SNARKs from Hash Functions

Nick Spooner

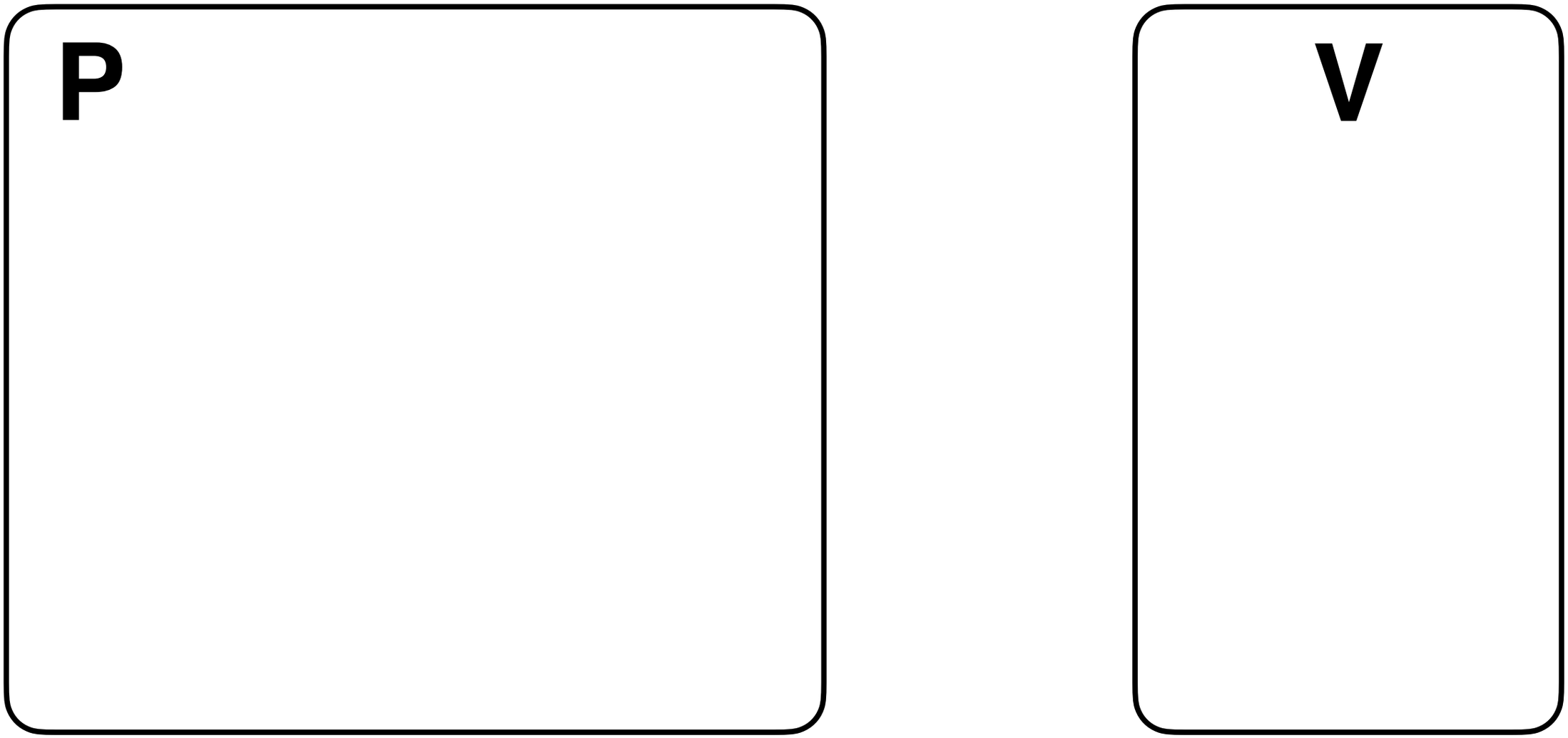
Boston University (➡️ Warwick)

In the Beginning:

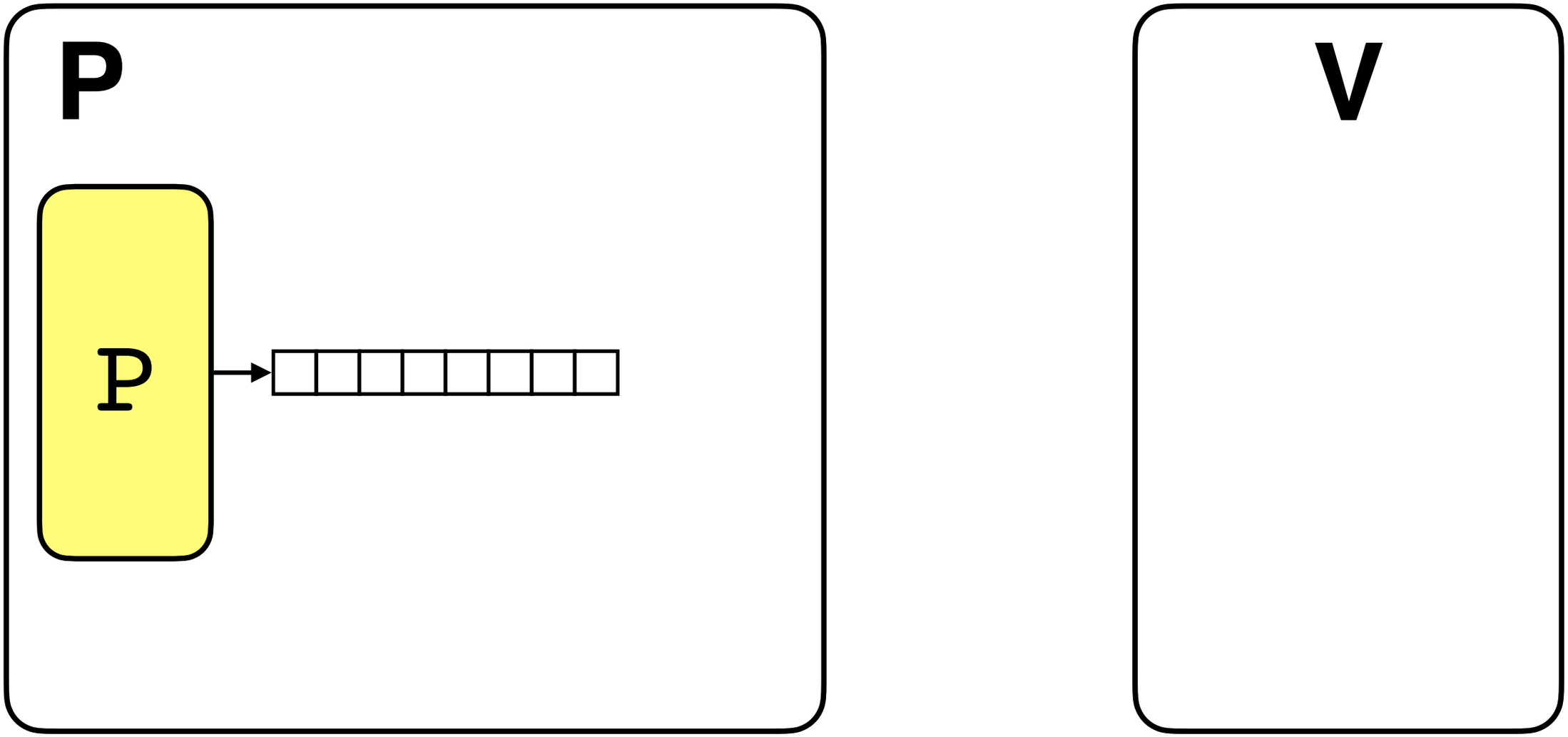
SNARKs from PCPs

Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.

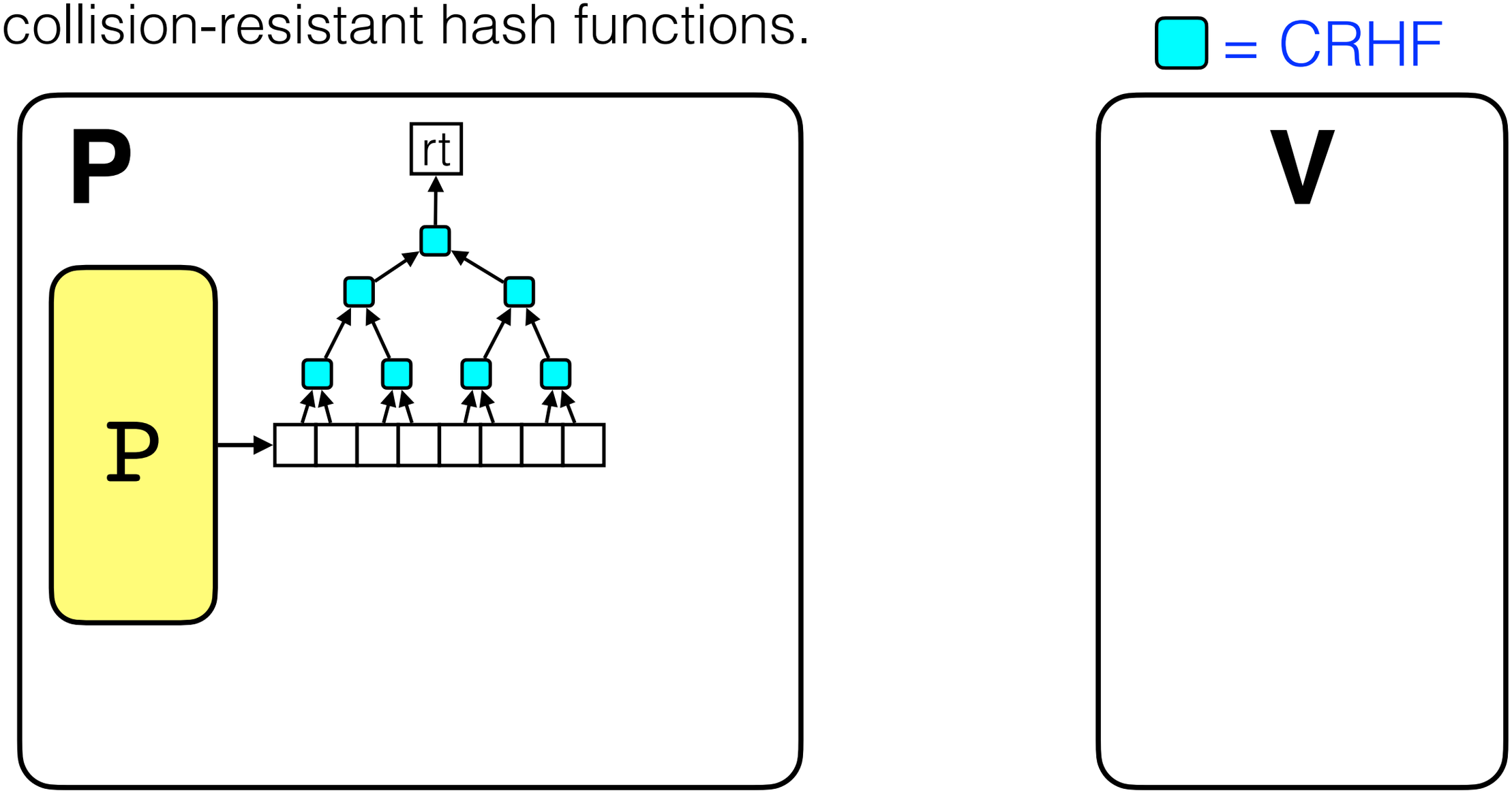
Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.



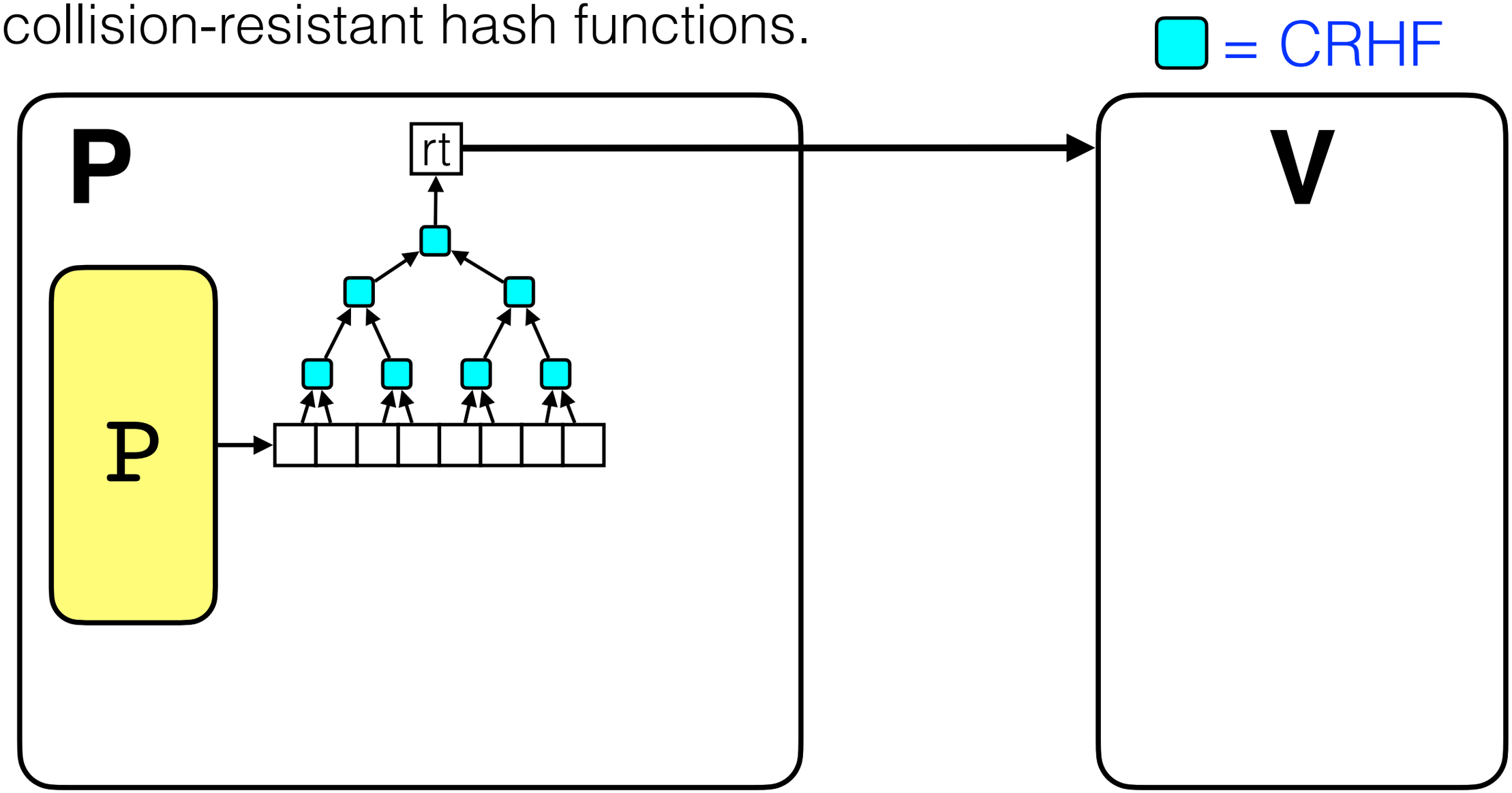
Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.



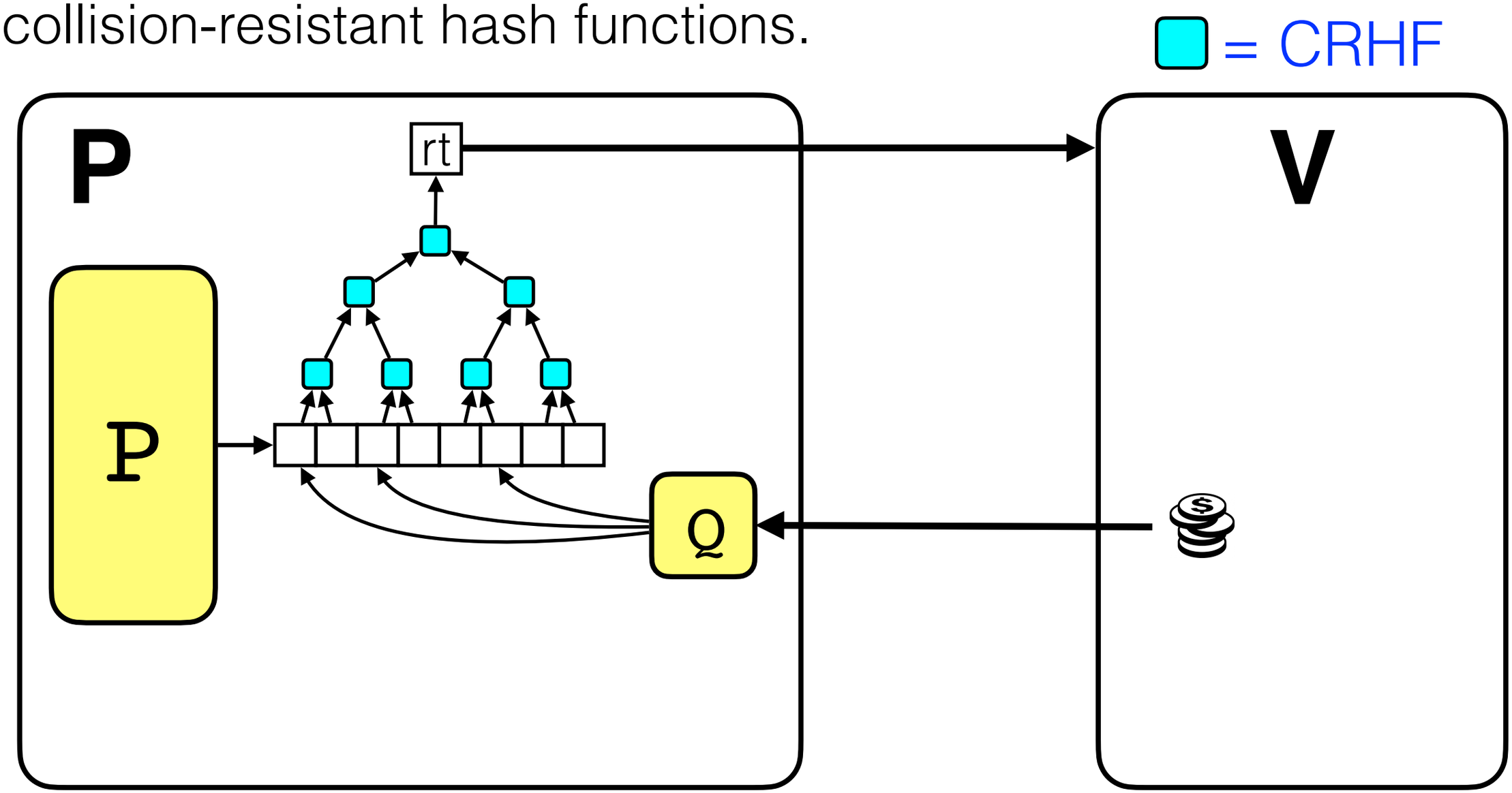
Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.



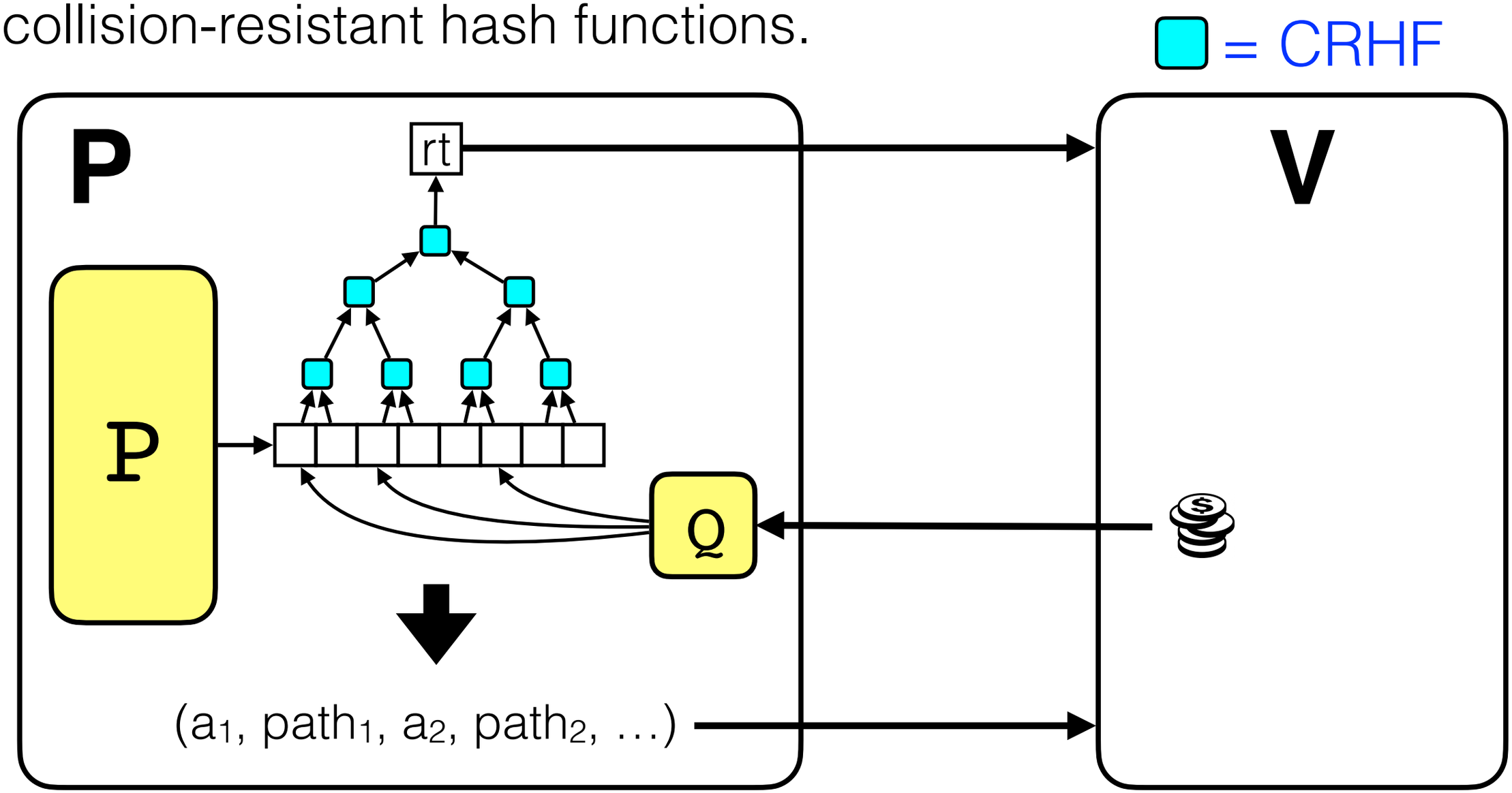
Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.



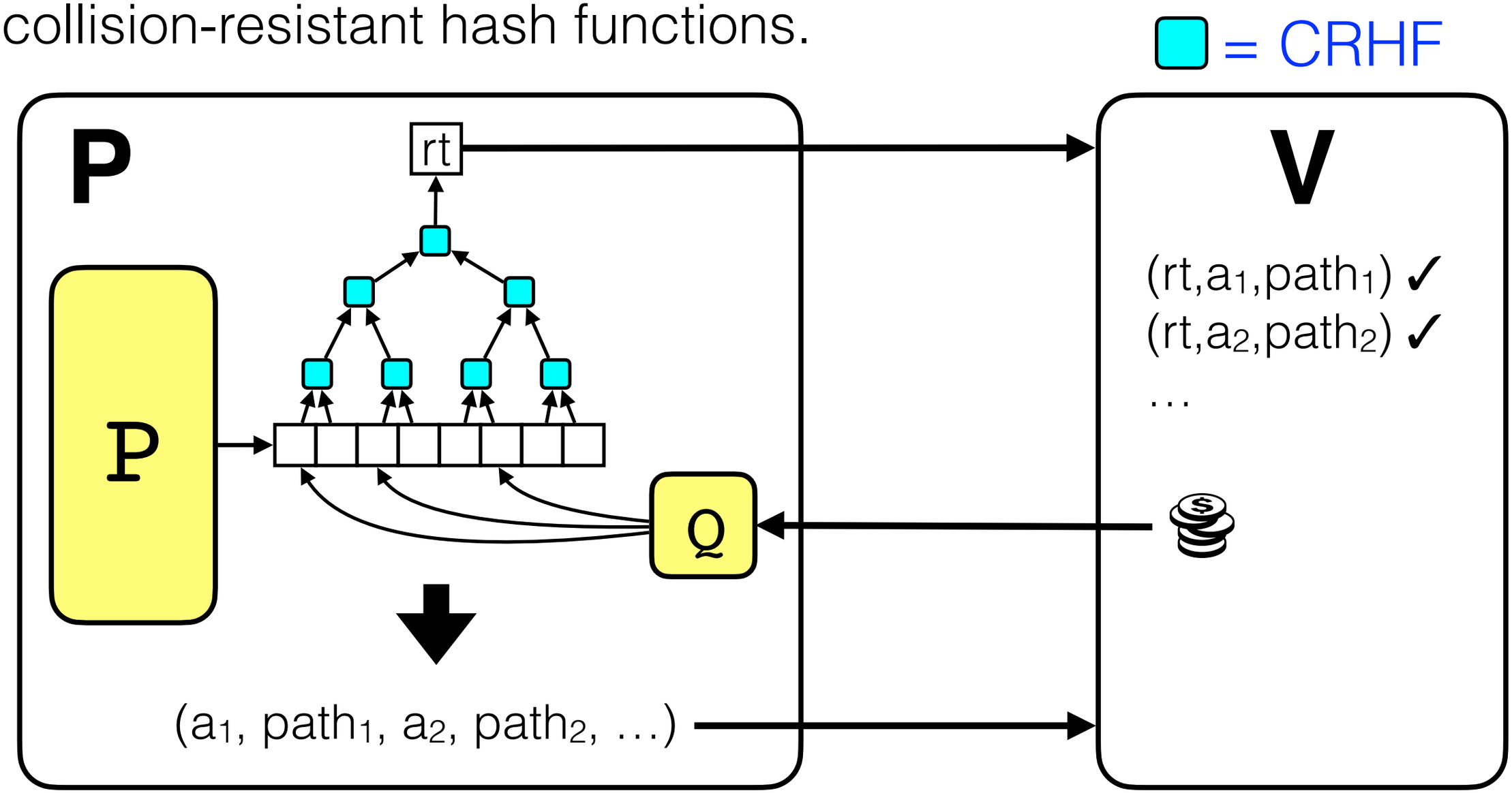
Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.



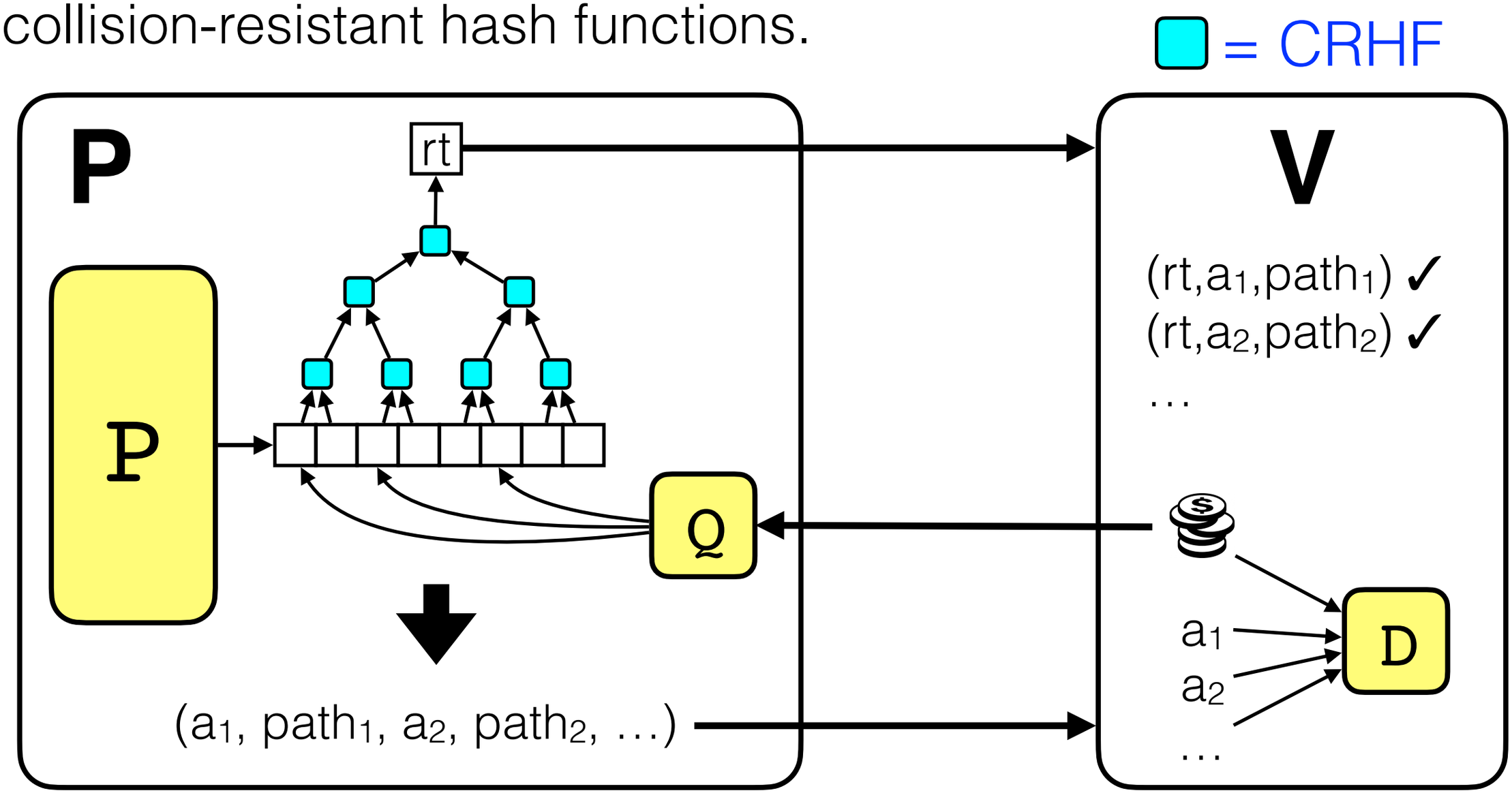
Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.



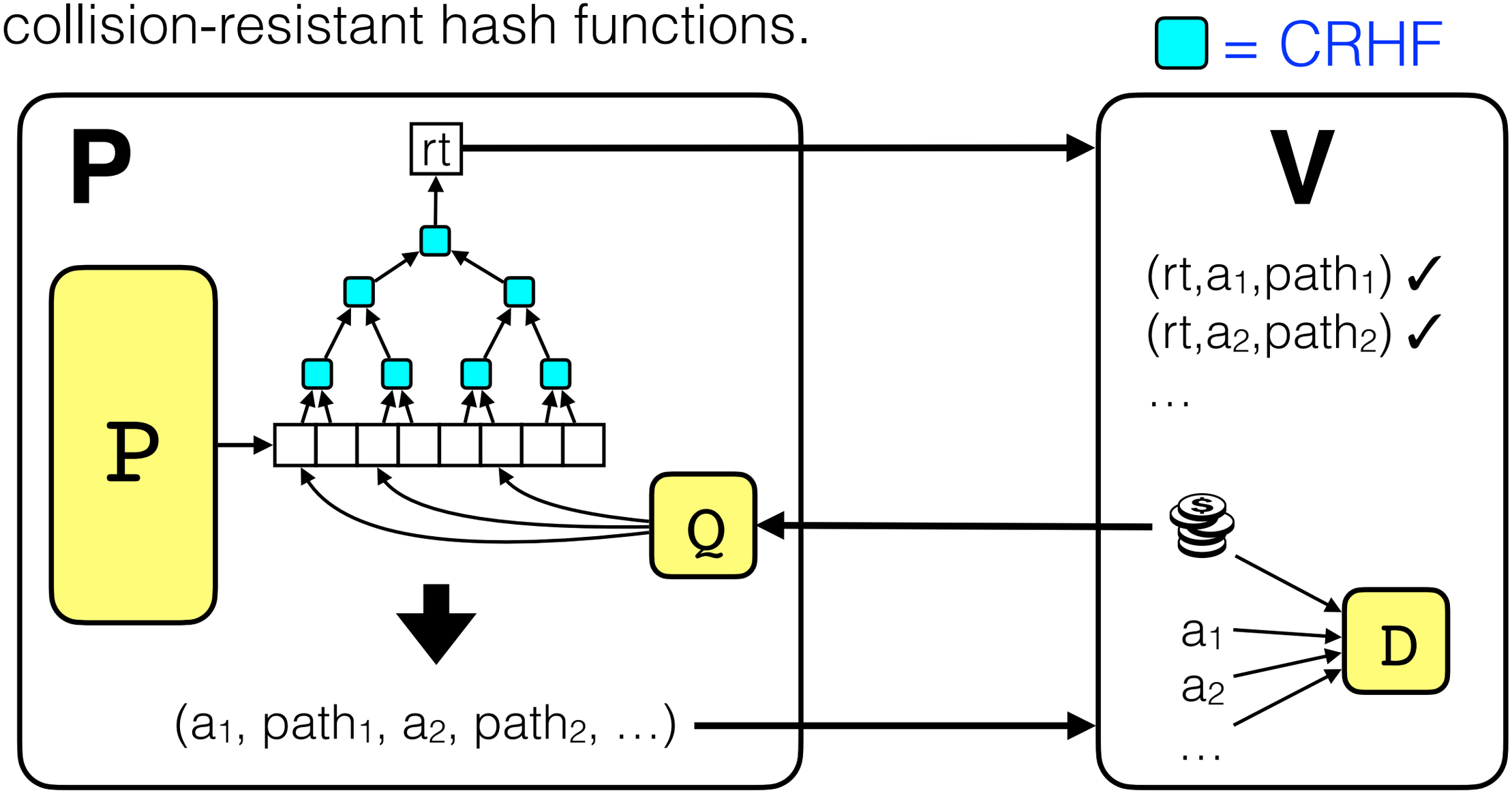
Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.



Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.

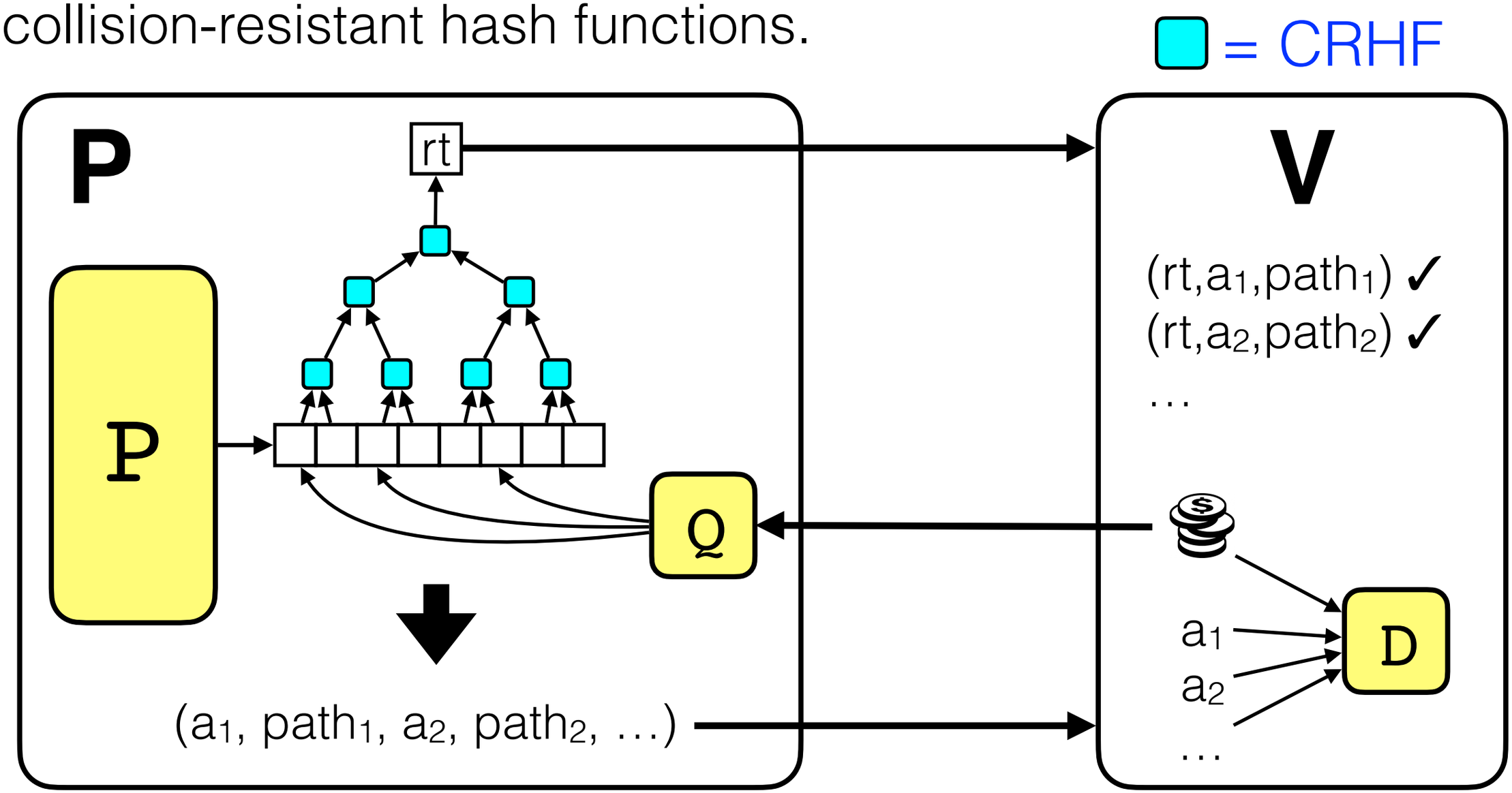


Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.



PCP parameters:
- ε_{PCP} is soundness
- ℓ is proof length
- Σ is proof alphabet
- q is number of queries

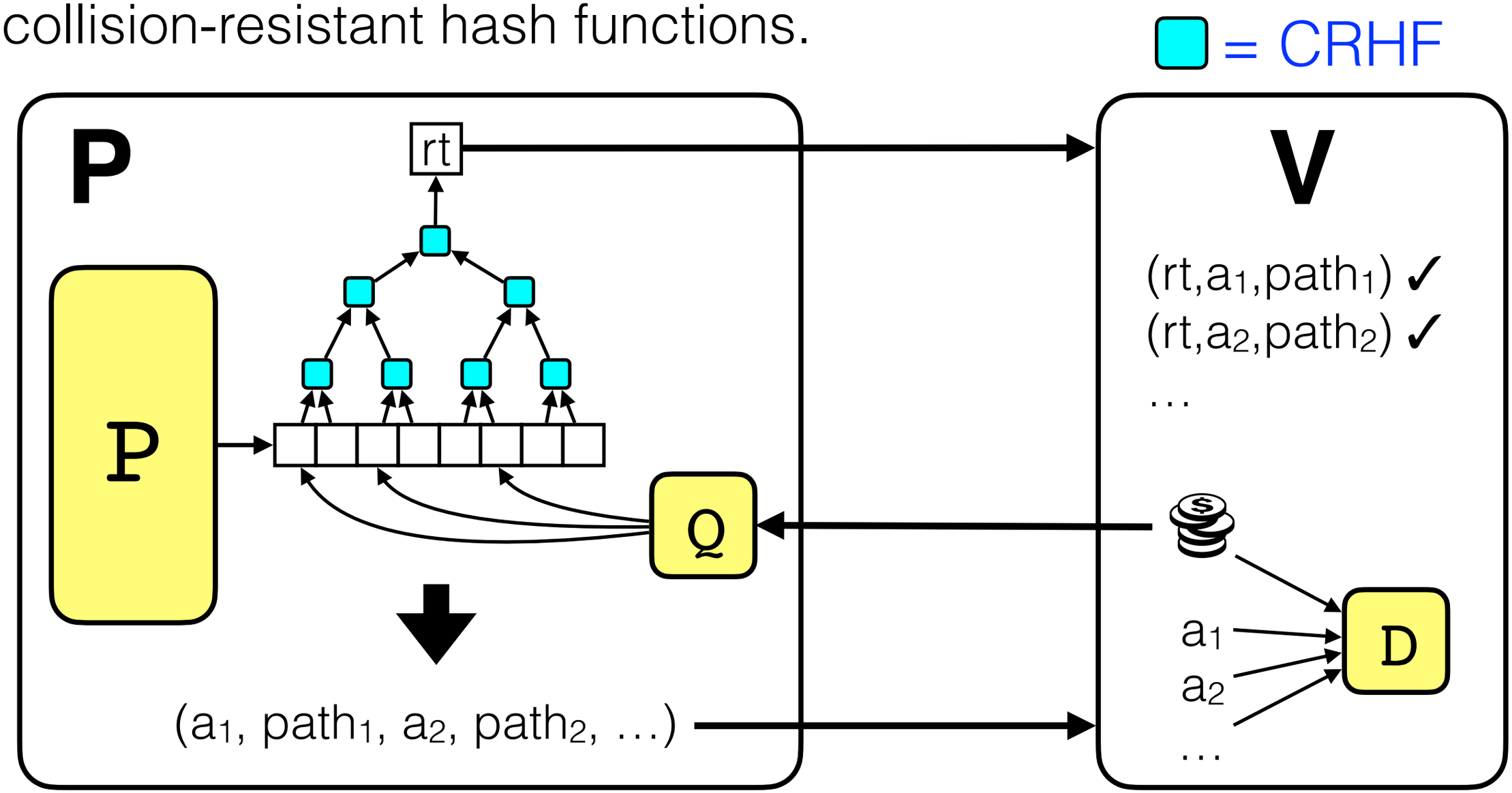
Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.



soundness = $\varepsilon_{PCP} + \varepsilon_{CRHF}$

PCP parameters:
- ε_{PCP} is soundness
- ℓ is proof length
- Σ is proof alphabet
- q is number of queries

Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.

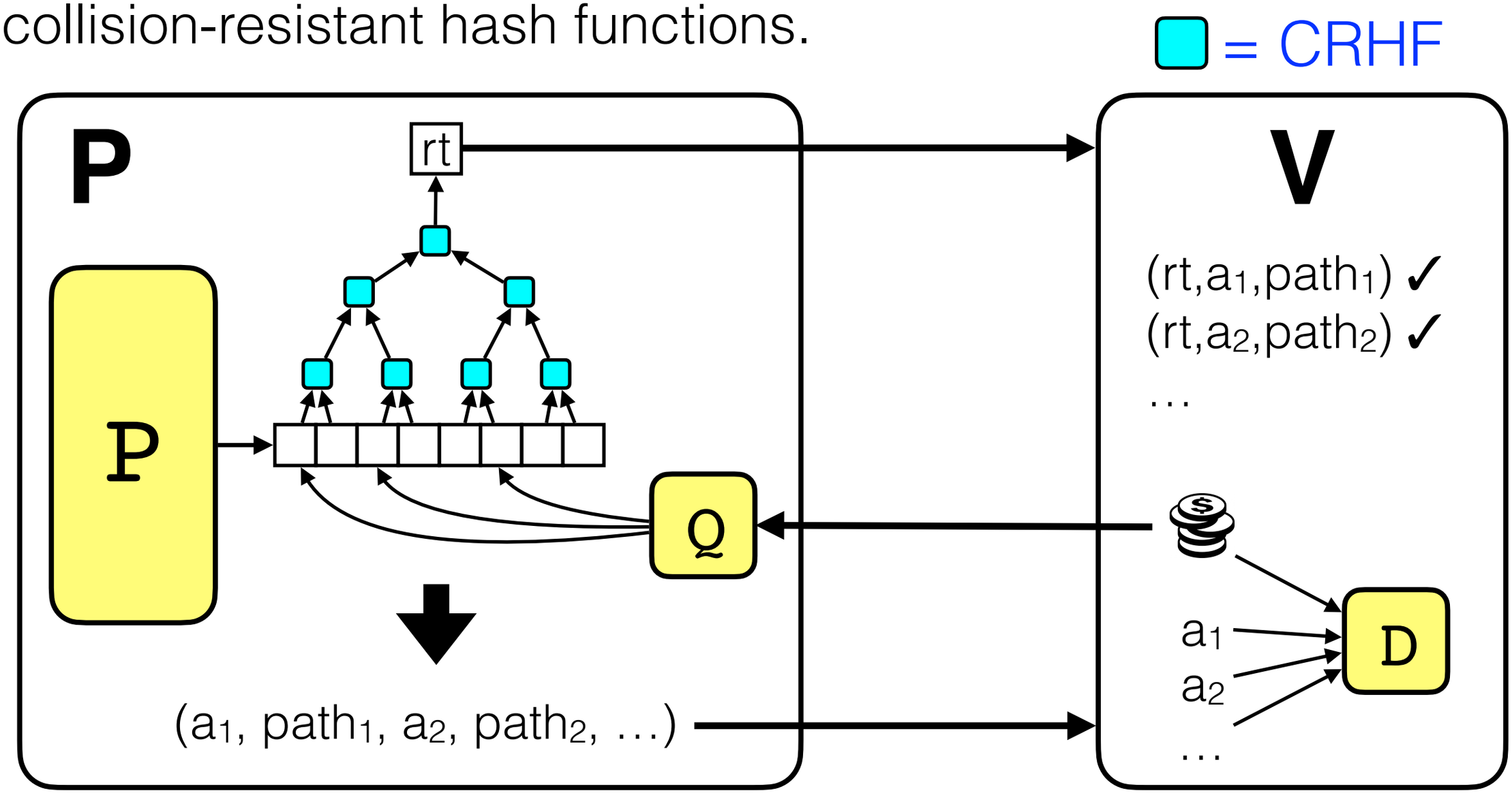


soundness = $\varepsilon_{PCP} + \varepsilon_{CRHF}$

proving time = PCP-prove + $t_H \times \ell$

PCP parameters:
- ε_{PCP} is soundness
- ℓ is proof length
- Σ is proof alphabet
- q is number of queries

Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.



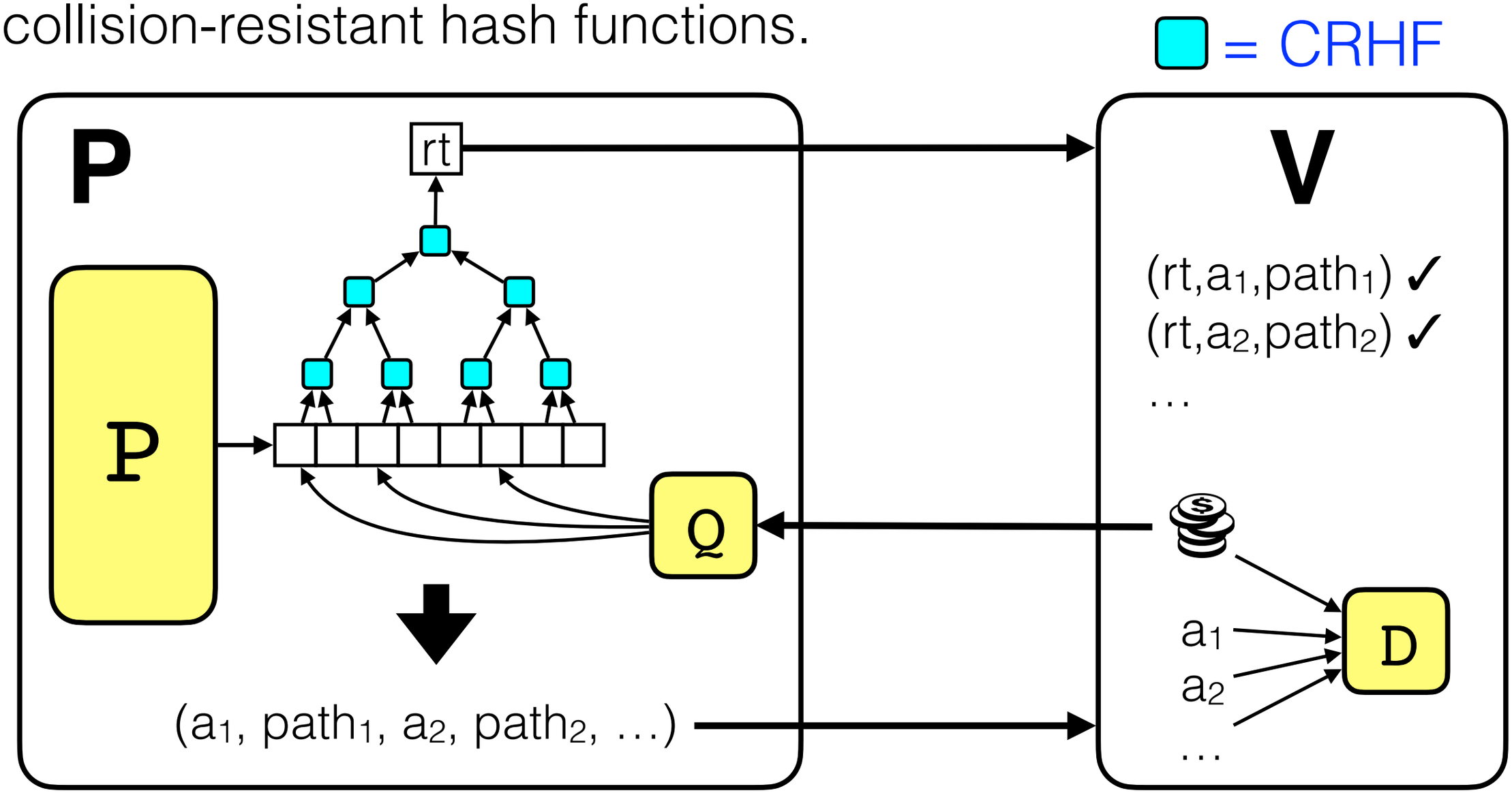
soundness = $\varepsilon_{PCP} + \varepsilon_{CRHF}$

proving time = PCP-prove + $t_H \times \ell$

verification time = PCP-verify + $t_H \times q \times \log \ell$

PCP parameters:
- ε_{PCP} is soundness
- ℓ is proof length
- Σ is proof alphabet
- q is number of queries

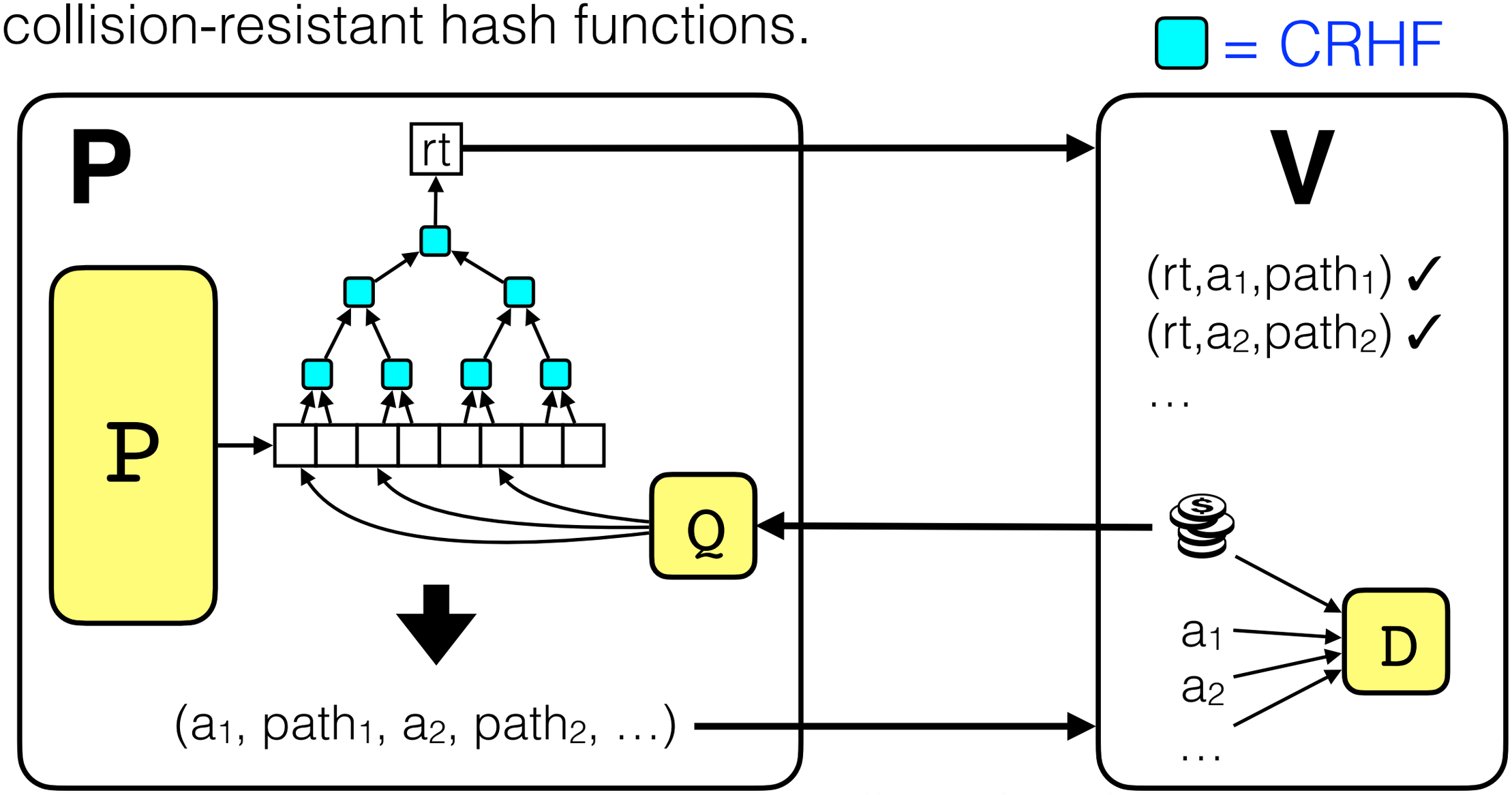
Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.



soundness = $\varepsilon_{PCP} + \varepsilon_{CRHF}$
proving time = PCP-prove + $t_H \times \ell$
verification time = PCP-verify + $t_H \times q \times \log\ell$
communication = $q \times (\log|\Sigma| + 2\kappa \log\ell)$

PCP parameters:
- ε_{PCP} is soundness
- ℓ is proof length
- Σ is proof alphabet
- q is number of queries

Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.



small overheads

soundness = $\varepsilon_{PCP} + \varepsilon_{CRHF}$

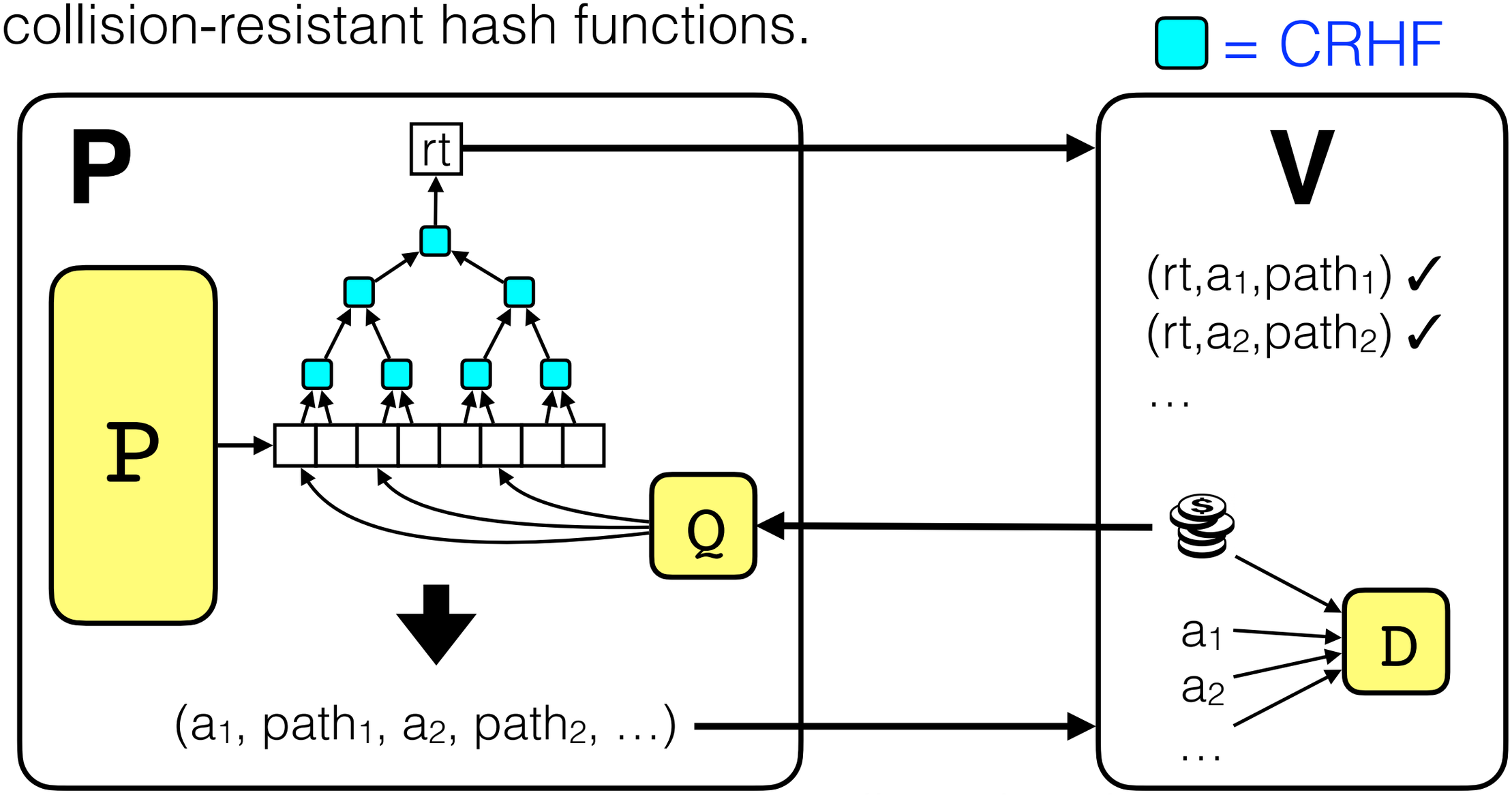
proving time = PCP-prove + $t_H \times \ell$

verification time = PCP-verify + $t_H \times q \times \log \ell$

communication = $q \times (\log|\Sigma| + 2\kappa \log \ell)$

PCP parameters:
- ε_{PCP} is soundness
- ℓ is proof length
- Σ is proof alphabet
- q is number of queries

Theorem [Kilian 92]
There exists a succinct **interactive** argument for NP, assuming collision-resistant hash functions.



small overheads

soundness = $\varepsilon_{PCP} + \varepsilon_{CRHF}$

proving time = PCP-prove + $t_H \times \ell$

verification time = PCP-verify + $t_H \times q \times \log \ell$

communication = $q \times (\log|\Sigma| + 2\kappa \log \ell) = \text{poly}(\kappa)$

PCP parameters:
- ε_{PCP} is soundness
- ℓ is proof length
- Σ is proof alphabet
- q is number of queries

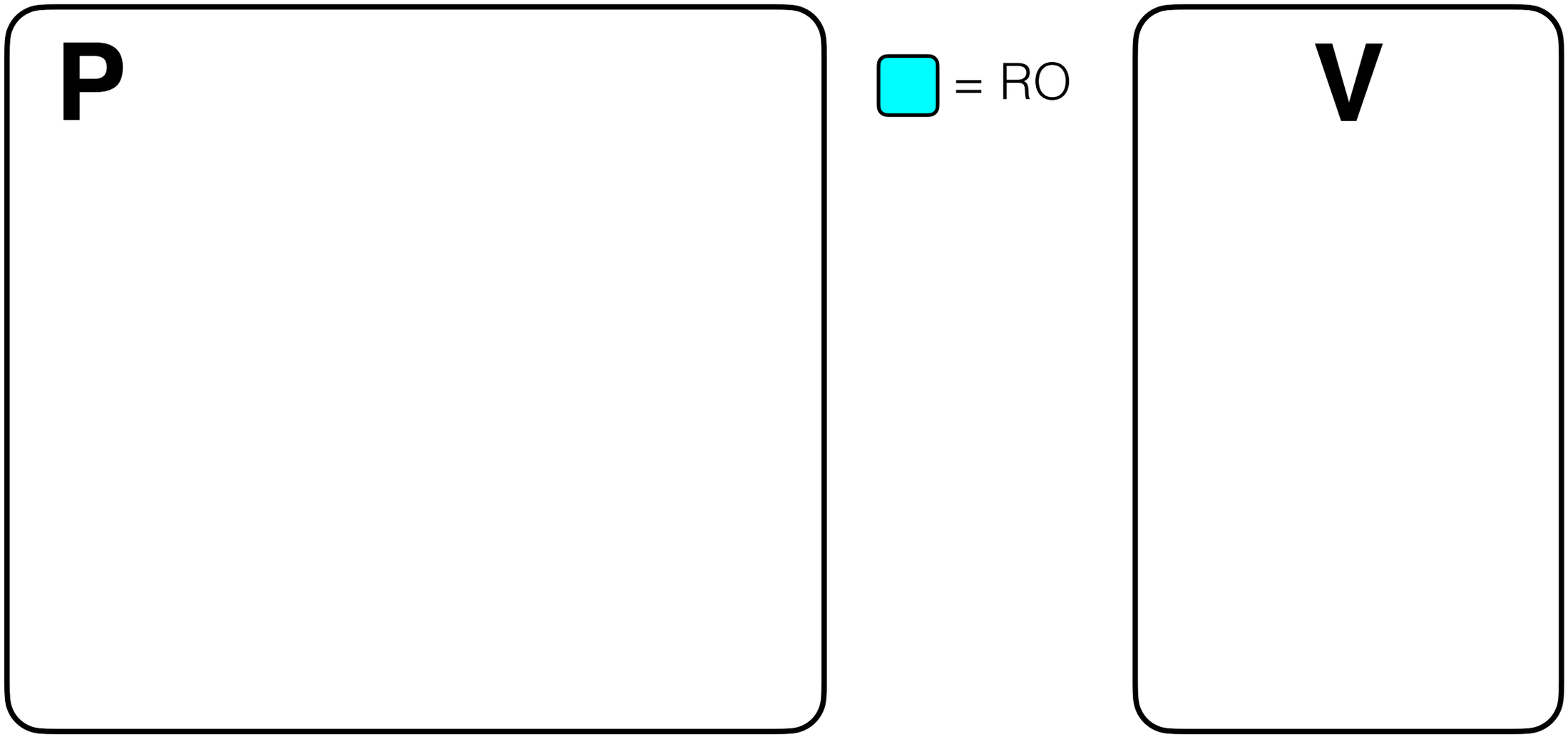
Theorem [Micali 94]: There is a SNARG for NP in the random oracle model (unconditionally). *(Actually a zkSNARK [Valiant 08, IMSX15].)*

Theorem [Micali 94]: There is a SNARG for NP in the random oracle model (unconditionally). *(Actually a zkSNARK [Valiant 08, IMSX15].)*

Idea: apply Fiat-Shamir transformation to Kilian's protocol!

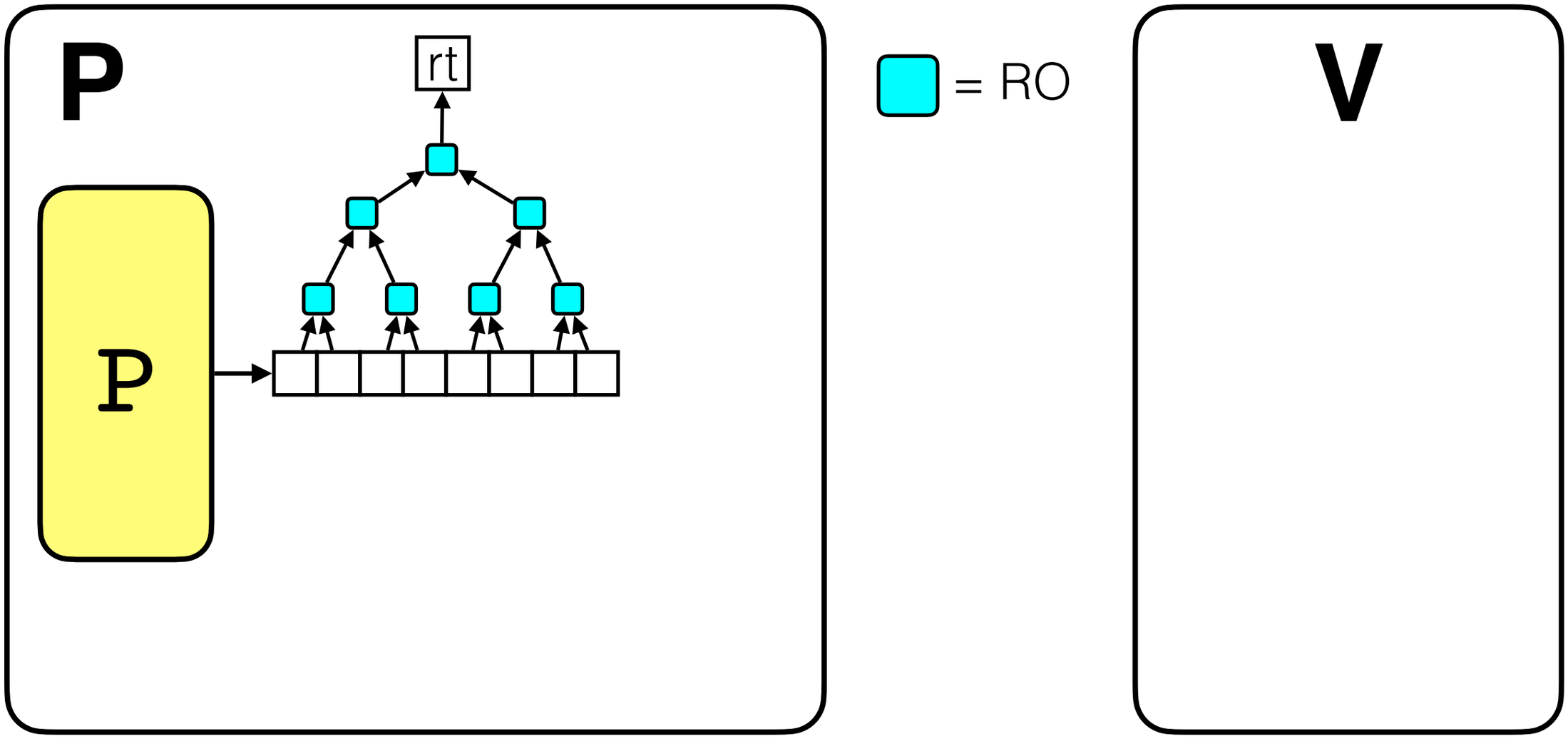
Theorem [Micali 94]: There is a SNARG for NP in the random oracle model (unconditionally). *(Actually a zkSNARK [Valiant 08, IMSX15].)*

Idea: apply Fiat-Shamir transformation to Kilian's protocol!



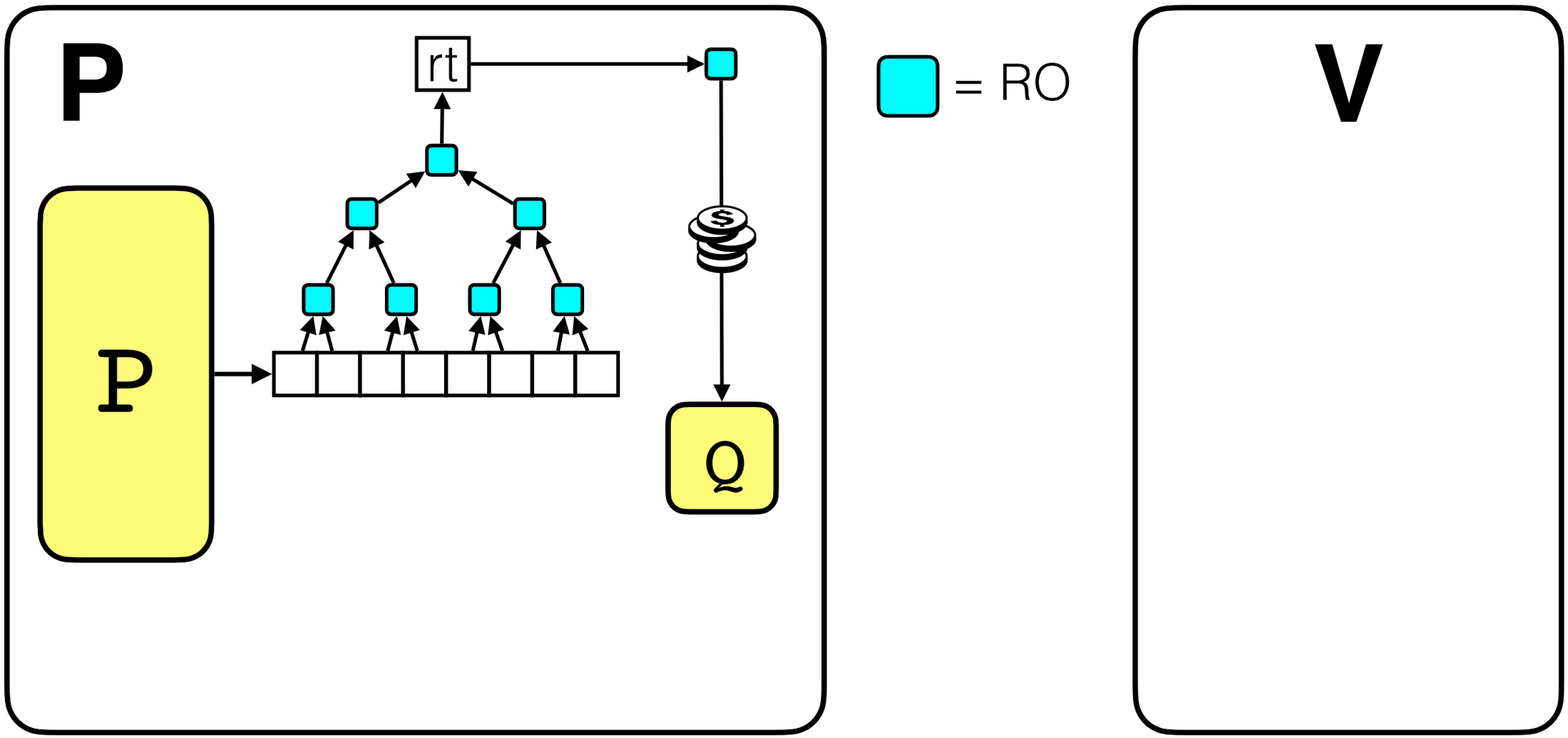
Theorem [Micali 94]: There is a SNARG for NP in the random oracle model (unconditionally). *(Actually a zkSNARK [Valiant 08, IMSX15].)*

Idea: apply Fiat-Shamir transformation to Kilian's protocol!



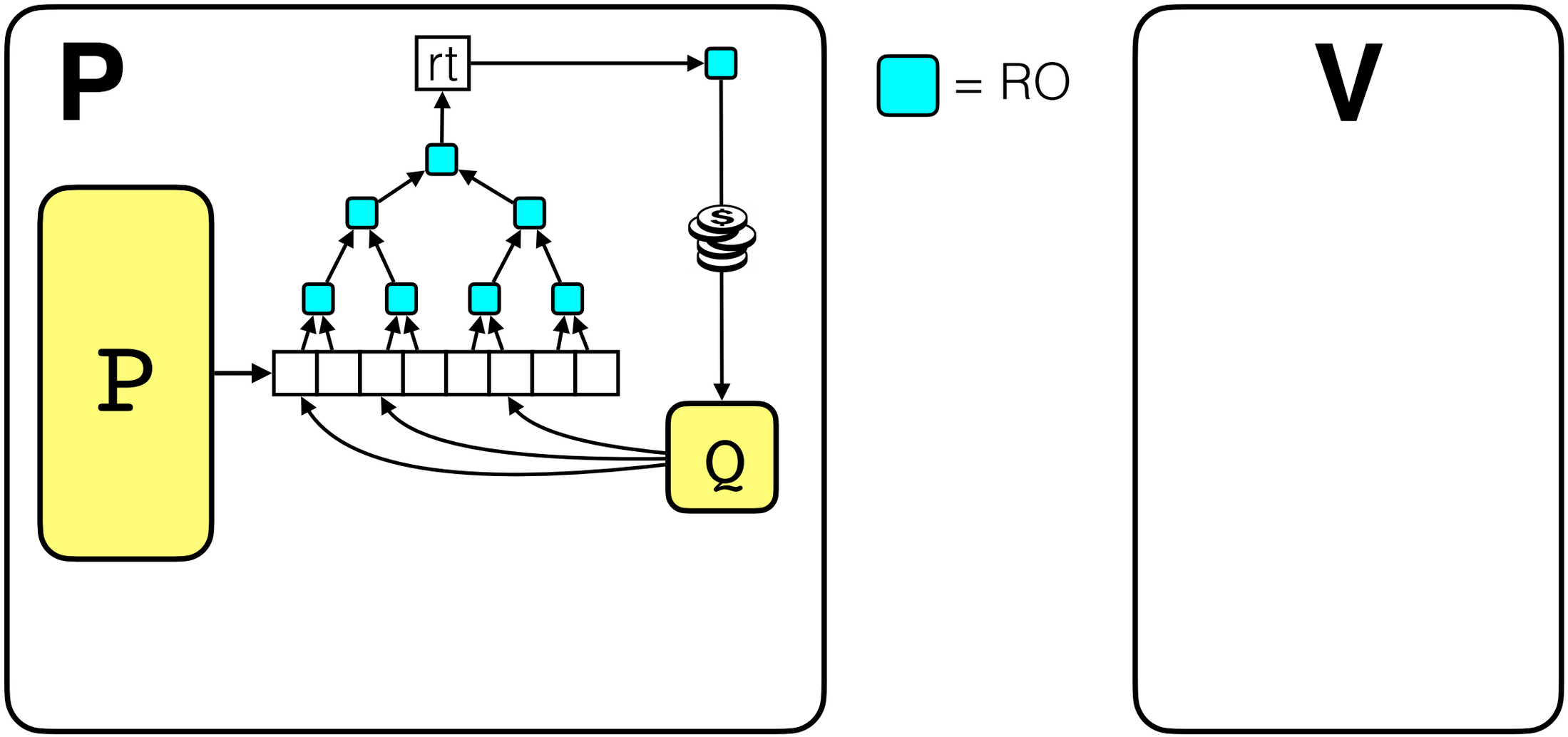
Theorem [Micali 94]: There is a SNARG for NP in the random oracle model (unconditionally). *(Actually a zkSNARK [Valiant 08, IMSX15].)*

Idea: apply Fiat-Shamir transformation to Kilian's protocol!



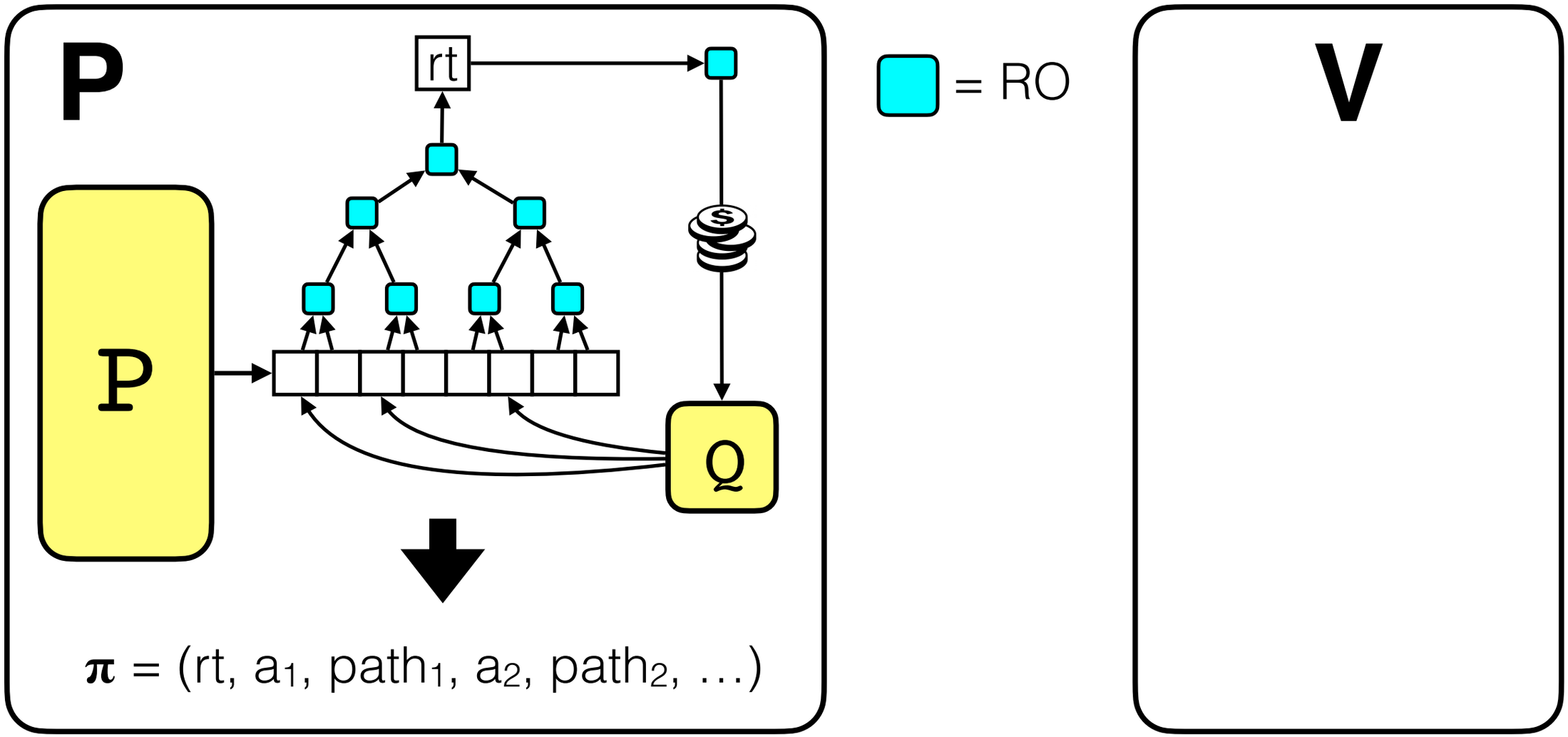
Theorem [Micali 94]: There is a SNARG for NP in the random oracle model (unconditionally). *(Actually a zkSNARK [Valiant 08, IMSX15].)*

Idea: apply Fiat-Shamir transformation to Kilian's protocol!



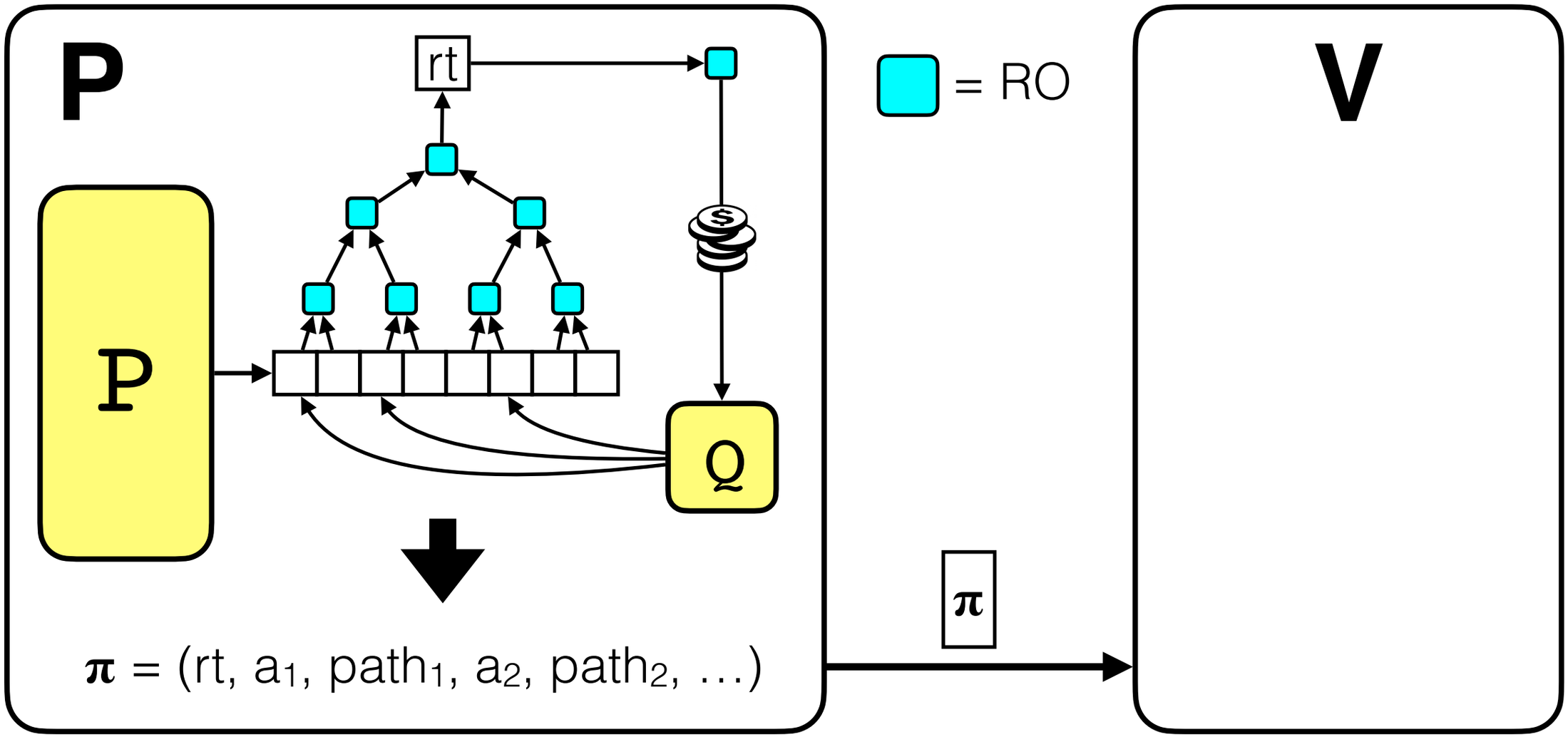
Theorem [Micali 94]: There is a SNARG for NP in the random oracle model (unconditionally). *(Actually a zkSNARK [Valiant 08, IMSX15].)*

Idea: apply Fiat-Shamir transformation to Kilian's protocol!



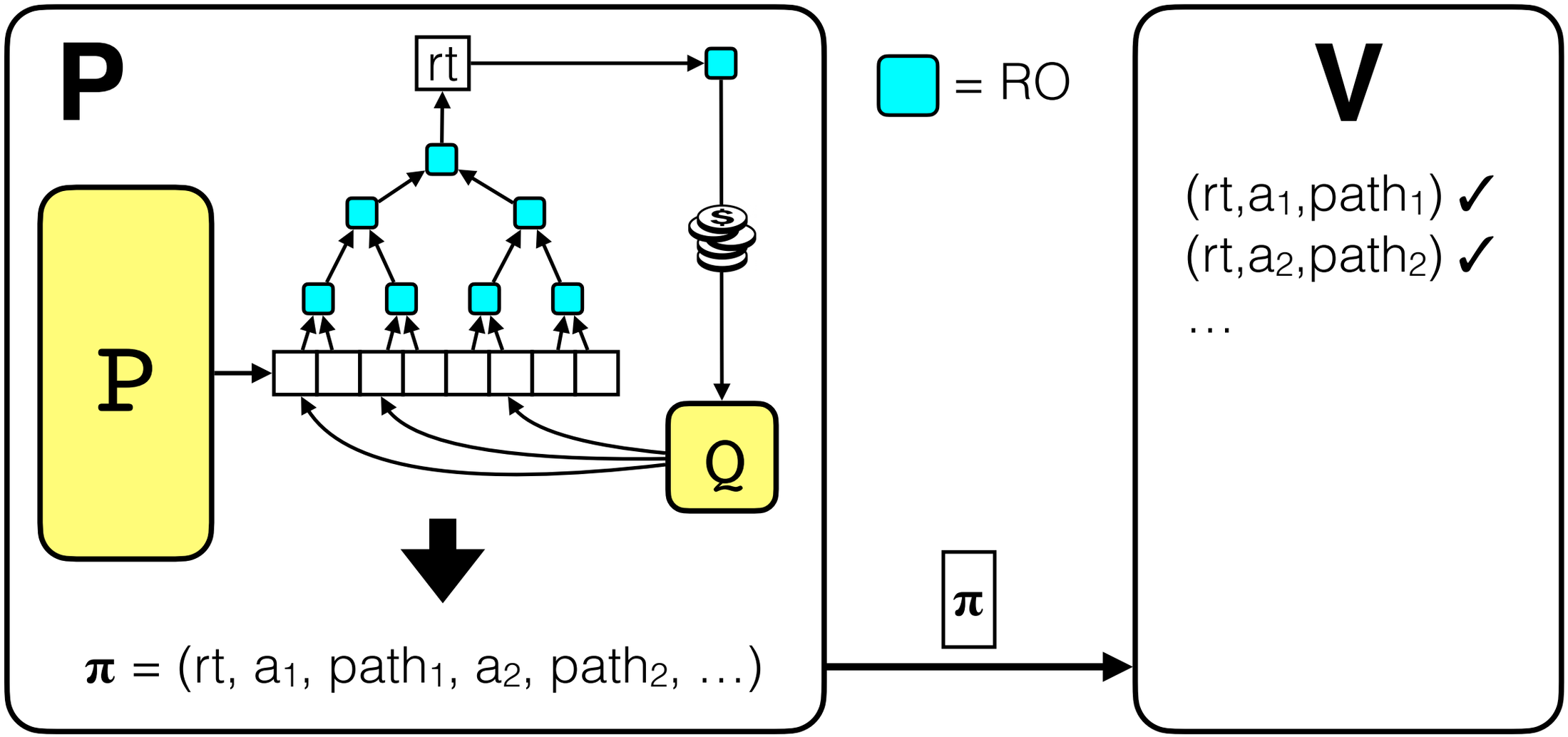
Theorem [Micali 94]: There is a SNARG for NP in the random oracle model (unconditionally). *(Actually a zkSNARK [Valiant 08, IMSX15].)*

Idea: apply Fiat-Shamir transformation to Kilian's protocol!



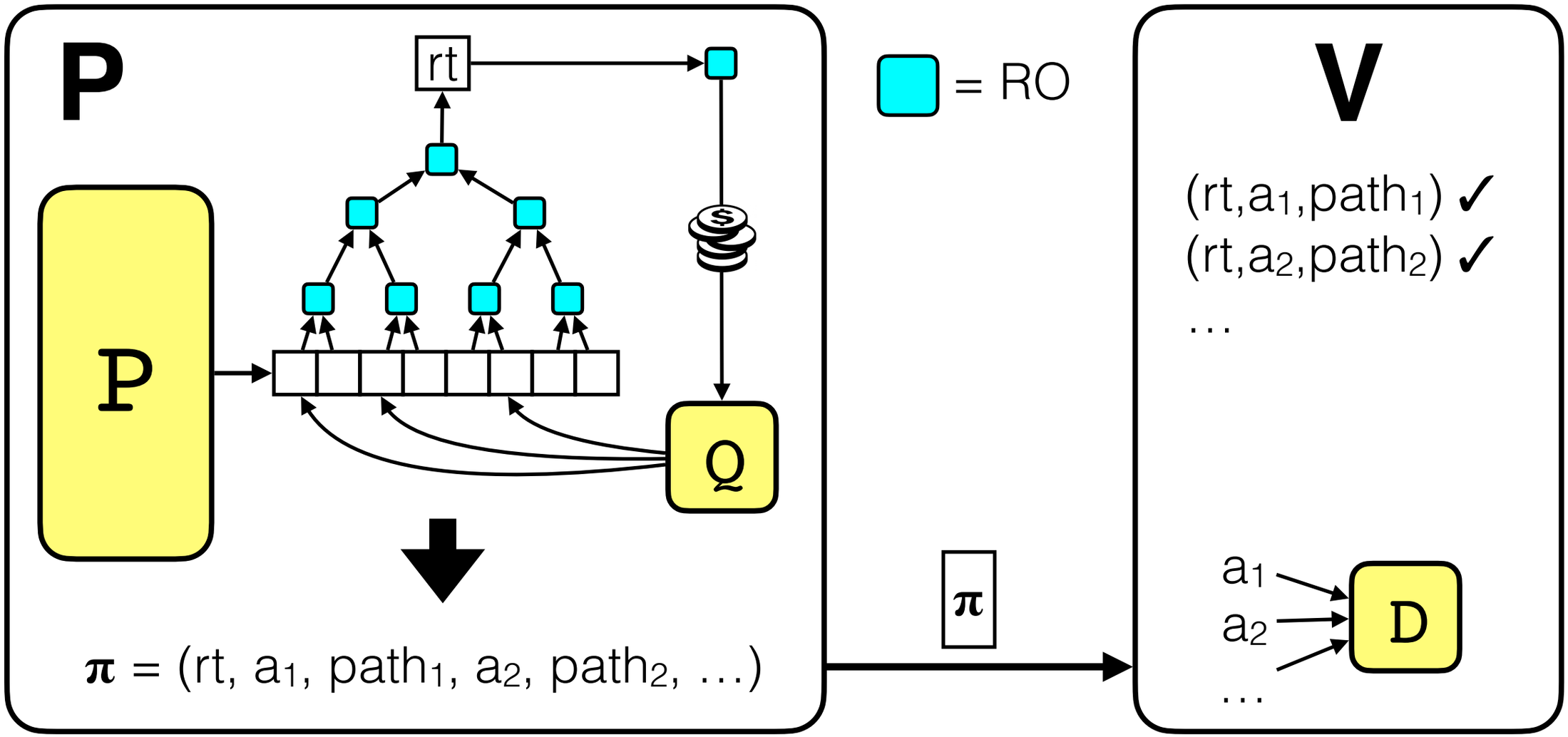
Theorem [Micali 94]: There is a SNARG for NP in the random oracle model (unconditionally). *(Actually a zkSNARK [Valiant 08, IMSX15].)*

Idea: apply Fiat-Shamir transformation to Kilian's protocol!



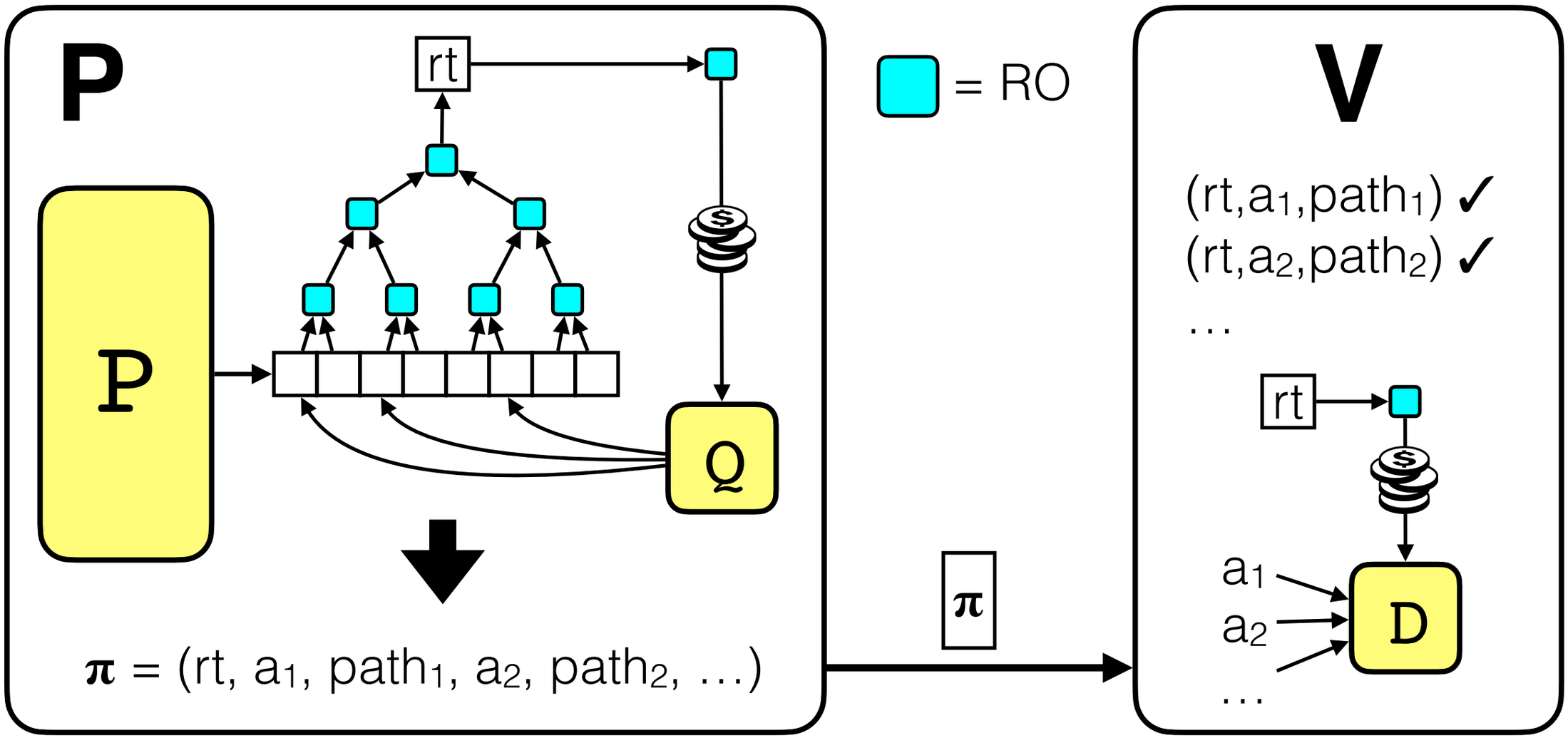
Theorem [Micali 94]: There is a SNARG for NP in the random oracle model (unconditionally). *(Actually a zkSNARK [Valiant 08, IMSX15].)*

Idea: apply Fiat-Shamir transformation to Kilian's protocol!



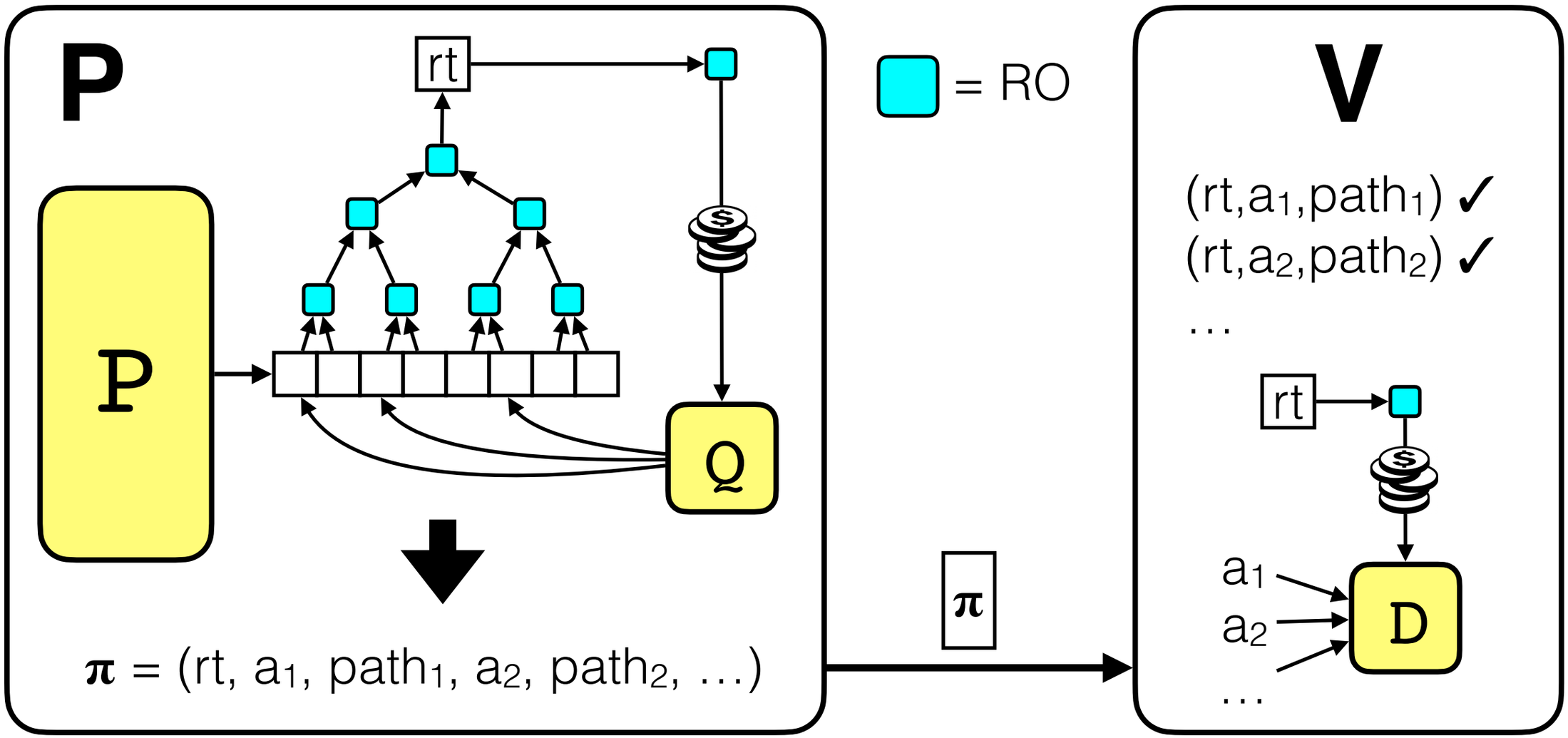
Theorem [Micali 94]: There is a SNARG for NP in the random oracle model (unconditionally). *(Actually a zkSNARK [Valiant 08, IMSX15].)*

Idea: apply Fiat-Shamir transformation to Kilian's protocol!



Theorem [Micali 94]: There is a SNARG for NP in the random oracle model (unconditionally). *(Actually a zkSNARK [Valiant 08, IMSX15].)*

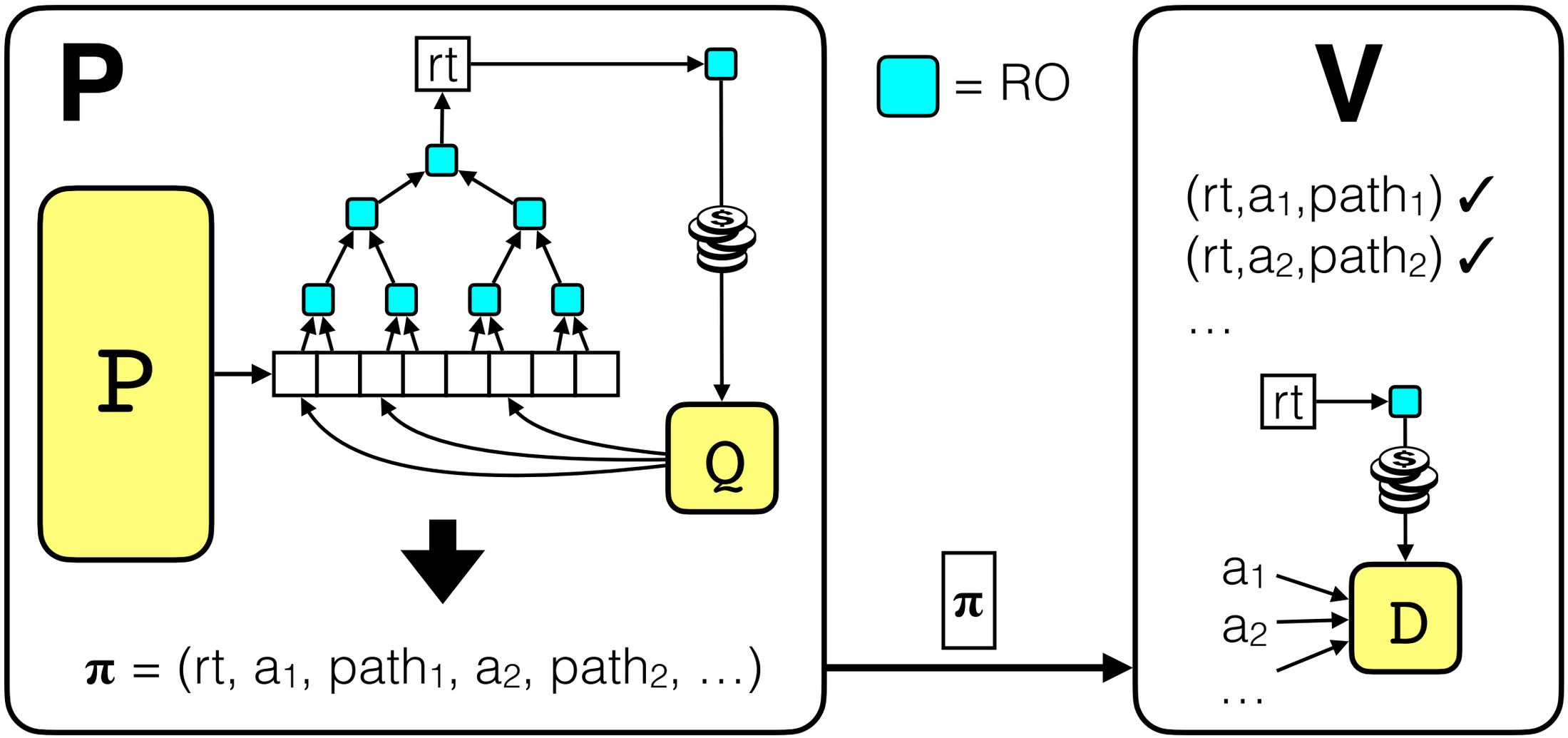
Idea: apply Fiat-Shamir transformation to Kilian's protocol!



soundness = $T \cdot \varepsilon_{PCP} + T^2 \cdot 2^{-\kappa}$ for T-query adversaries

Theorem [Micali 94]: There is a SNARG for NP in the random oracle model (unconditionally). *(Actually a zkSNARK [Valiant 08, IMSX15].)*

Idea: apply Fiat-Shamir transformation to Kilian's protocol!

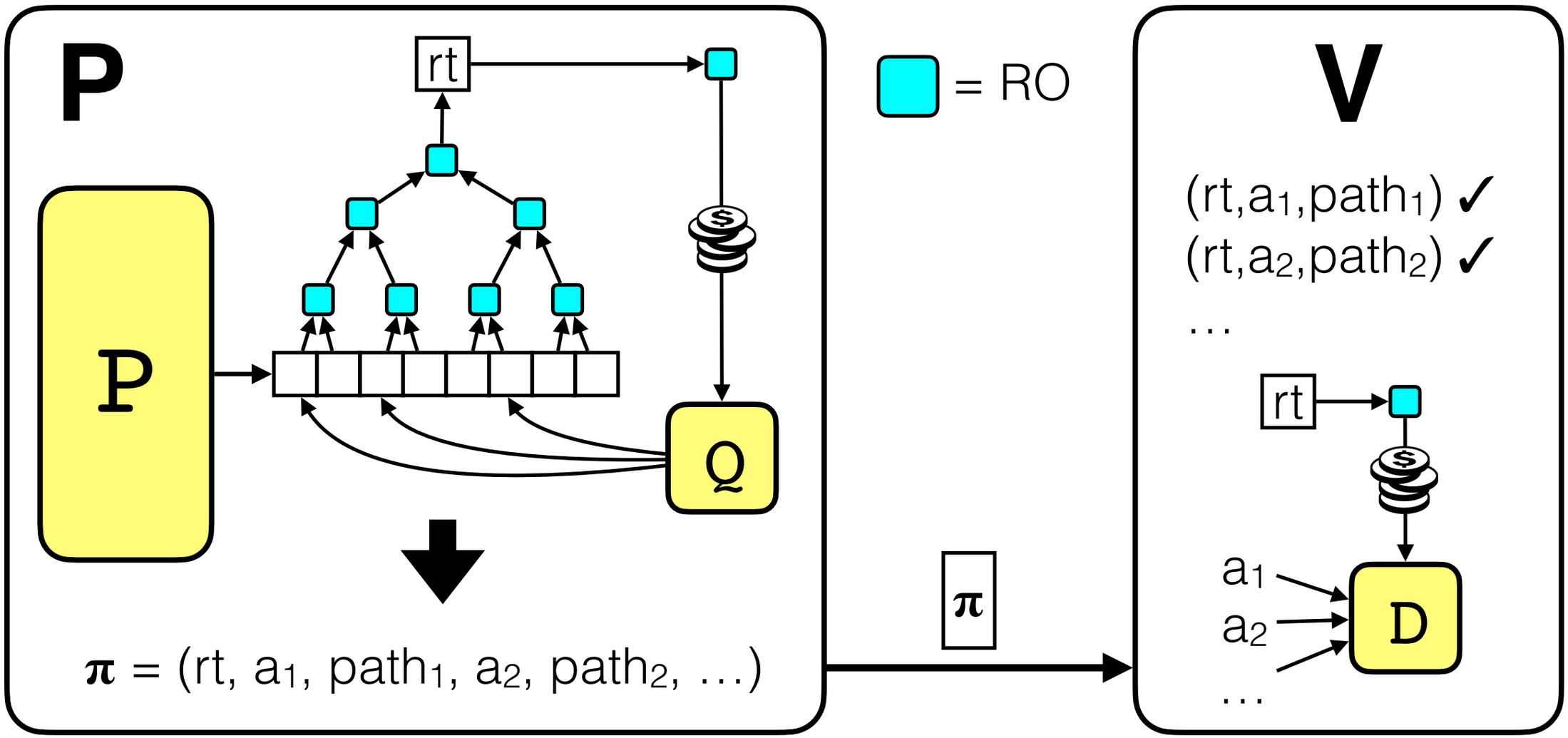


soundness = $T \cdot \varepsilon_{PCP} + T^2 \cdot 2^{-\kappa}$ for T-query adversaries

$= \varepsilon_{CRHF}$

Theorem [Micali 94]: There is a SNARG for NP in the random oracle model (unconditionally). *(Actually a zkSNARK [Valiant 08, IMSX15].)*

Idea: apply Fiat-Shamir transformation to Kilian's protocol!



soundness = $T \cdot \varepsilon_{PCP} + T^2 \cdot 2^{-\kappa}$ for T-query adversaries

adversary can attack PCP! needs better PCP soundness than Kilian, less efficient

$= \varepsilon_{CRHF}$

PCP-Based SNARGs

GOOD

BAD

PCP-Based SNARGs

GOOD

BAD

- random oracles don't exist*

PCP-Based SNARGs

GOOD

- random oracle is “robust”

BAD

- random oracles don’t exist*

PCP-Based SNARGs

GOOD

- random oracle is “robust”
- black-box use of lightweight crypto (any hash function)

SNARG-prove ≈ PCP-prove

SNARG-verify ≈ PCP-verify

BAD

- random oracles don’t exist*

PCP-Based SNARGs

GOOD

- random oracle is “robust”

- black-box use of lightweight crypto (any hash function)

  SNARG-prove ≈ PCP-prove

  SNARG-verify ≈ PCP-verify

- transparent (public-coin) setup

  system parameters

  =

  choice of hash function

BAD

- random oracles don’t exist*

PCP-Based SNARGs

GOOD

- random oracle is “robust”

- black-box use of lightweight crypto (any hash function)

  SNARG-prove ≈ PCP-prove

  SNARG-verify ≈ PCP-verify

- transparent (public-coin) setup

  system parameters

  =

  choice of hash function

- post-quantum

BAD

- random oracles don’t exist*

PCP-Based SNARGs

GOOD

- random oracle is “robust”
- black-box use of lightweight crypto (any hash function)

  SNARG-prove ≈ PCP-prove

  SNARG-verify ≈ PCP-verify
- transparent (public-coin) setup

  system parameters

  =

  choice of hash function
- post-quantum

BAD

- random oracles don’t exist*
- **PCPs are expensive!**

PCP-Based SNARGs

GOOD

- random oracle is “robust”
- black-box use of lightweight crypto (any hash function)

  SNARG-prove ≈ PCP-prove

  SNARG-verify ≈ PCP-verify
- transparent (public-coin) setup

  system parameters

  =

  choice of hash function
- post-quantum

BAD

- random oracles don’t exist*
- **PCPs are expensive!**

What do we do?

The Present Day:
SNARKs from IOPs

The Present Day:
SNARKs from IOPs

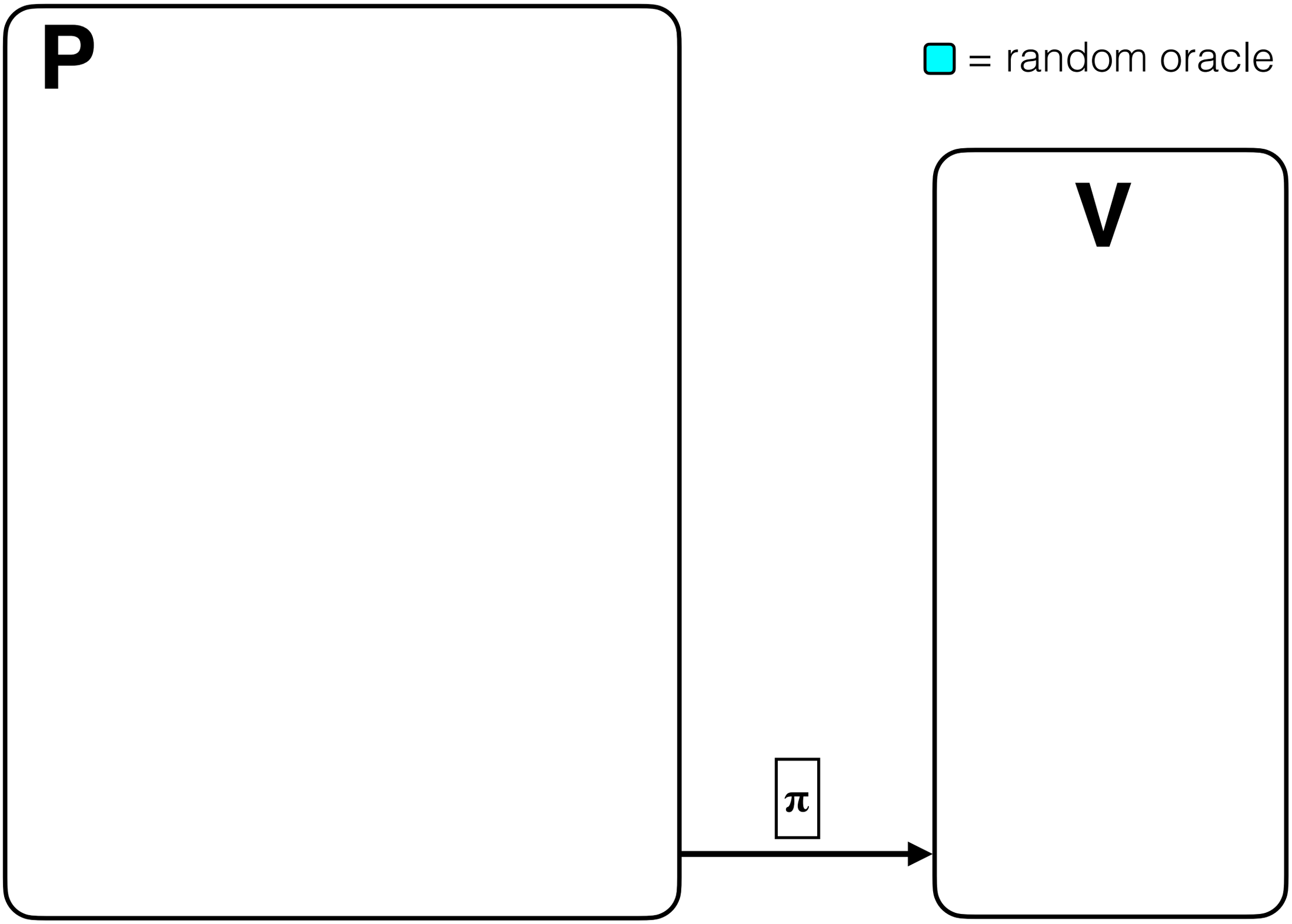
Alessandro's talk: known IOPs are more efficient than known PCPs

[BCS16]

Idea: generalize Kilian to IOPs + apply Fiat-Shamir

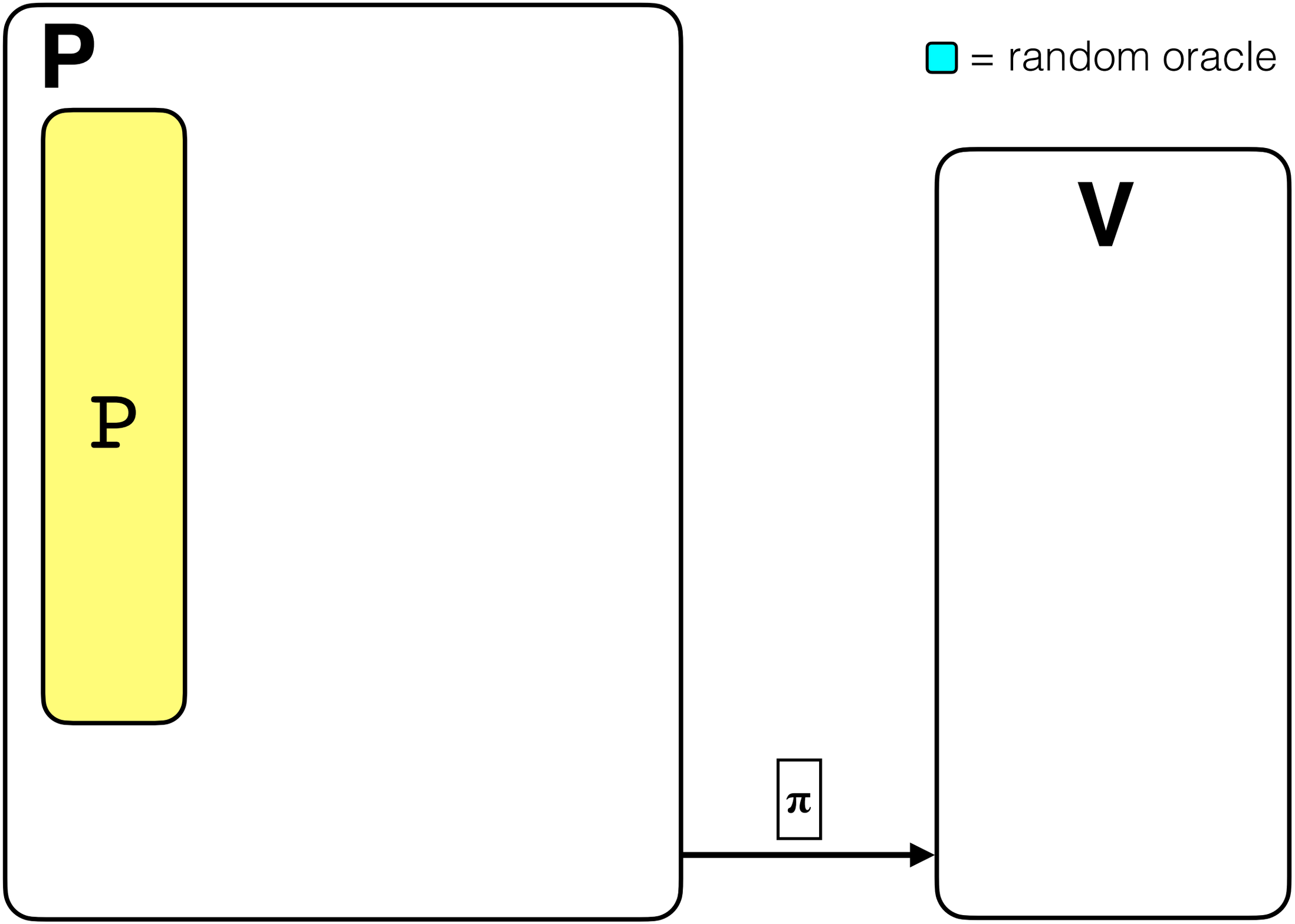
[BCS16]

Idea: generalize Kilian to IOPs + apply Fiat-Shamir



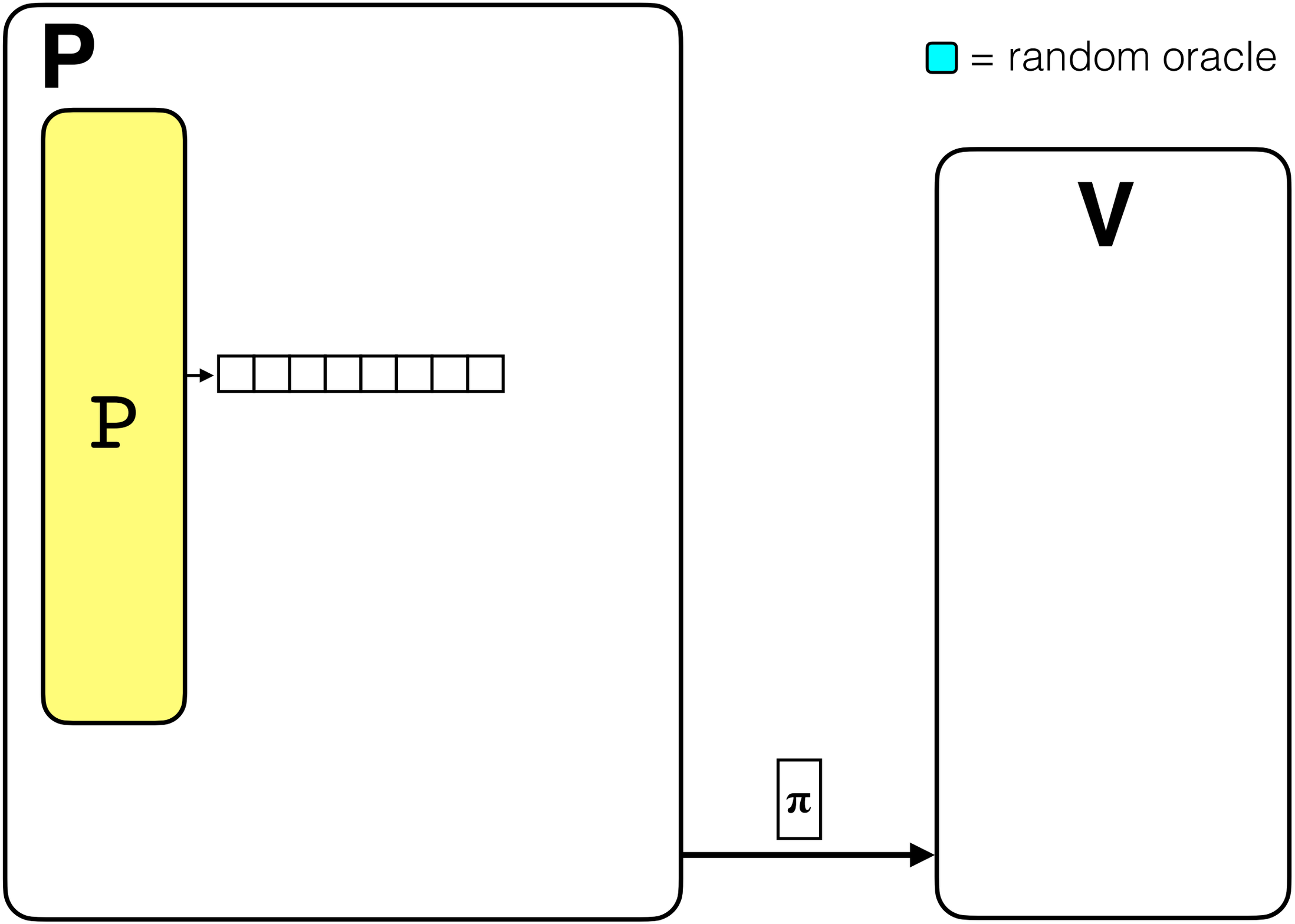
[BCS16]

Idea: generalize Kilian to IOPs + apply Fiat-Shamir



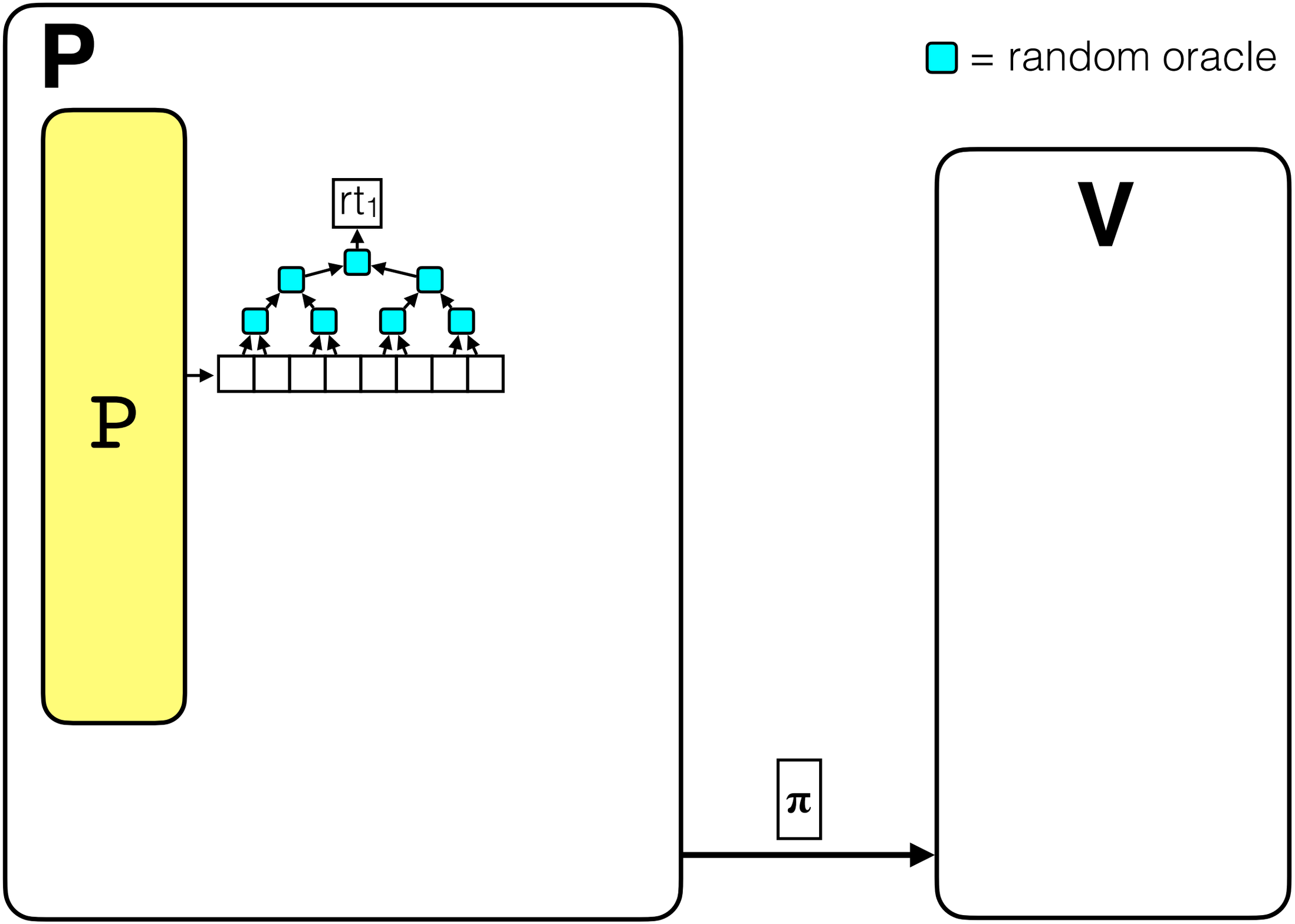
[BCS16]

Idea: generalize Kilian to IOPs + apply Fiat-Shamir



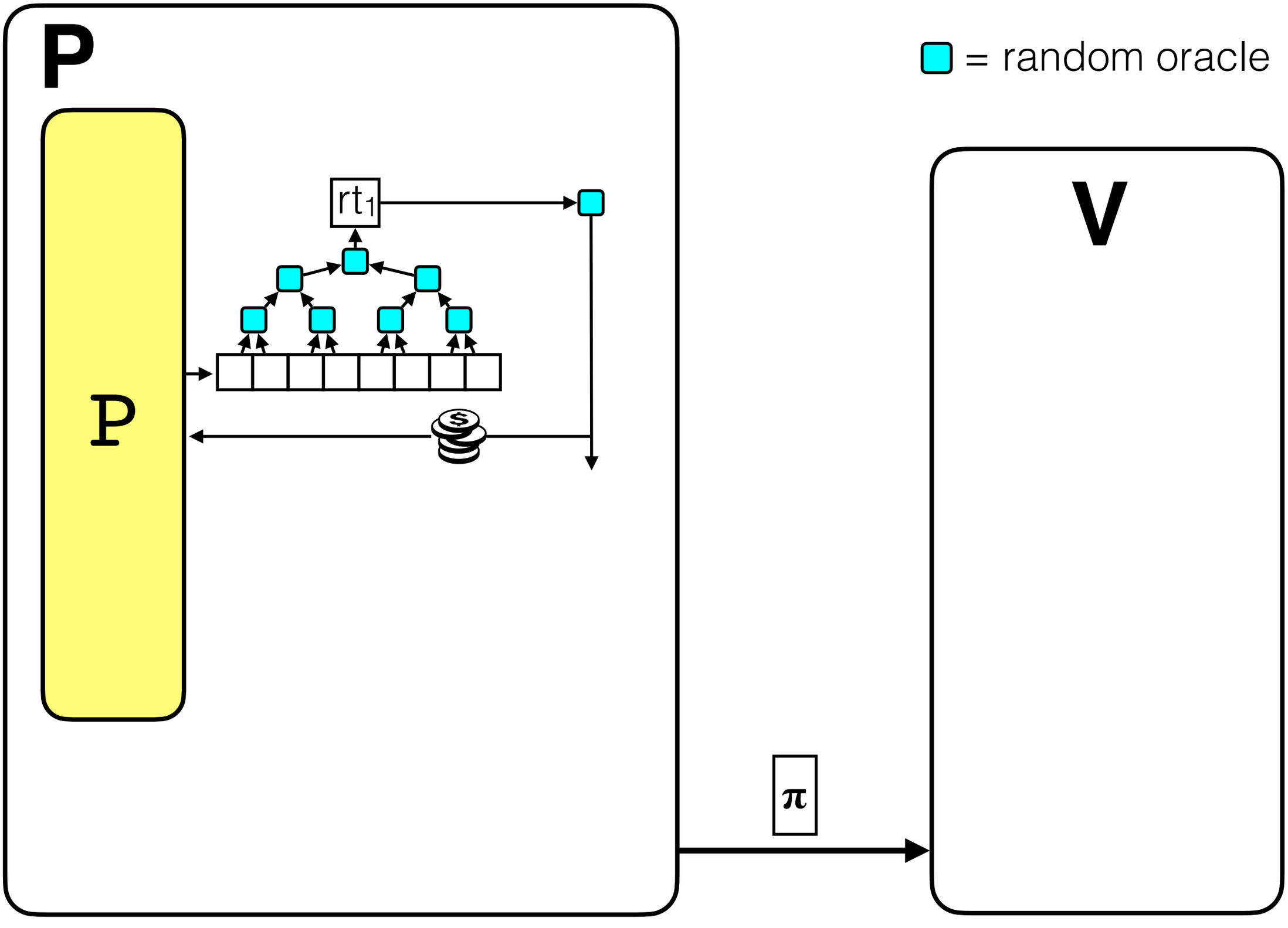
[BCS16]

Idea: generalize Kilian to IOPs + apply Fiat-Shamir



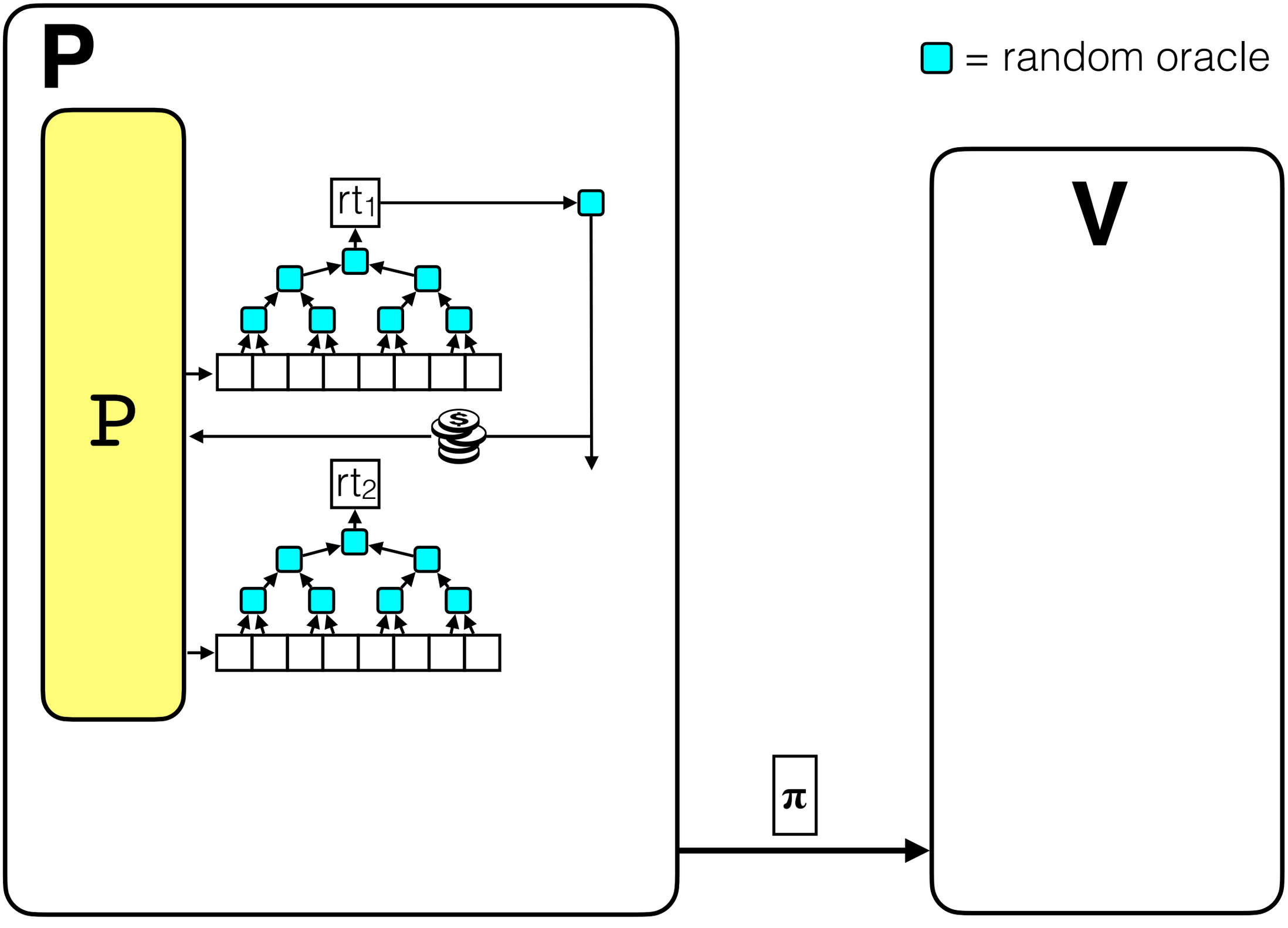
[BCS16]

Idea: generalize Kilian to IOPs + apply Fiat-Shamir



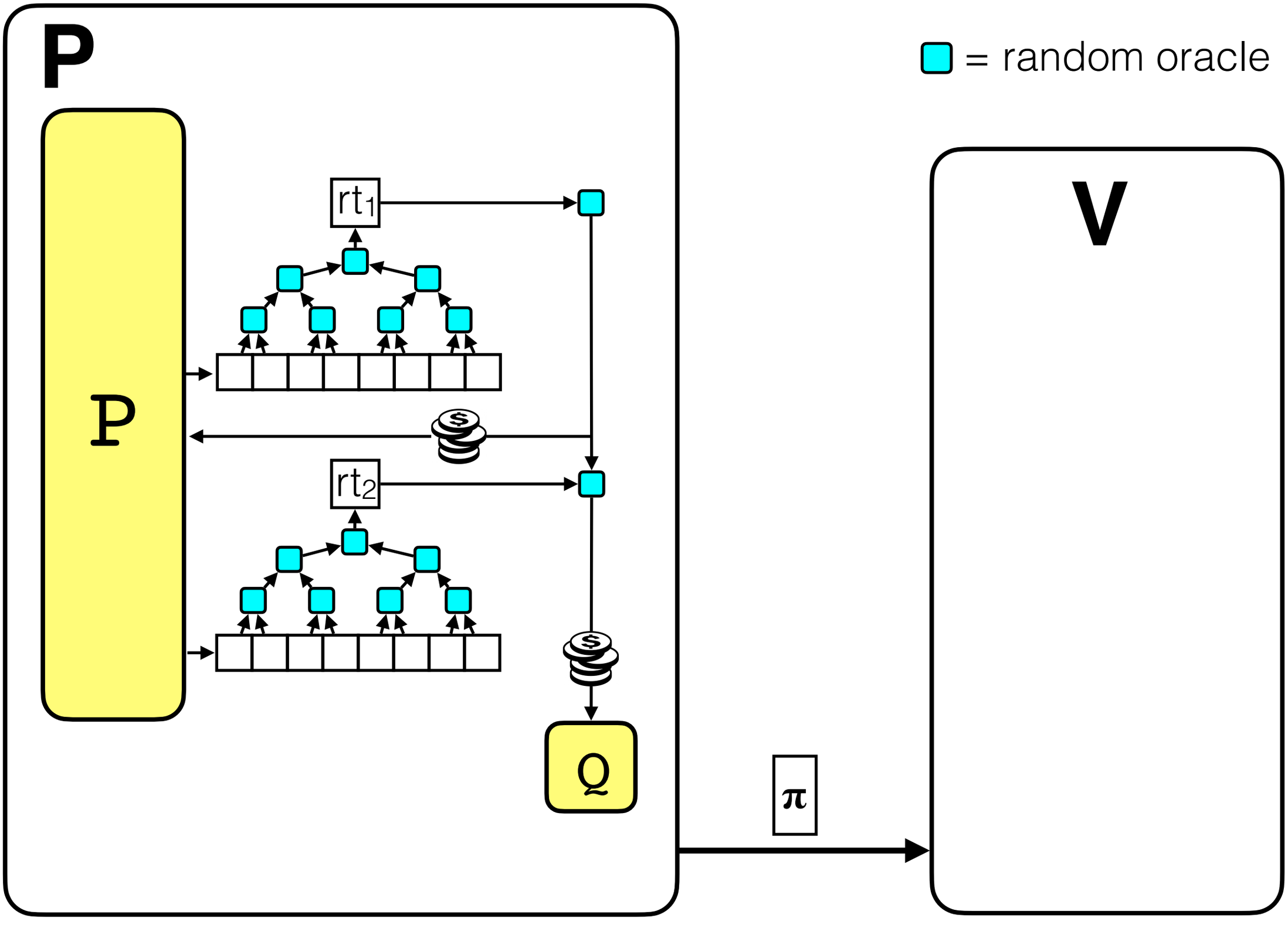
[BCS16]

Idea: generalize Kilian to IOPs + apply Fiat-Shamir



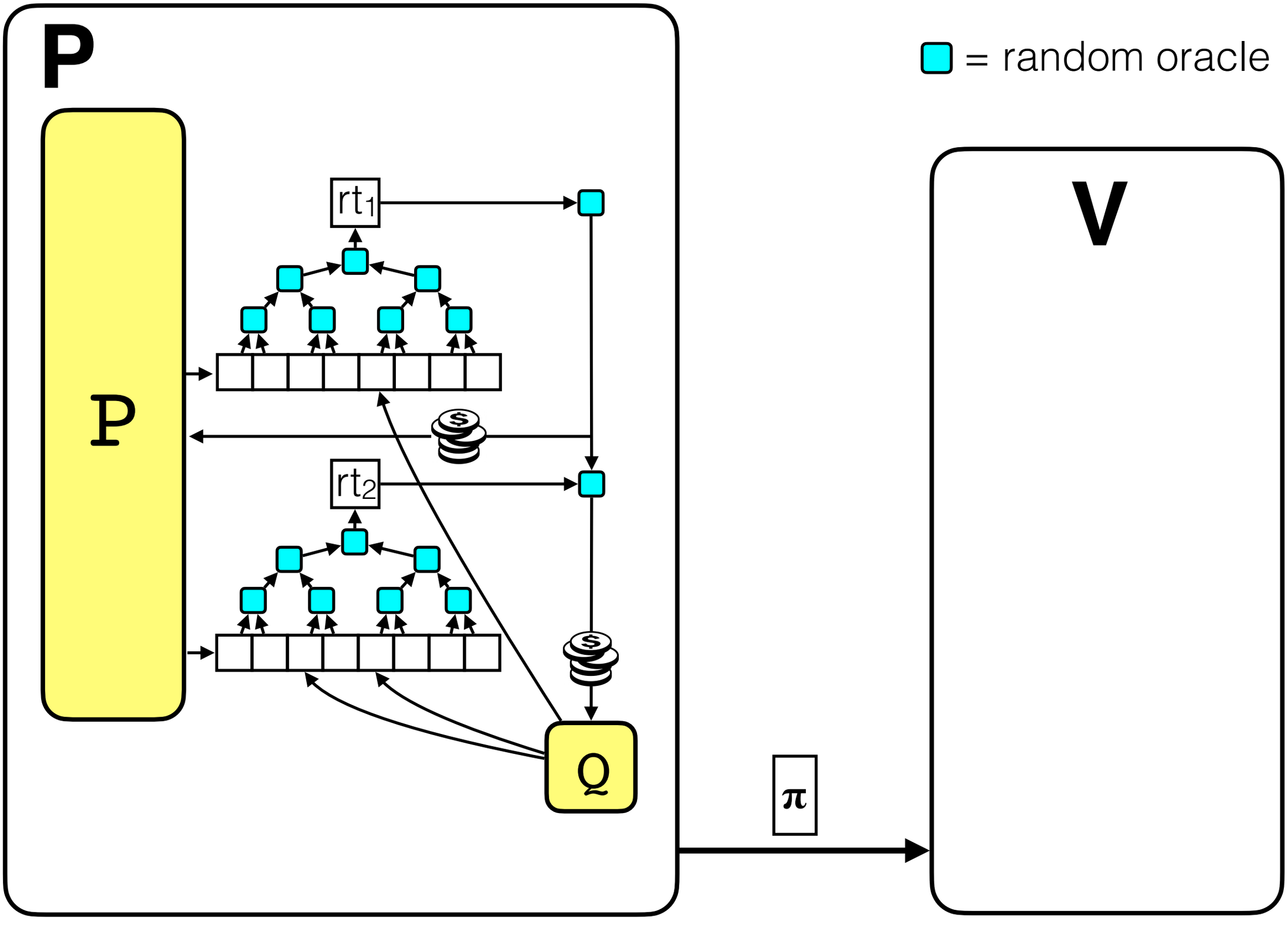
[BCS16]

Idea: generalize Kilian to IOPs + apply Fiat-Shamir



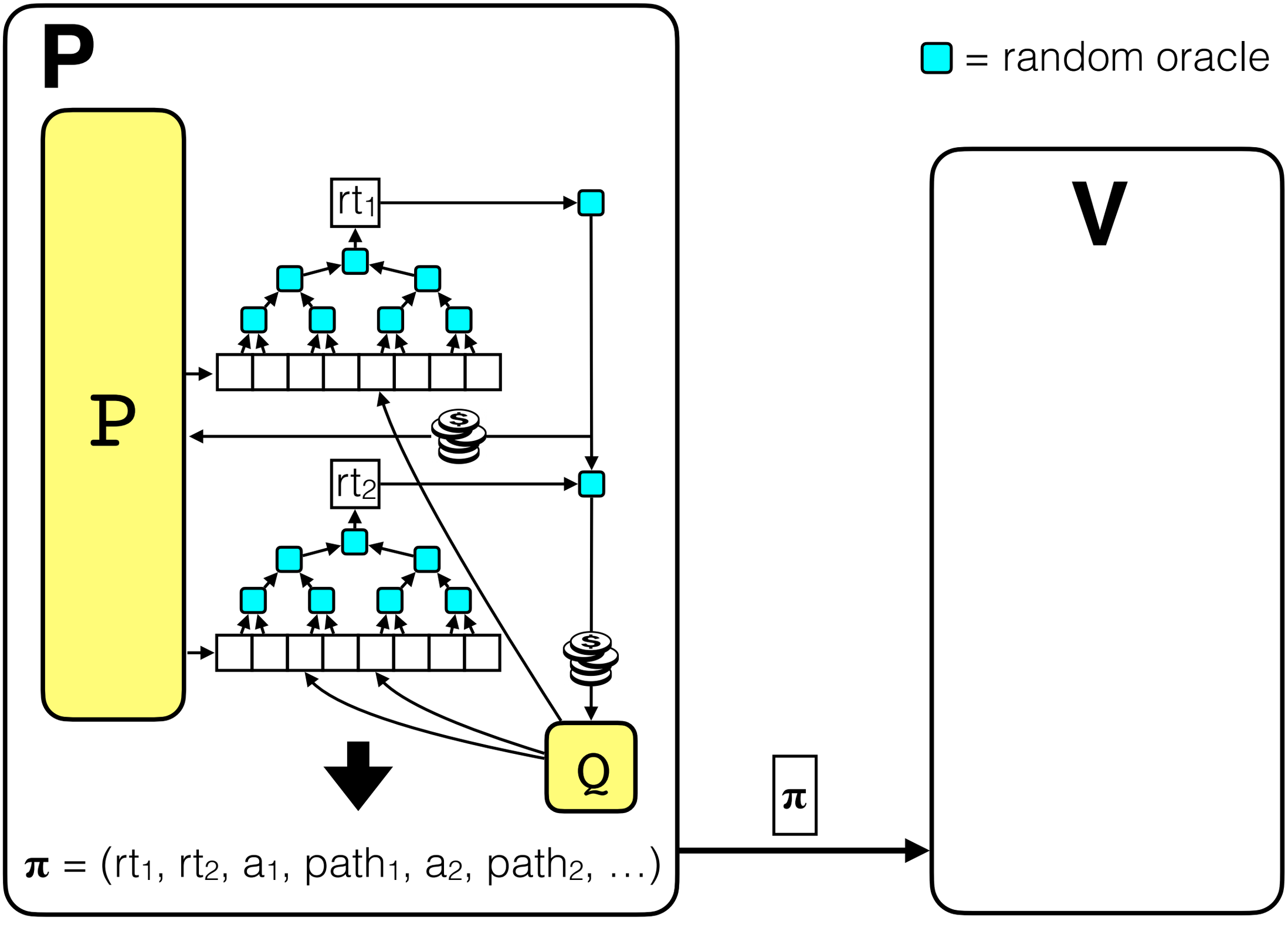
[BCS16]

Idea: generalize Kilian to IOPs + apply Fiat-Shamir



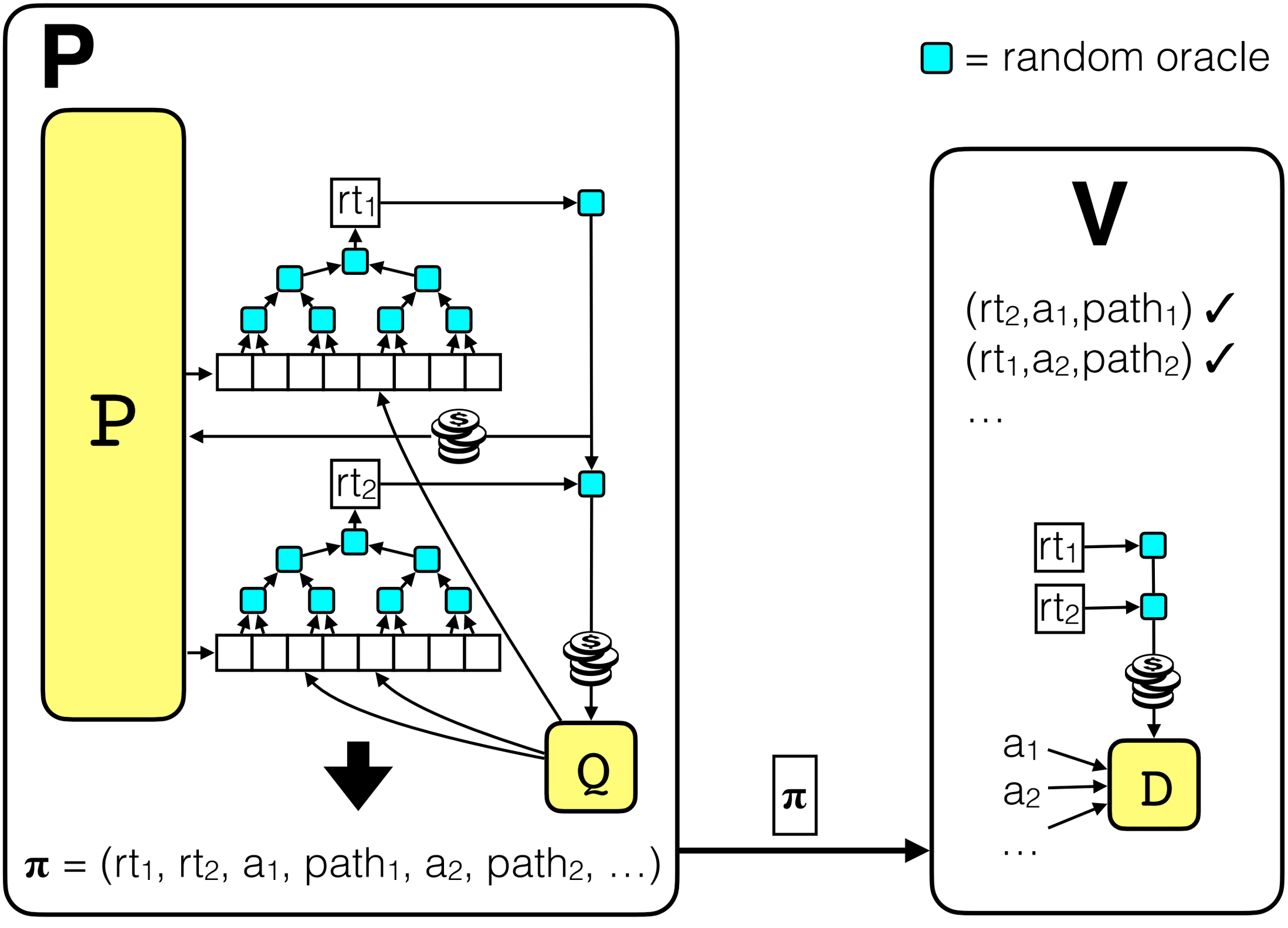
[BCS16]

Idea: generalize Kilian to IOPs + apply Fiat-Shamir



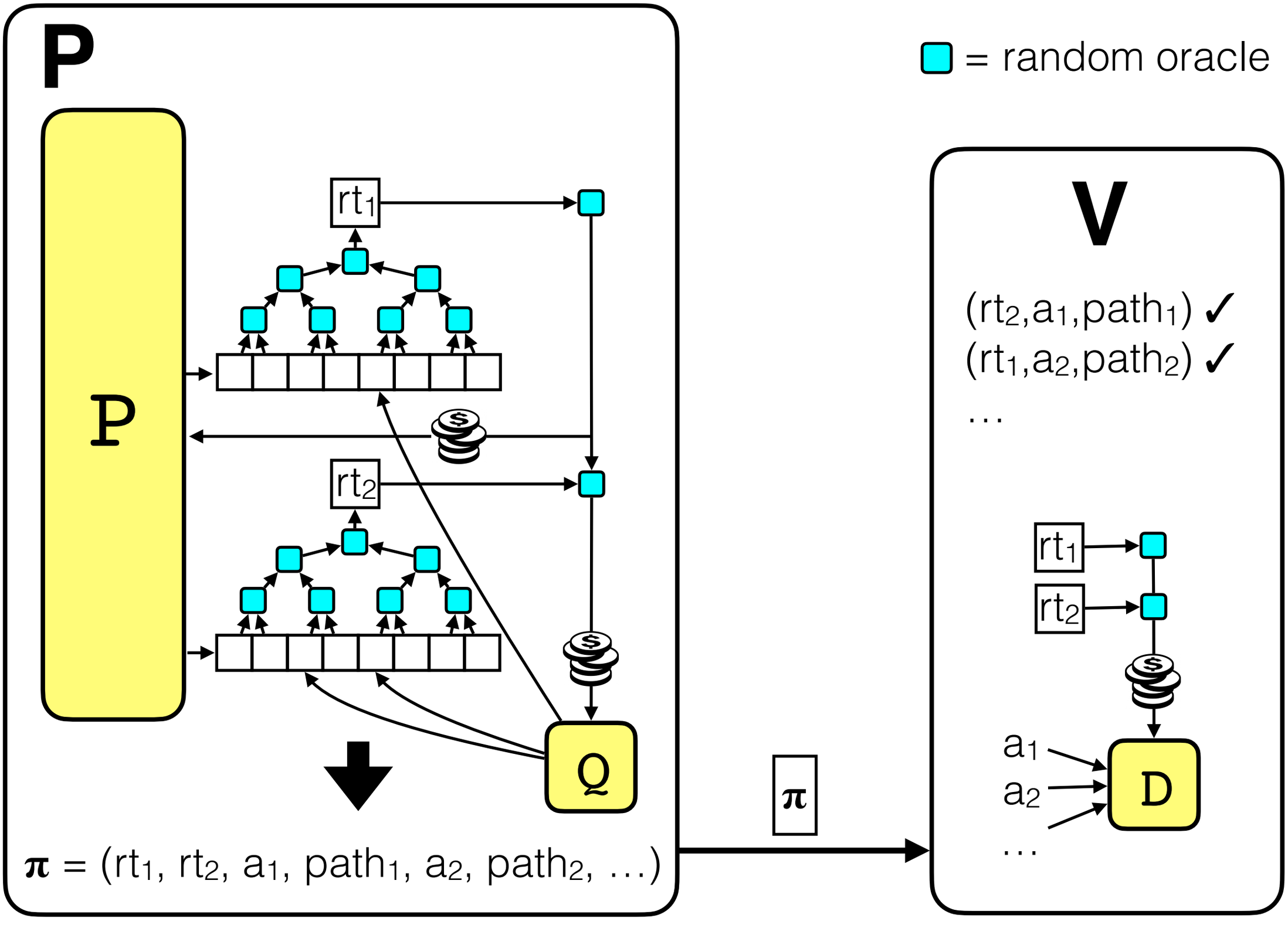
[BCS16]

Idea: generalize Kilian to IOPs + apply Fiat-Shamir



[BCS16]

Idea: generalize Kilian to IOPs + apply Fiat-Shamir



Interaction ➡️ more efficient *non-interactive* arguments

Theorem [BCS16]

IOP-based construction in previous slide is a SNARG
that is unconditionally secure in the random oracle model.

Theorem [BCS16]

IOP-based construction in previous slide is a SNARG
that is unconditionally secure in the random oracle model.

The SNARG costs are analogous to before:

Theorem [BCS16]

IOP-based construction in previous slide is a SNARG
that is unconditionally secure in the random oracle model.

The SNARG costs are analogous to before:

IOP parameters:
- ℓ is proof length (**sum** of each round's length)
- Σ is proof alphabet
- q is number of queries (**sum** of each round's queries)

Theorem [BCS16]

IOP-based construction in previous slide is a SNARG
that is unconditionally secure in the random oracle model.

The SNARG costs are analogous to before:

$$\begin{aligned} \text{proving time} &= \text{IOP-prove} + \boxed{t_{\mathsf{RO}} \times \ell} \\ \text{verification time} &= \text{IOP-verify} + \boxed{t_{\mathsf{RO}} \times q \times \log \ell} \\ \text{argument size} &= q \times (\log|\Sigma| + 2\kappa \log \ell) = \text{poly}(\kappa) \end{aligned}$$

— small overheads

IOP parameters:
- ℓ is proof length (**sum** of each round's length)
- Σ is proof alphabet
- q is number of queries (**sum** of each round's queries)

Theorem [BCS16]

IOP-based construction in previous slide is a SNARG
that is unconditionally secure in the random oracle model.

The SNARG costs are analogous to before:

$$\begin{aligned} \text{proving time} &= \text{IOP-prove} + \boxed{t_{\mathsf{RO}} \times \ell} \\ \text{verification time} &= \text{IOP-verify} + \boxed{t_{\mathsf{RO}} \times q \times \log \ell} \\ \text{argument size} &= q \times (\log|\Sigma| + 2\kappa \log \ell) = \text{poly}(\kappa) \end{aligned}$$

— small overheads

IOP parameters:
- ℓ is proof length (**sum** of each round's length)
- Σ is proof alphabet
- q is number of queries (**sum** of each round's queries)

Punchline: ROM + **IOP** → SNARGs.

Theorem [BCS16]

IOP-based construction in previous slide is a SNARG
that is unconditionally secure in the random oracle model.

The SNARG costs are analogous to before:

$$\begin{aligned} \text{proving time} &= \text{IOP-prove} + \boxed{t_{\mathsf{RO}} \times \ell} \\ \text{verification time} &= \text{IOP-verify} + \boxed{t_{\mathsf{RO}} \times q \times \log \ell} \\ \text{argument size} &= q \times (\log|\Sigma| + 2\kappa \log \ell) = \text{poly}(\kappa) \end{aligned}$$

— small overheads

IOP parameters:
- ℓ is proof length (**sum** of each round's length)
- Σ is proof alphabet
- q is number of queries (**sum** of each round's queries)

Punchline: ROM + **IOP** → SNARGs.

Lemma: The transformation preserves ZK and PoK:
if IOP is ZK and PoK then you get a zkSNARK.

What about **soundness**?

What about **soundness**?

Theorem [BCS16]:
SNARG soundness = IOP **state restoration soundness**

What about **soundness**?

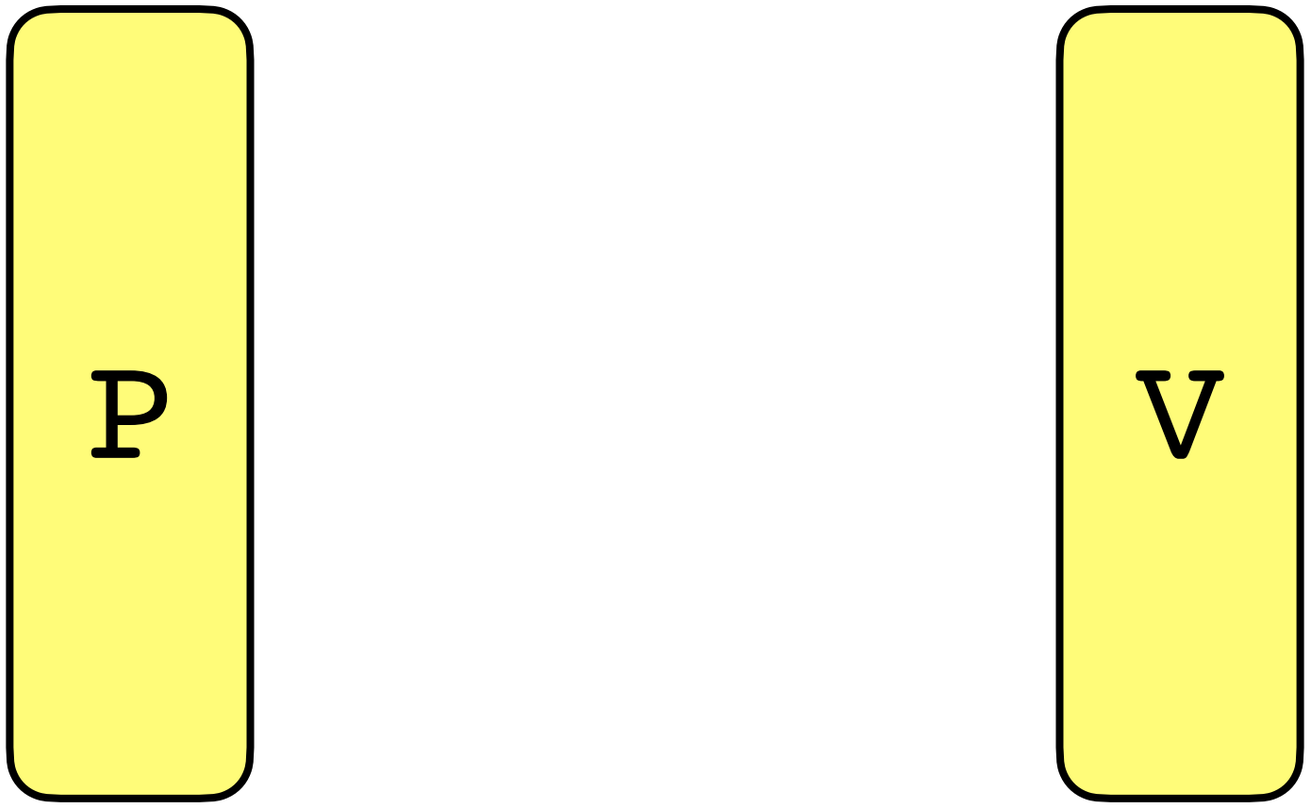
also applies to Fiat-Shamir for IPs

Theorem [BCS16]:
SNARG soundness = IOP **state restoration soundness**

What about **soundness**?

also applies to Fiat-Shamir for IPs

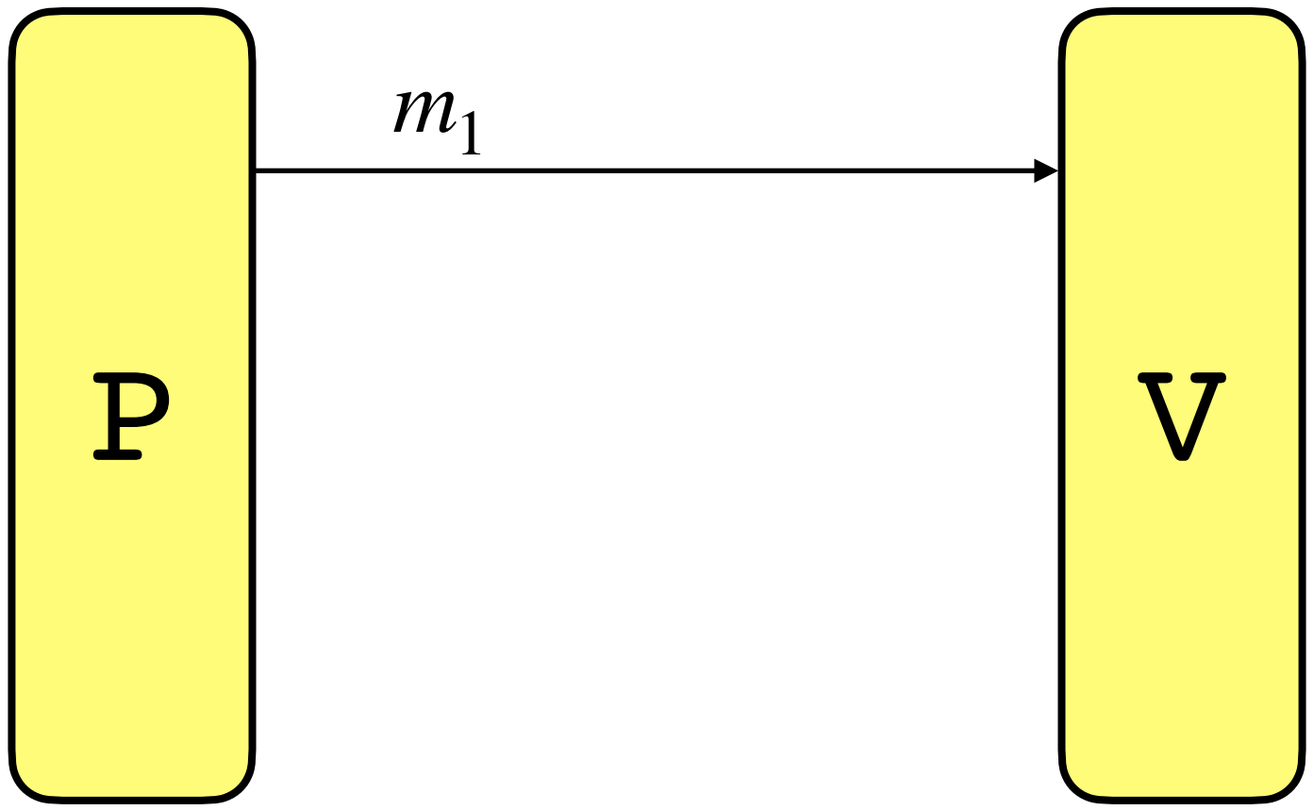
Theorem [BCS16]:
SNARG soundness = IOP **state restoration soundness**



What about **soundness**?

also applies to Fiat-Shamir for IPs

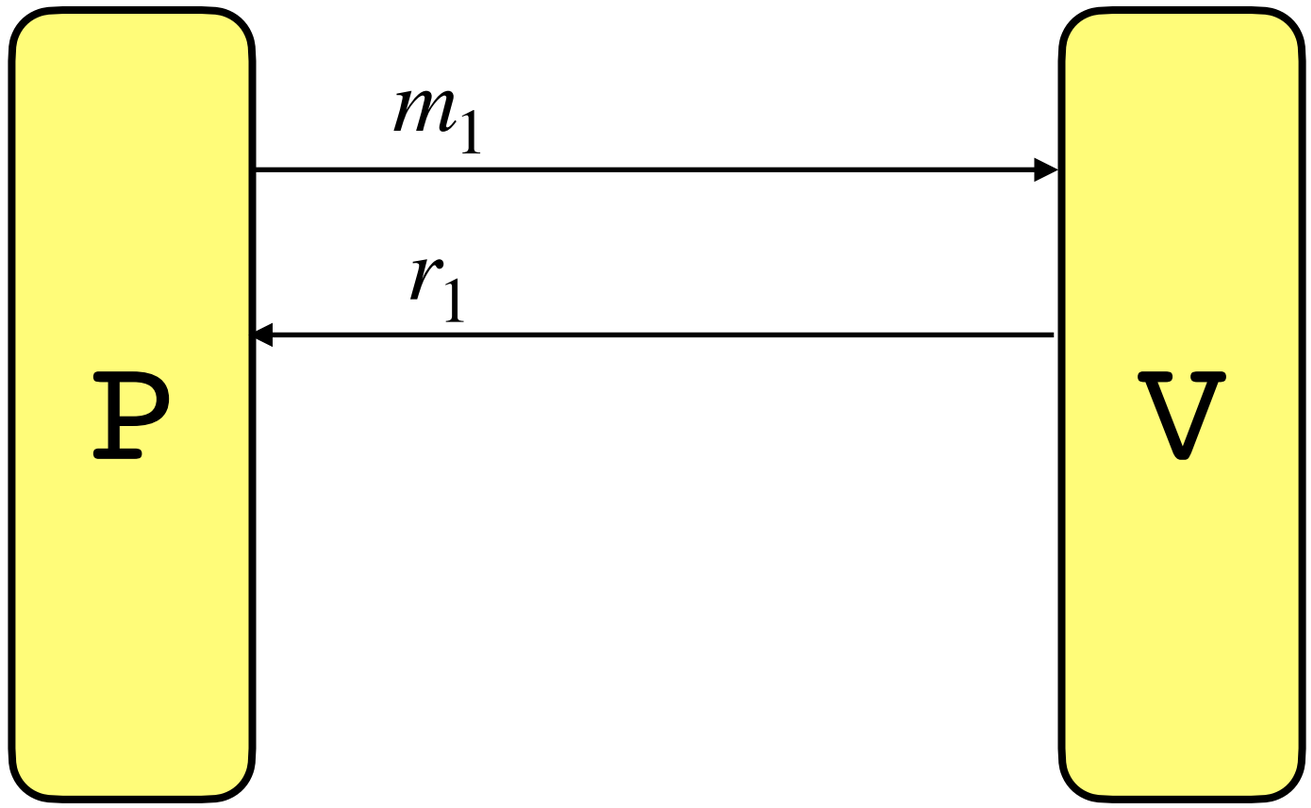
Theorem [BCS16]:
SNARG soundness = IOP **state restoration soundness**



What about **soundness**?

also applies to Fiat-Shamir for IPs

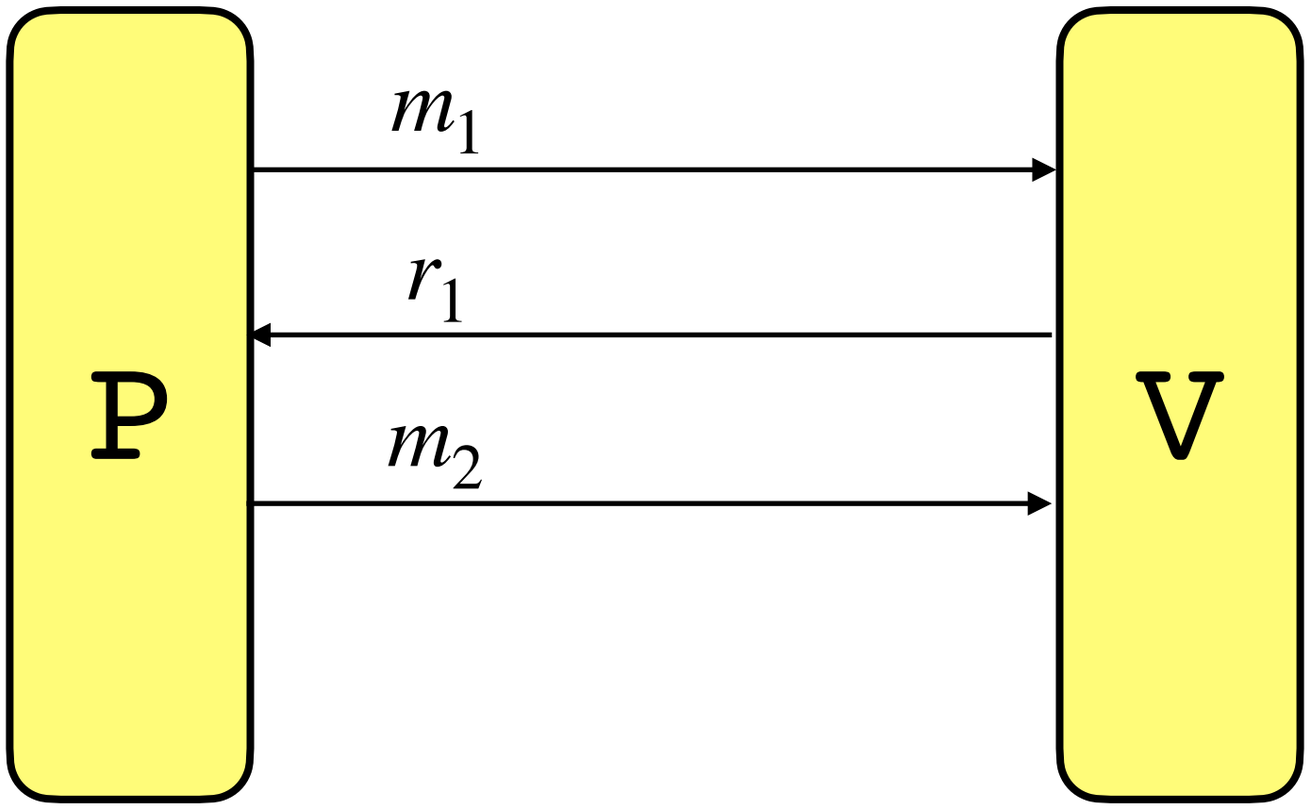
Theorem [BCS16]:
SNARG soundness = IOP **state restoration soundness**



What about **soundness**?

also applies to Fiat-Shamir for IPs

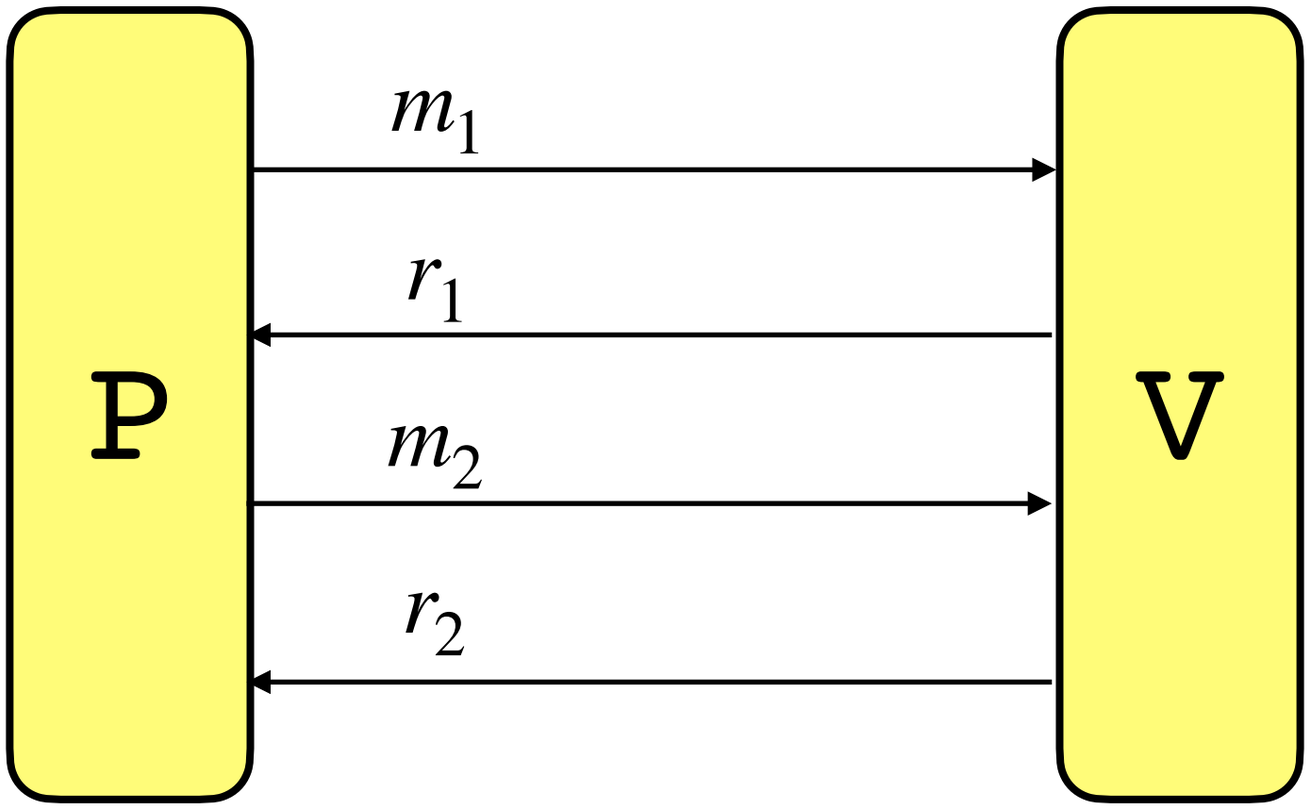
Theorem [BCS16]:
SNARG soundness = IOP **state restoration soundness**



What about **soundness**?

also applies to Fiat-Shamir for IPs

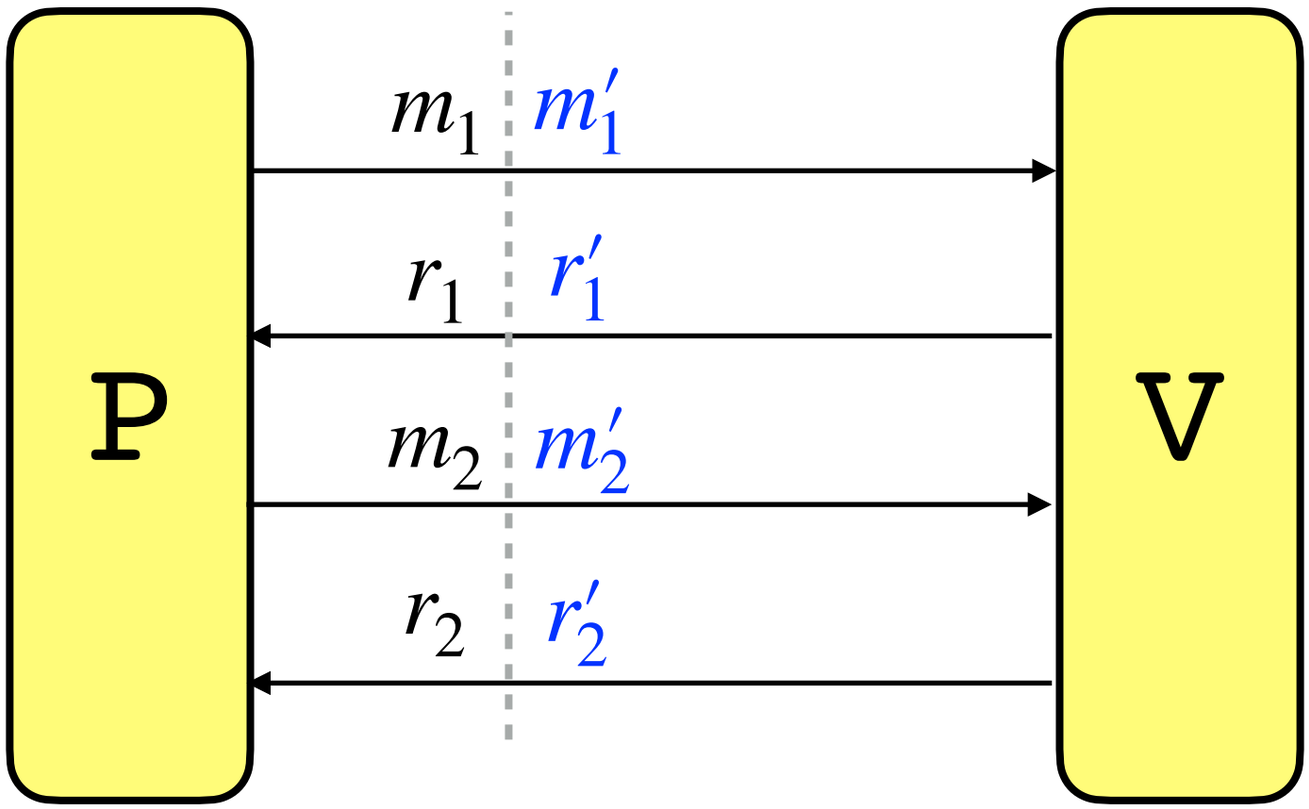
Theorem [BCS16]:
SNARG soundness = IOP **state restoration soundness**



What about **soundness**?

also applies to Fiat-Shamir for IPs

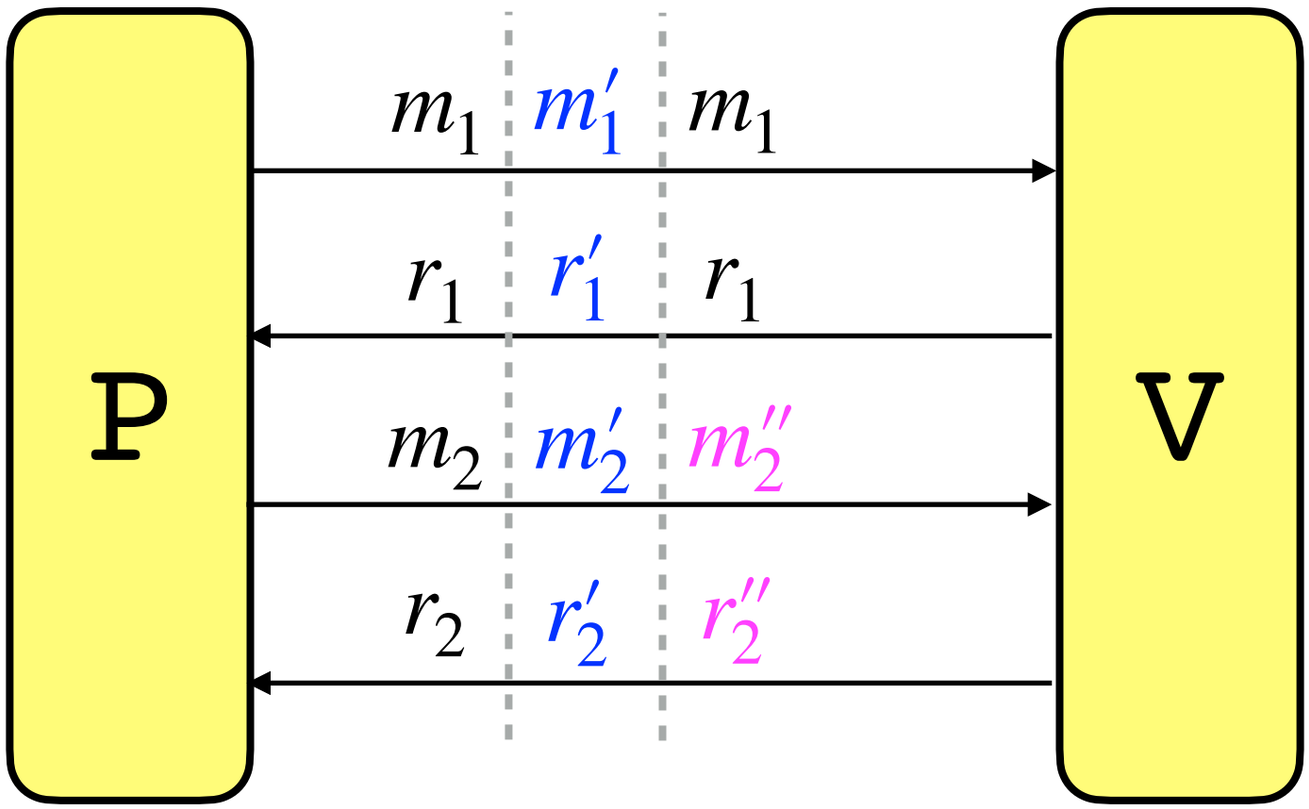
Theorem [BCS16]:
SNARG soundness = IOP **state restoration soundness**



What about **soundness**?

also applies to Fiat-Shamir for IPs

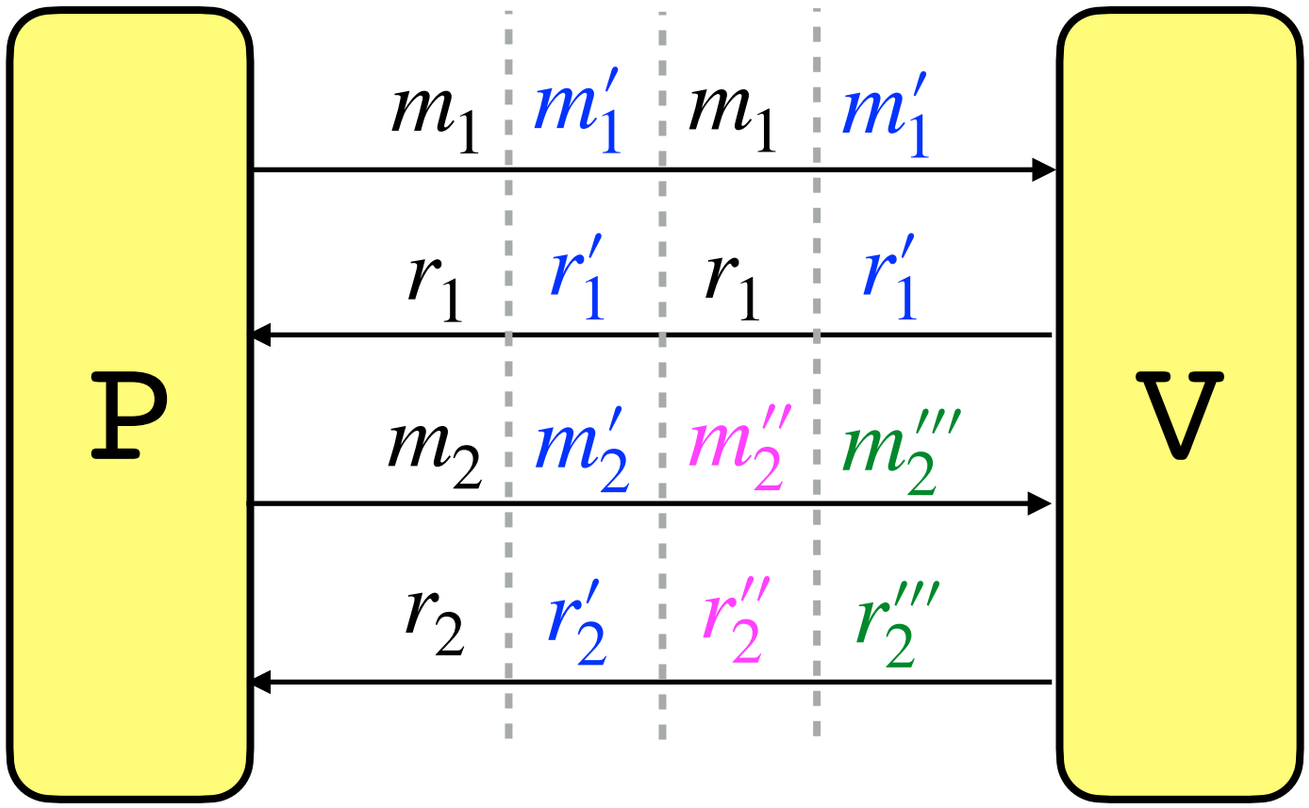
Theorem [BCS16]:
SNARG soundness = IOP **state restoration soundness**



What about **soundness**?

also applies to Fiat-Shamir for IPs

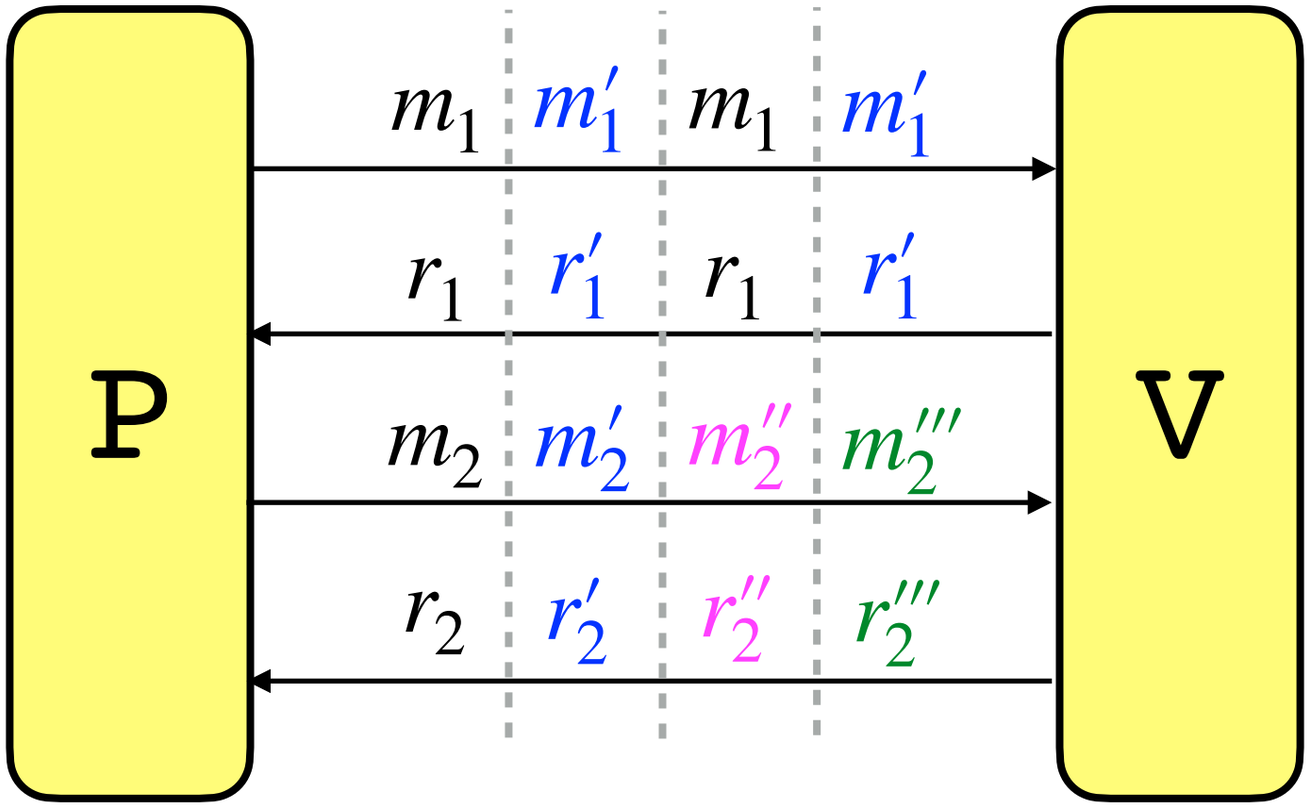
Theorem [BCS16]:
SNARG soundness = IOP **state restoration soundness**



What about **soundness**?

also applies to Fiat-Shamir for IPs

Theorem [BCS16]:
SNARG soundness = IOP **state restoration soundness**

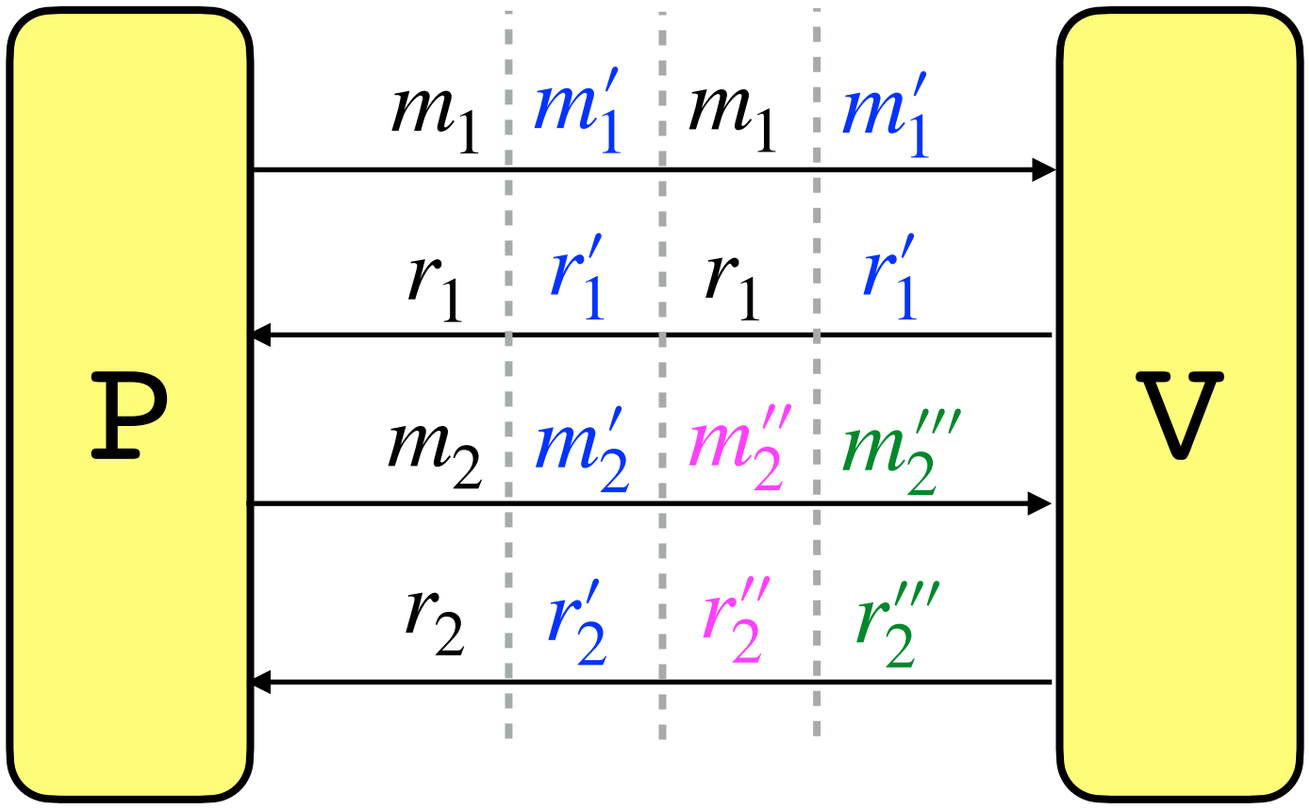


Cheating prover wins by finding **any** accepting transcript

What about **soundness**?

also applies to Fiat-Shamir for IPs

Theorem [BCS16]:
SNARG soundness = IOP **state restoration soundness**



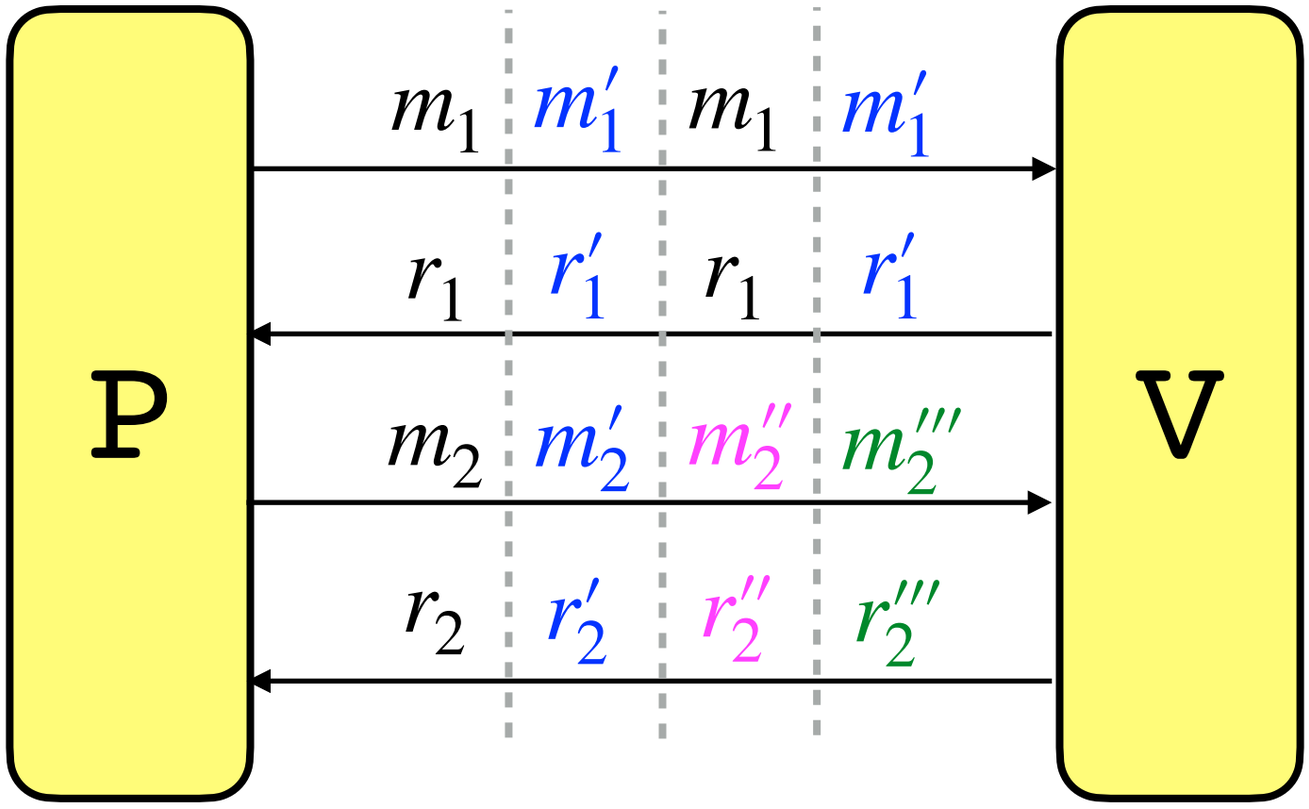
Cheating prover wins by finding **any** accepting transcript

Lemma [BCS16]: for any k-round protocol, $s_{SR} \leq s \cdot T^k$
and this is tight for some protocols

What about **soundness**?

also applies to Fiat-Shamir for IPs

Theorem [BCS16]:
SNARG soundness = IOP **state restoration soundness**



Cheating prover wins by finding **any** accepting transcript

Lemma [BCS16]: for any k-round protocol, $s_{SR} \leq s \cdot T^k$
and this is tight for some protocols

Thankfully, widely-used protocols have good SR soundness!
(**but** it is still important to prove it [AFK21])

IOP-Based SNARGs

GOOD

BAD

IOP-Based SNARGs

GOOD

BAD

- random oracles don't exist*

as before

IOP-Based SNARGs

GOOD

- random oracle is “robust”
- black-box use of lightweight crypto (any hash function)

SNARG-prove ≈ PCP-prove
SNARG-verify ≈ PCP-verify

- transparent (public-coin) setup

system parameters
=
choice of hash function

- post-quantum

as before

BAD

- random oracles don’t exist*

as before

IOP-Based SNARGs

GOOD

- random oracle is “robust”
- black-box use of lightweight crypto (any hash function)

SNARG-prove ≈ PCP-prove
SNARG-verify ≈ PCP-verify

- transparent (public-coin) setup

system parameters
=
choice of hash function

- post-quantum

as before

BAD

- random oracles don’t exist*

as before

- ~~**PCPs are expensive!**~~

IOP-Based SNARGs

GOOD

- random oracle is “robust”
- black-box use of lightweight crypto (any hash function)

SNARG-prove ≈ PCP-prove
SNARG-verify ≈ PCP-verify

- transparent (public-coin) setup

system parameters
=
choice of hash function

- post-quantum

as before

BAD

- random oracles don’t exist*

as before

- ~~**PCPs are expensive!**~~

recall: today IOPs are much more efficient than PCPs!

IOP-based SNARGs

IOP-based SNARGs

asymptotic
improvements
in IOP protocols

IOP-based SNARGs

asymptotic improvements in IOP protocols

concrete improvements in IOP protocols

IOP-based SNARGs

asymptotic improvements in IOP protocols ⟶ concrete improvements in IOP protocols

⟵

IOP-based SNARGs

IOP-based SNARGs

asymptotic improvements in IOP protocols ⟷ concrete improvements in IOP protocols → improved IOP-based SNARGs

	SNARG size (representative)	main tool to get there

IOP-based SNARGs

asymptotic improvements in IOP protocols ⟵ ⟶ concrete improvements in IOP protocols ⟶ improved IOP-based SNARGs

	SNARG size (representative)	main tool to get there
[BBCGGHPRSTV 17] **SCI**	10s of MB	interactive proof composition [BCGRS16] applied to the BS PCPP for RS

IOP-based SNARGs

asymptotic improvements in IOP protocols ⟵ ⟶ concrete improvements in IOP protocols ⟶ improved IOP-based SNARGs

	SNARG size (representative)	main tool to get there
[BBCGGHPRSTV 17] **SCI**	10s of MB	interactive proof composition [BCGRS16] applied to the BS PCPP for RS
[AHIV 17] **Ligero**	1s of MB	IOP for testing Interleave(RS)

IOP-based SNARGs

asymptotic improvements in IOP protocols ⟵ ⟶ concrete improvements in IOP protocols ⟶ improved IOP-based SNARGs

	SNARG size (representative)	main tool to get there
[BBCGGHPRSTV 17] **SCI**	10s of MB	interactive proof composition [BCGRS16] applied to the BS PCPP for RS
[AHIV 17] **Ligero**	1s of MB	IOP for testing Interleave(RS)
[BBHR 18b] **Stark**	100 kB	IOPP for RS with linear length and logarithmic query complexity [BBHR18a]

IOP-based SNARGs

asymptotic improvements in IOP protocols ⟵ ⟶ concrete improvements in IOP protocols ⟶ improved IOP-based SNARGs

	SNARG size (representative)	main tool to get there
[BBCGGHPRSTV 17] **SCI**	10s of MB	interactive proof composition [BCGRS16] applied to the BS PCPP for RS
[AHIV 17] **Ligero**	1s of MB	IOP for testing Interleave(RS)
[BBHR 18b] **Stark**	100 kB	IOPP for RS with linear length and logarithmic query complexity [BBHR18a]
[BCRSVW 18] **Aurora**	100 kB	sumcheck protocol for univariate polynomials

IOP-based SNARGs

asymptotic improvements in IOP protocols ⟷ concrete improvements in IOP protocols → improved IOP-based SNARGs

	SNARG size (representative)	main tool to get there
[BBCGGHPRSTV 17] **SCI**	10s of MB	interactive proof composition [BCGRS16] applied to the BS PCPP for RS
[AHIV 17] **Ligero**	1s of MB	IOP for testing Interleave(RS)
[BBHR 18b] **Stark**	100 kB	IOPP for RS with linear length and logarithmic query complexity [BBHR18a]
[BCRSVW 18] **Aurora**	100 kB	sumcheck protocol for univariate polynomials
FRONTIER	**10 kB? 1 kB?**	**???**

Limits of Argument Size

Limits of Argument Size

For soundness $2^{-\kappa}$ using Micali/BCS:

	ℓ	$\lvert\Sigma\rvert$	q	argument size $q \times (\log\lvert\Sigma\rvert + 2\kappa \log\ell)$

Limits of Argument Size

For soundness $2^{-\kappa}$ using Micali/BCS:

	ℓ	$\lvert\Sigma\rvert$	q	argument size $q \times (\log\lvert\Sigma\rvert + 2\kappa \log\ell)$
best used in practice	O(n)	O(n)	$O(\kappa \log n)$	$O(\kappa^2 \log^2 n)$

Limits of Argument Size

For soundness $2^{-\kappa}$ using Micali/BCS:

	ℓ	$\lvert\Sigma\rvert$	q	argument size $q \times (\log\lvert\Sigma\rvert + 2\kappa \log\ell)$
best used in practice	O(n)	O(n)	$O(\kappa \log n)$	$O(\kappa^2 \log^2 n)$
best known asymptotics	O(n)	O(n)	$O(\kappa)$	$O(\kappa^2 \log n)$

Limits of Argument Size

For soundness $2^{-\kappa}$ using Micali/BCS:

	ℓ	$\lvert\Sigma\rvert$	q	argument size $q \times (\log\lvert\Sigma\rvert + 2\kappa \log\ell)$
best used in practice	O(n)	O(n)	$O(\kappa \log n)$	$O(\kappa^2 \log^2 n)$
best known asymptotics	O(n)	O(n)	$O(\kappa)$	$O(\kappa^2 \log n)$
open: "Sliding Scale Conjecture" for IOPs?	O(n)	poly(n)	$O(\kappa/\log n)$	$O(\kappa^2)$

Limits of Argument Size

For soundness $2^{-\kappa}$ using Micali/BCS:

	ℓ	$\lvert\Sigma\rvert$	q	argument size $q \times (\log\lvert\Sigma\rvert + 2\kappa \log\ell)$
best used in practice	O(n)	O(n)	$O(\kappa \log n)$	$O(\kappa^2 \log^2 n)$
best known asymptotics	O(n)	O(n)	$O(\kappa)$	$O(\kappa^2 \log n)$
open: "Sliding Scale Conjecture" for IOPs?	O(n)	poly(n)	$O(\kappa/\log n)$	$O(\kappa^2)$

Exciting direction that
can get argument size < 1 kB!

Limits of Argument Size

For soundness $2^{-\kappa}$ using Micali/BCS:

	ℓ	$\lvert\Sigma\rvert$	q	argument size $q \times (\log\lvert\Sigma\rvert + 2\kappa \log\ell)$
best used in practice	O(n)	O(n)	$O(\kappa \log n)$	$O(\kappa^2 \log^2 n)$
best known asymptotics	O(n)	O(n)	$O(\kappa)$	$O(\kappa^2 \log n)$
open: "Sliding Scale Conjecture" for IOPs?	O(n)	poly(n)	$O(\kappa/\log n)$	$O(\kappa^2)$

Best possible: need $q \log \ell = \Omega(\kappa)$
so arg size = $\Omega(\kappa^2)$

Exciting direction that
can get argument size < 1 kB!

Limits of Argument Size

For soundness $2^{-\kappa}$ using Micali/BCS:

	ℓ	$\lvert\Sigma\rvert$	q	argument size $q \times (\log\lvert\Sigma\rvert + 2\kappa \log\ell)$
best used in practice	O(n)	O(n)	$O(\kappa \log n)$	$O(\kappa^2 \log^2 n)$
best known asymptotics	O(n)	O(n)	$O(\kappa)$	$O(\kappa^2 \log n)$
open: "Sliding Scale Conjecture" for IOPs?	O(n)	poly(n)	$O(\kappa/\log n)$	$O(\kappa^2)$

Best possible: need $q \log \ell = \Omega(\kappa)$
so arg size = $\Omega(\kappa^2)$

Exciting direction that
can get argument size < 1 kB!

[CY21]: can do a bit better than $O(\kappa^2)$ by optimizing Micali

Post-Quantum Security

Post-Quantum Security

China will open a $10 billion quantum computer center
and others also investing in quantum computing Next Big Future, 2017.10.10

Post-Quantum Security

China will open a $10 billion quantum computer center and others also investing in quantum computing Next Big Future, 2017.10.10

Europe shows first cards in €1-billion quantum bet

Nature, 2018.10.29

Post-Quantum Security

China will open a $10 billion quantum computer center and others also investing in quantum computing Next Big Future, 2017.10.10

Europe shows first cards in €1-billion quantum bet
Nature, 2018.10.29

President Trump has signed a $1.2 billon law to boost US quantum tech MIT Tech Review, 2018.12.22

Post-Quantum Security

China will open a $10 billion quantum computer center and others also investing in quantum computing Next Big Future, 2017.10.10

Europe shows first cards in €1-billion quantum bet
Nature, 2018.10.29

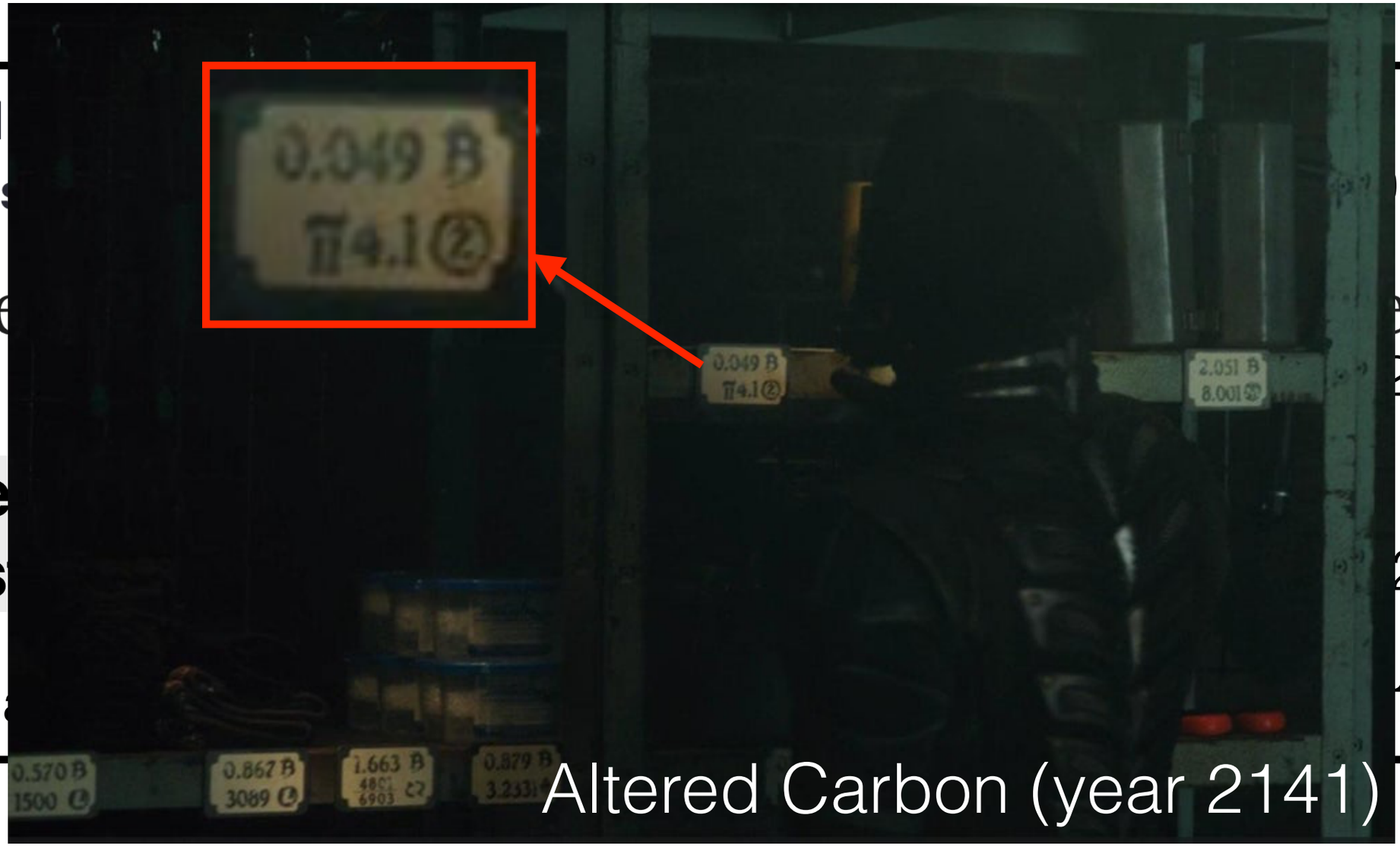
President Trump has signed a $1.2 billon law to boost US quantum tech MIT Tech Review, 2018.12.22

NIST's call for post-quantum cryptography **CLOSED** in 2017.11.30.

Post-Quantum Security

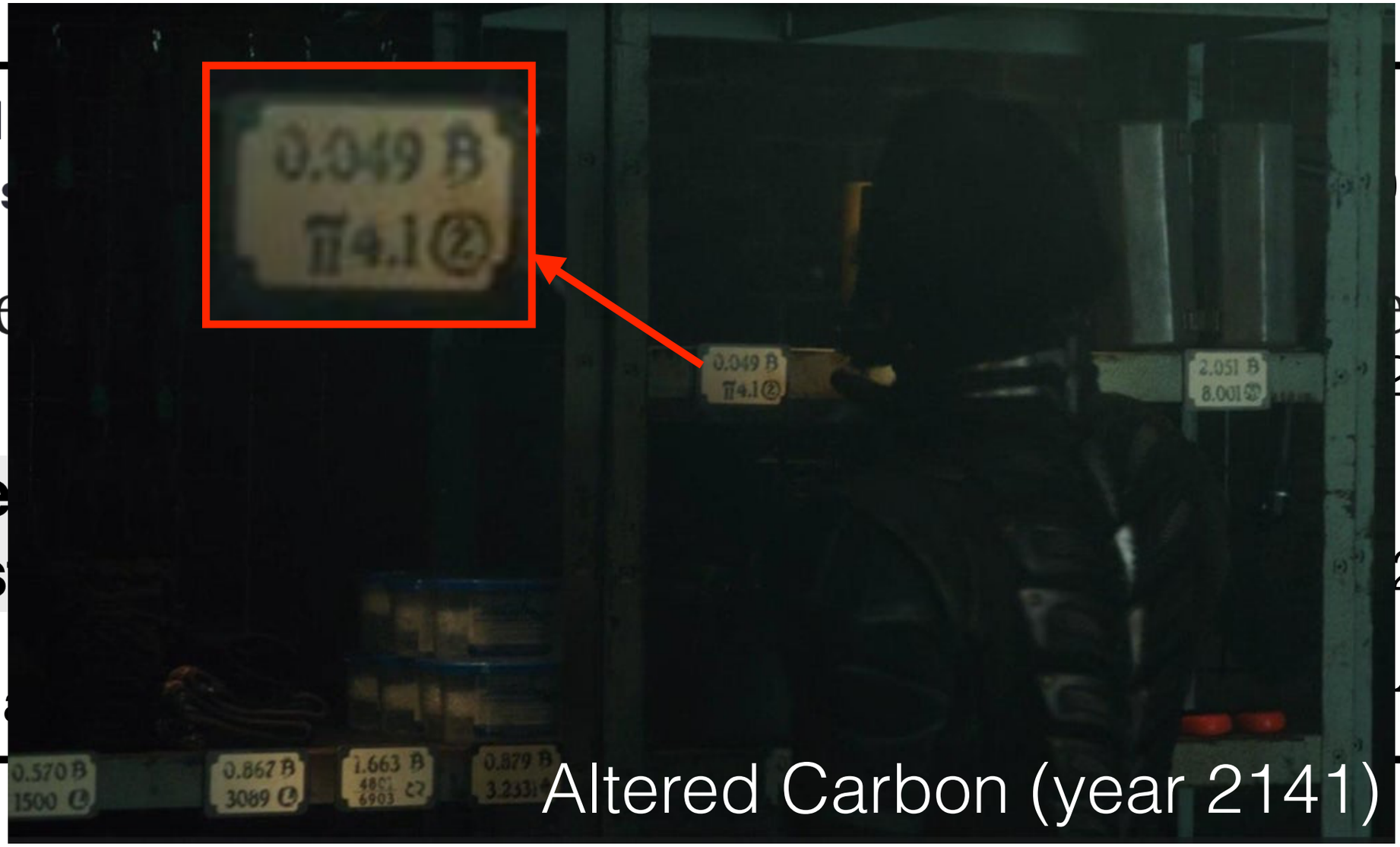
Altered Carbon (year 2141)

Post-Quantum Security



Altered Carbon (year 2141)

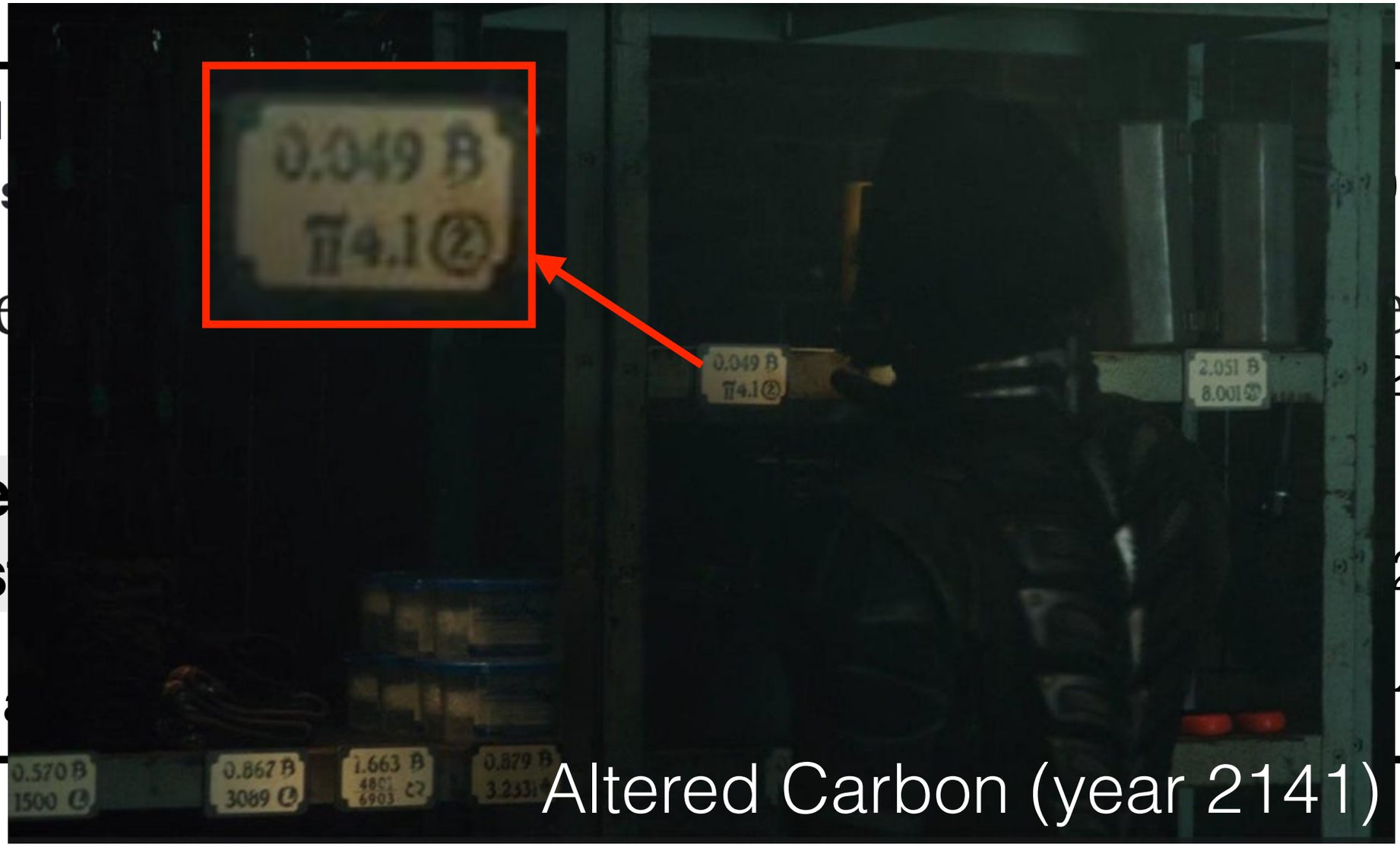
Post-Quantum Security



Altered Carbon (year 2141)

All DL-based constructions are **insecure against quantum adversaries**.

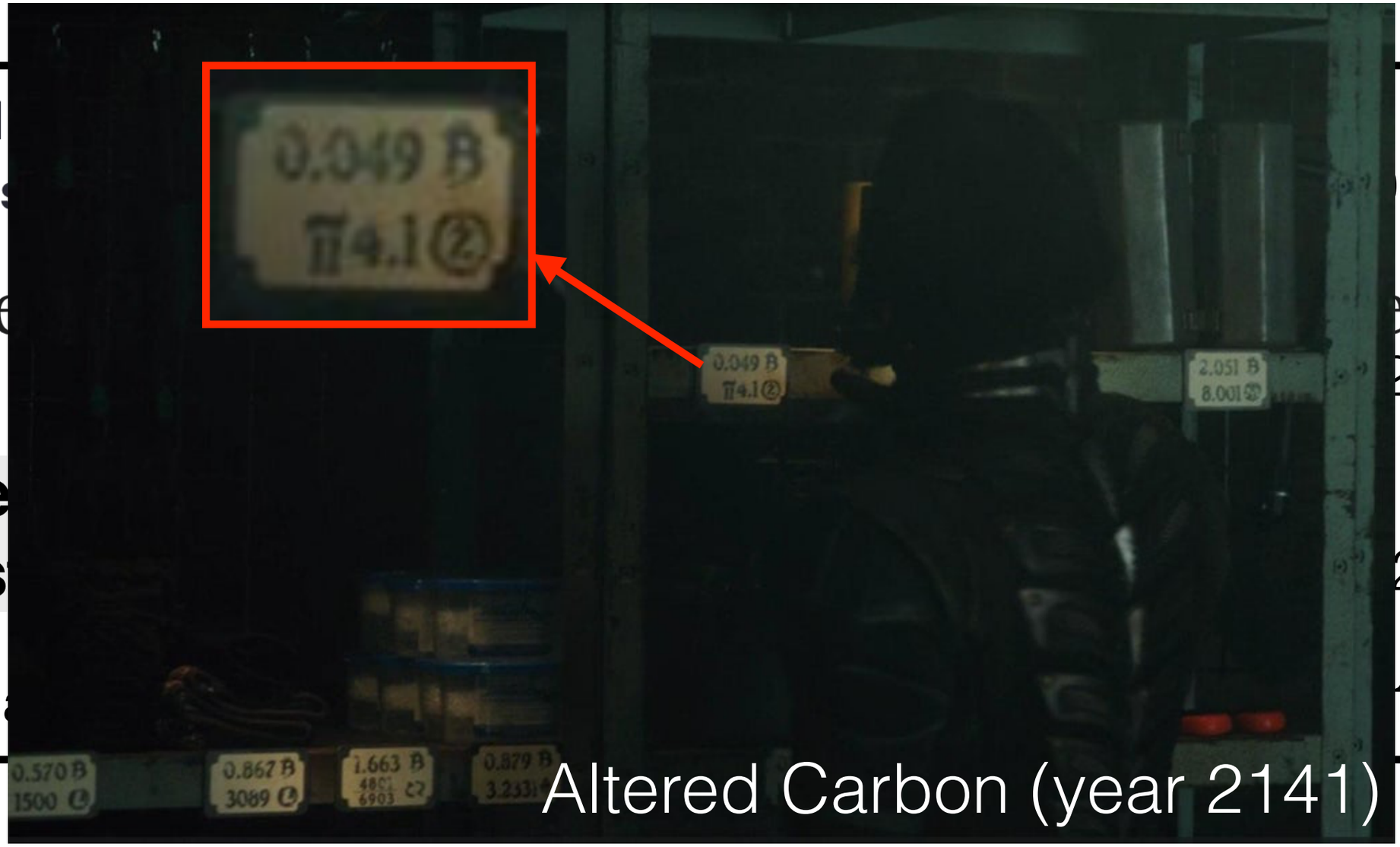
Post-Quantum Security



All DL-based constructions are **insecure against quantum adversaries**.

If we build long-term public infrastructure using zkSNARKs
then security against quantum adversaries is **paramount.**

Post-Quantum Security



All DL-based constructions are **insecure against quantum adversaries**.

If we build long-term public infrastructure using zkSNARKs
then security against quantum adversaries is **paramount.**

All constructions presented in this talk are post-quantum secure!

2018: “plausibly” post-quantum

2018: “plausibly” post-quantum

2021: *provably* post-quantum

2018: “plausibly” post-quantum

2021: *provably* post-quantum

Theorem [CMS19]: The [BCS16] transformation is secure in the quantum random oracle model (QROM)

2018: “plausibly” post-quantum

2021: *provably* post-quantum

Theorem [CMS19]: The [BCS16] transformation is secure in the quantum random oracle model (QROM)

QROM models **superposition access** to hash functions

2018: “plausibly” post-quantum

2021: *provably* post-quantum

Theorem [CMS19]: The [BCS16] transformation is secure in the quantum random oracle model (QROM)

QROM models **superposition access** to hash functions

Caveat: IOP must satisfy **round-by-round** soundness, stronger than SR

2018: “plausibly” post-quantum

2021: *provably* post-quantum

Theorem [CMS19]: The [BCS16] transformation is secure in the quantum random oracle model (QROM)

QROM models **superposition access** to hash functions

Caveat: IOP must satisfy **round-by-round** soundness, stronger than SR

Theorem [CMSZ21]: Kilian’s interactive protocol (for PCPs) is post-quantum secure assuming QLWE.

2018: “plausibly” post-quantum

2021: *provably* post-quantum

Theorem [CMS19]: The [BCS16] transformation is secure in the quantum random oracle model (QROM)

QROM models **superposition access** to hash functions

Caveat: IOP must satisfy **round-by-round** soundness, stronger than SR

Theorem [CMSZ21]: Kilian’s interactive protocol (for PCPs) is post-quantum secure assuming QLWE.

This is highly non-trivial to show: classical proof is by **rewinding** which is **impossible** for quantum adversaries

2018: “plausibly” post-quantum

2021: *provably* post-quantum

Theorem [CMS19]: The [BCS16] transformation is secure in the quantum random oracle model (QROM)

QROM models **superposition access** to hash functions

Caveat: IOP must satisfy **round-by-round** soundness, stronger than SR

Theorem [CMSZ21]: Kilian’s interactive protocol (for PCPs) is post-quantum secure assuming QLWE.

This is highly non-trivial to show: classical proof is by **rewinding** which is **impossible** for quantum adversaries

---

Many open questions: security of Kilian for IOPs, security from “plain” CRHFs, security of “lattice Bulletproofs”, more efficient lattice-based constructions…

2018: “plausibly” post-quantum

2021: *provably* post-quantum

Theorem [CMS19]: The [BCS16] transformation is secure in the quantum random oracle model (QROM)

QROM models **superposition access** to hash functions

Caveat: IOP must satisfy **round-by-round** soundness, stronger than SR

Theorem [CMSZ21]: Kilian’s interactive protocol (for PCPs) is post-quantum secure assuming QLWE.

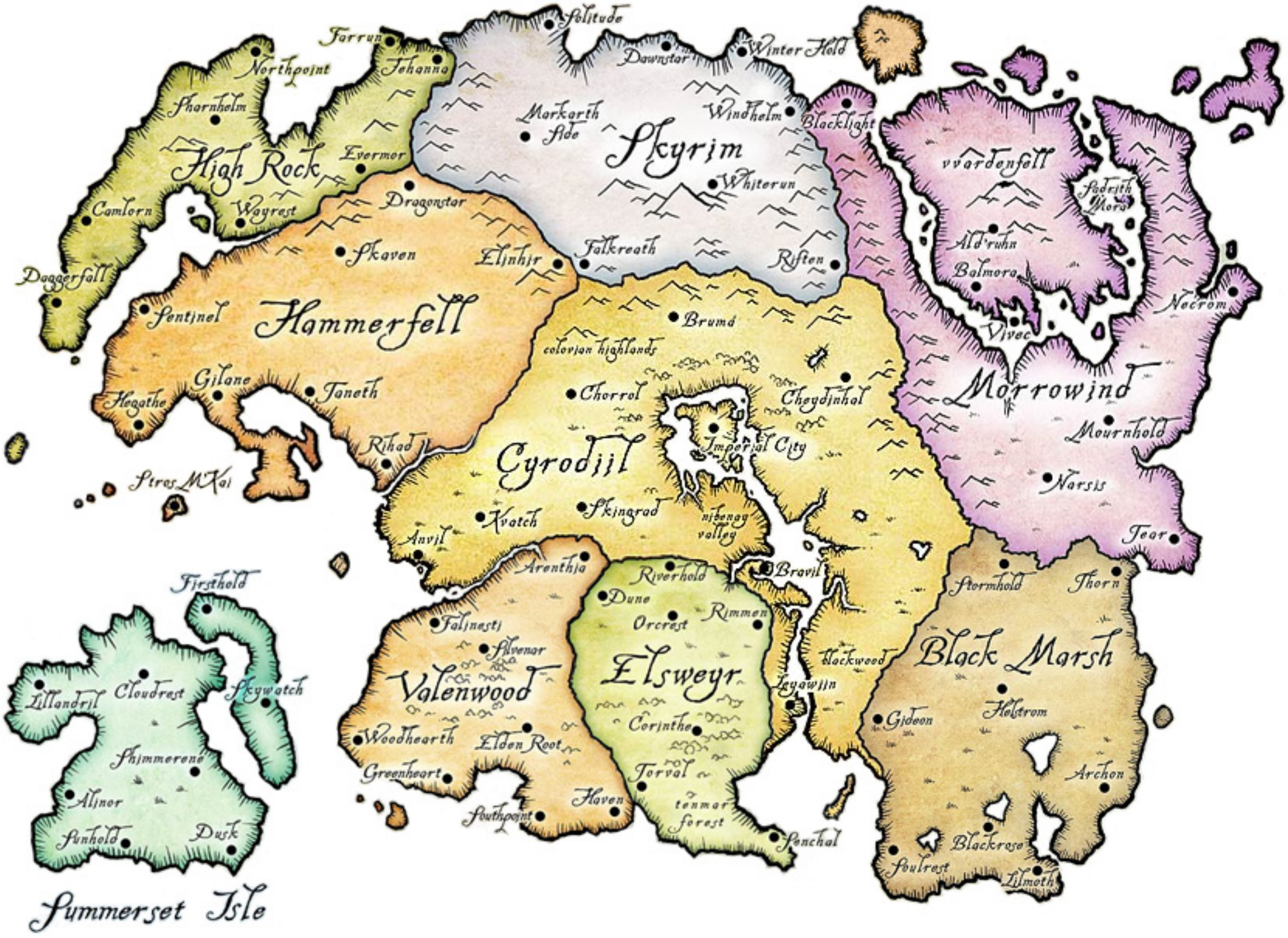
This is highly non-trivial to show: classical proof is by **rewinding** which is **impossible** for quantum adversaries

---

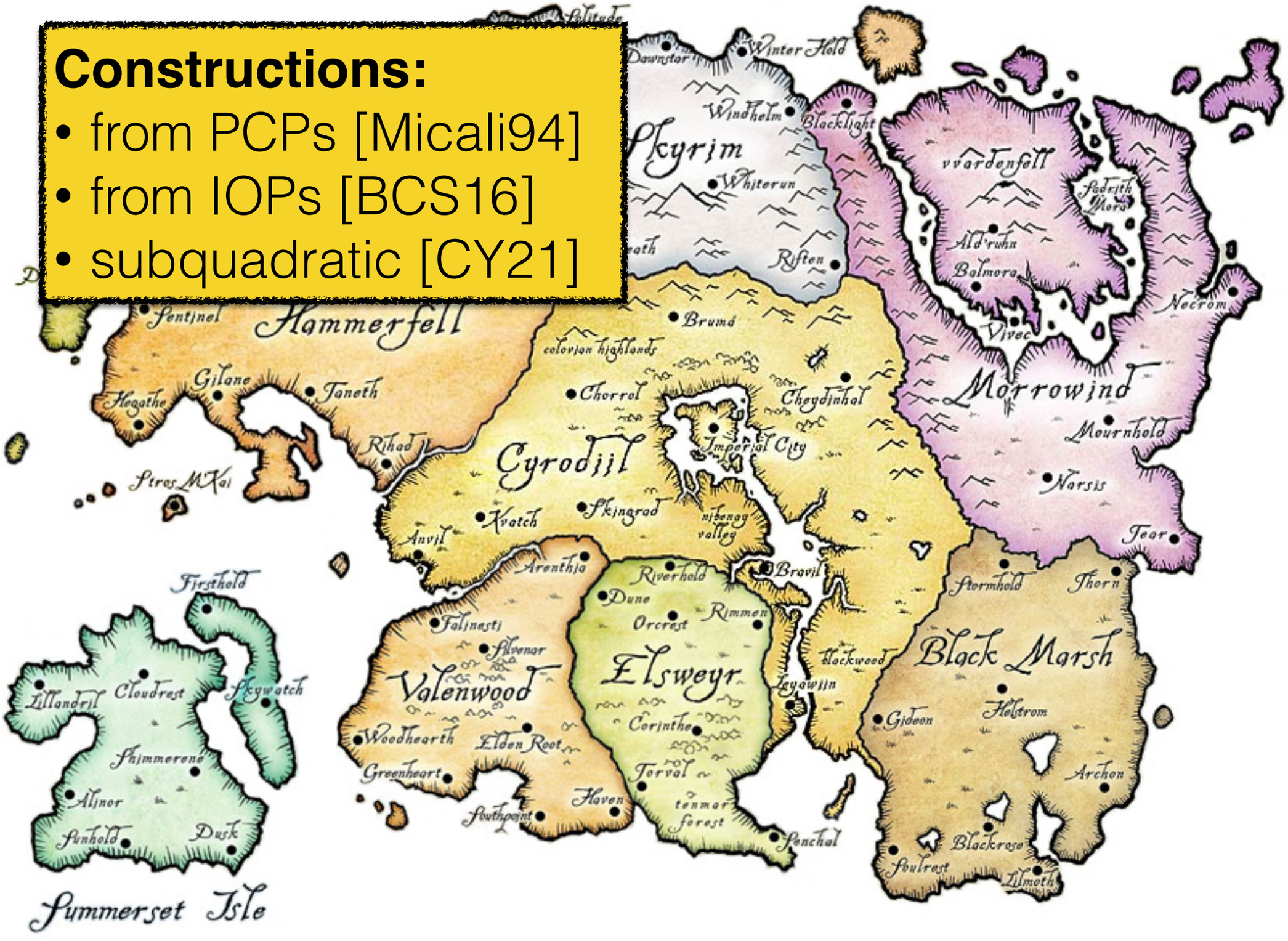
Many open questions: security of Kilian for IOPs, security from “plain” CRHFs, security of “lattice Bulletproofs”, more efficient lattice-based constructions…

Very active research area (talk to me if interested)

Conclusion: The World of Hash-Based SNARKs



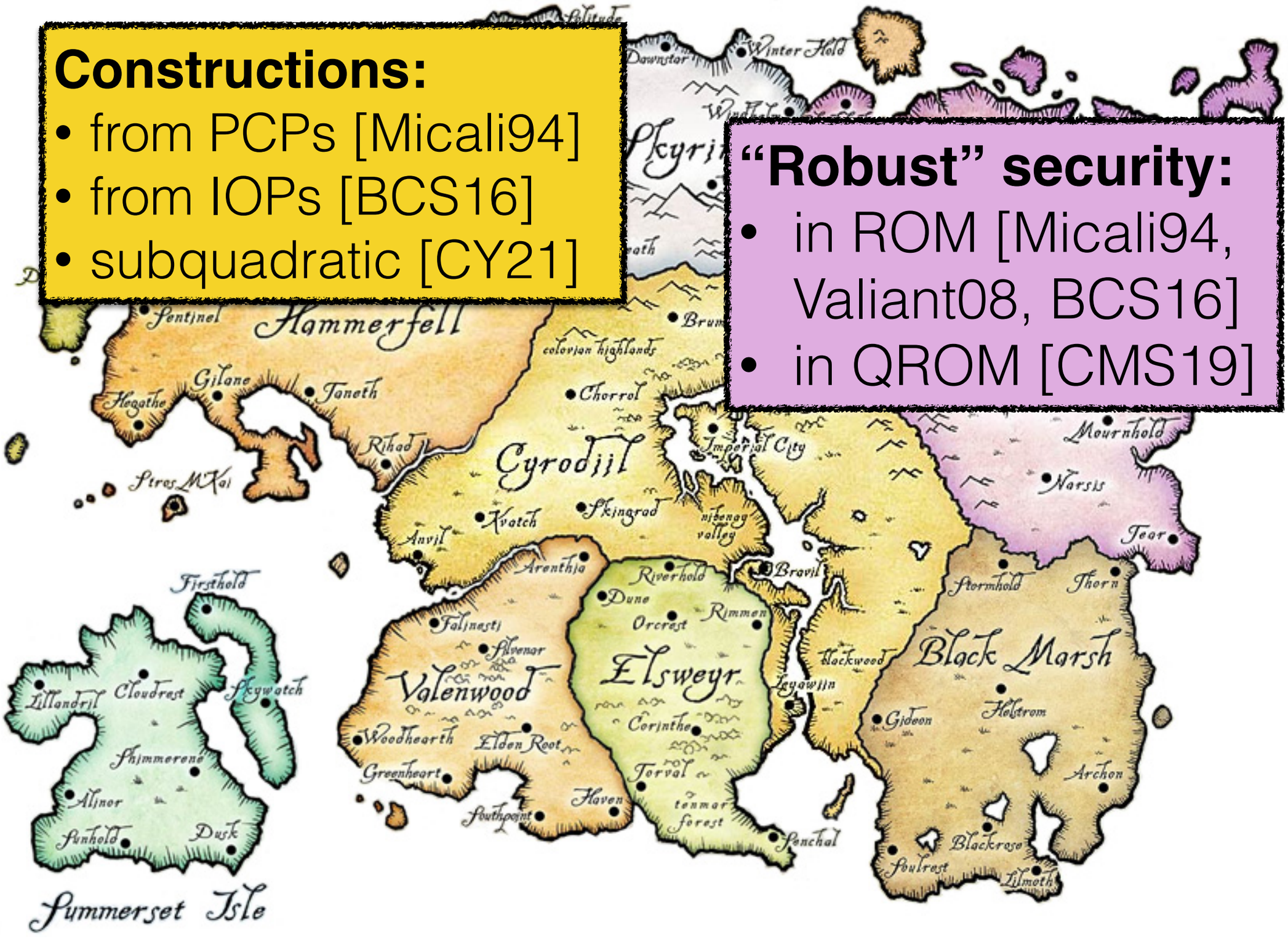
Conclusion: The World of Hash-Based SNARKs



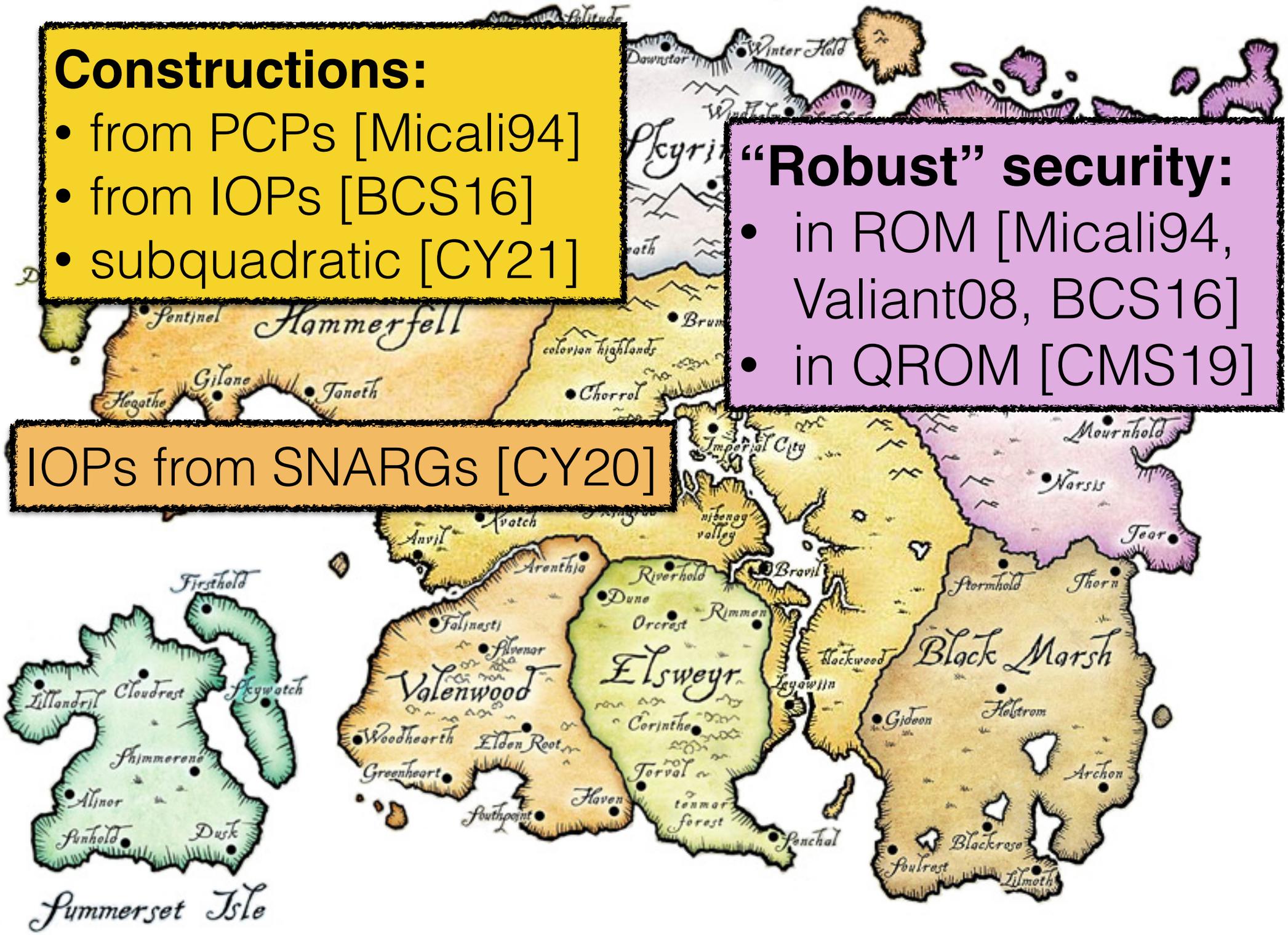
Conclusion: The World of Hash-Based SNARKs

Constructions:
- from PCPs [Micali94]
- from IOPs [BCS16]
- subquadratic [CY21]

"Robust" security:
- in ROM [Micali94, Valiant08, BCS16]
- in QROM [CMS19]



Conclusion: The World of Hash-Based SNARKs



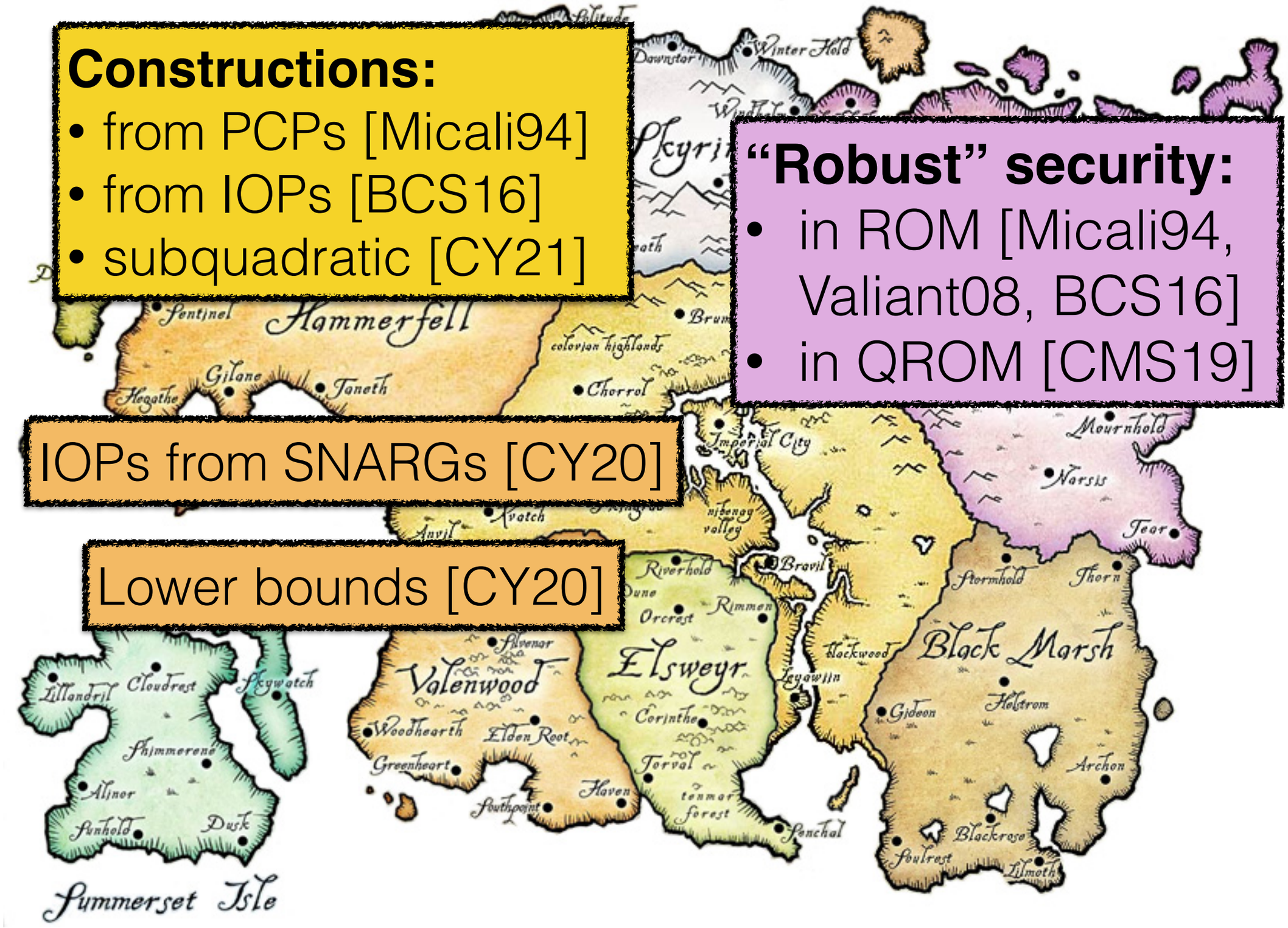
Conclusion: The World of Hash-Based SNARKs

Constructions:
- from PCPs [Micali94]
- from IOPs [BCS16]
- subquadratic [CY21]

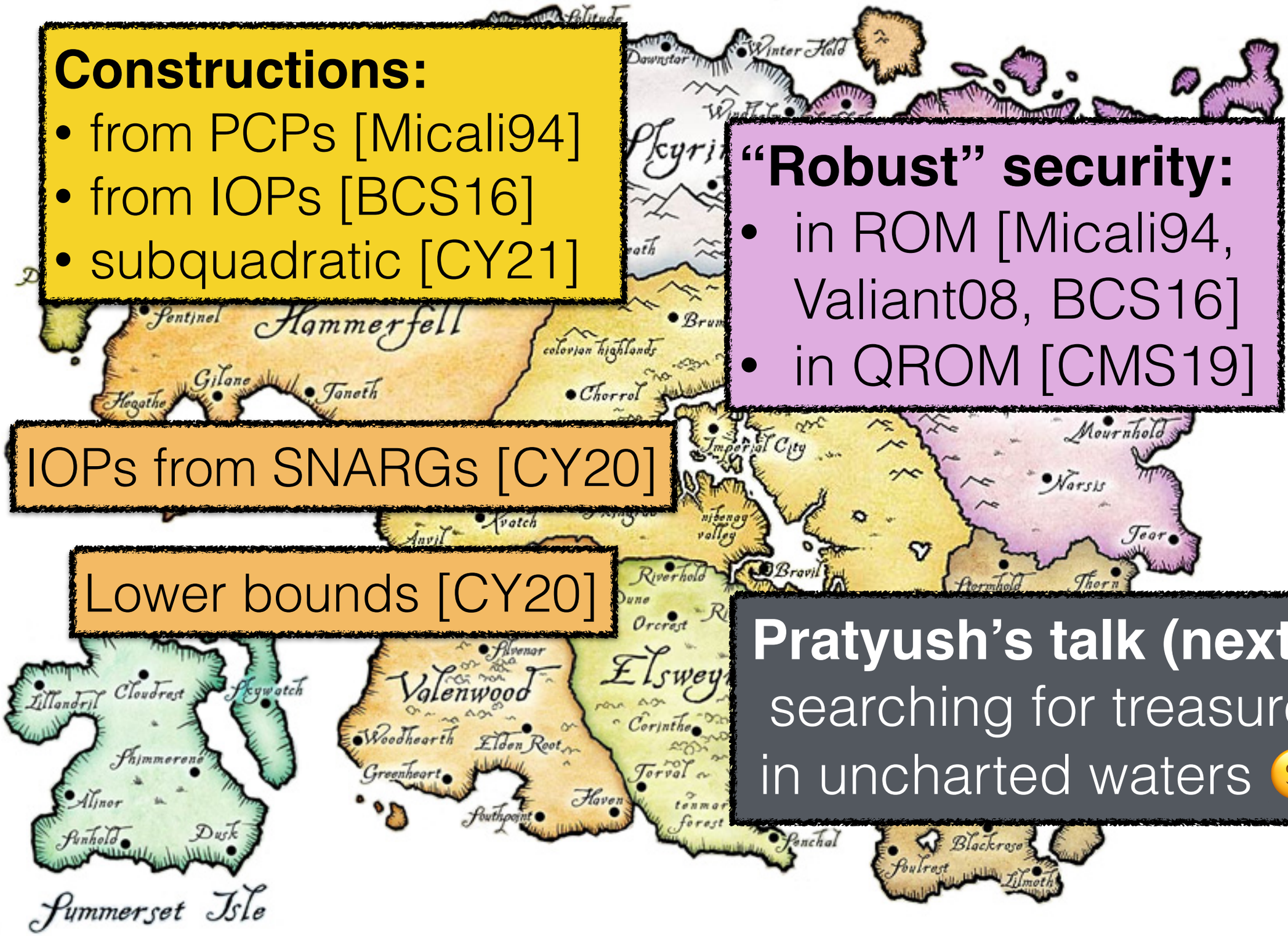
"Robust" security:
- in ROM [Micali94, Valiant08, BCS16]
- in QROM [CMS19]

IOPs from SNARGs [CY20]

Lower bounds [CY20]



Conclusion: The World of Hash-Based SNARKs



Thanks!