Untangling the Security of Kilian's Protocol: Upper and Lower Bounds

Alessandro Chiesa, Marcel Dall'Agnol, Ziyi Guan, Nick Spooner, Eylon Yogev

Funded by the European Union
European Research Council
Established by the European Commission
ethereum foundation

Interactive proofs

Interactive proofs

Prover

$P(x, w)$

Interactive proofs

Prover

$P(x, w)$

Verifier

$V(x)$

Interactive proofs

Prover

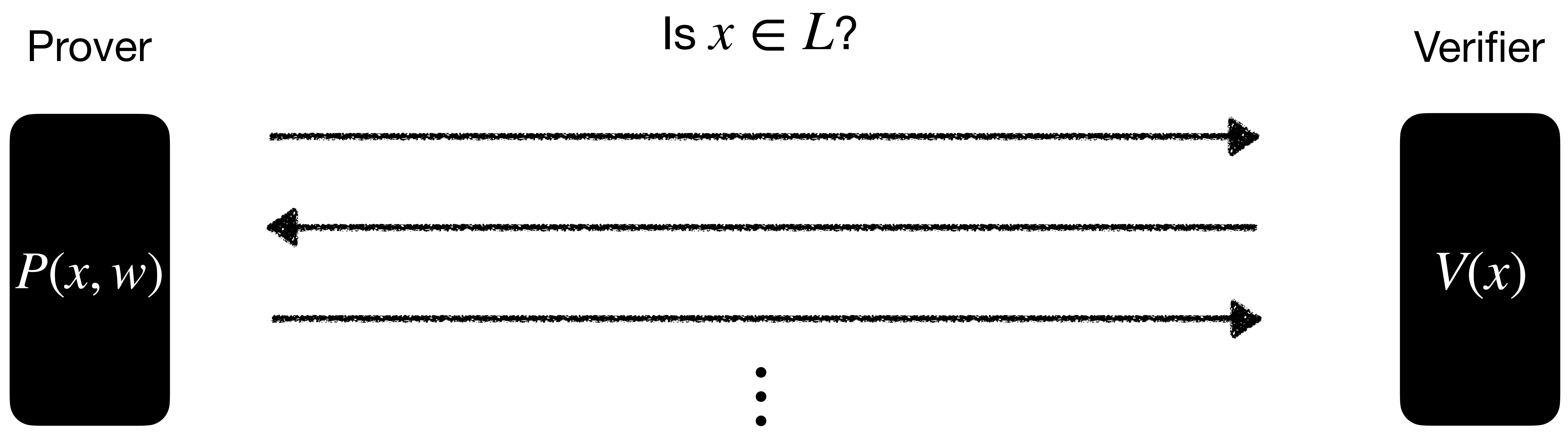
Is $x \in L$?

Verifier

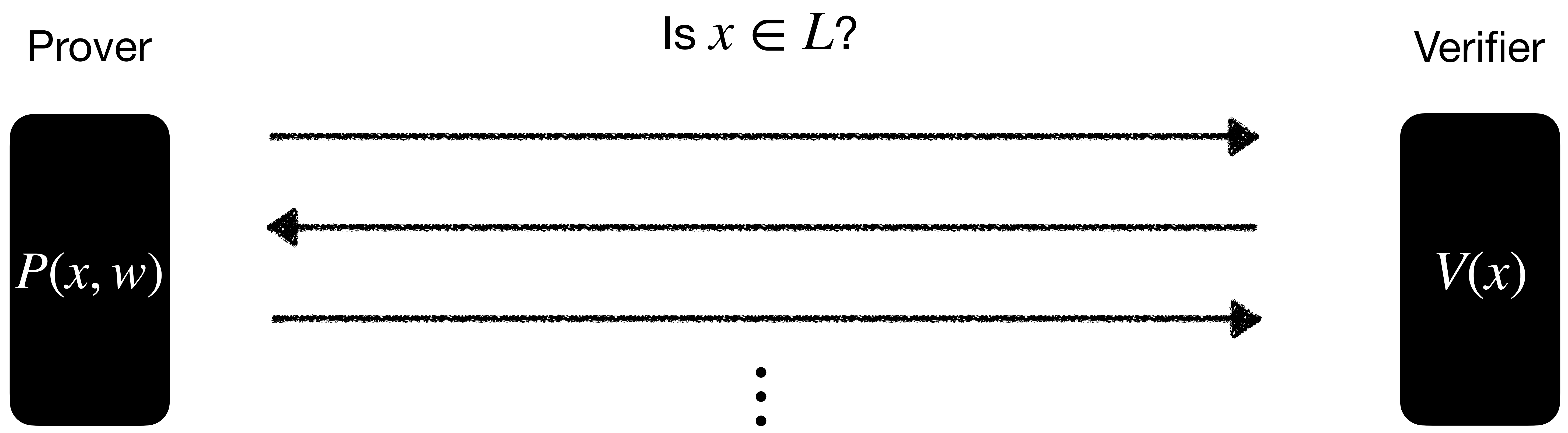
$P(x, w)$

$V(x)$

Interactive proofs

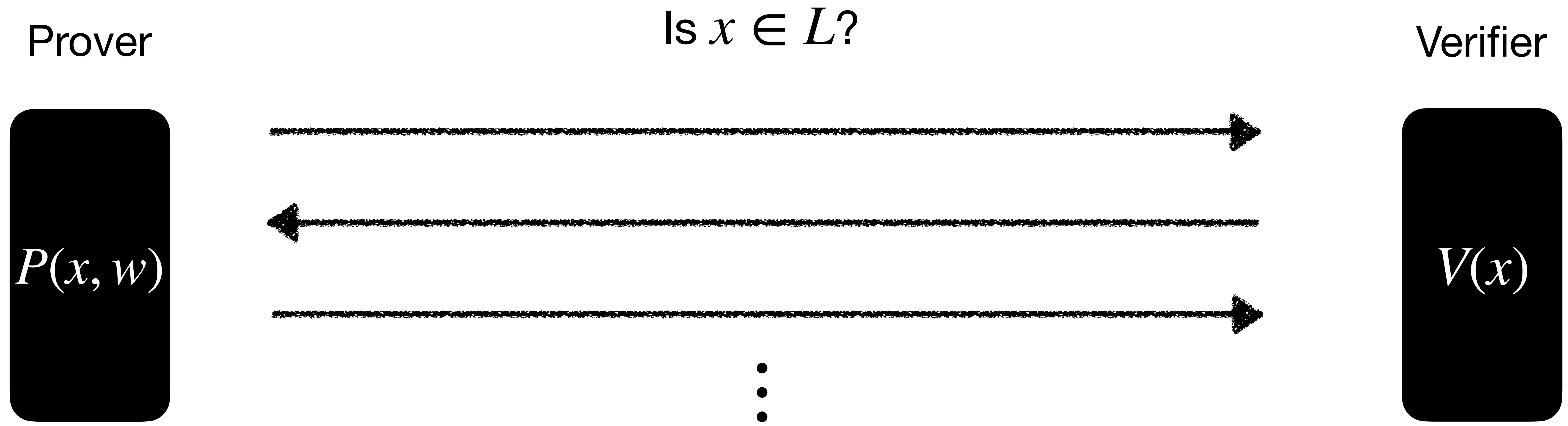


Interactive proofs



Perfect completeness: For every instance $x \in L$,

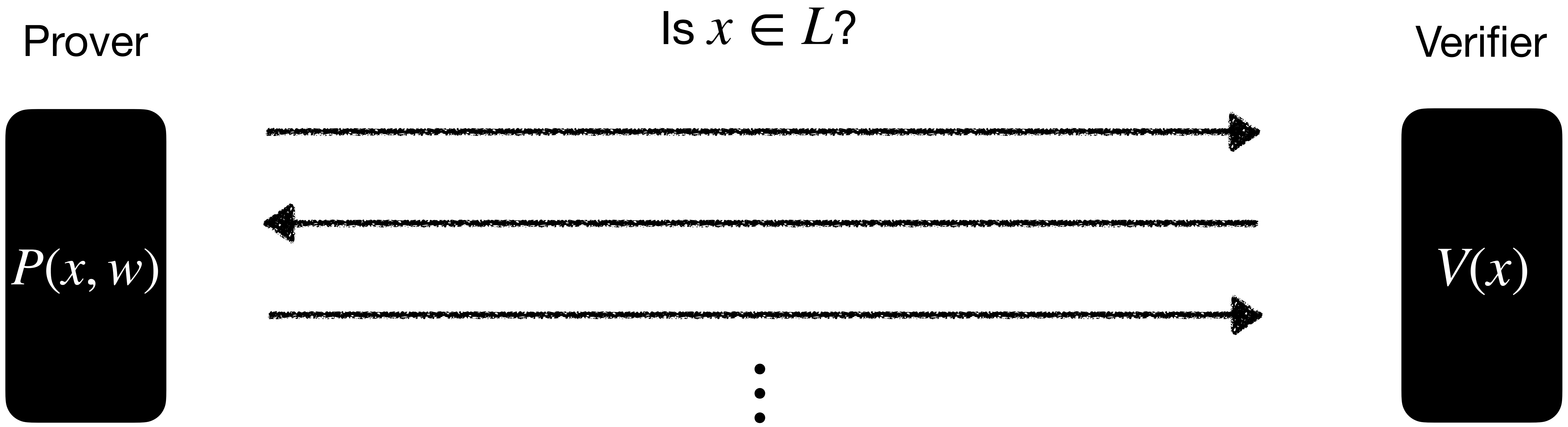
Interactive proofs



Perfect completeness: For every instance $x \in L$,

$$\Pr\left[\langle P(x,w), V(x)\rangle = 1\right] = 1.$$

Interactive proofs

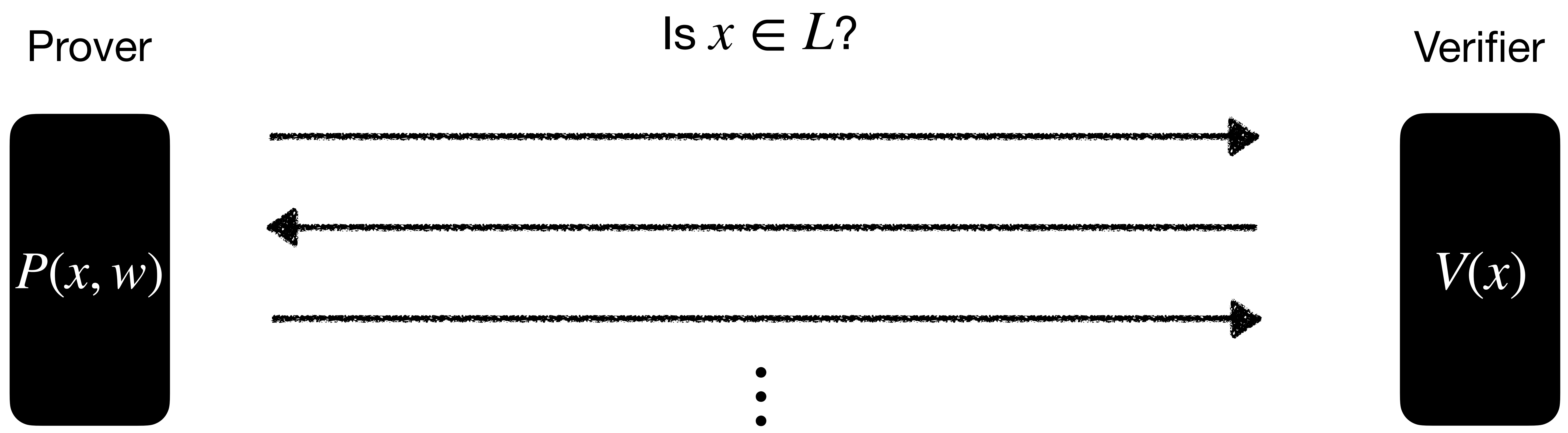


Perfect completeness: For every instance $x \in L$,

$$\Pr\left[\langle P(x,w), V(x)\rangle = 1\right] = 1.$$

Soundness: For every instance $x \notin L$ and adversary $\tilde{P}$,

Interactive proofs



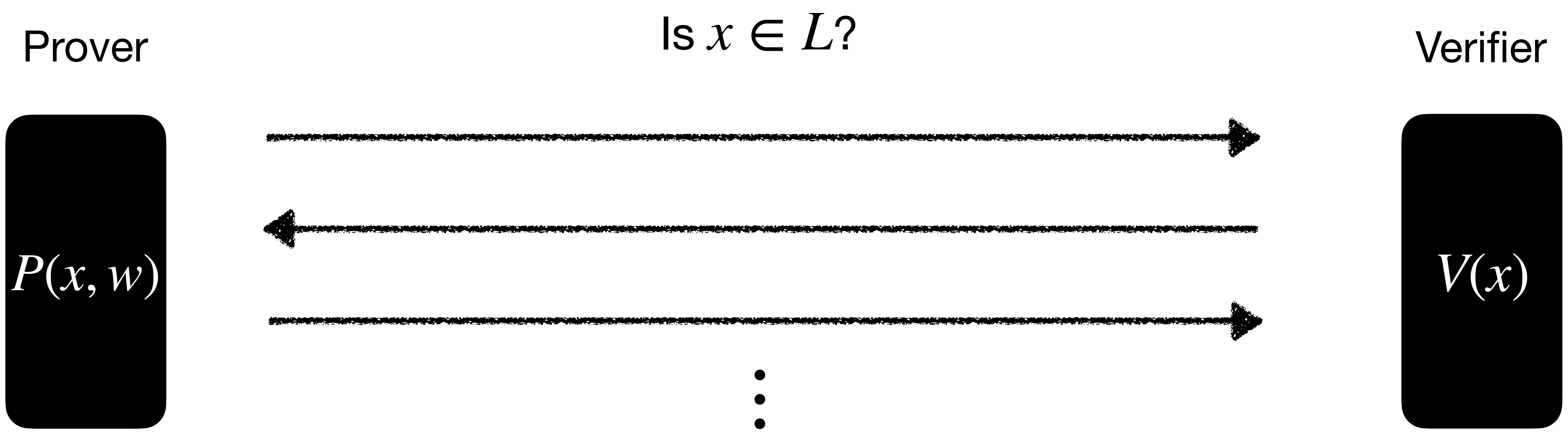
Perfect completeness: For every instance $x \in L$,

$$\Pr\left[\langle P(x,w), V(x)\rangle = 1\right] = 1.$$

Soundness: For every instance $x \notin L$ and adversary $\tilde{P}$,

$$\Pr\left[\langle \tilde{P}, V(x)\rangle = 1\right] \leq \epsilon(x).$$

Interactive proofs



Perfect completeness: For every instance $x \in L$,

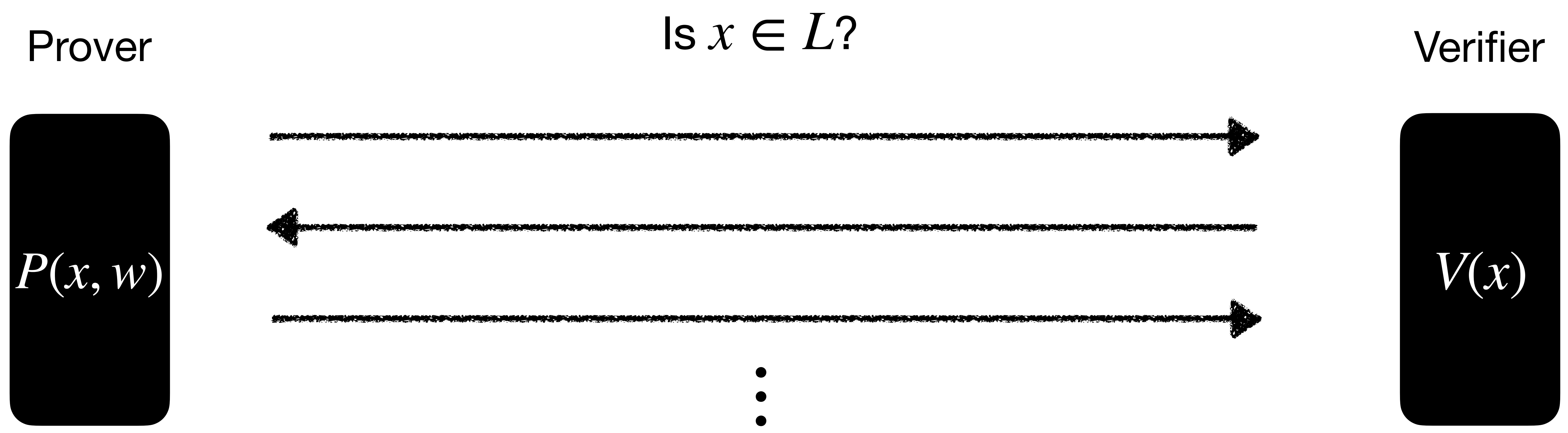
$$\Pr\left[\langle P(x,w), V(x)\rangle = 1\right] = 1.$$

Soundness: For every instance $x \notin L$ and adversary $\tilde{P}$,

$$\Pr\left[\langle \tilde{P}, V(x)\rangle = 1\right] \leq \epsilon(x).$$

Basic efficiency metric: **COMMUNICATION COMPLEXITY** (number of bits exchanged during the interaction).

Interactive proofs



Perfect completeness: For every instance $x \in L$,

$$\Pr\left[\langle P(x,w), V(x)\rangle = 1\right] = 1.$$

Soundness: For every instance $x \notin L$ and adversary $\tilde{P}$,

$$\Pr\left[\langle \tilde{P}, V(x)\rangle = 1\right] \leq \epsilon(x).$$

Basic efficiency metric: **COMMUNICATION COMPLEXITY** (number of bits exchanged during the interaction).

Limitation: NP-complete languages do not have IPs with cc $\ll |w|$ (or else the language would be easy).

(Indeed, [GH97] proved that, in general, $\mathrm{IP}[\mathrm{cc}] \subseteq \mathrm{BPTIME}[2^{\mathrm{cc}}]$.)

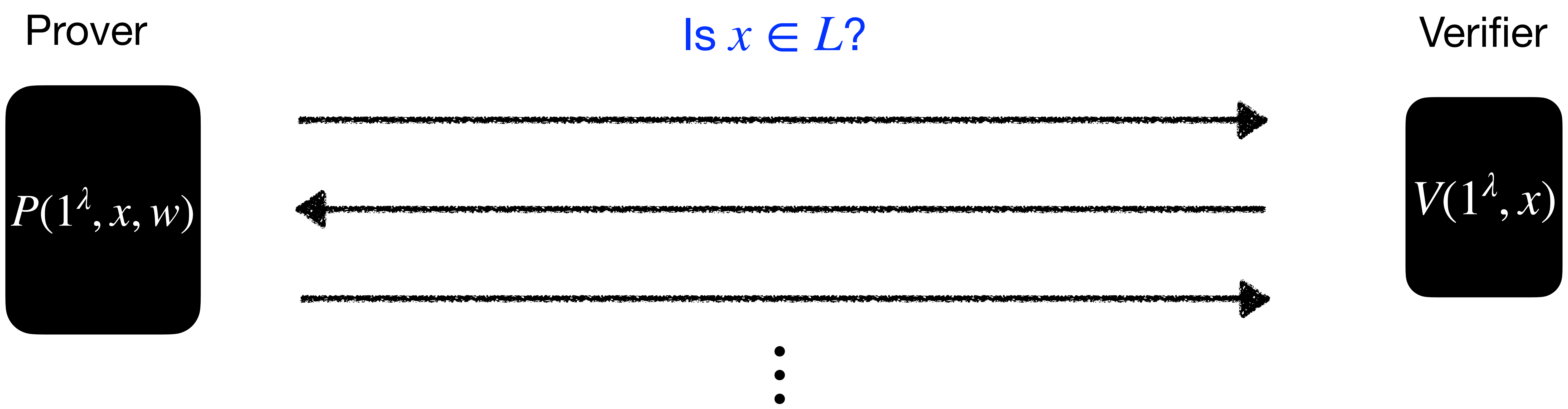
Interactive arguments

Interactive arguments

Interactive proofs with computational soundness

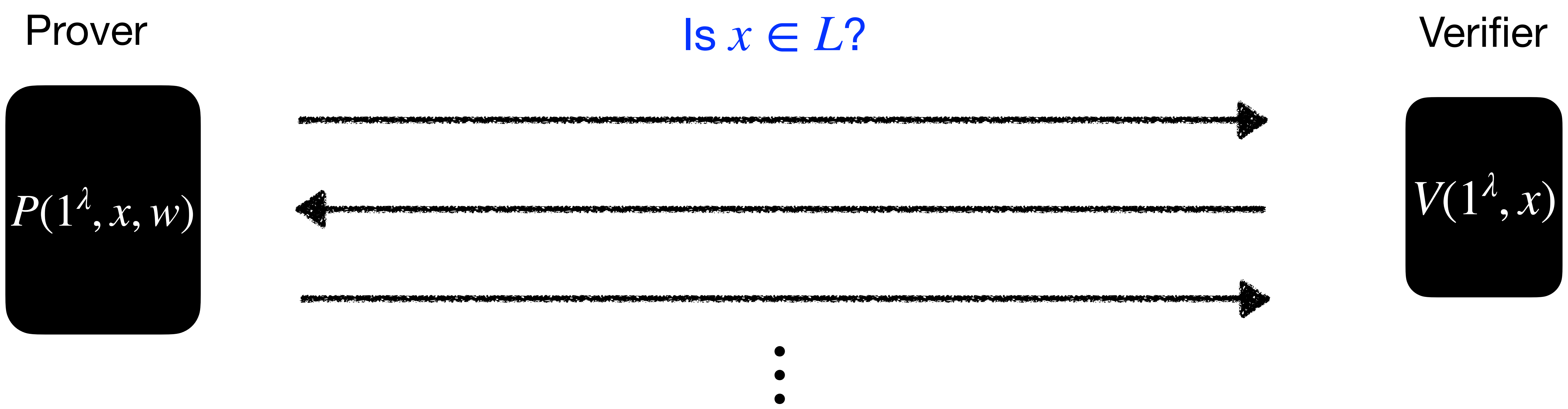
Interactive arguments

Interactive proofs with computational soundness



Interactive arguments

Interactive proofs with computational soundness

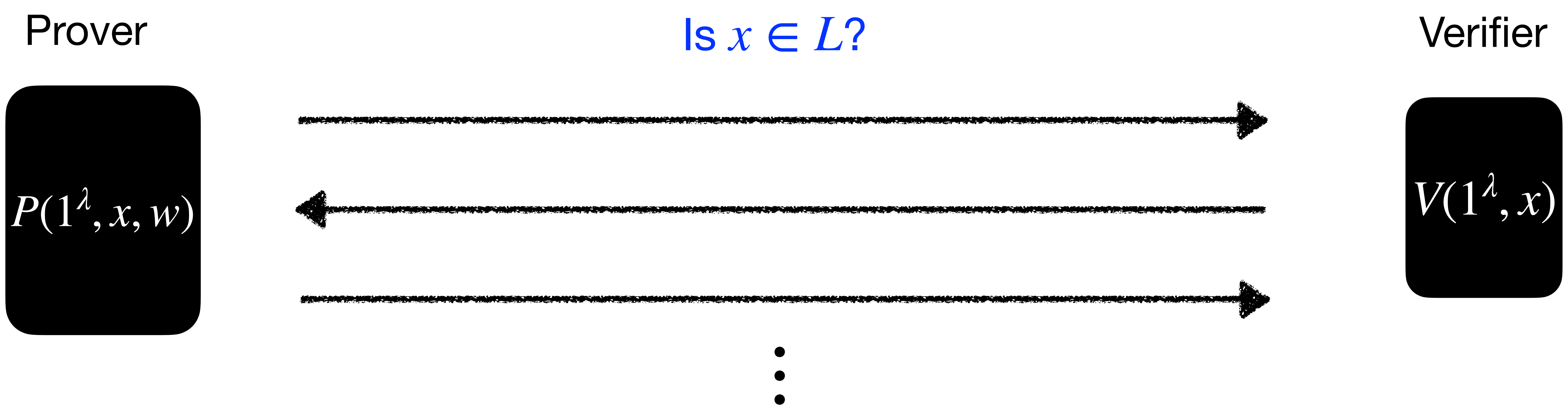


Computational soundness: For every $x \notin L$, security parameter $\lambda \in \mathbb{N}$, and t_{ARG}-bounded adversary $\tilde{P}$,

$$\Pr\left[\langle \tilde{P}, V(1^\lambda, x)\rangle = 1\right] \leq \epsilon_{\mathsf{ARG}}(\lambda, x, t_{\mathsf{ARG}}).$$

Interactive arguments

Interactive proofs with computational soundness



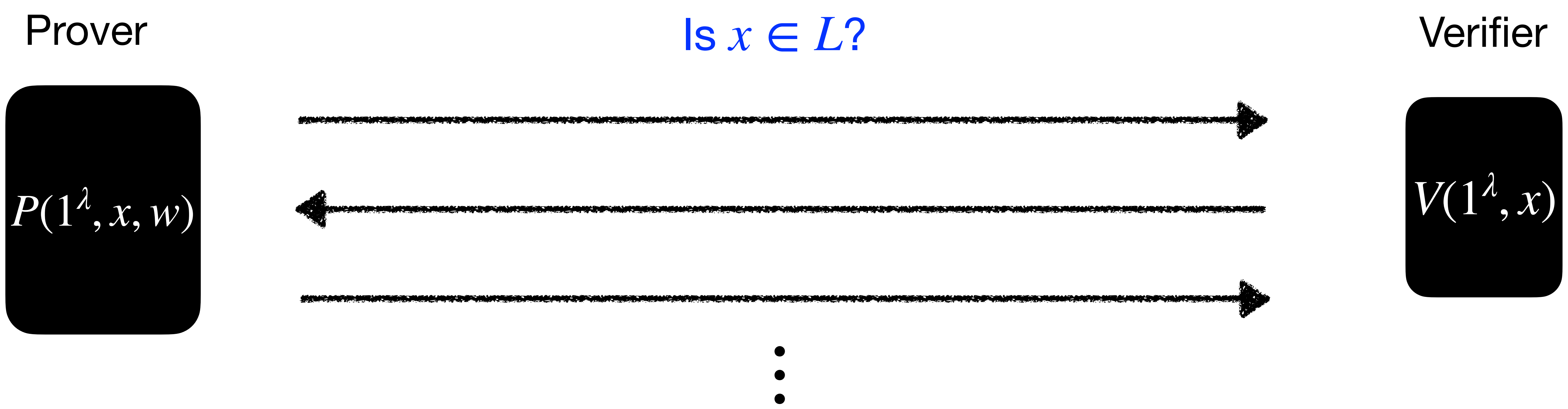
relaxes the soundness guarantee of interactive proofs

Computational soundness: For every $x \notin L$, security parameter $\lambda \in \mathbb{N}$, and t_{ARG}-bounded adversary $\tilde{P}$,

$$\Pr\left[\langle \tilde{P}, V(1^\lambda, x)\rangle = 1\right] \leq \epsilon_{\mathsf{ARG}}(\lambda, x, t_{\mathsf{ARG}}).$$

Interactive arguments

Interactive proofs with computational soundness



relaxes the soundness guarantee of interactive proofs

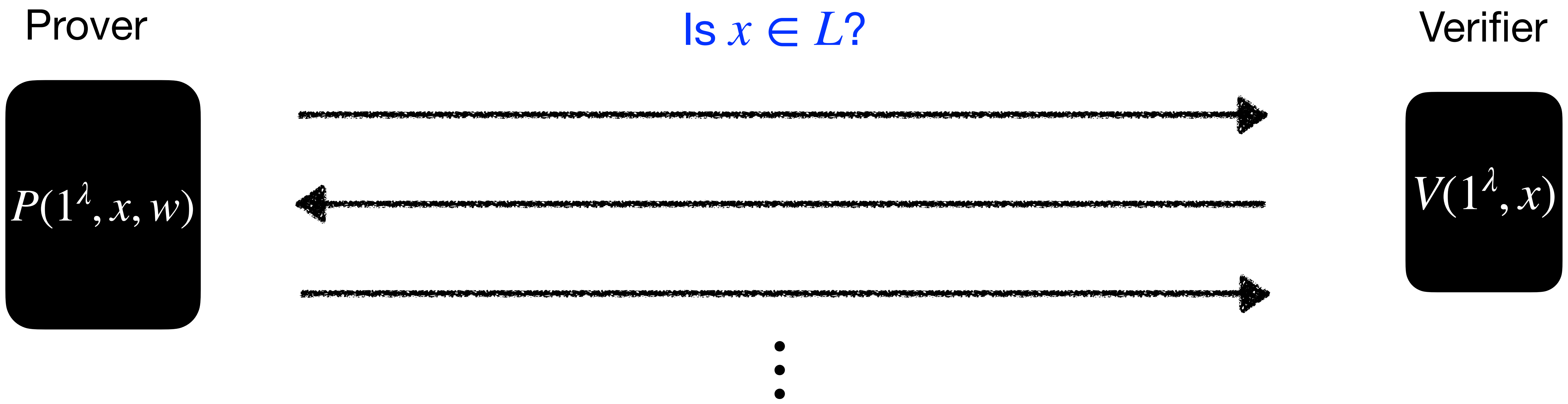
Computational soundness: For every $x \notin L$, security parameter $\lambda \in \mathbb{N}$, and t_{ARG}-bounded adversary $\tilde{P}$,

$$\Pr\left[\langle \tilde{P}, V(1^\lambda, x)\rangle = 1\right] \le \epsilon_{\mathsf{ARG}}(\lambda, x, t_{\mathsf{ARG}}).$$

Limitations on the communication complexity of interactive proofs no longer hold.

Interactive arguments

Interactive proofs with computational soundness



relaxes the soundness guarantee of interactive proofs

Computational soundness: For every $x \notin L$, security parameter $\lambda \in \mathbb{N}$, and t_{ARG}-bounded adversary $\tilde{P}$,

$$\Pr\left[\langle \tilde{P}, V(1^\lambda, x)\rangle = 1\right] \leq \epsilon_{\mathsf{ARG}}(\lambda, x, t_{\mathsf{ARG}}).$$

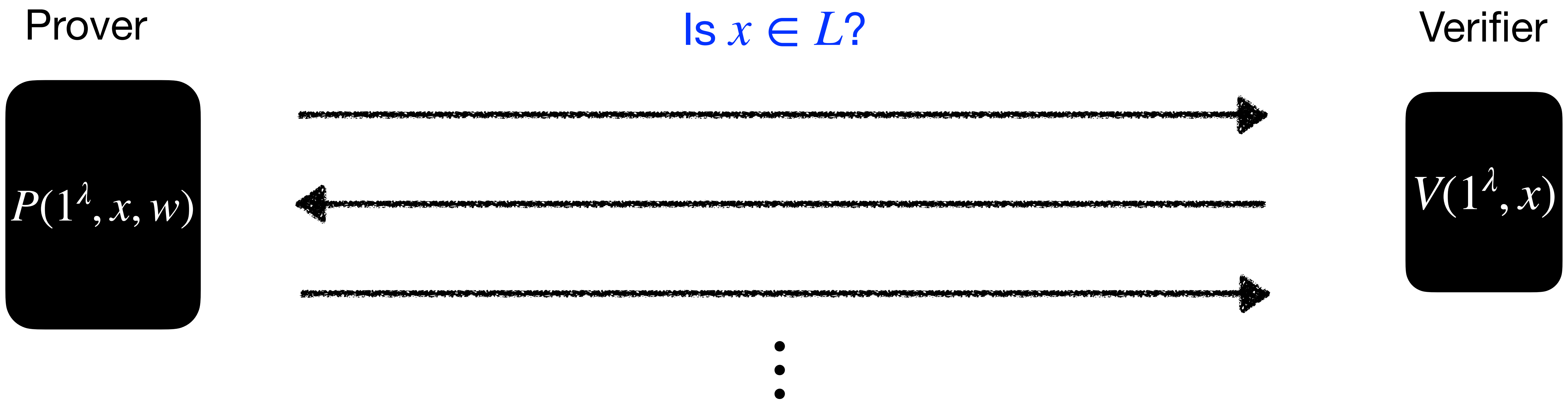
Limitations on the communication complexity of interactive proofs no longer hold.

AMAZING: there exist interactive arguments for NP with cc $\ll |w|$ (given basic cryptography)

These are known as **Succinct Interactive Arguments**.

Interactive arguments

Interactive proofs with computational soundness



relaxes the soundness guarantee of interactive proofs

Computational soundness: For every $x \notin L$, security parameter $\lambda \in \mathbb{N}$, and t_{ARG}-bounded adversary $\tilde{P}$,

$$\Pr\left[\langle \tilde{P}, V(1^\lambda, x)\rangle = 1\right] \leq \epsilon_{\text{ARG}}(\lambda, x, t_{\text{ARG}}).$$

Further relaxation: Expected-time computational soundness $\epsilon^\star_{\text{ARG}}$ against adversaries with bounded expected running time $t^\star_{\text{ARG}}$.

Limitations on the communication complexity of interactive proofs no longer hold.

AMAZING: there exist interactive arguments for NP with $\text{cc} \ll |w|$ (given basic cryptography)

These are known as **Succinct Interactive Arguments**.

Why study succinct interactive arguments?

Why study succinct interactive arguments?

A **fundamental primitive** known to exist assuming only simple cryptography (e.g. collision-resistant hash functions).

Why study succinct interactive arguments?

A **fundamental primitive** known to exist assuming only simple cryptography (e.g. collision-resistant hash functions).

The savings in communication ($\mathsf{cc} \ll |w|$) or even verification ($\mathsf{time}(V) \ll |w|$) are remarkably useful.

Why study succinct interactive arguments?

A **fundamental primitive** known to exist assuming only simple cryptography (e.g. collision-resistant hash functions).

The savings in communication ($\mathsf{cc} \ll |w|$) or even verification ($\mathsf{time}(V) \ll |w|$) are remarkably useful.

Succinct arguments play a key role in notable applications
(e.g., zero-knowledge with non-black-box simulation, malicious MPC, ...).

Why study succinct interactive arguments?

A **fundamental primitive** known to exist assuming only simple cryptography (e.g. collision-resistant hash functions).

The savings in communication ($\mathsf{cc} \ll |w|$) or even verification ($\mathsf{time}(V) \ll |w|$) are remarkably useful.

Succinct arguments play a key role in notable applications
(e.g., zero-knowledge with non-black-box simulation, malicious MPC, ...).

They also serve as a stepping stone towards succinct **non-interactive** arguments (SNARGs).

Why study succinct interactive arguments?

A **fundamental primitive** known to exist assuming only simple cryptography (e.g. collision-resistant hash functions).

The savings in communication ($\mathsf{cc} \ll |w|$) or even verification ($\mathsf{time}(V) \ll |w|$) are remarkably useful.

Succinct arguments play a key role in notable applications
(e.g., zero-knowledge with non-black-box simulation, malicious MPC, ...).

They also serve as a stepping stone towards succinct **non-interactive** arguments (SNARGs).

Recall: SNARGs for NP cannot be realized via a black-box reduction to a falsifiable assumption [GW11].

Often (though not always): SNARG = succinct interactive argument + non-falsifiable assumption / idealized model

Why study succinct interactive arguments?

A **fundamental primitive** known to exist assuming only simple cryptography (e.g. collision-resistant hash functions).

The savings in communication (cc $\ll |w|$) or even verification (time$(V) \ll |w|$) are remarkably useful.

Succinct arguments play a key role in notable applications
(e.g., zero-knowledge with non-black-box simulation, malicious MPC, ...).

They also serve as a stepping stone towards succinct **non-interactive** arguments (SNARGs).

Recall: SNARGs for NP cannot be realized via a black-box reduction to a falsifiable assumption [GW11].

Often (though not always): SNARG = succinct interactive argument + non-falsifiable assumption / idealized model

Kilian's protocol, the first and simplest succinct argument

Kilian’s protocol

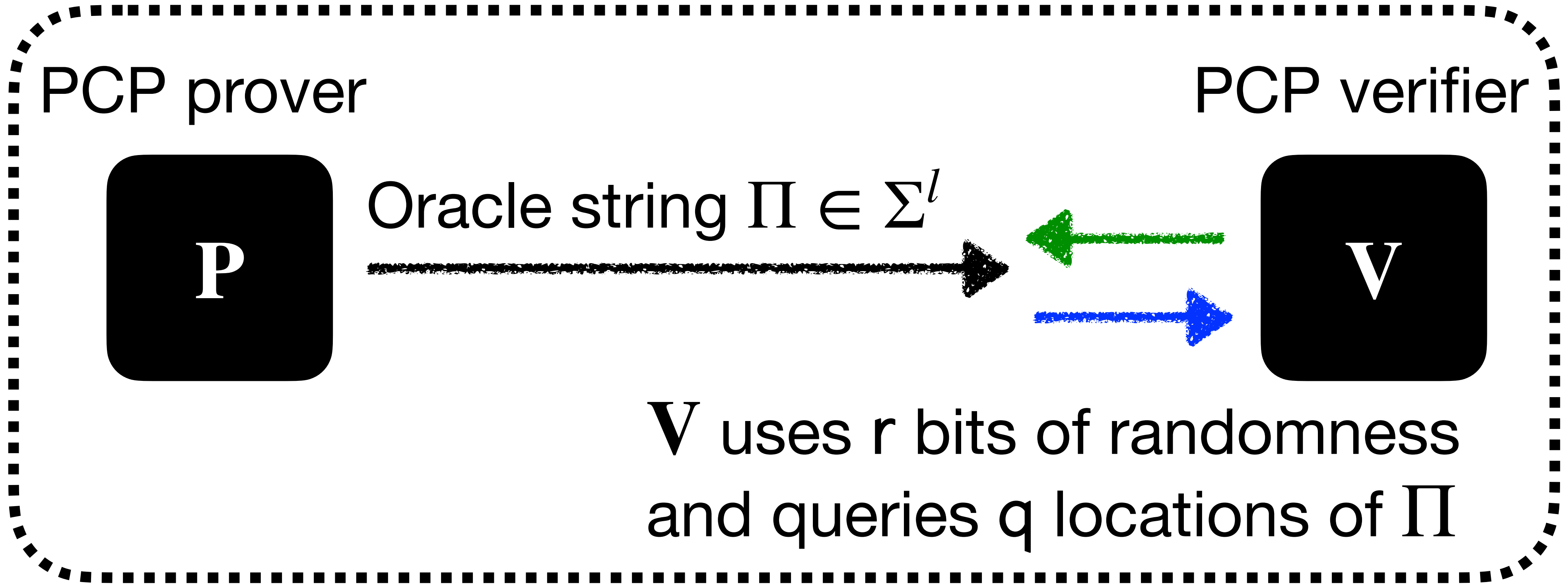
[Kilian92]

Kilian's protocol

Building block #1: probabilistically checkable proof (PCP)

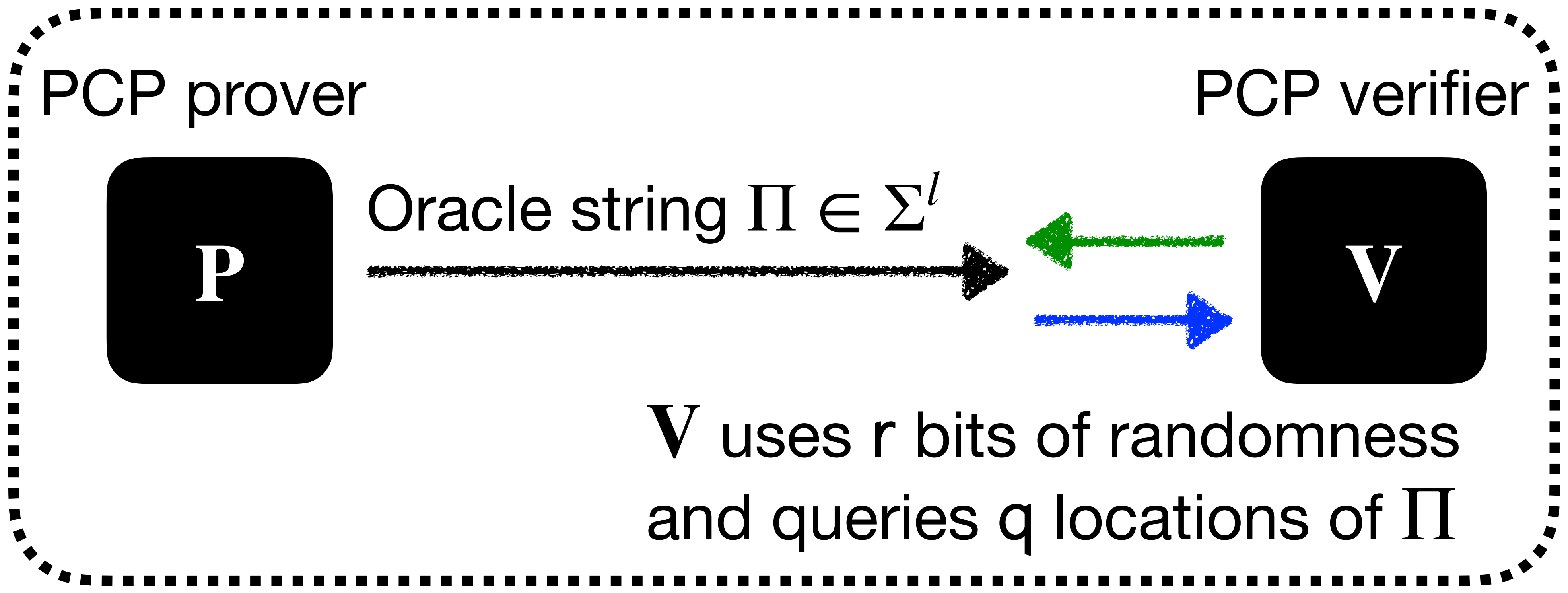
Kilian's protocol

Building block #1: probabilistically checkable proof (PCP)



Kilian's protocol

Building block #1: probabilistically checkable proof (PCP)



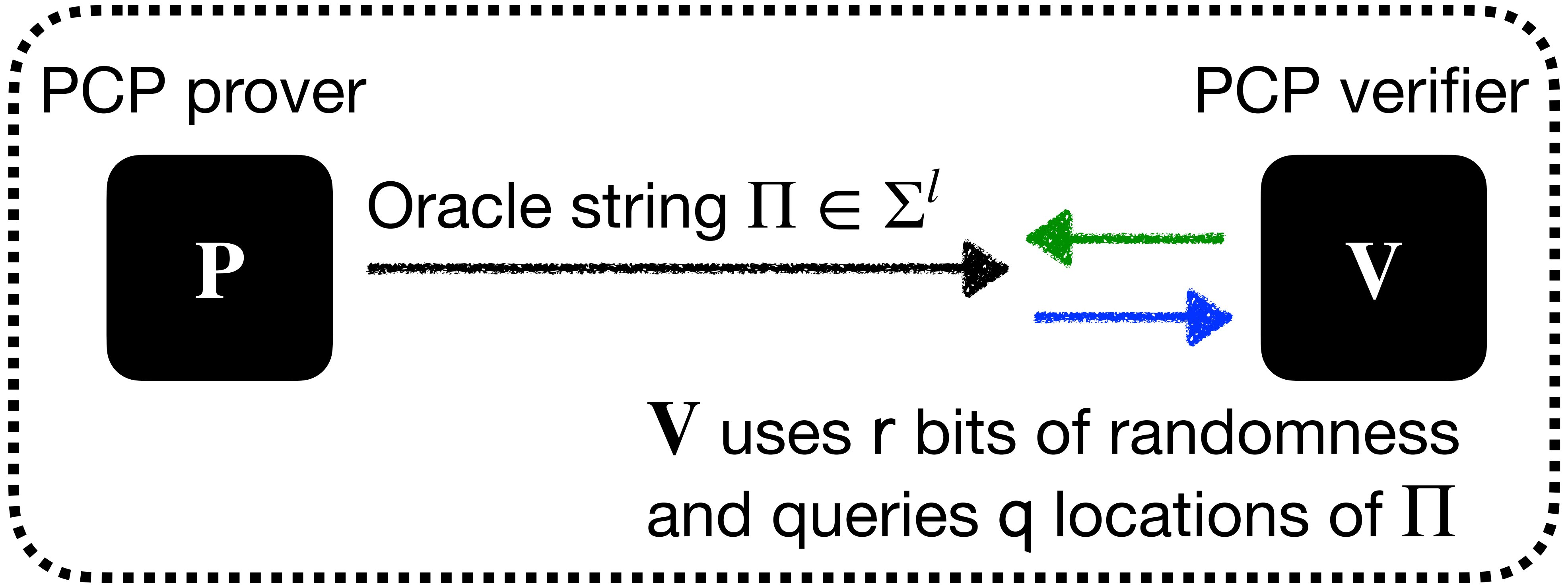
Building block #2: vector commitment scheme (VC)

[Kilian92]

Kilian's protocol

abstraction for a succinct commitment with local openings (e.g. Merkle tree)

Building block #1: probabilistically checkable proof (PCP)

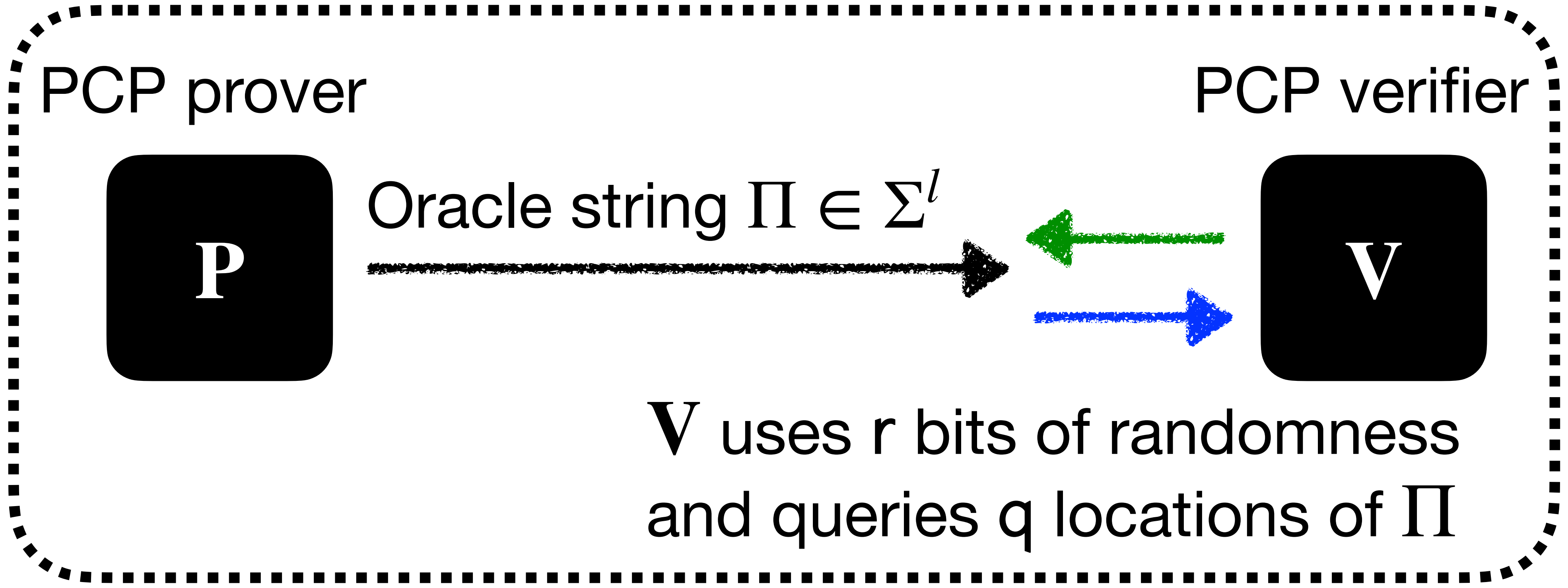


Building block #2: vector commitment scheme (VC)

Kilian’s protocol

abstraction for a succinct commitment with local openings (e.g. Merkle tree)

Building block #1: probabilistically checkable proof (PCP)



Building block #2: vector commitment scheme (VC)

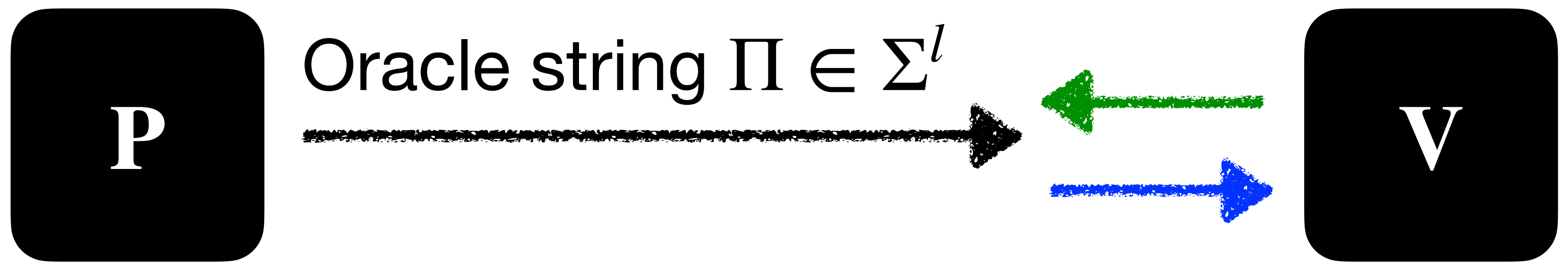
Kilian's protocol

abstraction for a succinct commitment with local openings (e.g. Merkle tree)

Building block #1: probabilistically checkable proof (PCP)

PCP prover

PCP verifier



$\mathbf{V}$ uses r bits of randomness and queries q locations of Π

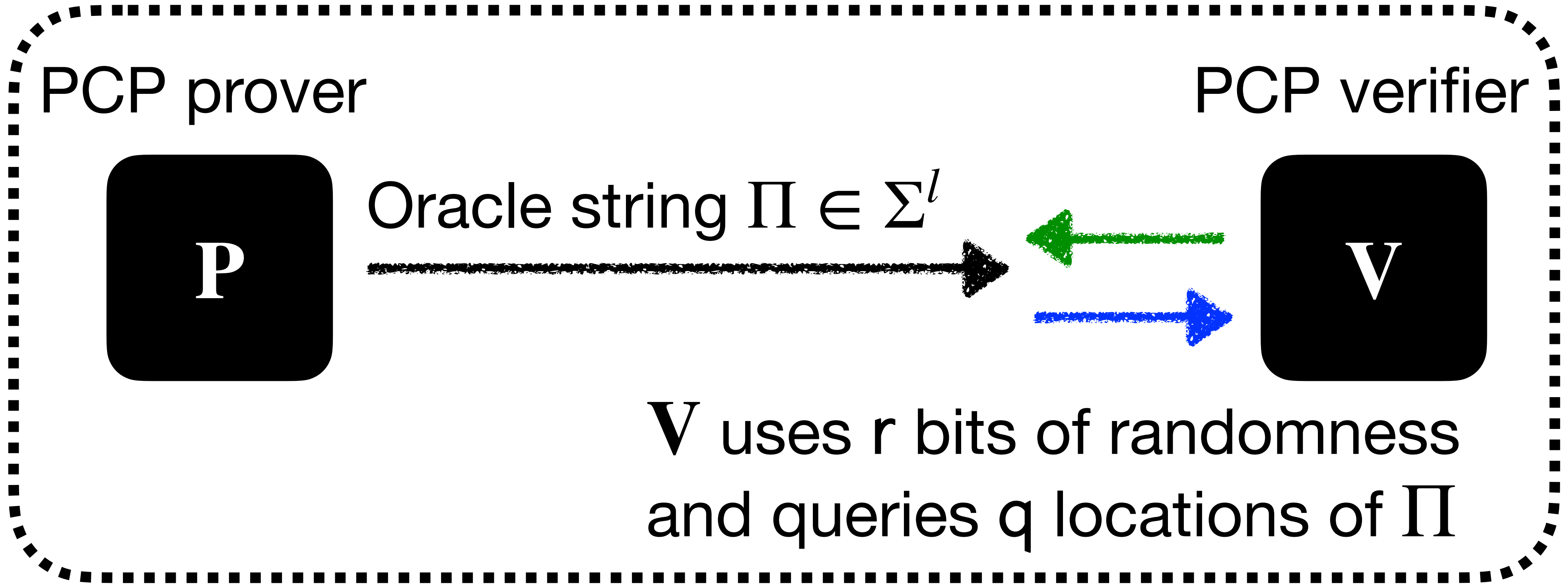
Building block #2: vector commitment scheme (VC)

Commit phase

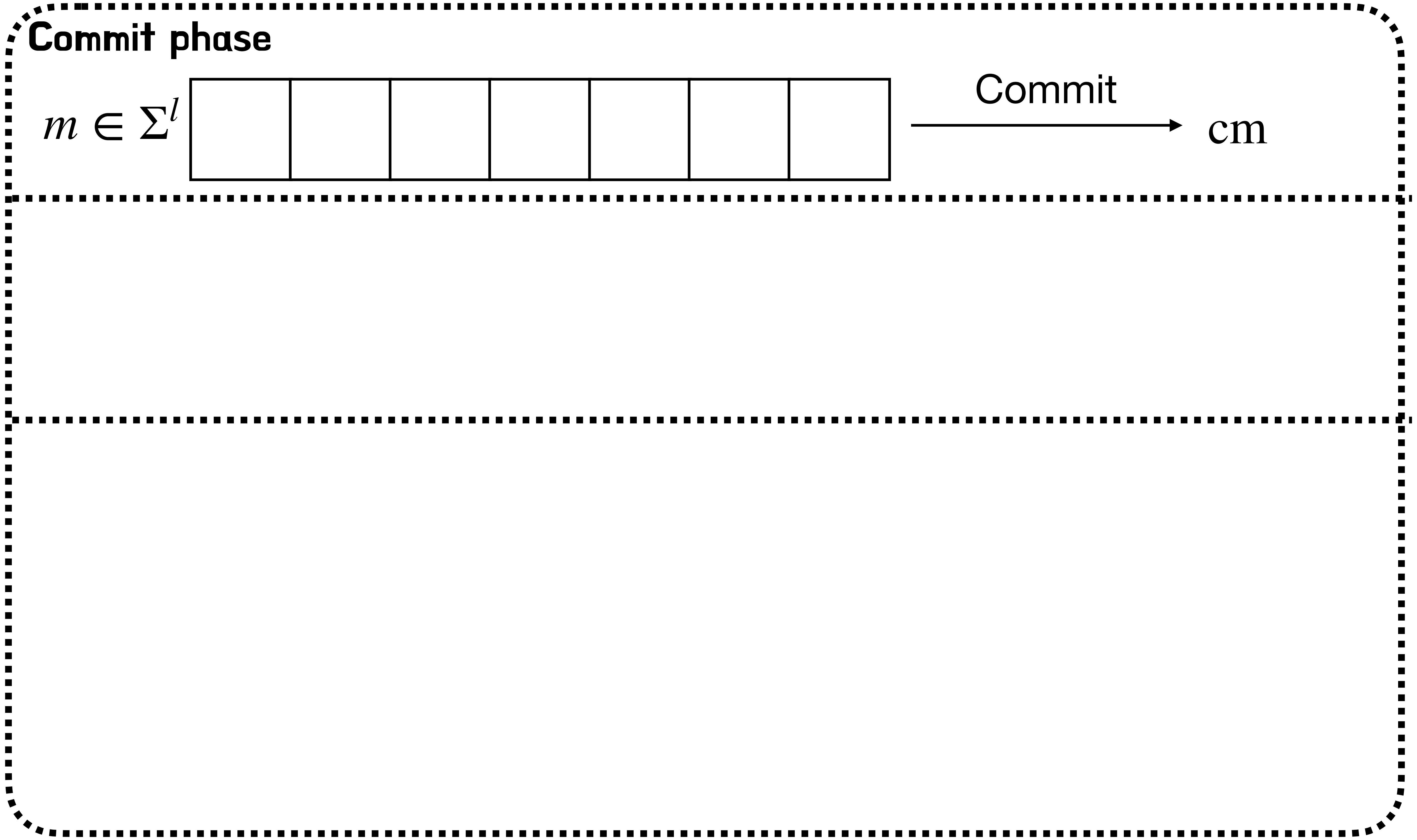
Kilian's protocol

abstraction for a succinct commitment with local openings (e.g. Merkle tree)

Building block #1: probabilistically checkable proof (PCP)



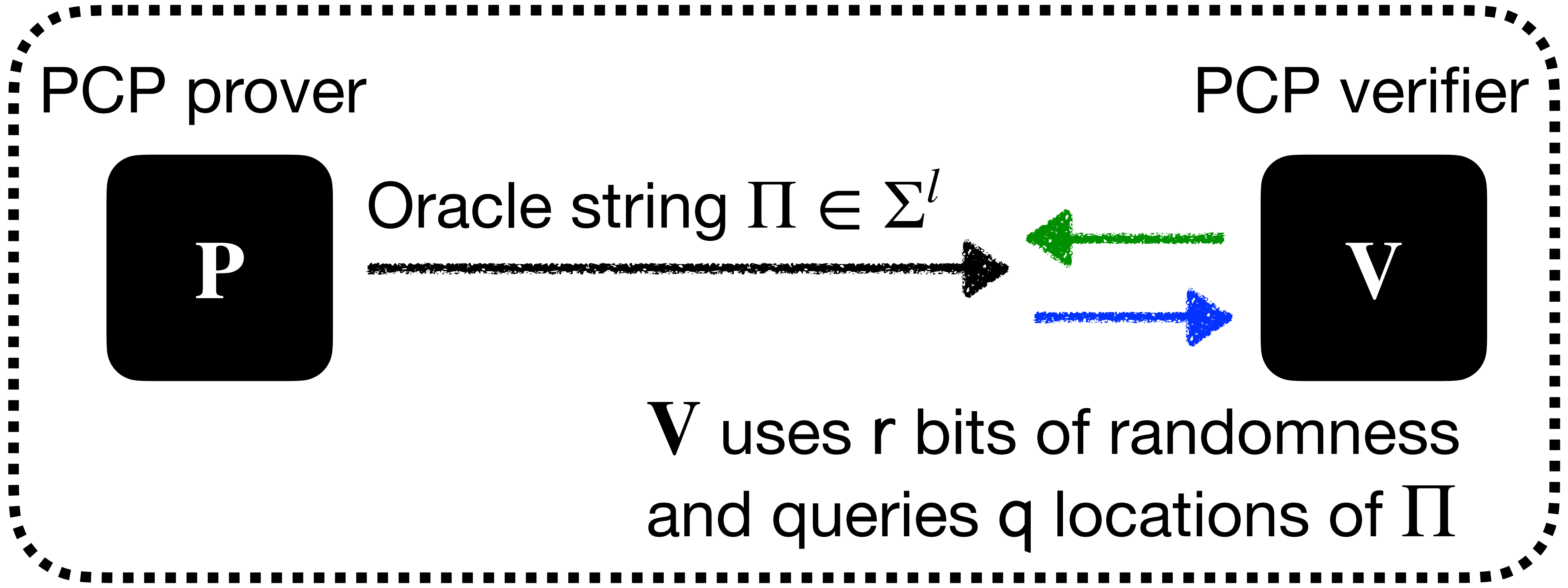
Building block #2: vector commitment scheme (VC)



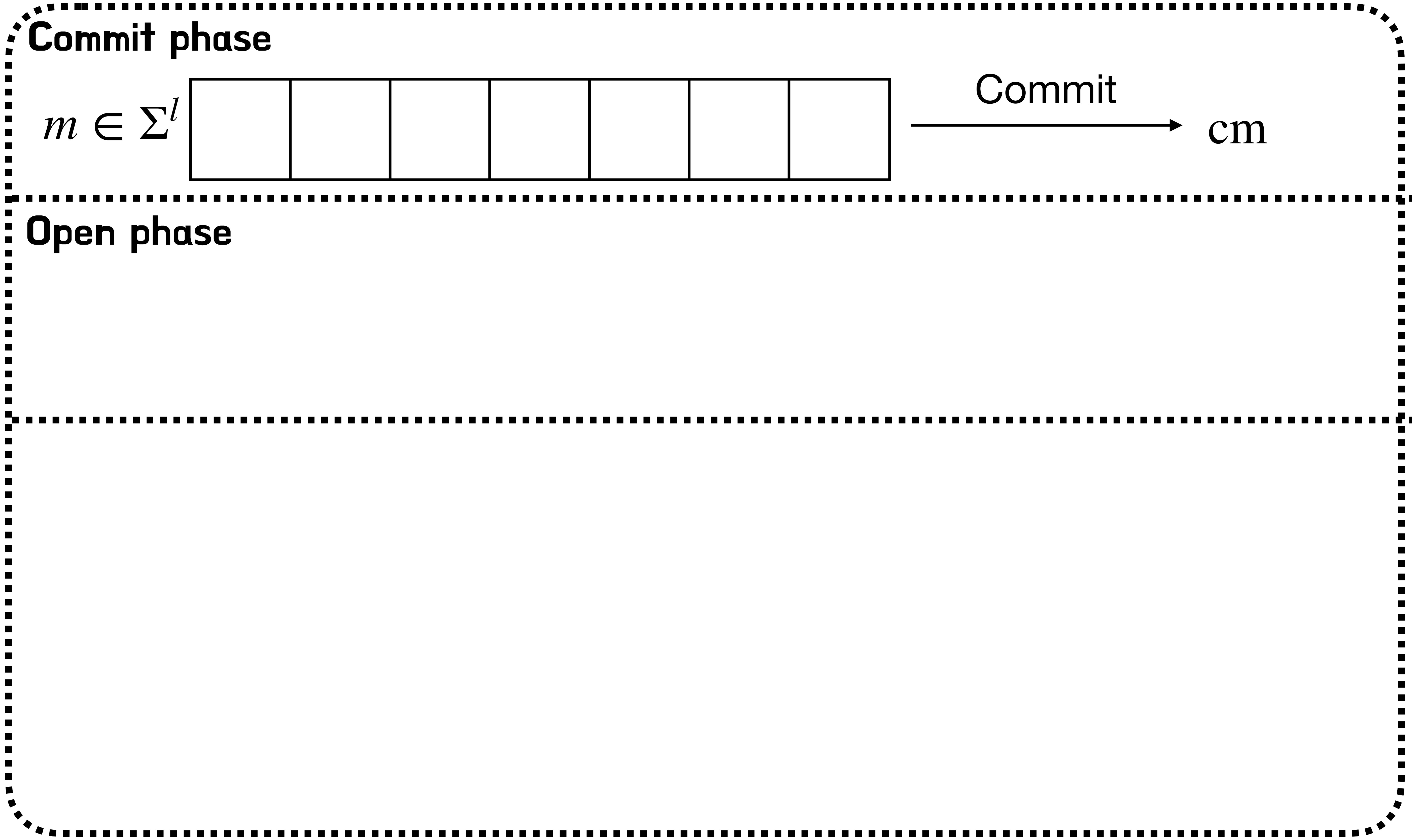
Kilian's protocol

abstraction for a succinct commitment
with local openings (e.g. Merkle tree)

Building block #1: probabilistically checkable proof (PCP)



Building block #2: vector commitment scheme (VC)



Kilian's protocol

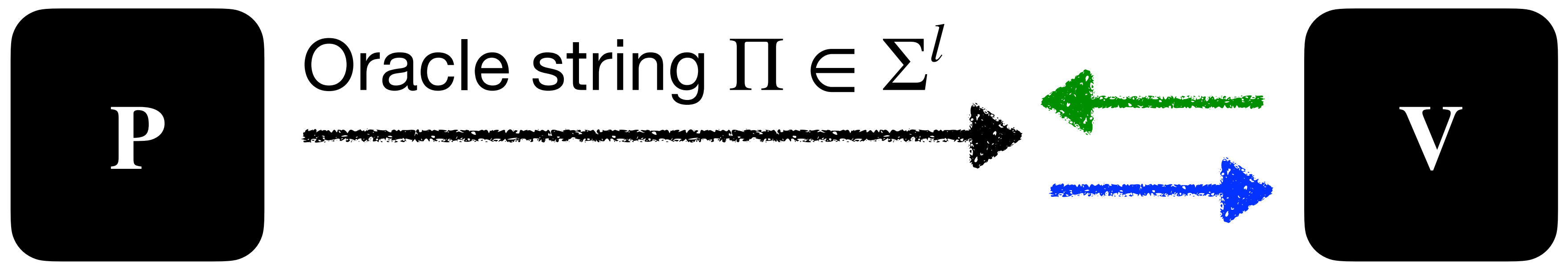
abstraction for a succinct commitment with local openings (e.g. Merkle tree)

Building block #1: probabilistically checkable proof (PCP)

PCP prover

PCP verifier

Oracle string $\Pi \in \Sigma^l$



$\mathbf{V}$ uses r bits of randomness and queries q locations of Π

Building block #2: vector commitment scheme (VC)

Commit phase

$m \in \Sigma^l$ — Commit → cm

Open phase

ans

Query set $Q \subseteq [l]$ — Open from cm → pf

If ans $:= m[Q]$ then Check$(\mathrm{cm}, Q, \text{ans}, \text{pf}) = 1$.

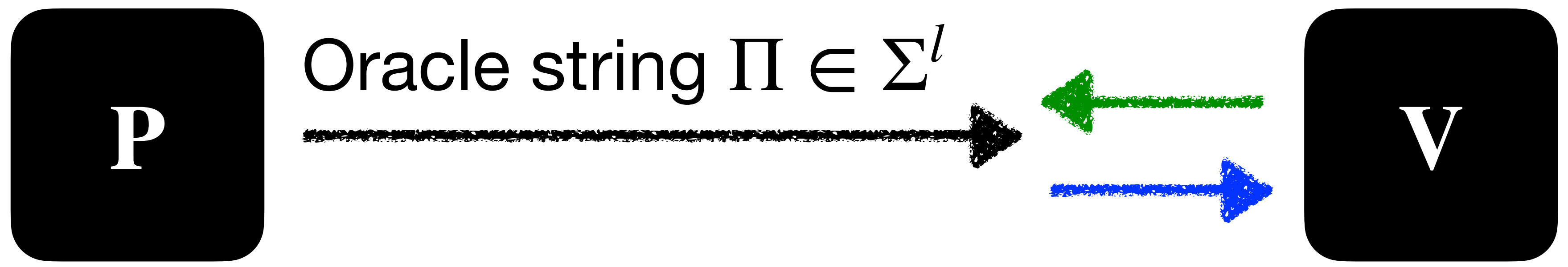
Kilian's protocol

abstraction for a succinct commitment
with local openings (e.g. Merkle tree)

Building block #1: probabilistically checkable proof (PCP)

PCP prover

PCP verifier

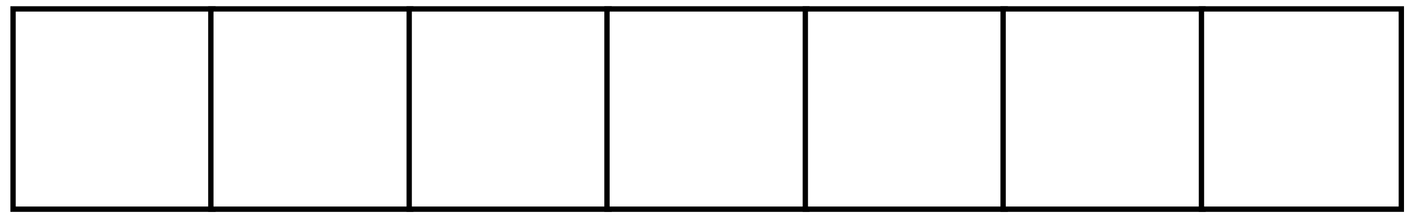


$\mathbf{V}$ uses r bits of randomness
and queries q locations of Π

Building block #2: vector commitment scheme (VC)

Commit phase

$m \in \Sigma^l$

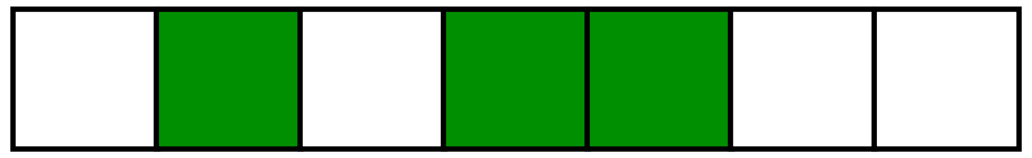


Commit → cm

Open phase

ans

Query set $Q \subseteq [l]$



Open from cm → pf

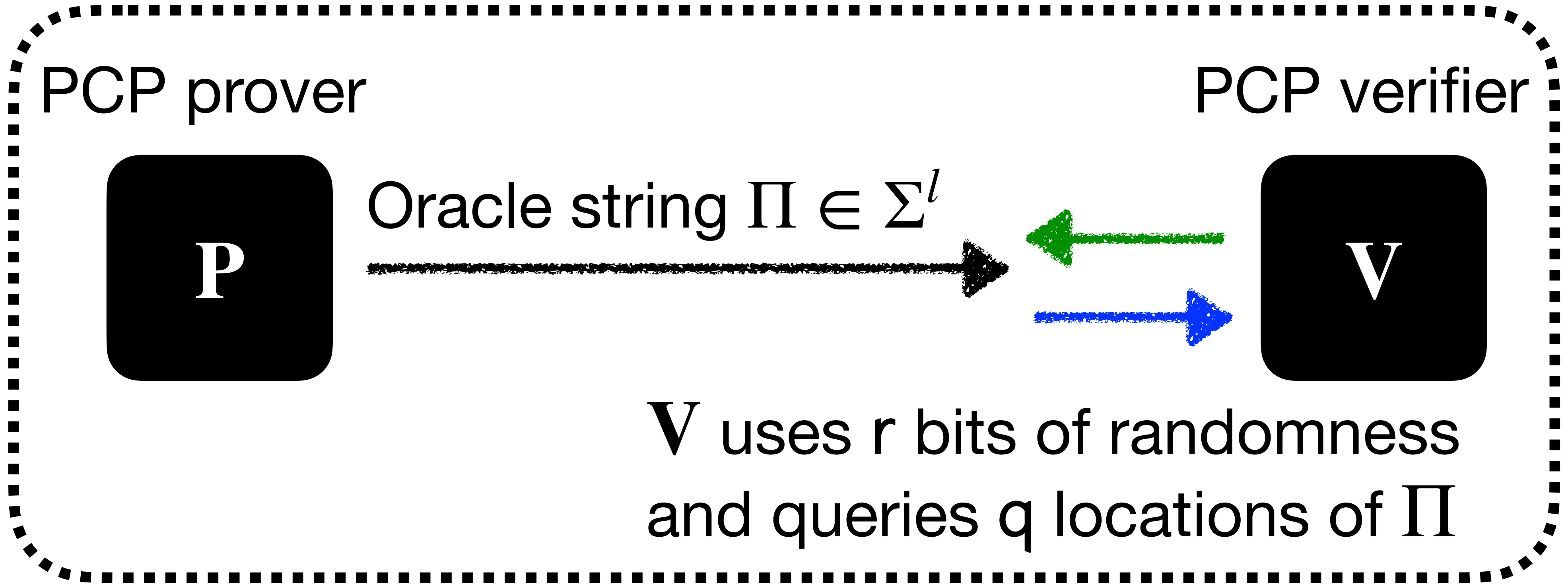
If ans $:= m[Q]$ then Check$(\mathrm{cm}, Q, \text{ans}, \text{pf}) = 1$.

(Expected-time) position binding

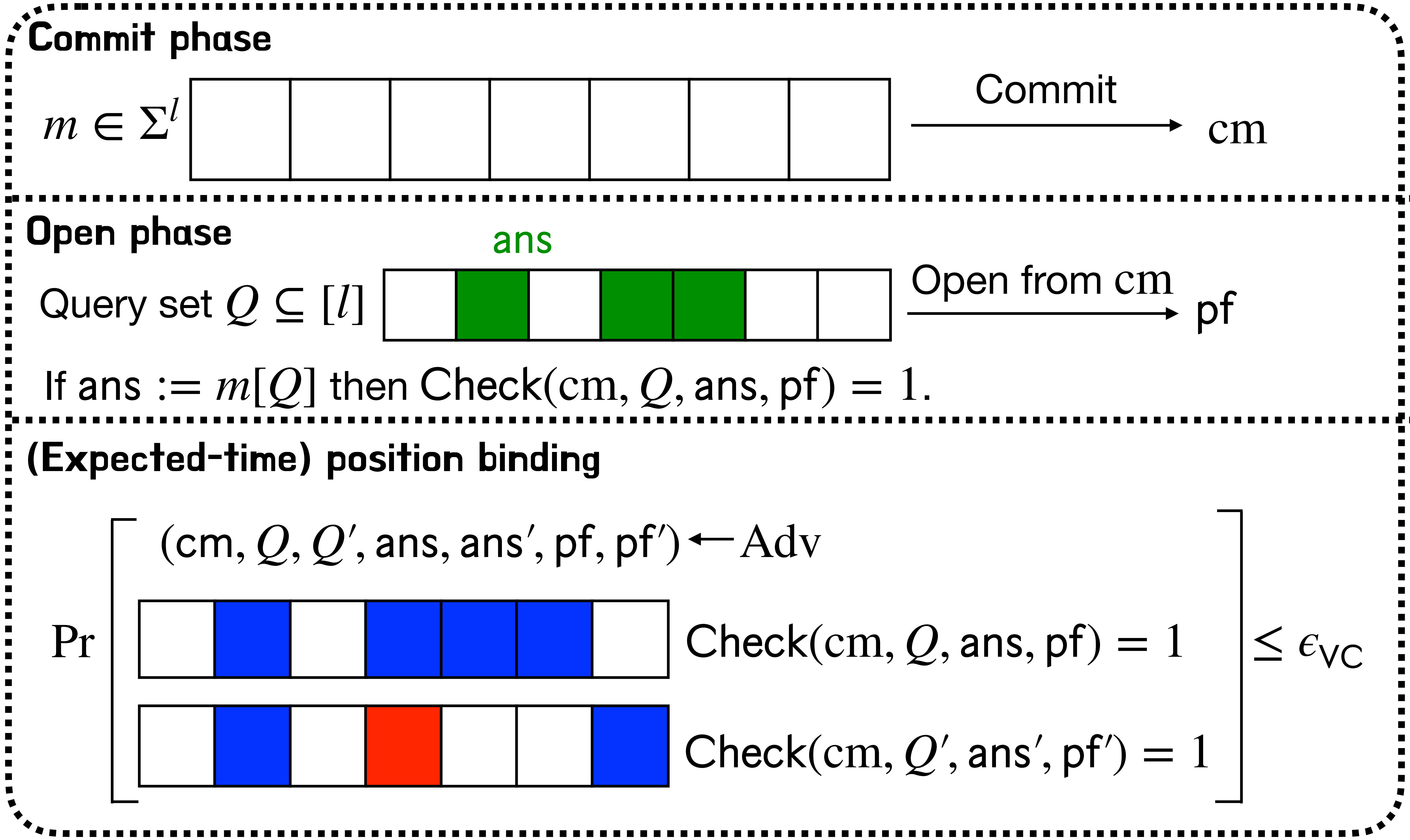
Kilian's protocol

abstraction for a succinct commitment with local openings (e.g. Merkle tree)

Building block #1: probabilistically checkable proof (PCP)



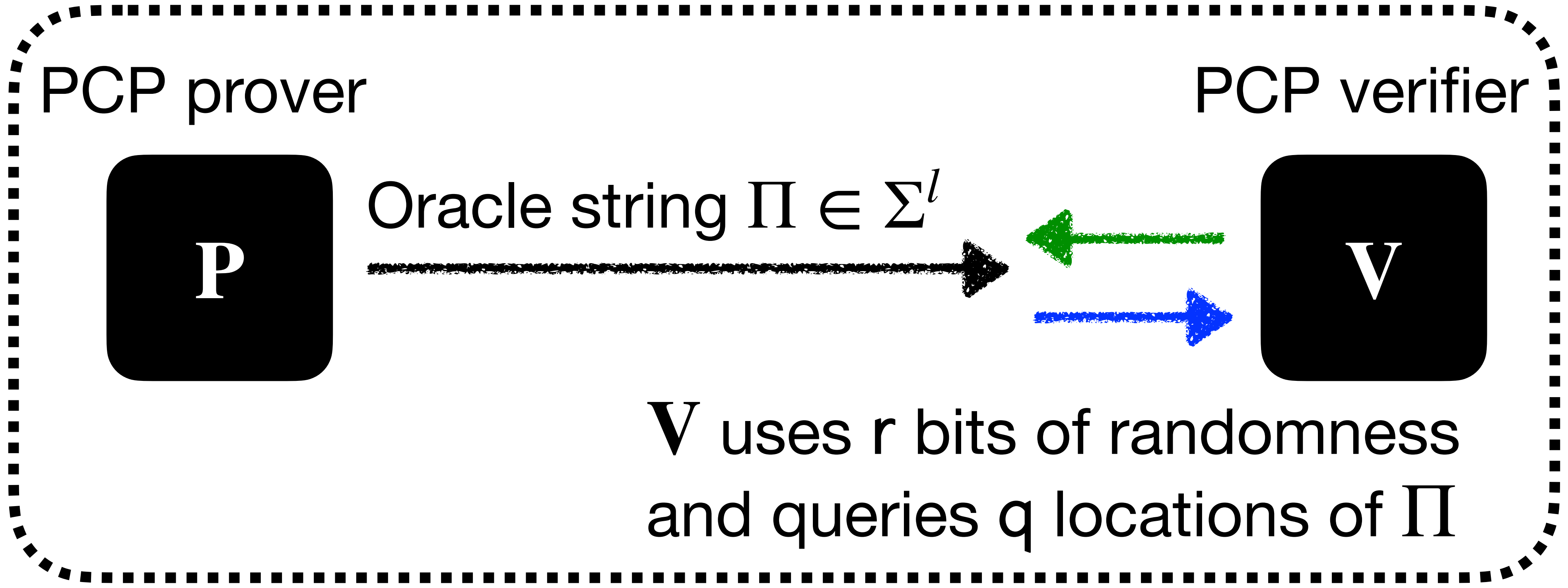
Building block #2: vector commitment scheme (VC)



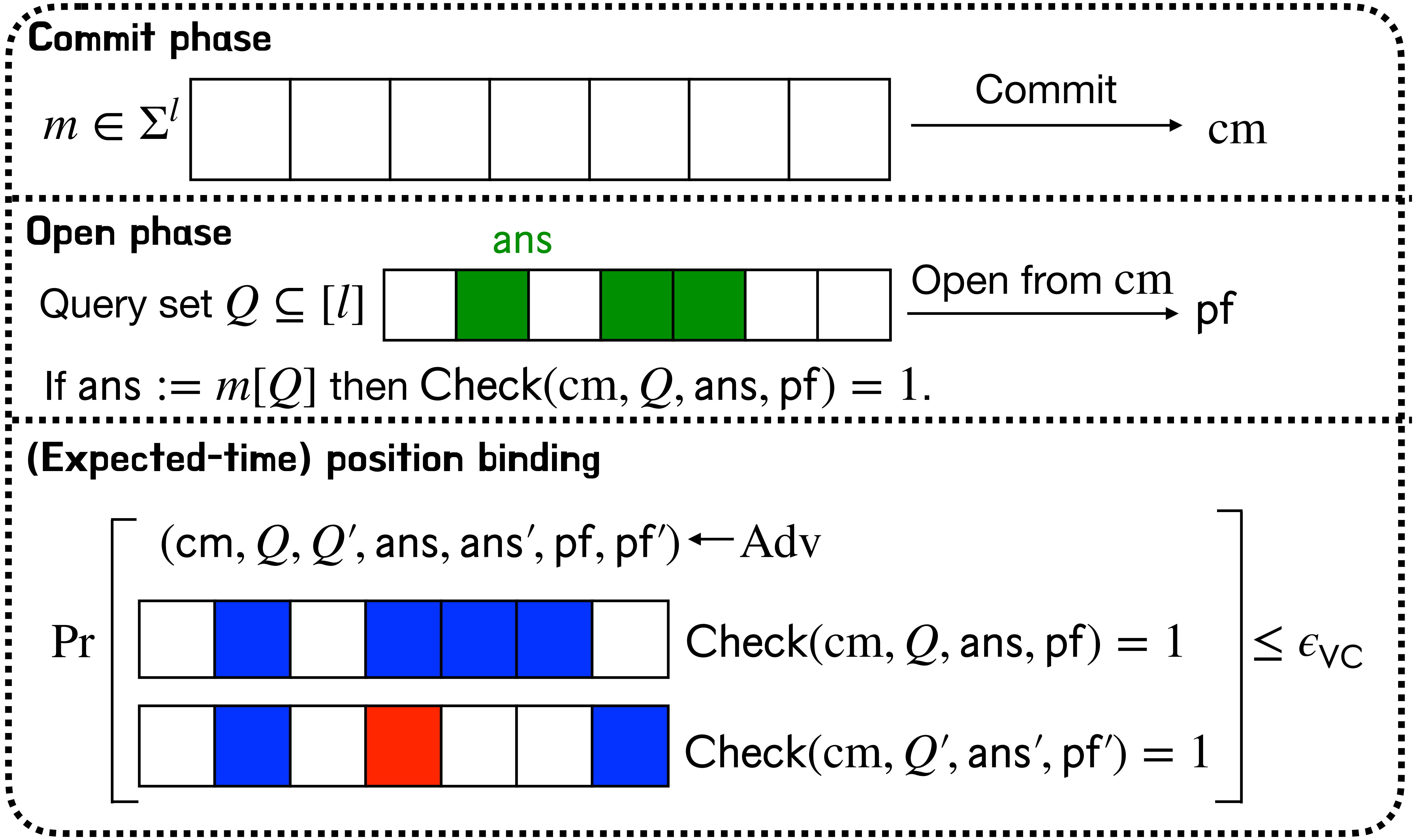
Kilian's protocol

abstraction for a succinct commitment
with local openings (e.g. Merkle tree)

Building block #1: probabilistically checkable proof (PCP)



Building block #2: vector commitment scheme (VC)

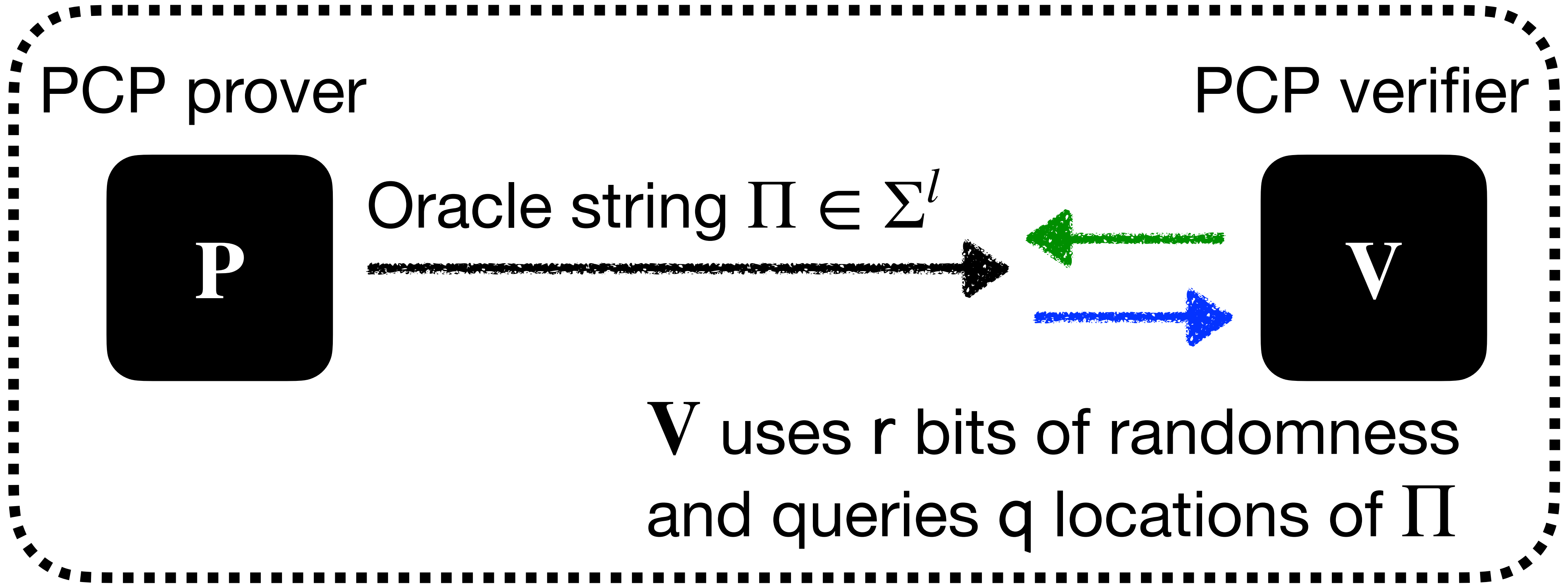


The protocol:

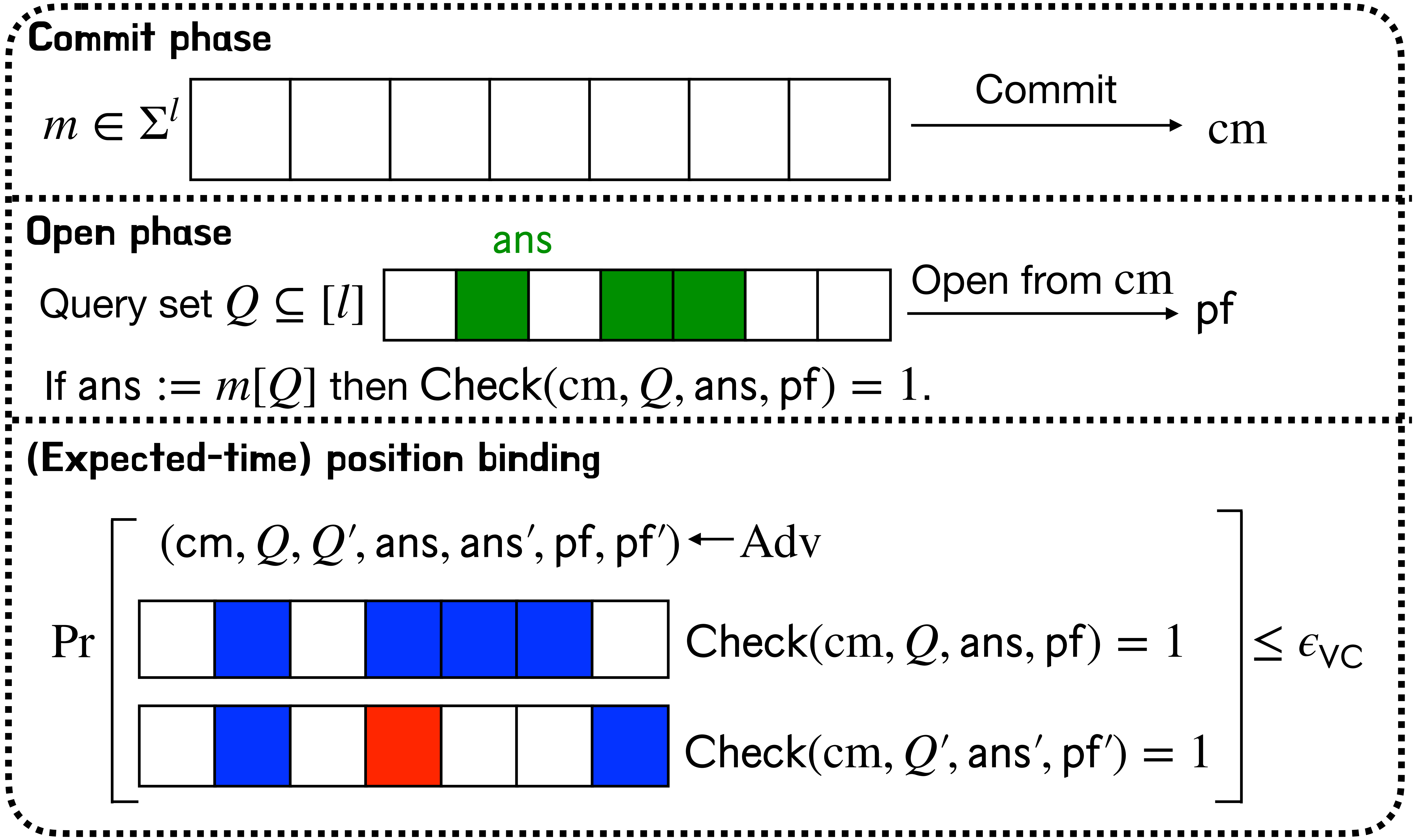
Kilian's protocol

abstraction for a succinct commitment with local openings (e.g. Merkle tree)

Building block #1: probabilistically checkable proof (PCP)



Building block #2: vector commitment scheme (VC)



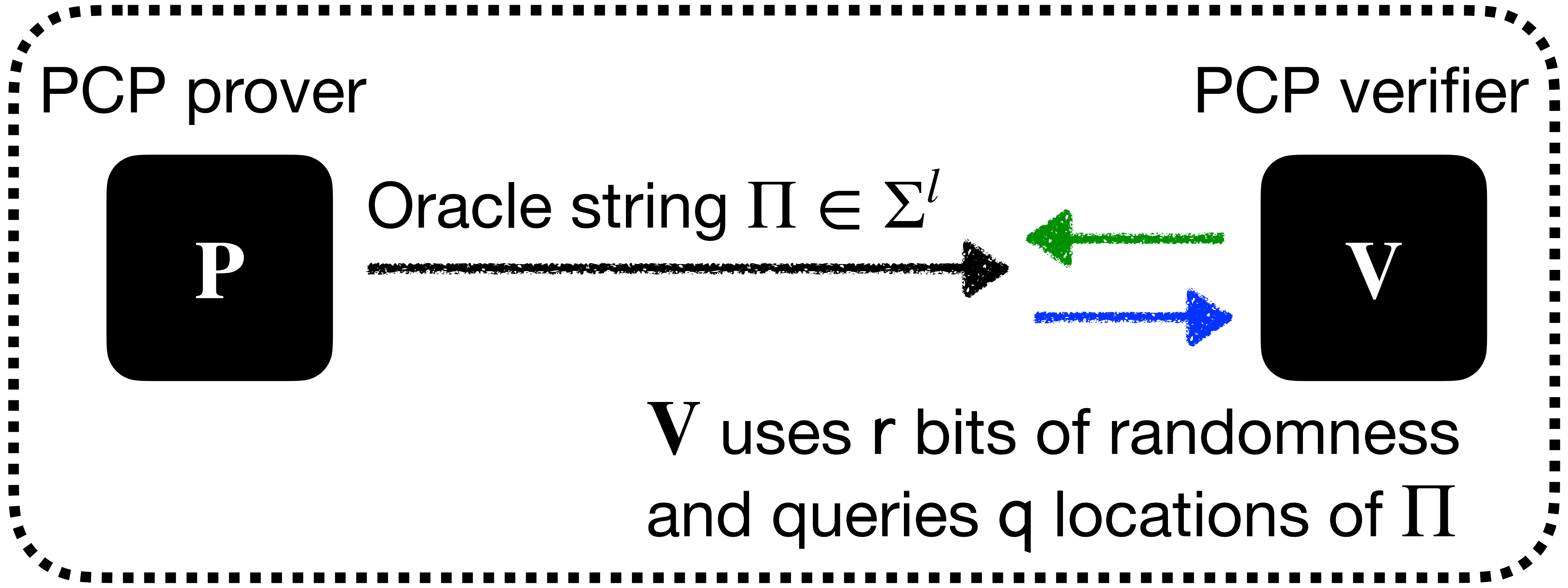
The protocol:

[Kilian92]

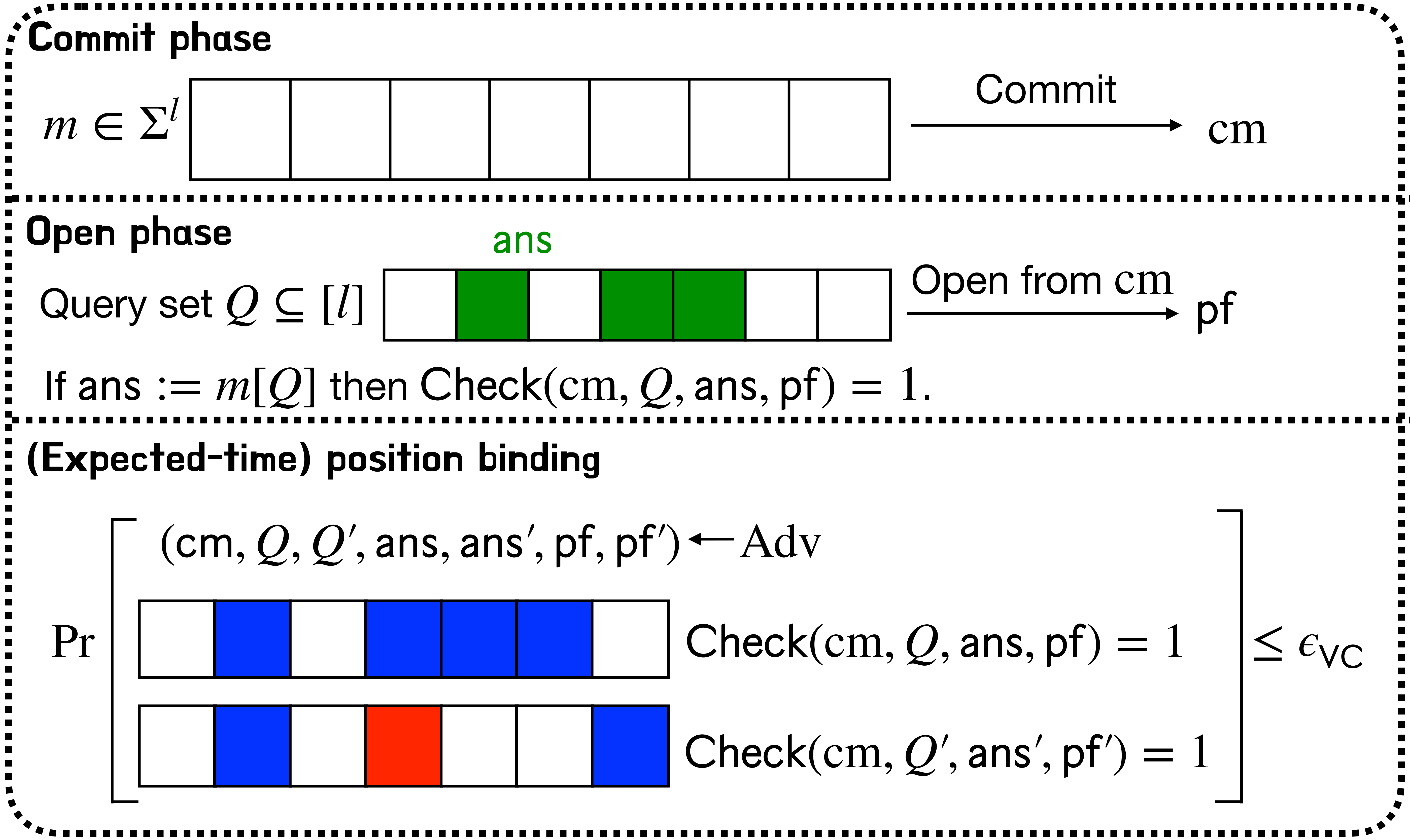
Kilian's protocol

abstraction for a succinct commitment
with local openings (e.g. Merkle tree)

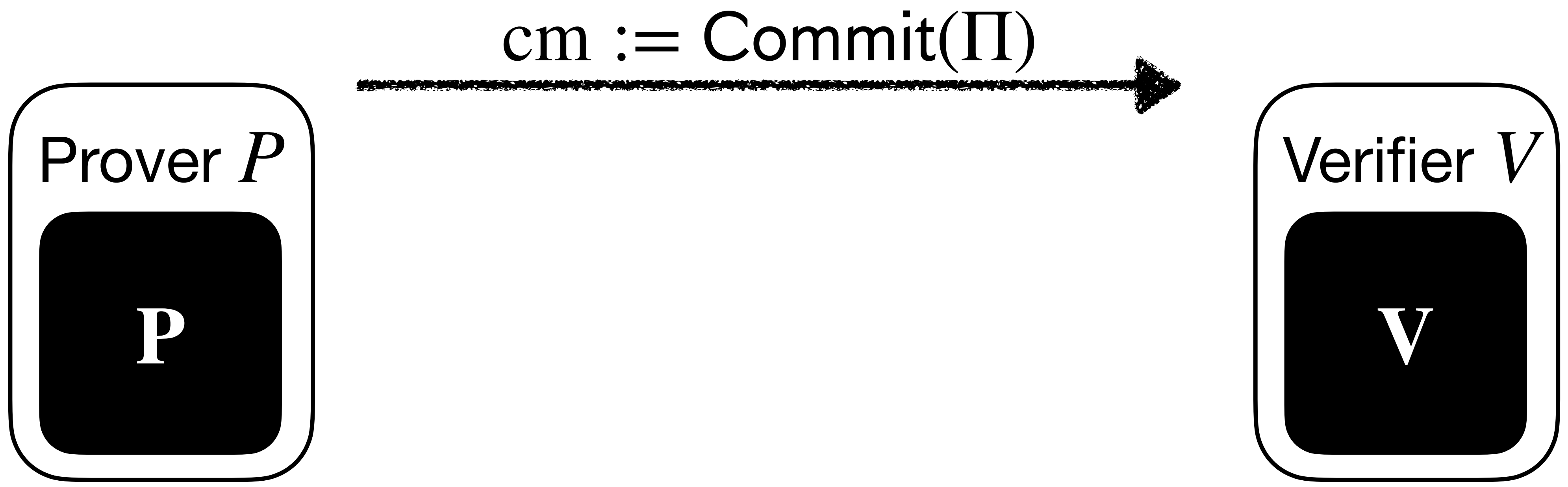
Building block #1: probabilistically checkable proof (PCP)



Building block #2: vector commitment scheme (VC)



The protocol:

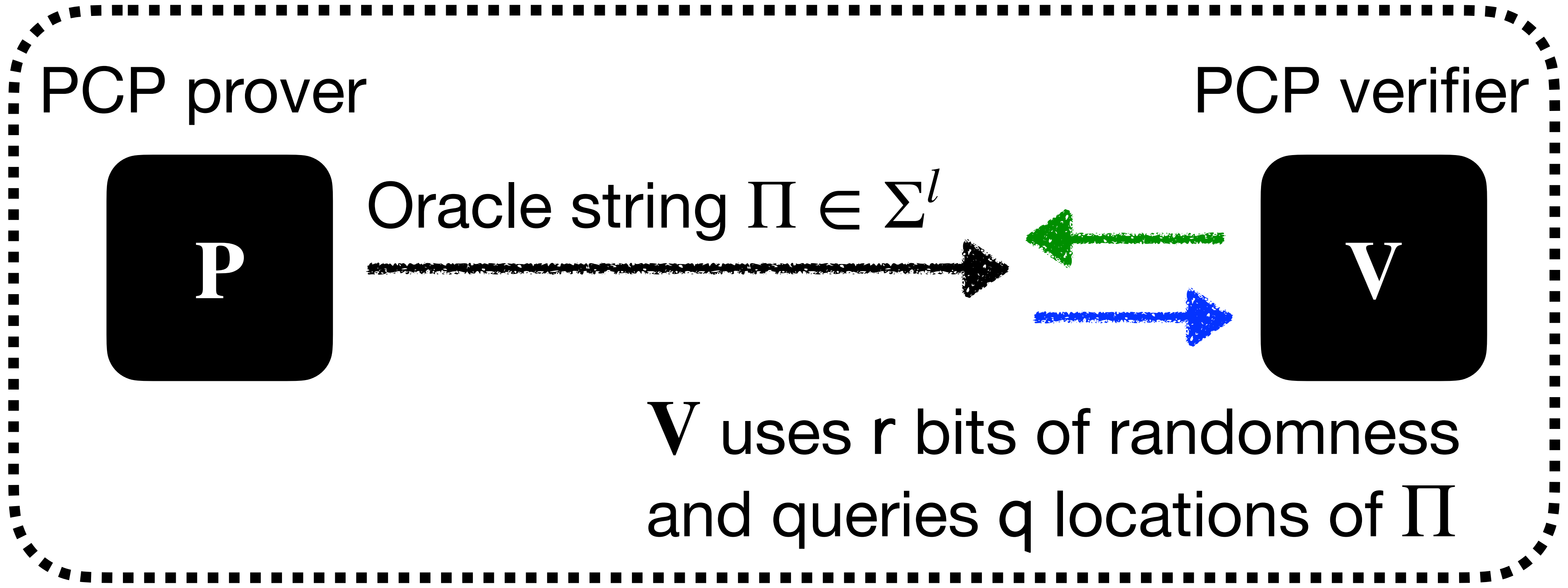


[Kilian92]

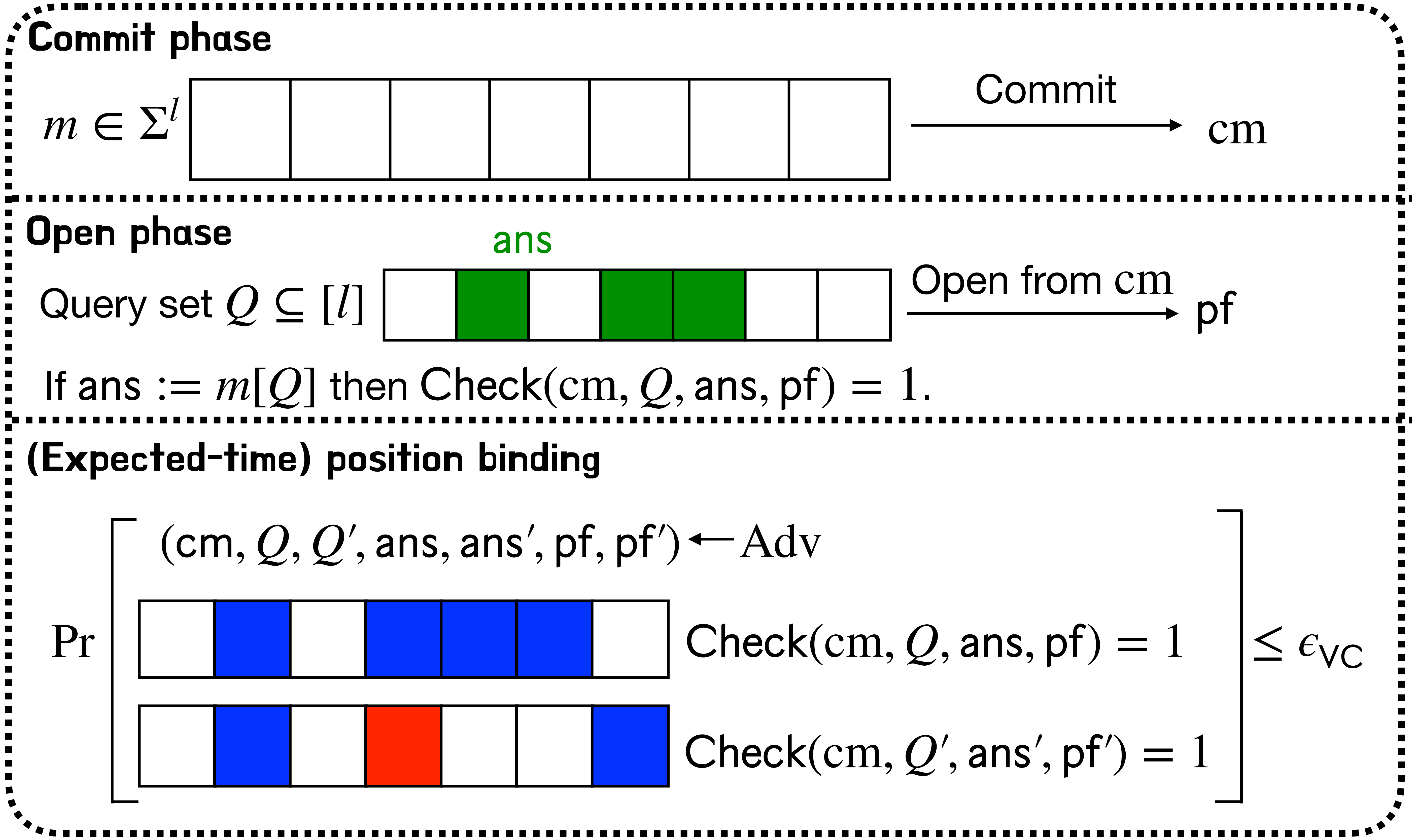
Kilian's protocol

abstraction for a succinct commitment
with local openings (e.g. Merkle tree)

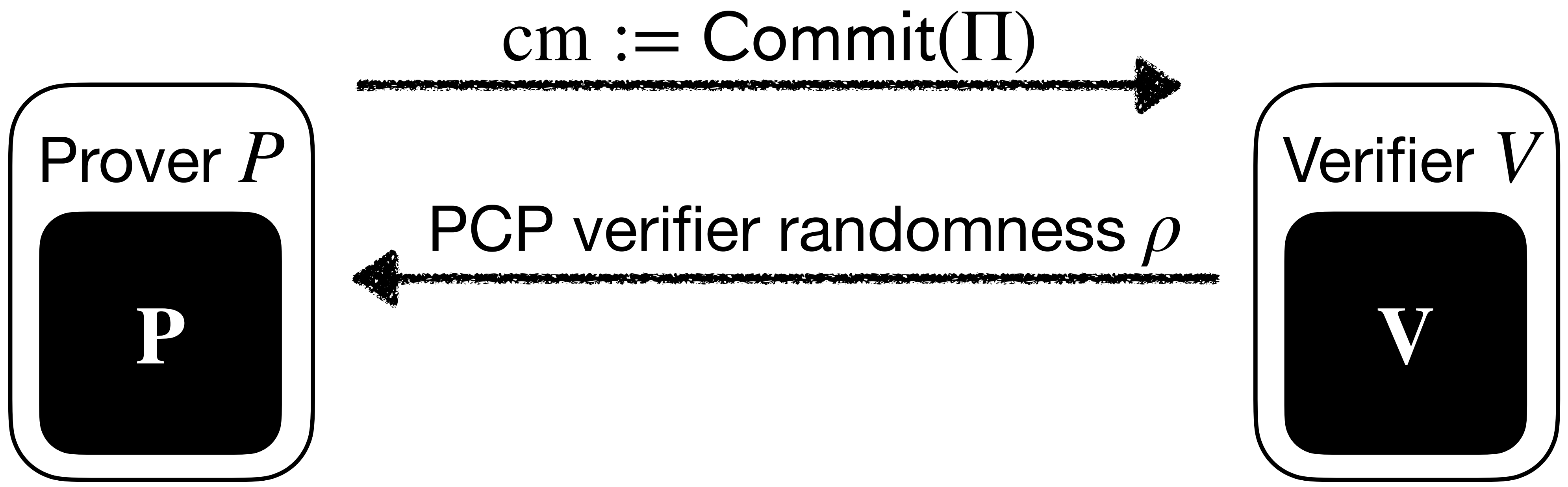
Building block #1: probabilistically checkable proof (PCP)



Building block #2: vector commitment scheme (VC)



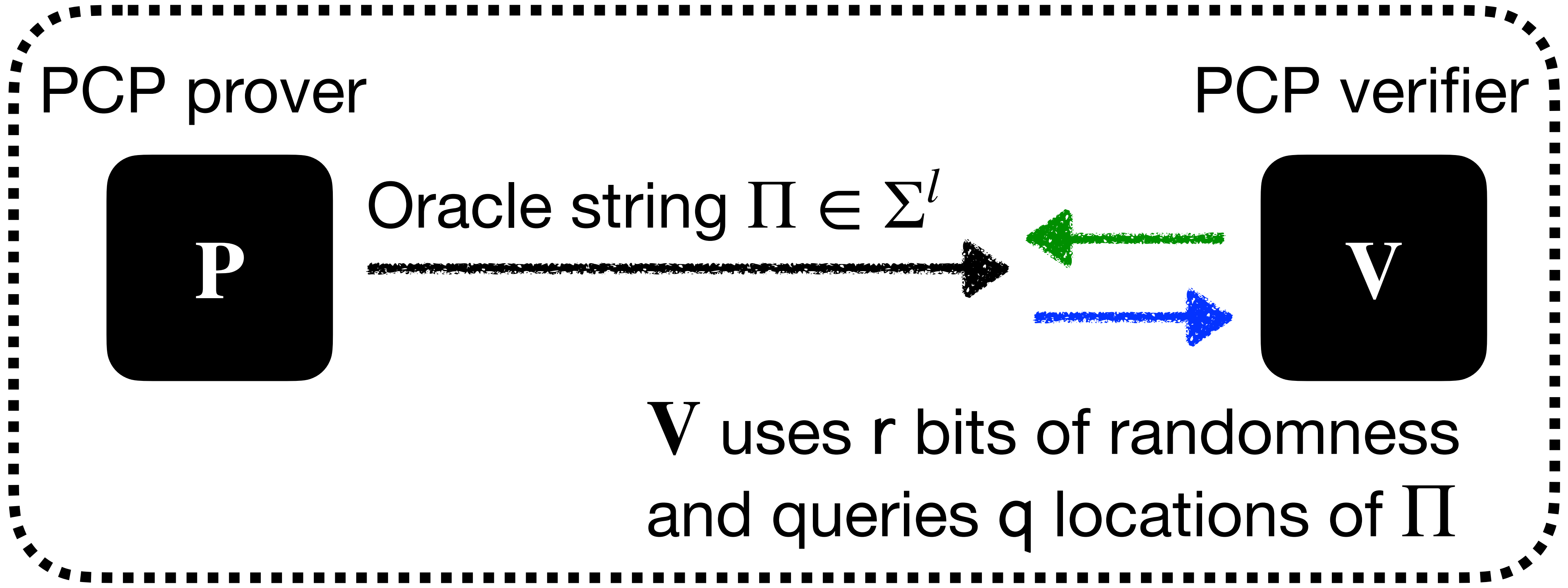
The protocol:



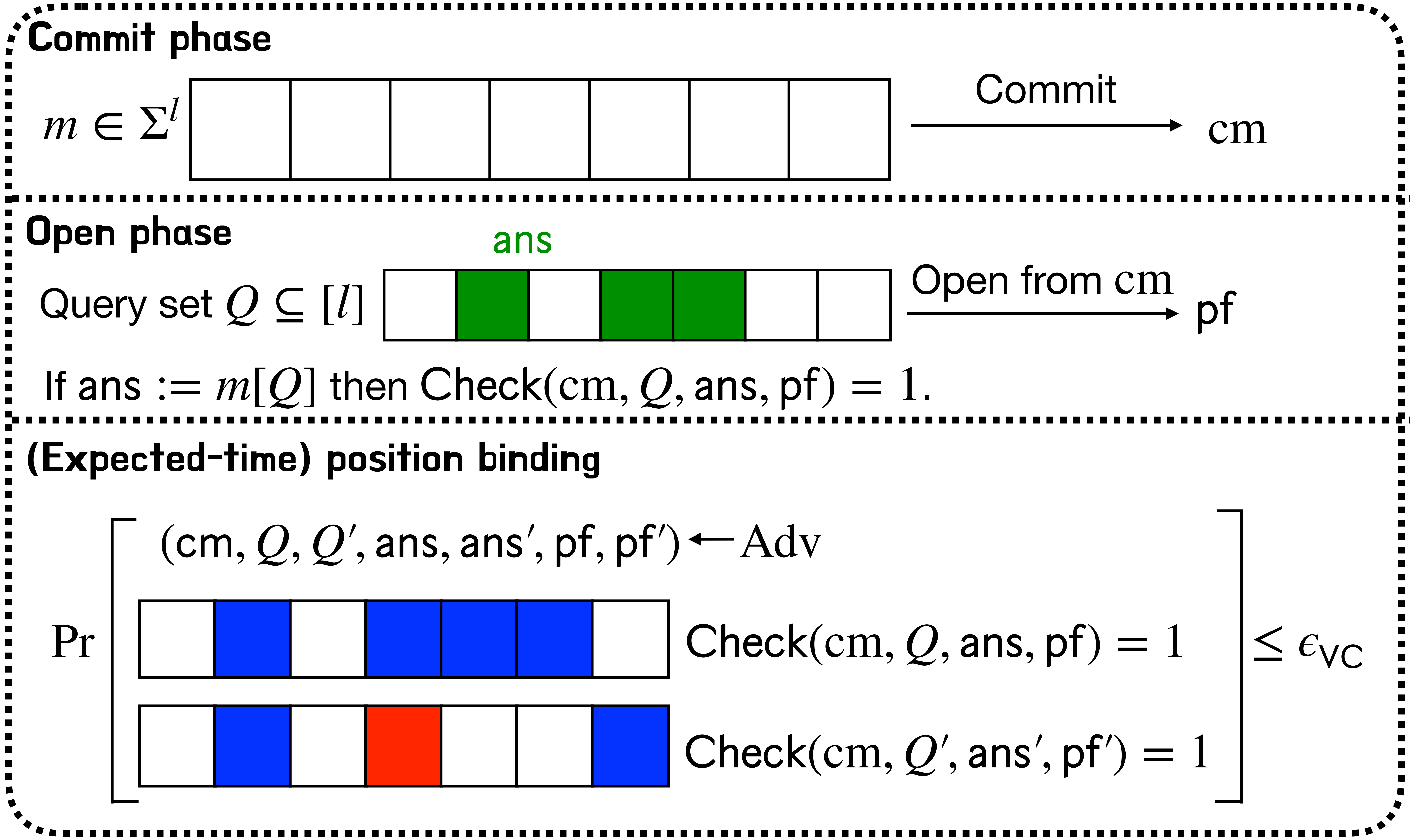
Kilian's protocol

abstraction for a succinct commitment with local openings (e.g. Merkle tree)

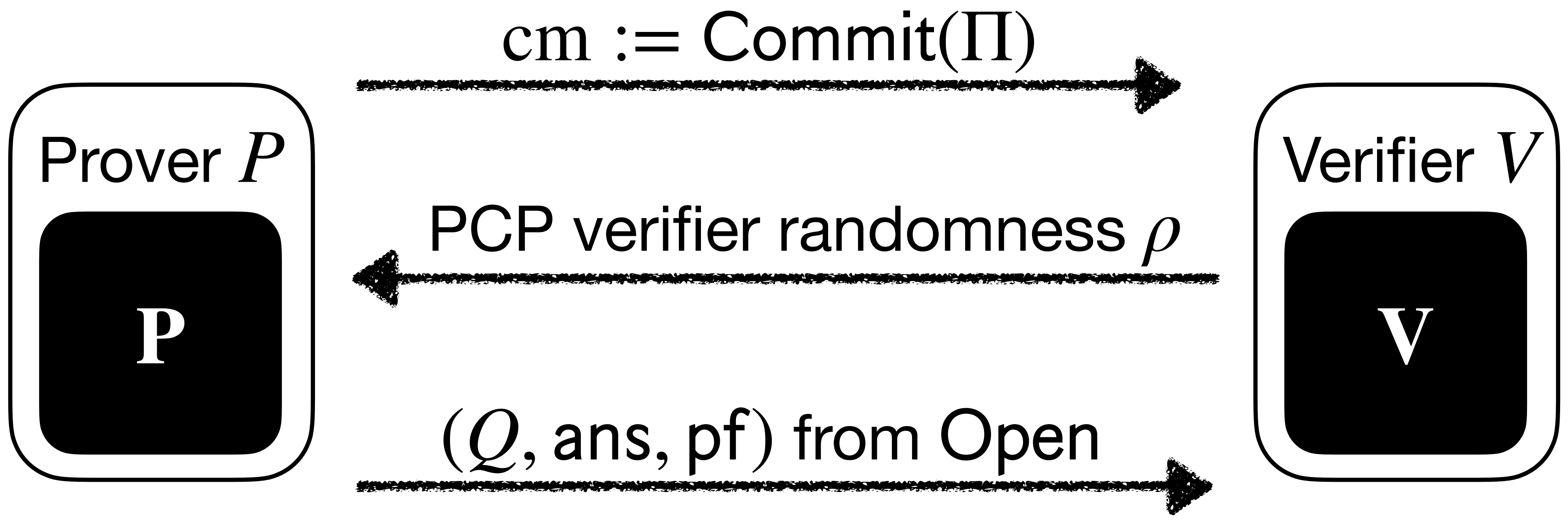
Building block #1: probabilistically checkable proof (PCP)



Building block #2: vector commitment scheme (VC)

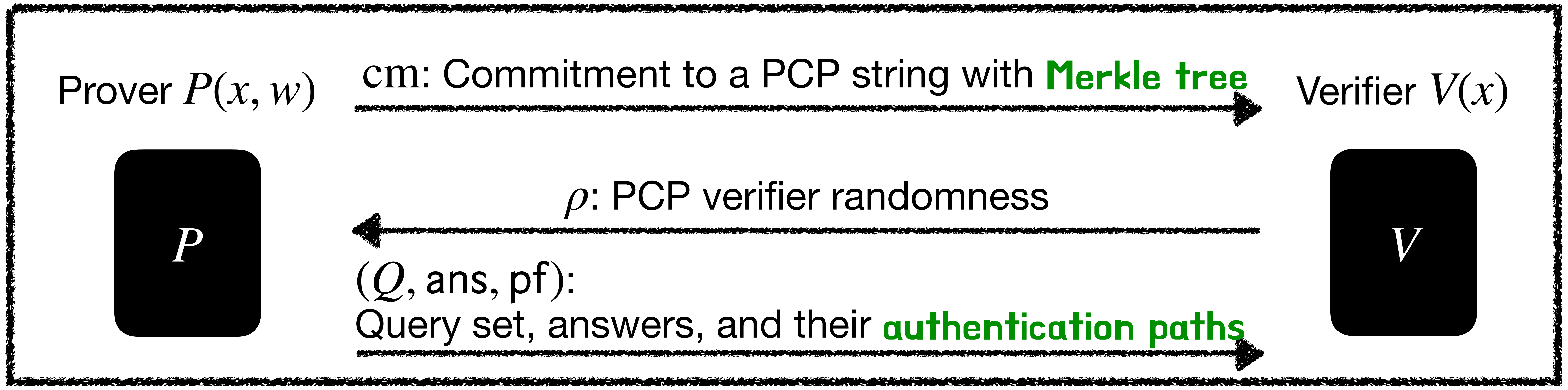


The protocol:

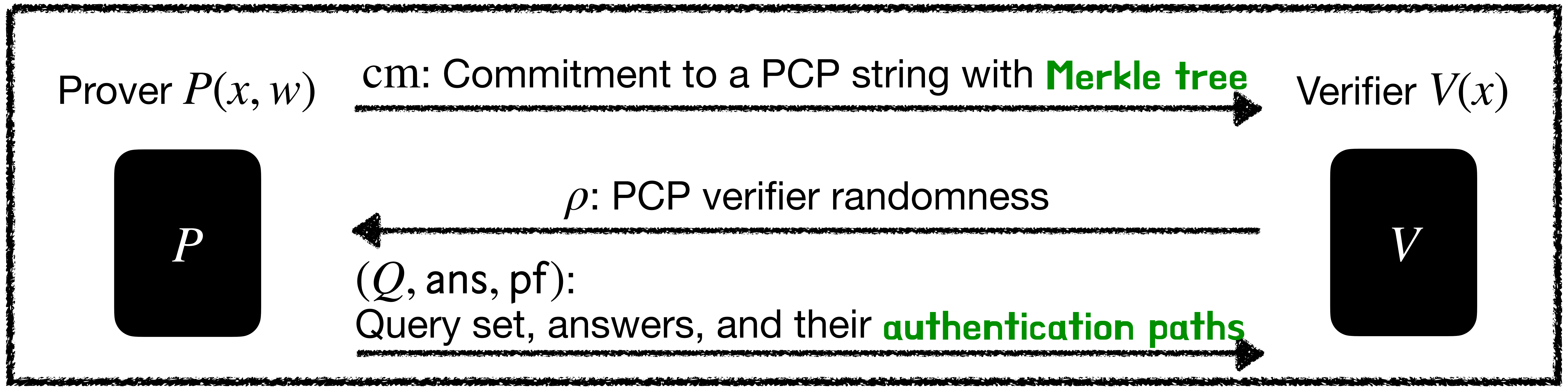


Fundamental question:
What is the security of Kilian's protocol?

What is the security of Kilian's protocol?

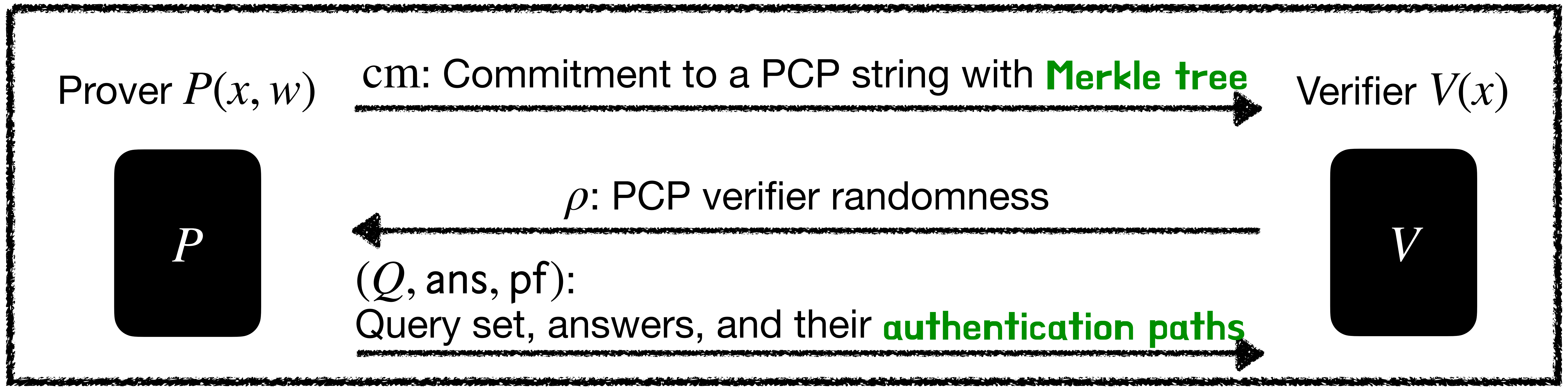


What is the security of Kilian's protocol?



Previously:

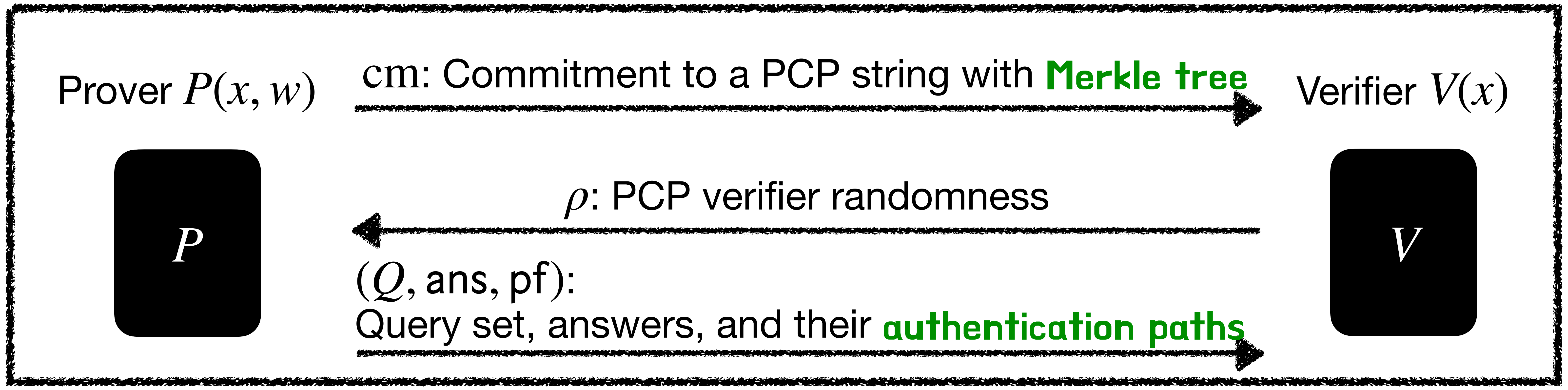
What is the security of Kilian's protocol?



Previously:

- Folklore: well-understood, if ϵ_{PCP} and ϵ_{VC} if negligible, then ϵ_{ARG} is negligible.

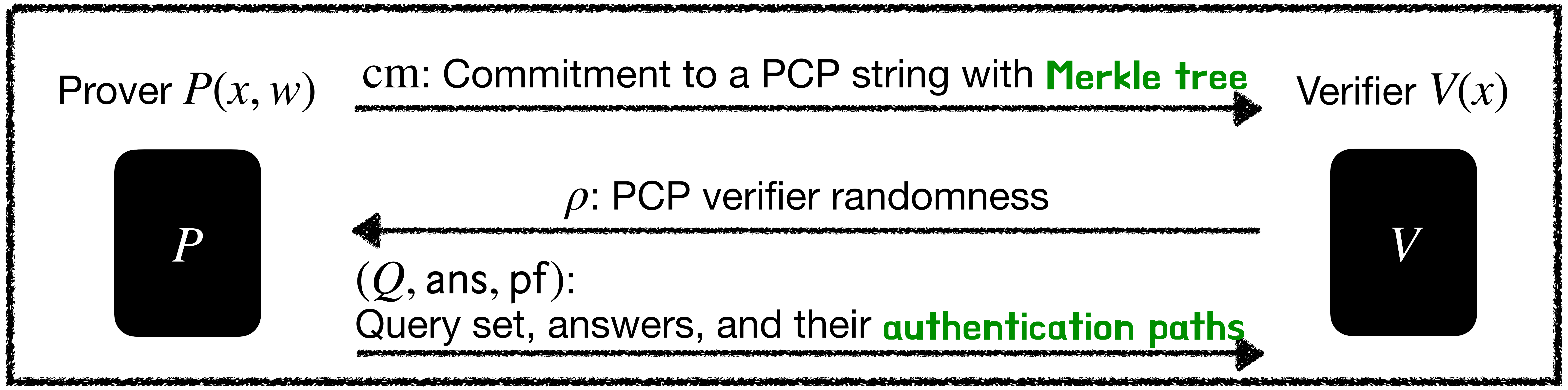
What is the security of Kilian's protocol?



Previously:

- Folklore: well-understood, if ϵ_{PCP} and ϵ_{VC} if negligible, then ϵ_{ARG} is negligible.
- [Kilian92] gives an informal analysis.

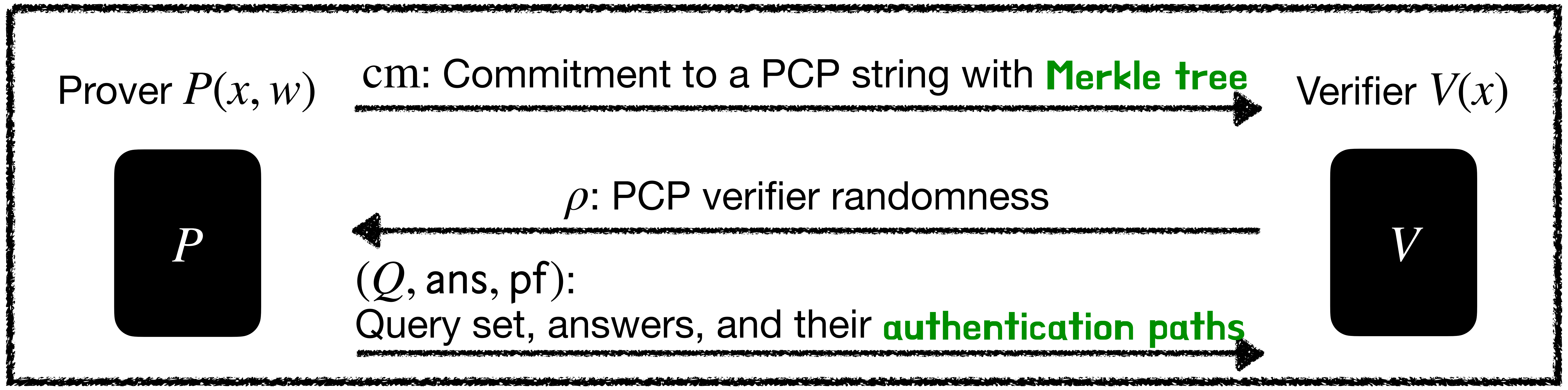
What is the security of Kilian's protocol?



Previously:

- Folklore: well-understood, if ϵ_{PCP} and ϵ_{VC} if negligible, then ϵ_{ARG} is negligible.
- [Kilian92] gives an informal analysis.
- [BG08] proves security of Kilian's protocol **assuming** the underlying PCP is non-adaptive and reverse-samplable.

What is the security of Kilian's protocol?

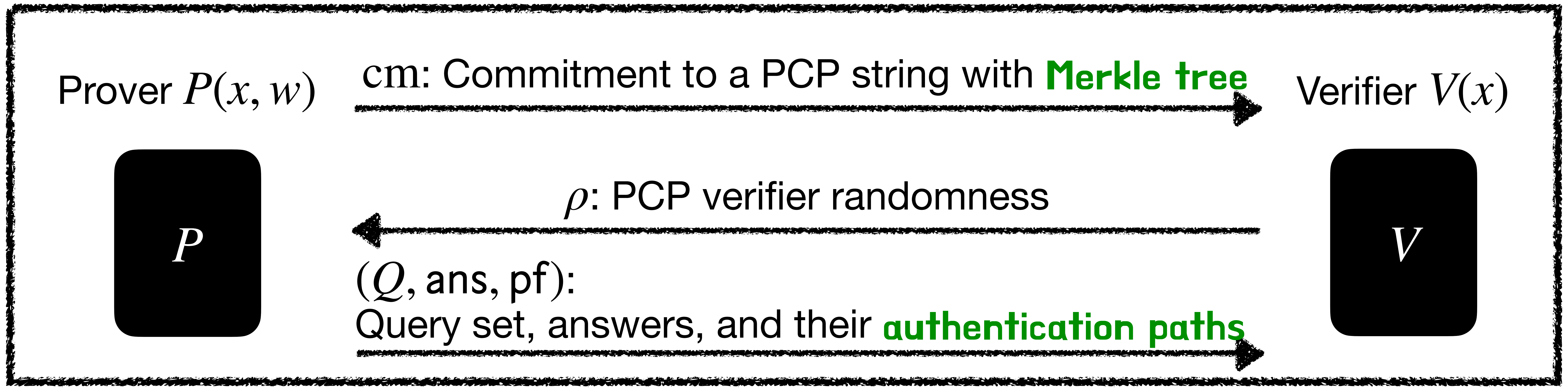


Previously:

- Folklore: well-understood, if ϵ_{PCP} and ϵ_{VC} if negligible, then ϵ_{ARG} is negligible.
- [Kilian92] gives an informal analysis.
- [BG08] proves security of Kilian's protocol **assuming** the underlying PCP is non-adaptive and reverse-samplable.

non-trivial restrictions on the PCP.

What is the security of Kilian's protocol?

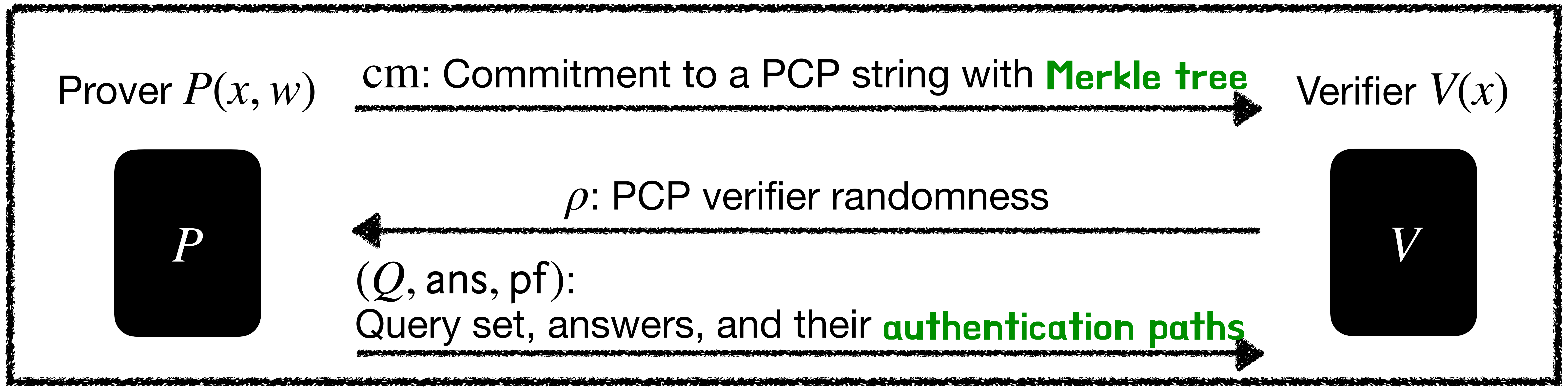


Previously:

- Folklore: well-understood, if ϵ_{PCP} and ϵ_{VC} if negligible, then ϵ_{ARG} is negligible.
- [Kilian92] gives an informal analysis.
- [BG08] proves security of Kilian's protocol **assuming** the underlying PCP is non-adaptive and reverse-samplable.
  Their analysis is NOT tight: roughly $\epsilon_{\text{ARG}} \leq 8 \cdot \epsilon_{\text{PCP}} + \sqrt[3]{\epsilon_{\text{VC}}}$ (multiplicative constant overhead).

non-trivial restrictions on the PCP.

What is the security of Kilian’s protocol?



Previously:

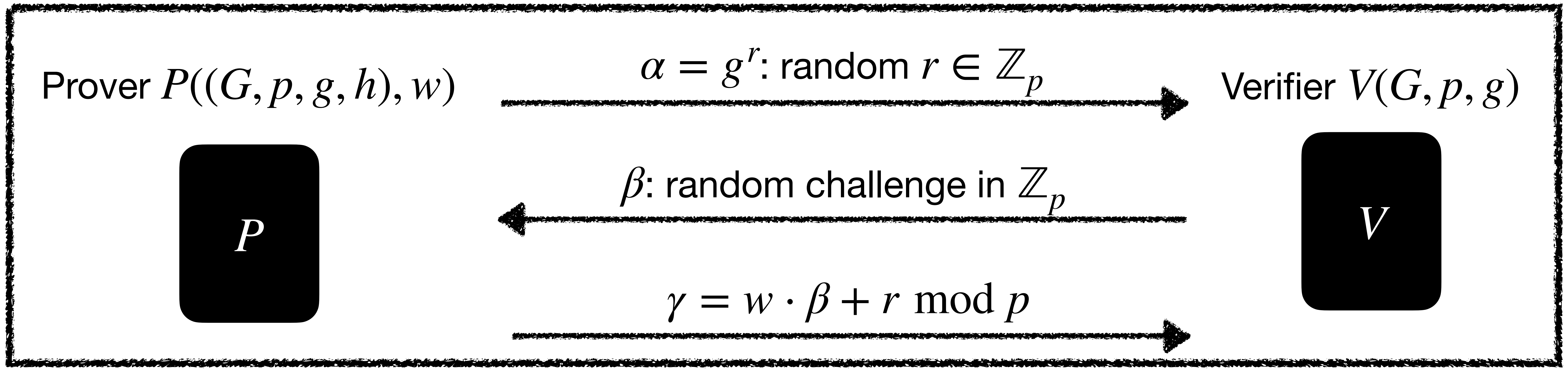
- Folklore: well-understood, if ϵ_{PCP} and ϵ_{VC} if negligible, then ϵ_{ARG} is negligible.
- [Kilian92] gives an informal analysis.

non-trivial restrictions on the PCP.

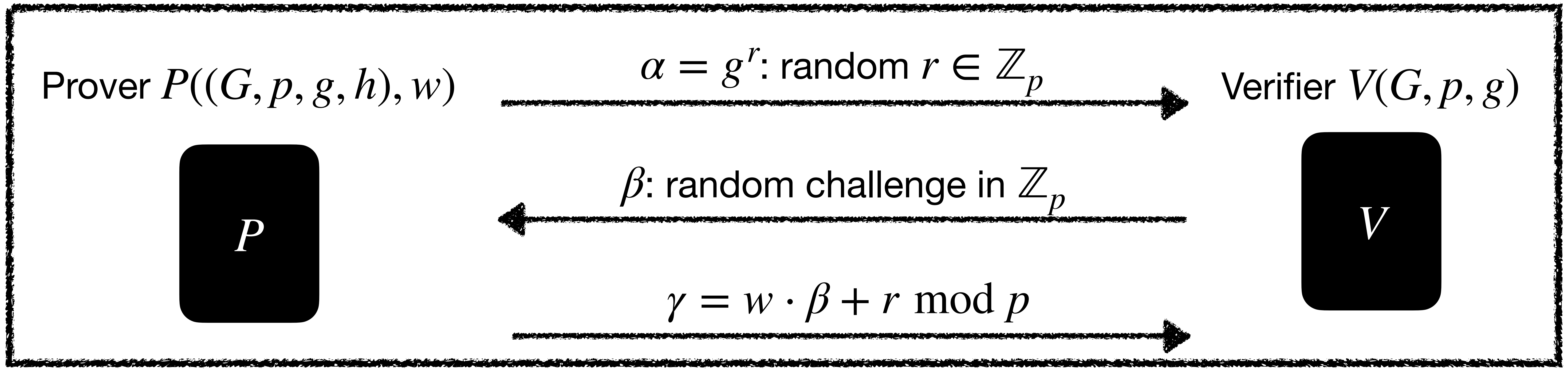
- [BG08] proves security of Kilian’s protocol **assuming** the underlying PCP is non-adaptive and reverse-samplable.
  Their analysis is NOT tight: roughly $\epsilon_{\text{ARG}} \leq 8 \cdot \epsilon_{\text{PCP}} + \sqrt[3]{\epsilon_{\text{VC}}}$ (multiplicative constant overhead).
- Kilian's protocol is widely used across cryptography but lacks a security proof in the general case.

A similar protocol: Schnorr identification scheme

A similar protocol: Schnorr identification scheme

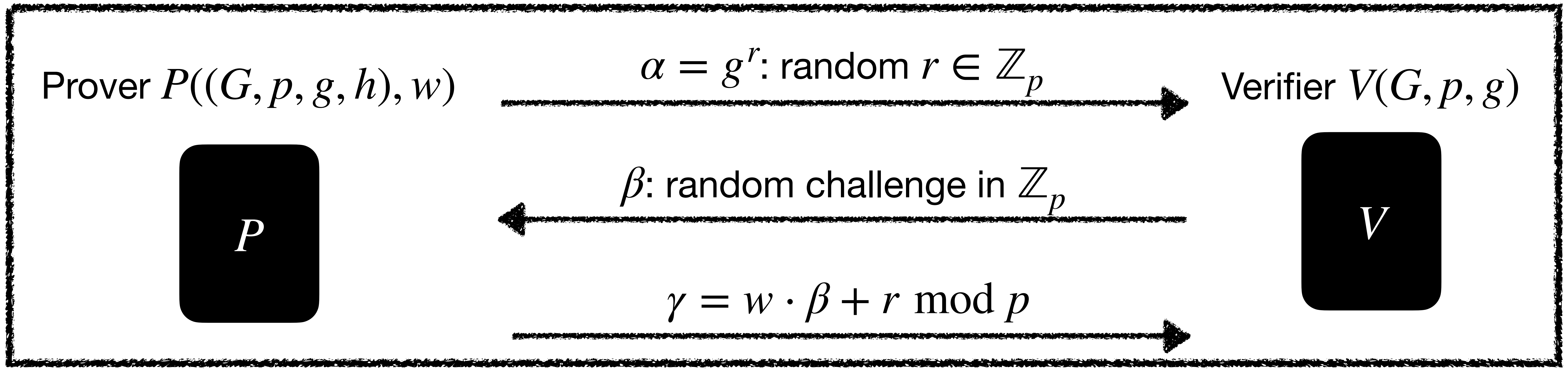


A similar protocol: Schnorr identification scheme



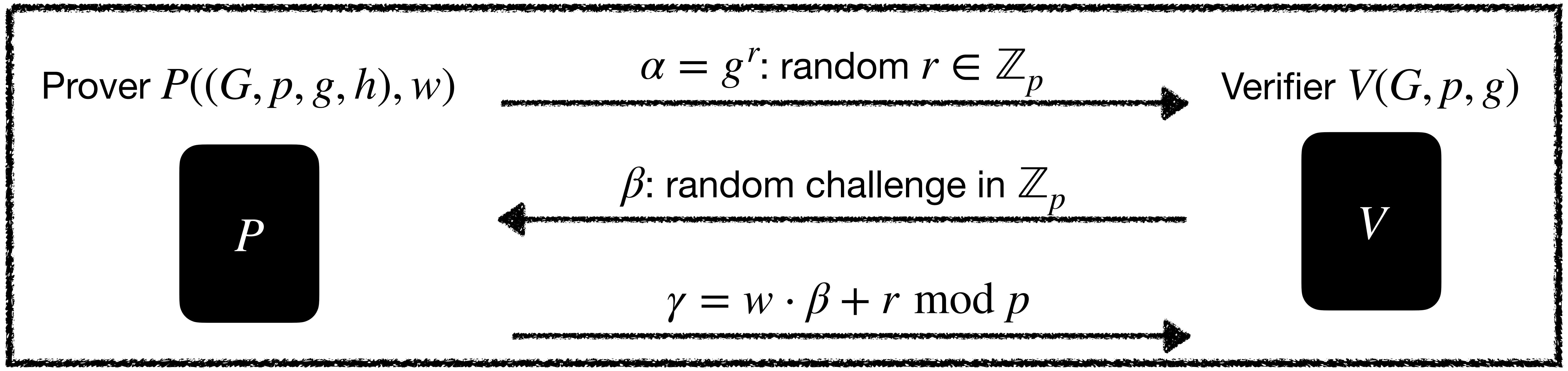
Numerous works study the security of Schnorr identification and its variants in different settings [Sho97,PS00,BP02,FPS20,BD20,RS21,SSY23]

A similar protocol: Schnorr identification scheme



Numerous works study the security of Schnorr identification and its variants in different settings [Sho97,PS00,BP02,FPS20,BD20,RS21,SSY23]
Yet, there are gaps in our understanding of Schnorr's protocol - challenging open questions

A similar protocol: Schnorr identification scheme

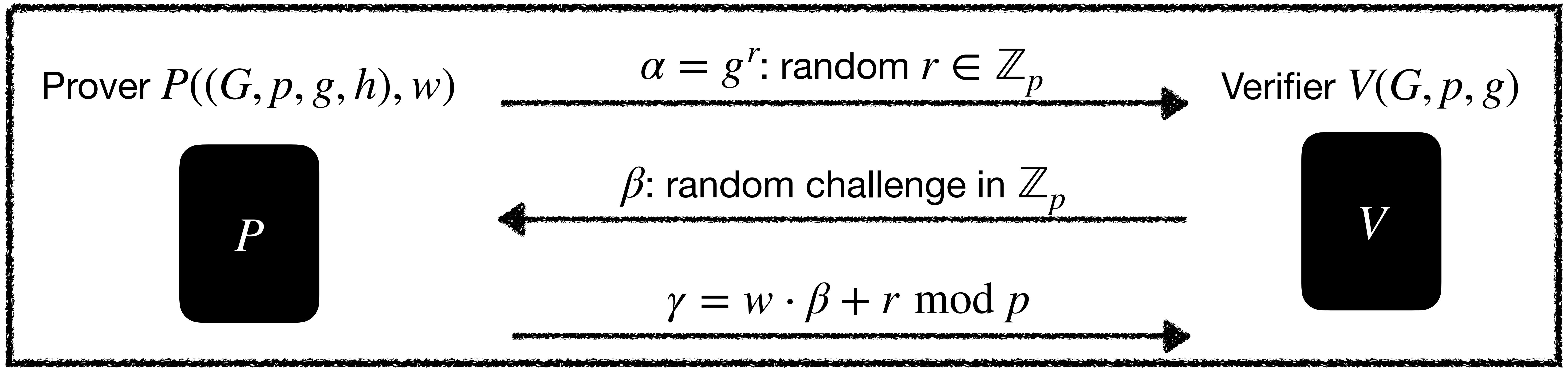


Numerous works study the security of Schnorr identification and its variants in different settings [Sho97,PS00,BP02,FPS20,BD20,RS21,SSY23]

Yet, there are gaps in our understanding of Schnorr's protocol - challenging open questions

Our contribution:

A similar protocol: Schnorr identification scheme



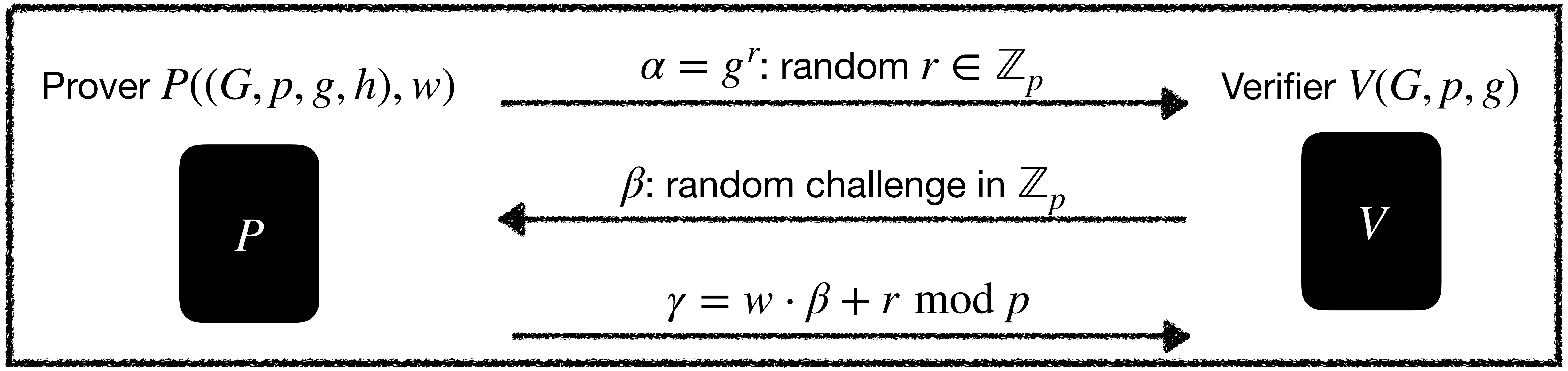
Numerous works study the security of Schnorr identification and its variants in different settings [Sho97,PS00,BP02,FPS20,BD20,RS21,SSY23]

Yet, there are gaps in our understanding of Schnorr's protocol - challenging open questions

Our contribution:

- Proving the security of Kilian's protocol is as hard as that of Schnorr's protocol.

A similar protocol: Schnorr identification scheme



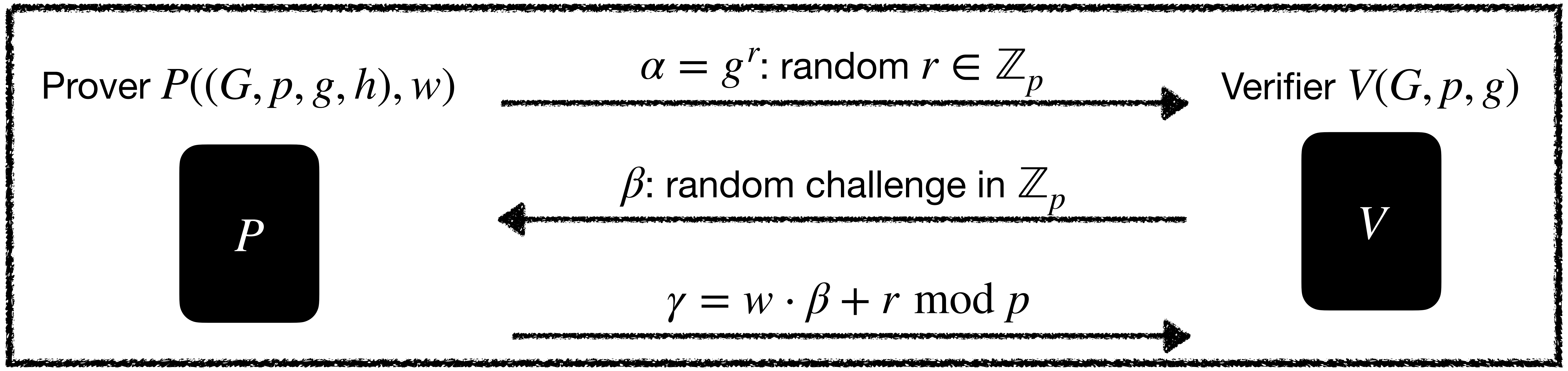
Numerous works study the security of Schnorr identification and its variants in different settings [Sho97,PS00,BP02,FPS20,BD20,RS21,SSY23]

Yet, there are gaps in our understanding of Schnorr's protocol - challenging open questions

Our contribution:

- Proving the security of Kilian's protocol is as hard as that of Schnorr's protocol.
  - Is Kilian's protocol really "well-understood"?

A similar protocol: Schnorr identification scheme

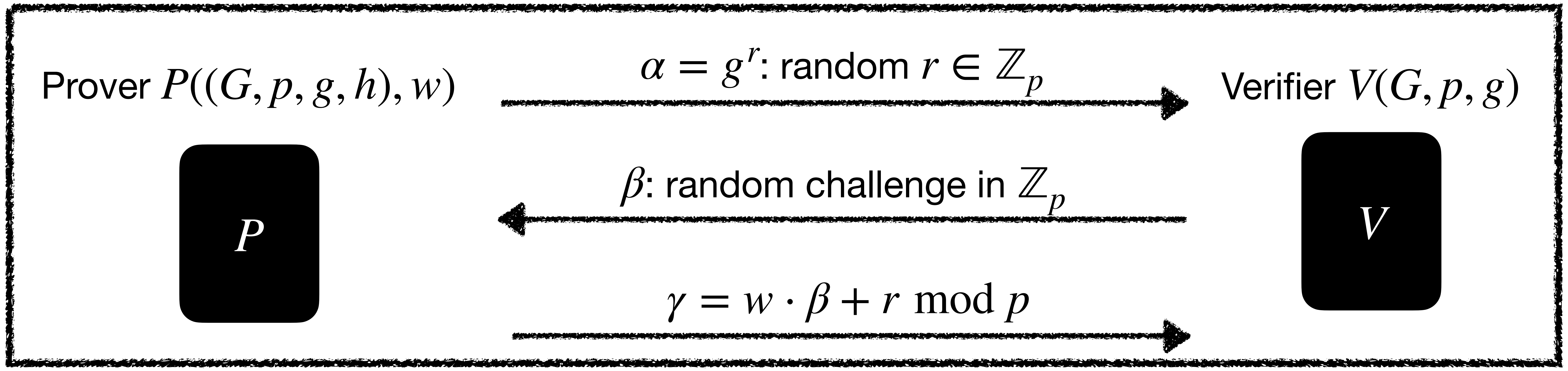


Numerous works study the security of Schnorr identification and its variants in different settings [Sho97,PS00,BP02,FPS20,BD20,RS21,SSY23]
Yet, there are gaps in our understanding of Schnorr's protocol - challenging open questions

Our contribution:

- Proving the security of Kilian's protocol is as hard as that of Schnorr's protocol.
  - Is Kilian's protocol really "well-understood"?
- A general and tightest known security analysis of Kilian's protocol.

A similar protocol: Schnorr identification scheme



Numerous works study the security of Schnorr identification and its variants in different settings [Sho97,PS00,BP02,FPS20,BD20,RS21,SSY23]

Yet, there are gaps in our understanding of Schnorr's protocol - challenging open questions

Our contribution:

- Proving the security of Kilian's protocol is as hard as that of Schnorr's protocol.
  - Is Kilian's protocol really "well-understood"?
- A general and tightest known security analysis of Kilian's protocol.
  - Gaps and barriers remain.

Our results

Our results

Upper Bounds.

Our results

Upper Bounds.

PCP for language L with
- proof length l
- query complexity q
- soundness error ϵ_{PCP}

Our results

Upper Bounds.

PCP for language L with
- proof length l
- query complexity q
- soundness error ϵ_{PCP}

Vector commitment scheme with
position binding error ϵ_{VC}
(or expected-time position binding error $\epsilon^{\star}_{\mathsf{VC}}$)

Our results

Upper Bounds.

PCP for language L with
- proof length l
- query complexity q
- soundness error ϵ_{PCP}

→ **ARG := Kilian[PCP, VC]** ←

Vector commitment scheme with
position binding error ϵ_{VC}
(or expected-time position binding error $\epsilon^{\star}_{\mathsf{VC}}$)

Our results

Upper Bounds.

PCP for language L with
- proof length l
- query complexity q
- soundness error ϵ_{PCP}

→ ARG := Kilian[PCP, VC] ←

Vector commitment scheme with position binding error ϵ_{VC} (or expected-time position binding error $\epsilon^{\star}_{\mathsf{VC}}$)

For every $x \notin L$ and $\epsilon > 0$,

Our results

Upper Bounds.

PCP for language L with
- proof length l
- query complexity q
- soundness error ϵ_{PCP}

→ ARG := Kilian[PCP, VC] ←

Vector commitment scheme with
position binding error ϵ_{VC}
(or expected-time position binding error $\epsilon^{\star}_{\text{VC}}$)

For every $x \notin L$ and $\epsilon > 0$,

$$\epsilon_{\text{ARG}}(\lambda, x, t_{\text{ARG}}) \leq \epsilon_{\text{PCP}}(x) + \epsilon_{\text{VC}}(\lambda, t_{\text{VC}}) + \epsilon, \text{ where } t_{\text{VC}} = O\left(t_{\text{ARG}} \cdot l/\epsilon\right);$$

Our results

Upper Bounds.

PCP for language L with
- proof length l
- query complexity q
- soundness error ϵ_{PCP}

→ ARG := Kilian[PCP, VC] ←

Vector commitment scheme with
position binding error ϵ_{VC}
(or expected-time position binding error $\epsilon^{\star}_{\mathsf{VC}}$)

For every $x \notin L$ and $\epsilon > 0$,

$$\epsilon_{\mathsf{ARG}}(\lambda, x, t_{\mathsf{ARG}}) \leq \epsilon_{\mathsf{PCP}}(x) + \epsilon_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) + \epsilon, \text{ where } t_{\mathsf{VC}} = O\left(t_{\mathsf{ARG}} \cdot l/\epsilon\right);$$

$$\epsilon^{\star}_{\mathsf{ARG}}(\lambda, x, t^{\star}_{\mathsf{ARG}}) \leq \epsilon_{\mathsf{PCP}}(x) + \epsilon^{\star}_{\mathsf{VC}}(\lambda, t^{\star}_{\mathsf{VC}}) + \epsilon, \text{ where } t^{\star}_{\mathsf{VC}} = O\left(t^{\star}_{\mathsf{ARG}} \cdot \log(\mathsf{q}/\epsilon)\right).$$

Our results

Upper Bounds.

PCP for language L with
- proof length l
- query complexity q
- soundness error ϵ_{PCP}

→ **ARG := Kilian[PCP, VC]** ←

Vector commitment scheme with
position binding error ϵ_{VC}
(or expected-time position binding error $\epsilon^{\star}_{\mathsf{VC}}$)

For every $x \notin L$ and $\epsilon > 0$,

$$\epsilon_{\mathsf{ARG}}(\lambda, x, t_{\mathsf{ARG}}) \leq \epsilon_{\mathsf{PCP}}(x) + \epsilon_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) + \epsilon, \text{ where } t_{\mathsf{VC}} = O\left(t_{\mathsf{ARG}} \cdot l/\epsilon\right);$$

$$\epsilon^{\star}_{\mathsf{ARG}}(\lambda, x, t^{\star}_{\mathsf{ARG}}) \leq \epsilon_{\mathsf{PCP}}(x) + \epsilon^{\star}_{\mathsf{VC}}(\lambda, t^{\star}_{\mathsf{VC}}) + \epsilon, \text{ where } t^{\star}_{\mathsf{VC}} = O\left(t^{\star}_{\mathsf{ARG}} \cdot \log(\mathsf{q}/\epsilon)\right).$$

Lower Bounds. Bounding the soundness error of Kilian's protocol is as hard as that of the *Schnorr identification scheme*.

Our results

Upper Bounds.

PCP for language L with
- proof length l
- query complexity q
- soundness error ϵ_{PCP}

→ ARG := Kilian[PCP, VC] ←

Vector commitment scheme with
position binding error ϵ_{VC}
(or expected-time position binding error $\epsilon^{\star}_{\mathsf{VC}}$)

For every $x \notin L$ and $\epsilon > 0$,

$$\epsilon_{\mathsf{ARG}}(\lambda, x, t_{\mathsf{ARG}}) \leq \epsilon_{\mathsf{PCP}}(x) + \epsilon_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) + \epsilon, \text{ where } t_{\mathsf{VC}} = O\left(t_{\mathsf{ARG}} \cdot l/\epsilon\right);$$

$$\epsilon^{\star}_{\mathsf{ARG}}(\lambda, x, t^{\star}_{\mathsf{ARG}}) \leq \epsilon_{\mathsf{PCP}}(x) + \epsilon^{\star}_{\mathsf{VC}}(\lambda, t^{\star}_{\mathsf{VC}}) + \epsilon, \text{ where } t^{\star}_{\mathsf{VC}} = O\left(t^{\star}_{\mathsf{ARG}} \cdot \log(\mathsf{q}/\epsilon)\right).$$

Lower Bounds. Bounding the soundness error of Kilian's protocol is as hard as that of the *Schnorr identification scheme*.

There exists PCP and VC such that, for every $x \notin L$,

Our results

Upper Bounds.

PCP for language L with
- proof length l
- query complexity q
- soundness error ϵ_{PCP}

→ **ARG := Kilian[PCP, VC]** ←

Vector commitment scheme with
position binding error ϵ_{VC}
(or expected-time position binding error $\epsilon^{\star}_{\mathsf{VC}}$)

For every $x \notin L$ and $\epsilon > 0$,

$$\epsilon_{\mathsf{ARG}}(\lambda, x, t_{\mathsf{ARG}}) \leq \epsilon_{\mathsf{PCP}}(x) + \epsilon_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) + \epsilon, \text{ where } t_{\mathsf{VC}} = O\left(t_{\mathsf{ARG}} \cdot l/\epsilon\right);$$

$$\epsilon^{\star}_{\mathsf{ARG}}(\lambda, x, t^{\star}_{\mathsf{ARG}}) \leq \epsilon_{\mathsf{PCP}}(x) + \epsilon^{\star}_{\mathsf{VC}}(\lambda, t^{\star}_{\mathsf{VC}}) + \epsilon, \text{ where } t^{\star}_{\mathsf{VC}} = O\left(t^{\star}_{\mathsf{ARG}} \cdot \log(\mathsf{q}/\epsilon)\right).$$

Lower Bounds. Bounding the soundness error of Kilian's protocol is as hard as that of the *Schnorr identification scheme*.

There exists PCP and VC such that, for every $x \notin L$,

$$\epsilon_{\mathsf{Schnorr}}(\lambda, t_{\mathsf{Schnorr}}) \leq \epsilon_{\mathsf{ARG}}(\lambda, x, t_{\mathsf{ARG}}), \text{ where } t_{\mathsf{ARG}} = O(t_{\mathsf{Schnorr}});$$

Our results

Upper Bounds.

PCP for language L with
- proof length l
- query complexity q
- soundness error ϵ_{PCP}

→ ARG := Kilian[PCP, VC] ←

Vector commitment scheme with
position binding error ϵ_{VC}
(or expected-time position binding error $\epsilon^{\star}_{\mathsf{VC}}$)

For every $x \notin L$ and $\epsilon > 0$,

$$\epsilon_{\mathsf{ARG}}(\lambda, x, t_{\mathsf{ARG}}) \leq \epsilon_{\mathsf{PCP}}(x) + \epsilon_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) + \epsilon, \text{ where } t_{\mathsf{VC}} = O\left(t_{\mathsf{ARG}} \cdot l/\epsilon\right);$$
$$\epsilon^{\star}_{\mathsf{ARG}}(\lambda, x, t^{\star}_{\mathsf{ARG}}) \leq \epsilon_{\mathsf{PCP}}(x) + \epsilon^{\star}_{\mathsf{VC}}(\lambda, t^{\star}_{\mathsf{VC}}) + \epsilon, \text{ where } t^{\star}_{\mathsf{VC}} = O\left(t^{\star}_{\mathsf{ARG}} \cdot \log(\mathsf{q}/\epsilon)\right).$$

Lower Bounds. Bounding the soundness error of Kilian's protocol is as hard as that of the *Schnorr identification scheme*.

There exists PCP and VC such that, for every $x \notin L$,

$$\epsilon_{\mathsf{Schnorr}}(\lambda, t_{\mathsf{Schnorr}}) \leq \epsilon_{\mathsf{ARG}}(\lambda, x, t_{\mathsf{ARG}}), \text{ where } t_{\mathsf{ARG}} = O(t_{\mathsf{Schnorr}});$$
$$\epsilon^{\star}_{\mathsf{Schnorr}}(\lambda, t^{\star}_{\mathsf{Schnorr}}) \leq \epsilon^{\star}_{\mathsf{ARG}}(\lambda, x, t^{\star}_{\mathsf{ARG}}), \text{ where } t^{\star}_{\mathsf{ARG}} = O(t^{\star}_{\mathsf{Schnorr}}).$$

How tight are the bounds?

Strict-time setting.

- Setting $\epsilon_{\mathsf{DLOG}}(\lambda, t) \leq O(t^2/2^\lambda)$.
- Best known analysis of the Schnorr identification scheme:

$$\epsilon_{\mathsf{Schnorr}}(\lambda, t_{\mathsf{Schnorr}}) \leq \sqrt{\epsilon_{\mathsf{DLOG}}(\lambda, O(t_{\mathsf{Schnorr}}))} \leq O\left(\sqrt{t_{\mathsf{Schnorr}}^2/2^\lambda}\right).$$

- Our bound:

$$\epsilon_{\mathsf{ARG}}(\lambda, x, t_{\mathsf{ARG}}) \leq 2^{-\lambda} + \epsilon_{\mathsf{DLOG}}(\lambda, t_{\mathsf{ARG}} \cdot l/\epsilon) + \epsilon \leq 2^{-\lambda} + l^{2/3} \cdot O\left(\sqrt[3]{t_{\mathsf{ARG}}^2/2^\lambda}\right).$$

Polynomial gap

Expected-time setting.

- Best known analysis of the Schnorr identification scheme:

$$\epsilon^\star_{\mathsf{Schnorr}}(\lambda, t^\star_{\mathsf{Schnorr}}) \leq \epsilon^\star_{\mathsf{DLOG}}(\lambda, O(t^\star_{\mathsf{Schnorr}})).$$

- Our bound:

$$\epsilon^\star_{\mathsf{ARG}}(\lambda, x, t_{\mathsf{ARG}}) \leq 2^{-\lambda} + \epsilon^\star_{\mathsf{DLOG}}(\lambda, t^\star_{\mathsf{ARG}} \cdot \log(\mathsf{q}/\epsilon)) + \epsilon.$$

Polylogarithmic gap
Almost tight

Our followup: Quantum Rewinding for IOP-Based Succinct Arguments

Alessandro Chiesa, Marcel Dall'Agnol, Zijing Di, **Ziyi Guan**, Nick Spooner

Quantum Rewinding for IOP-Based Succinct Arguments

Alessandro Chiesa, Marcel Dall Agnol, Zijing Di, Ziyi Guan, Nicholas Spooner

We analyze the post-quantum security of succinct interactive arguments constructed from interactive oracle proofs (IOPs) and vector commitment schemes. We prove that an interactive variant of the BCS transformation is secure in the standard model against quantum adversaries when the vector commitment scheme is collapsing. Our proof builds on and extends prior work on the post-quantum security of Kilians succinct interactive argument, which is instead based on probabilistically checkable proofs (PCPs). We introduce a new quantum rewinding strategy that works across any number of rounds. As a consequence of our results, we obtain standard-model post-quantum secure succinct arguments with the best asymptotic complexity known.

Thank you!

https://eprint.iacr.org/2024/1434

On the price of rewinding

Goal: achieve $\epsilon_{\text{ARG}} = 2^{-40}$ against adversaries of size 2^{60} for Kilian's protocol.

Standard model

$t_{\text{VC}} = O\left(\frac{l}{\epsilon} \cdot t_{\text{ARG}}\right)$

For every $x \notin L$ and $\epsilon > 0$,
$\epsilon_{\text{ARG}}(\lambda, x, t_{\text{ARG}}) \leq \epsilon_{\text{PCP}}(x) + \epsilon_{\text{VC}}(\lambda, l(x), \mathsf{q}(x), t_{\text{VC}}) + \epsilon.$

- Suppose $\epsilon_{\text{PCP}} = 2^{-42}$ with $l = 2^{30}$.
- Suppose $\epsilon_{\text{VC}} = (\lambda, l, \mathsf{q}, t_{\text{VC}}) \leq \frac{t_{\text{VC}}^2}{2^\lambda}$ (achieved by ideal Merkle trees).
- Setting $\epsilon := 2^{-42}$:
  - $t_{\text{VC}} \leq 4 \cdot \frac{2^{30}}{2^{-42}} \cdot t_{\text{ARG}} < 2^{80} \cdot t_{\text{ARG}}$
  - $\epsilon_{\text{VC}} \leq \frac{(2^{80} \cdot t_{\text{ARG}})^2}{2^\lambda} = 2^{160-\lambda} \cdot t_{\text{ARG}}^2 = 2^{280-\lambda}$
- Set $\lambda = 322$ to achieve the desired bound.

Random oracle model

For every $x \notin L$, [CY24]

$\epsilon_{\text{ARG}}(\lambda, x, t_{\text{ARG}}) \leq \epsilon_{\text{PCP}}(x) + \frac{t_{\text{ARG}}^2}{2^\lambda}.$

- Suppose $\epsilon_{\text{PCP}} = 2^{-42}$
- $\epsilon_{\text{VC}} \leq \frac{t_{\text{ARG}}^2}{2^\lambda} = 2^{120-\lambda}$
- Set $\lambda = 162$ to achieve the desired bound.

- If the hash function is assumed ideal then extraction is straightline.
- If the hash function is merely collision-resistant then extraction is rewinding.
These computations illustrate the **PRICE OF REWINDING**.