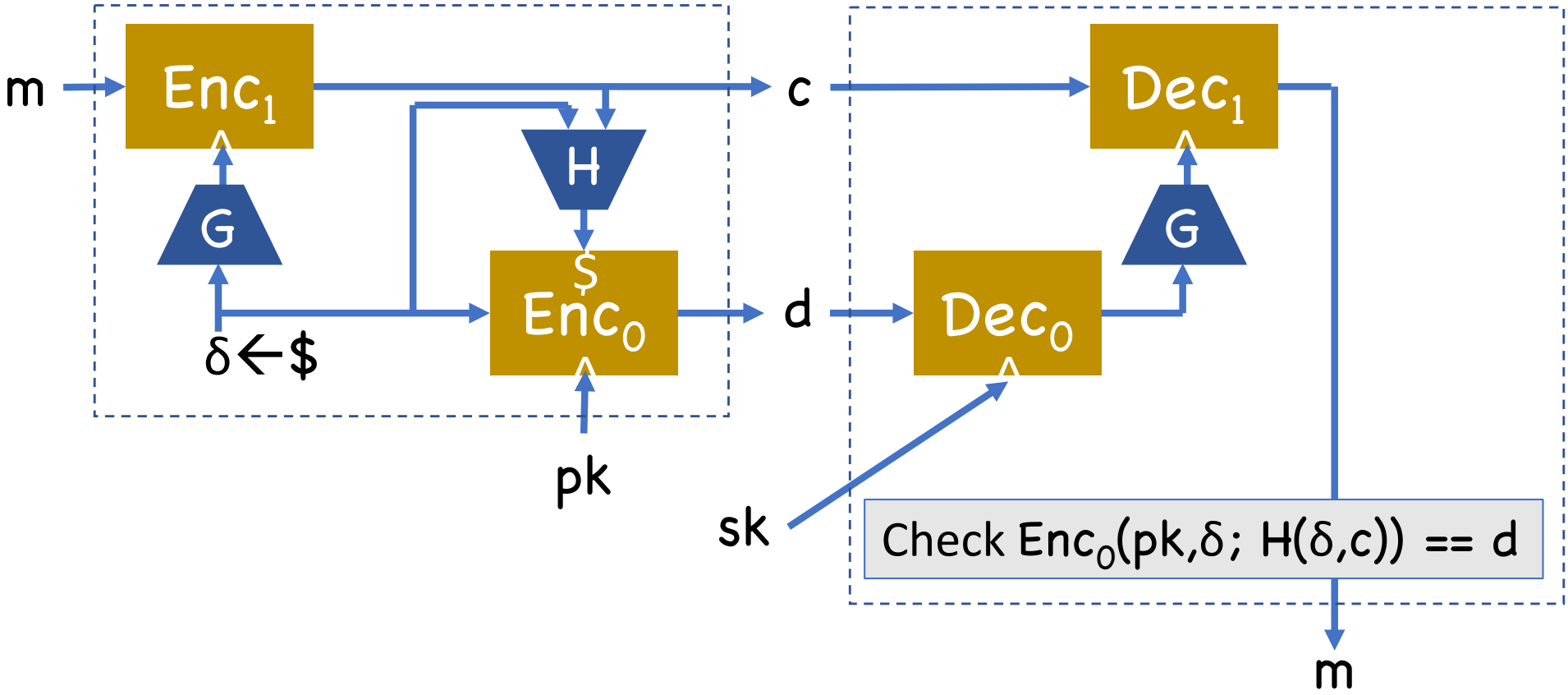
Security: $Enc_0(pk,m)$ one-way

Building Block: One-*time* SKE

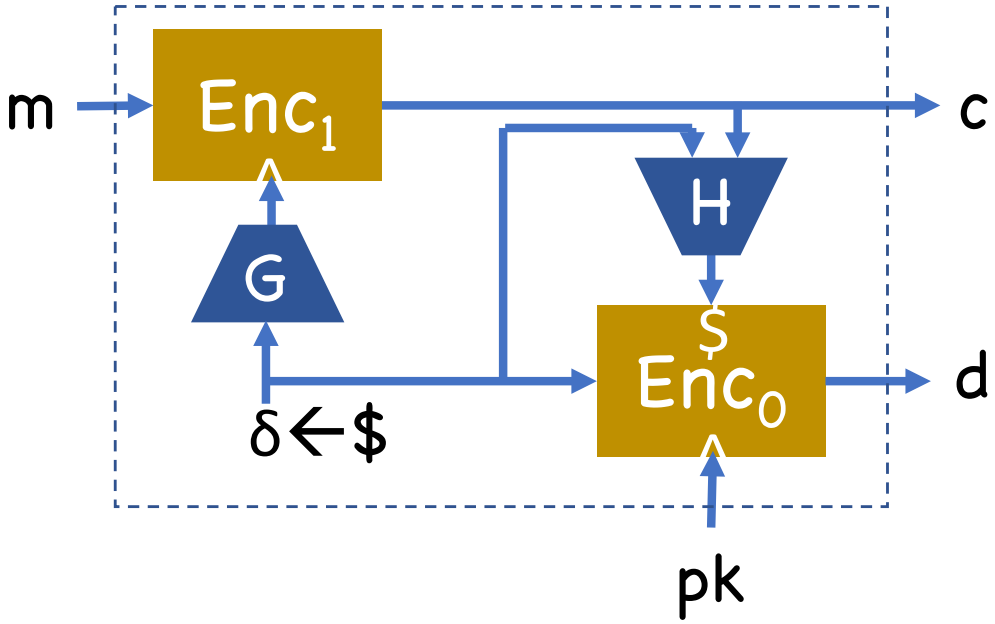


Security: $Enc_1(k,m_0) \approx Enc_1(k,m_1)$,
$H_\infty(Enc(k,m))$ large

Example: Fujisaki-Okamoto



Example: Fujisaki-Okamoto



CCA security intuition:
Only way to obtain valid (c,d) is to have queried H on some (δ,c)

→Look at prior queries to H to answer CCA queries

QROM problem: CPReds can't look at prior RO queries!

Example: Fujisaki-Okamoto

CPRed Impossibility? Open for FO, but I expect one exists

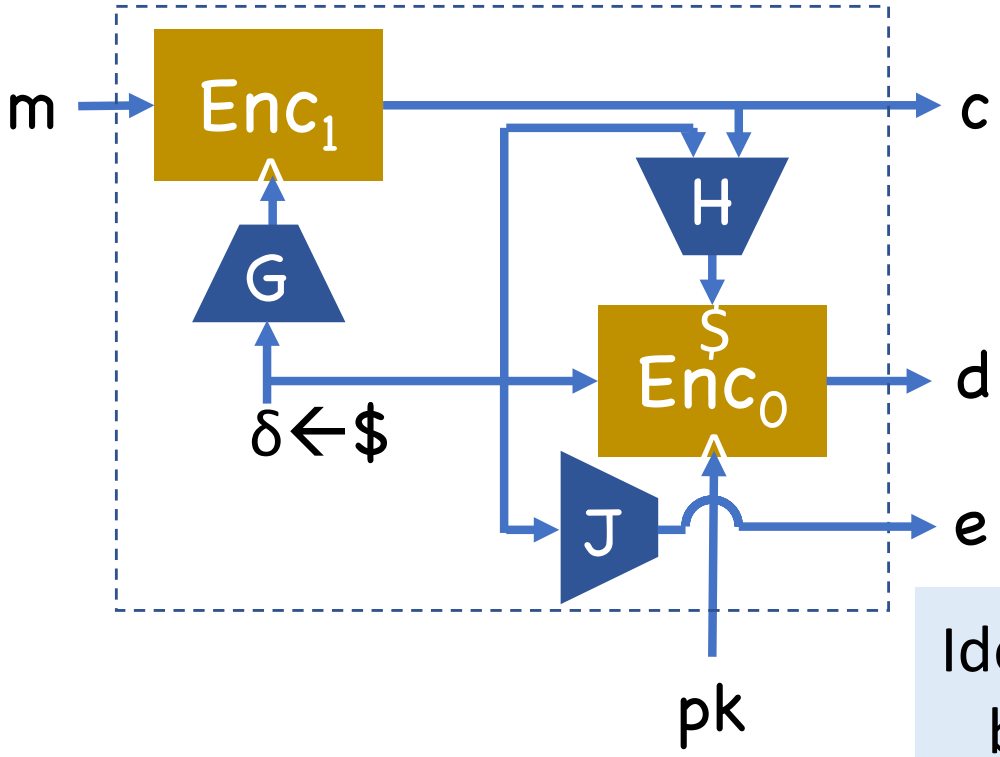
Given (c,d), no way to even tell which RO inputs or outputs used

→ RO seems useless

Impos. of CPReds for OAEP [Zhang-Yu-Feng-Fan-Zhang'19]

A Tweaked Conversion

[Targhi-Unruh'15]

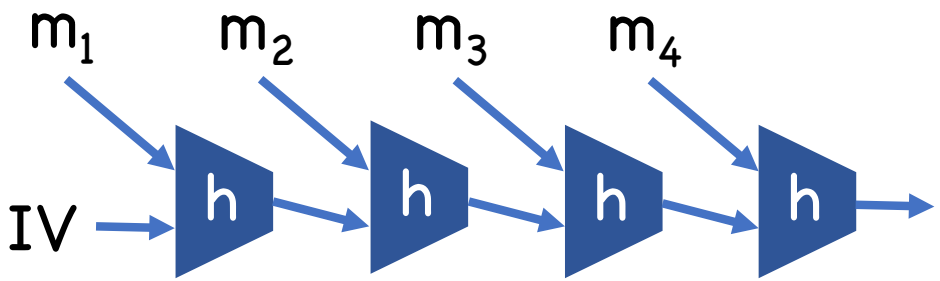


Idea: answer CCA queries by computing $\delta = J^{-1}(e)$

Example: Domain Extension for RO

Most hash functions built from lower-level objects

E.g. Merkle-Damgård (SHA1,SHA2)



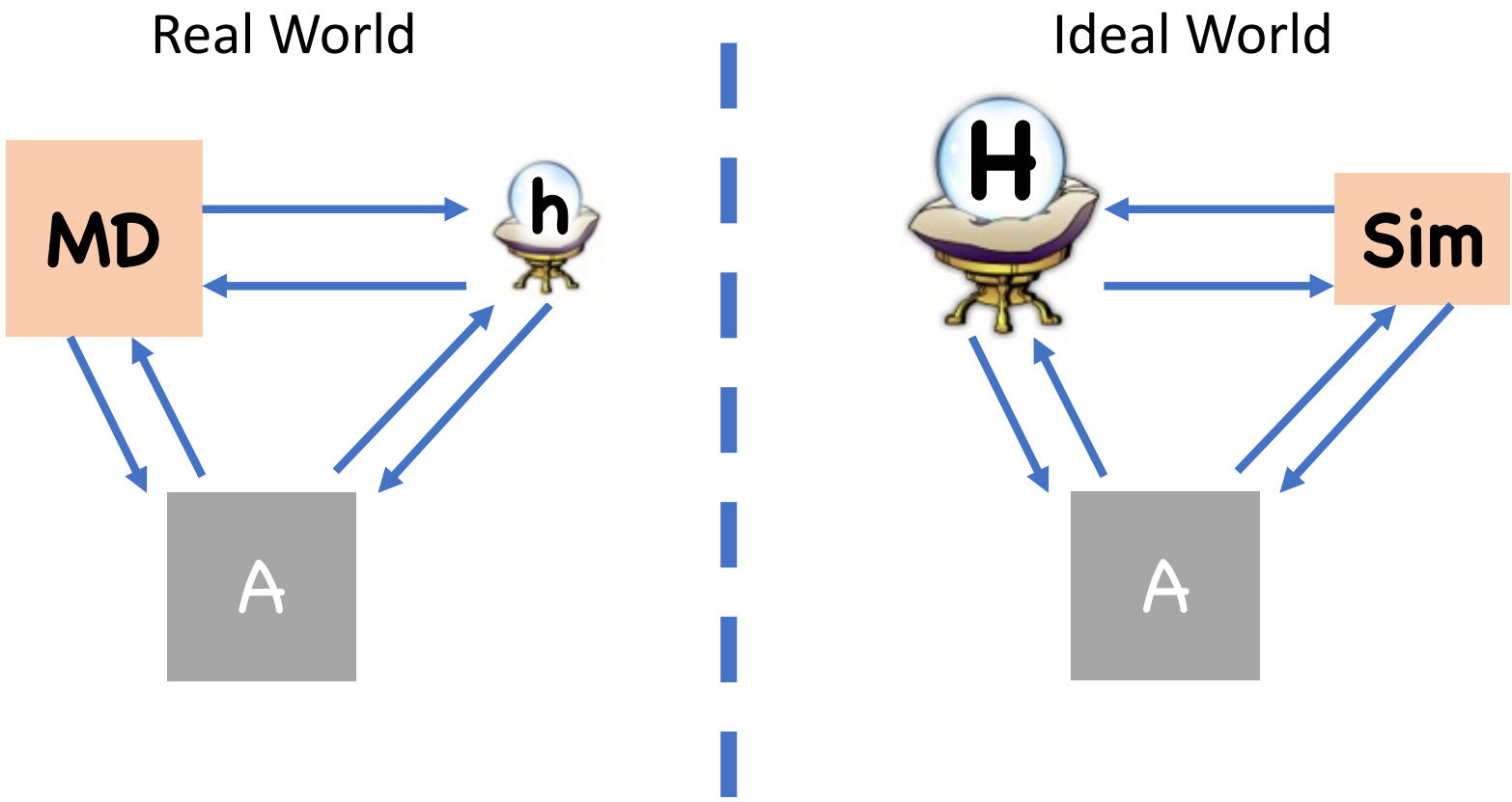
Problem: sometimes structure can be exploited for attack, even if **h** is assumed ideal

Example: Domain Extension for RO

Can we nevertheless justify the “RO Assumption”, despite structure?

Yes(ish): indifferentiability [Maurer-Renner-Holenstein’04]

Indifferentiability



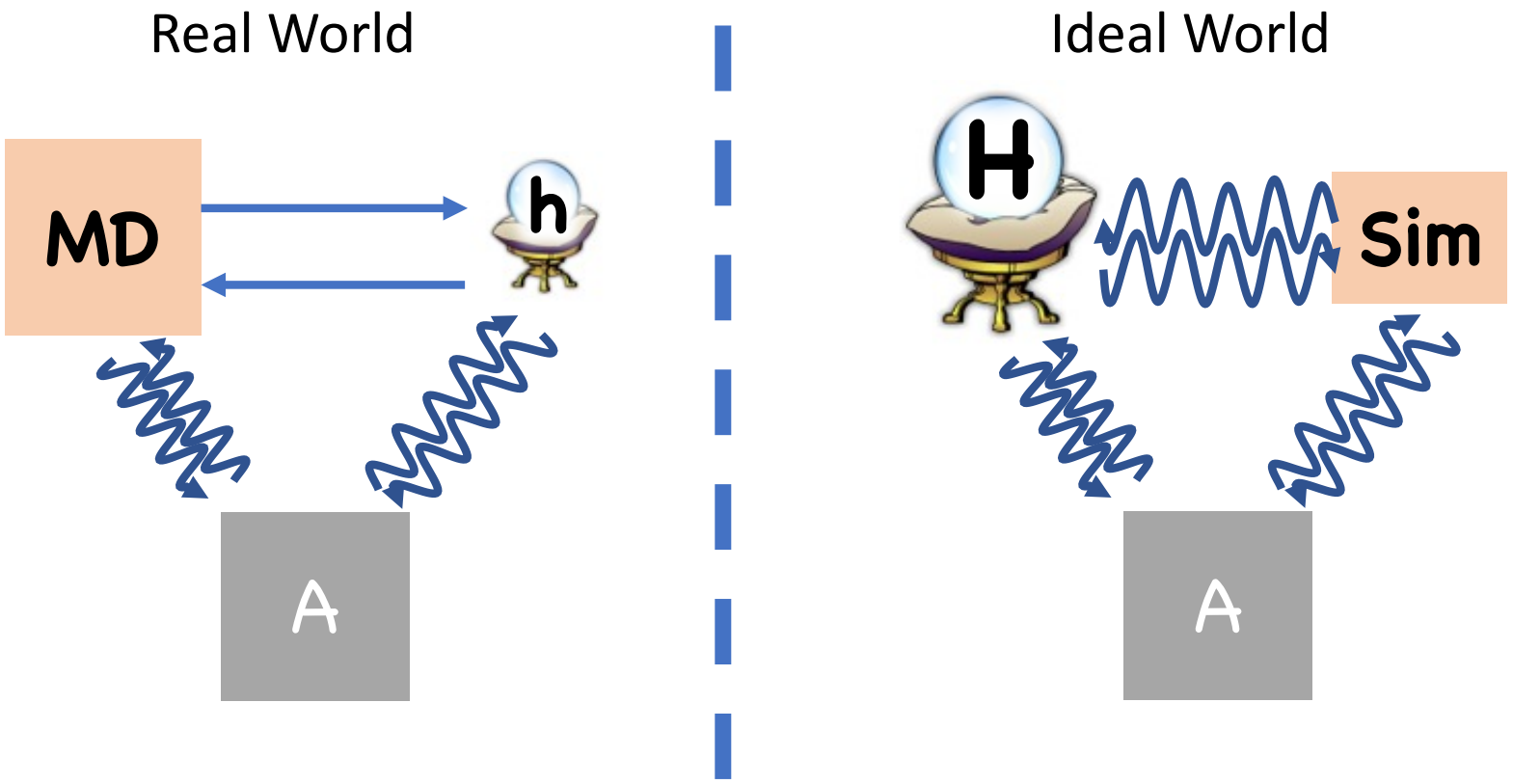
Indifferentiability

Thm [Ristenpart-Shacham-Shrimpton'11]:
Indifferentiability $\Rightarrow$ as good as RO for "single stage games"

Thm [Coron-Dodis-Malinaud-Puniya'05]
MD is classically indifferentiable under appropriate padding

Proof idea: Simulator can figure out when ~~A~~ is trying to evaluate MD by looking at past oracle queries

Quantum Indifferentiability



Quantum Indifferentiability

Fact: No CPRed (stateless simulator) for indifferentiable domain extension, *regardless of construction*

Proof idea:

- Size(truth table of $\mathsf{Sim}^{\mathsf{H}}$) << Size(truth table of H)
- And yet, $\mathsf{Sim}^{\mathsf{H}}$ allows for computing H

  → Compression for random strings

What's next?

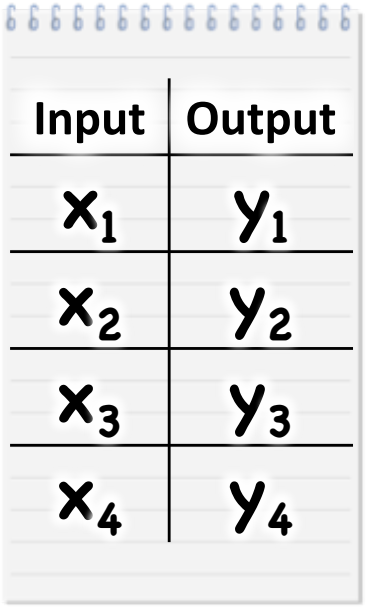
Certain protocols, and even certain tasks, are unprovable under CPReds

Final hour: non-committed programming reductions

Quantum Random Oracle Model, Part 3

Mark Zhandry (Princeton & NTT Research)

Recall: Typical Classical ROM Proof: On-the-fly Simulation



Input	Output
x_1	y_1
x_2	y_2
x_3	y_3
x_4	y_4

Query(x, D):

If **(x,y)∈D**:

Return(y,D)

Else:

y ←$ Y

D′ = D+(x,y)

Return(y,D′)

Recall: Typical Classical ROM Proof: On-the-fly Simulation

Allows us to:

- Know the inputs adversary cares about ✓
- Know the corresponding outputs ✓
- (Adaptively) program the outputs ✓

CPReds?

Allows us to:

- Know the inputs adversary cares about ✘
- Know the corresponding outputs ✘
- (Adaptively) program the outputs ✓ / ✘

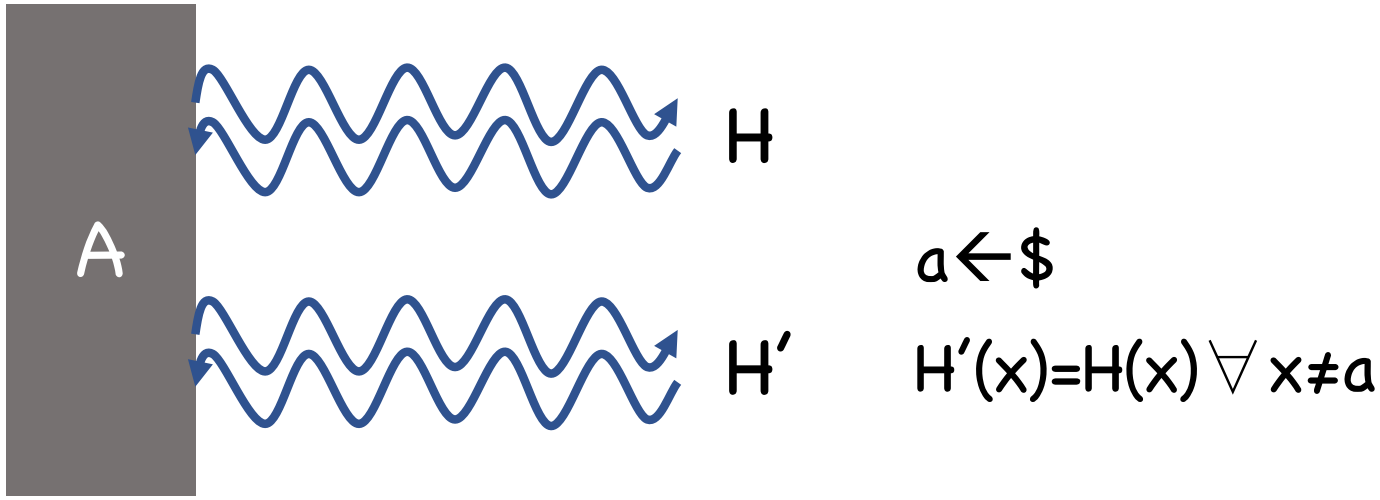
Beyond Committed Programming

How do we change oracle without detection?

Problem: repeated queries?

Problem: distinguishing attack

Random points



Negligible query mass on a, so change undetectable

Used, e.g. for NIZKs [Unruh'16]

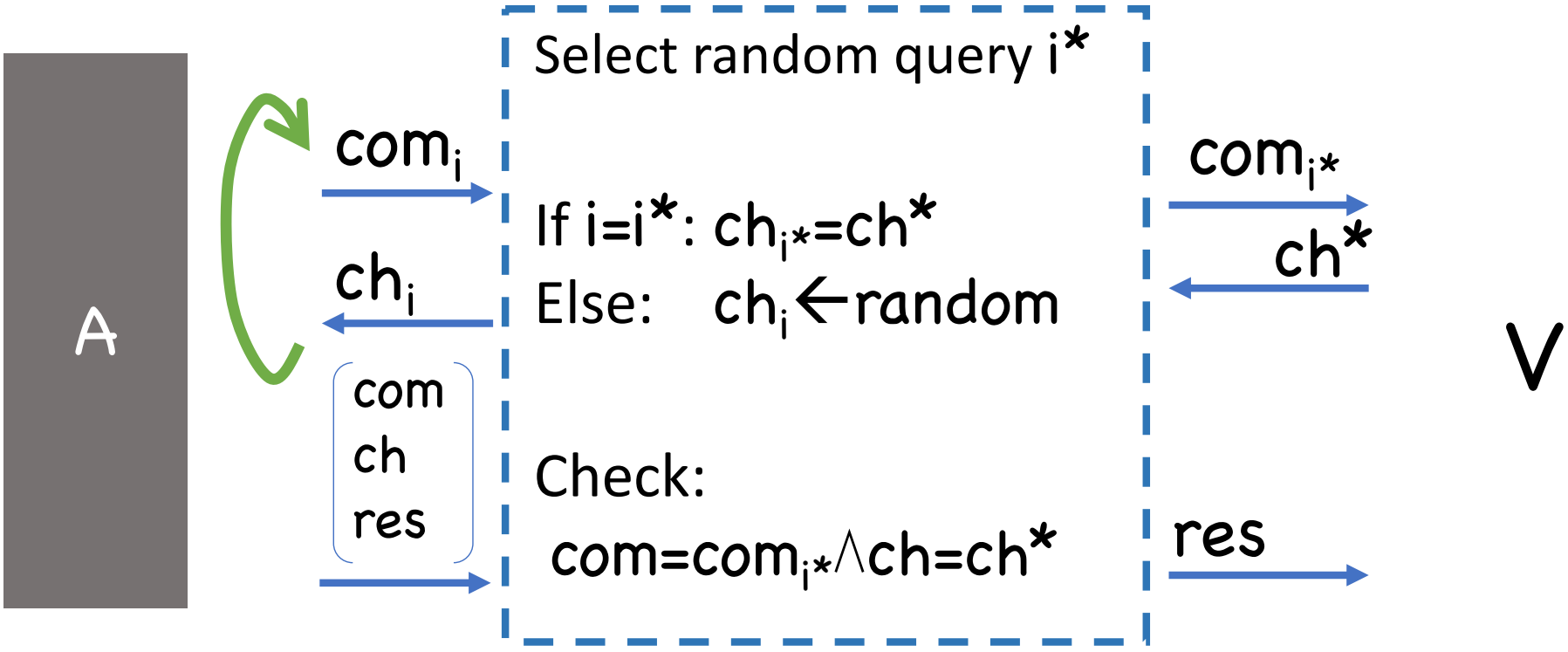
Newer Techniques

Very recently (last 2 years), new techniques have emerged that allow for better programming

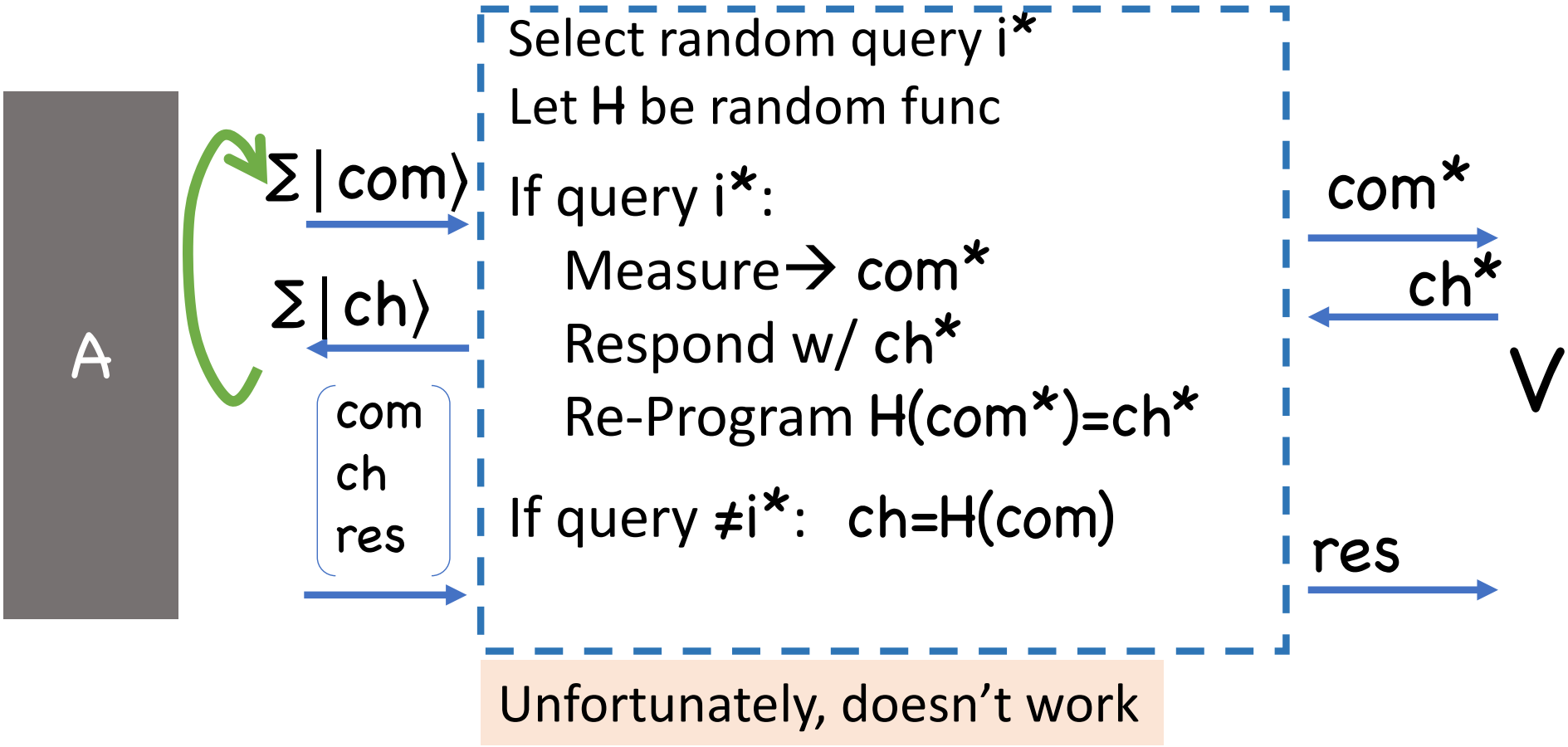
Will highlight some techniques

Fiat Shamir

Recall: Classical Fiat-Shamir Proof

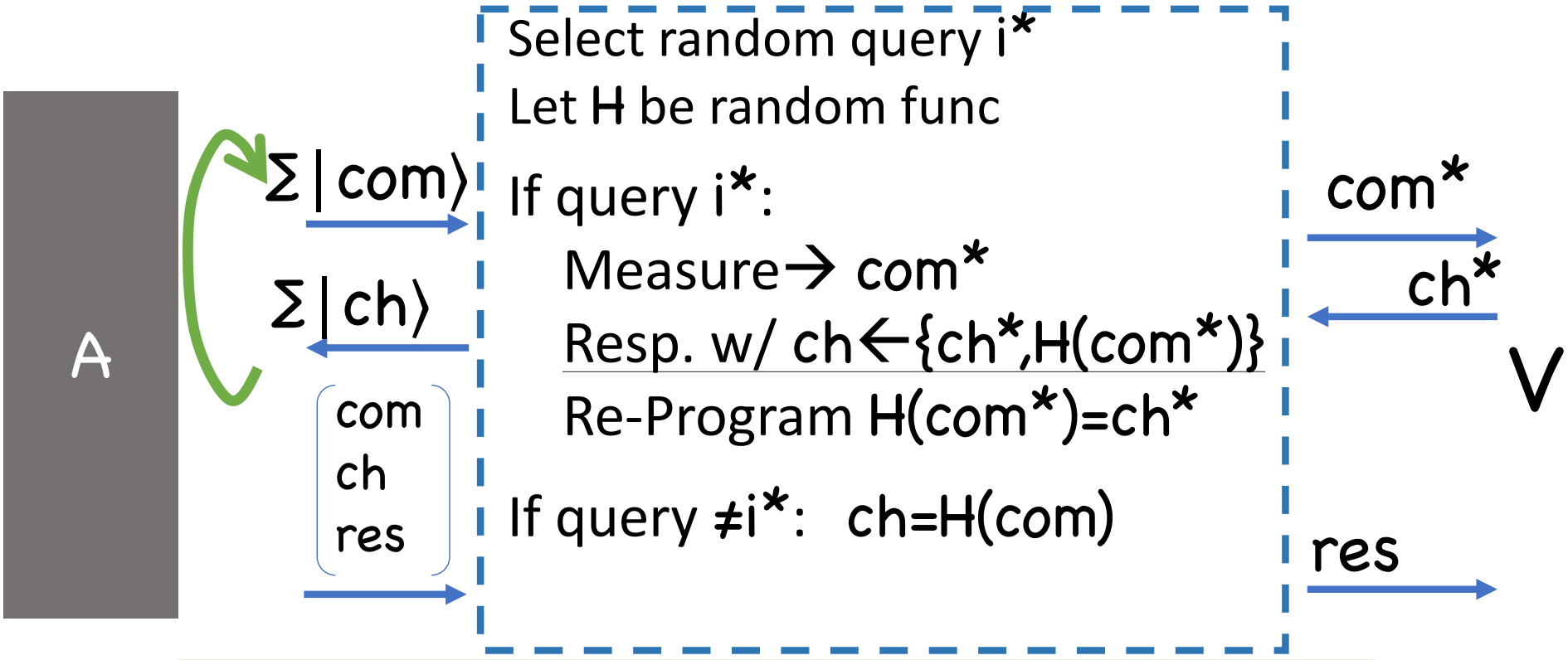


Failed Quantum Fiat-Shamir Proof



Unfortunately, doesn't work

Fixed Quantum Fiat-Shamir Proof



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

Other Applications

[Don-Fehr-Majenz'20]: Multi-round Fiat-Shamir

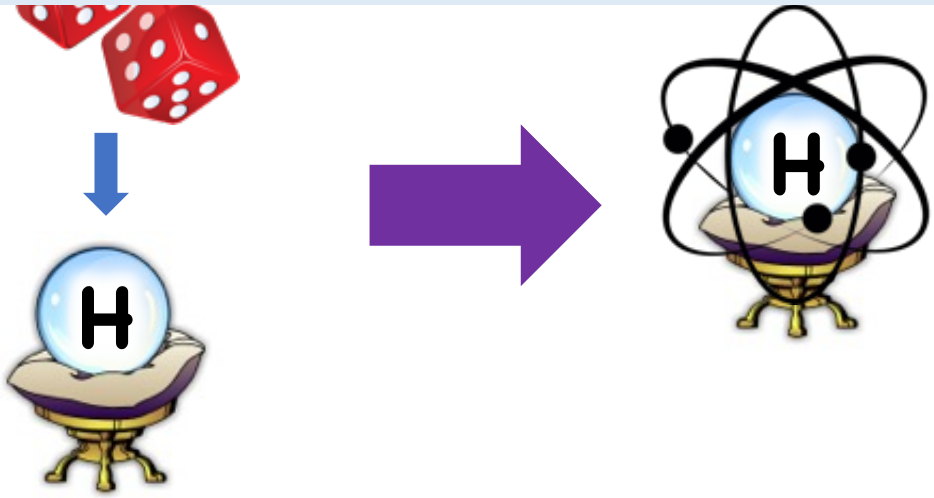
"Lifting Theorem" [Yamakawa-Z'20]:
If *search-type* game, and challenger makes *constant* number of queries to RO, classical ROM proof → QROM proof (w/ polynomial security loss)

Compressed Oracles

Step 1: Quantum-ify (aka Purify)

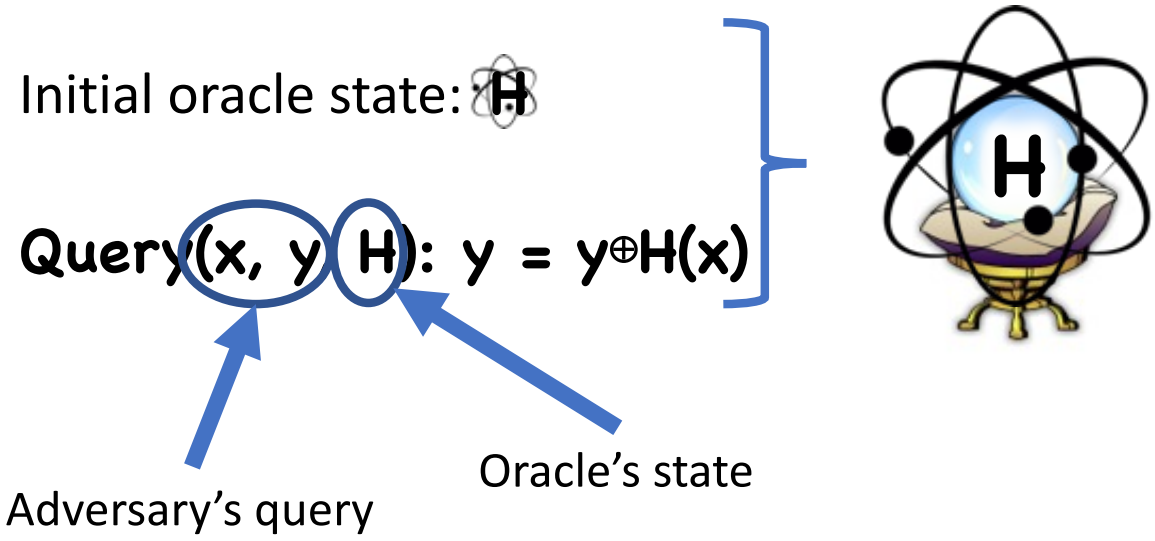
Quantum-ifying (aka purifying) random oracle:

➡ ~~A~~ +  now single quantum system

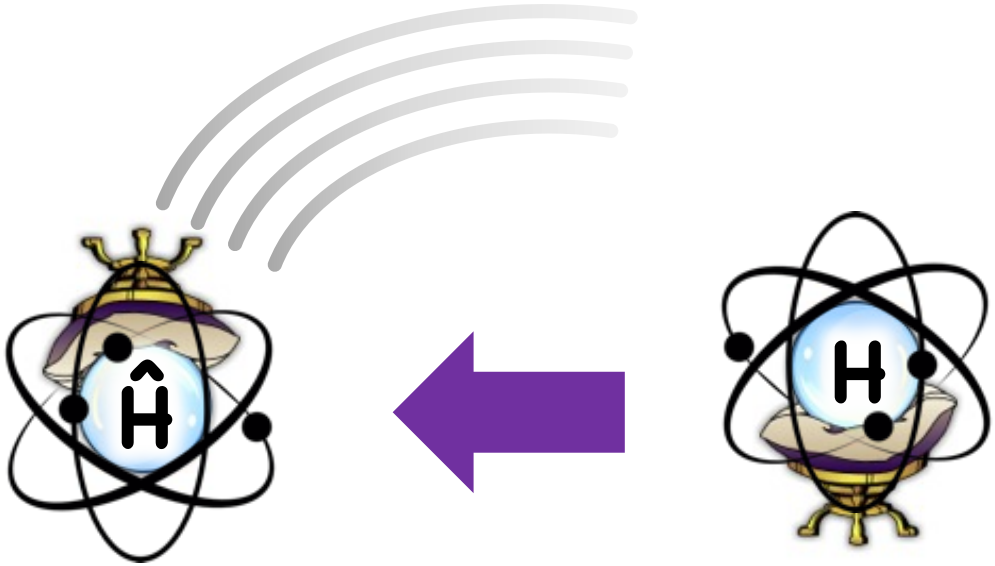


Reminiscent of old impossibilities for unconditional quantum protocols [Lo'97,Lo-Chau'97,Mayers'97,Nayak'99]

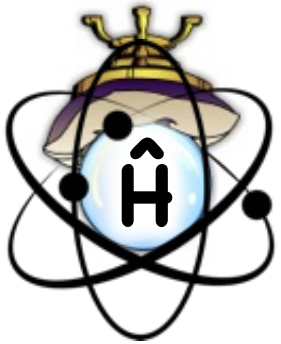
Step 1: Superposition of Oracles



Step 2: Look at Fourier Domain



Step 2: Look at Fourier Domain

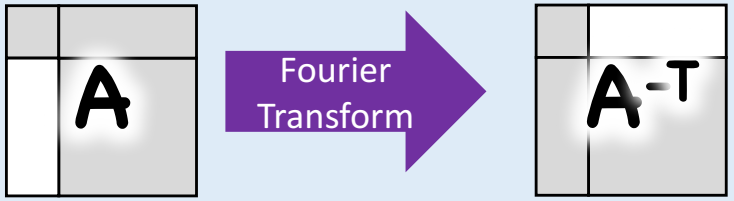


Initial oracle state: $Z(x) = 0$

Query$(x, y, \hat{H})$: $\hat{H} = \hat{H} \oplus P_{x,y}$

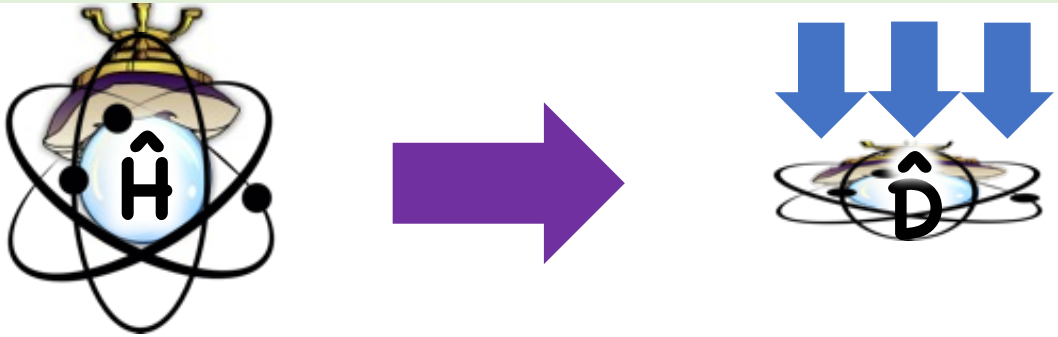
$$P_{x,y}(x') = \begin{cases} y \text{ if } x=x' \\ 0 \text{ else} \end{cases}$$

Proof:



Step 3: Compress

Observation:
After **q** queries, $\hat{H}$ is non-zero on at most **q** points



Step 3: Compress

Initial oracle state: **{}**

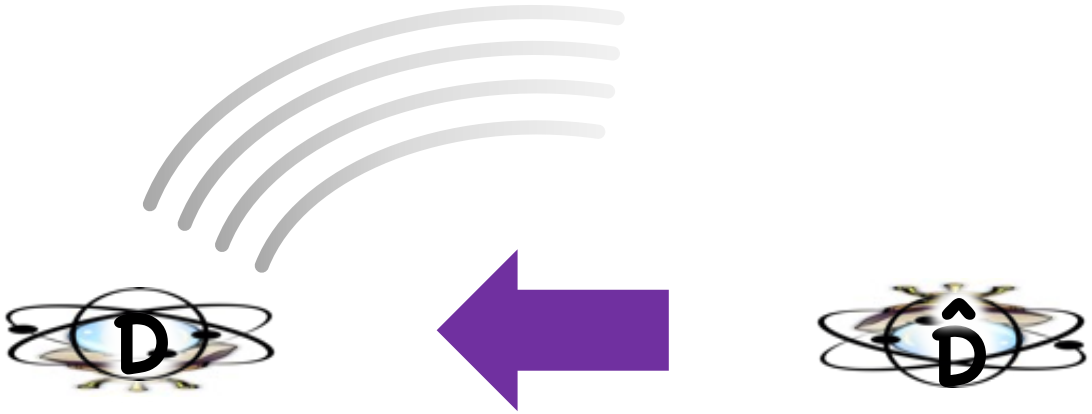
Query(x, y, $\hat{D}$):

(1) If $\nexists$**(x,y′)**$\in\hat{D}$**:** $\hat{D} = \hat{D}+(x,0)$

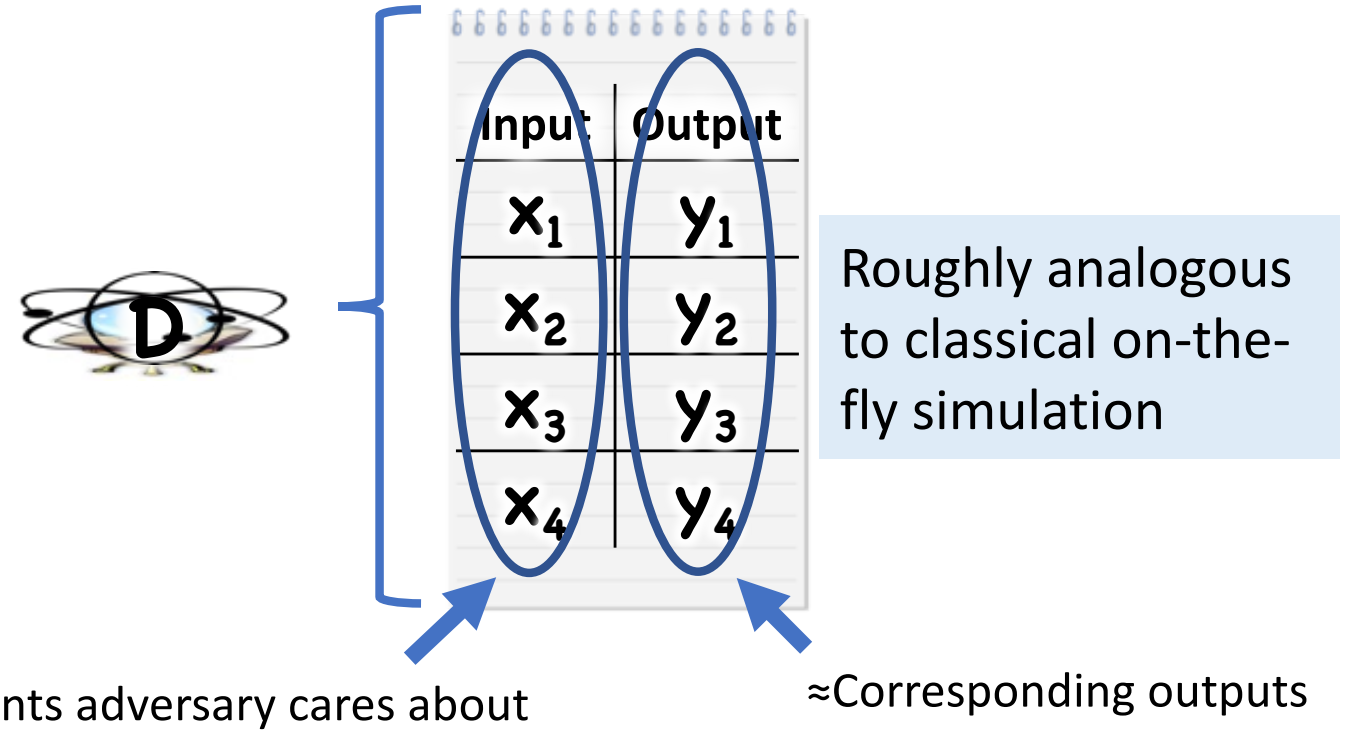
(2) Replace **(x,y′)**$\in\hat{D}$
with **(x,y′⊕y)**

(3) If **(x,0)**$\in\hat{D}$**:** remove it

Step 4: Revert back to Primal Domain



Step 4: Revert back to Primal Domain



Compressed Oracles

Allows us to:

- Know the inputs adversary cares about? ✓
- Know the corresponding outputs? ✓
- (Adaptively) program the outputs? ✓ (with some work)

So, what happened?

Observer Effect:
Learning anything about quantum system disturbs it

Motivation for CPReds:

answers obliviously,
so no disturbance

Reduction must answer obliviously, too?

Beyond CPReds:

A learns about 

through queries

gets disturbed

Compressed oracles decode such disturbance

Caveats

Outputs in database **≠0** in Fourier domain

➡ **y** values aren't exactly query outputs

Examining **x,y** values perturbs state

➡ Still must be careful about how we use them

But, still good enough for many applications…

Some Applications

[Z'19]: Indifferentiability of MD

[Alagic-Majenz-Russell-Song'18]: Quantum-secure signature separation

[Liu-Z'19a]: Tight bounds for multi-collision problem

[Liu-Z'19b]: Fiat-Shamir

([Don-Fehr-Majenz-Schaffner'19]: direct proof)

[Hosoyamada-Iwata'19]: 4-round Luby-Rackoff

[Unruh'21]: Collision resistance of Sponge

[Chiesa-Manohar-Spooner'19]: zk-SNARKs

[Bindel-Hamburg-Hülsing-Persichetti'19]: Tighter CCA security proofs

Summary

- Now have numerous techniques for proving QROM security
- Many schemes of interest now have QROM proof
- Major lingering issues:
  - Tightness of reductions
  - Indifferentiability (Sponge, ideal ciphers from RO)
  - Constant-query lifting theorem for indistinguishability?
  - Still various missing pieces