Quantum Security and Fiat-Shamir for Cryptographic Protocols

Fermi Ma

Cryptographic Protocols

A cryptographic protocol is an **interaction** between parties that achieves:

Cryptographic Protocols

A cryptographic protocol is an **interaction** between parties that achieves:

1) Functionality

Ex: verify computation, compute on private inputs

Cryptographic Protocols

A cryptographic protocol is an **interaction** between parties that achieves:

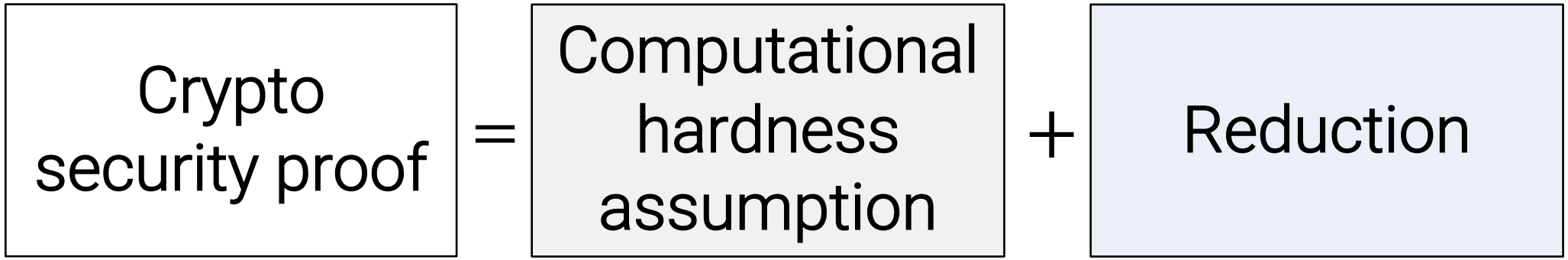
1) **Functionality**

Ex: verify computation, compute on private inputs

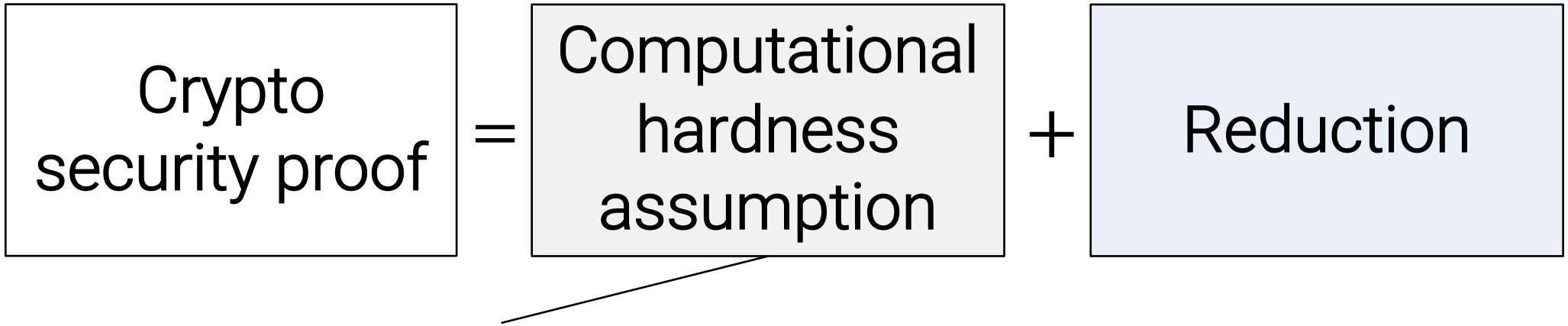
2) **Security against adversarial behavior**

Ex: can't fool verifier, can't learn other party's input

How do we prove security?

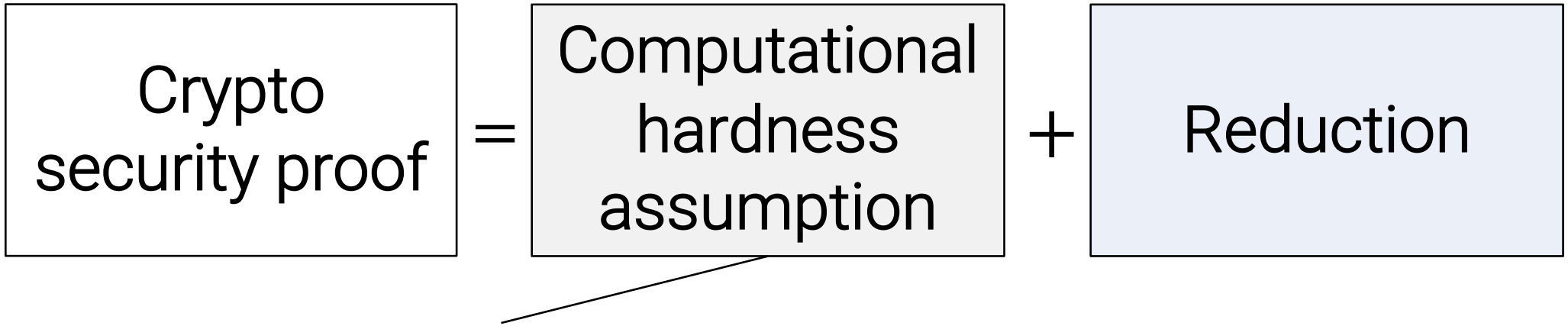


How do we prove security?



Ex: invert one-way function, factoring, discrete log, lattice problems, etc.

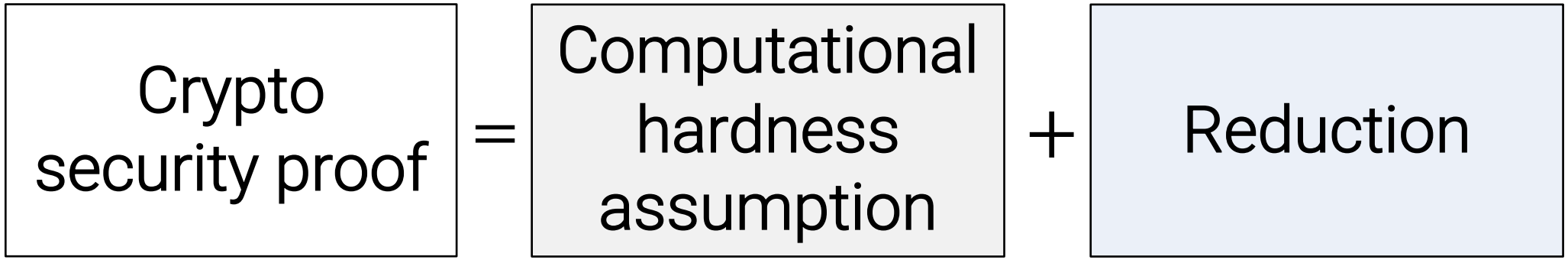
How do we prove security?



Ex: invert one-way function, factoring, discrete log, lattice problems, etc.

Any ***efficient*** attack on the protocol
→ Break underlying hardness assumption

How do we prove security?



Many amazing results based on this formula:

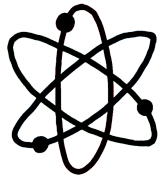
- Zero-knowledge proofs [GMR84]
- Secure multi-party computation [Yao86, GMW86]
- Succinct arguments [Kilian92]

But even in settings where secure protocols are known, key challenges remain.

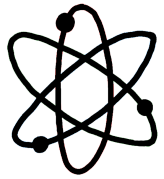
Challenge 1: Quantum Computers

Most security proofs consider classical attackers.

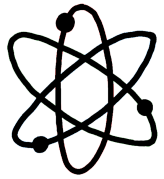
Does security still hold if a quantum computer is built?



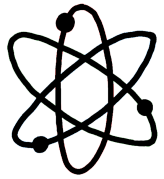
Quantum Computers



- Use quantum physics to perform computation



Quantum Computers



- Use quantum physics to perform computation
- Likely more powerful than classical computers (e.g., Shor's algorithm enables factoring)

Quantum Computers

- Use quantum physics to perform computation
- Likely more powerful than classical computers (e.g., Shor's algorithm enables factoring)
- Not far away?


NEWS · 23 OCTOBER 2019

Hello quantum world! Google publishes landmark quantum supremacy claim

[Nature 2019]


THE WALL STREET JOURNAL.

English Edition | Print Edition | Video | Podcasts | Latest Headlines

World U.S. Politics Economy Business Tech Markets Opinion Life & Arts Real Estate

CIO JOURNAL

Google Aims for Commercial-Grade Quantum Computer by 2029

[WSJ 2021]

Goal 1:

Understand what happens to crypto if/when a quantum computer is built.

Challenge 2: Removing Interaction

Protocols typically use *interaction*, but interaction is unwieldy in practice.

Through heuristics such as Fiat-Shamir, some protocols can be made *non-interactive*.

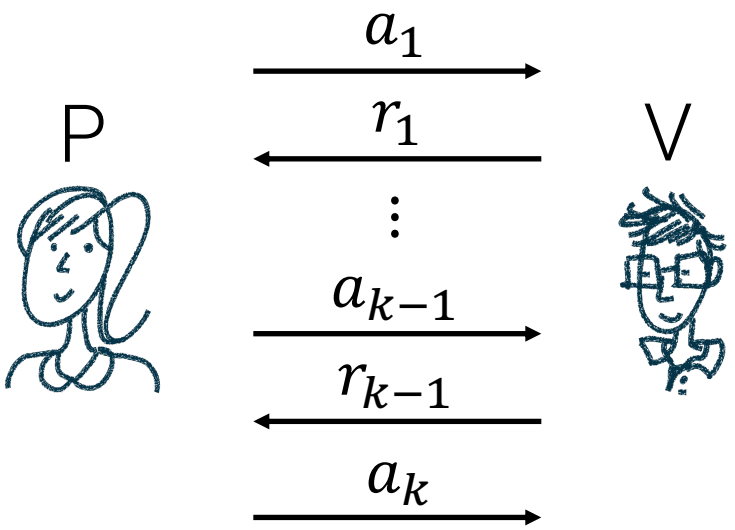
Is this secure?

Fiat-Shamir Heuristic [FS86]

Magical compiler that removes interaction from public-coin interactive protocols.

Fiat-Shamir Heuristic [FS86]

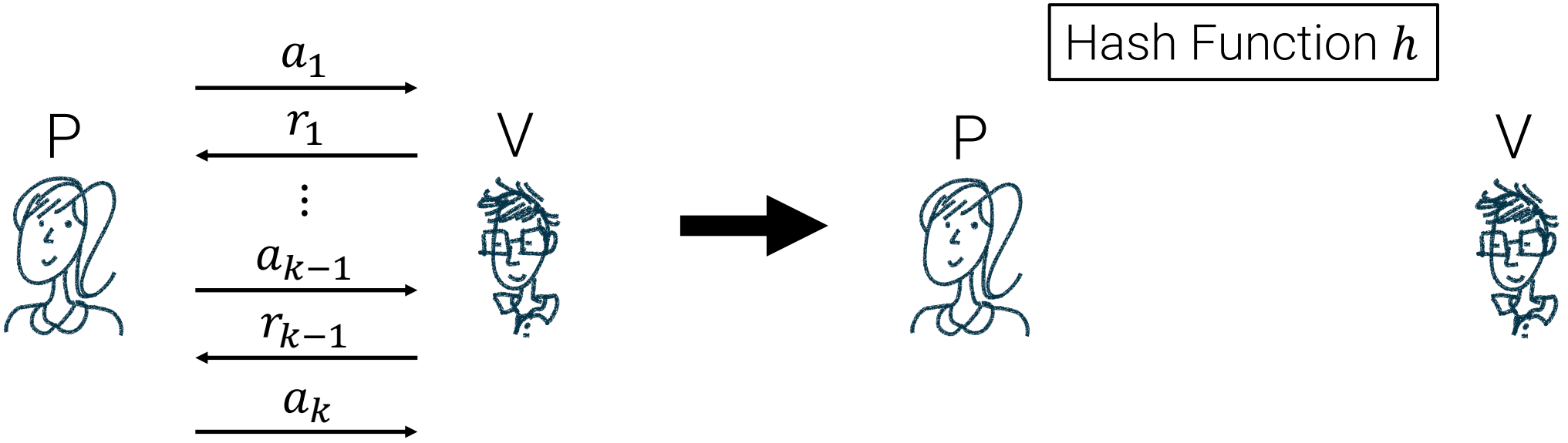
Magical compiler that removes interaction from public-coin interactive protocols.



Public-Coin
Interactive Protocol

Fiat-Shamir Heuristic [FS86]

Magical compiler that removes interaction from public-coin interactive protocols.

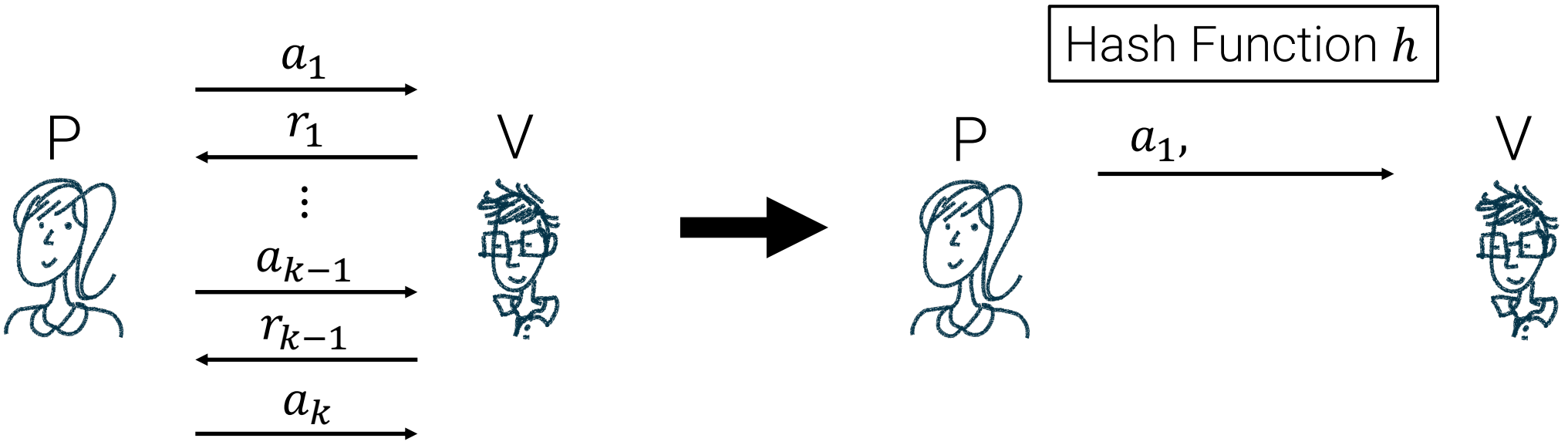


Public-Coin Interactive Protocol

Non-Interactive Argument

Fiat-Shamir Heuristic [FS86]

Magical compiler that removes interaction from public-coin interactive protocols.

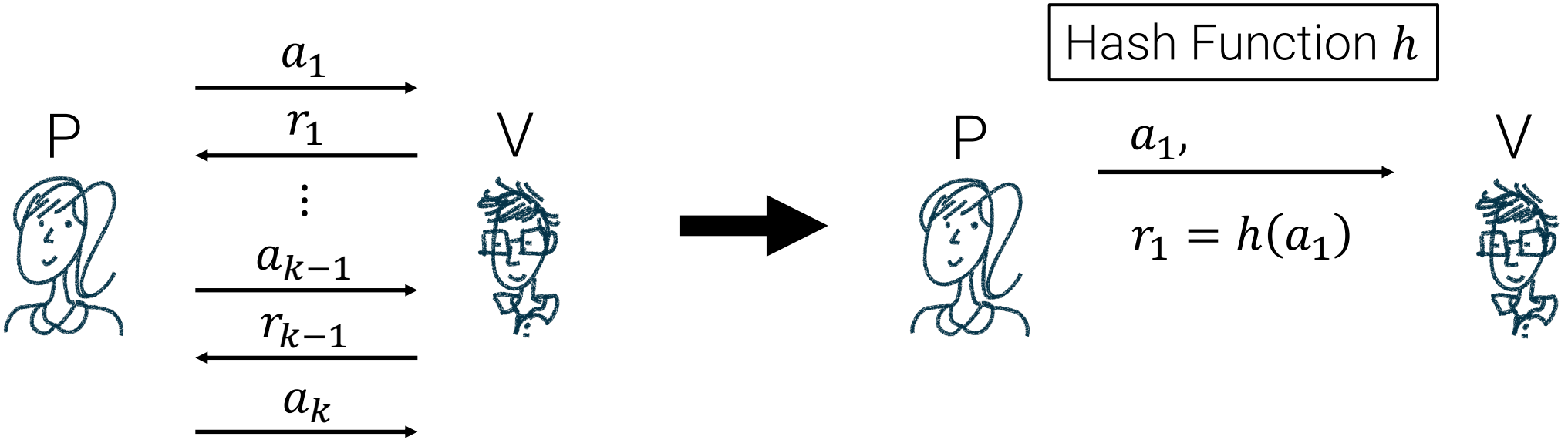


Public-Coin Interactive Protocol

Non-Interactive Argument

Fiat-Shamir Heuristic [FS86]

Magical compiler that removes interaction from public-coin interactive protocols.

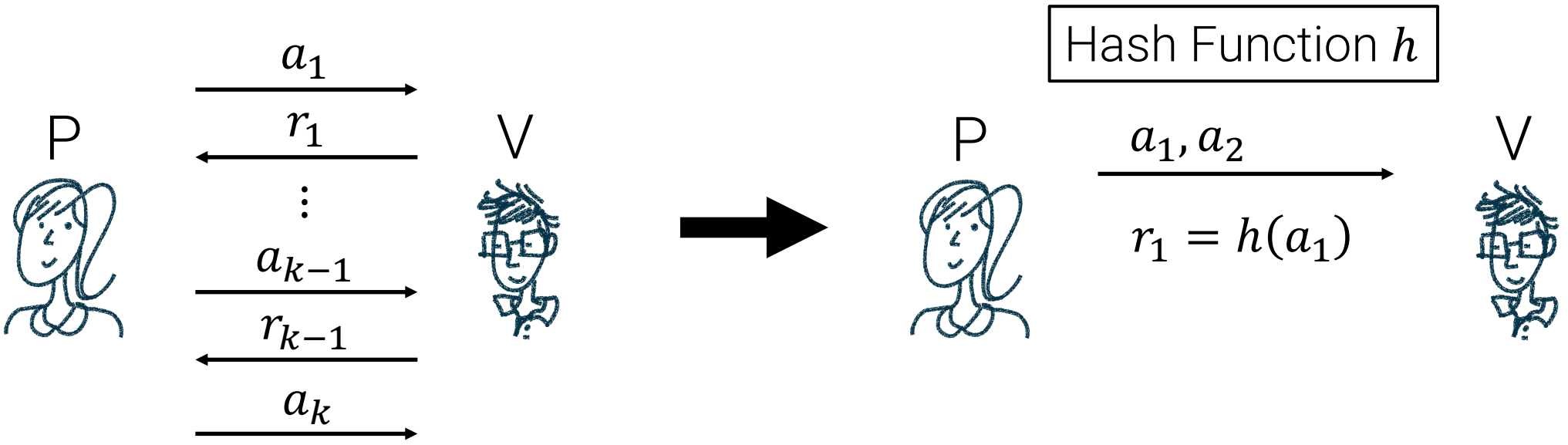


Public-Coin Interactive Protocol

Non-Interactive Argument

Fiat-Shamir Heuristic [FS86]

Magical compiler that removes interaction from public-coin interactive protocols.

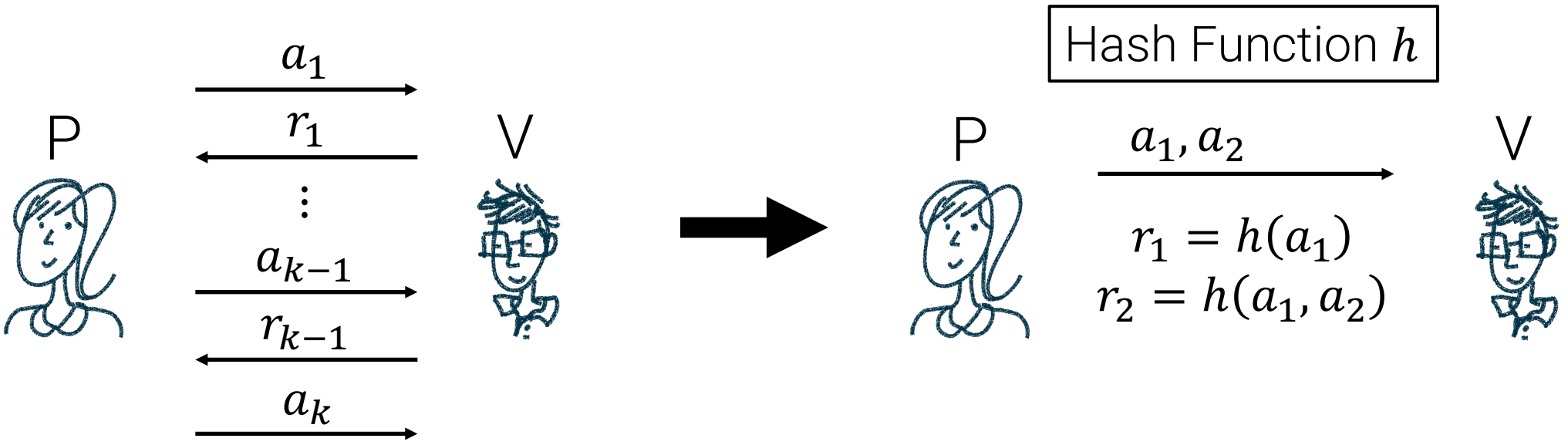


Public-Coin Interactive Protocol

Non-Interactive Argument

Fiat-Shamir Heuristic [FS86]

Magical compiler that removes interaction from public-coin interactive protocols.

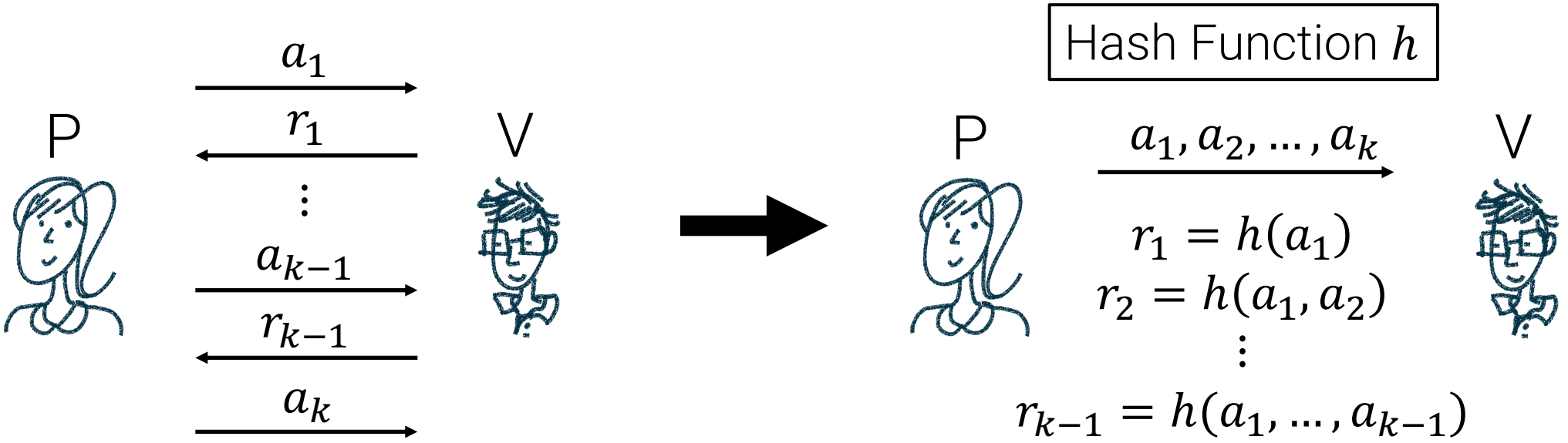


Public-Coin Interactive Protocol

Non-Interactive Argument

Fiat-Shamir Heuristic [FS86]

Magical compiler that removes interaction from public-coin interactive protocols.

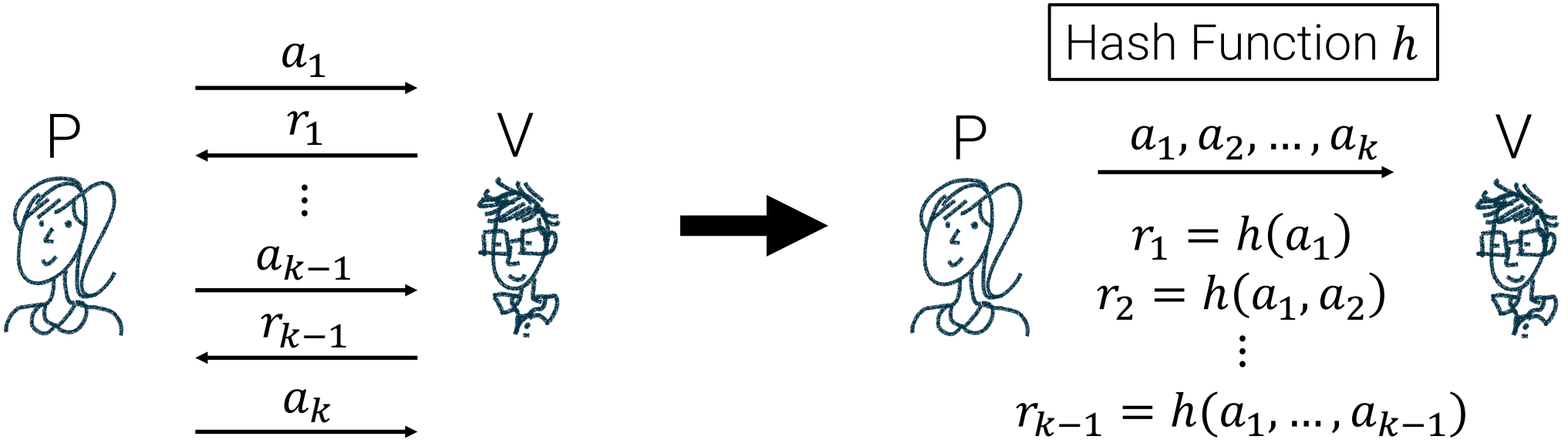


Public-Coin Interactive Protocol

Non-Interactive Argument

Fruitful approach:

Construct ***interactive*** protocol for some functionality (e.g., identification, verifiable computation), apply Fiat-Shamir.

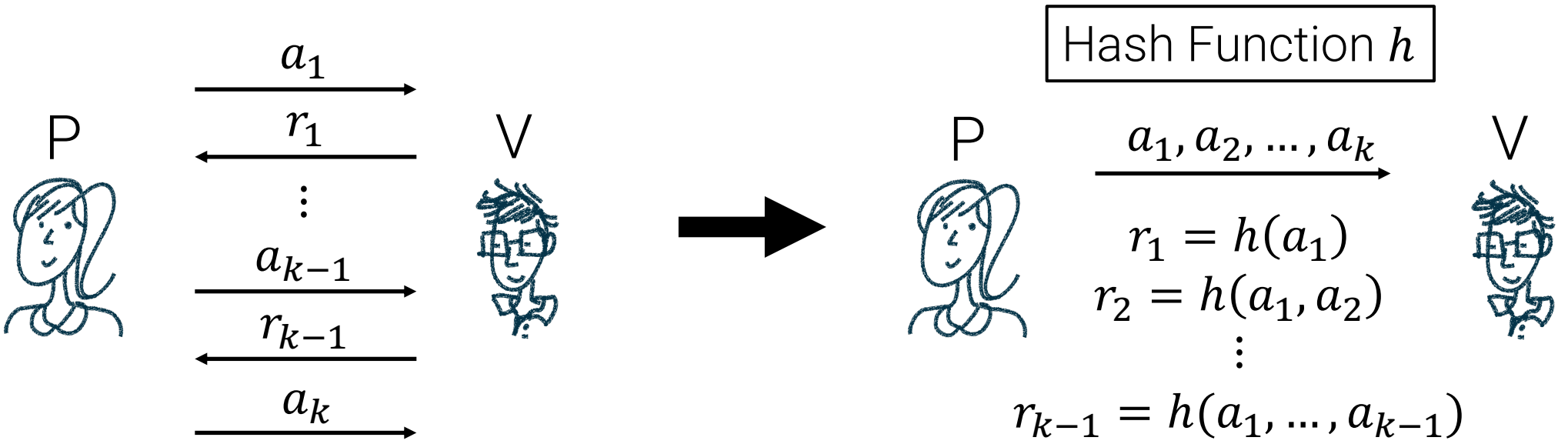


Public-Coin Interactive Protocol

Non-Interactive Argument

Important Caveat:

Not immediately clear if soundness is preserved!



Public-Coin Interactive Protocol

Non-Interactive Argument

Goal 2:

Understand when the Fiat-Shamir transform preserves soundness.

In sum, this thesis addresses two challenges:

1) Our understanding of what constitutes an **efficient algorithm** has changed.

2) Our demands for how protocols **will be used** have changed.

Thesis Contributions

Part 1: Quantum

[CMSZ21]: Post-quantum succinct arguments via rewinding.

Thesis Contributions

Part 1: Quantum

[CMSZ21]: Post-quantum succinct arguments via rewinding.

[BCKM21]: One-way functions imply secure multi-party computation (MPC) for quantum users.

Thesis Contributions

Part 1: Quantum

[CMSZ21]: Post-quantum succinct arguments via rewinding.

[BCKM21]: One-way functions imply secure multi-party computation (MPC) for quantum users.

Part 2: Fiat-Shamir

[BBHMR19]: Barriers to provably-secure succinct non-interactive arguments from Fiat-Shamir.

Thesis Contributions

Part 1: Quantum

[CMSZ21]: Post-quantum succinct arguments via rewinding.

[BCKM21]: One-way functions imply secure multi-party computation (MPC) for quantum users.

Part 2: Fiat-Shamir

[BBHMR19]: Barriers to provably-secure succinct non-interactive arguments from Fiat-Shamir.

[CLMQ21]: Investigate whether Fiat-Shamir hash function must be “cryptographic"

Thesis Contributions

Part 1: Quantum

This talk

[CMSZ21]: Post-quantum succinct arguments via rewinding.

[BCKM21]: One-way functions imply secure multi-party computation (MPC) for quantum users.

Part 2: Fiat-Shamir

[BBHMR19]: Barriers to provably-secure succinct non-interactive arguments from Fiat-Shamir.

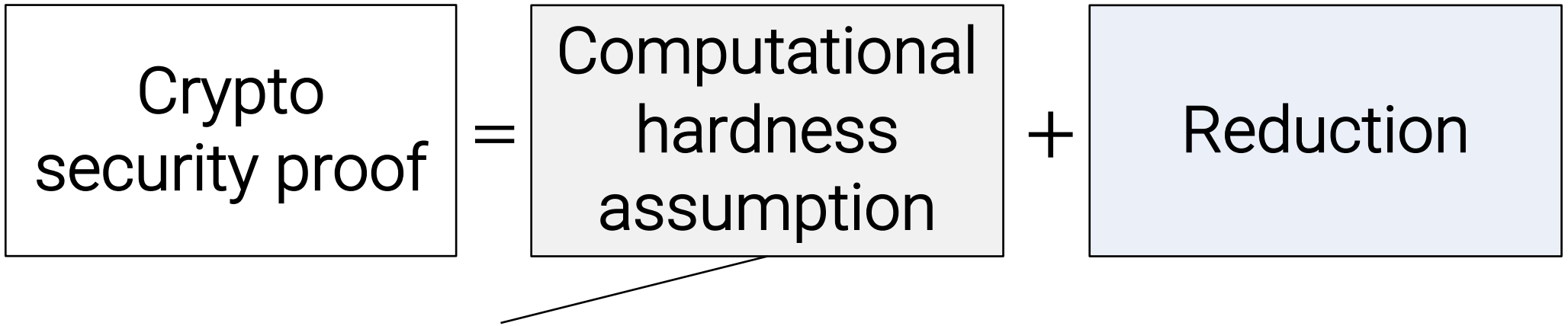
[CLMQ21]: Investigate whether Fiat-Shamir hash function must be “cryptographic"

Up next:

[CMSZ21]: Post-quantum succinct arguments via rewinding.

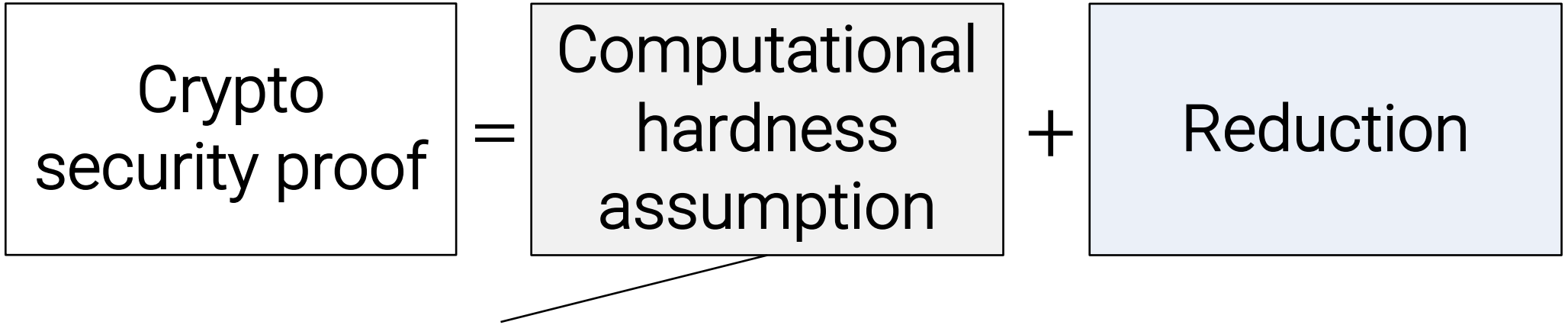
How Will Quantum Computers Impact Crypto?

How Will Quantum Computers Impact Crypto?



Ex: invert one-way function, factoring, discrete log, lattice problems, etc.

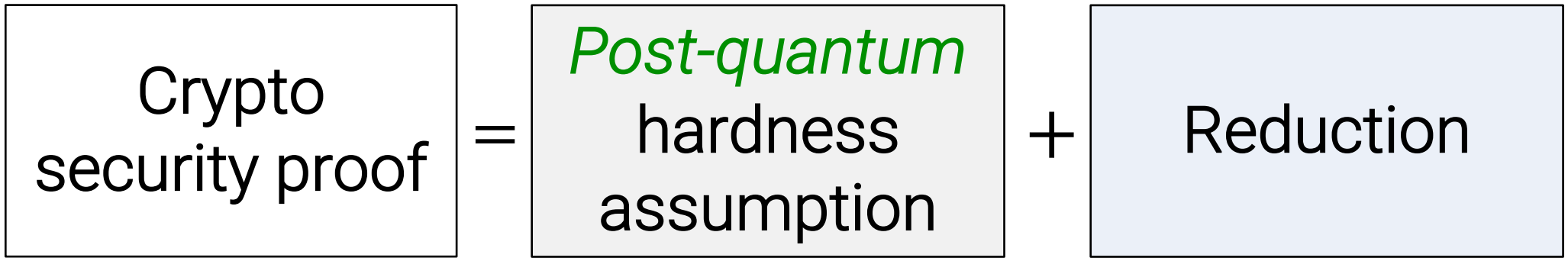
How Will Quantum Computers Impact Crypto?



Ex: invert one-way function, ~~factoring~~, ~~discrete log~~, lattice problems, etc.

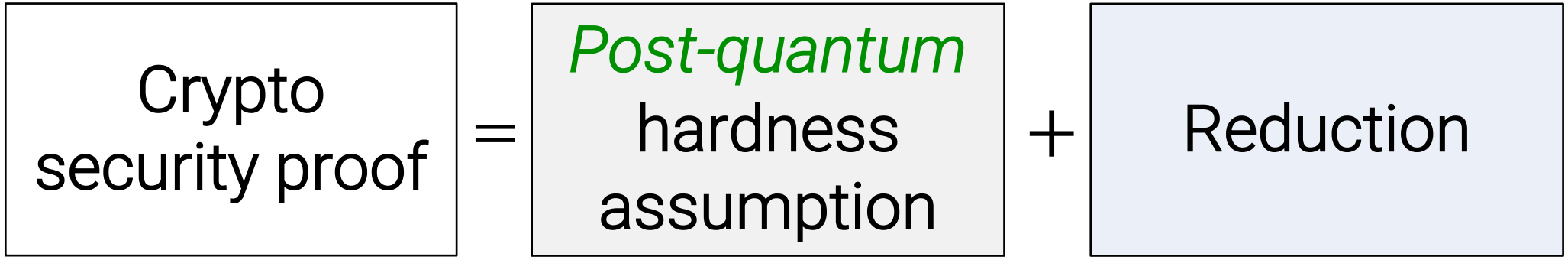
Immediate consequence: Shor's algorithm breaks commonly used assumptions

Post-Quantum Cryptography



Minimum requirement for *post-quantum* crypto:
hard problem should resist quantum attacks

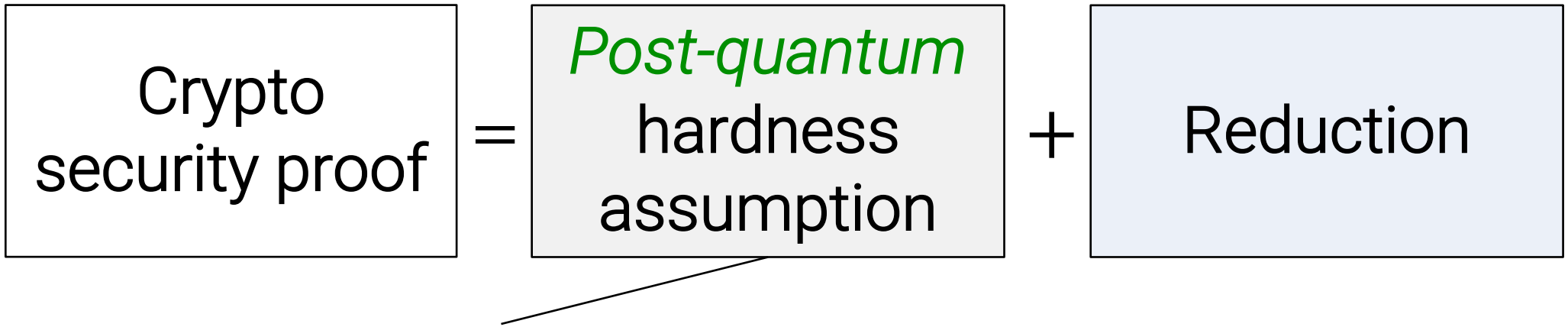
Post-Quantum Cryptography



Minimum requirement for *post-quantum* crypto:
hard problem should resist quantum attacks

Fortunately, we have many candidate hard problems.

Post-Quantum Cryptography

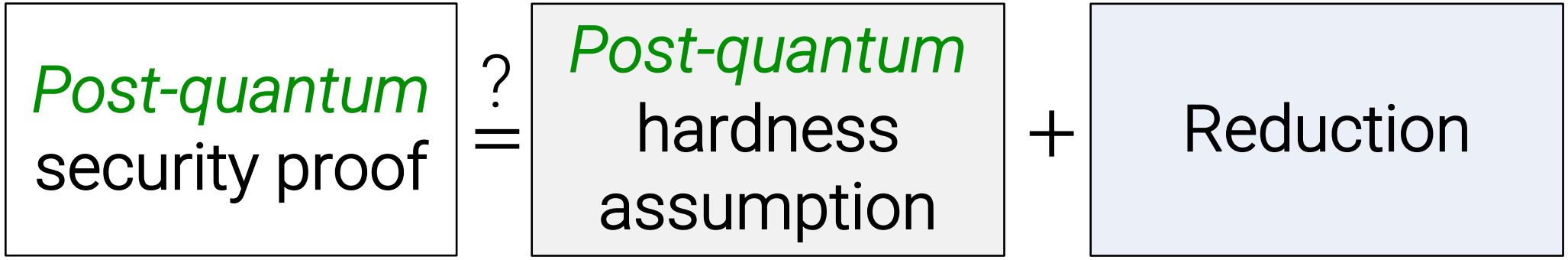


Ex: post-quantum one-way function, lattice problems, etc.

Minimum requirement for *post-quantum* crypto:
hard problem should resist quantum attacks

Fortunately, we have many candidate hard problems.

Post-Quantum Cryptography



Important point:
the *security reduction* must be *quantum-compatible*!

Post-Quantum Cryptography

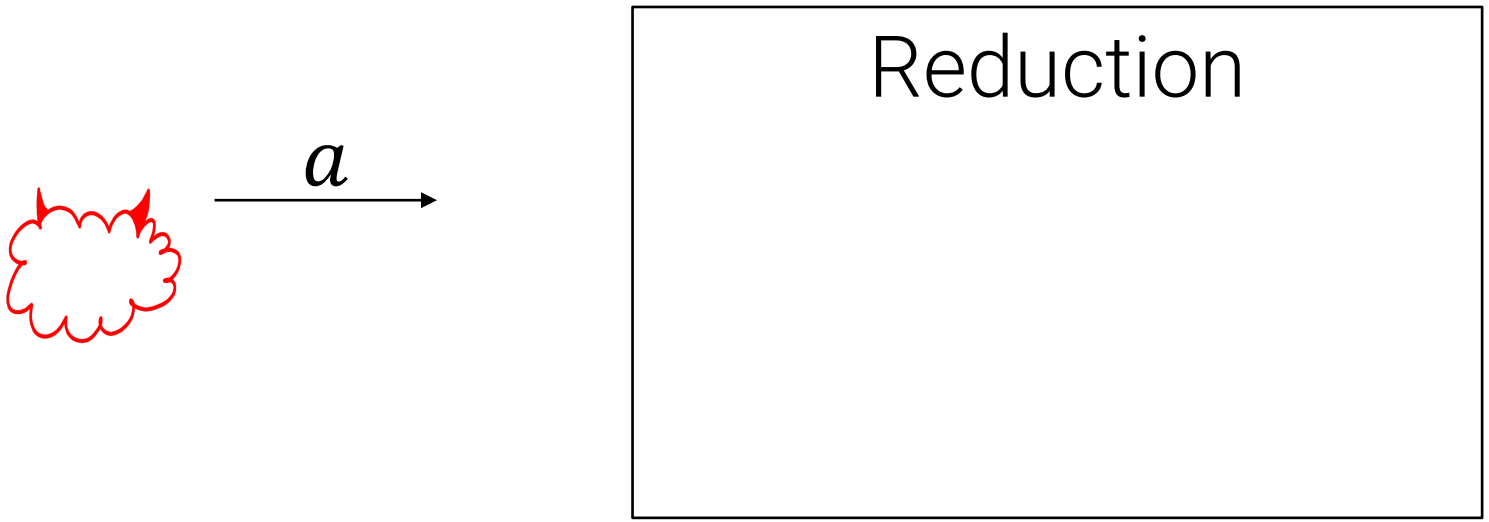
Post-quantum security proof	=	*Post-quantum* hardness assumption	+	*Quantum-compatible* Reduction

Important point:
the ***security reduction*** must be *quantum-compatible*!

Reduction must imply:	Any ***quantum*** attack on the protocol → ***quantum*** attack on the assumption

Some classical reductions are quantum-compatible, but problems arise if the reduction *rewinds the adversary*.

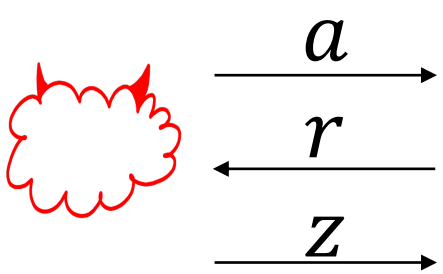
Some classical reductions are quantum-compatible, but problems arise if the reduction ***rewinds the adversary***.



Some classical reductions are quantum-compatible, but problems arise if the reduction ***rewinds the adversary***.

Some classical reductions are quantum-compatible, but problems arise if the reduction ***rewinds the adversary***.

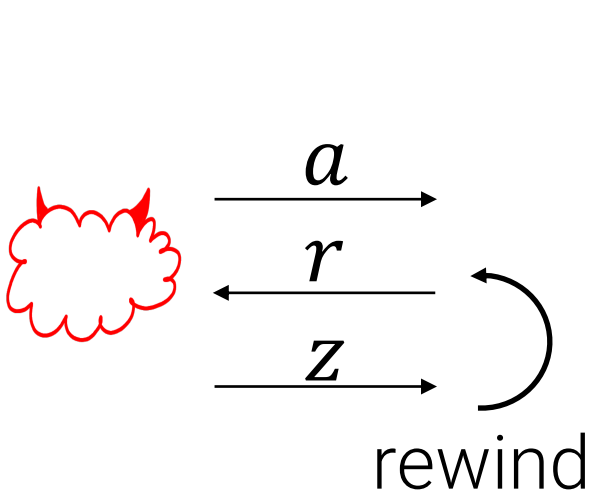
Some classical reductions are quantum-compatible, but problems arise if the reduction ***rewinds the adversary***.



Reduction

1) Record (a, r, z).

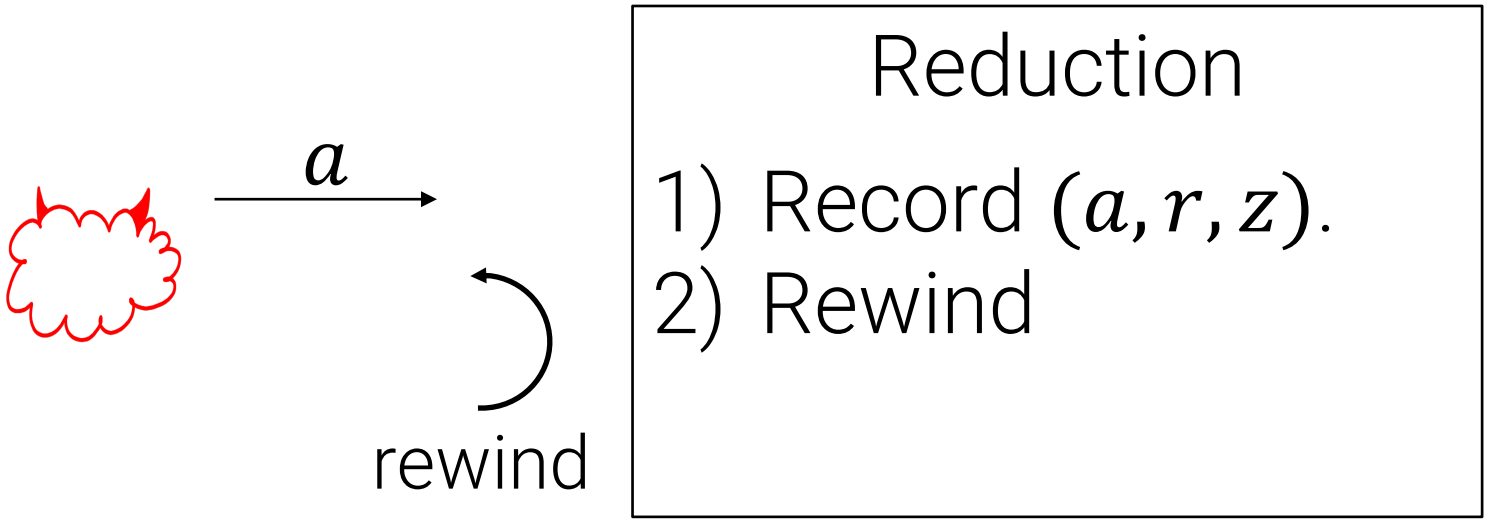
Some classical reductions are quantum-compatible, but problems arise if the reduction ***rewinds the adversary***.



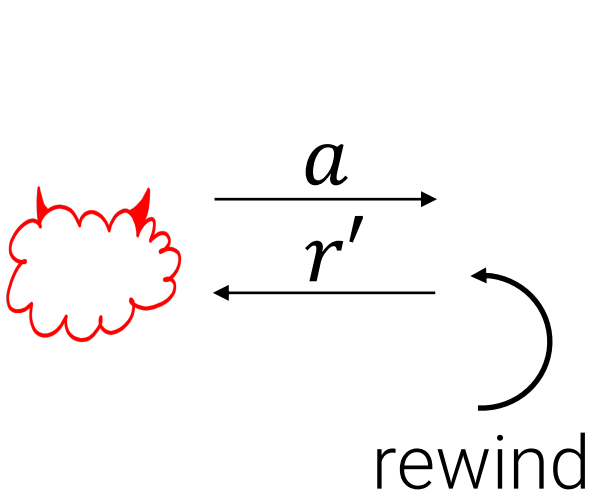
Reduction

1) Record (a, r, z).
2) Rewind

Some classical reductions are quantum-compatible, but problems arise if the reduction ***rewinds the adversary***.



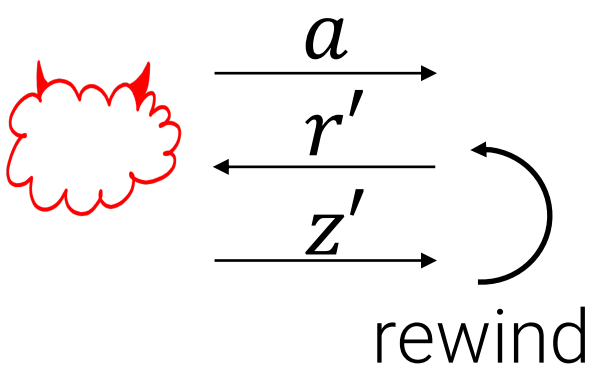
Some classical reductions are quantum-compatible, but problems arise if the reduction ***rewinds the adversary***.



Reduction

1) Record (a, r, z).
2) Rewind

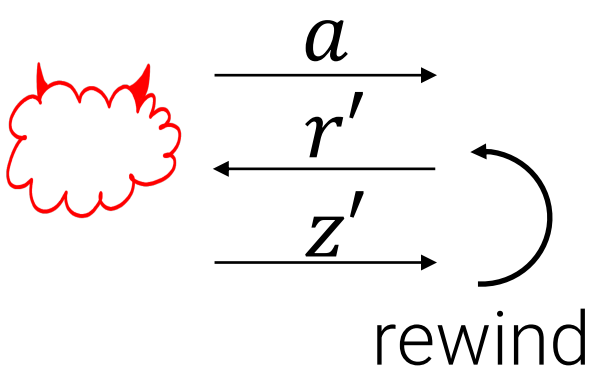
Some classical reductions are quantum-compatible, but problems arise if the reduction ***rewinds the adversary***.



Reduction

1) Record (a, r, z).
2) Rewind

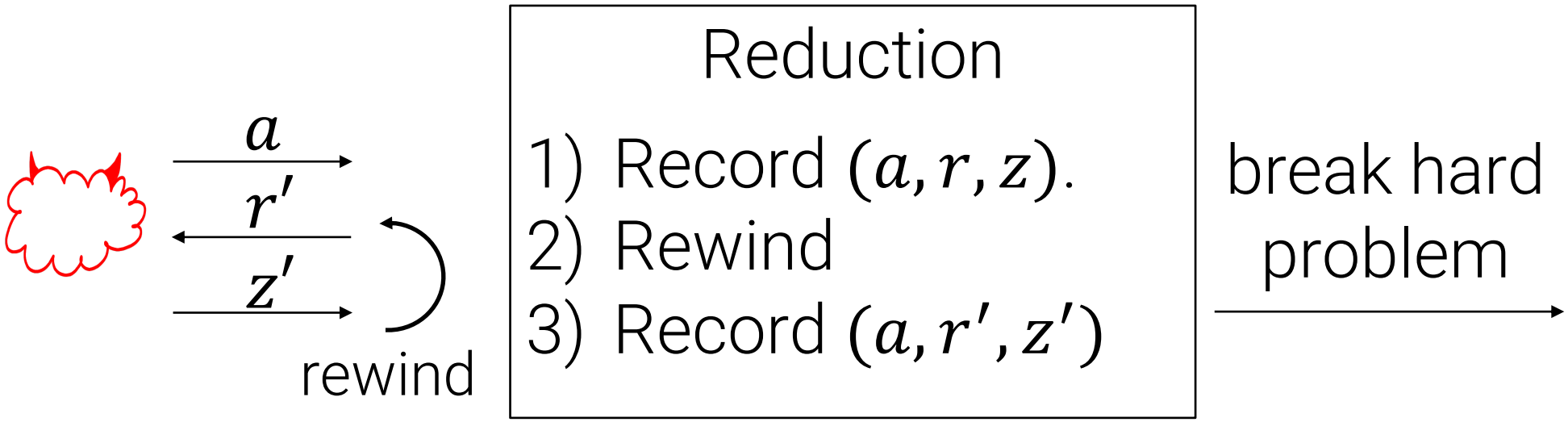
Some classical reductions are quantum-compatible, but problems arise if the reduction ***rewinds the adversary***.



Reduction

1) Record (a, r, z).
2) Rewind
3) Record (a, r', z')

Some classical reductions are quantum-compatible, but problems arise if the reduction ***rewinds the adversary***.



Problem: unclear how to rewind a quantum adversary since running the adversary may disturb its state!

Problem: unclear how to rewind a quantum adversary since running the adversary may disturb its state!

Some quantum rewinding techniques are known [Watrous06,U12], but have very limited applications.

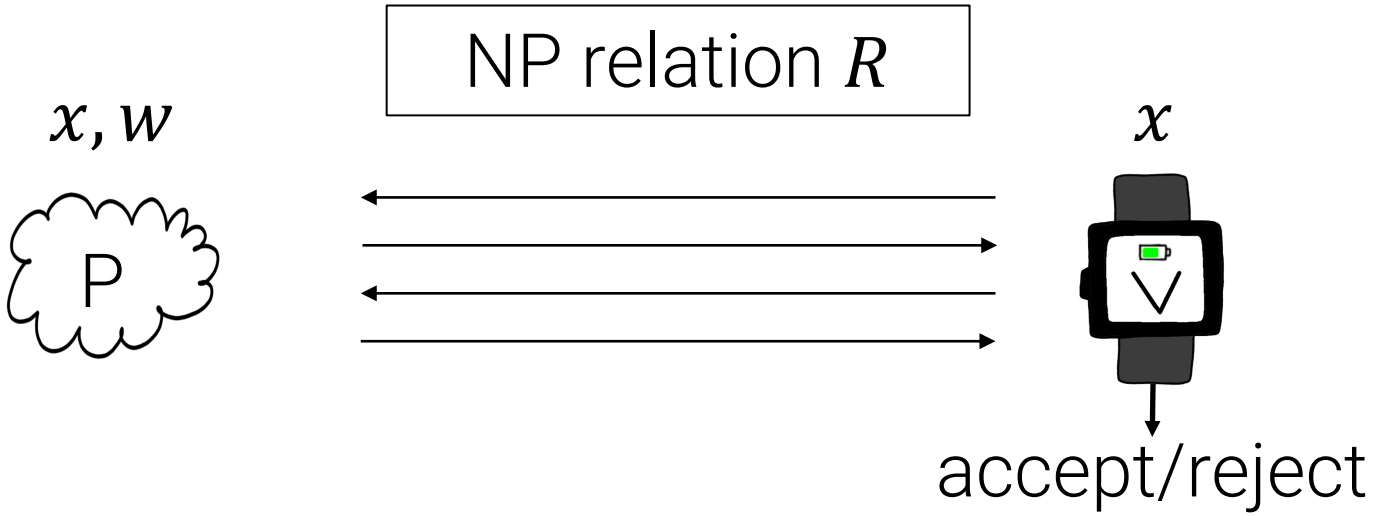
Problem: unclear how to rewind a quantum adversary since running the adversary may disturb its state!

Some quantum rewinding techniques are known [Watrous06,U12], but have very limited applications.

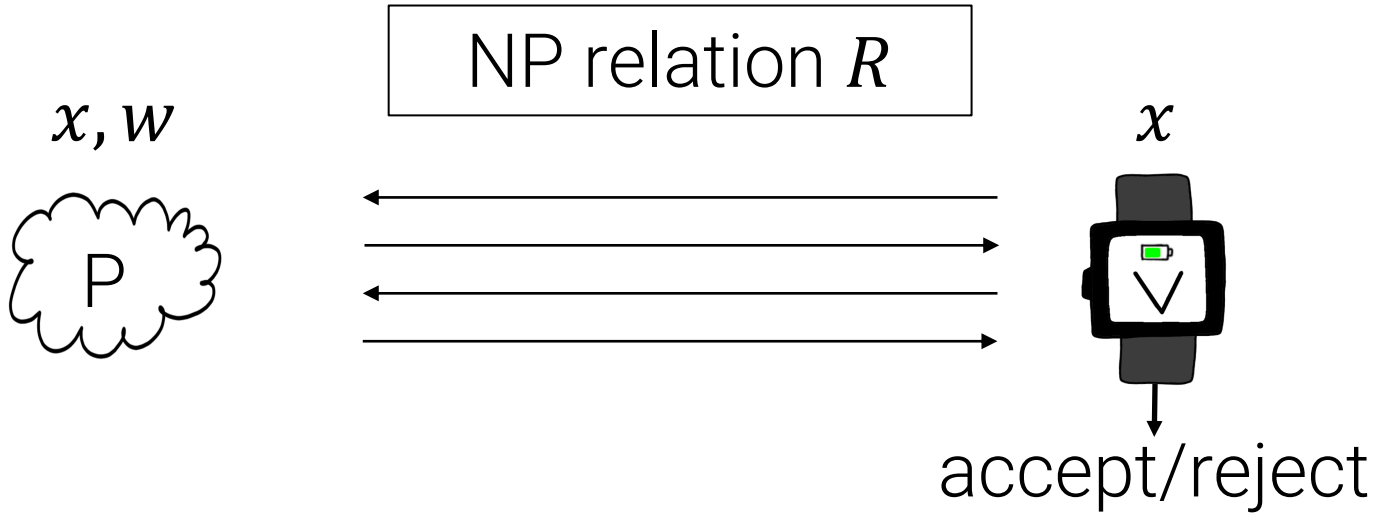
Important application where known techniques fail:

Succinct Arguments

Succinct Arguments for NP

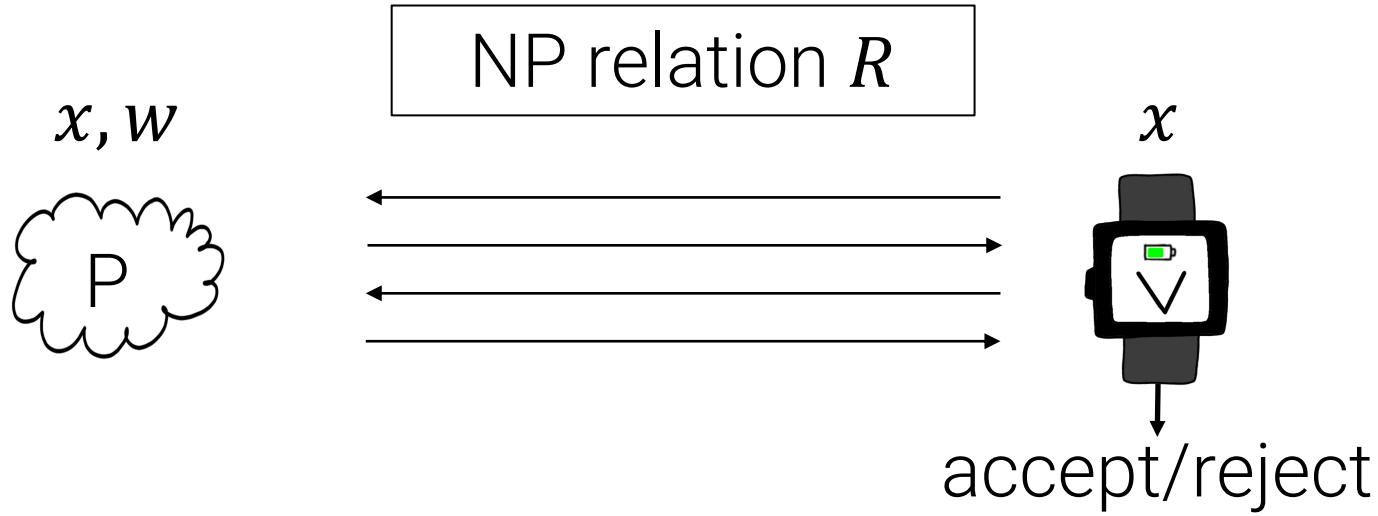


Succinct Arguments for NP



Succinct $= poly(\lambda, \log(|x| + |w|))$ communication.

Succinct Arguments for NP

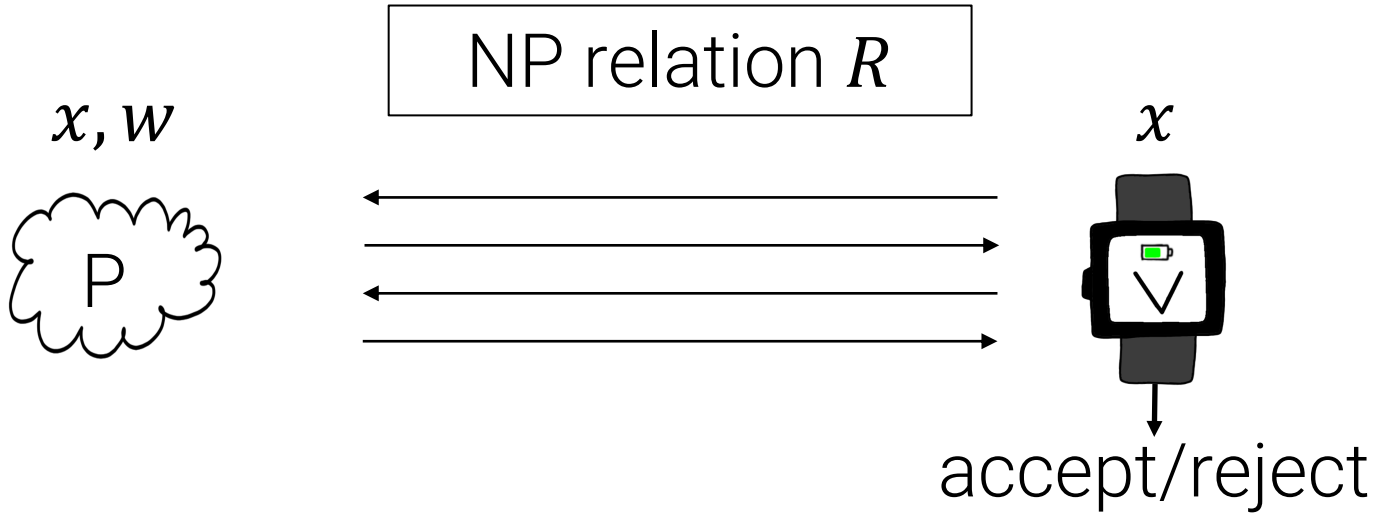


Succinct $= poly(\lambda, \log(|x| + |w|))$ communication.

Argument = complete + computationally sound

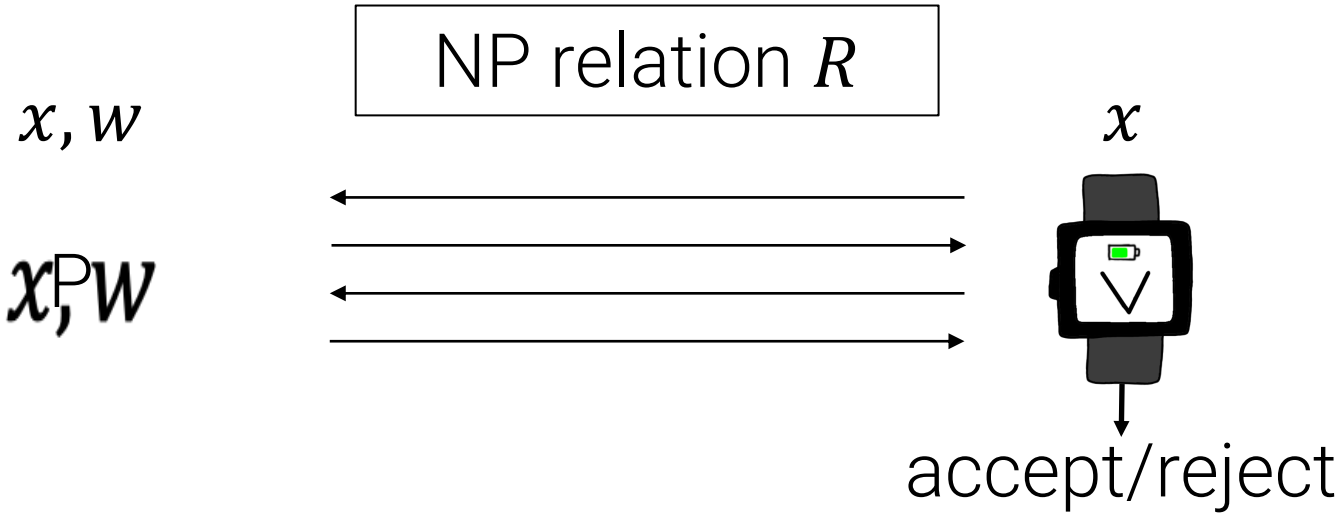
- Complete: if $(x, w) \in R$,  accepts.
- Sound: if $x \notin L(R)$, malicious poly-time  can't fool 

Succinct Arguments for NP



[Kilian92]: Succinct arguments for NP exist assuming collision-resistant hash functions (CRHFs).

Succinct Arguments for NP

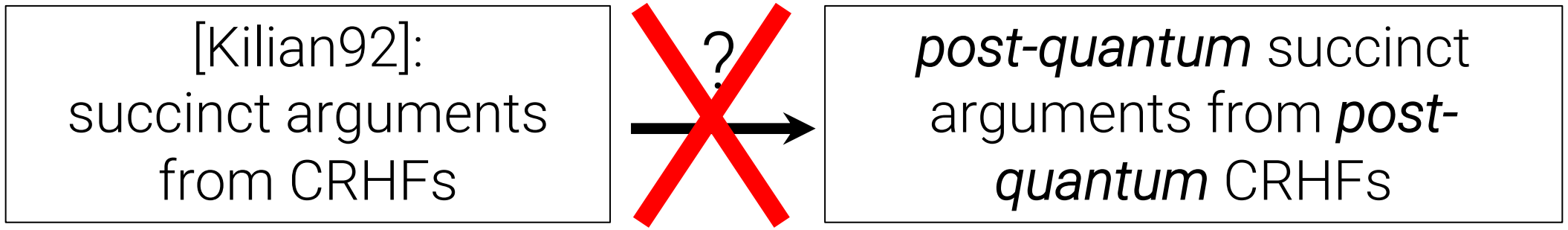


[Kilian92]: Succinct arguments for NP exist assuming collision-resistant hash functions (CRHFs).

Many applications: Succinct non-interactive arguments (SNARGs) [Micali94], Universal arguments [BG01], non-black-box zero knowledge [Barak01], …

Kilian’s security proof fundamentally relies on rewinding, and does not extend to quantum attackers.

Kilian's security proof fundamentally relies on rewinding, and does not extend to quantum attackers.



[CMSZ21] Result

Kilian's protocol is post-quantum secure when instantiated with a post-quantum hash function*.

*collapsing hash function [U16]

[CMSZ21] Result

Kilian's protocol is post-quantum secure when instantiated with a post-quantum hash function*.

*collapsing hash function [U16]

Technique

Extract unbounded number of accepting transcripts from quantum adversary.

[CMSZ21] Result

Kilian's protocol is post-quantum secure when instantiated with a post-quantum hash function*.

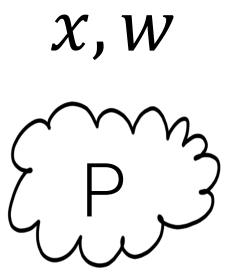
*collapsing hash function [U16]

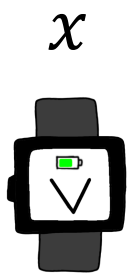
Technique

Extract unbounded number of accepting transcripts from quantum adversary.

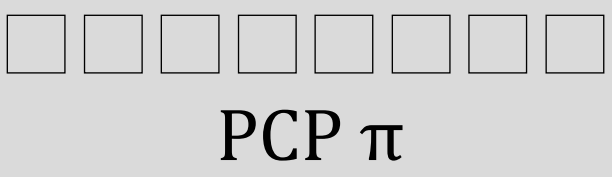
Prior work [U12,U16]: extract constant number of transcripts.

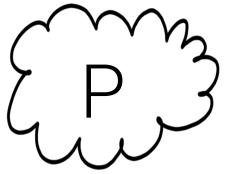
Kilian's protocol





Encode w as PCP

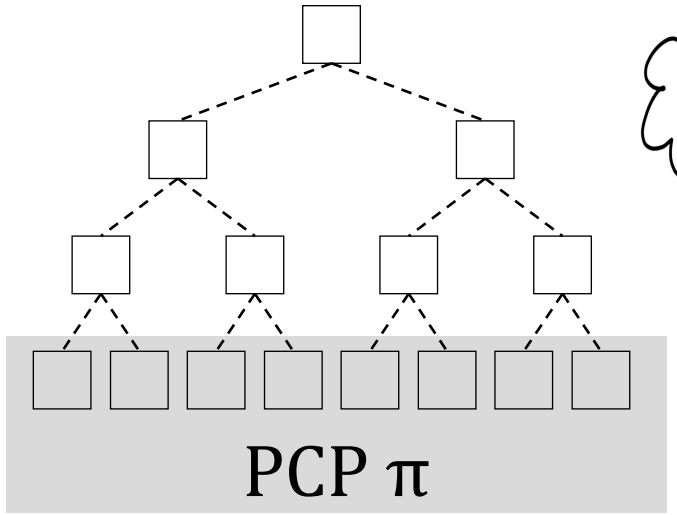




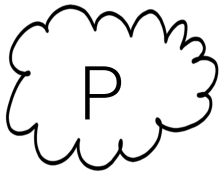
sends short commitment to PCP π.

Kilian's protocol

$\text{com} = \text{Merkle}_h(\pi)$



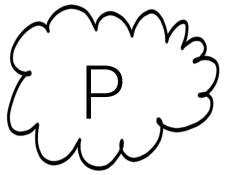
x, w



x

CRHF h

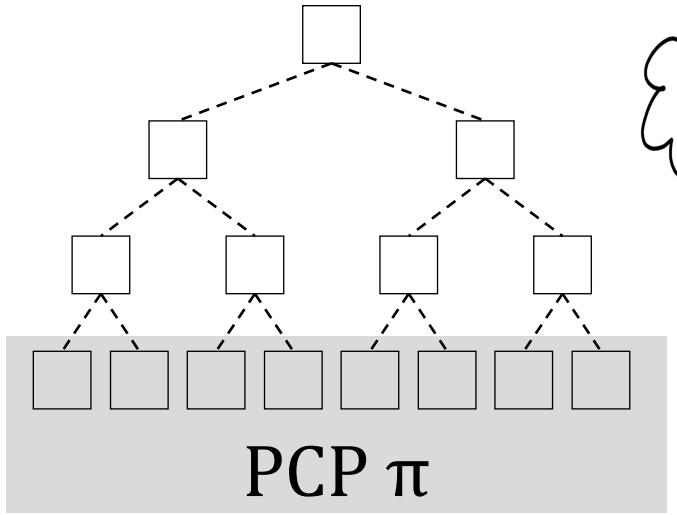
com



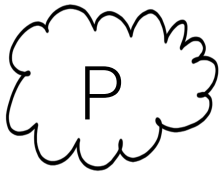
sends short commitment to PCP π.

Kilian's protocol

$\text{com} = \text{Merkle}_h(\pi)$



x, w



x

CRHF h

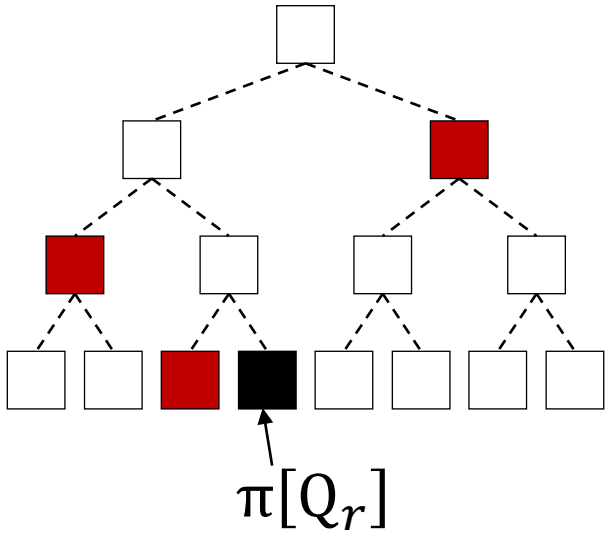
com

r

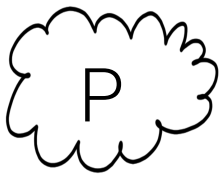
samples PCP verifier coins $r \leftarrow R$.

Kilian's protocol

$\mathrm{com} = \mathrm{Merkle}_h(\pi)$



$\mathrm{open}[Q_r] =$ ■



x, w

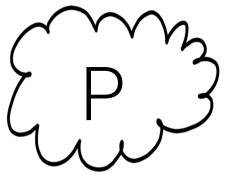
x

CRHF h

com

r

$\pi[Q_r], \mathrm{open}[Q_r]$

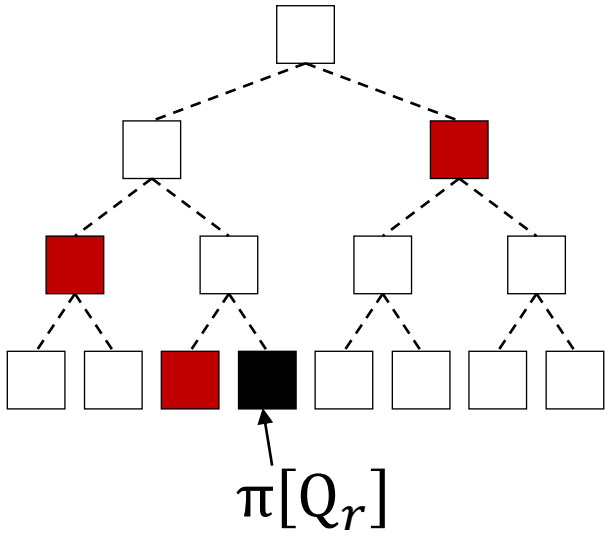


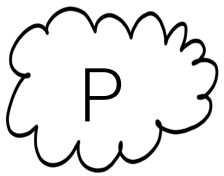
sends $\pi[Q_r]$ + short opening proofs

Q_r = indices PCP verifier checks on random coins r

Kilian's protocol

$\mathrm{com} = \mathrm{Merkle}_h(\pi)$





x, w

x

CRHF h

com

r

$\pi[Q_r], \mathrm{open}[Q_r]$

accept
or reject

accepts if openings valid
+ PCP verifier accepts

Proving Security, Classically

$x \notin L$

Reduction

Assume  fools  into accepting on $x \notin L$.

Proving Security, Classically

$x \notin L$

CRHF h

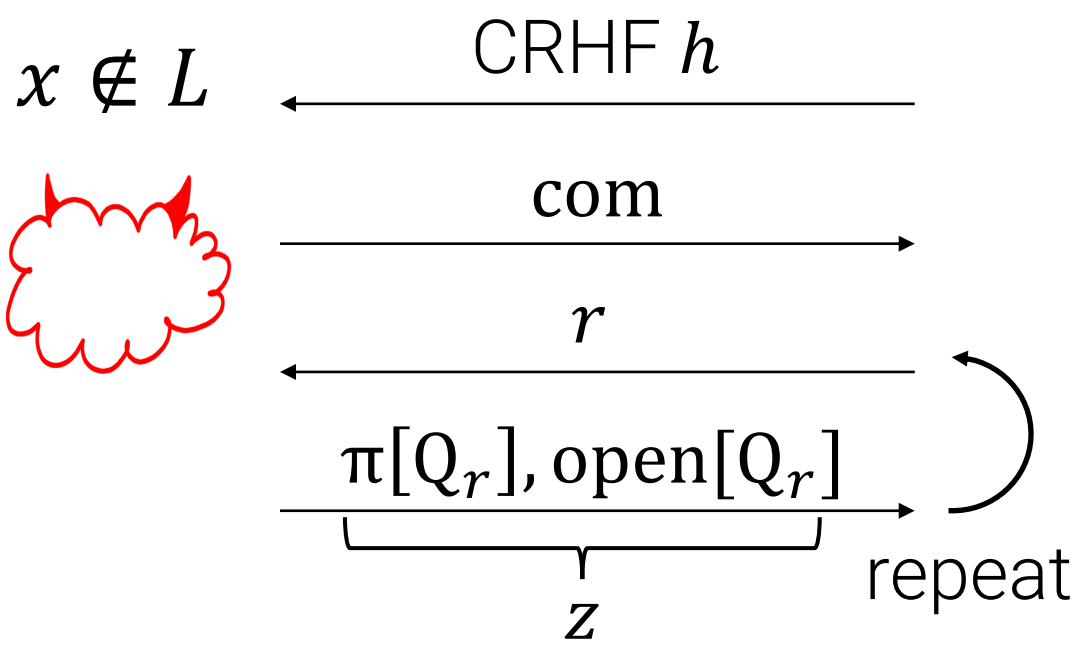
$\mathbf{com}$

Reduction

1) Run  to get $\mathbf{com}$.

Assume  fools  into accepting on $x \notin L$.

Proving Security, Classically

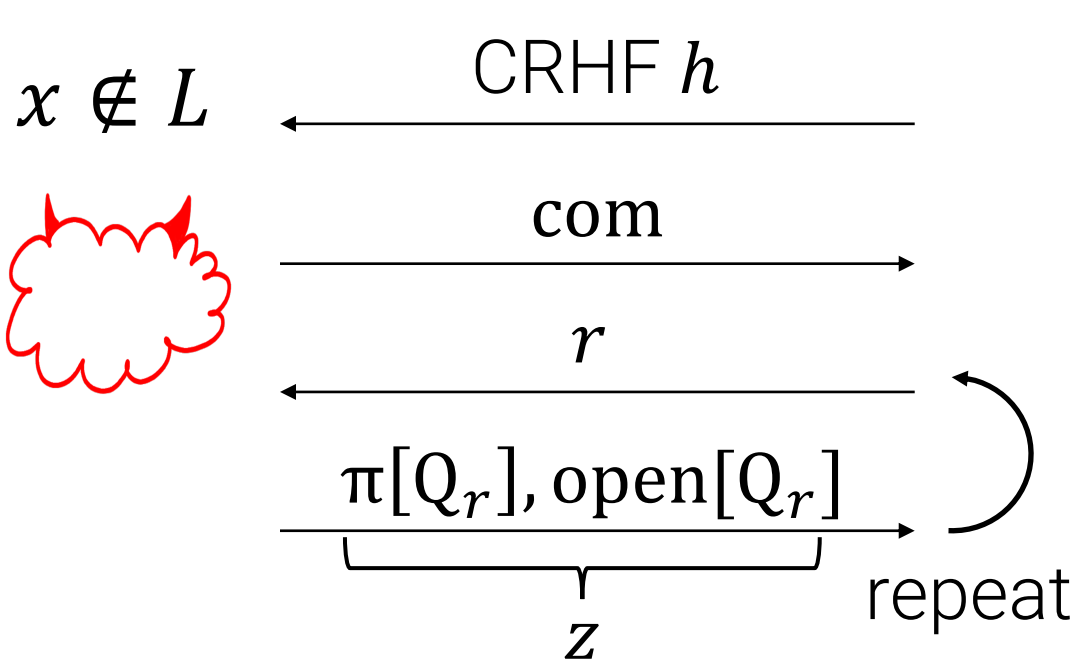


Reduction

1) Run  to get **com**.
2) Repeatedly ask  to answer on random r.

Reduction's goal: record ***many*** accepting transcripts (r_i, z_i)

Proving Security, Classically



Reduction

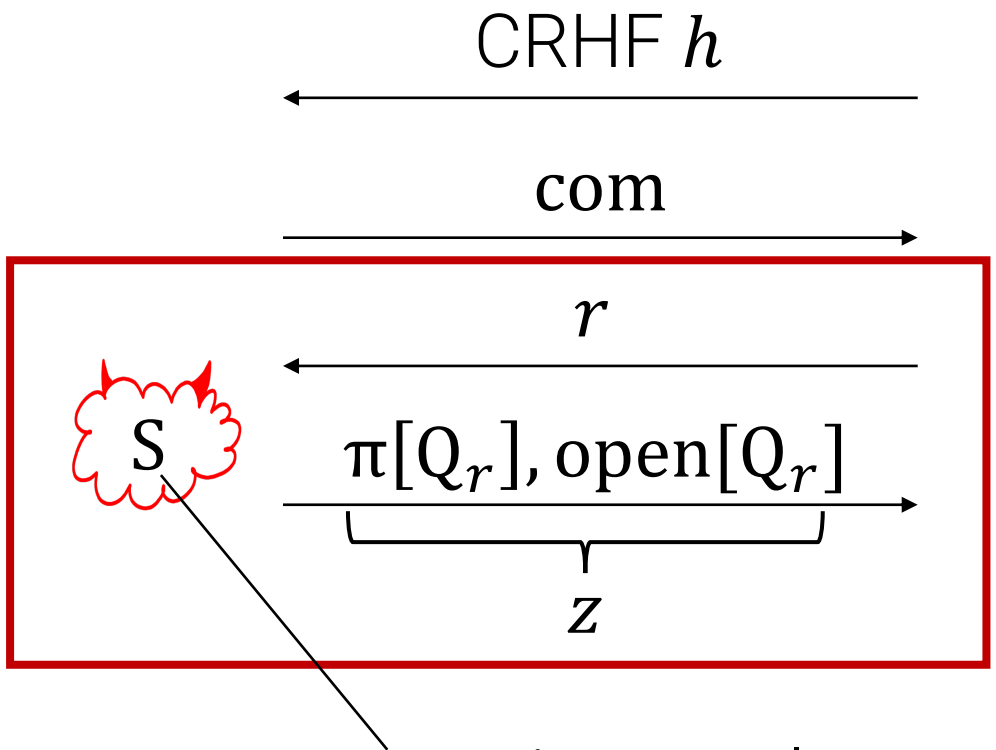
1) Run  to get **com**.
2) Repeatedly ask  to answer on random r.

Reduction's goal: record ***many*** accepting transcripts (r_i, z_i)

Eventually finds impossible π **OR** collision.

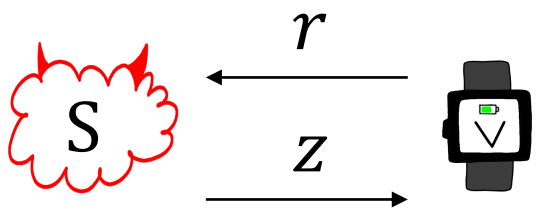
Pr[PCP verifier accepts π] > PCP soundness error

Proving Security, Classically



S = internal state after first two messages

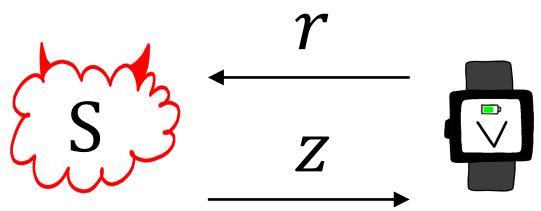
The Rewinding Problem



$$p := \Pr_{\substack{r \leftarrow R, \\ z \leftarrow S(r)}} [V \text{ accepts } (r, z)]$$

Goal: given S with large enough success probability p, obtain k accepting transcripts (r_i, z_i).

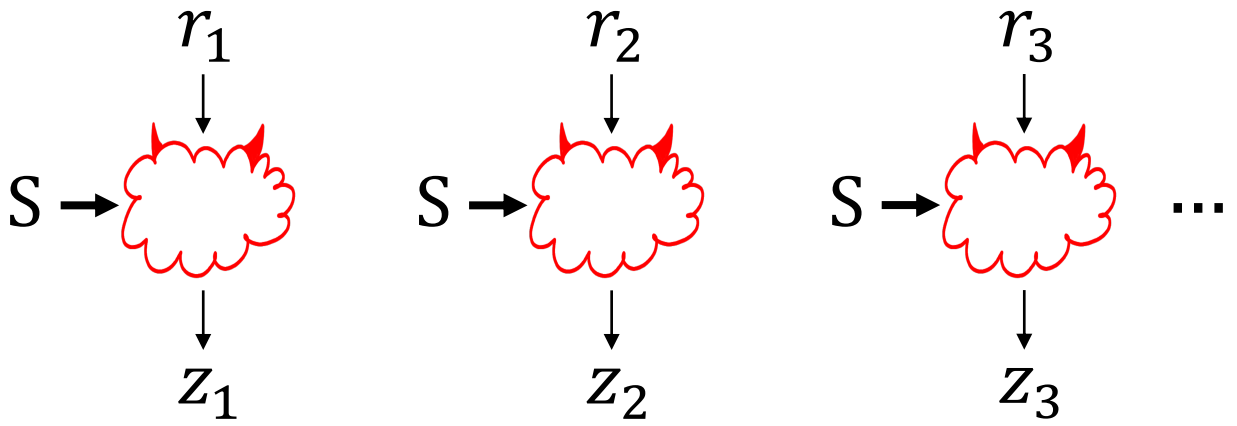
The Rewinding Problem



$$p := \Pr_{\substack{r \leftarrow R, \\ z \leftarrow S(r)}} [V \text{ accepts } (r, z)]$$

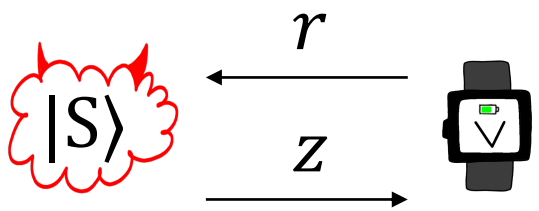
Goal: given S with large enough success probability p, obtain k accepting transcripts (r_i, z_i).

If S is classical, perform independent trials:



obtain k accepting transcripts in k/p trials (expected)

The Rewinding Problem

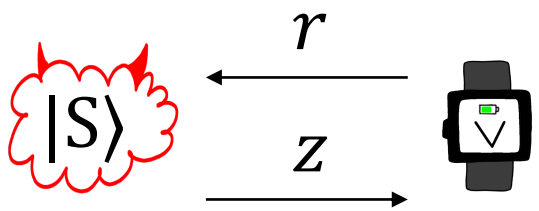


$$p := \Pr_{\substack{r \leftarrow R, \\ z \leftarrow |S\rangle(r)}} [\text{V accepts } (r, z)]$$

Goal: given $|S\rangle$ with large enough success probability p, obtain k accepting transcripts (r_i, z_i).

What happens when $|S\rangle$ is quantum?

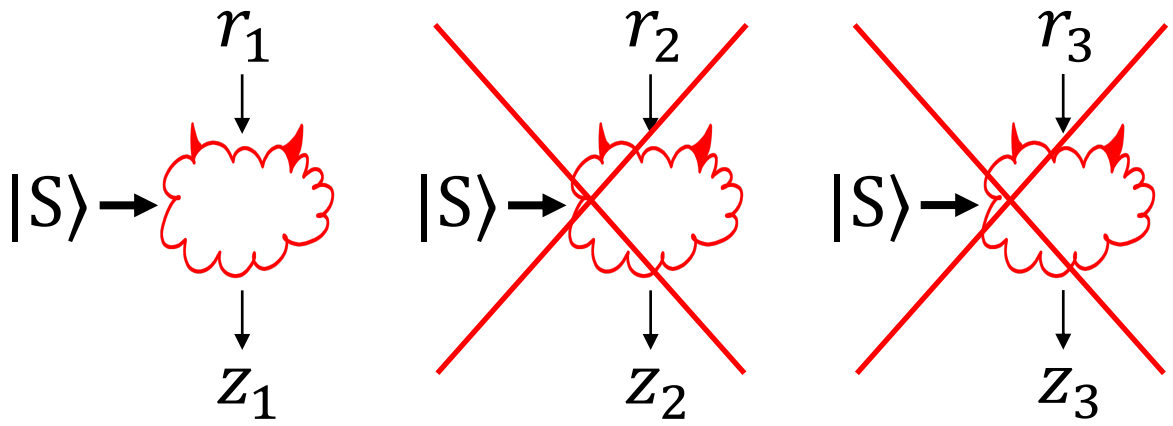
The Rewinding Problem



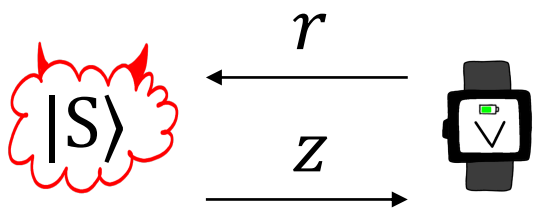
$$p := \Pr_{\substack{r \leftarrow R, \\ z \leftarrow |S\rangle(r)}} [\text{V accepts } (r, z)]$$

Goal: given $|S\rangle$ with large enough success probability p, obtain k accepting transcripts (r_i, z_i).

Independent trials violate no-cloning!



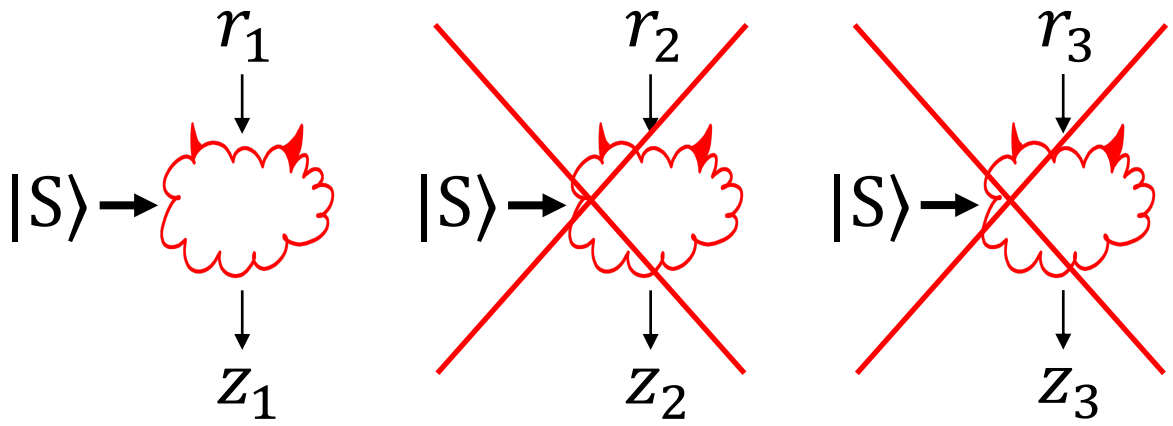
The Rewinding Problem



$$p := \Pr_{\substack{r \leftarrow R, \\ z \leftarrow |S\rangle(r)}} [\text{V accepts } (r, z)]$$

Goal: given $|S\rangle$ with large enough success probability p, obtain k accepting transcripts (r_i, z_i).

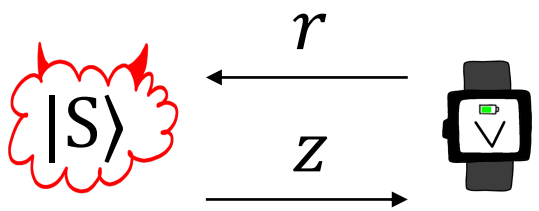
Independent trials violate no-cloning!



Idea: ***sequential trials.***

Run next trial on leftover state from previous trial

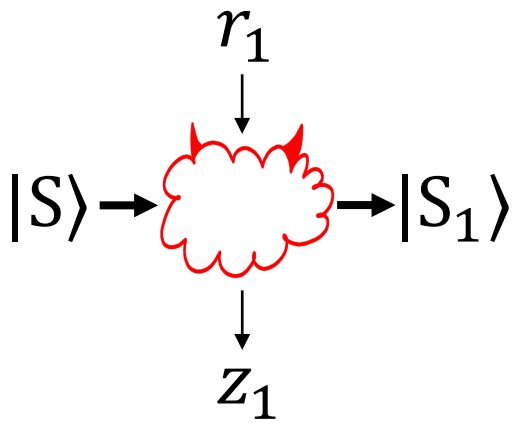
The Rewinding Problem



$$p := \Pr_{\substack{r \leftarrow R, \\ z \leftarrow |S\rangle(r)}} [\text{accepts } (r, z)]$$

Goal: given $|S\rangle$ with large enough success probability p, obtain k accepting transcripts (r_i, z_i).

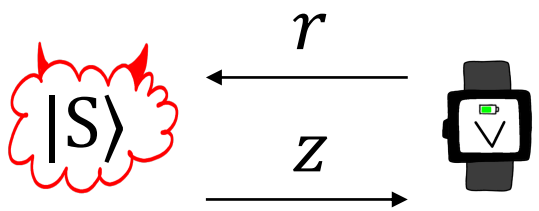
Independent trials violate no-cloning!



Idea: ***sequential trials.***

Run next trial on leftover state from previous trial

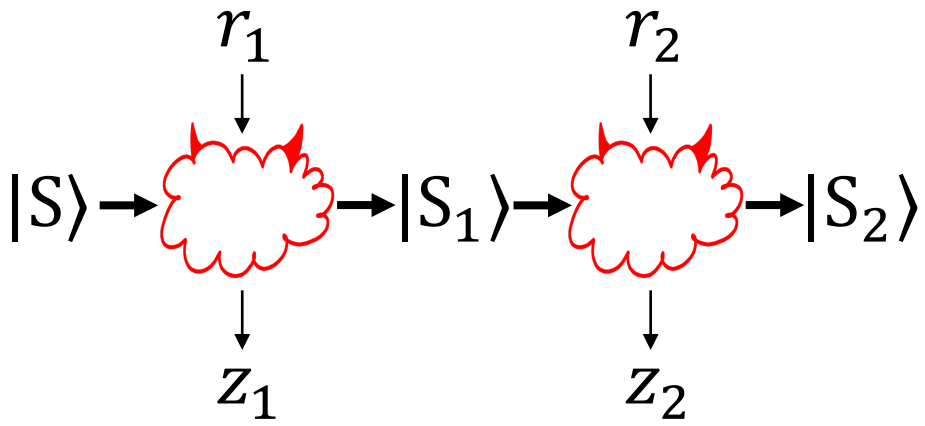
The Rewinding Problem



$$p := \Pr_{\substack{r \leftarrow R, \\ z \leftarrow |S\rangle(r)}} [\text{V accepts } (r, z)]$$

Goal: given $|S\rangle$ with large enough success probability p, obtain k accepting transcripts (r_i, z_i).

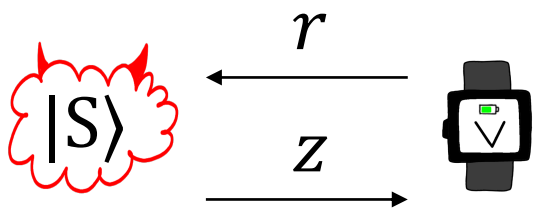
Independent trials violate no-cloning!



Idea: ***sequential trials.***

Run next trial on leftover state from previous trial

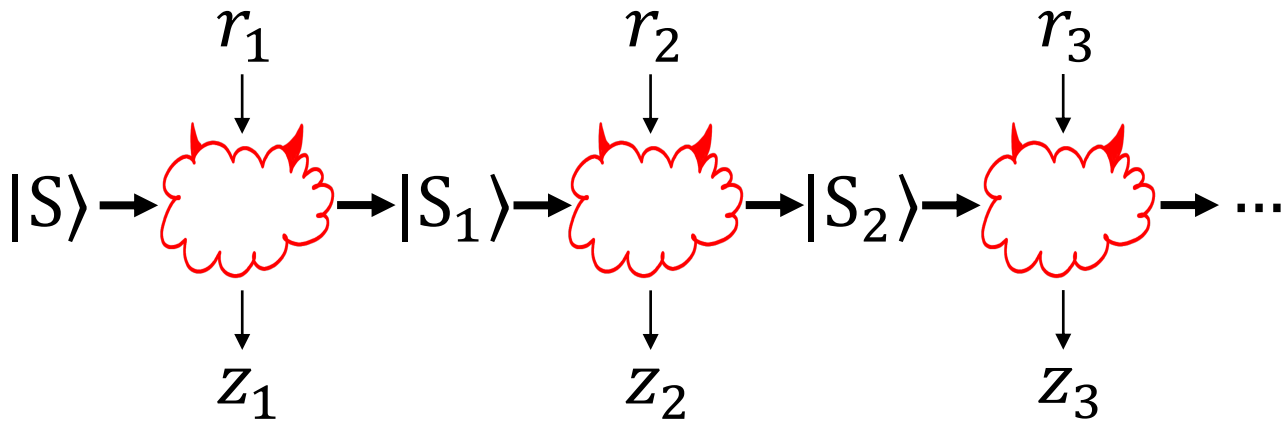
The Rewinding Problem



$$p := \Pr_{\substack{r \leftarrow R, \\ z \leftarrow |S\rangle(r)}} [\text{V accepts } (r, z)]$$

Goal: given $|S\rangle$ with large enough success probability p, obtain k accepting transcripts (r_i, z_i).

Independent trials violate no-cloning!



Idea: ***sequential trials.***

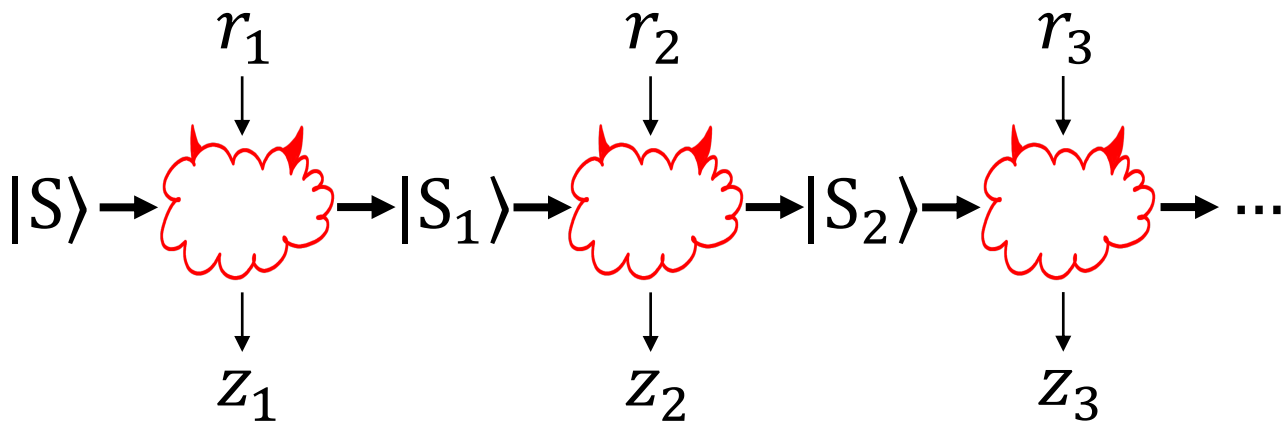
Run next trial on leftover state from previous trial

$$p_i := \Pr[\text{Trial } i \text{ successful}]$$

Unruh's Lemma [U12,DFMS19]:

$$\prod_{i=1}^{d} p_i \geq p_1^{2d-1}$$

In words: a constant number of sequential trials all succeed with non-negligible probability.



Idea: ***sequential trials.***

Run next trial on leftover state from previous trial

$$p_i := \Pr[\text{Trial } i \text{ successful}]$$

Unruh's Lemma [U12,DFMS19]:

$$\prod_{i=1}^{d} p_i \geq p_1^{2d-1}$$

In words: a constant number of sequential trials all succeed with non-negligible probability.

If we could improve this analysis,
we'd get better quantum rewinding!

$$p_i := \Pr[\text{Trial } i \text{ successful}]$$

Unruh's Lemma [U12,DFMS19]:

$$\prod_{i=1}^{d} p_i \geq p_1^{2d-1}$$

In words: a constant number of sequential trials all succeed with non-negligible probability.

Bad news: We show p_i can decay exponentially fast.

This means "sequential trials" rewinding is stuck at obtaining a constant number of transcripts.

What do we do differently in [CMSZ21]?

What do we do differently in [CMSZ21]?

Intuition: [U12,DFMS19] rewinding uses the fact that if the state remains close to the original $|S\rangle$, it retains some of the original success probability.

What do we do differently in [CMSZ21]?

Intuition: [U12,DFMS19] rewinding uses the fact that if the state remains close to the original $|S\rangle$, it retains some of the original success probability.

Our approach: no need to stay close to the original $|S\rangle$; rewinding "only" requires preserving success probability.

What do we do differently in [CMSZ21]?

Intuition: [U12,DFMS19] rewinding uses the fact that if the state remains close to the original $|S\rangle$, it retains some of the original success probability.

Our approach: no need to stay close to the original $|S\rangle$; rewinding "only" requires preserving success probability.

- We don't run the next trial on the leftover state from the previous trial.

What do we do differently in [CMSZ21]?

Intuition: [U12,DFMS19] rewinding uses the fact that if the state remains close to the original $|S\rangle$, it retains some of the original success probability.

Our approach: no need to stay close to the original $|S\rangle$; rewinding "only" requires preserving success probability.

- We don't run the next trial on the leftover state from the previous trial.
- Instead, run a procedure to ***repair*** the success probability of the leftover state before the next trial.

Quantum Background

- Quantum state: $\sum_x \alpha_x |x\rangle$ superposition over classical x

Quantum Background

- Quantum state: $\sum_x \alpha_x |x\rangle$ superposition over classical x
- Quantum measurement:

$$\sum_x \alpha_x |x\rangle \longrightarrow \boxed{\text{Measure}} \longrightarrow x \text{ with probability } |\alpha_x|^2$$

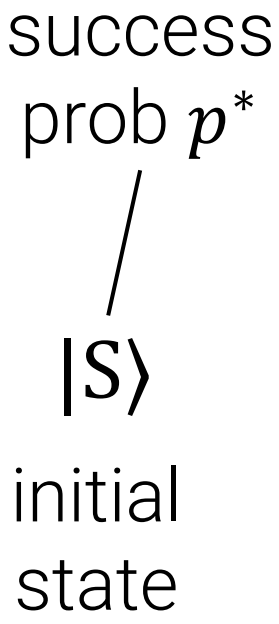
Quantum Background

- Quantum state: $\sum_x \alpha_x |x\rangle$ superposition over classical x
- Quantum measurement:

$$\sum_x \alpha_x |x\rangle \longrightarrow \boxed{\text{Measure}} \longrightarrow x \text{ with probability } |\alpha_x|^2$$

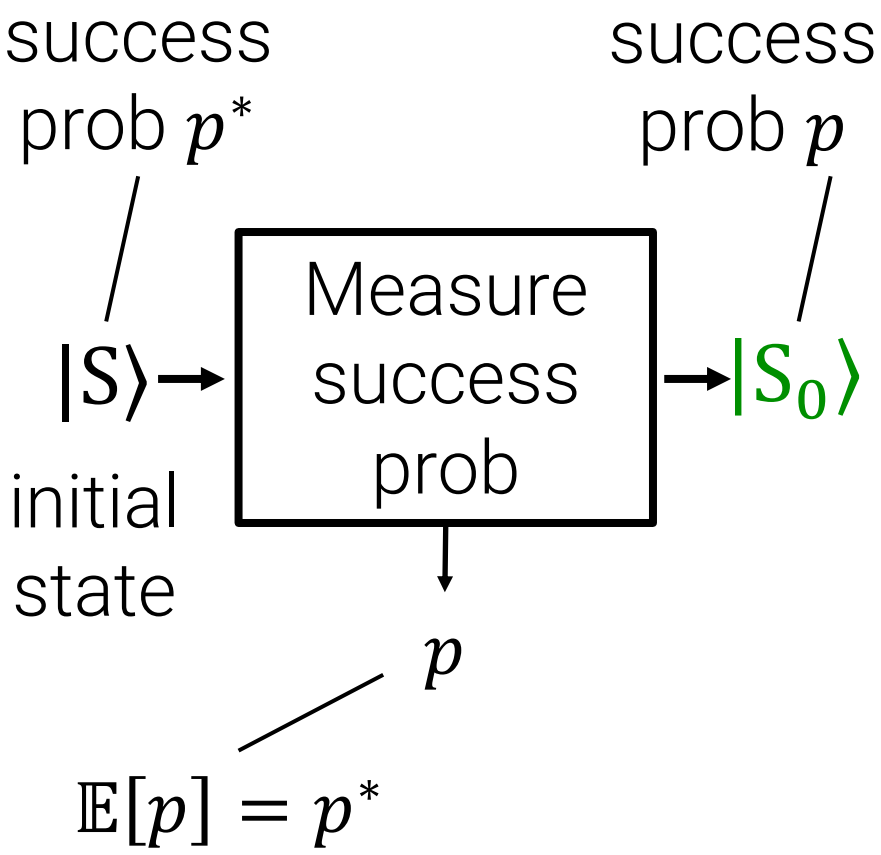
- Can do “partial” measurements that don’t fully collapse the state

[CMSZ21] Rewinding: A Bird's-Eye View



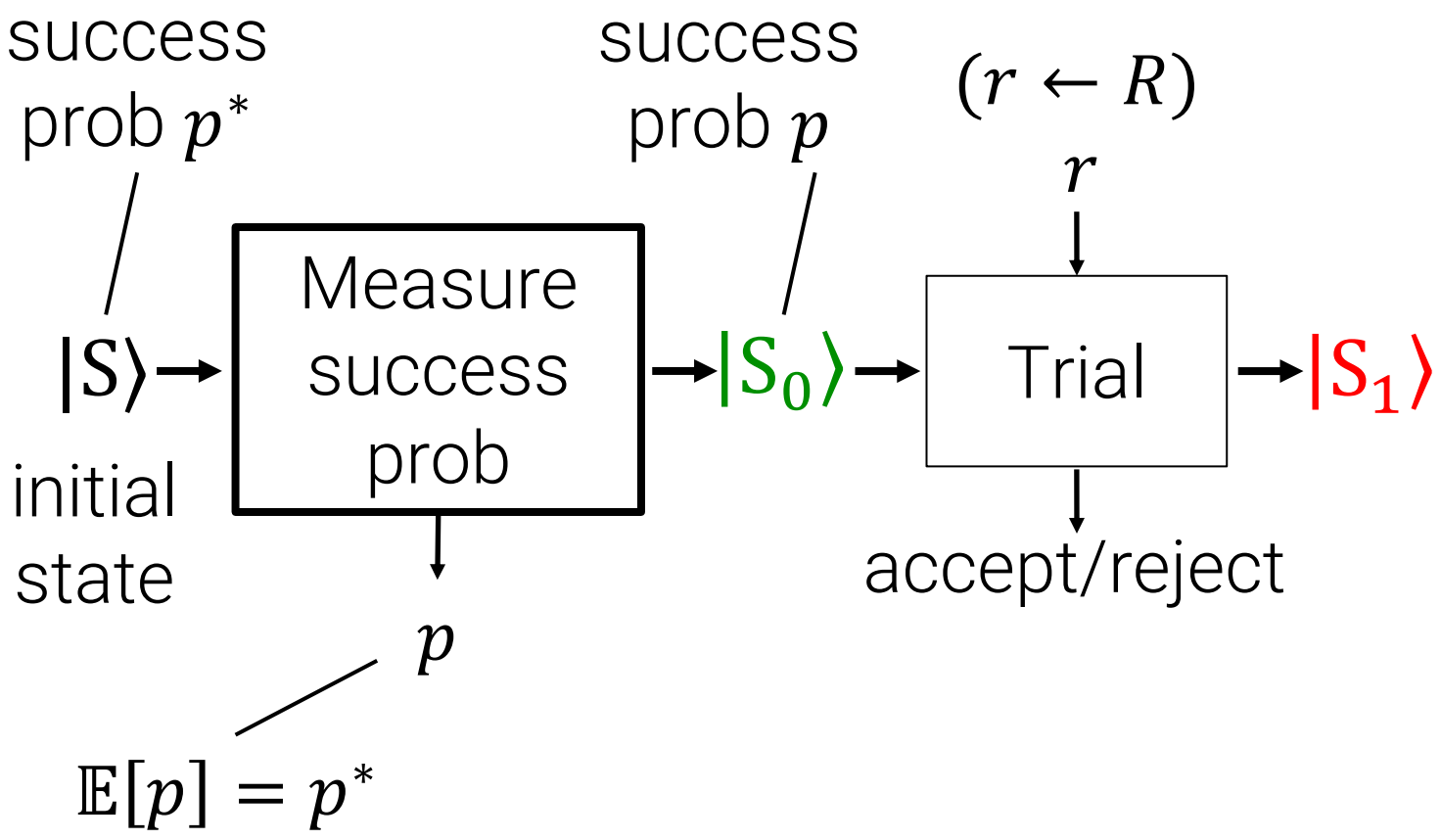
Before any trials, (somehow) measure the adversary's success probability

[CMSZ21] Rewinding: A Bird's-Eye View



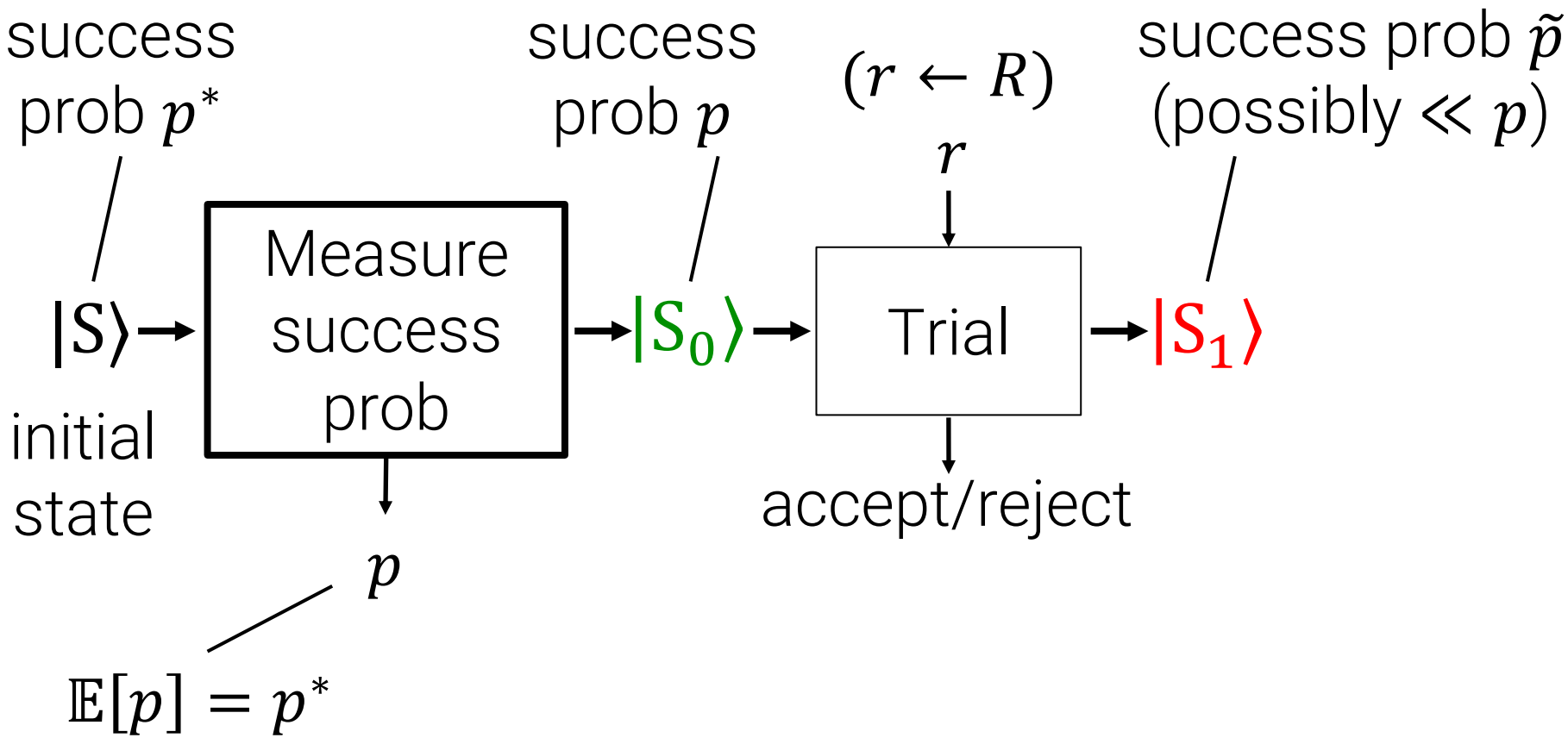
Before any trials, (somehow) measure the adversary's success probability

[CMSZ21] Rewinding: A Bird's-Eye View



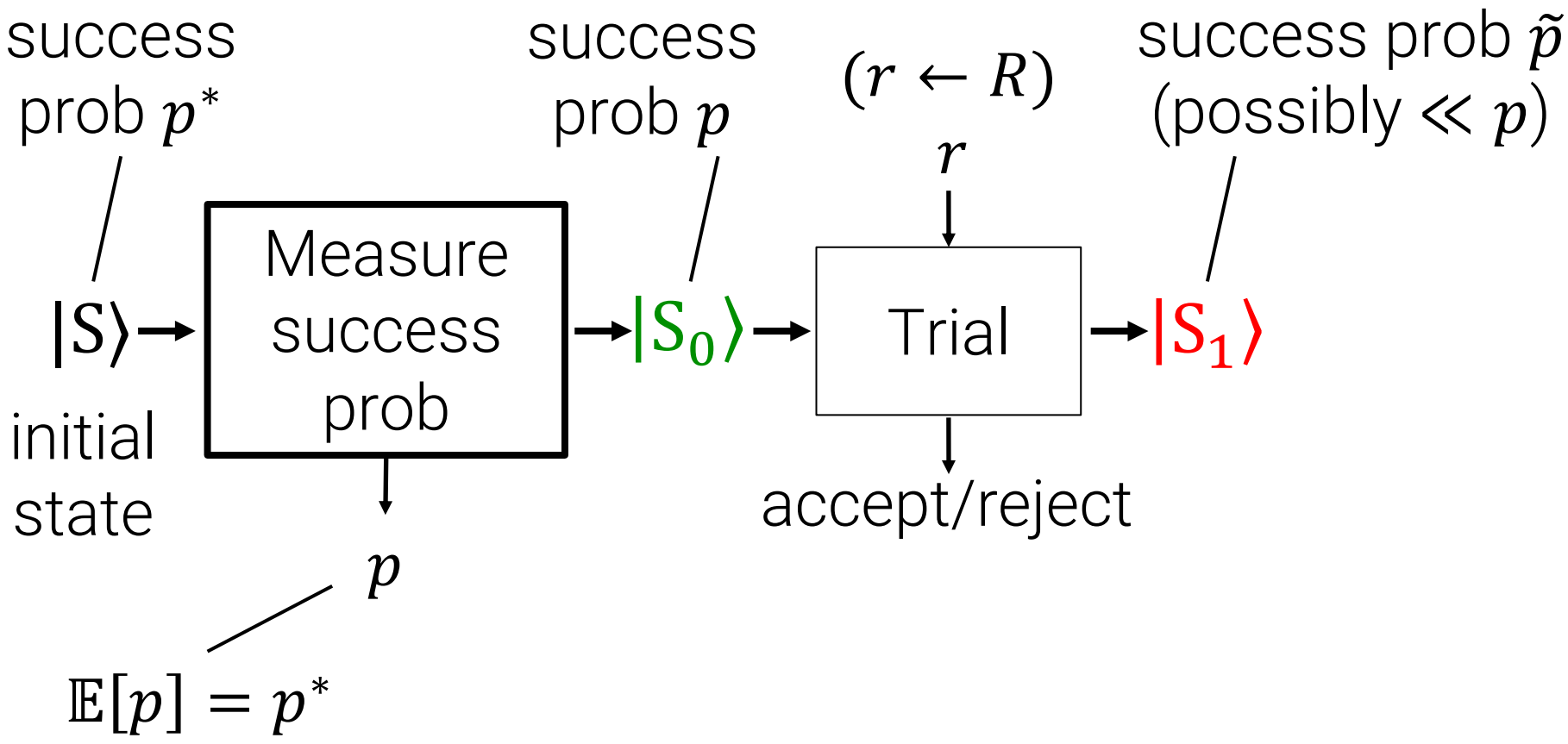
Next, perform trial on $|S_0\rangle$ with **known** success probability p.

[CMSZ21] Rewinding: A Bird's-Eye View



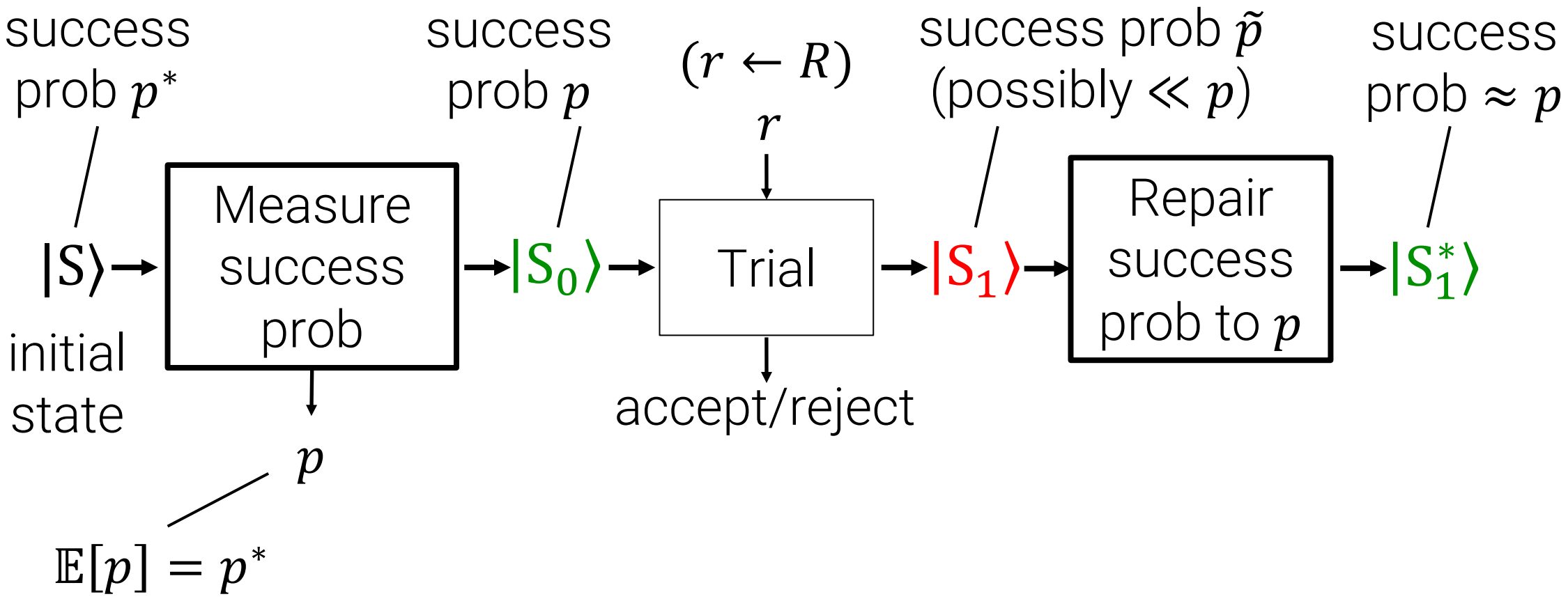
Next, perform trial on $|S_0\rangle$ with **known** success probability p.

[CMSZ21] Rewinding: A Bird's-Eye View



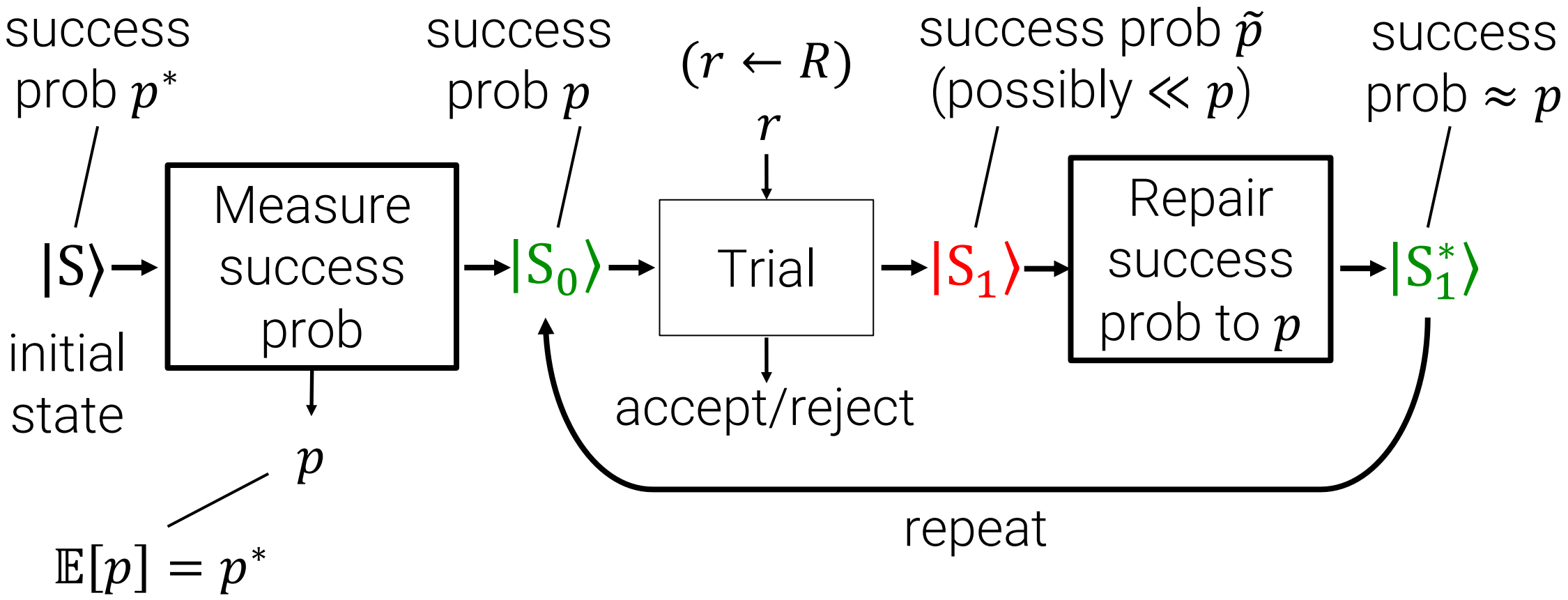
Before next trial, (somehow) **repair the success probability** back to $\approx p$.

[CMSZ21] Rewinding: A Bird's-Eye View



Before next trial, (somehow) **repair the success probability** back to $\approx p$.

[CMSZ21] Rewinding: A Bird's-Eye View



Before next trial, (somehow) **repair the success probability** back to $\approx p$.

Measure success prob

Repair success prob to p

How do we implement these procedures?

Fact: exists special basis $\{|\mathrm{T}_p\rangle\}_{p\in[0,1]}$ where each $|\mathrm{T}_p\rangle$ is adversary with success prob p.

Fact: exists special basis $\{|\mathrm{T}_p\rangle\}_{p\in[0,1]}$ where each $|\mathrm{T}_p\rangle$ is adversary with success prob p.

For $|\mathrm{S}\rangle = \sum_p \alpha_p |\mathrm{T}_p\rangle$ w/ success prob p^*,

$$p^* = \sum_{p\in[0,1]} |\alpha_p|^2 \cdot p$$

Fact: exists special basis $\{|\mathrm{T}_p\rangle\}_{p\in[0,1]}$ where each $|\mathrm{T}_p\rangle$ is adversary with success prob p.

For $|\mathrm{S}\rangle = \sum_p \alpha_p |\mathrm{T}_p\rangle$ w/ success prob p^*,

$$p^* = \sum_{p\in[0,1]} |\alpha_p|^2 \cdot p$$

[MW04,Z20]: Can ***approximately*** measure in this basis:

$|\mathrm{S}\rangle = \sum_p \alpha_p |\mathrm{T}_p\rangle$ collapses to $\approx |\mathrm{T}_p\rangle$ w/ prob $|\alpha_p|^2$.

Measure success prob

Repair success prob to p

[MW04,Z20]

Measure success prob

Repair success prob to p

[MW04,Z20]

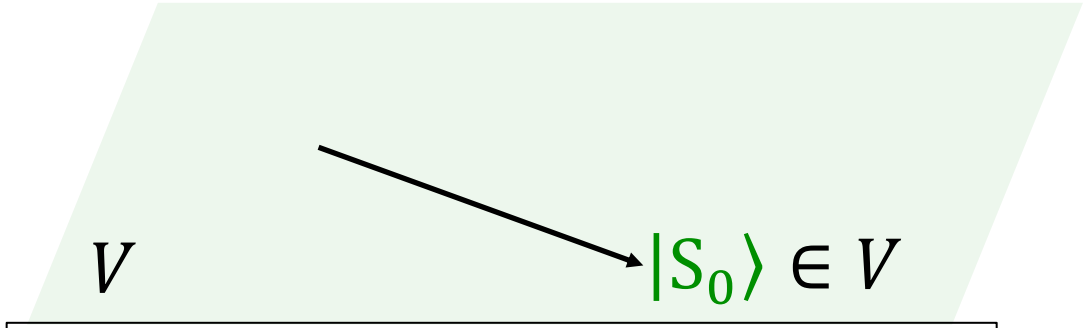
New techniques needed!

[CMSZ21] State Repair

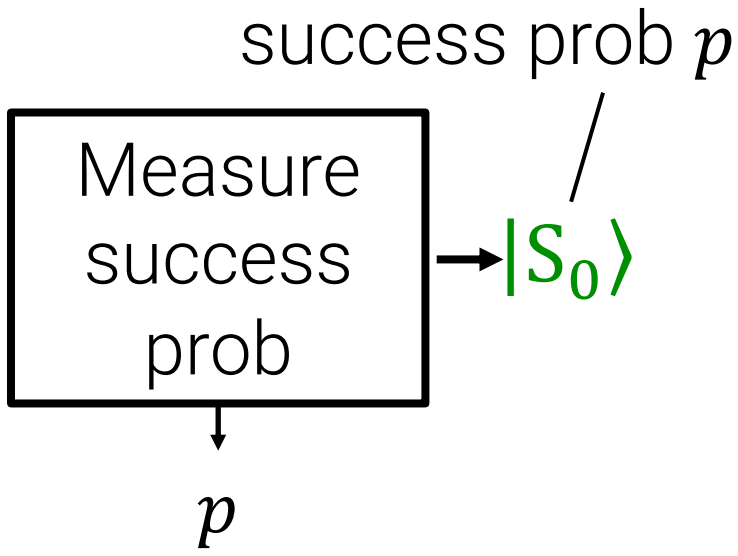
After we measure success prob p, $|S_0\rangle$ lies in subspace

$$V := \mathbf{span}\left(|T_q\rangle\right)_{q \geq p}$$

i.e., V = subspace of attacker states with success prob $\geq p$.



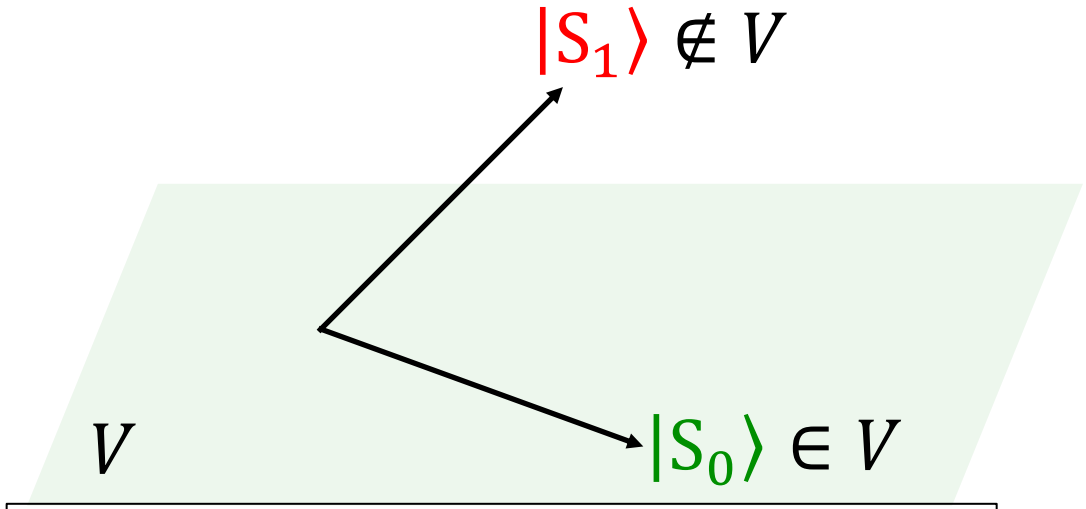
V = subspace of attacker states with success prob $\geq p$



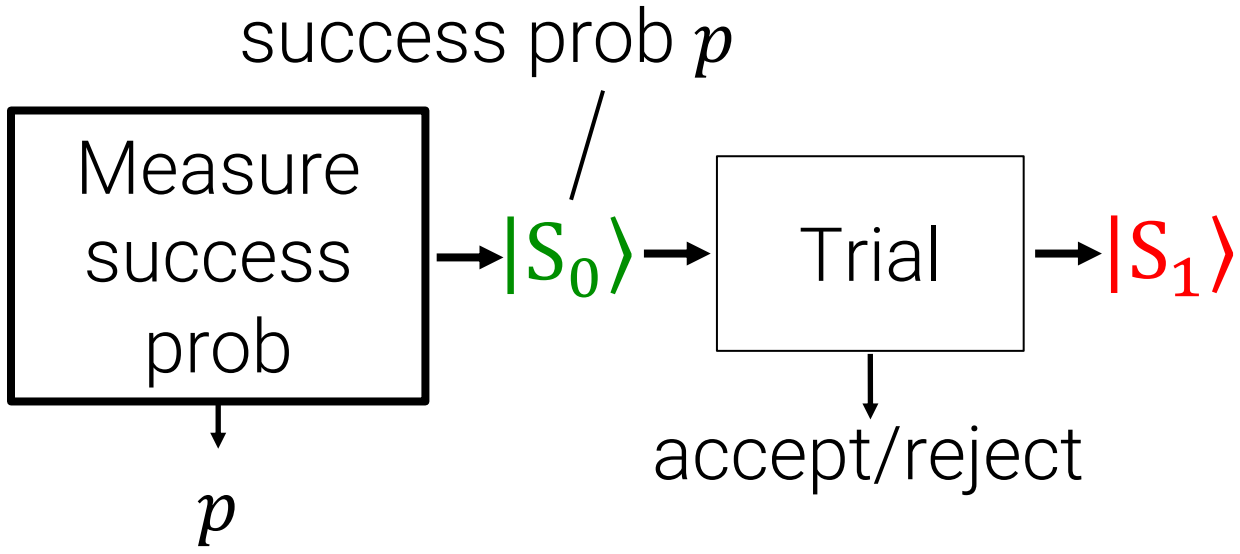
[CMSZ21] State Repair

After trial, state is $|S_1\rangle \notin V$.

We show: $|S_1\rangle$ not too far from V



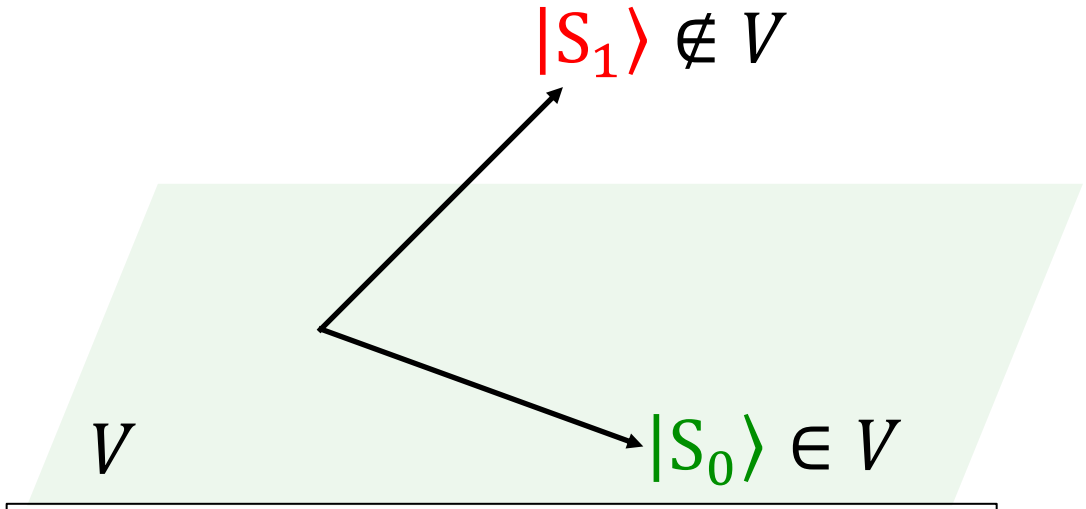
V = subspace of attacker states with success prob $\geq p$



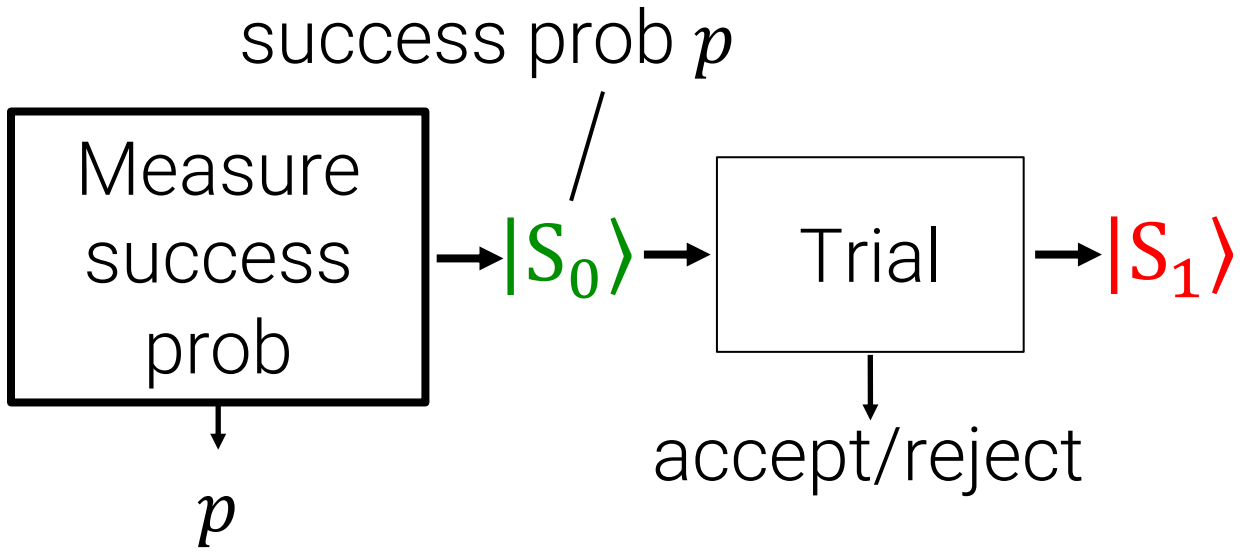
[CMSZ21] State Repair

If $|S_1\rangle$ not far from V, must have non-trivial component in V.

We show: can ***amplify*** this component to output $|S_1^*\rangle \in V$



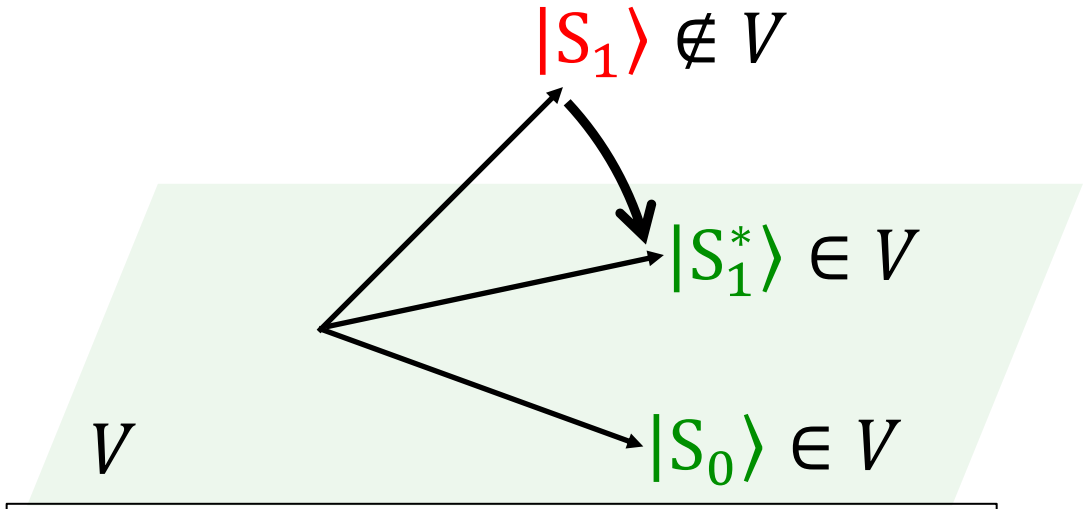
V = subspace of attacker states with success prob $\geq p$



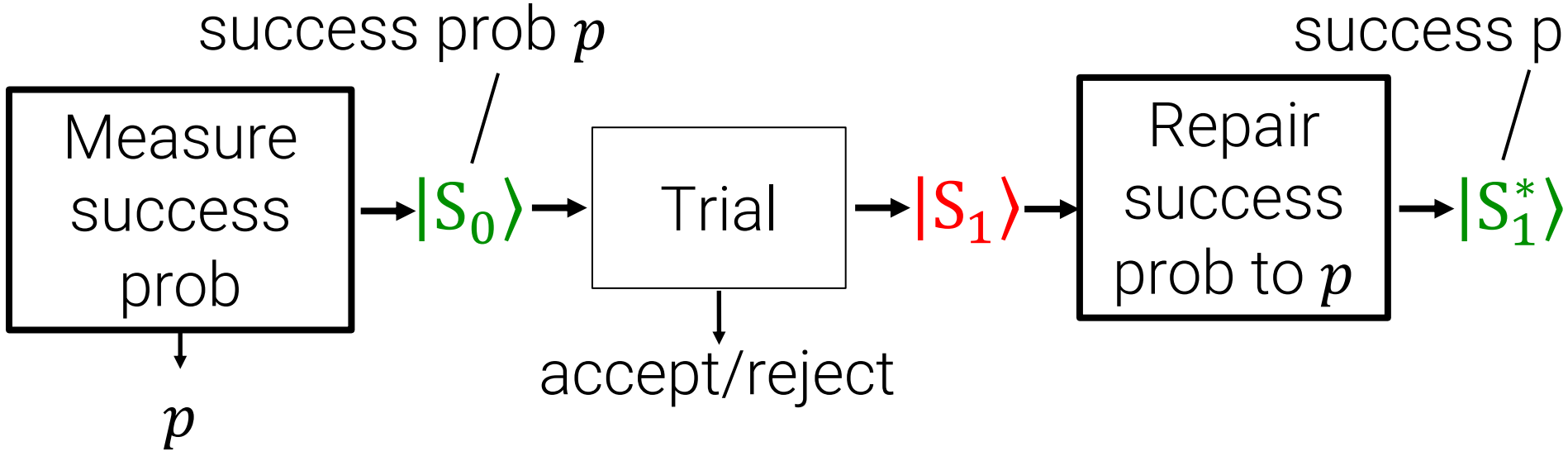
[CMSZ21] State Repair

If $|S_1\rangle$ not far from V, must have non-trivial component in V.

We show: can ***amplify*** this component to output $|S_1^*\rangle \in V$

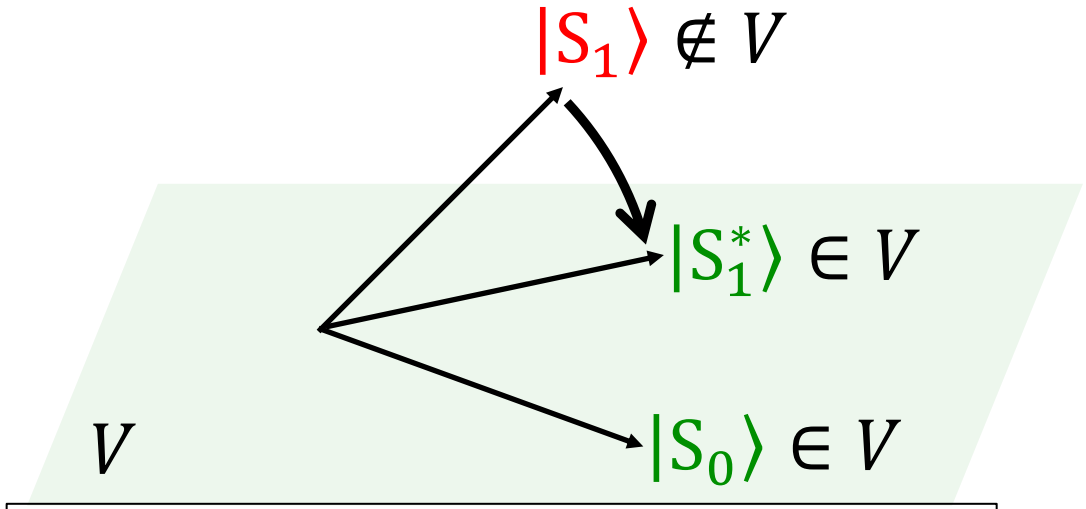


V = subspace of attacker states with success prob $\geq p$

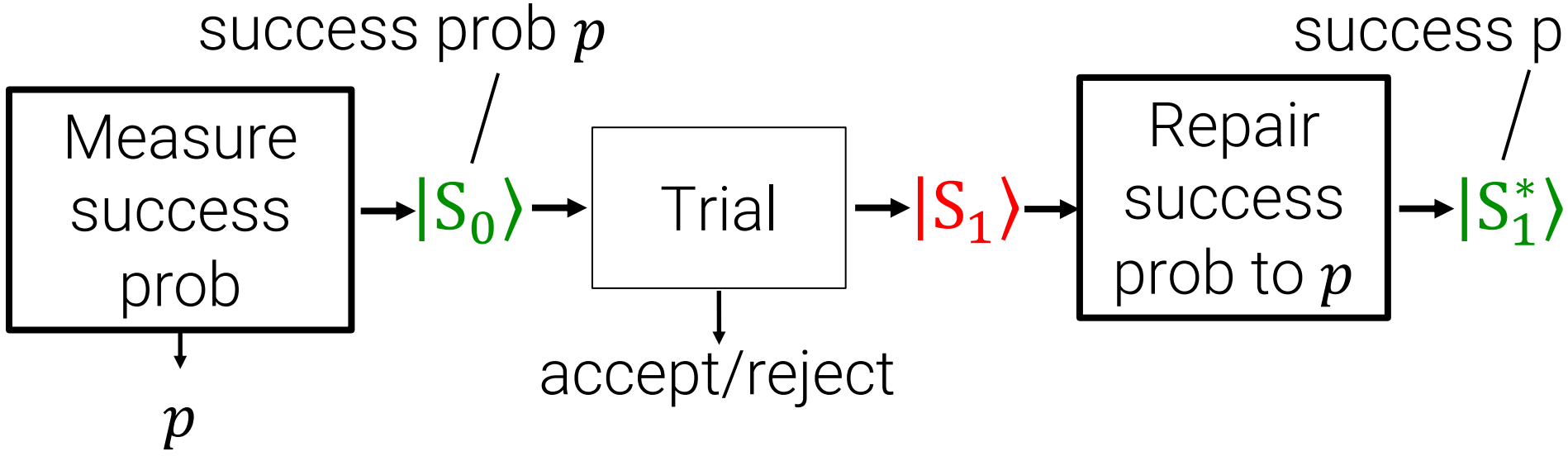


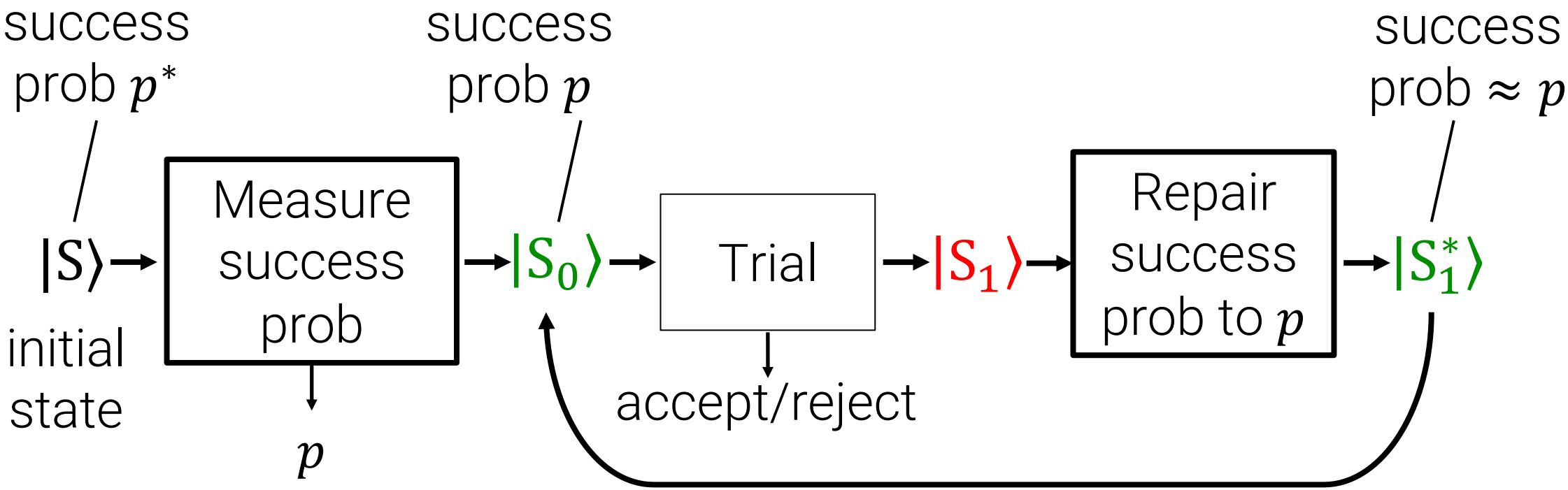
[CMSZ21] State Repair

Looks like ***amplitude amplification***, but requires care since we can only ***approximately*** project onto V.

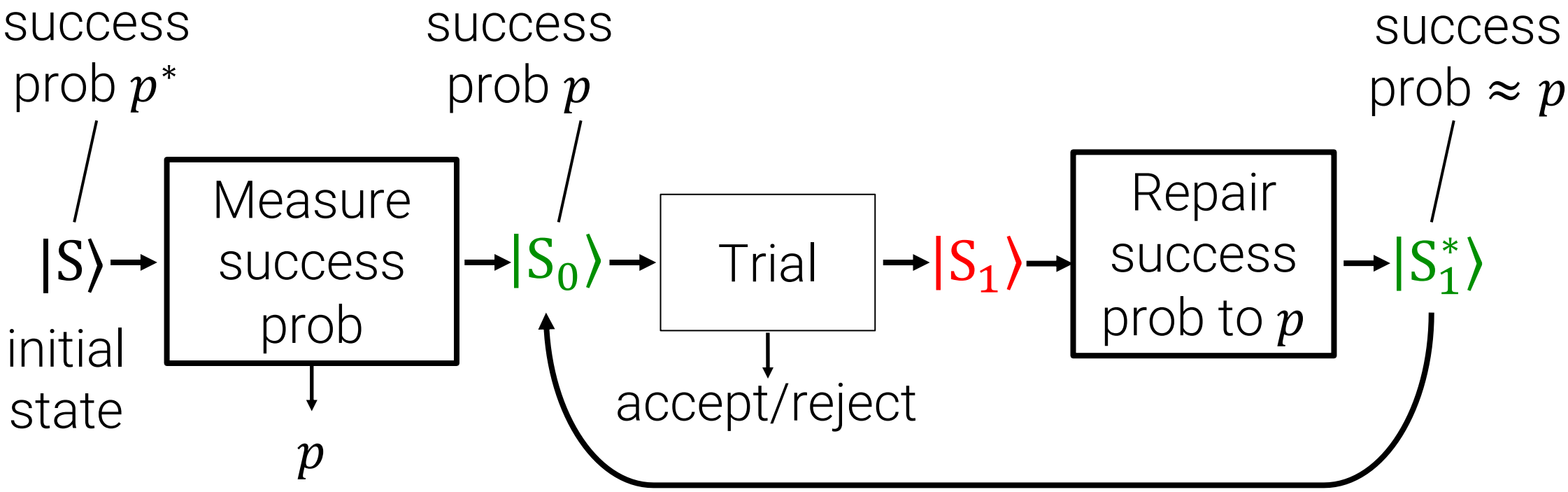


V = subspace of attacker states with success prob $\geq p$





[CMSZ21] Summary: our quantum rewinding approach extends many classical reductions to the quantum setting.



[CMSZ21] Summary: our quantum rewinding approach extends many classical reductions to the quantum setting.

Consequences: post-quantum succinct arguments + optimal post-quantum security guarantees for other protocols

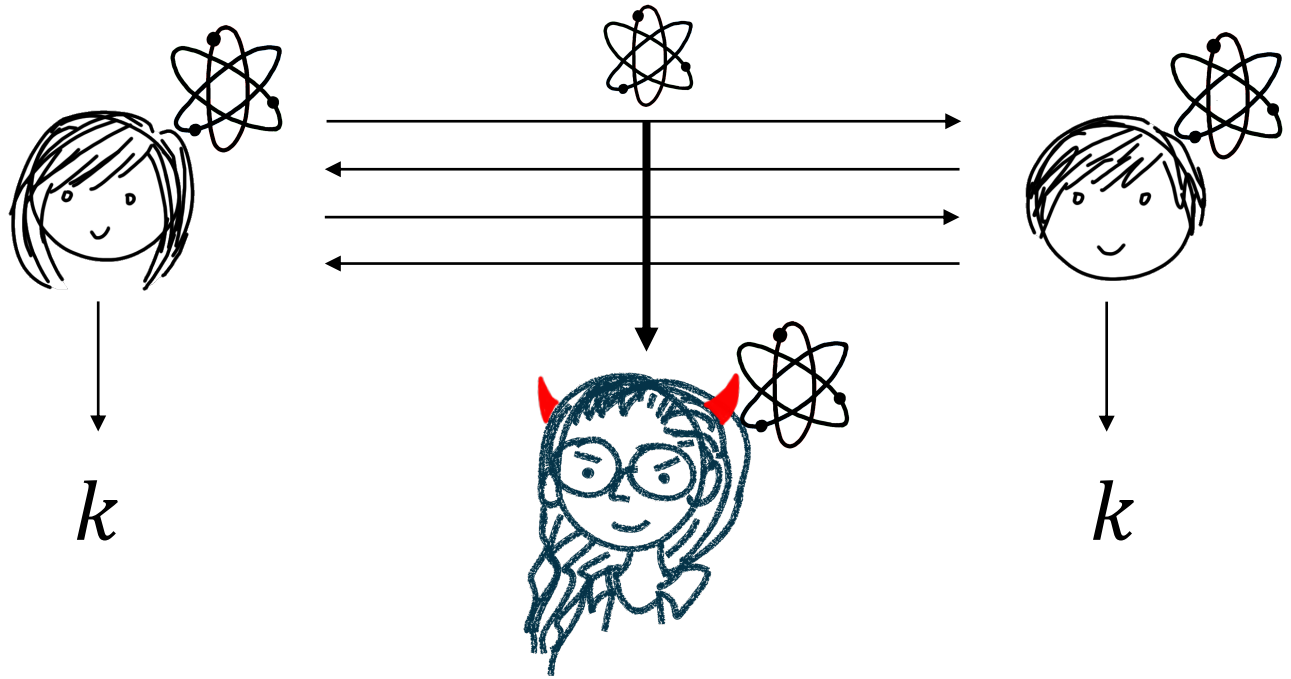
So far, we've considered quantum **adversaries**.

So far, we've considered quantum **adversaries**.

But in the long-term, even **honest parties** may possess quantum computers.

[BB84] Quantum Key Distribution

- Quantum parties + quantum channel agree on key
- Security is *information-theoretic* (i.e., no assumptions)



After [BB84], significant efforts (e.g. [BCJL93]) to build more information-theoretic quantum crypto.

After [BB84], significant efforts (e.g. [BCJL93]) to build more information-theoretic quantum crypto.

The hope:

Use quantum to build crypto without assumptions!

Bad news [M97,LC97]: Information-theoretic quantum bit commitments are impossible.

Bad news [M97,LC97]: Information-theoretic quantum bit commitments are impossible.

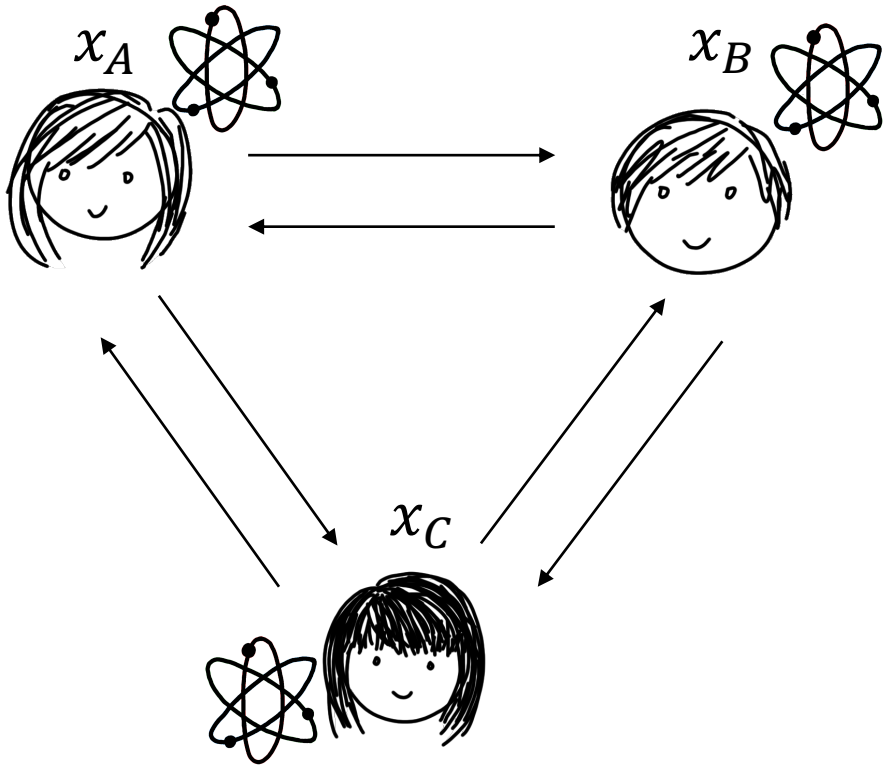
Does this mean for most crypto tasks, quantum information doesn't help?

Bad news [M97,LC97]: Information-theoretic quantum bit commitments are impossible.

Does this mean for most crypto tasks, quantum information doesn't help?

Not necessarily! May be possible to use quantum information to build crypto from *weaker assumptions*.

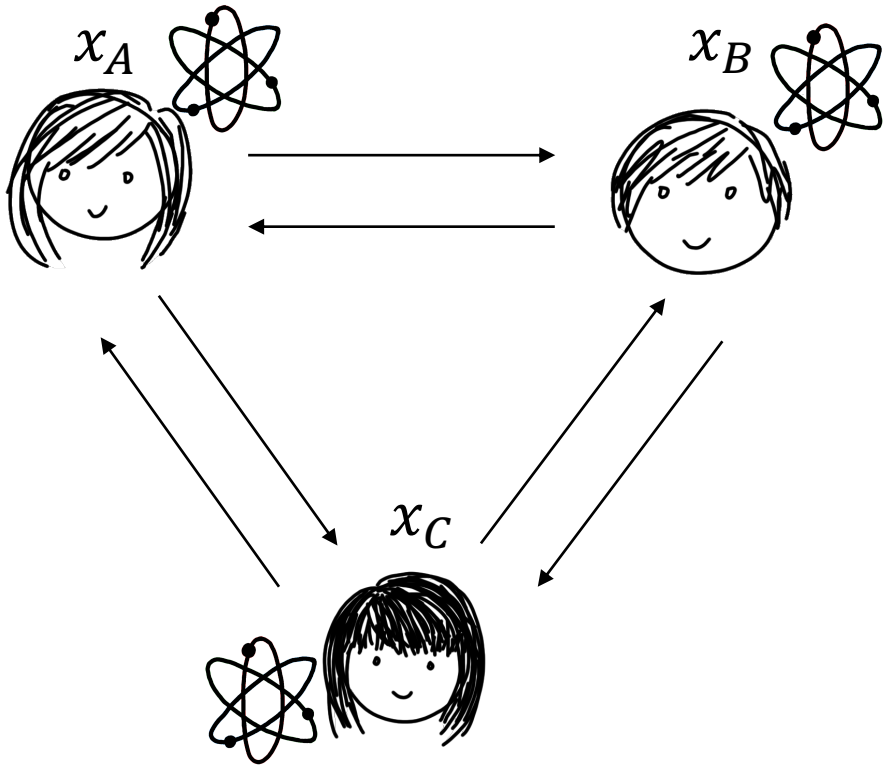
Multi-Party Computation (MPC) from One-Way Functions [BCKM21a]



Goal: Learn joint function $f(x_A, x_B, x_C)$ on private inputs

Security: reveal nothing else about x_A, x_B, x_C

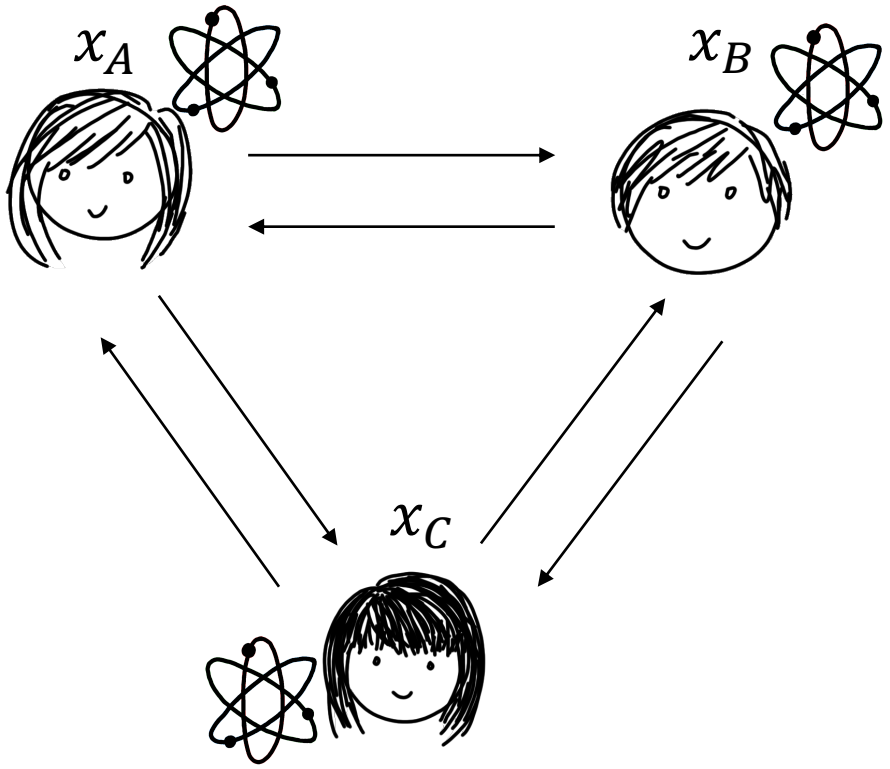
Multi-Party Computation (MPC) from One-Way Functions [BCKM21a]



[Yao86,GMW87]: Classical MPC

- Equivalent to “oblivious transfer”
- Not known from one-way functions (believed impossible)

Multi-Party Computation (MPC) from One-Way Functions [BCKM21a]

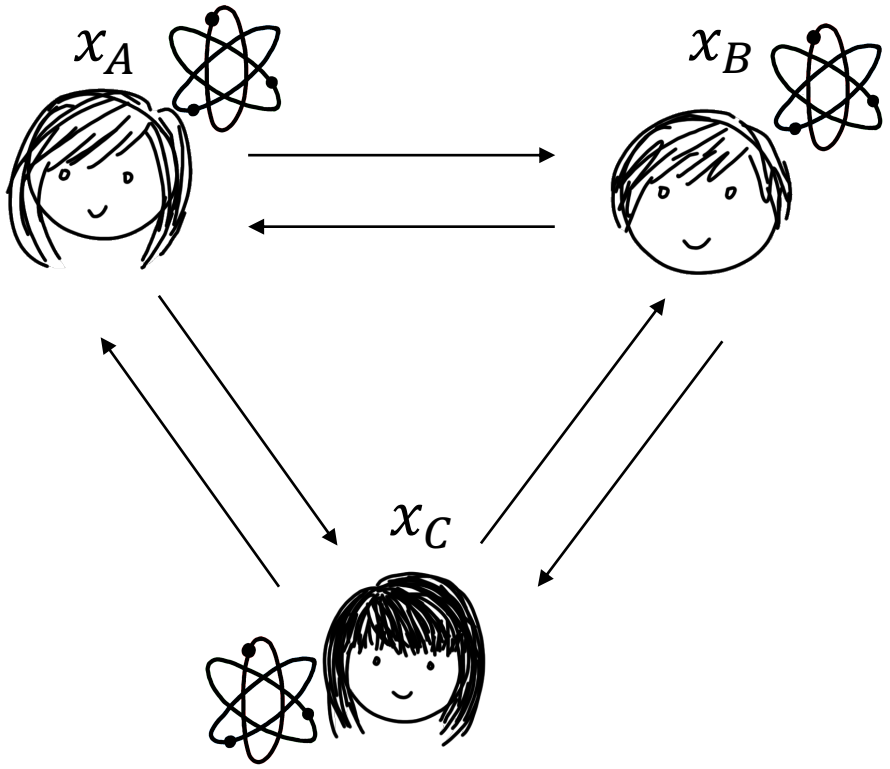


[Yao86,GMW87]: Classical MPC

- Equivalent to "oblivious transfer"
- Not known from one-way functions (believed impossible)

[BCKM21a]/[GLSV21]: Quantum MPC implied by one-way functions!

Multi-Party Computation (MPC) from One-Way Functions [BCKM21a]

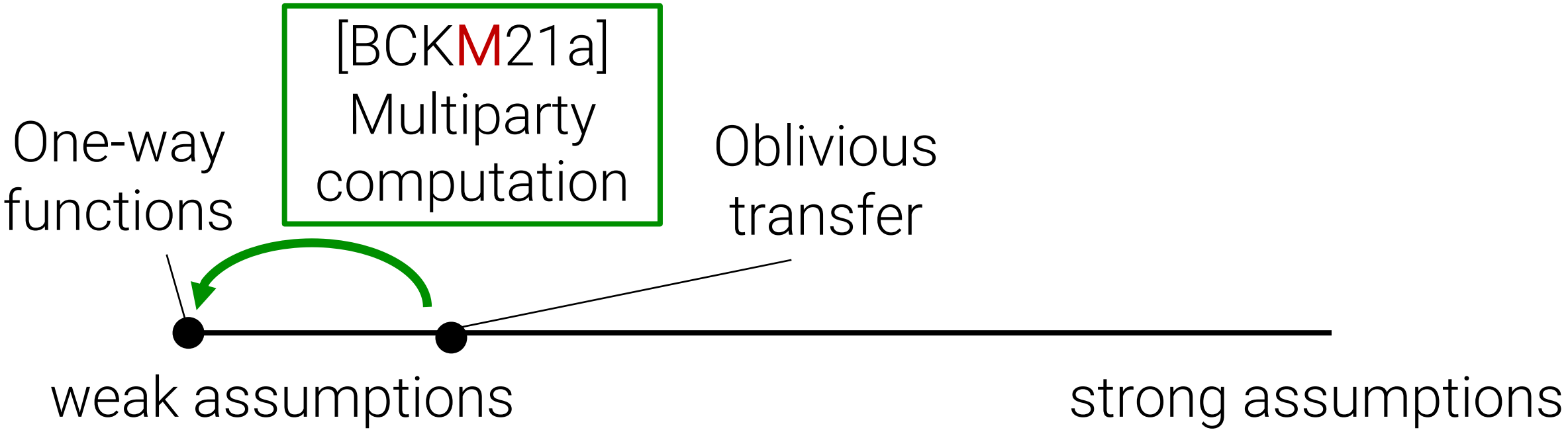


[Yao86,GMW87]: Classical MPC

- Equivalent to "oblivious transfer"
- Not known from one-way functions (believed impossible)

[BCKM21a]/[GLSV21]: Quantum MPC implied by one-way functions!

This is a prime example of quantum information enabling crypto from **weaker assumptions**



This is a prime example of quantum information enabling crypto from **weaker assumptions**

Recap

[CMSZ21]: Obtain quantum analogue of classical rewinding and prove Kilian's protocol secure against quantum.

[BCKM21a]: Construct secure multiparty computation from one-way functions + quantum information.

Recap

[CMSZ21]: Obtain quantum analogue of classical rewinding and prove Kilian's protocol secure against quantum.

[BCKM21a]: Construct secure multiparty computation from one-way functions + quantum information.

Upcoming work [LMS21]:

- obtain **state-preserving** version of [CMSZ21] rewinding
- one application: the [GMW86] graph non-isomorphism protocol is zero-knowledge against quantum verifiers

Recap

[CMSZ21]: Obtain quantum analogue of classical rewinding and prove Kilian's protocol secure against quantum.

[BCKM21a]: Construct secure multiparty computation from one-way functions + quantum information.

Rest of thesis: new results on Fiat-Shamir

[BBHMR19]: Barriers to provably-secure succinct non-interactive arguments from Fiat-Shamir.

[CLMQ21]: Investigate whether Fiat-Shamir hash function must be "cryptographic"

Thank You!

Slide Artwork by Eysa Lee