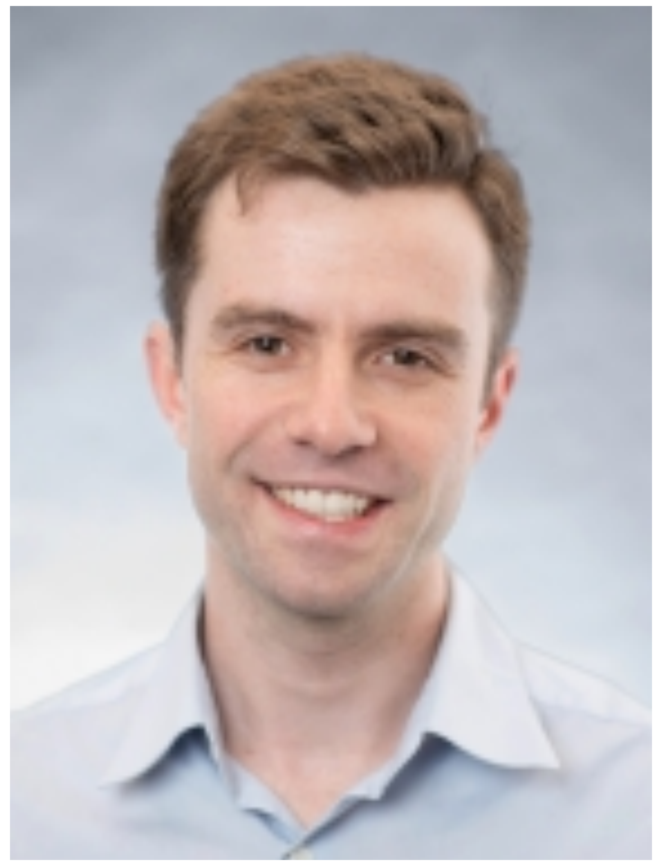
SNARGs for NP & Non-Signaling PCPs, Revisited

Surya Mathialagan

MIT → **NTT Research**

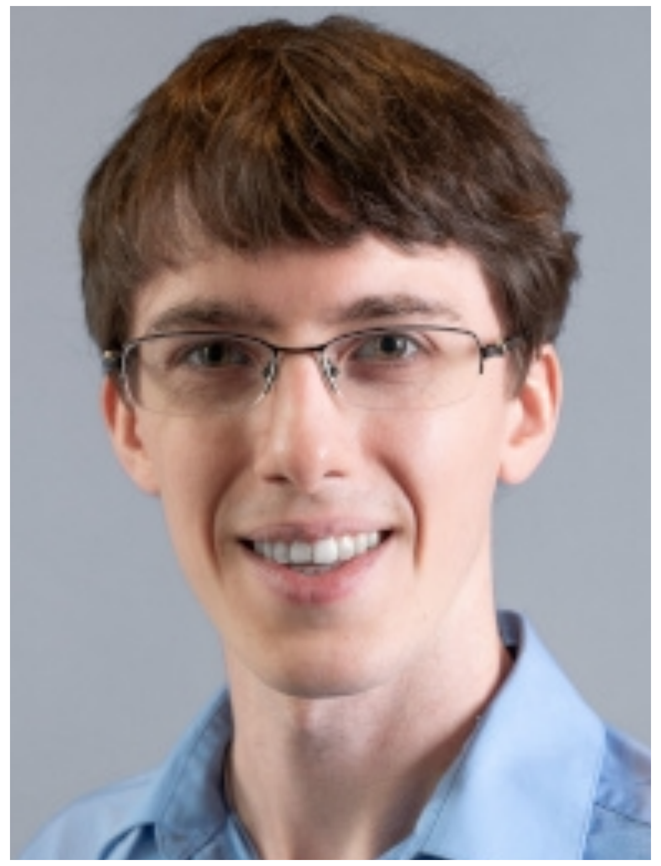
Lali Devadas
MIT



Sam Hopkins
MIT

Yael Kalai
MIT

Pravesh Kothari
Princeton



Alex Lombardi
Princeton


WANTED
DEAD OR ALIVE
CASH REWARD
$ 10.000


WANTED
PROVEN
OR ALIVE
CASH
REWARD
$ 10.000


WANTED
PROVEN
OR
DISPROVEN
CASH
REWARD
$ 10.000

WANTED

PROVEN OR **DISPROVEN**

CASH REWARD $ 10 **+ COOKIES**


WANTED
PROVEN
OR
DISPROVEN
Low-Norm
Nullstellensatz
Conjecture
CASH
REWARD
$ 10
+ COOKIES

TLDR

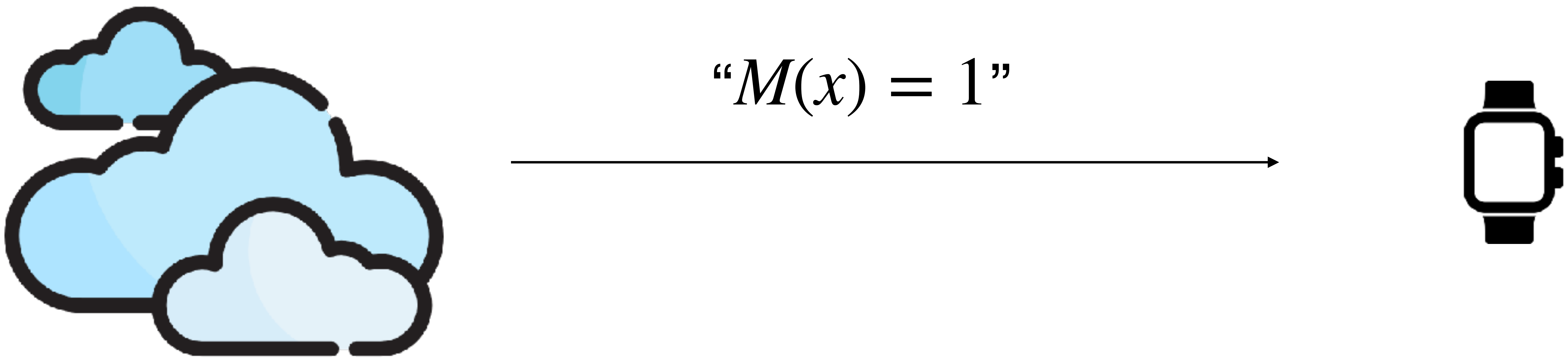
- **Theorem.** We construct SNARGs for NP assuming:
  - Hardness of LWE, Bilinear Maps or DDH,
  - A mathematical conjecture above multivariate polynomials of reals.
- **This talk:** I will talk about this fascinating connection between SNARGs and PCPs [BMW98, KRR14, BHK17, BKKSW18]
  - Giving you an open problem to solve :)

Delegation of Computation

Delegation of Computation

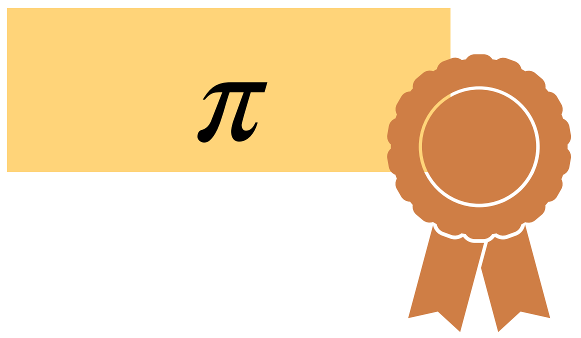
Delegation of Computation

Delegation of Computation

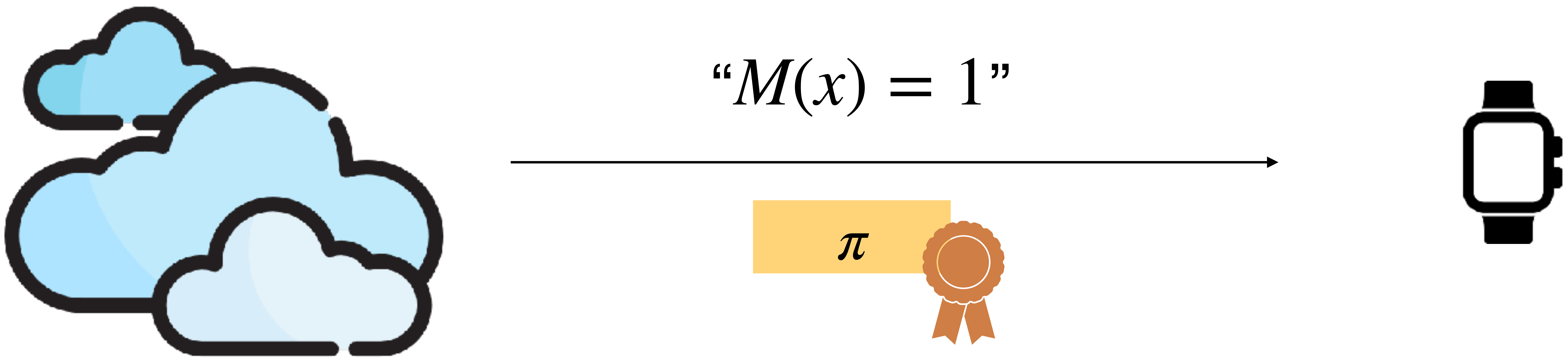


Delegation of Computation

“$M(x) = 1$”

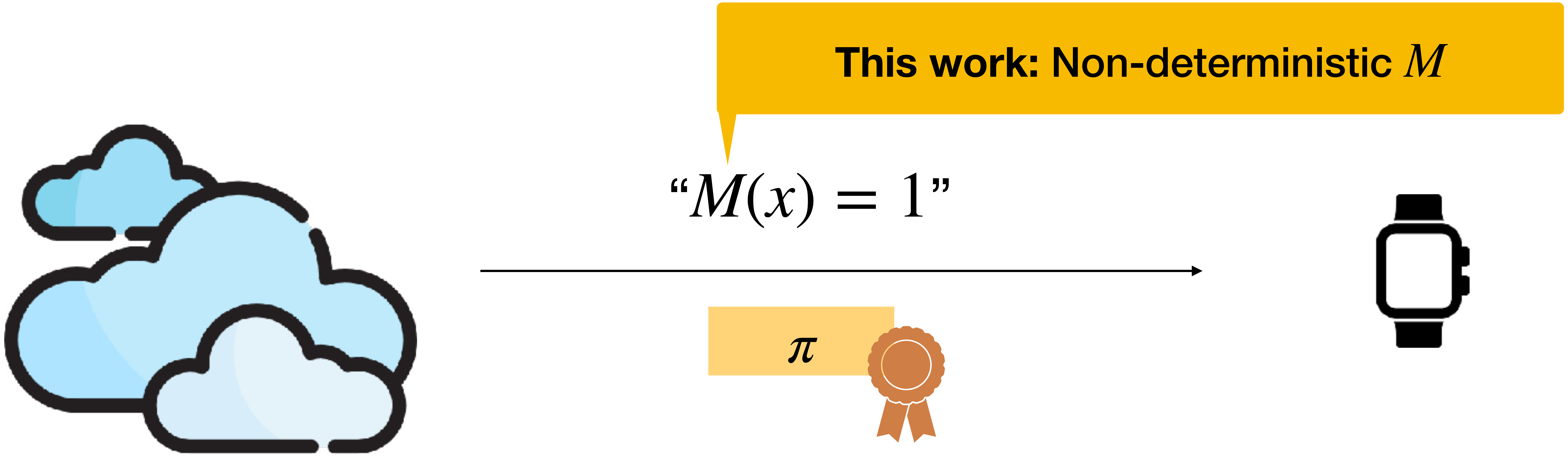


Delegation of Computation



Can the cloud attach a **small, efficiently verifiable proof** that he did the computation correctly?

Delegation of Computation



Can the cloud attach a **small, efficiently verifiable proof** that he did the computation correctly?

Succinct **N**on-Interactive **Arg**uments for NP

Succinct Non-Interactive Arguments for NP

$\mathcal{P}$

Succinct Non-Interactive Arguments for NP

"$x \in \mathscr{L}$"

$\mathscr{P}$

Succinct Non-Interactive Arguments for NP

“$x \in \mathscr{L}$”

$\mathscr{P}$ 

 $\mathscr{V}$

Succinct Non-Interactive Arguments for NP

"$x \in \mathscr{L}$"

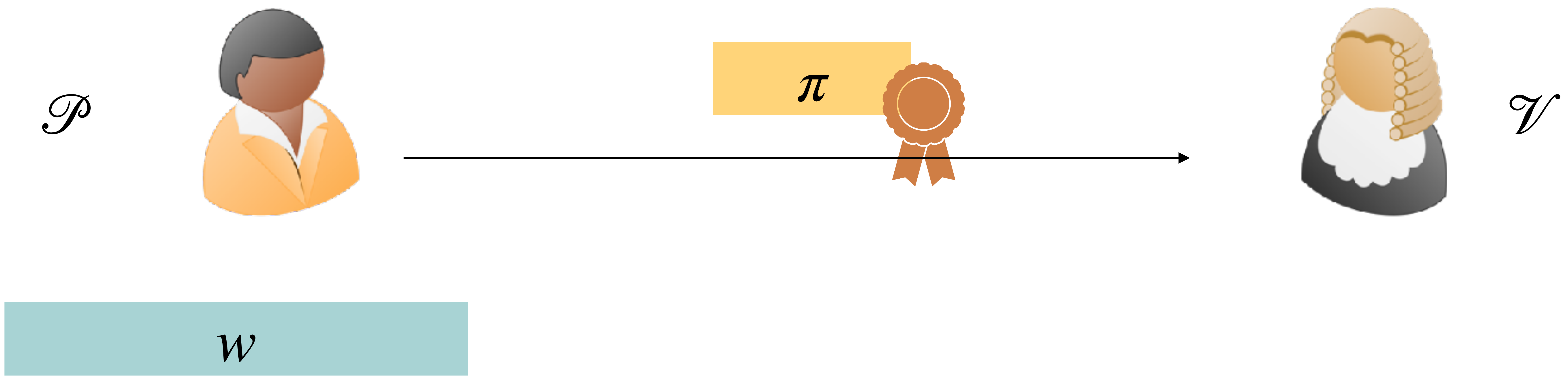
$\mathscr{P}$ 

 $\mathscr{V}$

w

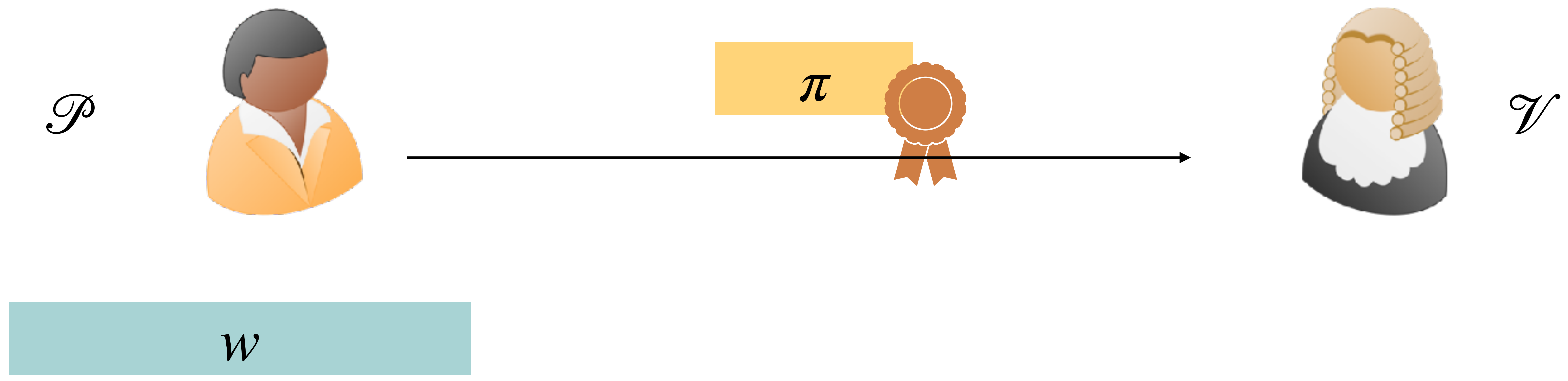
Succinct Non-Interactive Arguments for NP

"$x \in \mathscr{L}$"



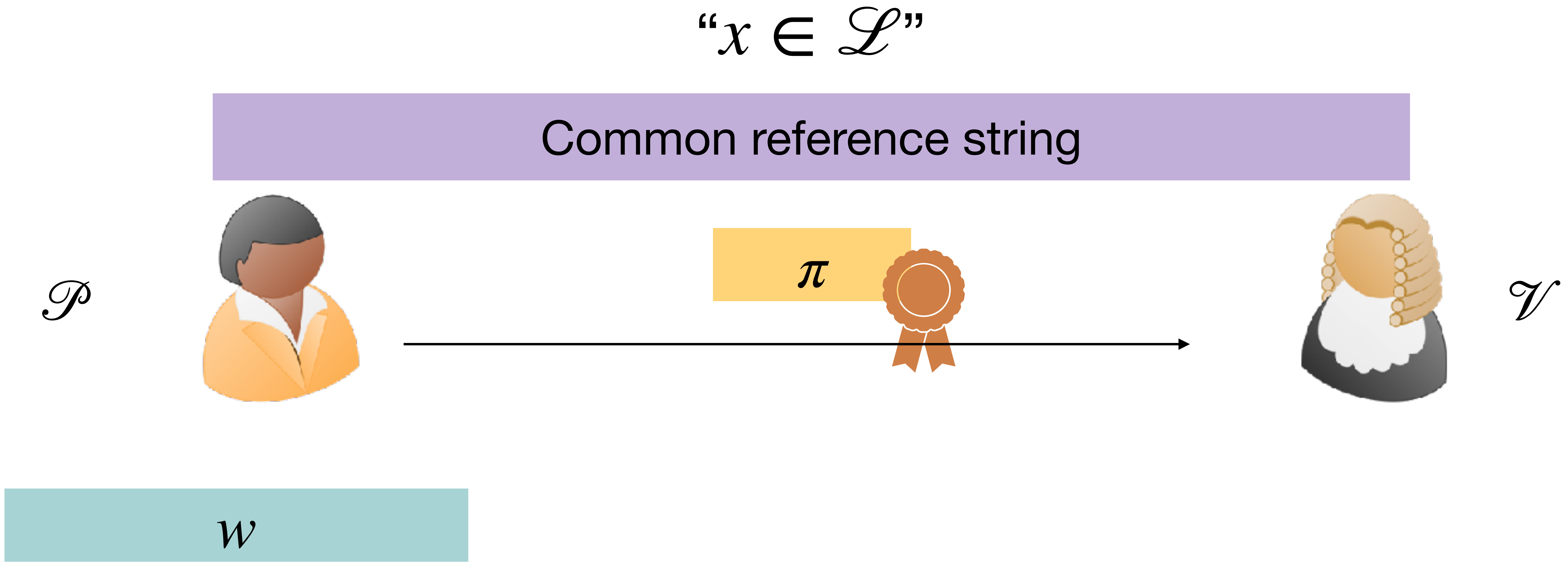
<u>S</u>uccinct <u>N</u>on-Interactive <u>Arg</u>uments for NP

"$x \in \mathscr{L}$"



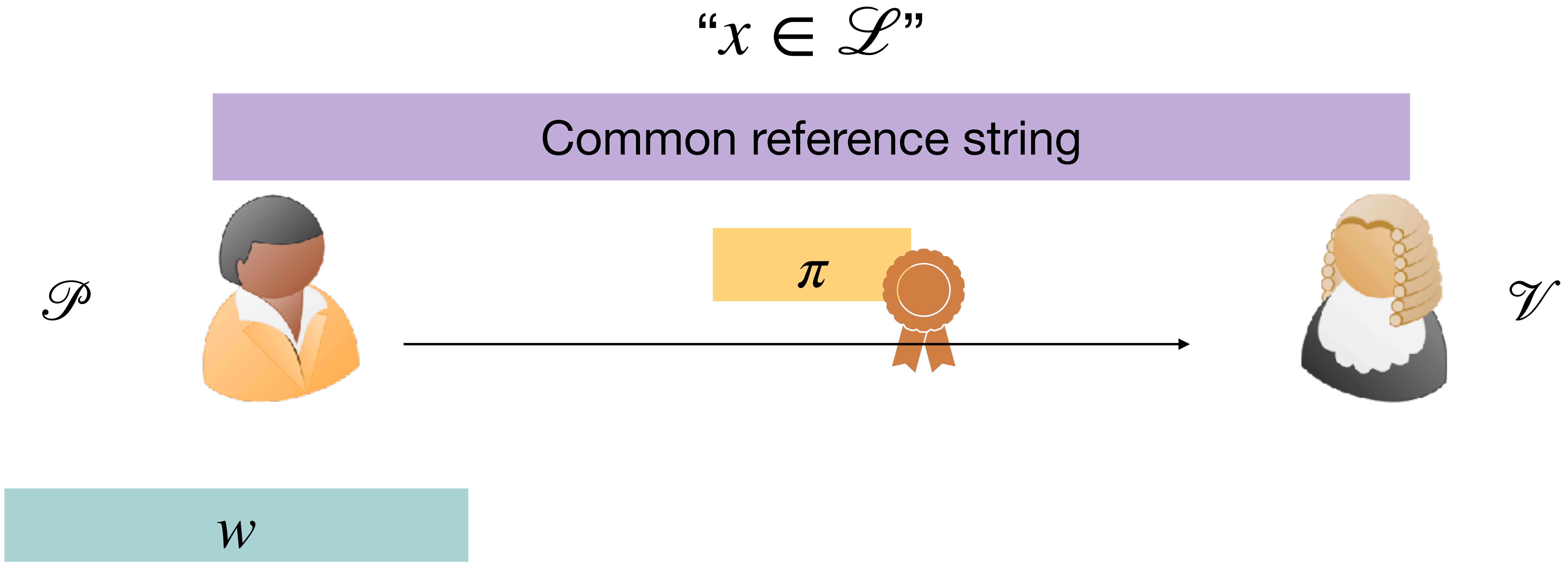
- **Succinctness:** $|\pi| \ll |w|$.

Succinct Non-Interactive Arguments for NP



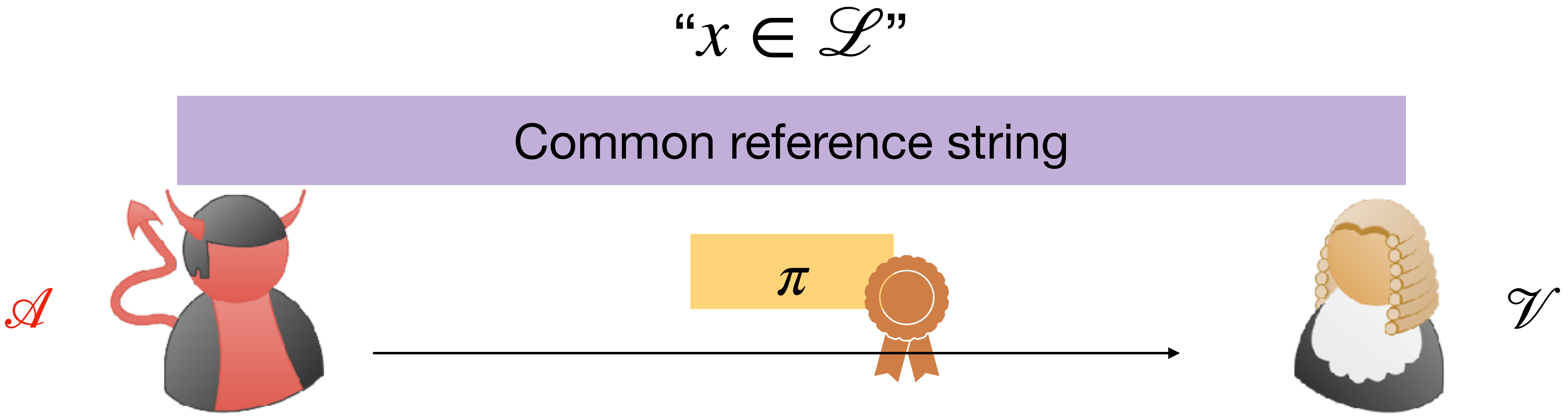
- **Succinctness:** $|\pi| \ll |w|$.

Succinct Non-Interactive Arguments for NP



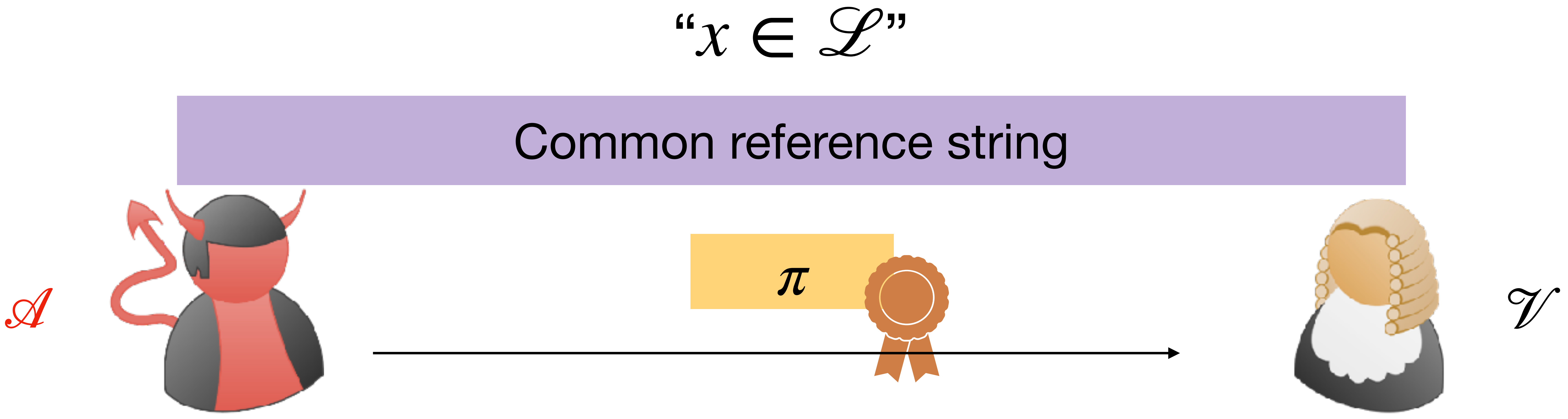
- **Succinctness:** $|\pi| \ll |w|$.
- **Correctness:** If π is honestly generated, $\mathscr{V}(\text{crs}, x, \pi)$ accepts.

Succinct Non-Interactive Arguments for NP



- **Succinctness:** $|\pi| \ll |w|$.
- **Correctness:** If π is honestly generated, $\mathscr{V}(\mathsf{crs}, x, \pi)$ accepts.

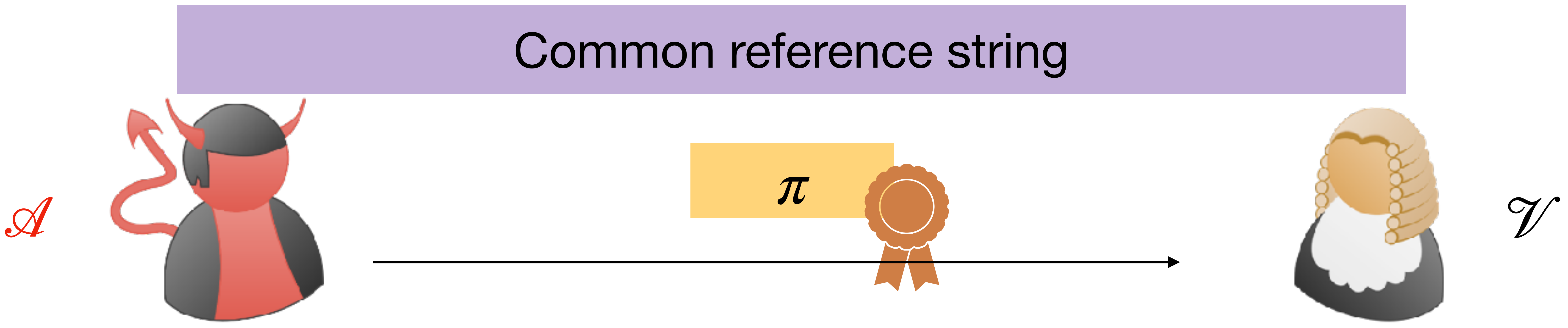
Succinct Non-Interactive Arguments for NP



- **Succinctness:** $|\pi| \ll |w|$.
- **Correctness:** If π is honestly generated, $\mathscr{V}(\text{crs}, x, \pi)$ accepts.
- **(Non-Adaptive) Soundness:** If $x \notin \mathscr{L}$, difficult for ppt $\mathscr{A}$ to come up with accepting proof.

<u>S</u>uccinct <u>N</u>on-Interactive <u>Arg</u>uments for NP

"$x \in \mathscr{L}$"



$$\Pr_{\text{crs}}[\pi \leftarrow \mathscr{A}(\text{crs}) \wedge \mathscr{V}(\text{crs}, x, \pi) = 1] \leq 2^{-\lambda}$$

- **Succinctness:** $|\pi| \ll |w|$.
- **Correctness:** If π is honestly generated, $\mathscr{V}(\text{crs}, x, \pi)$ accepts.
- **(Non-Adaptive) Soundness:** If $x \notin \mathscr{L}$, difficult for ppt $\mathscr{A}$ to come up with accepting proof.

SNARGs for NP?

SNARGs for NP?

Micali'00, IKO'07, GKR'08, IKOS'09, Groth'10, SBW'11, SMBW'12, Lipmaa'12, CMT'12, DFH'12, SVPBBW'12, TRMP'12, GGPR'13, BCIOP'13, BCCT'13, Thaler'13, BCGTV'13, PHGR'13, BSCGT'13, BCGGMTV'14, BCCGP'16, Groth'16, GMO'16, GLRT'17, AHIV'17, BSBCGGHPRST'17, WJBSTWW'17, BBBPWM'18, BCGMMW'18, BSBHR'18, WTSTW'18, WZCPS'18, GMNO'18, FKL'18, BBCGI'19, BBHR'19, BCRSVW'19, BSCRSVW'19, CFQ19, GWC'19, KPV'19, KPY'19, MBKM'19, Nitulescu'19, XZZPS'19, Gabizon'19, BBS'20, BSCIKS'20, BFHVXZ'20, COS'20, CHMMVW'20, KZ'20, KPPS'20, SGKS'20, SL'20, Setty'20, ZXZS'20, BMMTV'21, GLSTW'21, GMN21, GPR'21, Sta'21, ZLWZSXZ'21, Bay'22, CBBZ'22, XZCZZJBS'22, XZS'22, …

SNARGs for NP?

Micali'00, IKO'07, GKR'08, IKOS'09, Groth'10, SBW'11, SMBW'12, Lipmaa'12, CMT'12, DFH'12, SVPBBW'12, TRMP'12, GGPR'13, BCIOP'13, BCCT'13, Thaler'13, BCGTV'13, PHGR'13, BSCGT'13, BCGGMTV'14, BCCGP'16, Groth'16, GMO'16, GLRT'17, AHIV'17, BSBCGGHPRST'17, WJBSTWW'17, BBBPWM'18, BCGMMW'18, BSBHR'18, WTSTW'18, WZCPS'18, GMNO'18, FKL'18, BBCGI'19, BBHR'19, BCRSVW'19, BSCRSVW'19, CFQ'19, GWC'19, KPV'19, KPY'19, MBKM'19, Nitulescu'19, XZZPS'19, Gabizon'19, BBS'20, BSCIKS'20, BFHVXZ'20, COS'20, CHMMVW'20, KZ'20, KPPS'20, SGKS'20, SL'20, Setty'20, ZXZS'20, BMMTV'21, GLSTW'21, GMN21, GPR'21, Sta'21, ZLWZSXZ'21, Bay'22, CBBZ'22, XZCZZJBS'22, XZS'22, …

Random Oracle/Knowledge Assumptions

SNARGs for NP in Standard Model?

SNARGs for NP in Standard Model?

SNARGs for NP in Standard Model?

SNARGs for NP

SNARGs for NP in Standard Model?

SNARGs for NP

From indistinguishability obfuscation
[SW ’14, WW ’24, **M**PV ’24, WZ ’25, WW ’25]

SNARGs for NP in Standard Model?

SNARGs for NP

From indistinguishability obfuscation
[SW ’14, WW ’24, **M**PV ’24, WZ ’25, WW ’25]
From witness encryption* [JKL**M** ’25]

SNARGs for NP in Standard Model?

SNARGs for NP

Obfustopia

Not known from standard, post-quantum assumptions

SNARGs for NP in Standard Model?

SNARGs for NP

Obfustopia

Not known from standard,
post-quantum assumptions

From indistinguishability obfuscation
[SW ’14, WW ’24, **M**PV ’24, WZ ’25, WW ’25]
From witness encryption* [JKL**M** ’25]

SNARGs for NP in Standard Model?

SNARGs for NP

Obfustopia

Not known from standard, post-quantum assumptions

From Learning with Error, Bilinear Maps, Etc.

SNARGs for NP in Standard Model?

SNARGs for NP

Obfustopia

Not known from standard, post-quantum assumptions

From Learning with Error, Bilinear Maps, Etc.

P or Batch-NP [KRR14, BHK17, CJJ21, KVZ21]

SNARGs for NP in Standard Model?

SNARGs for NP

From indistinguishability obfuscation [SW '14, WW '24, **M**PV '24, WZ '25, WW '25]

From witness encryption* [JKL**M** '25]

Obfustopia

Not known from standard, post-quantum assumptions

From Learning with Error, Bilinear Maps, Etc.

P or Batch-NP [KRR14, BHK17, CJJ21, KVZ21]

Monotone-Policy Batch-NP [KRR14, BHK17, CJJ21, KVZ21]

SNARGs for NP in Standard Model?

SNARGs for NP

Obfustopia

Not known from standard, post-quantum assumptions

From indistinguishability obfuscation [SW '14, WW '24, **M**PV '24, WZ '25, WW '25]

From witness encryption* [JKL**M** '25]

From Learning with Error, Bilinear Maps, Etc.

P or Batch-NP [KRR14, BHK17, CJJ21, KVZ21]

Monotone-Policy Batch-NP [KRR14, BHK17, CJJ21, KVZ21]

Natural subclasses of NP ∩ coNP [JKLV24, JKL**M**25]

SNARGs for NP in Standard Model?

SNARGs for NP

Obfustopia

Not known from standard,
post-quantum assumptions

From indistinguishability obfuscation
[SW '14, WW '24, **M**PV '24, WZ '25, WW '25]

From witness encryption* [JKL**M** '25]

From Learning with Error, Bilinear Maps, Etc.

Subclass of NP

P or Batch-NP [KRR14, BHK17, CJJ21, KVZ21]

Monotone-Policy Batch-NP [KRR14, BHK17, CJJ21, KVZ21]

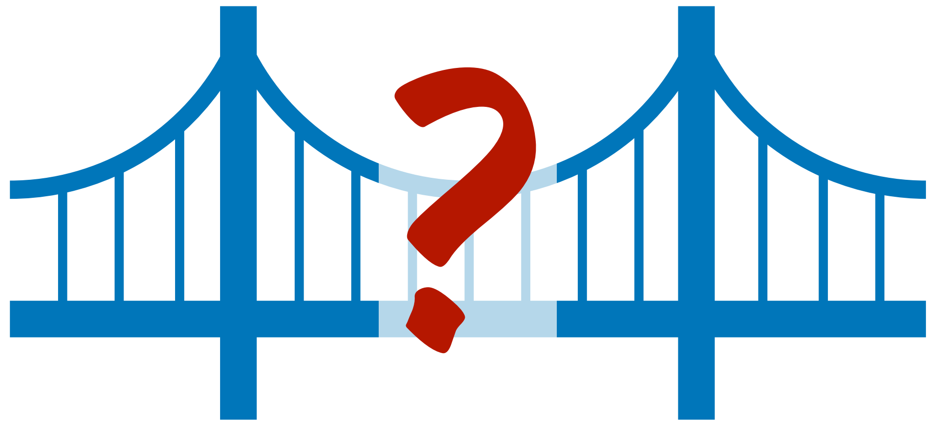
Natural subclasses of NP $\cap$ coNP [JKLV24, JKL**M**25]

SNARGs for NP in Standard Model?

SNARGs for NP

Obfustopia

Not known from standard, post-quantum assumptions



From Learning with Error, Bilinear Maps, Etc.

P or Batch-NP [KRR14, BHK17, CJJ21, KVZ21]

Subclass of NP

Natural subclasses of NP ∩ coNP [JKLV24, JKLM25]

SNARGs for NP in Standard Model?

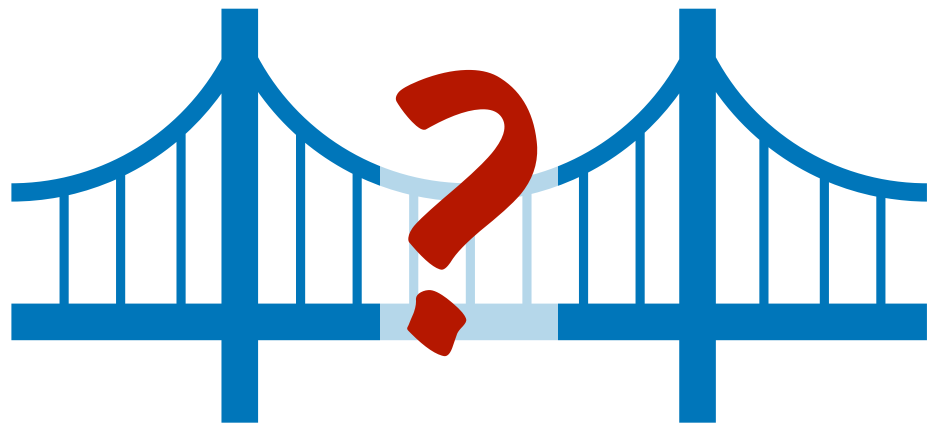
SNARGs for NP

Obfustopia

Not known from standard, post-quantum assumptions

From indistinguishability obfuscation [SW '14, WW '24, MPV '24, WZ '25, WW '25]

From witness encryption* [JKLM '25]



Can we build SNARGs from LWE/Bilinear Maps/etc?

From Learning with Error, Bilinear Maps, Etc.

Subclass of NP

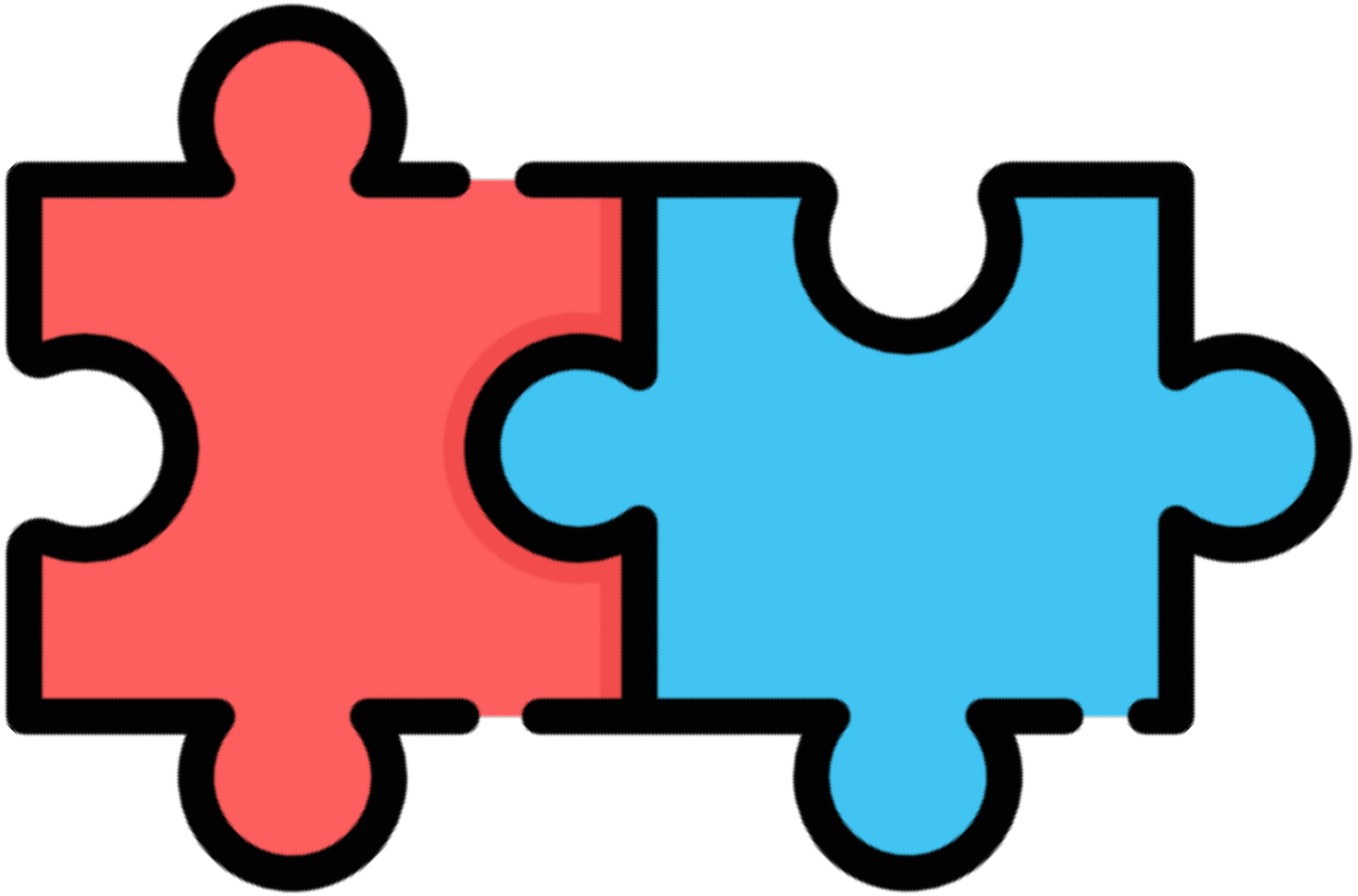
P or Batch-NP [KRR14, BHK17, CJJ21, KVZ21]

Monotone-Policy Batch-NP [KRR14, BHK17, CJJ21, KVZ21]

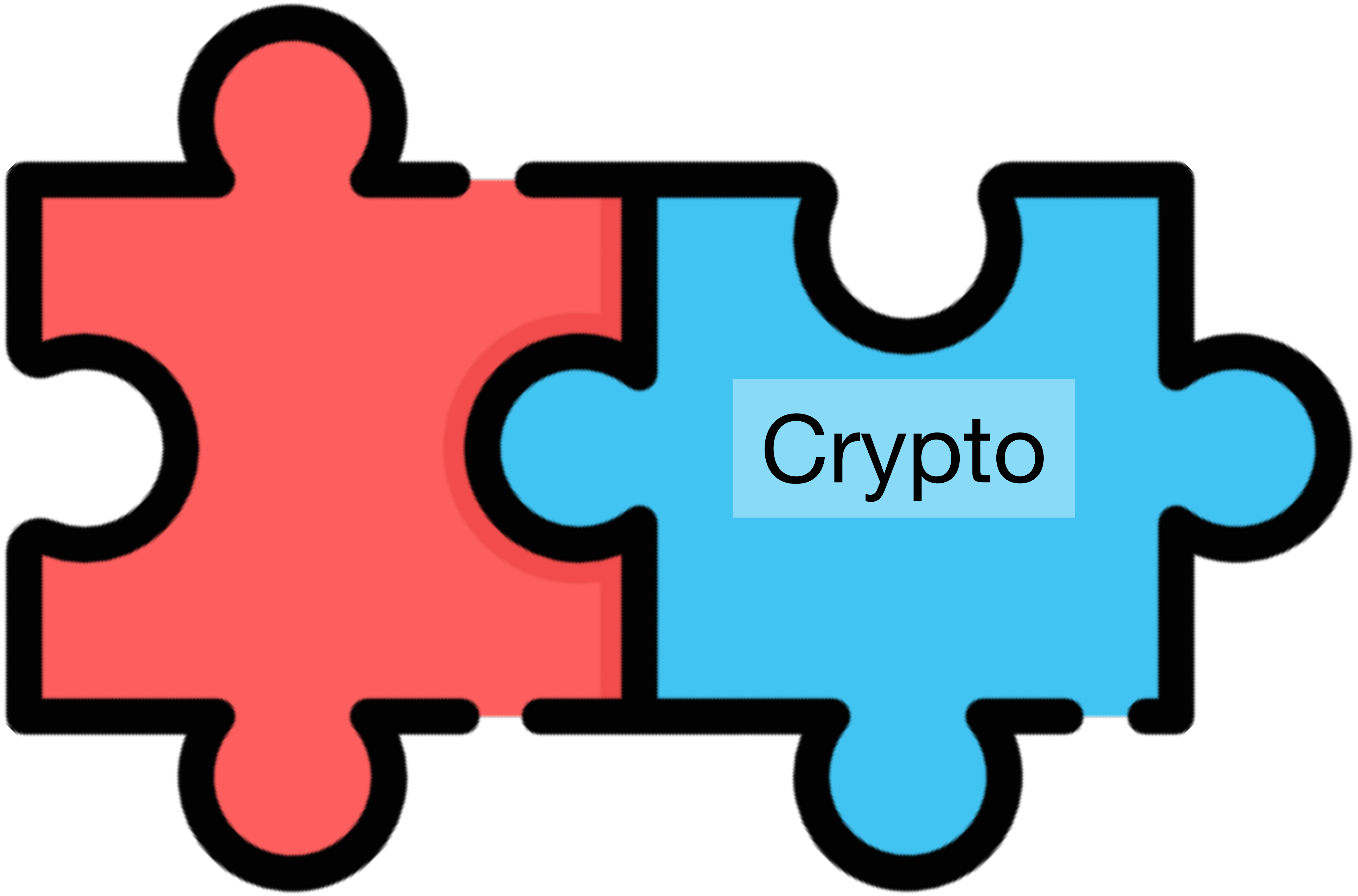
Natural subclasses of NP ∩ coNP [JKLV24, JKLM25]

How to construct SNARGs?

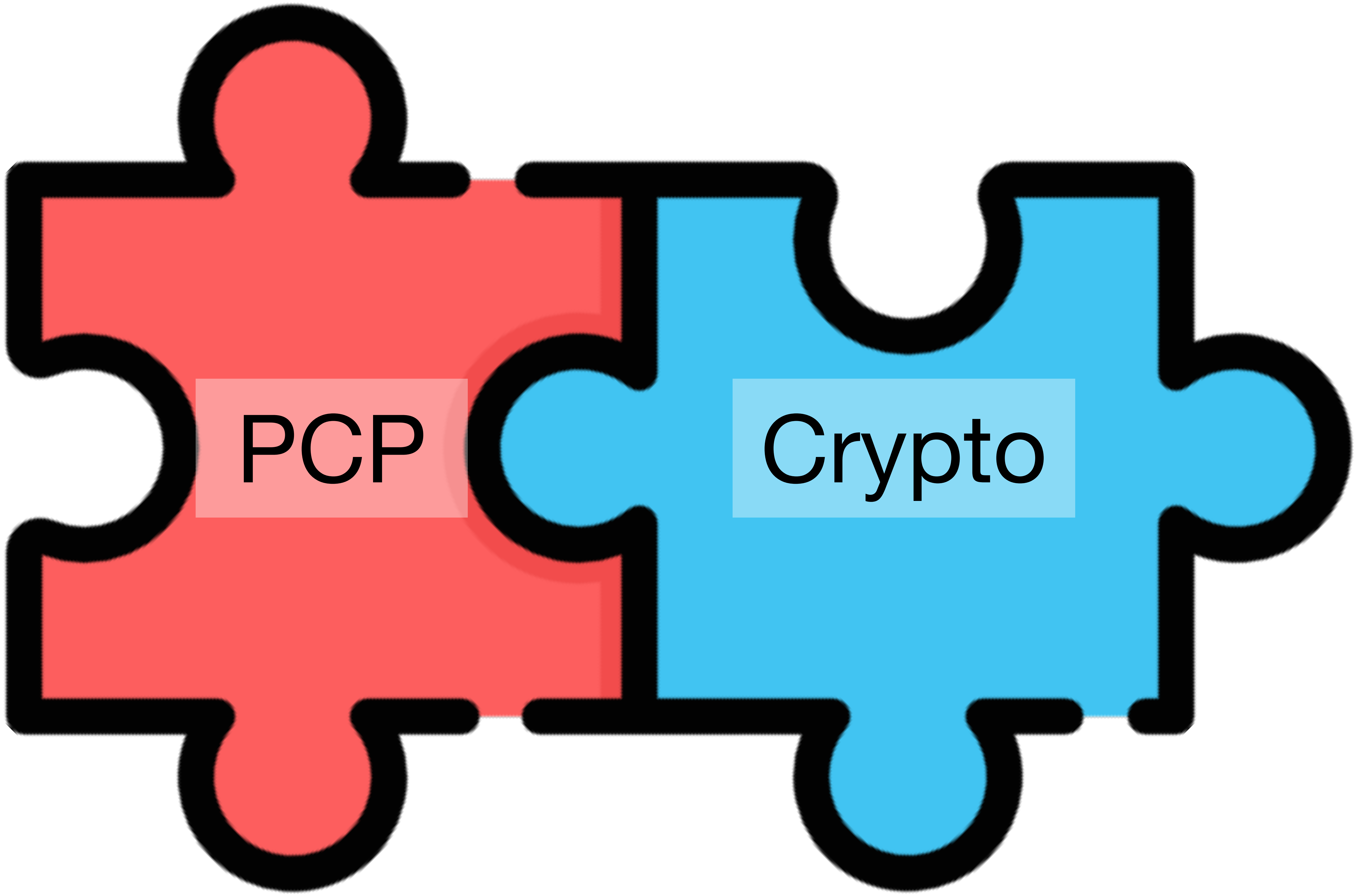
How to construct SNARGs?



How to construct SNARGs?



How to construct SNARGs?

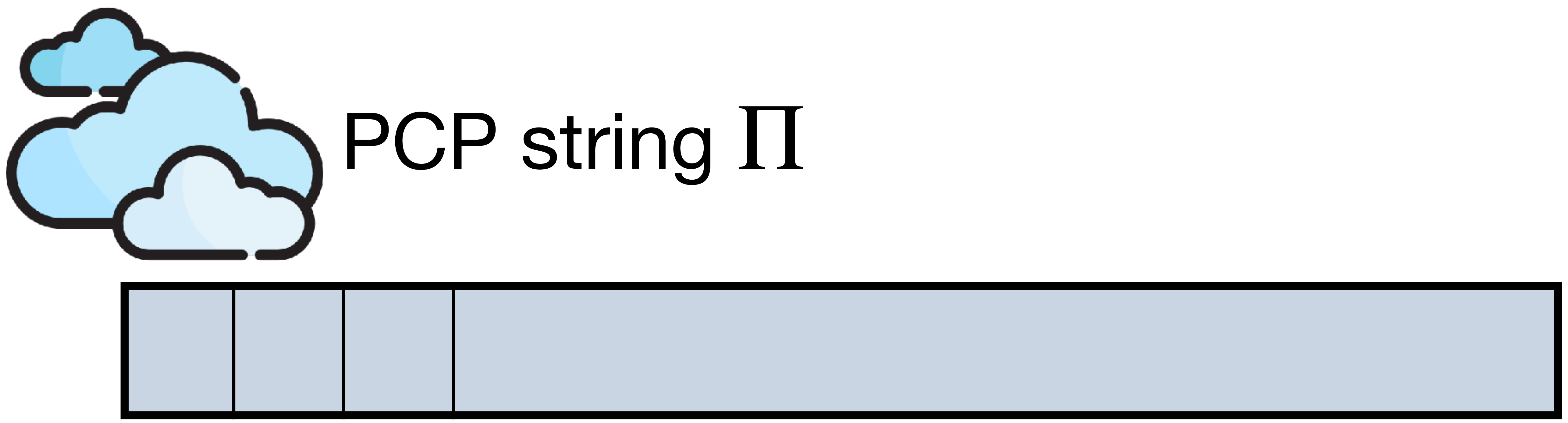


Probabilistically Checkable Proofs

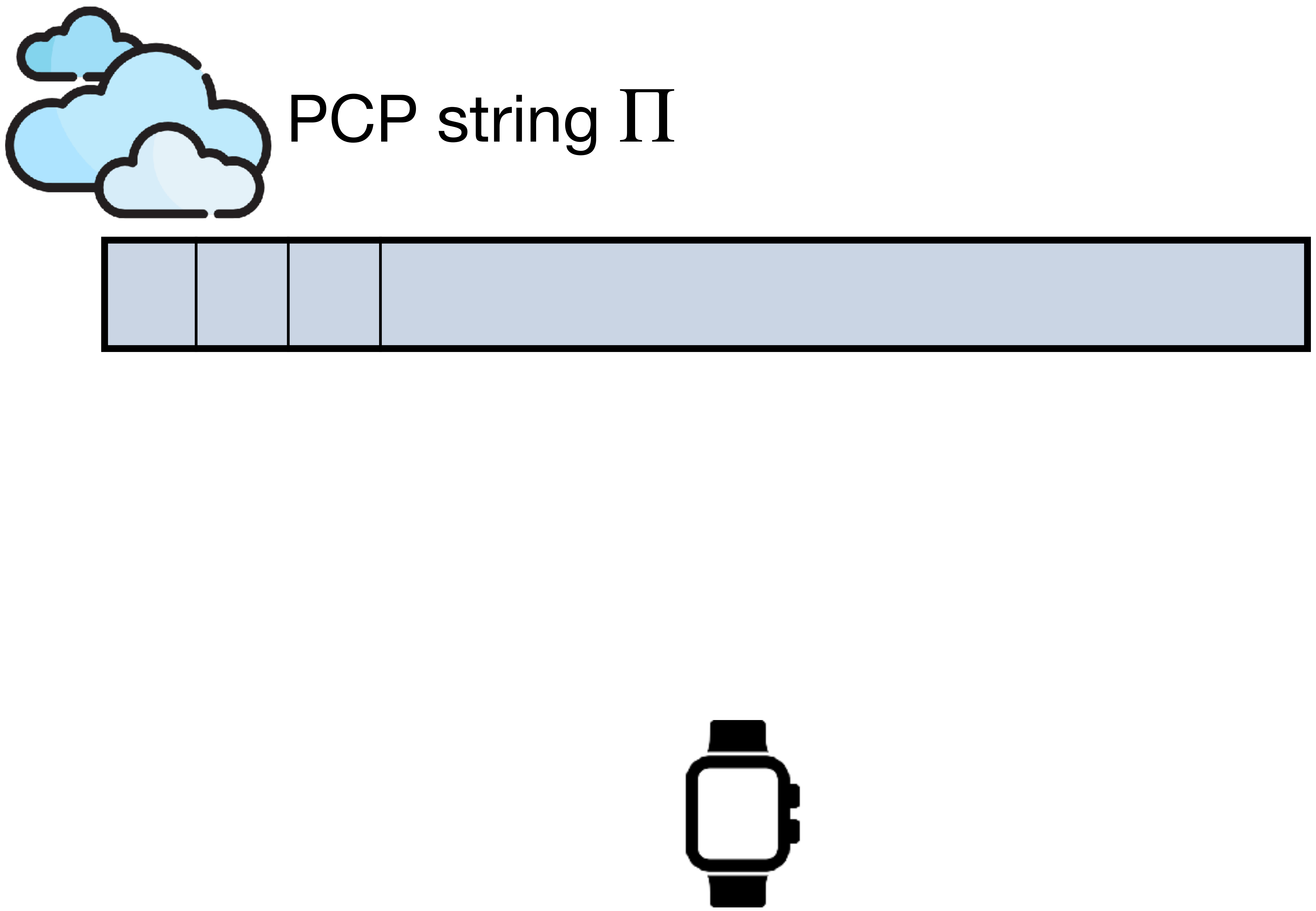
Probabilistically Checkable Proofs

PCP string Π

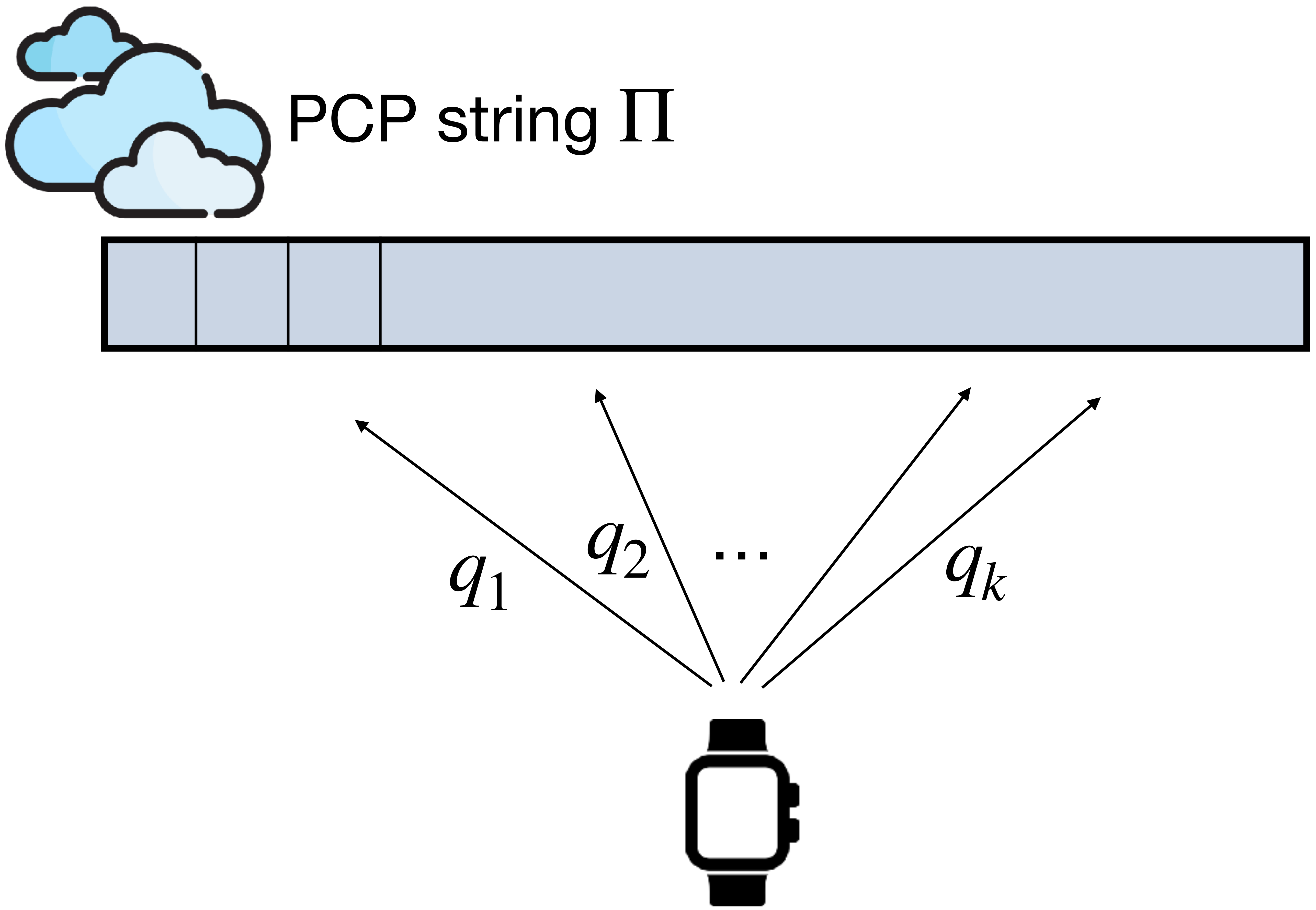
Probabilistically Checkable Proofs



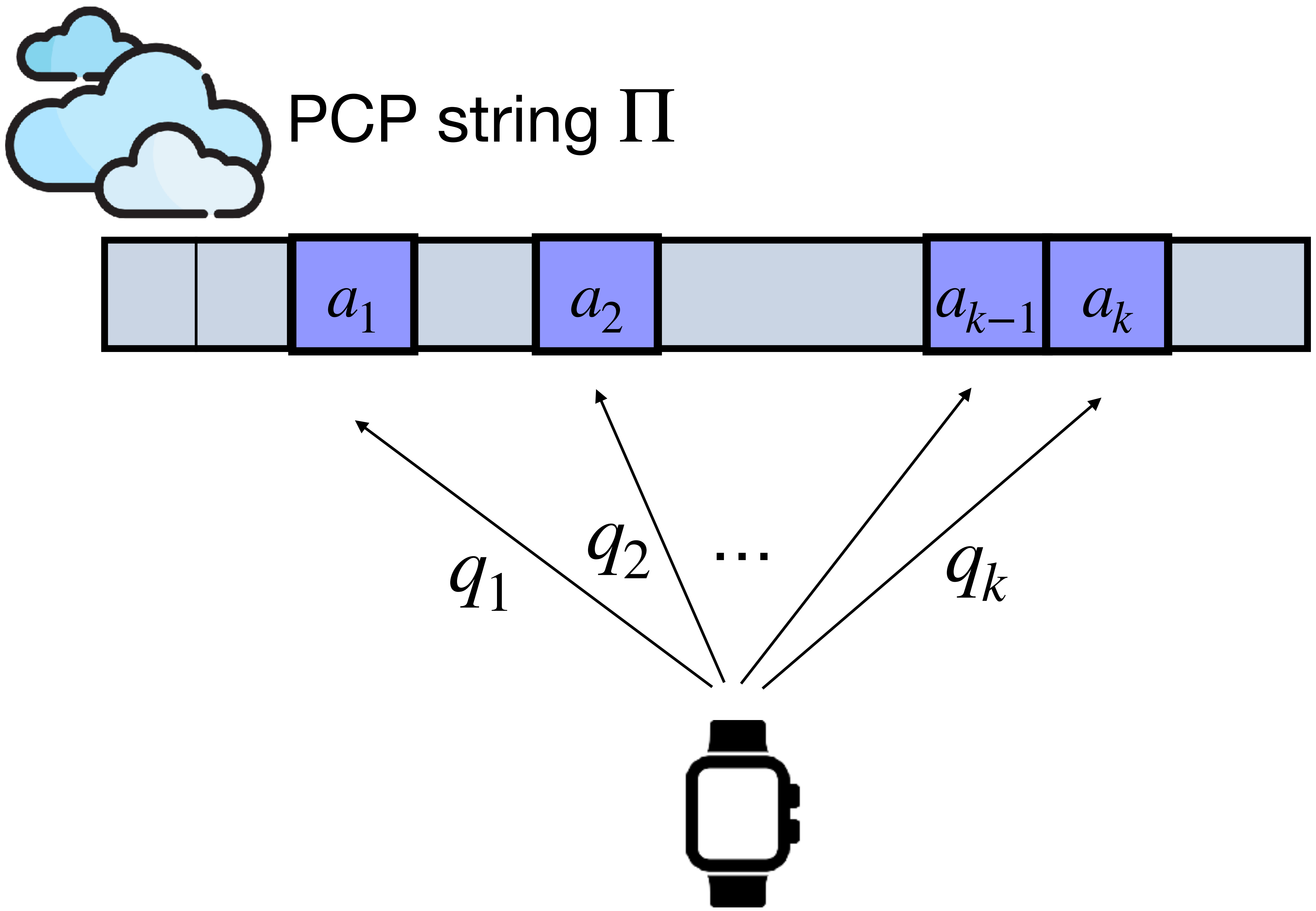
Probabilistically Checkable Proofs



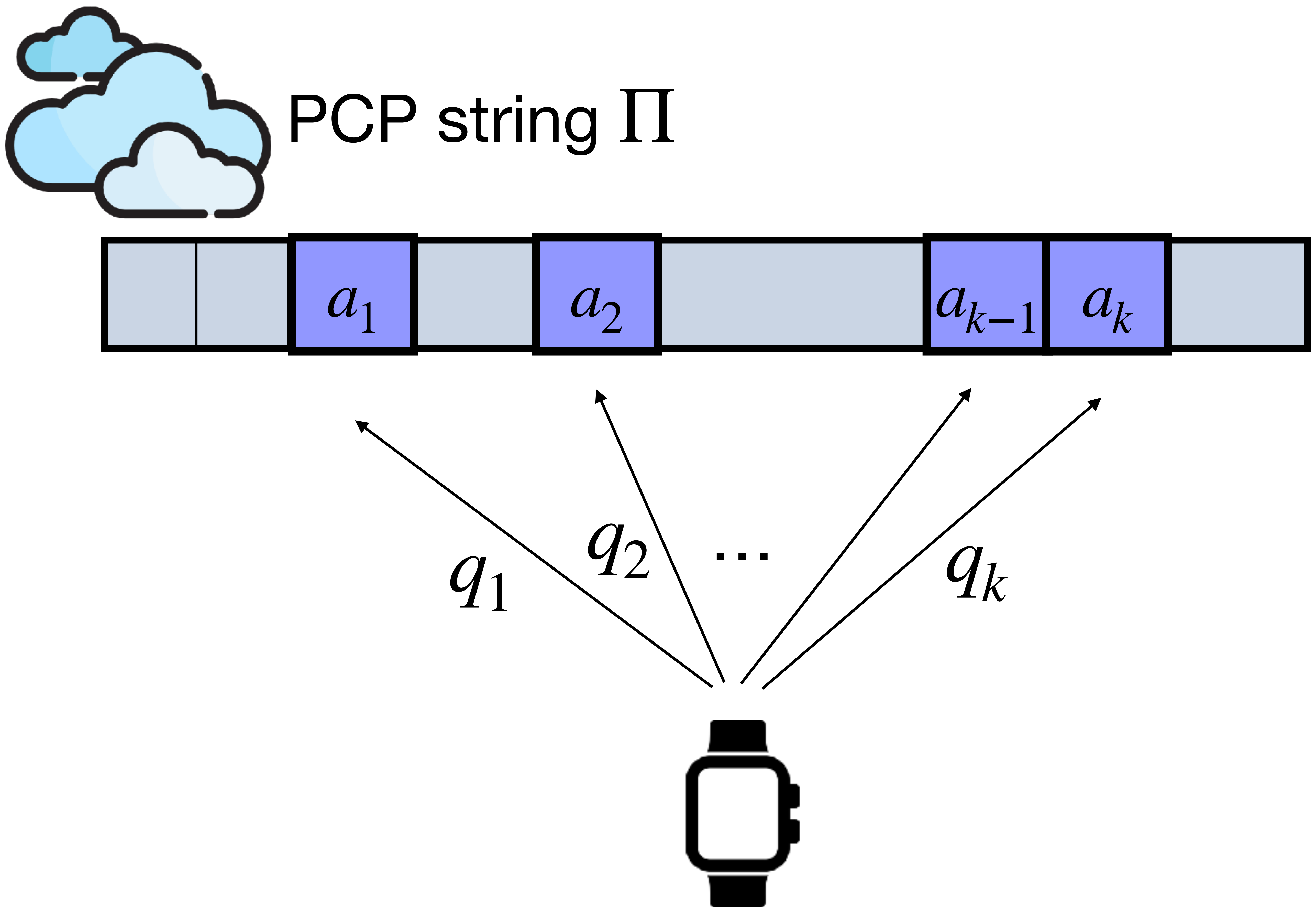
Probabilistically Checkable Proofs



Probabilistically Checkable Proofs

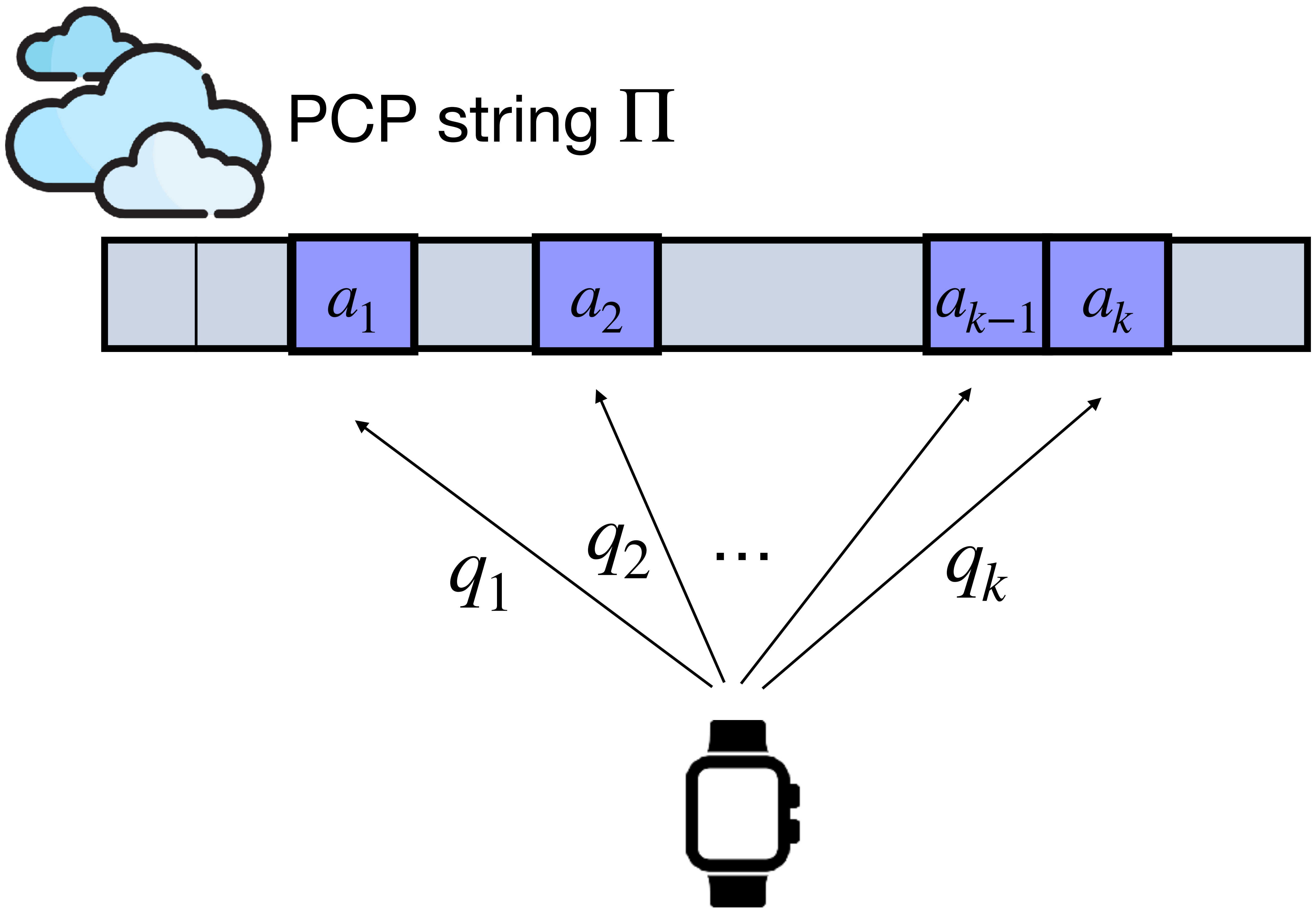


Probabilistically Checkable Proofs



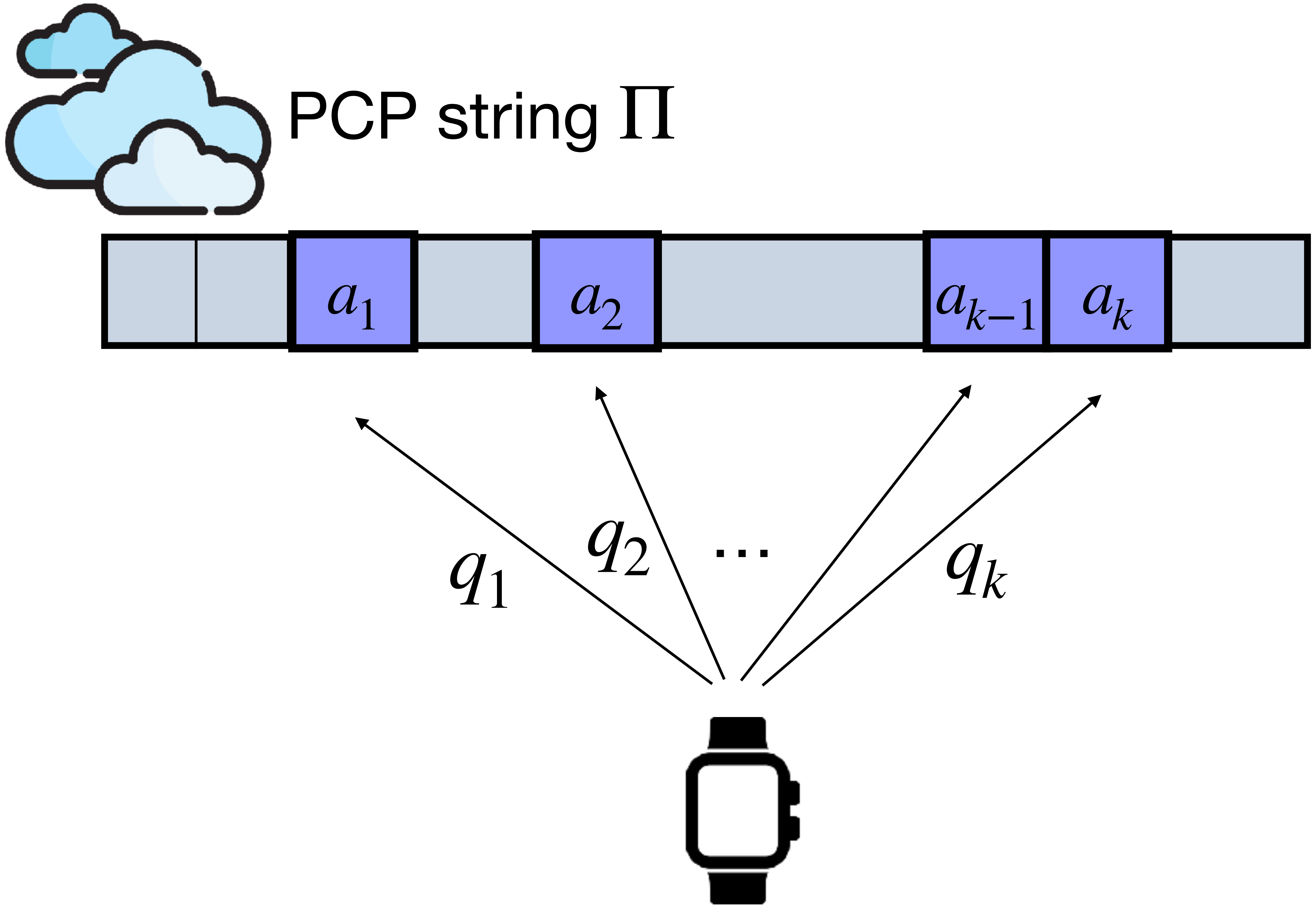
- **Locality:** $k \ll |w|$

Probabilistically Checkable Proofs



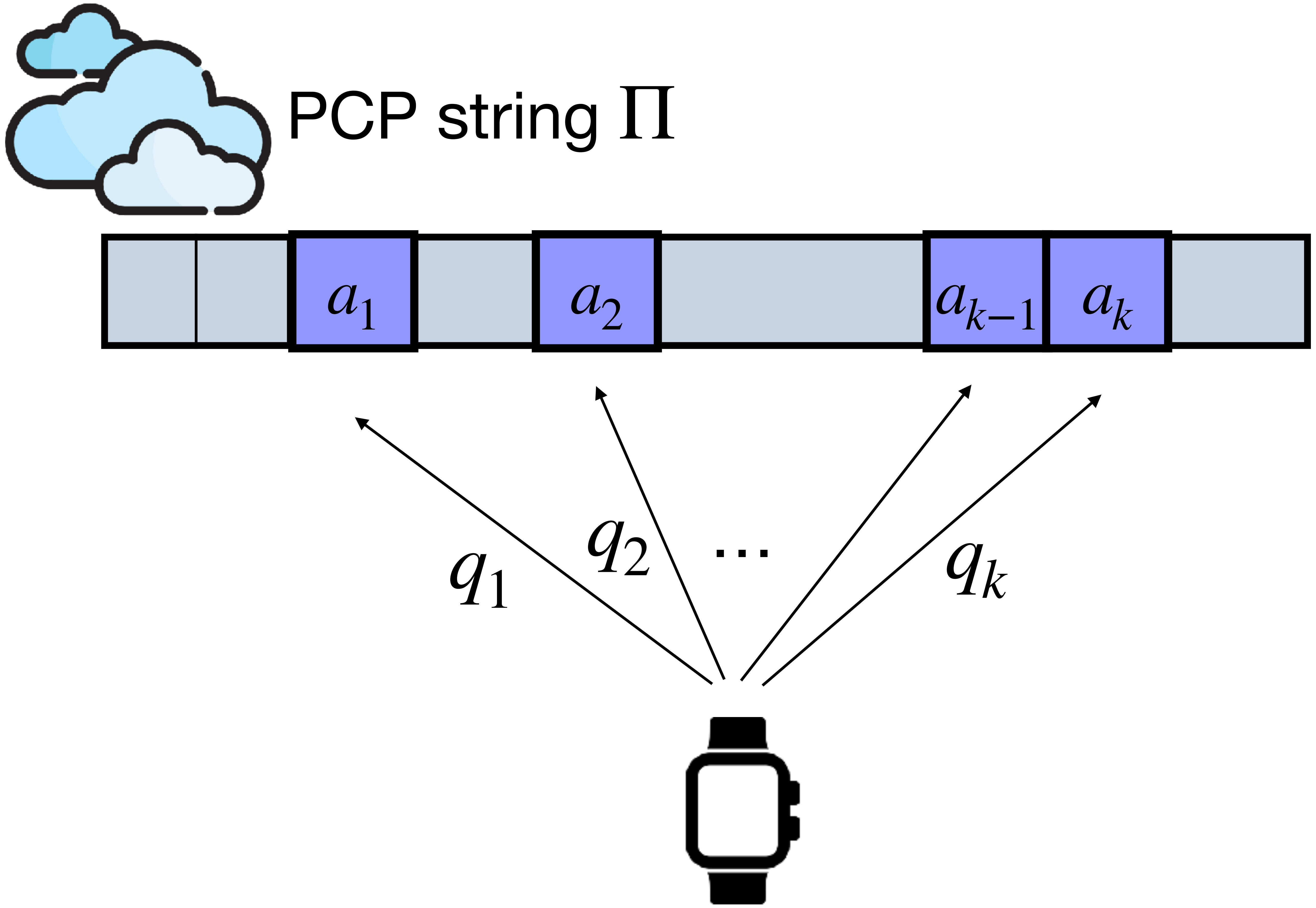
- **Locality:** $k \ll |w|$
- **Correctness:** For honest Π, V always accepts.

Probabilistically Checkable Proofs



- **Locality:** $k \ll |w|$
- **Correctness:** For honest Π, V always accepts.
- **Soundness:** For $x \notin \mathscr{L}$, $\Pr_Q[V^{\Pi}(x, Q) = 1] \leq 1/\text{poly(n)}$

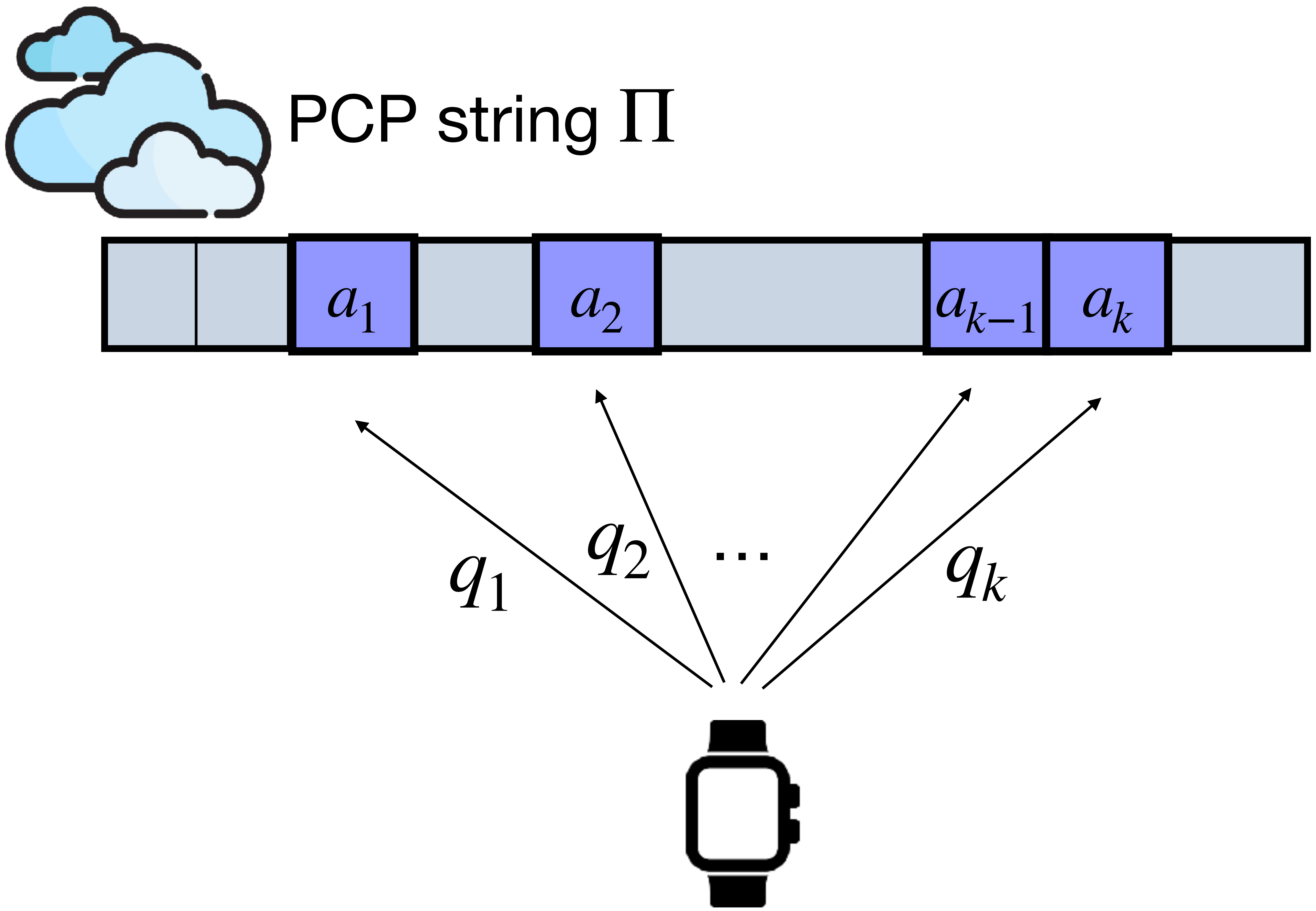
Probabilistically Checkable Proofs



- **Locality:** $k \ll |w|$
- **Correctness:** For honest Π, V always accepts.
- **Soundness:** For $x \notin \mathscr{L}$, $\Pr_Q[V^{\Pi}(x, Q) = 1] \leq 1/\text{poly(n)}$

Theorem [ALMSS '92]. There exists a PCP of length poly(n) and locality polylog(n)

PCP to SNARG?

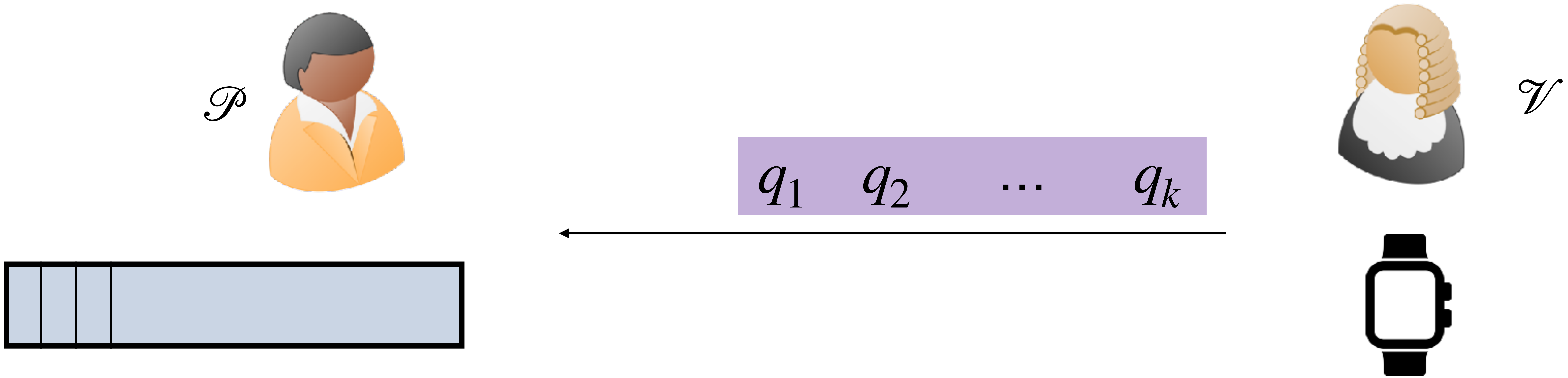


- **Locality:** $k \ll |w|$
- **Correctness:** For honest Π, V always accepts.
- **Soundness**: For $x \notin \mathscr{L}$, $\Pr_Q[V^{\Pi}(x, Q) = 1] \leq 1/\text{poly(n)}$

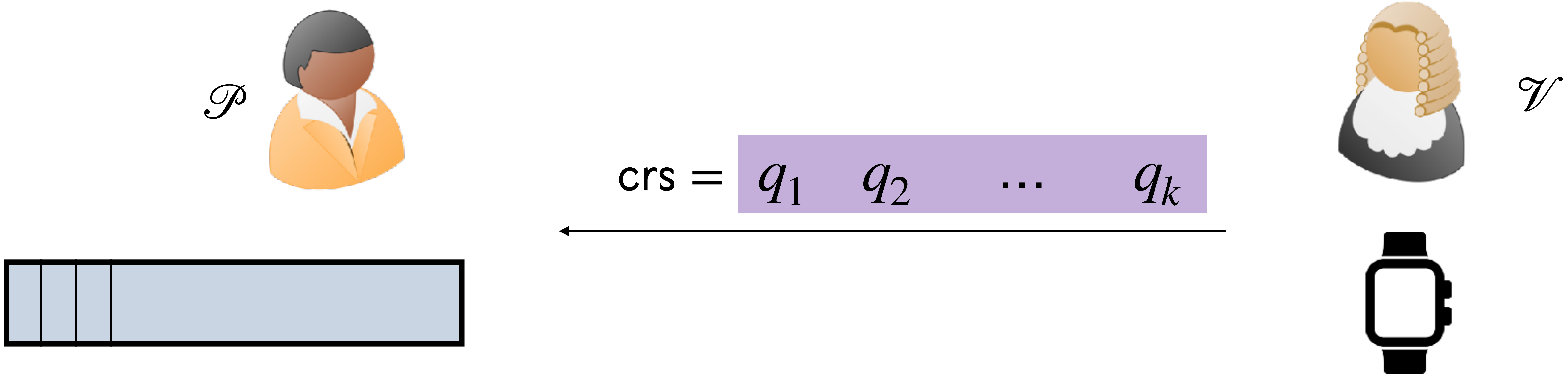
Theorem [ALMSS '92]. There exists a PCP of length poly(n) and locality polylog(n)

PCP to SNARG?

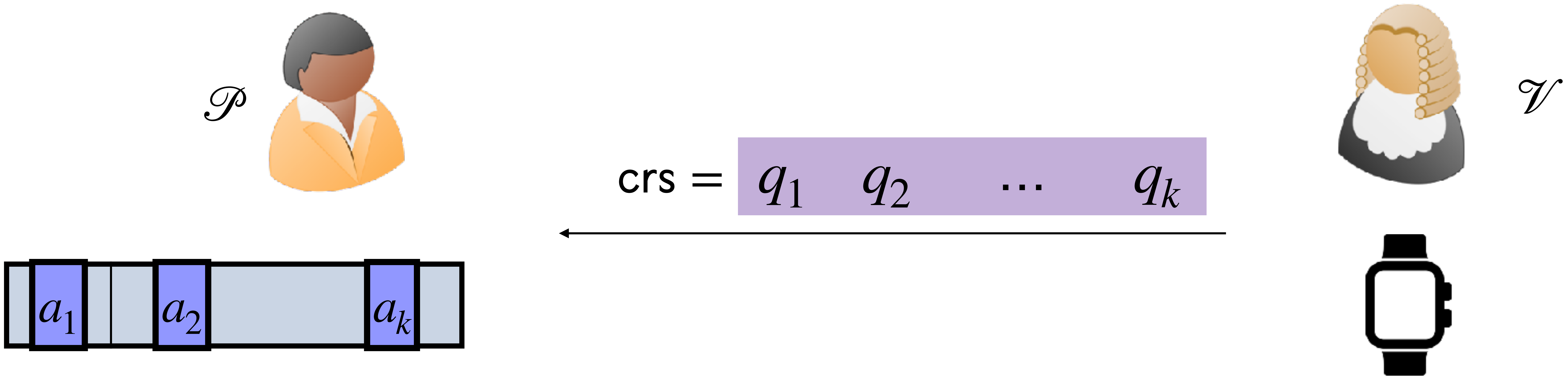
PCP to SNARG?



PCP to SNARG?

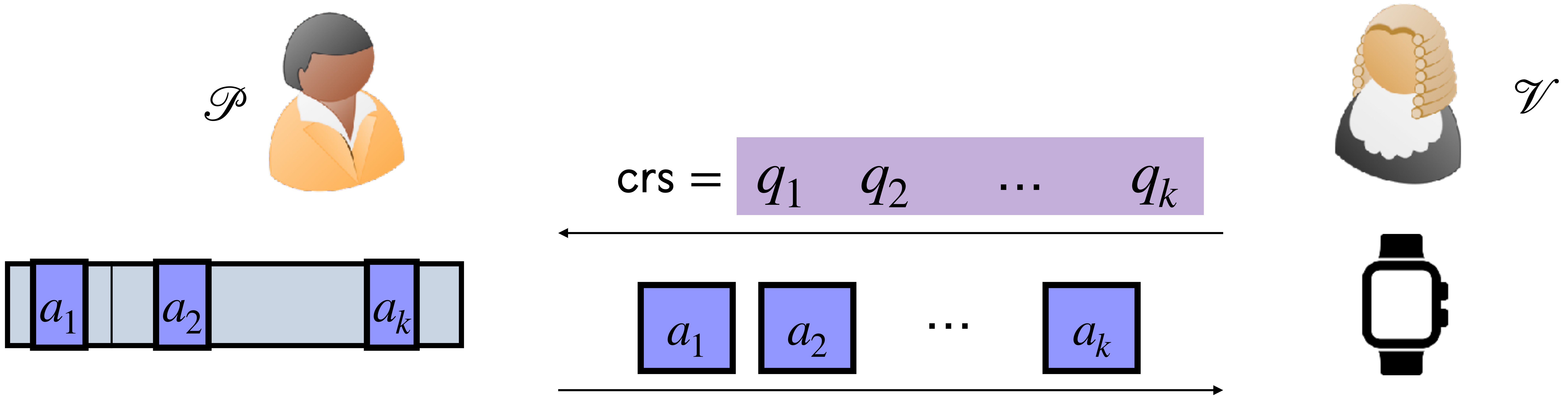


PCP to SNARG?

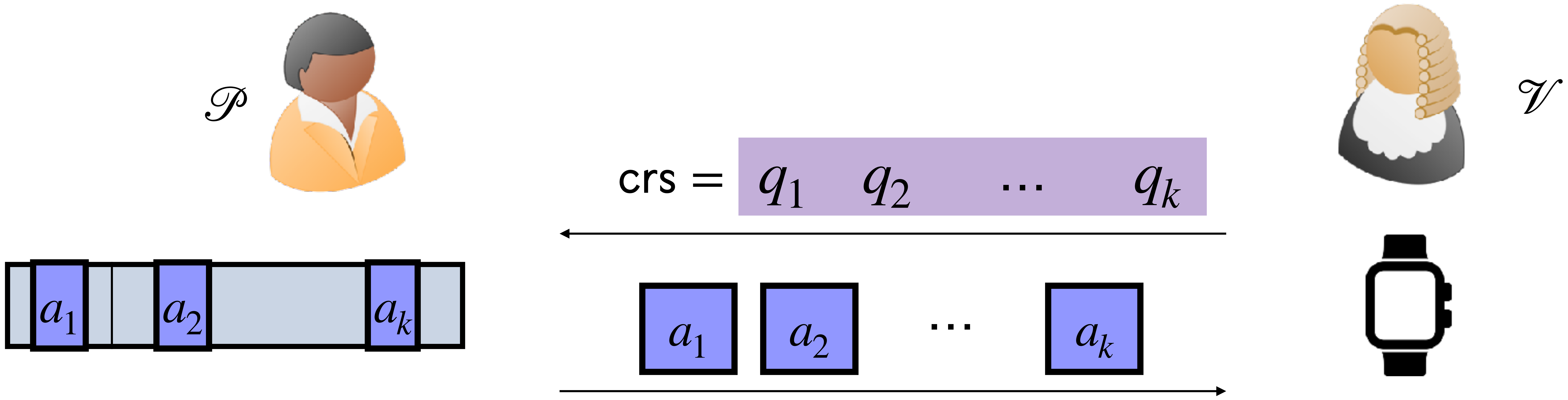


$\mathscr{P}$
$\mathscr{V}$
$\text{crs} = q_1 \quad q_2 \quad \cdots \quad q_k$
a_1
a_2
a_k

PCP to SNARG?

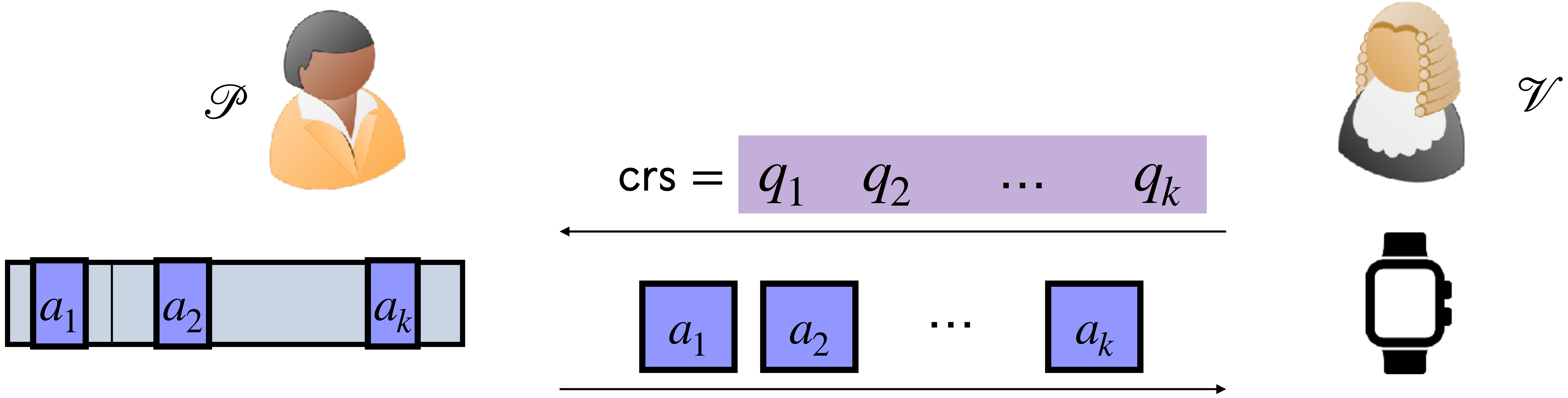


PCP to SNARG?



✓ Succinct and Correct

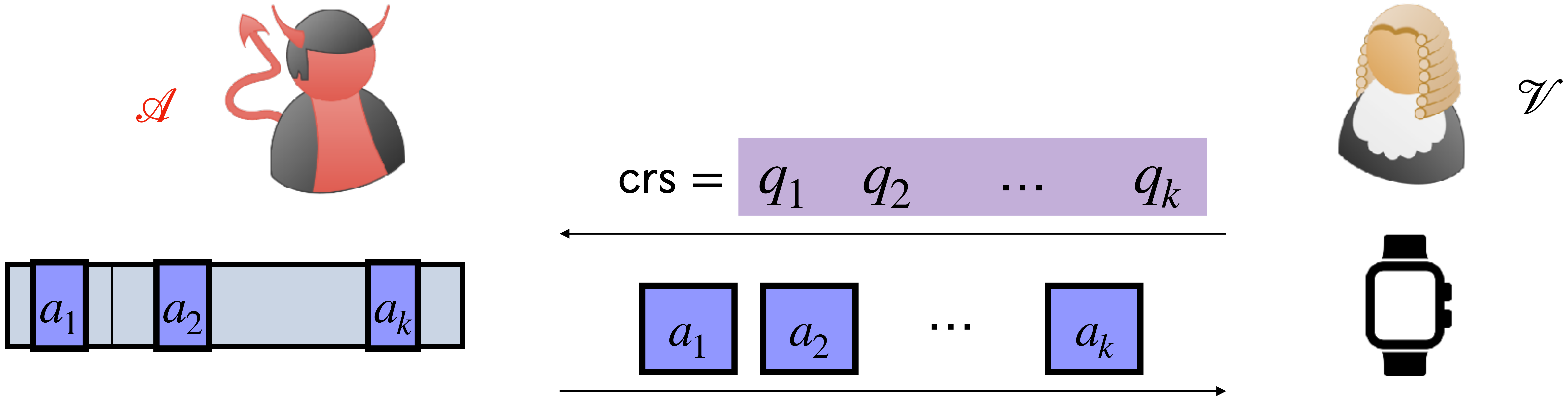
PCP to SNARG?



✓ Succinct and Correct

✗ **Not sound**

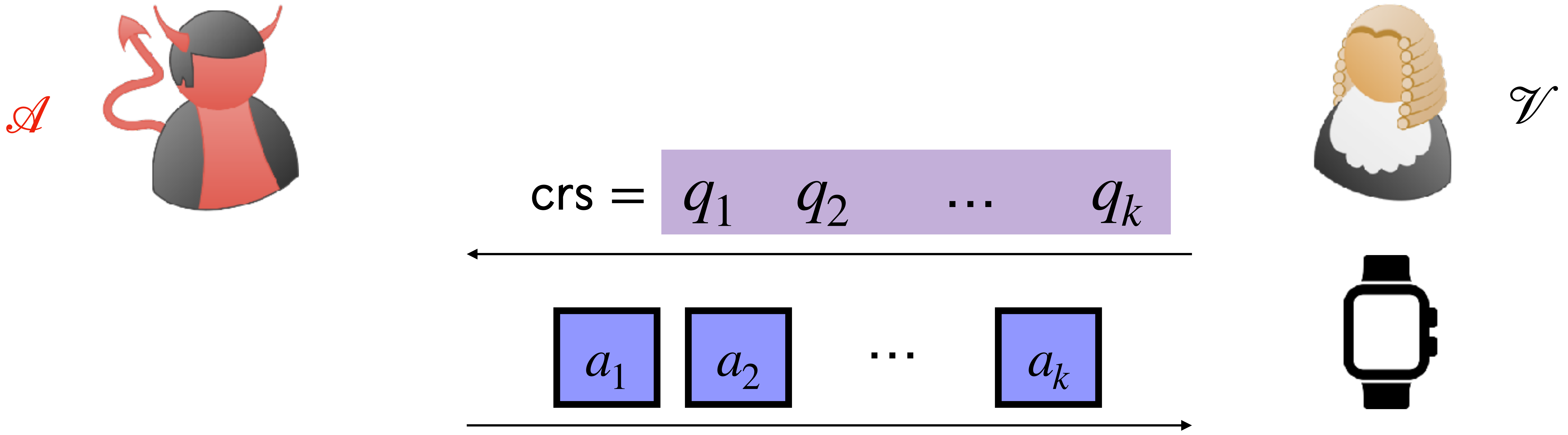
PCP to SNARG?



✓ Succinct and Correct

✗ **Not sound**

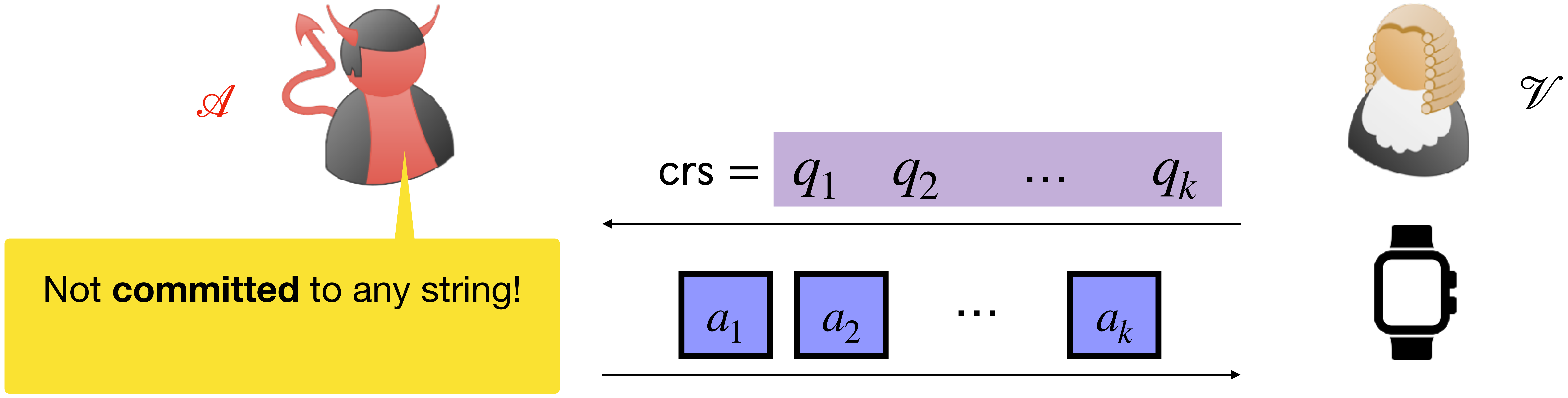
PCP to SNARG?



✓ Succinct and Correct

✗ **Not sound**

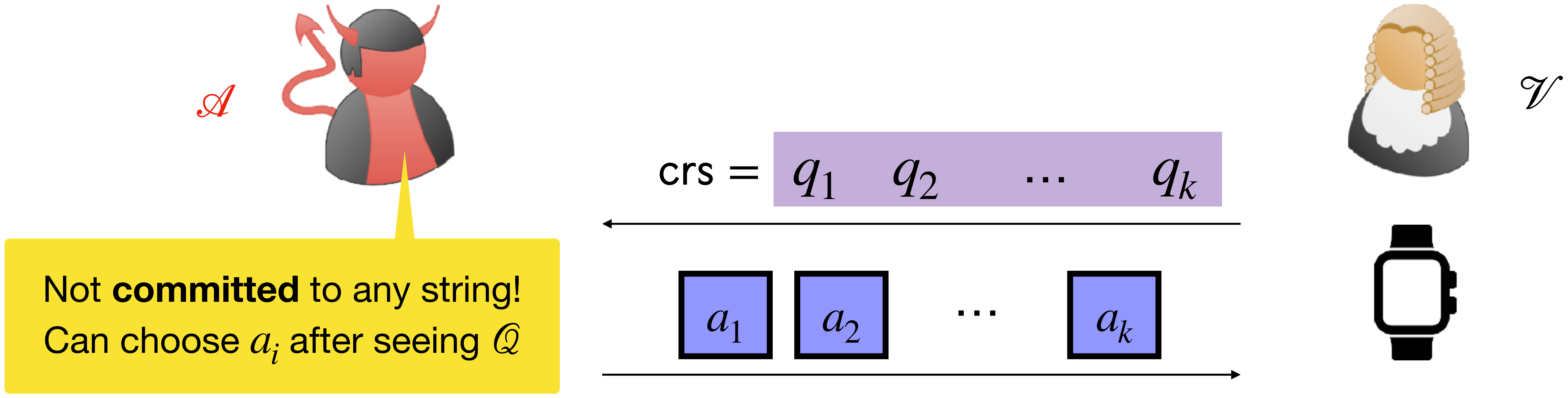
PCP to SNARG?



✓ Succinct and Correct

✗ **Not sound**

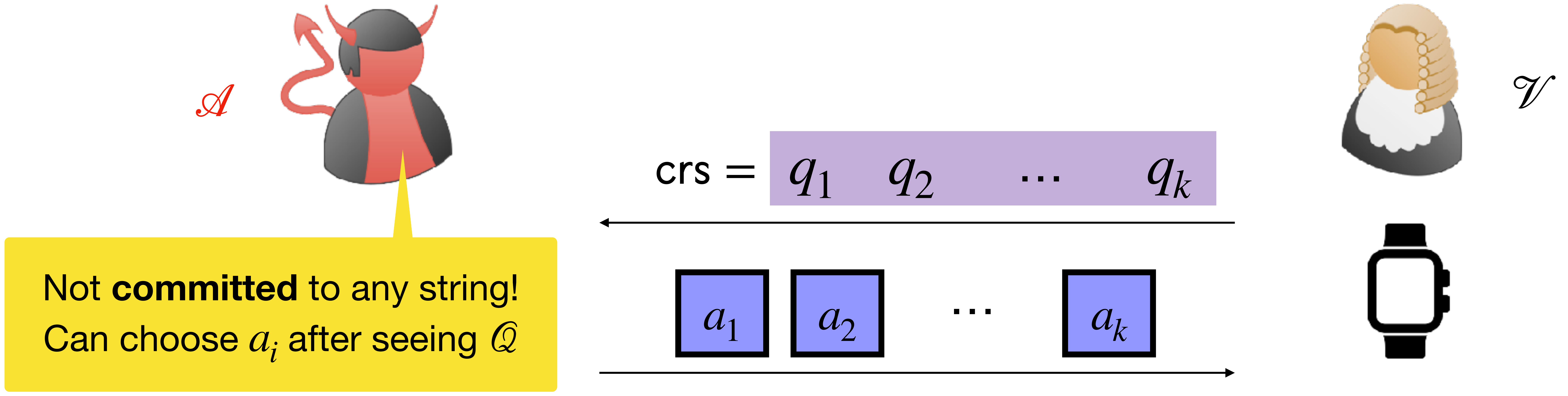
PCP to SNARG?



✓ Succinct and Correct

✗ **Not sound**

PCP to SNARG?



✓ Succinct and Correct

✗ **Not sound**

Need some cryptography in this compiler to restrict $\mathscr{A}$!

PCP to SNARG?

PCP to SNARG?

- **Recipe #1:** [Kilian ’92, Micali ’94] (not in talk)

PCP to SNARG?

- **Recipe #1:** [Kilian '92, Micali '94] (not in talk)

PCP +  Commitments

PCP to SNARG?

- **Recipe #1:** [Kilian ’92, Micali ’94] (not in talk)

PCP +  Commitments = **Interactive** Argument

PCP to SNARG?

- **Recipe #1:** [Kilian '92, Micali '94] (not in talk)

PCP +  Commitments

PCP to SNARG?

- **Recipe #1:** [Kilian ’92, Micali ’94] (not in talk)

PCP +  Commitments + Fiat-Shamir

PCP to SNARG?

- **Recipe #1:** [Kilian '92, Micali '94] (not in talk)

PCP +  Commitments + Fiat-Shamir

PCP to SNARG?

- **Recipe #1:** [Kilian ’92, Micali ’94] (not in talk)

PCP + 💍 Commitments + Fiat-Shamir

- **Recipe #2:** [BMW ’98, KRR ’14]

PCP to SNARG?

- **Recipe #1:** [Kilian ’92, Micali ’94] (not in talk)

PCP + Commitments + Fiat-Shamir

- **Recipe #2:** [BMW ’98, KRR ’14]

“Non-Signaling” PCP + FHE

PCP to SNARG?

- **Recipe #1:** [Kilian '92, Micali '94] (not in talk)

PCP + Commitments + Fiat-Shamir

- **Recipe #2:** [BMW '98, KRR '14]

"Non-Signaling" PCP + FHE

PCP to SNARG?

- **Recipe #1:** [Kilian ’92, Micali ’94] (not in talk)

PCP + Commitments + Fiat-Shamir

- **Recipe #2:** [BMW ’98, KRR ’14] ←

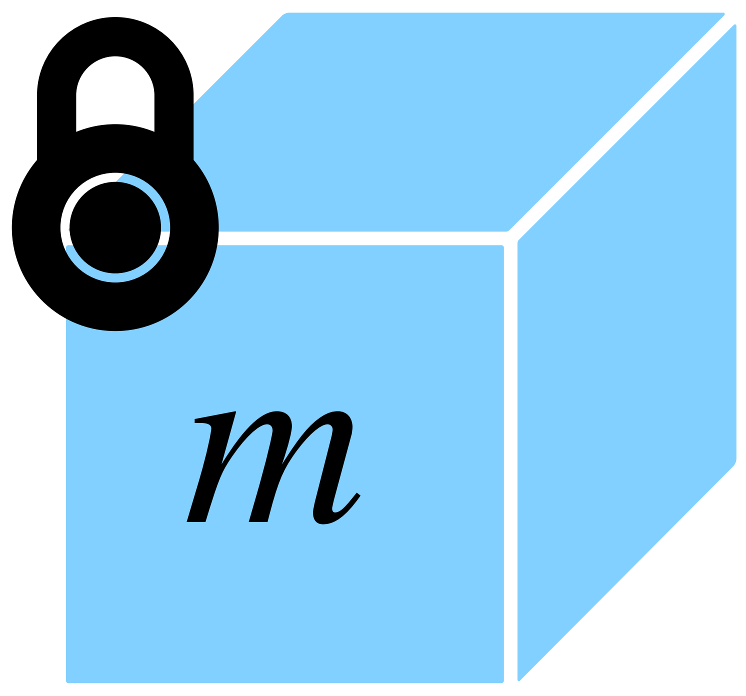
“Non-Signaling” PCP + FHE

Fully Homomorphic Encryption

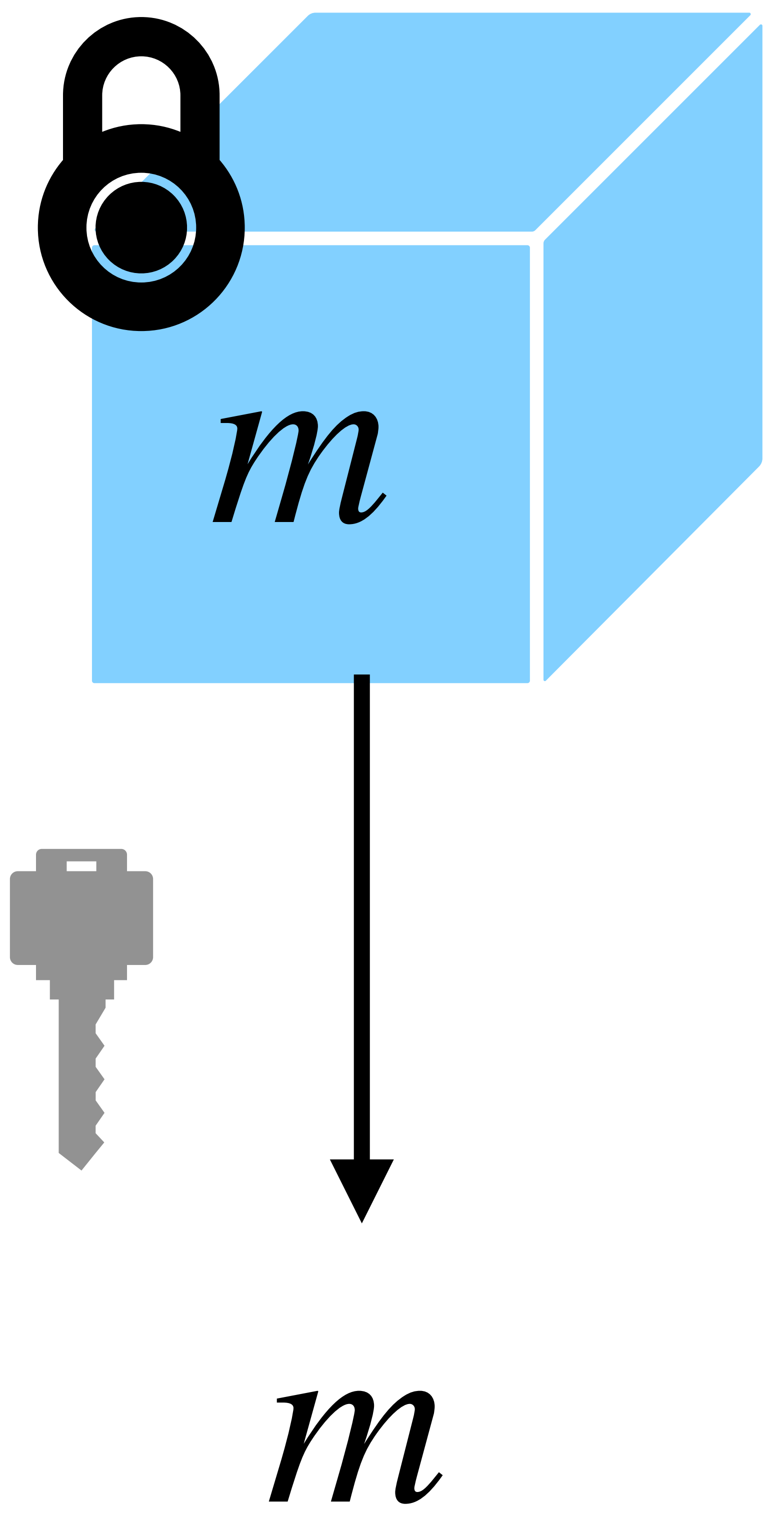
Fully Homomorphic Encryption

m

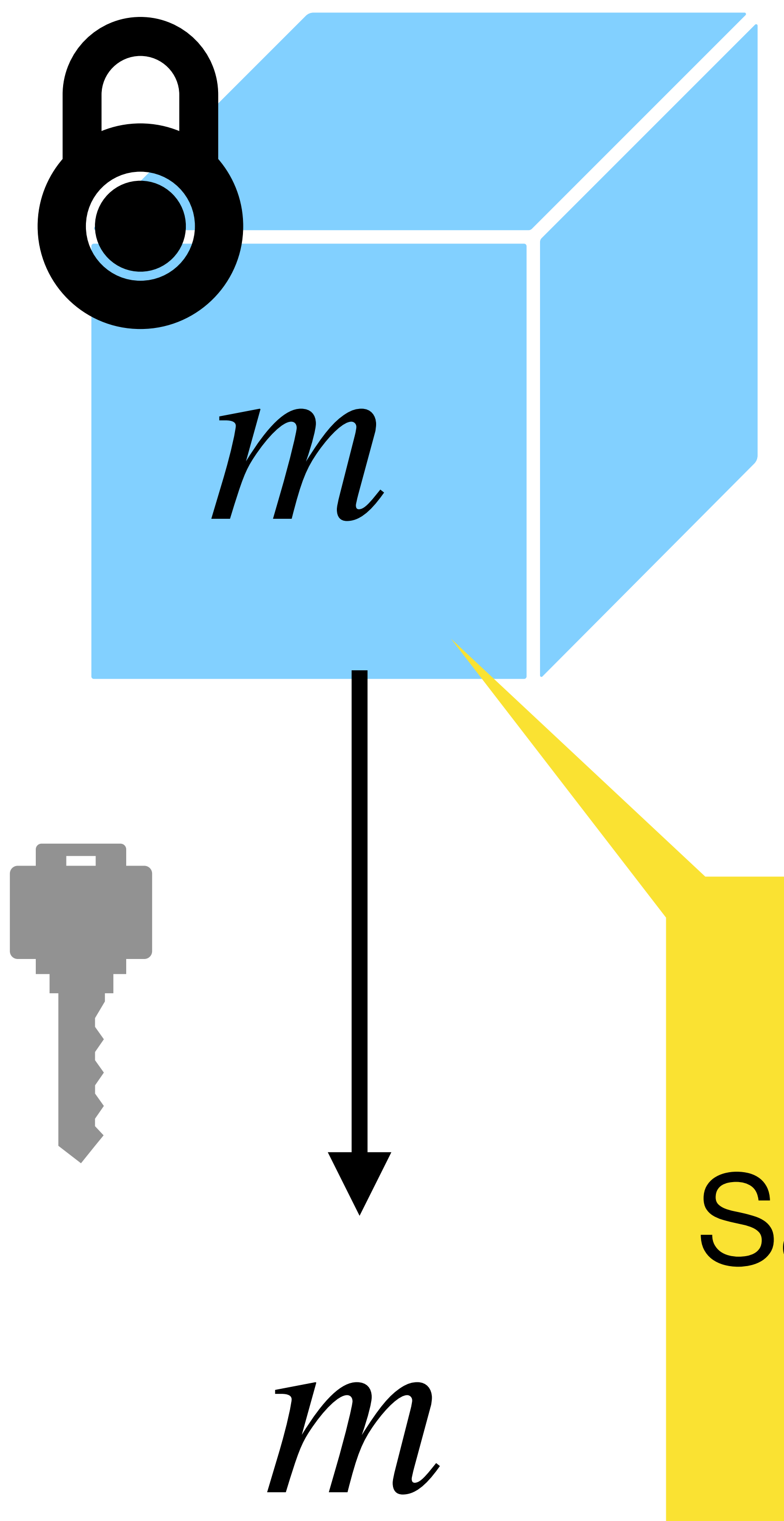
Fully Homomorphic Encryption



Fully Homomorphic Encryption

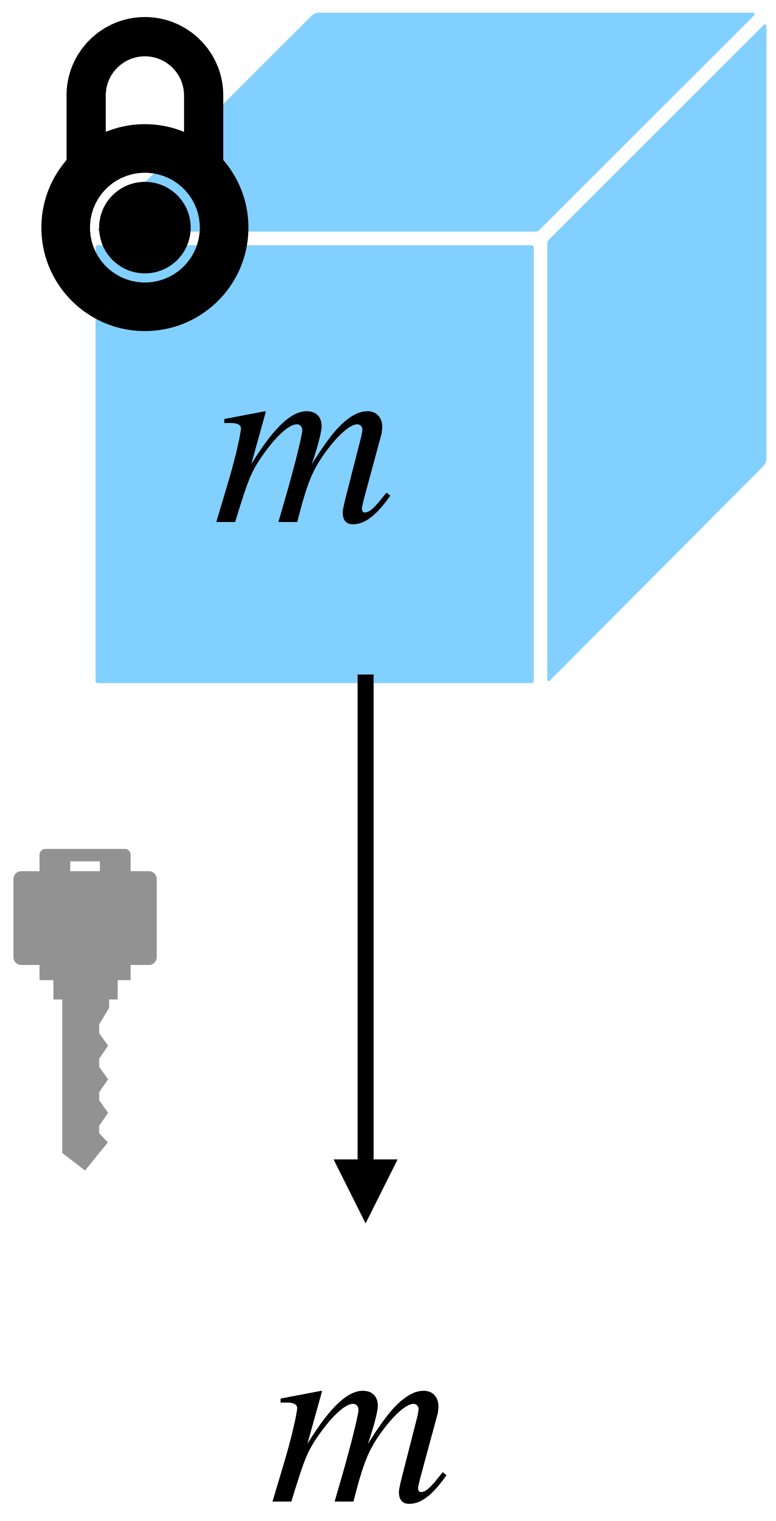


Fully Homomorphic Encryption

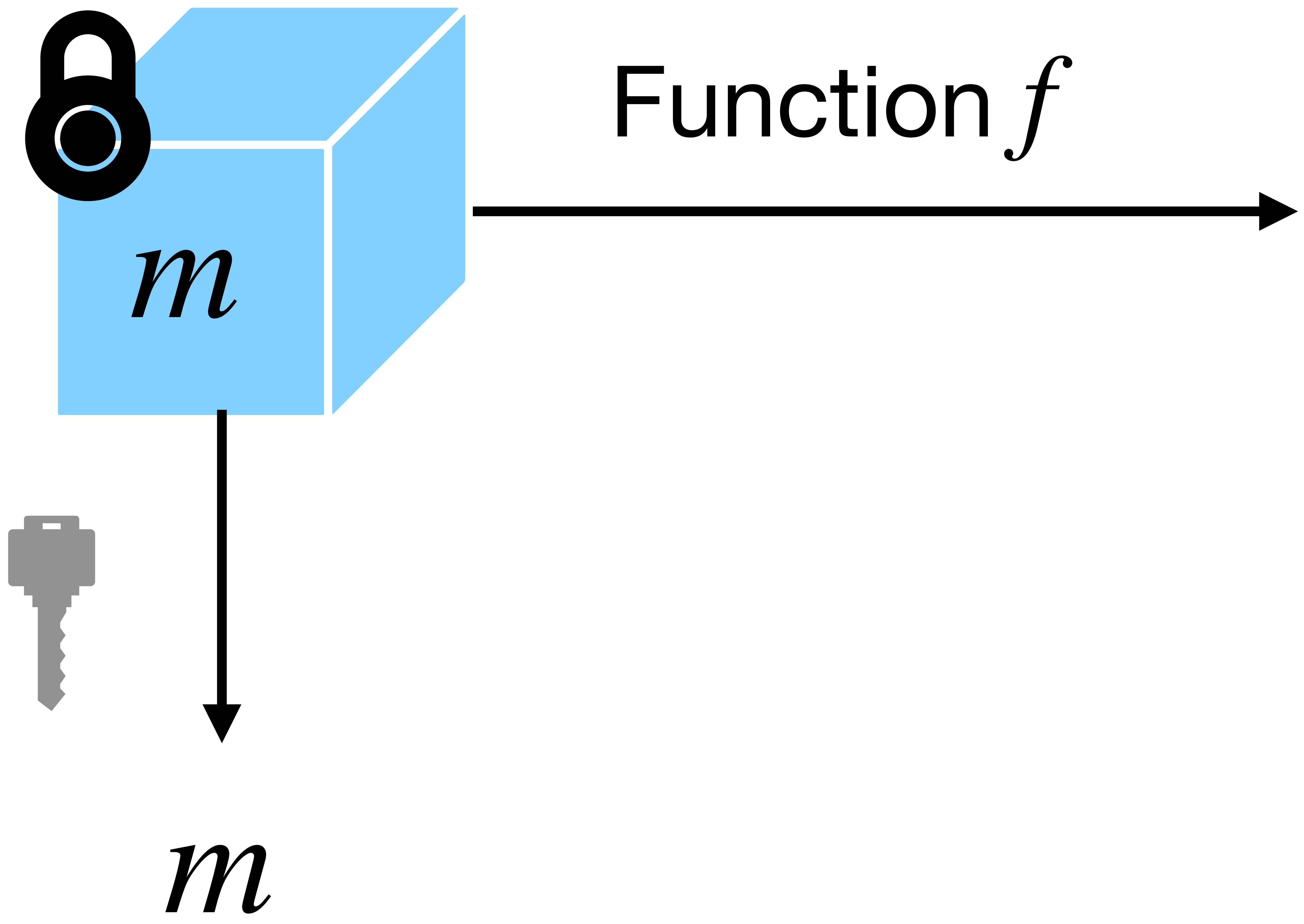


Satisfies usual properties of an encryption scheme

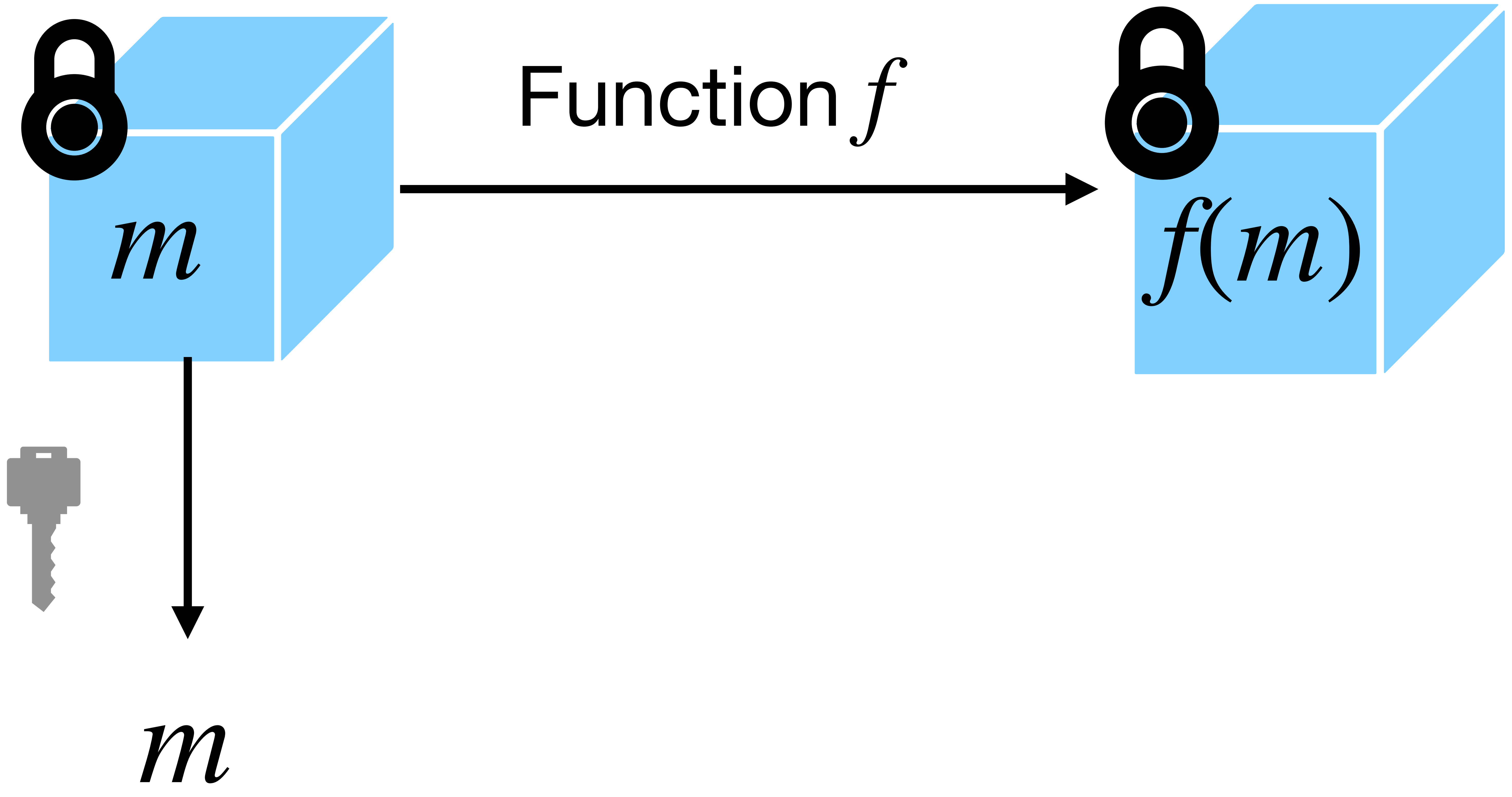
Fully Homomorphic Encryption



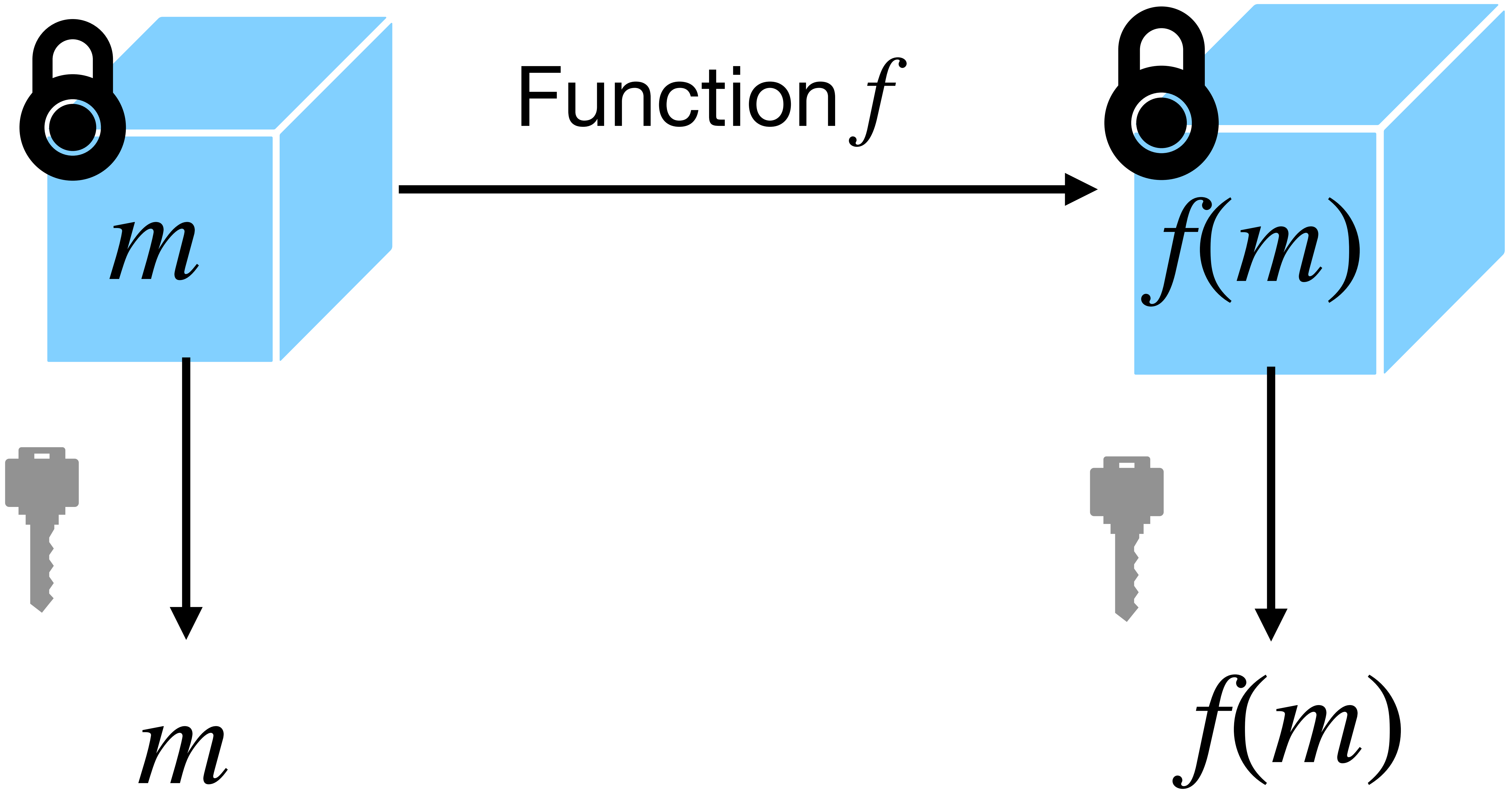
Fully Homomorphic Encryption



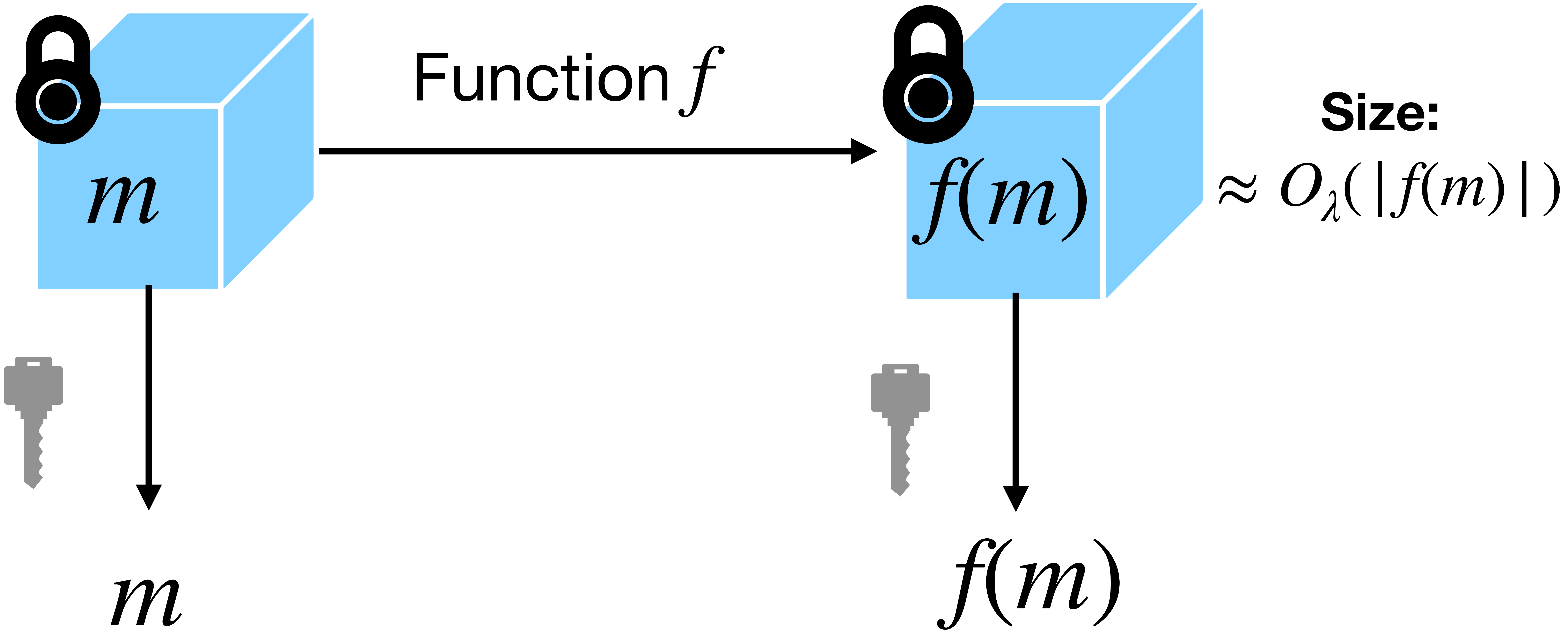
Fully Homomorphic Encryption



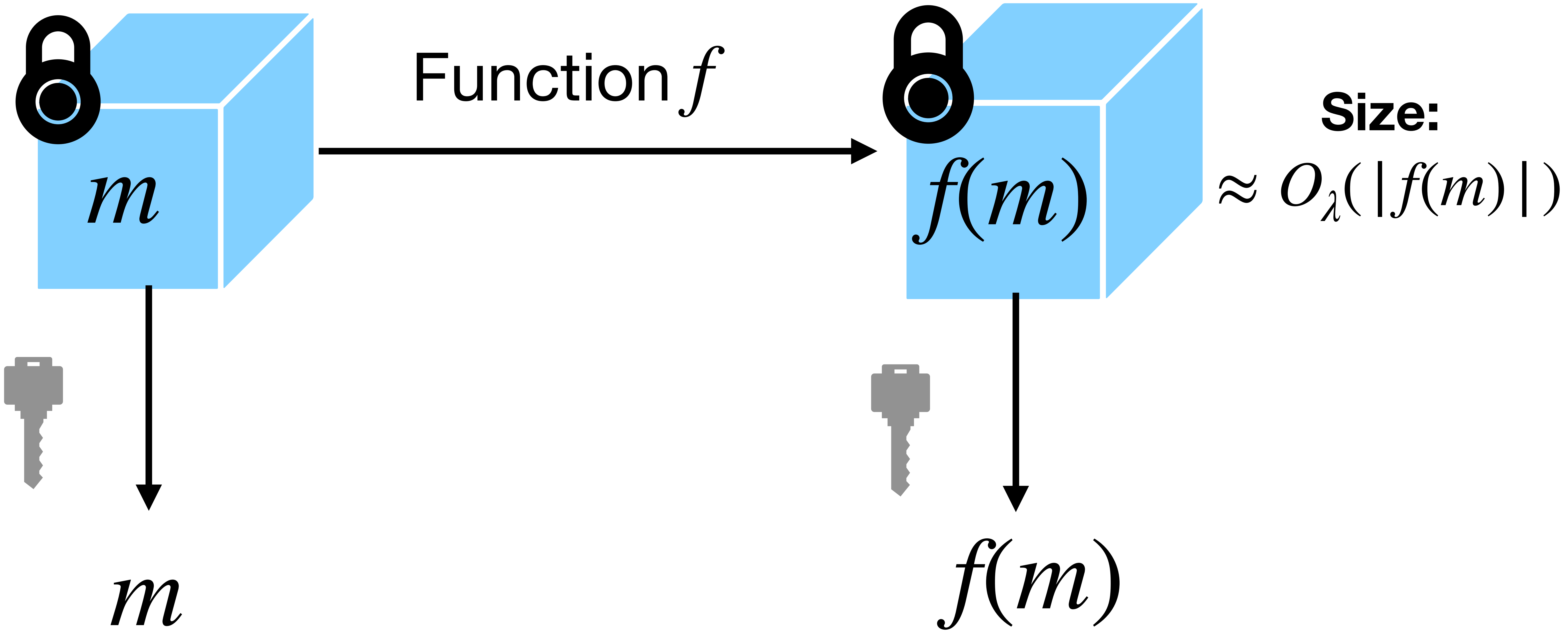
Fully Homomorphic Encryption



Fully Homomorphic Encryption



Fully Homomorphic Encryption



Theorem [G09, BV11]. Assuming polynomial hardness LWE, there exist (leveled) FHE.

KRR14 Construction

Based on [Biehl-Meyer-Wetzel ’98]

$\mathscr{P}$

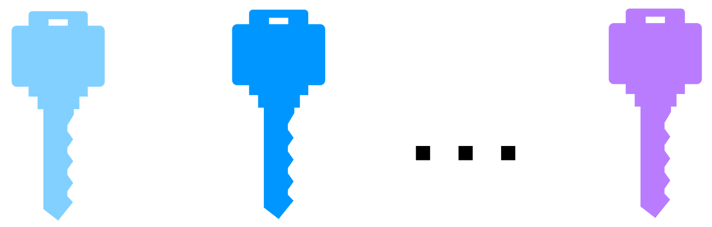
$\mathscr{V}$

KRR14 Construction

Based on [Biehl-Meyer-Wetzel '98]

$\mathscr{P}$

$\mathscr{V}$

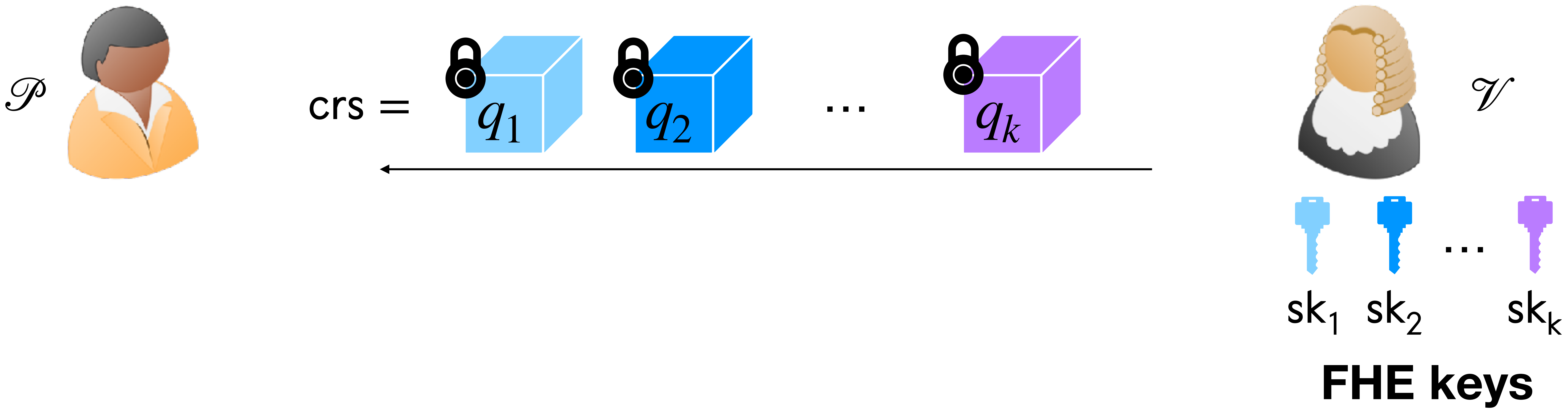


sk_1 sk_2 … sk_k

FHE keys

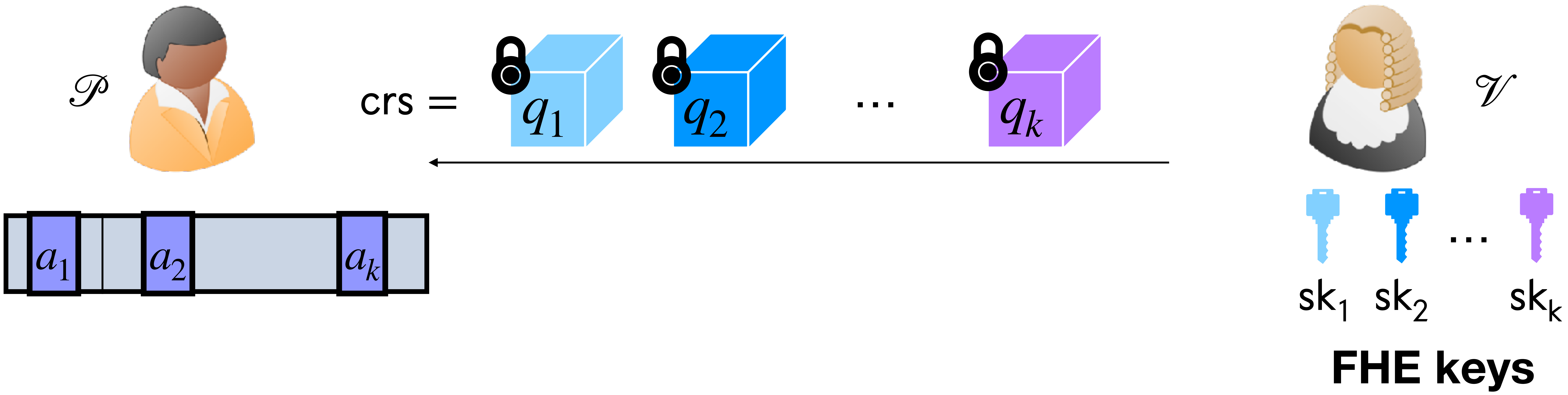
KRR14 Construction

Based on [Biehl-Meyer-Wetzel '98]



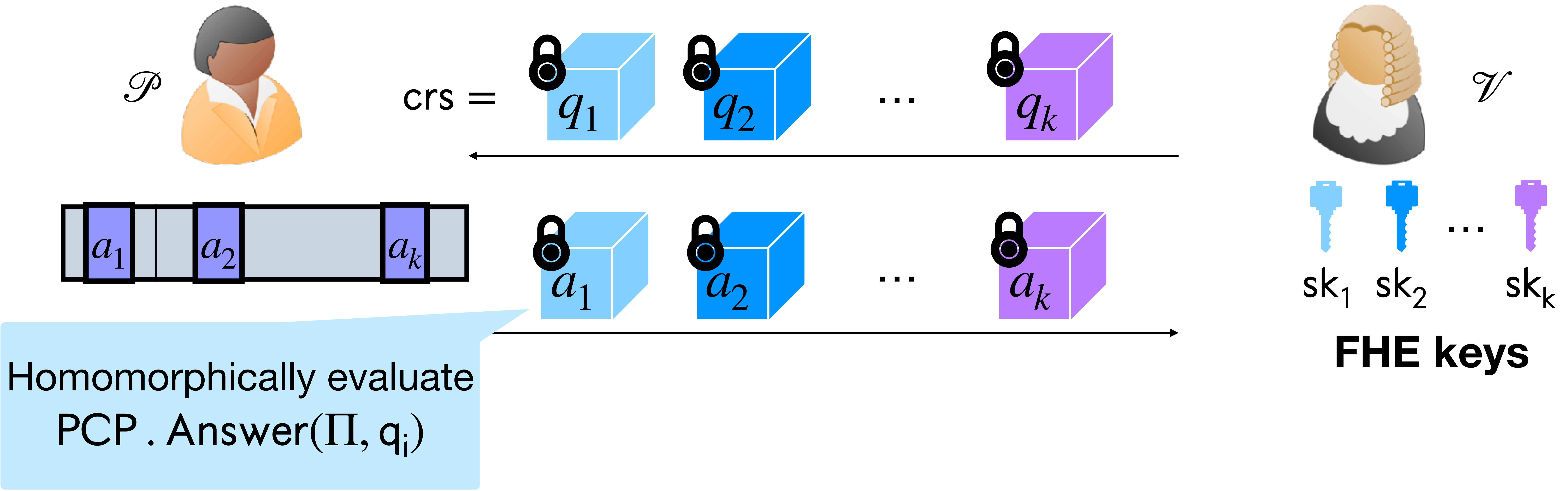
KRR14 Construction

Based on [Biehl-Meyer-Wetzel '98]



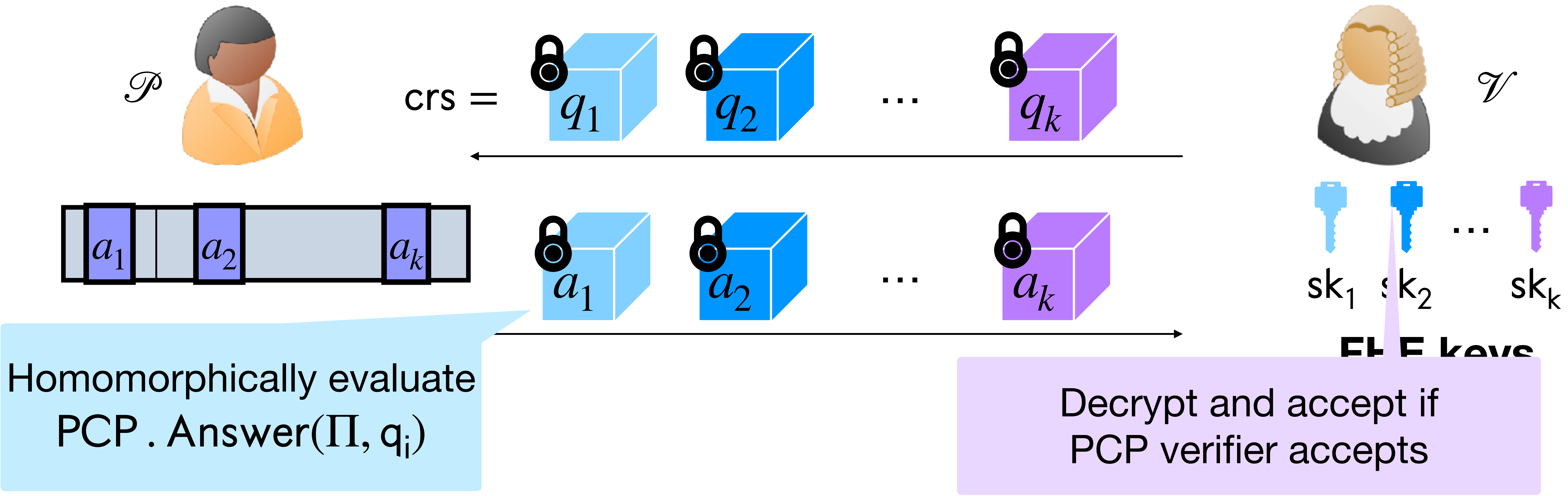
KRR14 Construction

Based on [Biehl-Meyer-Wetzel ’98]



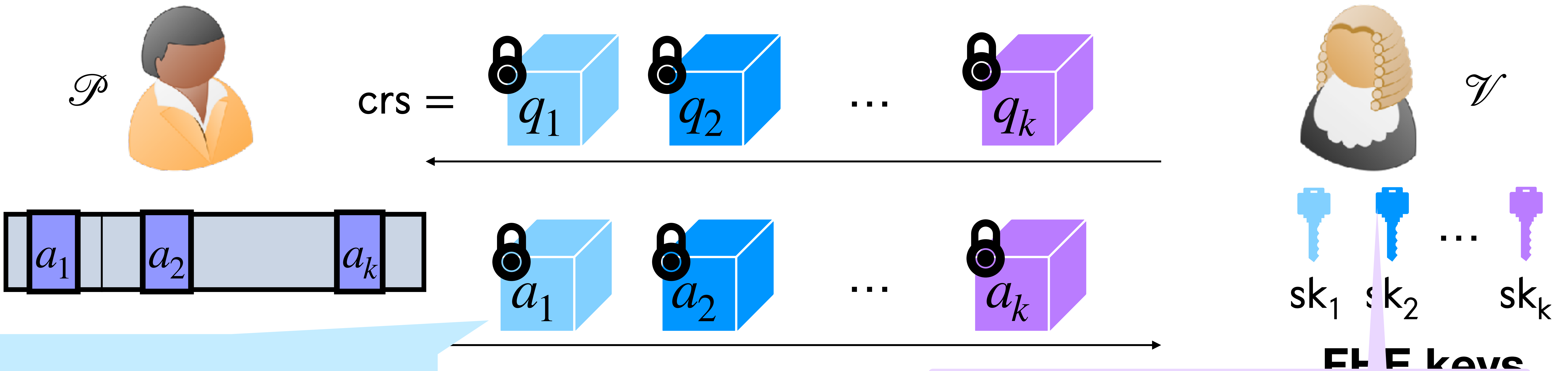
KRR14 Construction

Based on [Biehl-Meyer-Wetzel '98]



KRR14 Construction

Based on [Biehl-Meyer-Wetzel ’98]



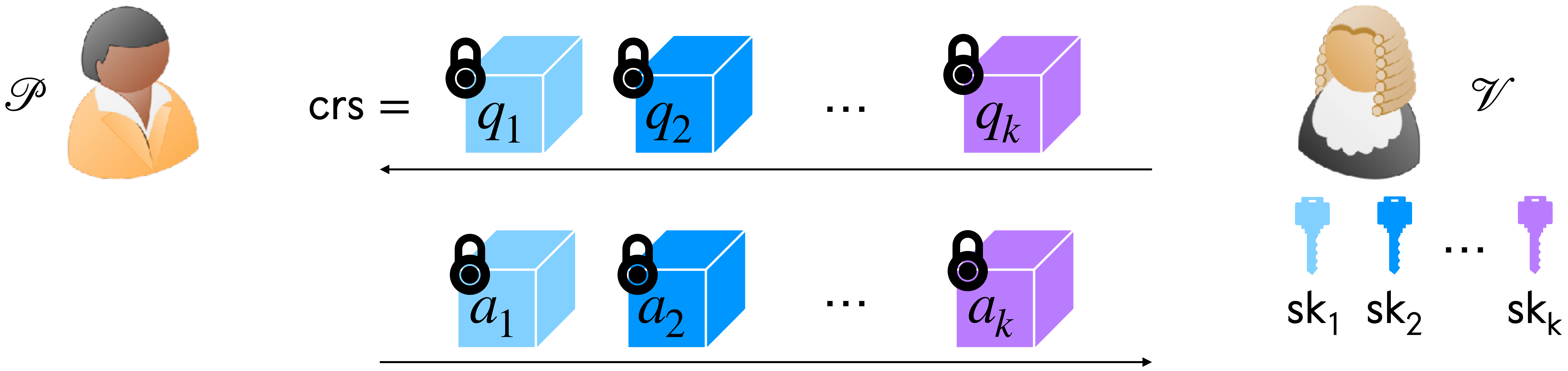
Homomorphically evaluate $\mathsf{PCP.Answer}(\Pi, q_i)$

Decrypt and accept if PCP verifier accepts

Intuition: How can $\mathscr{P}$ **cheat** if he doesn’t know what is being queried (FHE security)?

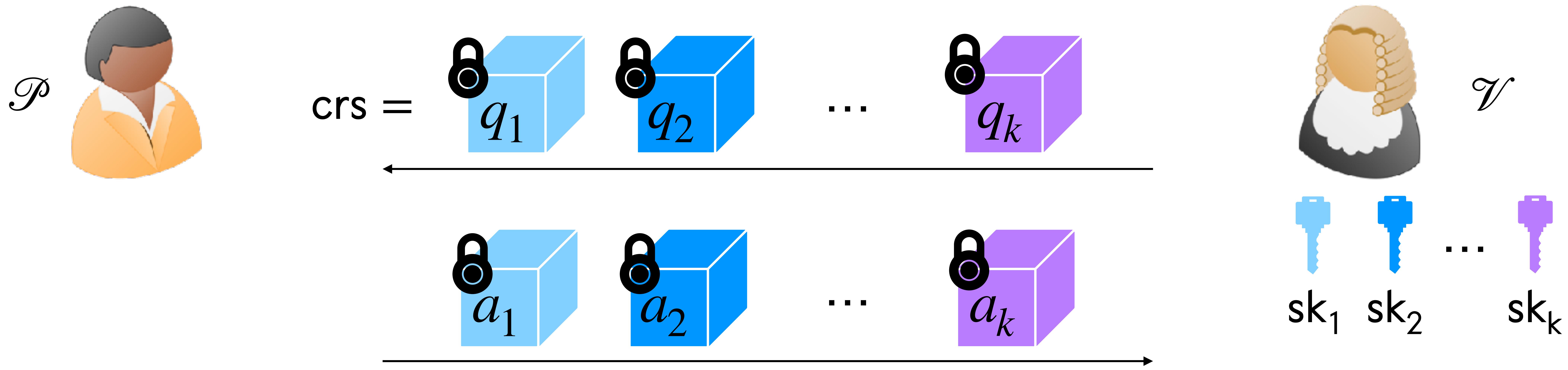
KRR14 Construction

Based on [Biehl-Meyer-Wetzel ’98]



KRR14 Construction

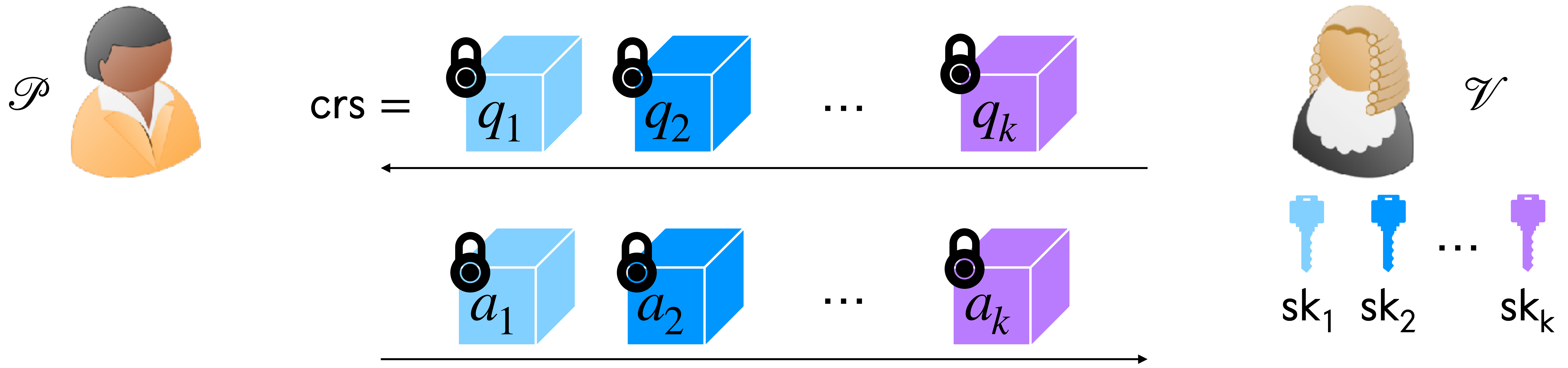
Based on [Biehl-Meyer-Wetzel ’98]



Issue 1: Need a **secret key** to verify (can be solved using [JKL**M**25])

KRR14 Construction

Based on [Biehl-Meyer-Wetzel ’98]

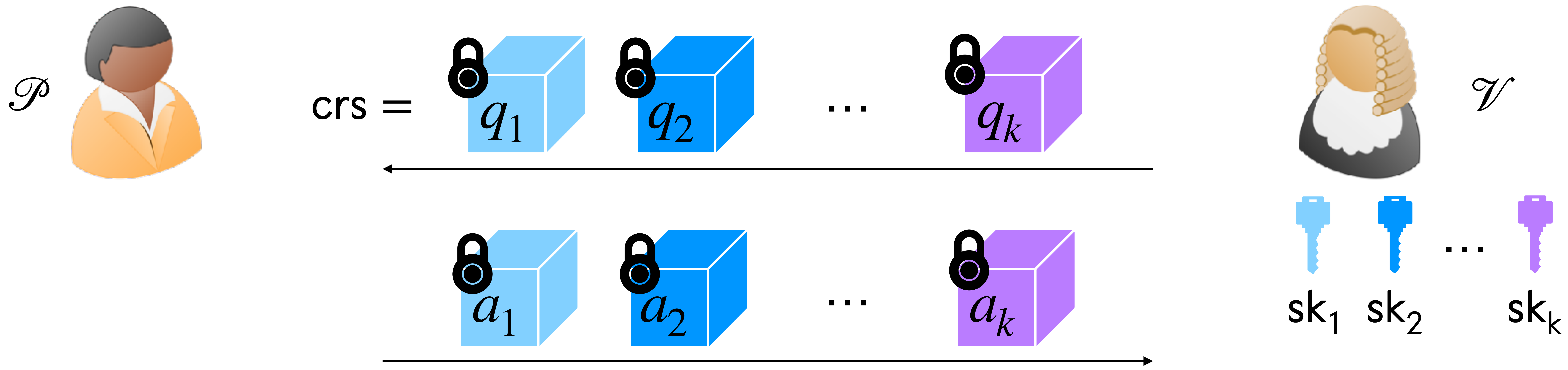


Issue 1: Need a **secret key** to verify (can be solved using [JKL**M**25])

Issue 2: There exist PCPs and FHE schemes for which this compiler is **not sound.** [DLNNR04, DHRW16]

KRR14 Construction

Based on [Biehl-Meyer-Wetzel '98]



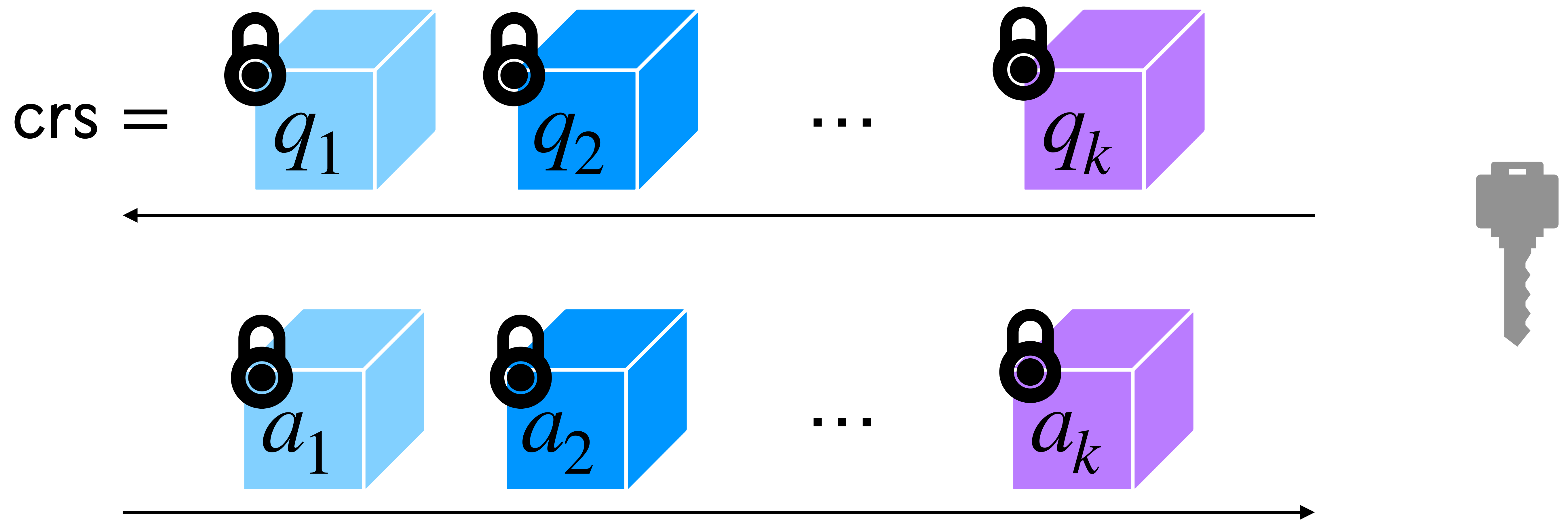
Issue 1: Need a **secret key** to verify (can be solved using [JKL**M**25])

Issue 2: There exist PCPs and FHE schemes for which this compiler is **not sound.** [DLNNR04, DHRW16]

- Still runs into the problem that the verifier is not committed to **one** Π!
  (Bonus slide demonstrating this)

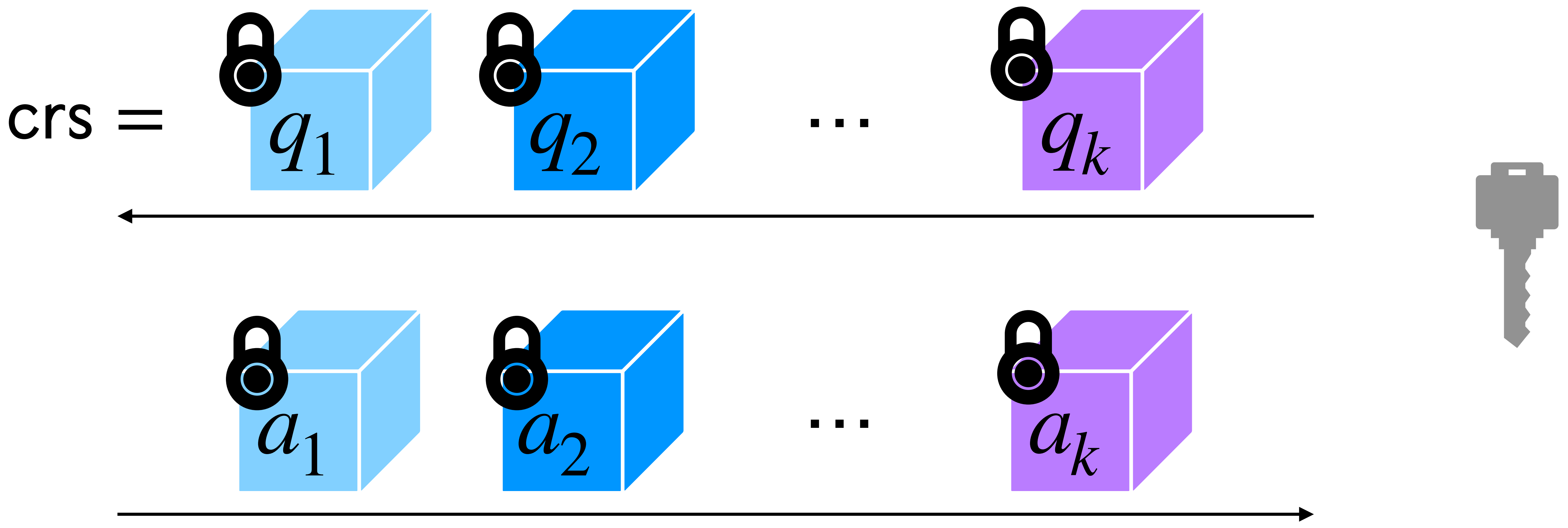
KRR14 Construction

Based on [Biehl-Meyer-Wetzel ’98]



KRR14 Construction

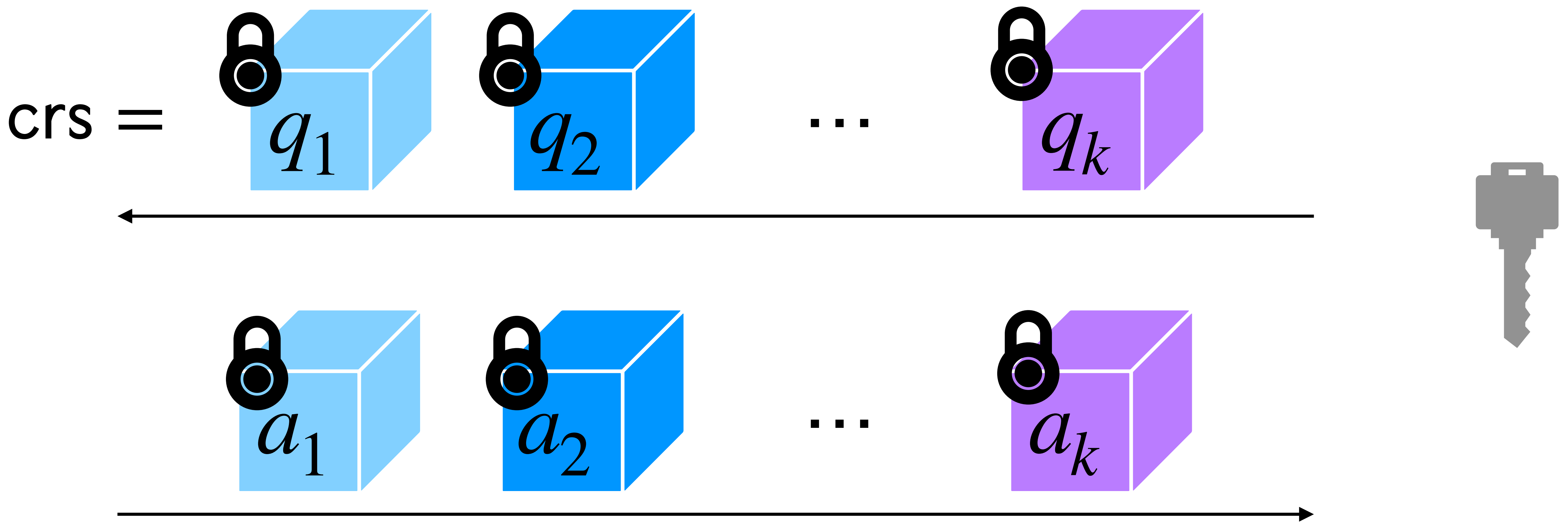
Based on [Biehl-Meyer-Wetzel ’98]



Independent keys: Adversary doesn’t “**see all**” $\{q_i\}_i$ anymore.

KRR14 Construction

Based on [Biehl-Meyer-Wetzel '98]

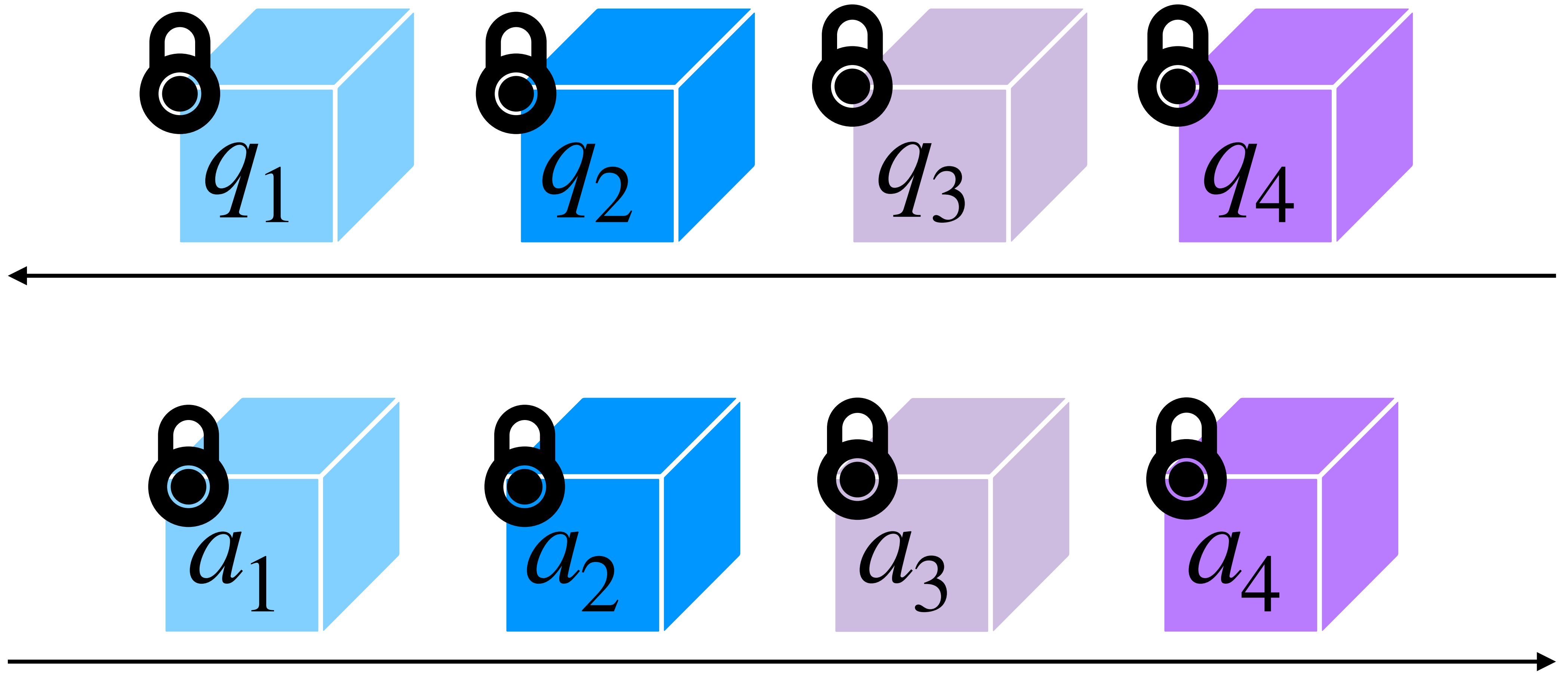


Independent keys: Adversary doesn't "**see all**" $\{q_i\}_i$ anymore.

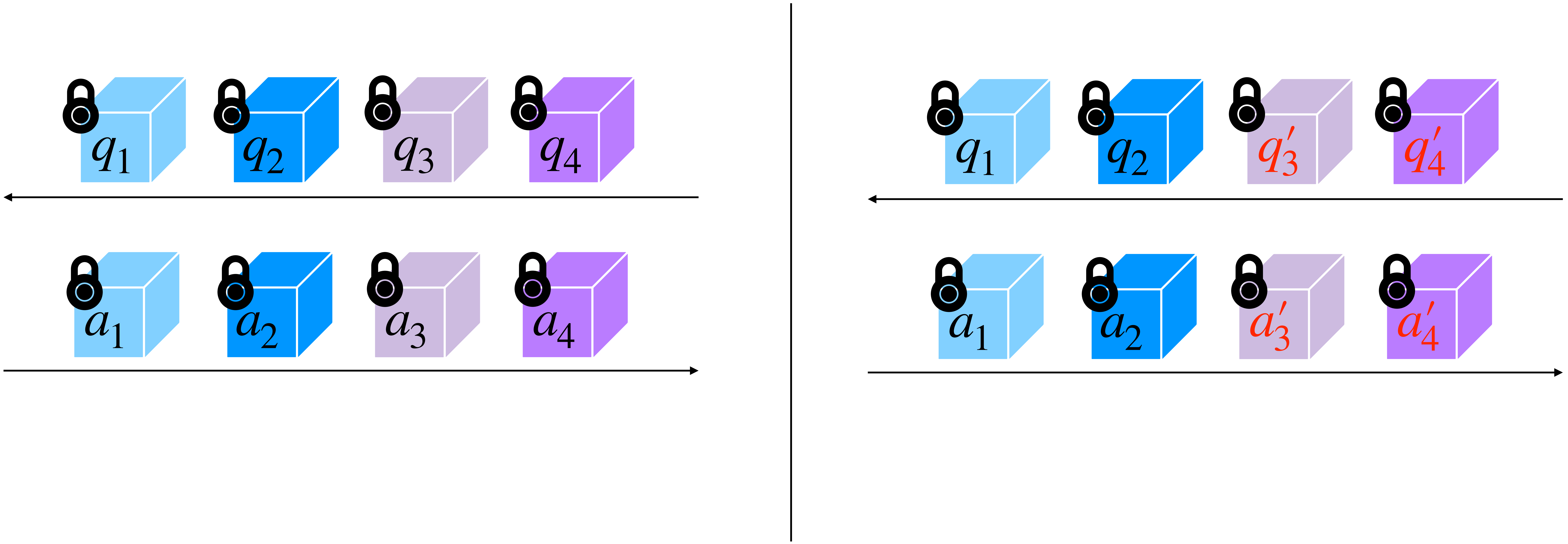
i.e. "Information" should **not** be transmitted between answers.

KRR14 Guarantee

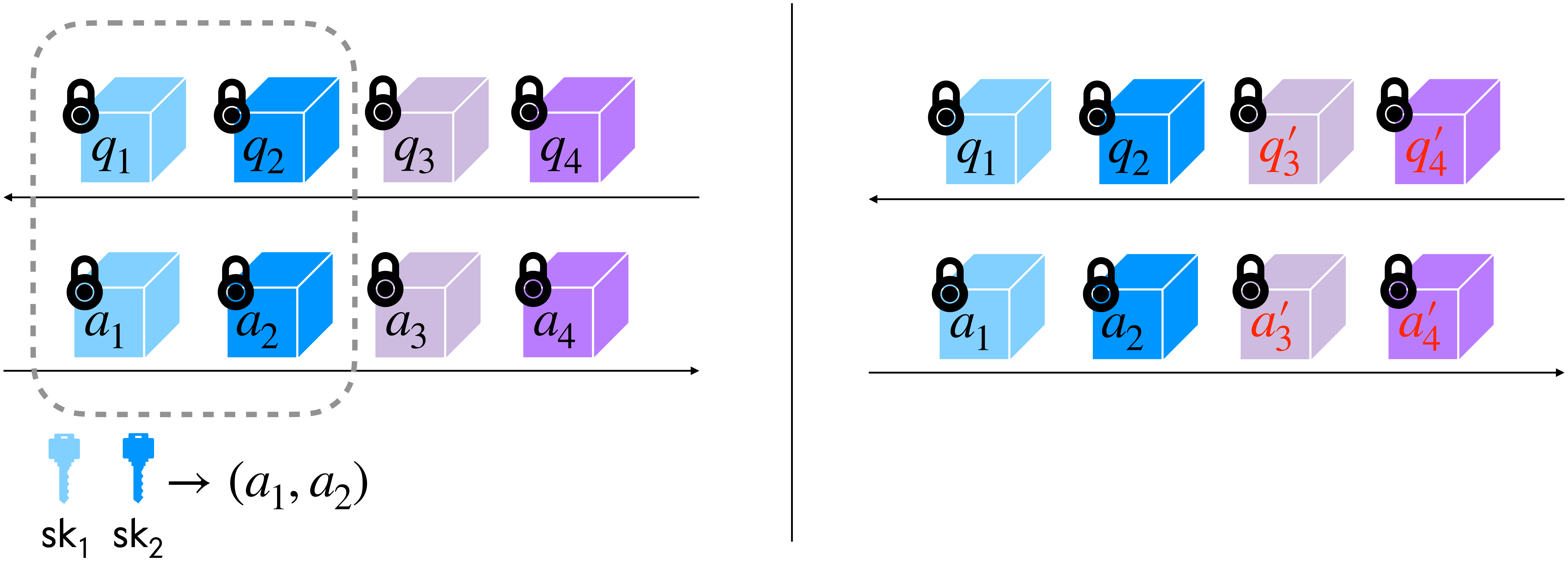
KRR14 Guarantee



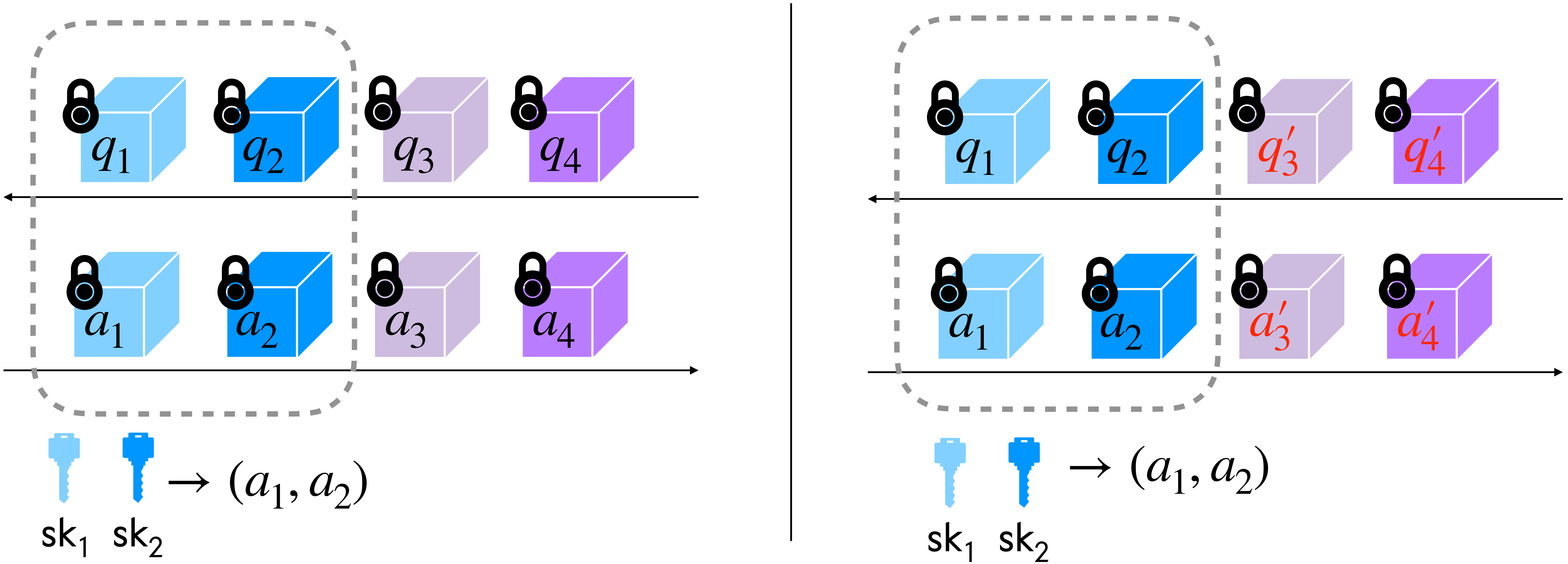
KRR14 Guarantee



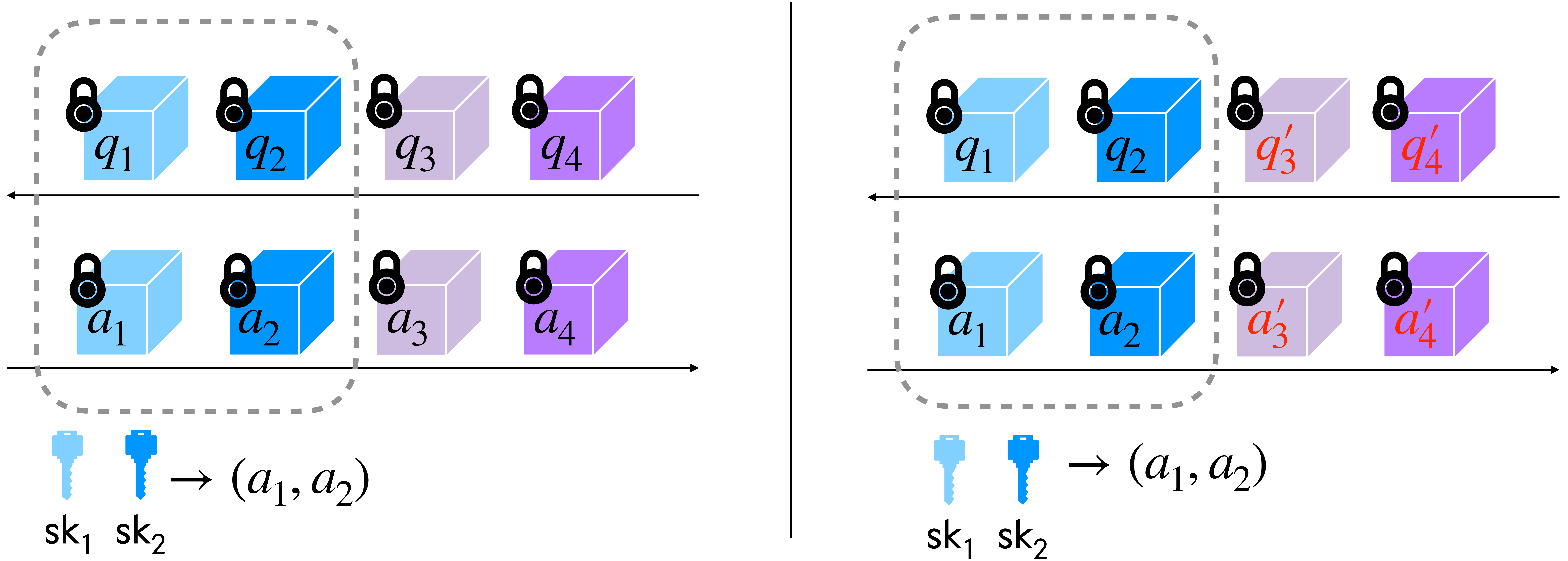
KRR14 Guarantee



KRR14 Guarantee



KRR14 Guarantee

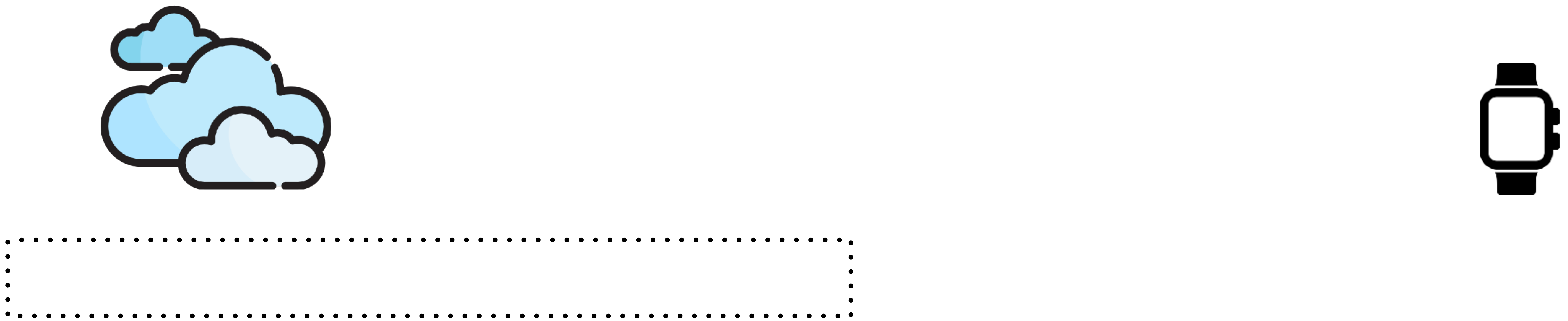


Semantic security of $(\mathsf{sk}_3, \mathsf{sk}_4)$:

PCPs answers (a_1, a_2) should be **indistinguishable** in both experiments.

Enter: Non-Signaling PCPs

Enter: Non-Signaling PCPs



Enter: Non-Signaling PCPs

Family of distributions:

$$\mathscr{D} = \{D_Q\}_Q$$

Enter: Non-Signaling PCPs

Family of distributions:

$$\mathscr{D} = \{D_Q\}_Q$$

Q

Enter: Non-Signaling PCPs

Family of distributions:

$$\mathscr{D} = \{D_Q\}_Q$$

Q

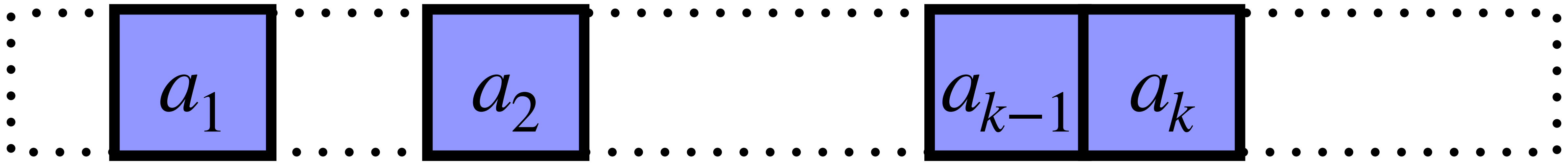
$A_Q \leftarrow D_Q$

Enter: Non-Signaling PCPs

Family of distributions:

$\mathscr{D} = \{D_Q\}_Q$

Q



$A_Q \leftarrow D_Q$

Enter: Non-Signaling PCPs

Family of distributions:

$\mathscr{D} = \{D_Q\}_Q$

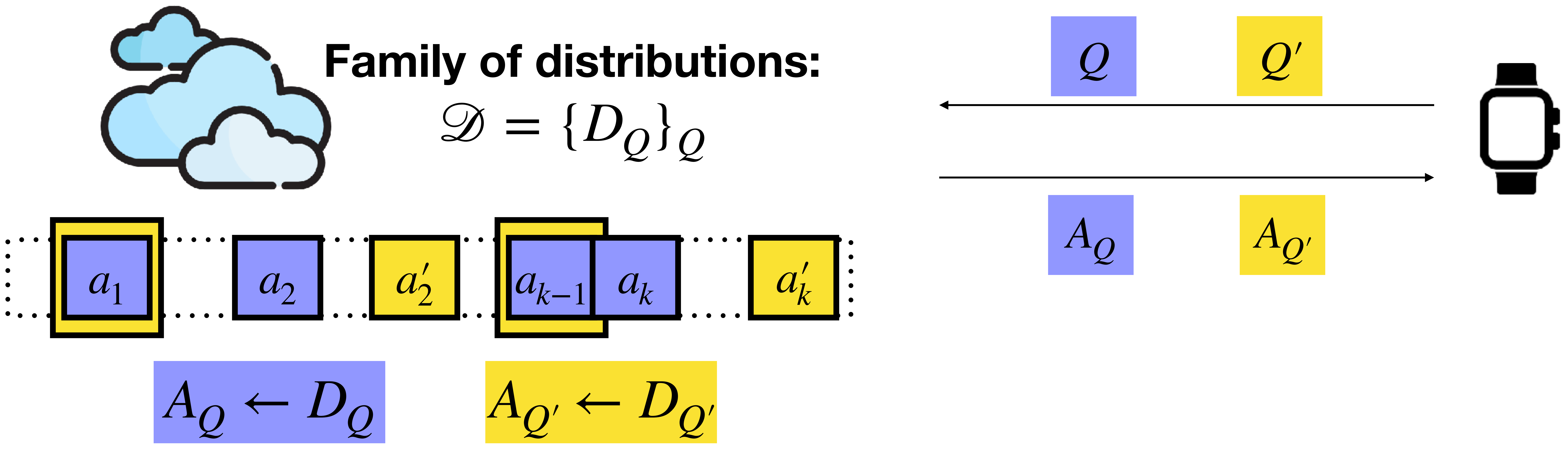
Q

A_Q

a_1 a_2 a_{k-1} a_k

$A_Q \leftarrow D_Q$

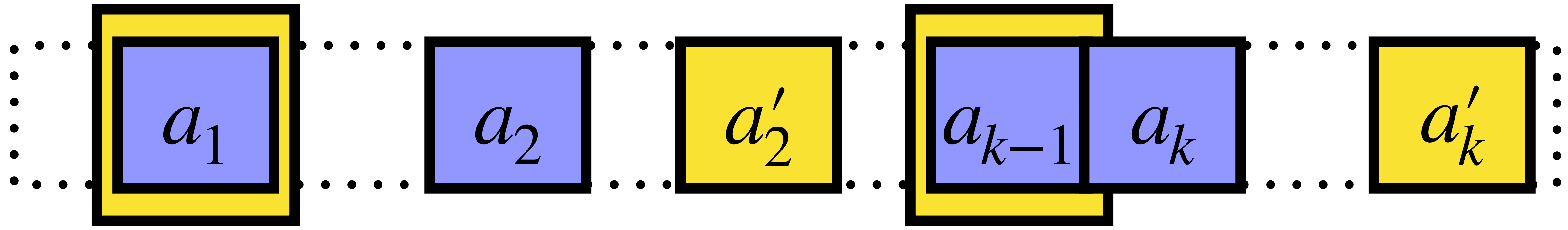
Enter: Non-Signaling PCPs



Enter: Non-Signaling PCPs

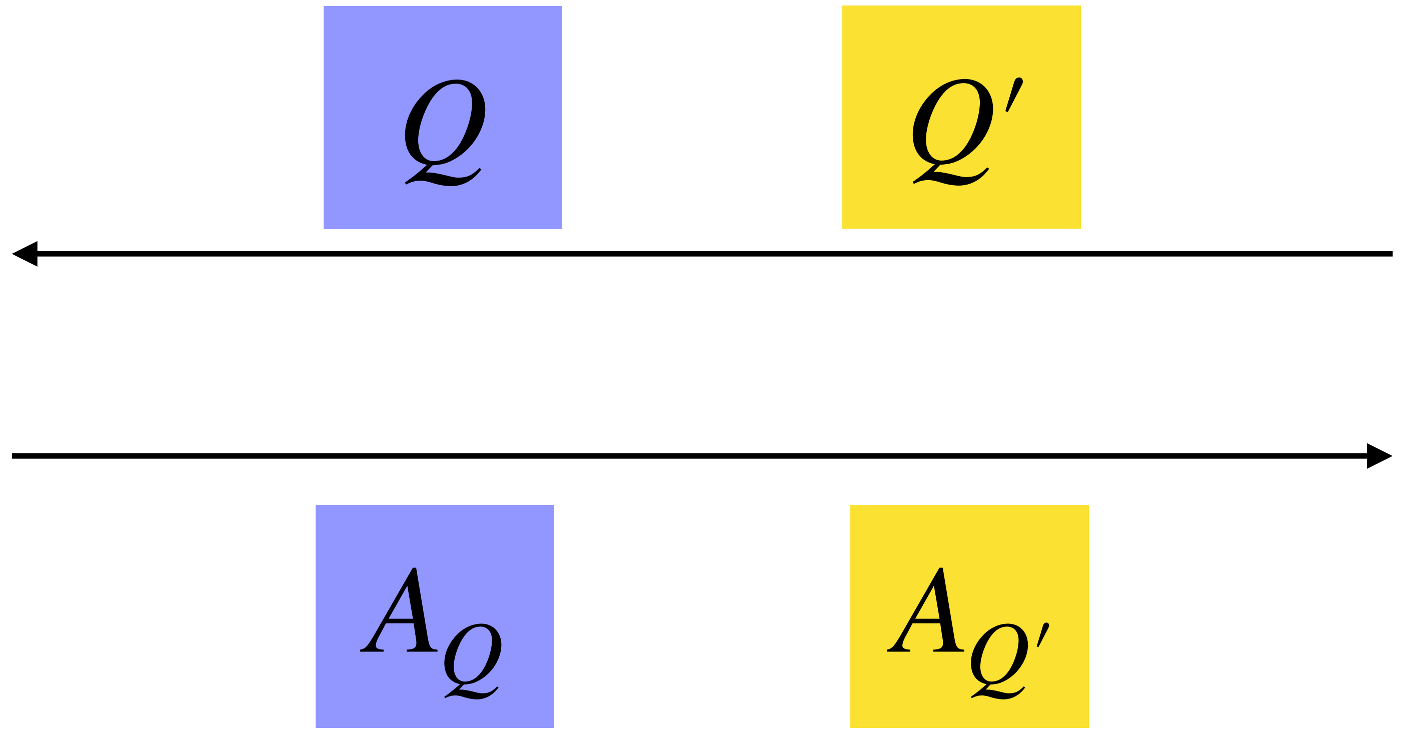
Family of distributions:

$\mathscr{D} = \{D_Q\}_Q$



$A_Q \leftarrow D_Q$

$A_{Q'} \leftarrow D_{Q'}$



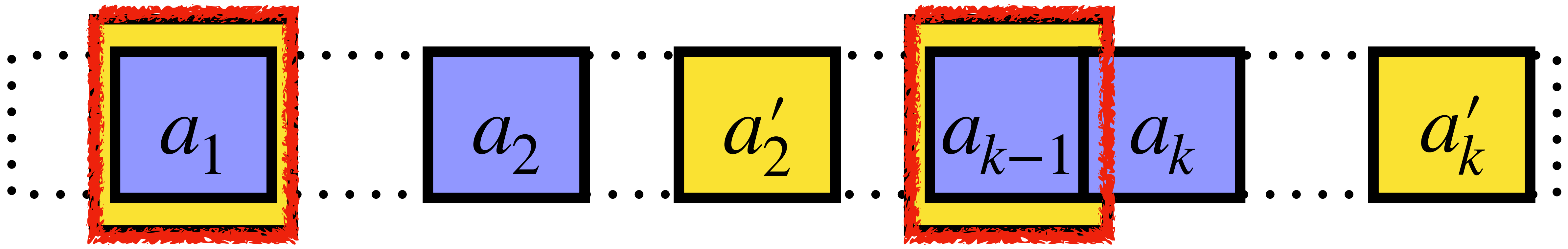
Non-signaling: Let $S = Q \cap Q'$.

$A_Q|_S \equiv A_{Q'}|_S$

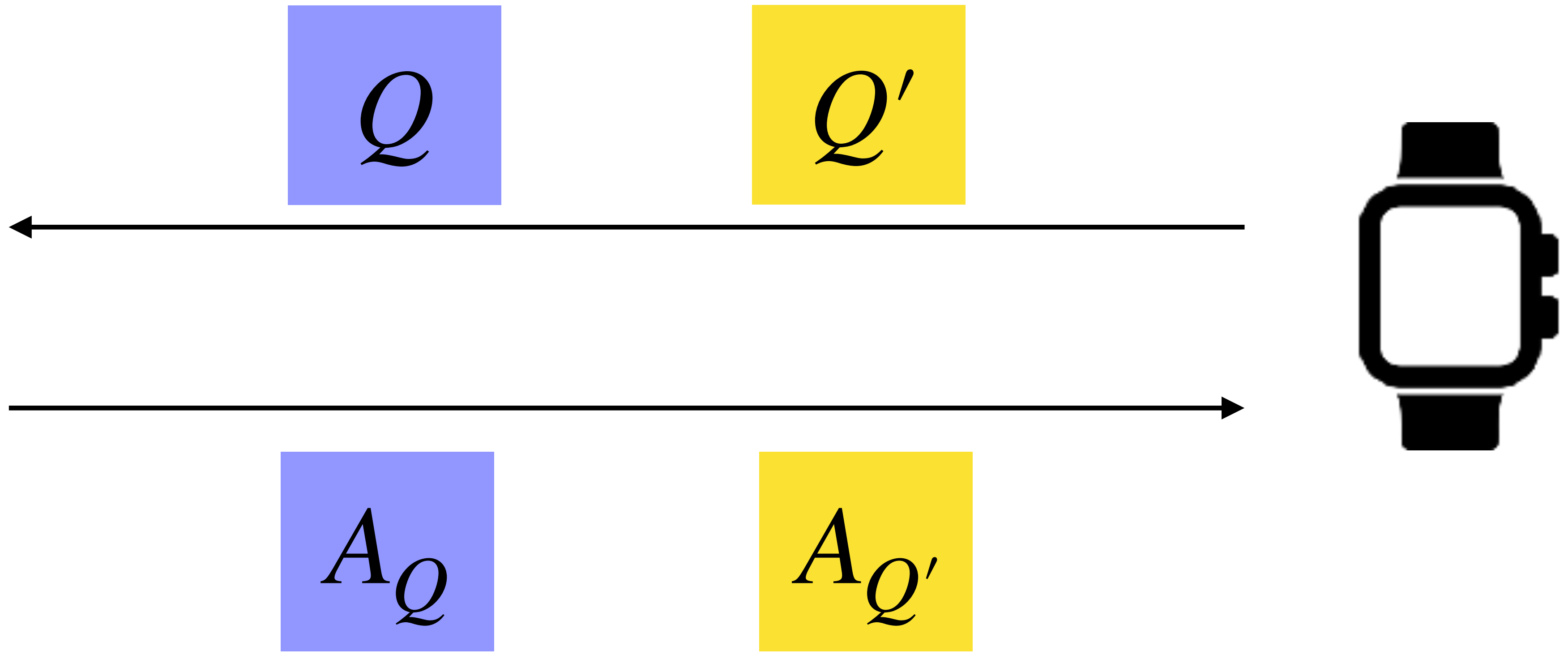
Enter: Non-Signaling PCPs

Family of distributions:

$\mathscr{D} = \{D_Q\}_Q$



$A_Q \leftarrow D_Q$ $A_{Q'} \leftarrow D_{Q'}$

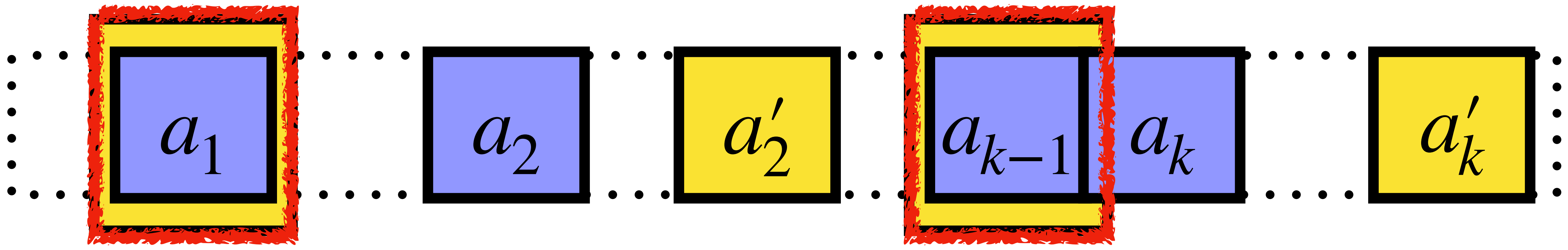


Non-signaling: Let $S = Q \cap Q'$.

$A_Q|_S \equiv A_{Q'}|_S$

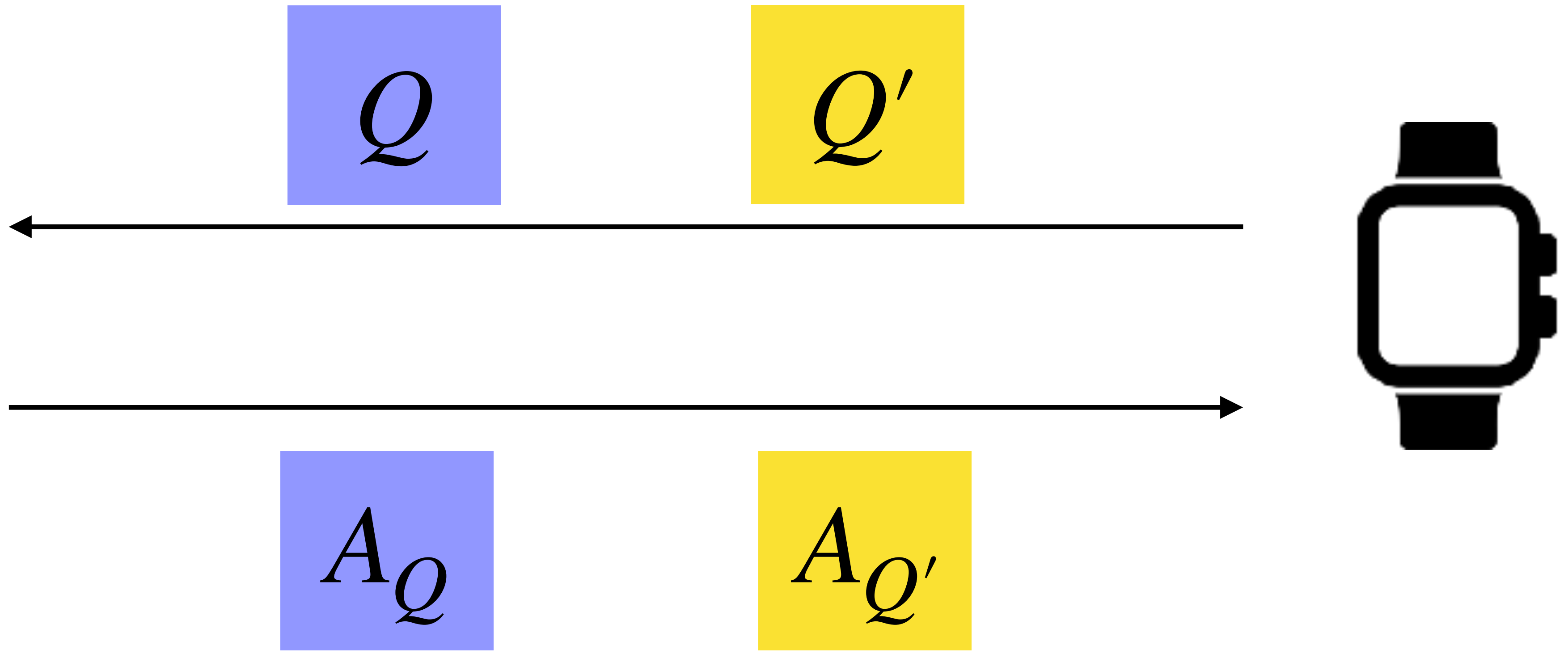
Enter: Non-Signaling PCPs

Family of distributions:
$\mathscr{D} = \{D_Q\}_Q$



$A_Q \leftarrow D_Q$ $\quad$ $A_{Q'} \leftarrow D_{Q'}$

Non-signaling: Let $S = Q \cap Q'$.
$$A_Q|_S \equiv A_{Q'}|_S$$



NS Soundness: If $x \notin \mathscr{L}$
$$\Pr_{Q,A_Q}[V(x, Q, A_Q) = 1] \leq \frac{1}{\text{poly(n)}}$$

Enter: Non-Signaling PCPs

[Alternate view]

Non-signaling:

$S = Q \cap Q'$.

$A_Q|_S \equiv A_{Q'}|_S$

Enter: Non-Signaling PCPs

[Alternate view]

Non-signaling:

$S = Q \cap Q'$.

$A_Q|_S \equiv A_{Q'}|_S$

Arbitrary correlation that obeys "locality"

Enter: Non-Signaling PCPs

[Alternate view]

Non-signaling:

$S = Q \cap Q'$.

$A_Q|_S \equiv A_{Q'}|_S$

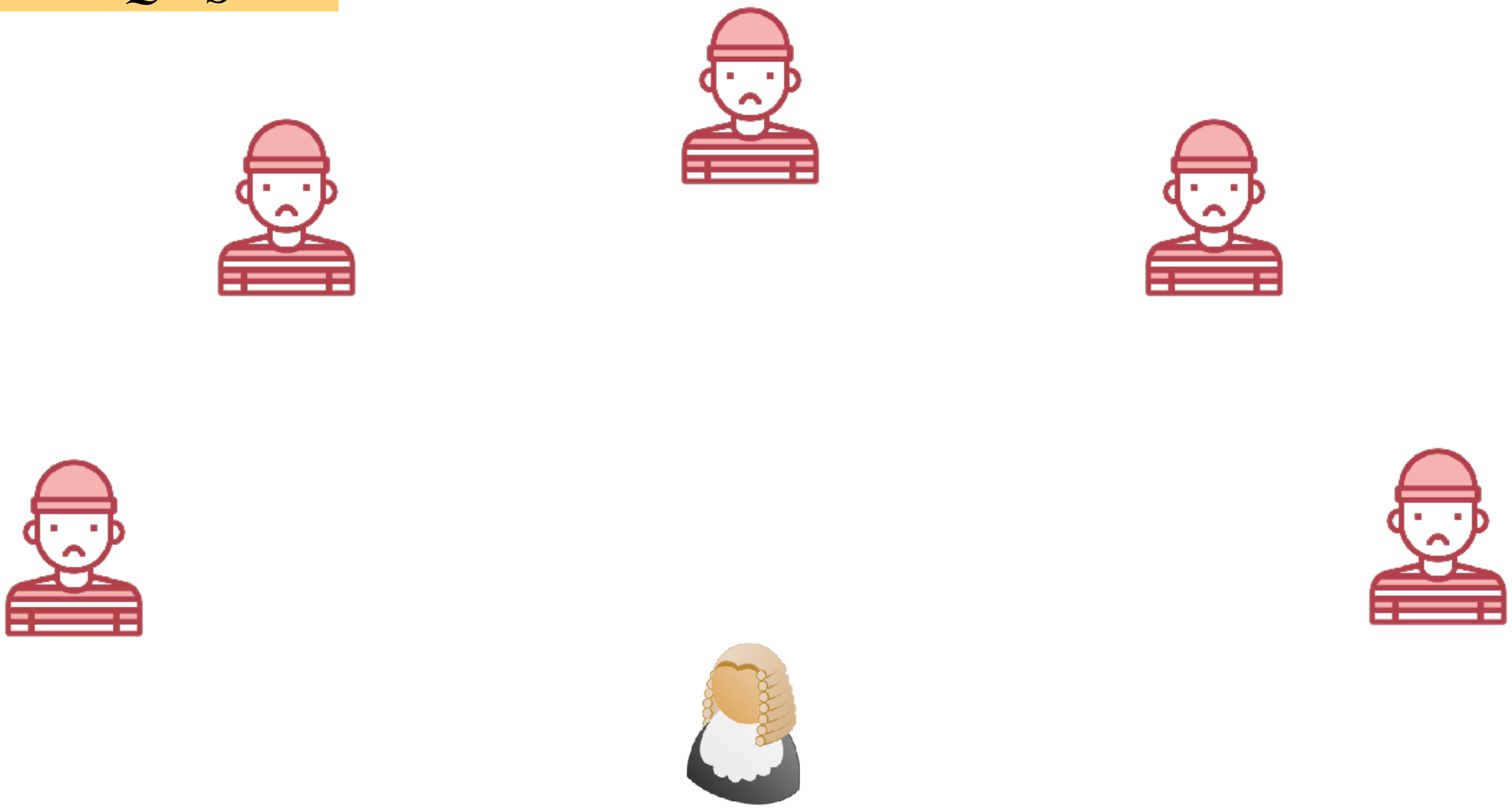
Arbitrary correlation that obeys "locality"

Enter: Non-Signaling PCPs

[Alternate view]

Non-signaling:
$S = Q \cap Q'$.
$A_Q|_S \equiv A_{Q'}|_S$

Arbitrary correlation that obeys "locality"

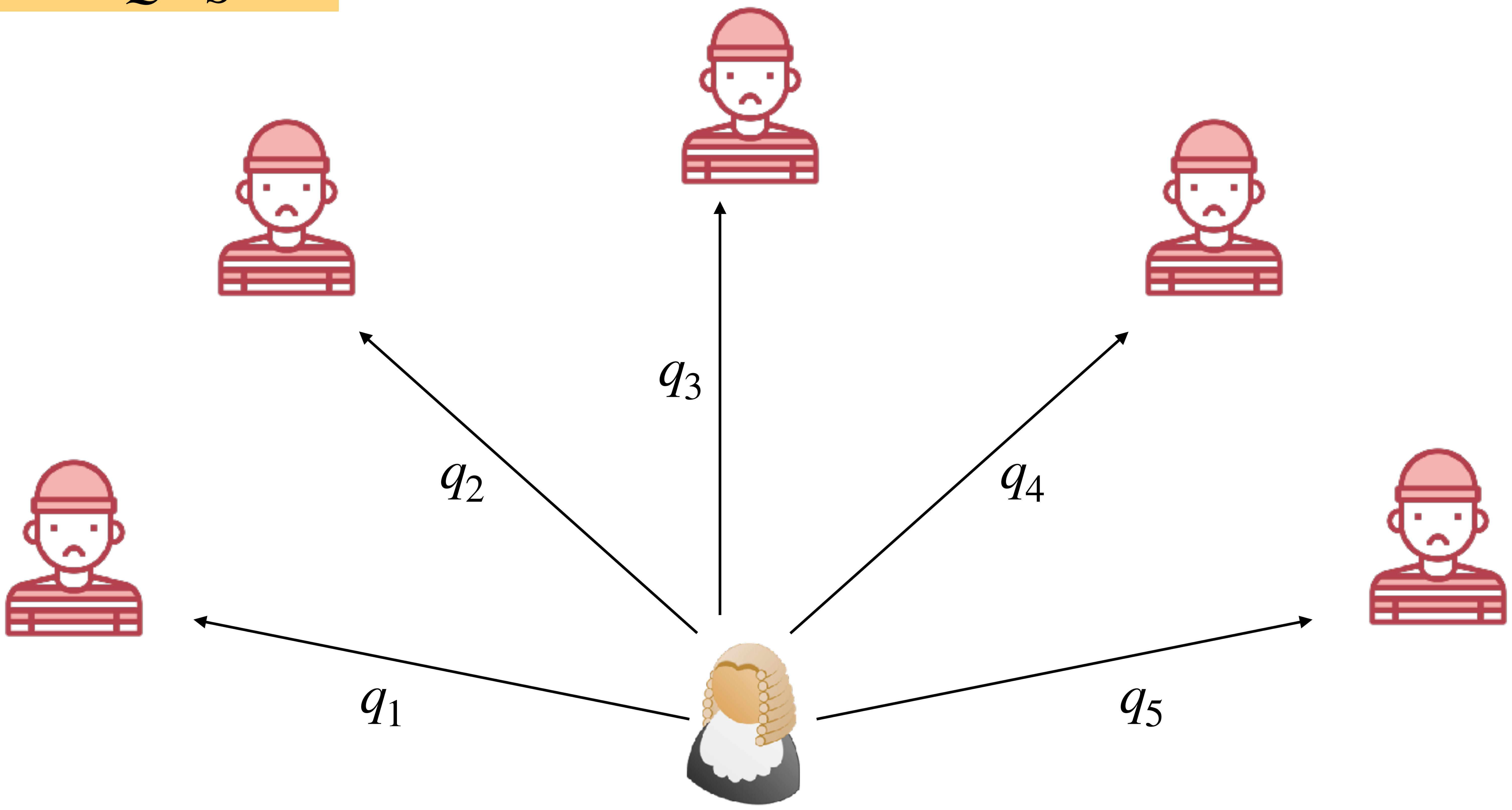


Enter: Non-Signaling PCPs

[Alternate view]

Non-signaling:
$S = Q \cap Q'$.
$A_Q|_S \equiv A_{Q'}|_S$

Arbitrary correlation that obeys "locality"

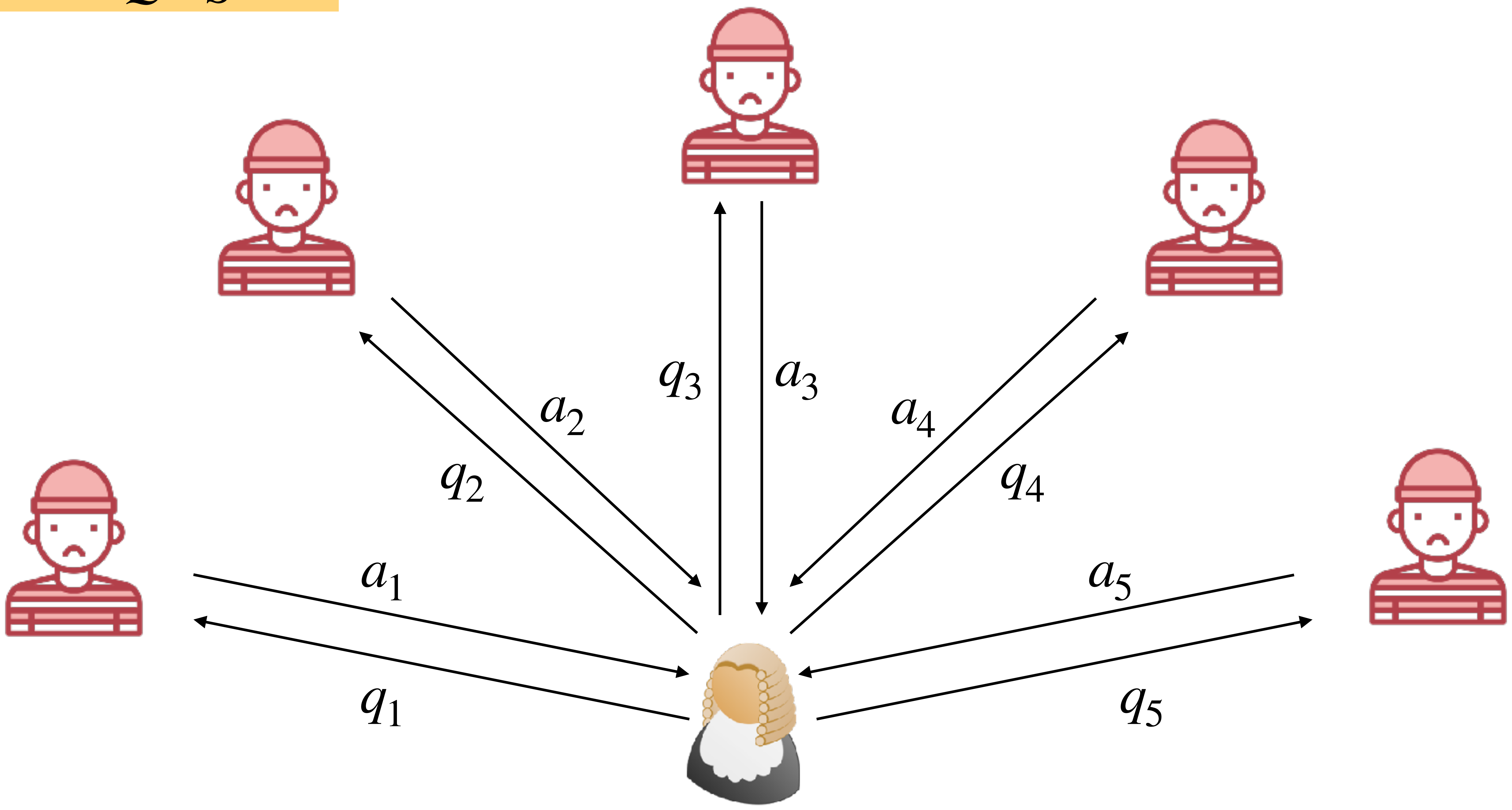


Enter: Non-Signaling PCPs

[Alternate view]

Non-signaling:
$S = Q \cap Q'$.
$A_Q|_S \equiv A_{Q'}|_S$

Arbitrary correlation that obeys "locality"



Enter: Non-Signaling PCPs

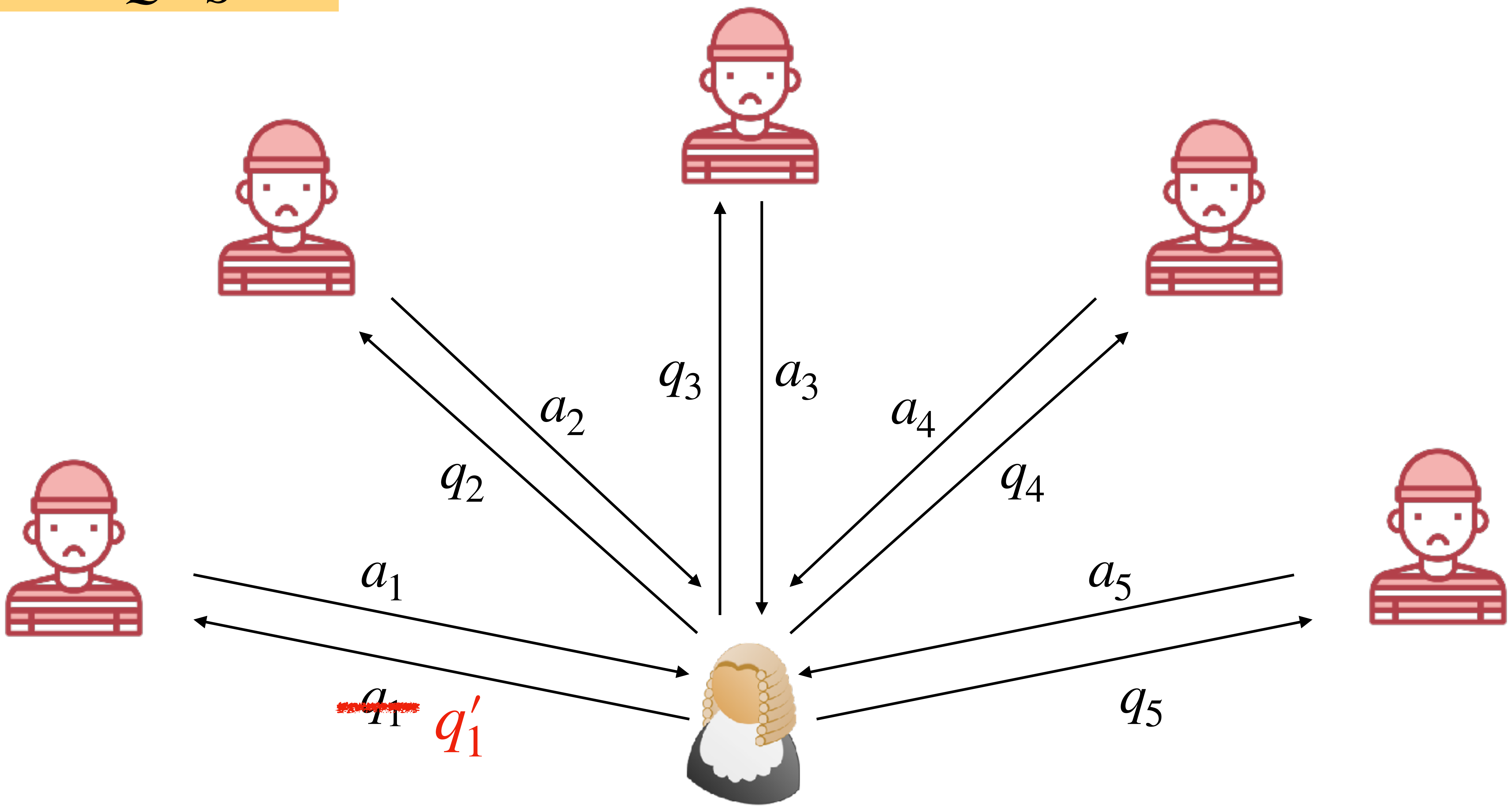
[Alternate view]

Non-signaling:

$S = Q \cap Q'$.

$A_Q|_S \equiv A_{Q'}|_S$

Arbitrary correlation that obeys "locality"



Enter: Non-Signaling PCPs

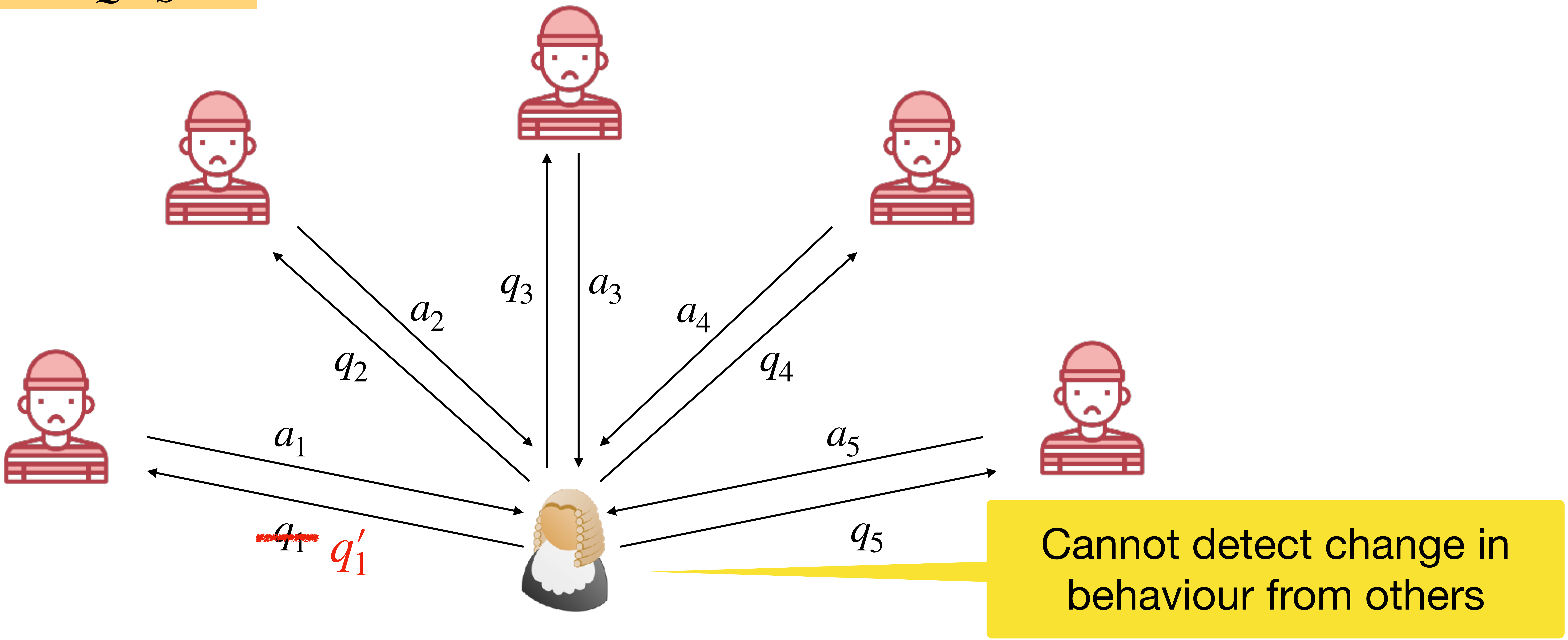
[Alternate view]

Non-signaling:

$S = Q \cap Q'$.

$A_Q|_S \equiv A_{Q'}|_S$

Arbitrary correlation that obeys "locality"



Enter: Non-Signaling PCPs

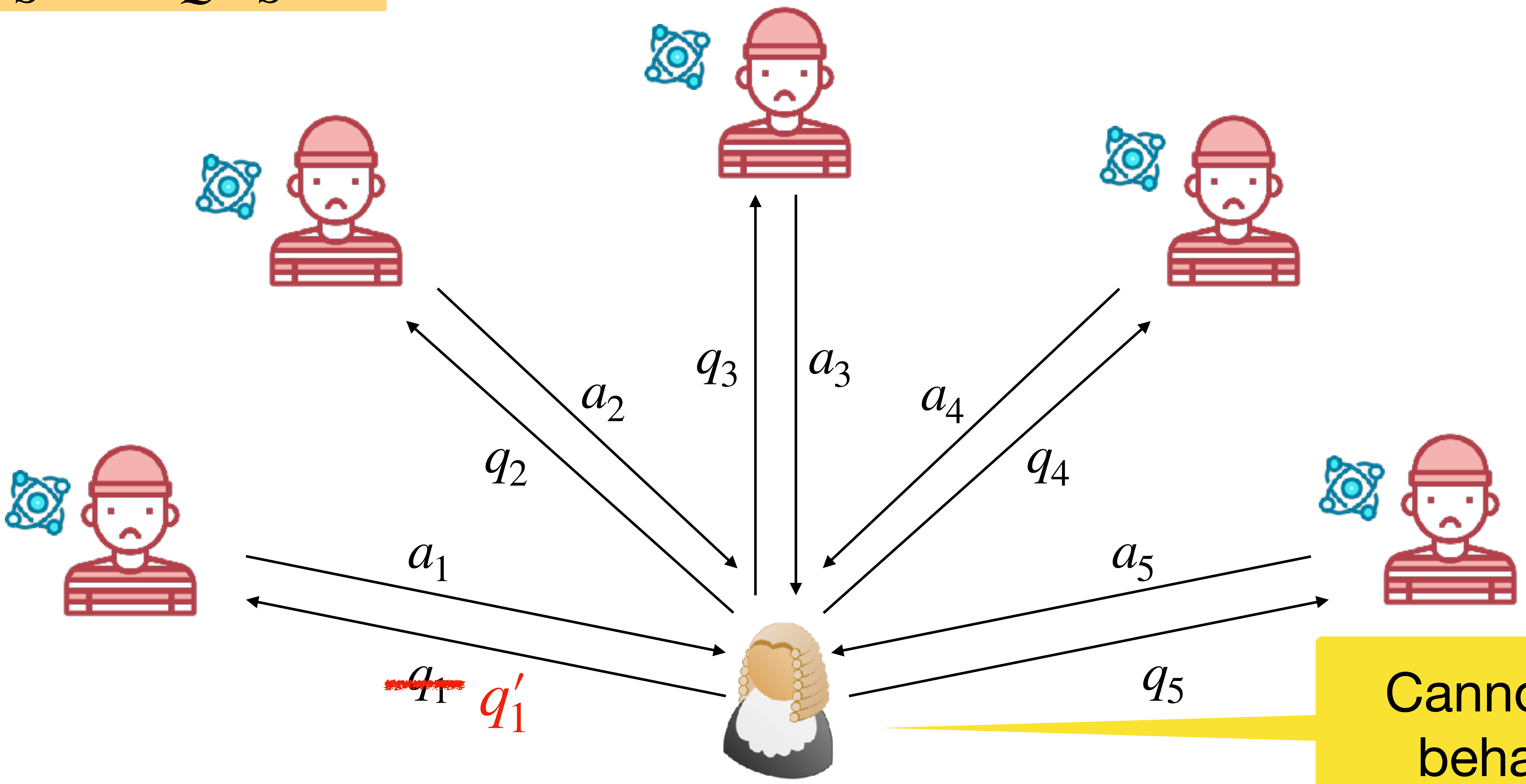
[Alternate view]

Non-signaling:

$S = Q \cap Q'$.

$A_Q|_S \equiv A_{Q'}|_S$

Arbitrary correlation that obeys "locality"



Cannot detect change in behaviour from others

Enter: Non-Signaling PCPs

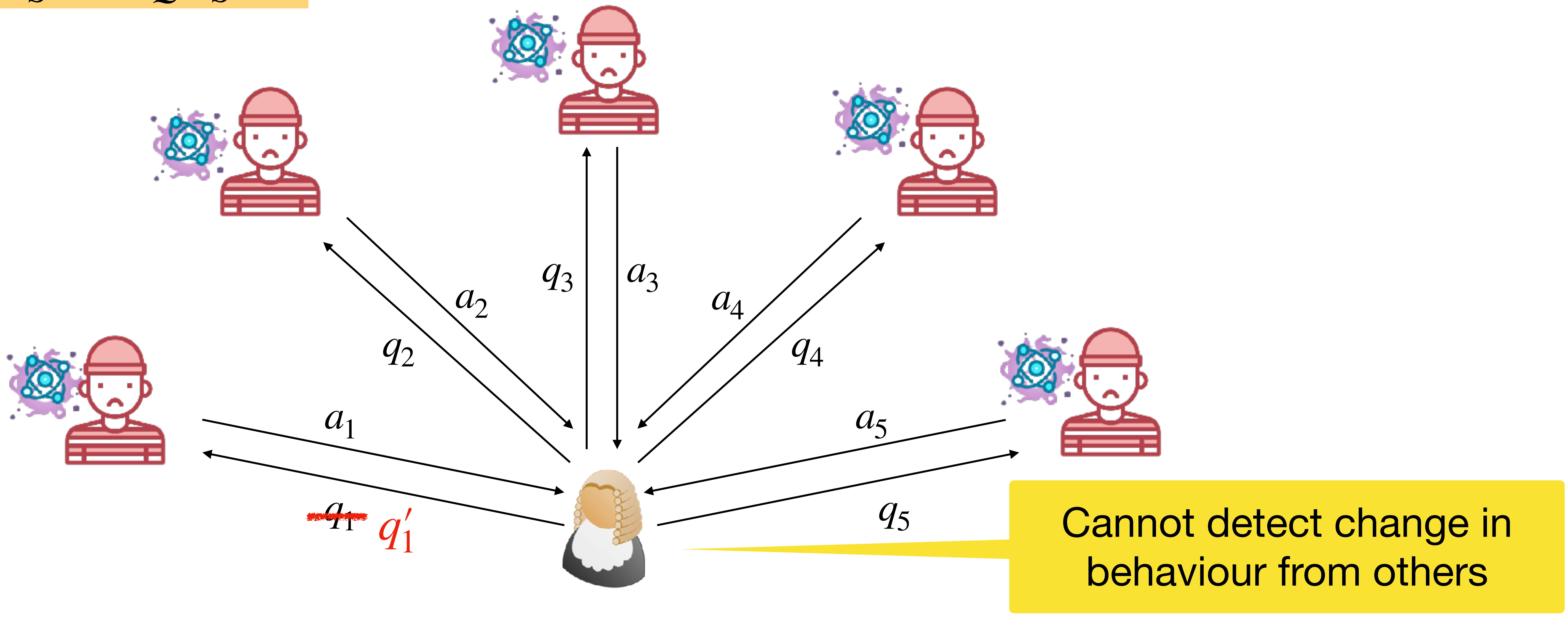
[Alternate view]

Non-signaling:

$S = Q \cap Q'$.

$A_Q|_S \equiv A_{Q'}|_S$

Arbitrary correlation that obeys "locality"

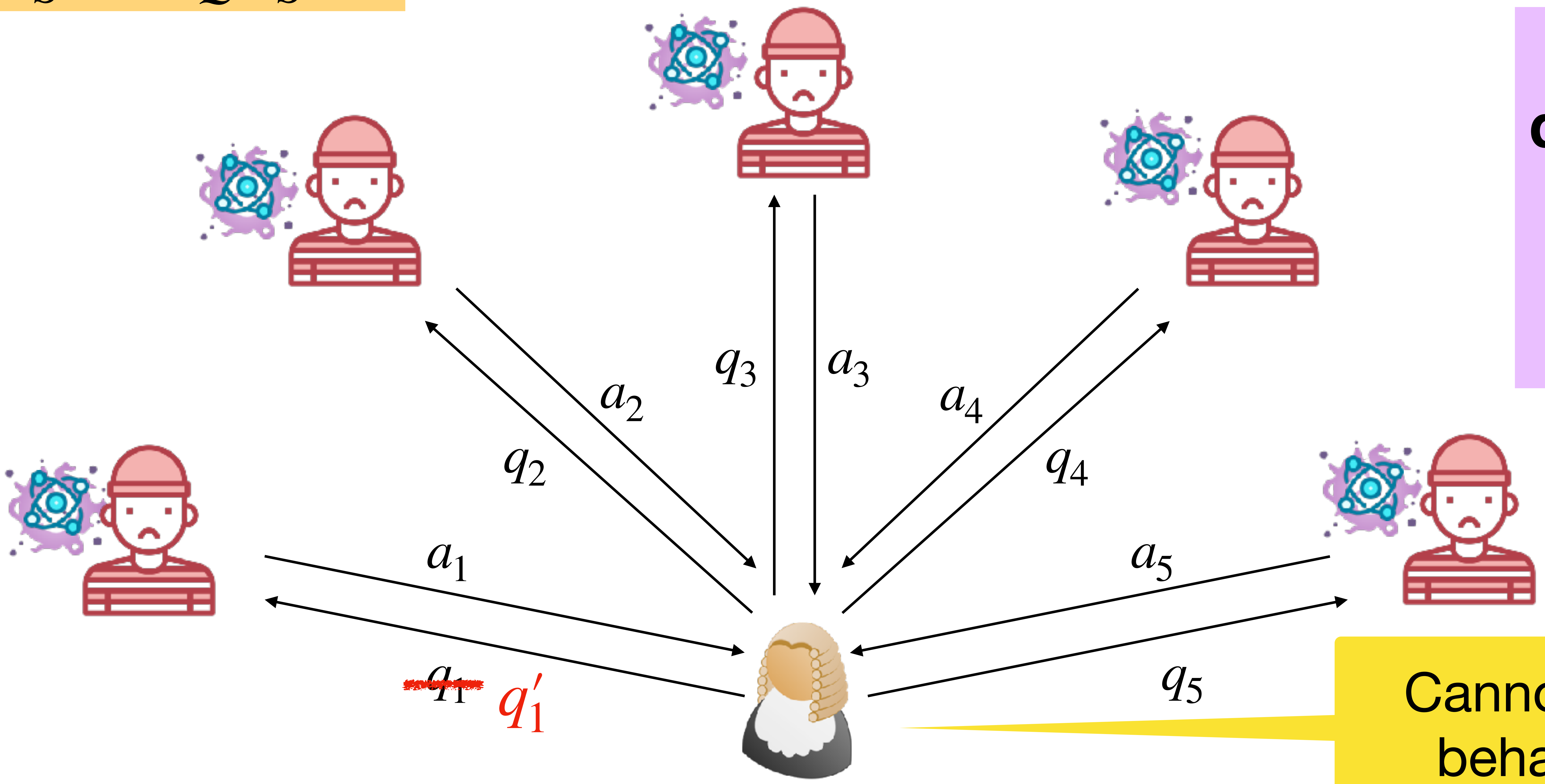


Enter: Non-Signaling PCPs

[Alternate view]

Non-signaling:
$S = Q \cap Q'$.
$A_Q|_S \equiv A_{Q'}|_S$

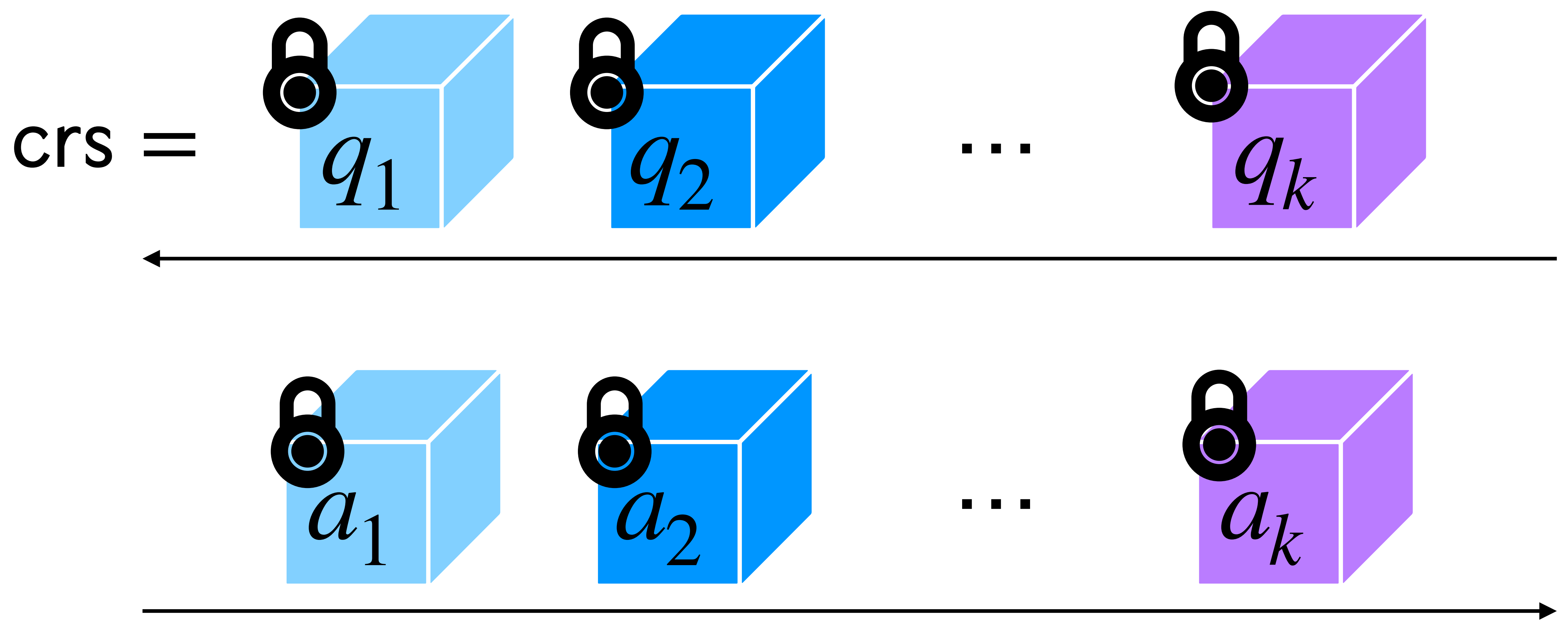
Arbitrary correlation that obeys "locality"



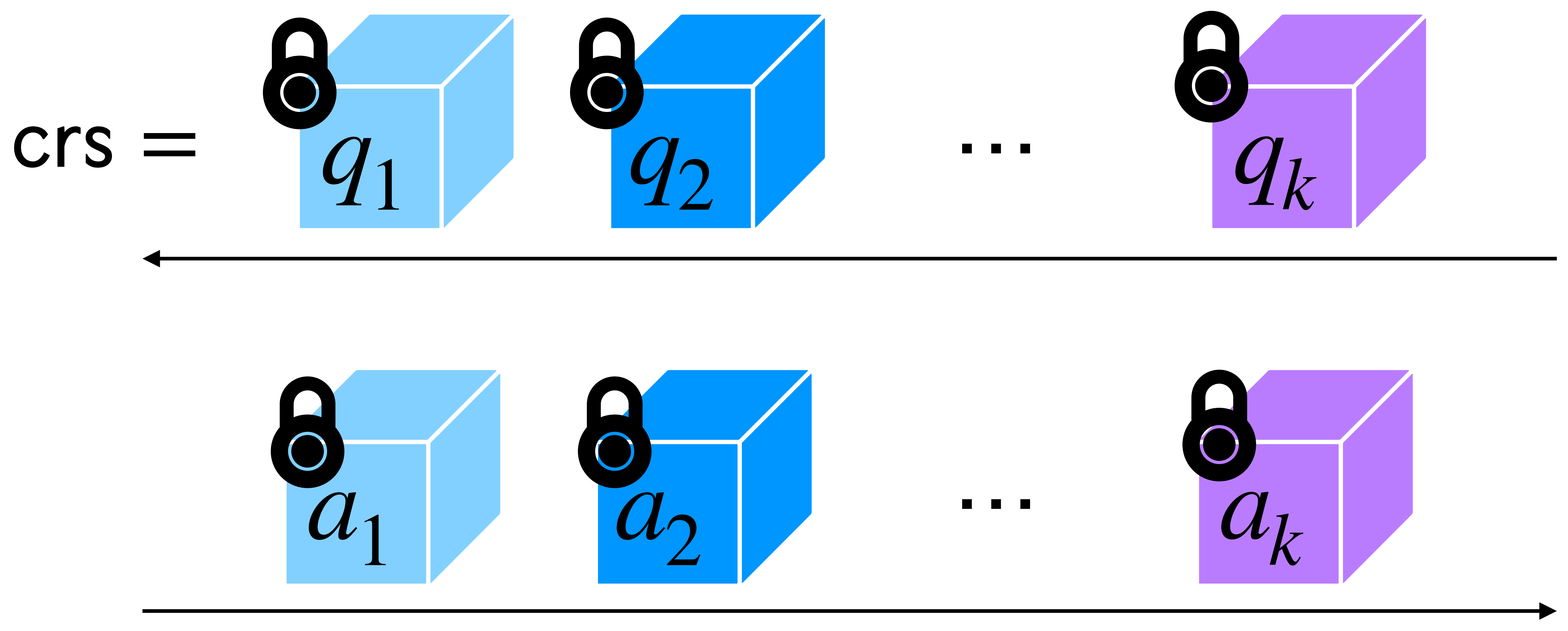
Generalization of quantum strategies.
E.g. CHSH game.
Quantum: ~0.85
NS: 1

Cannot detect change in behaviour from others

KRR Construction



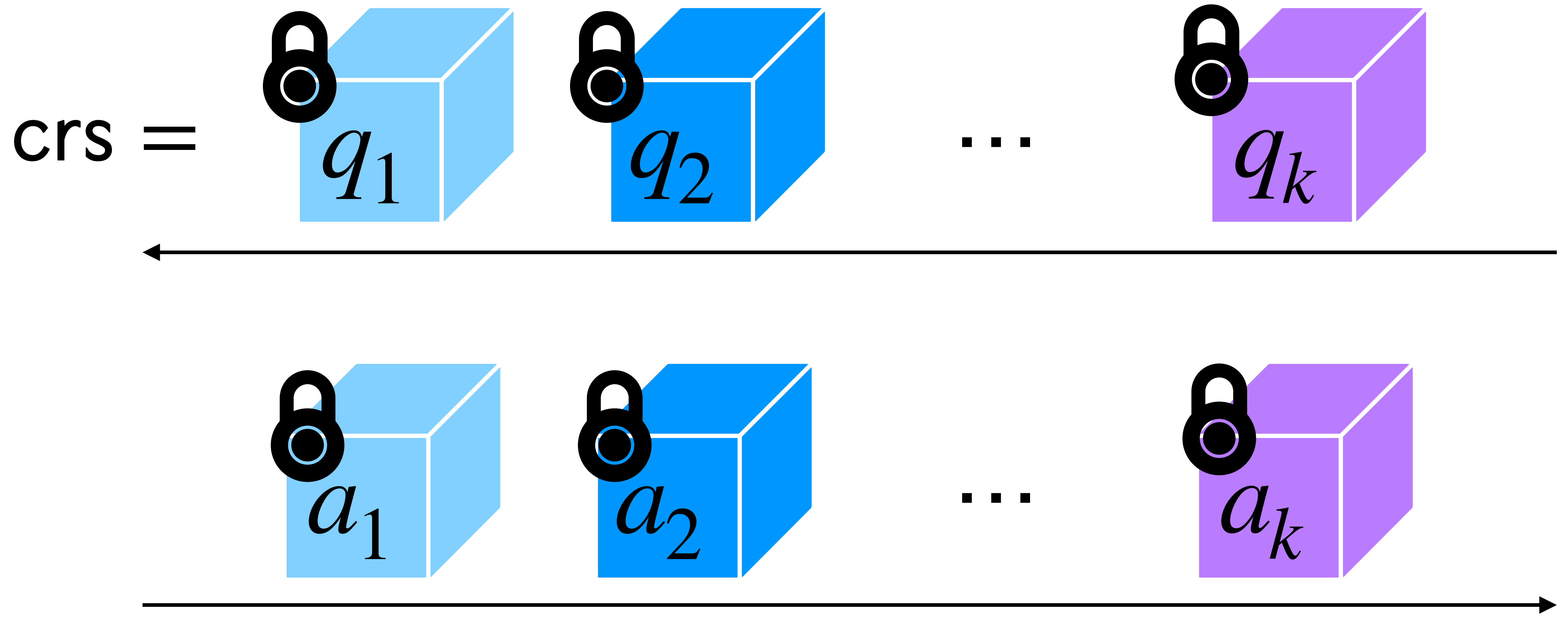
KRR Construction



Theorem statement:
FHE Security + NS-PCP = KRR Secure

Pf. Cheating $\mathscr{A}$ gives an NS PCP strategy ∎

KRR Construction



Theorem statement:
FHE Security + NS-PCP = KRR Secure

Pf. Cheating $\mathscr{A}$ gives an NS PCP strategy ∎

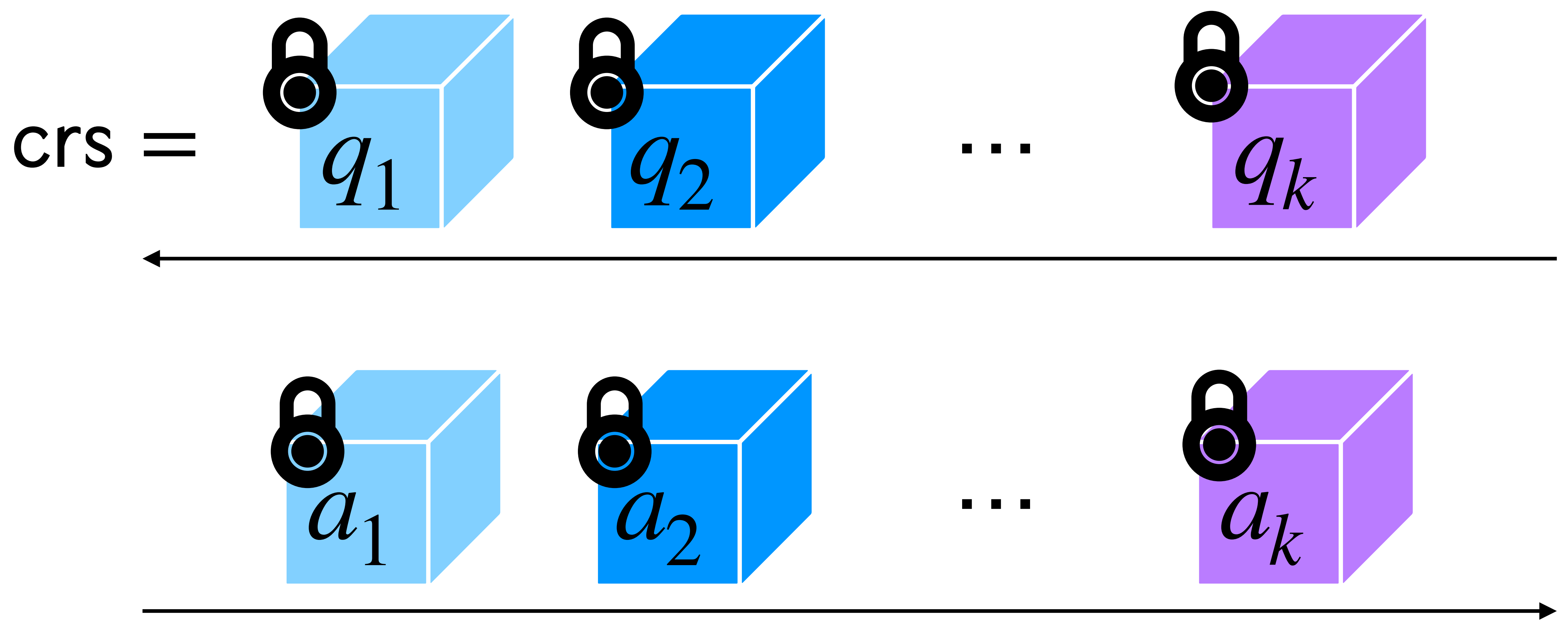
We have nsPCPs for:

Deterministic Computations
[KRR14, BHK17, CJJ21ab, KVZ21, …]

NTISP [BKK+17, KVZ21]

Batch-NP [CJJ21ab, WW22, …]

KRR Construction



Theorem statement:
FHE Security + NS-PCP = KRR Secure

Pf. Cheating $\mathscr{A}$ gives an NS PCP strategy ∎

We have nsPCPs for:

Deterministic Computations
[KRR14, BHK17, CJJ21ab, KVZ21, …]

NTISP [BKK+17, KVZ21]

Batch-NP [CJJ21ab, WW22, …]

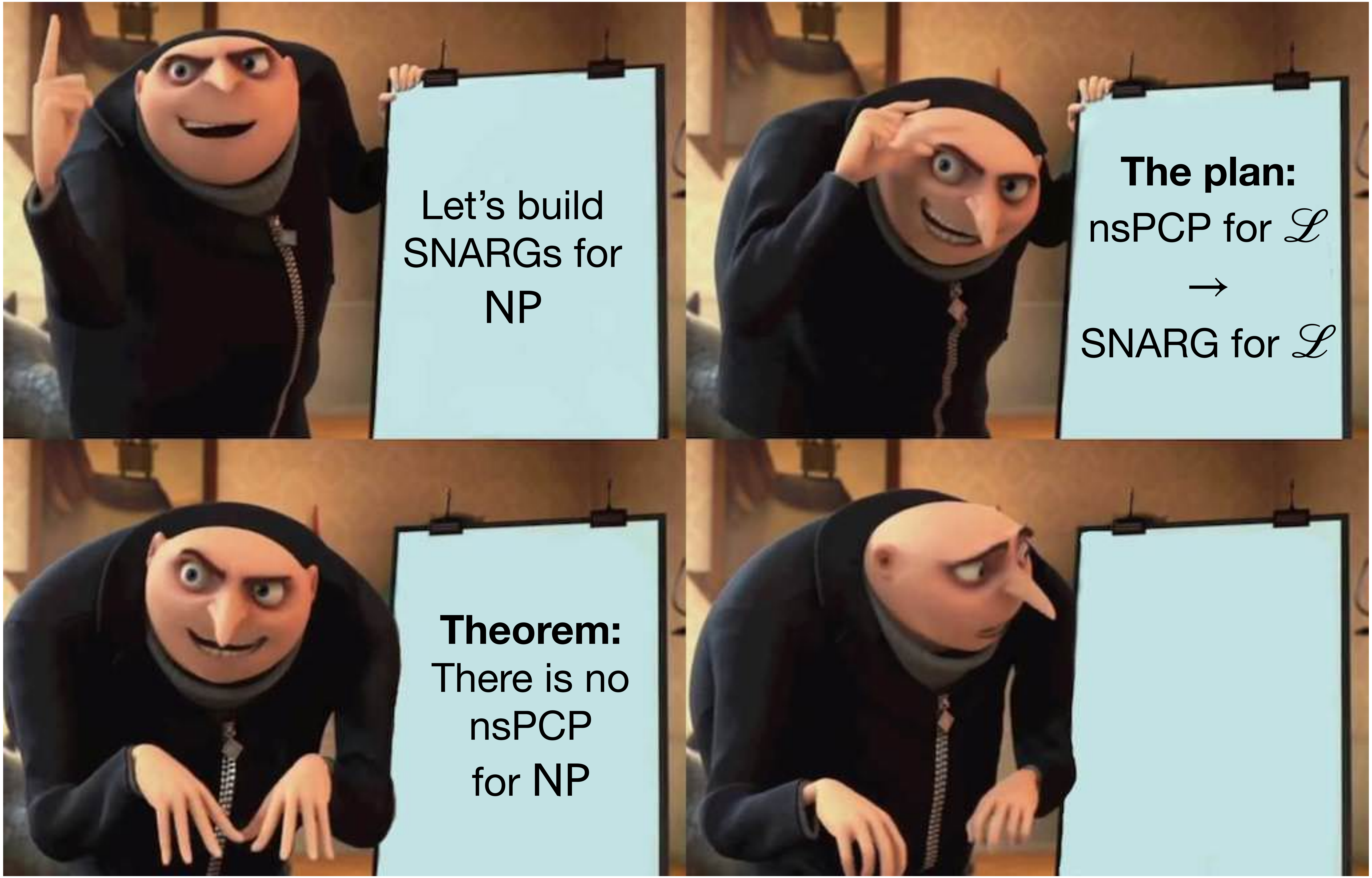
What about NP?

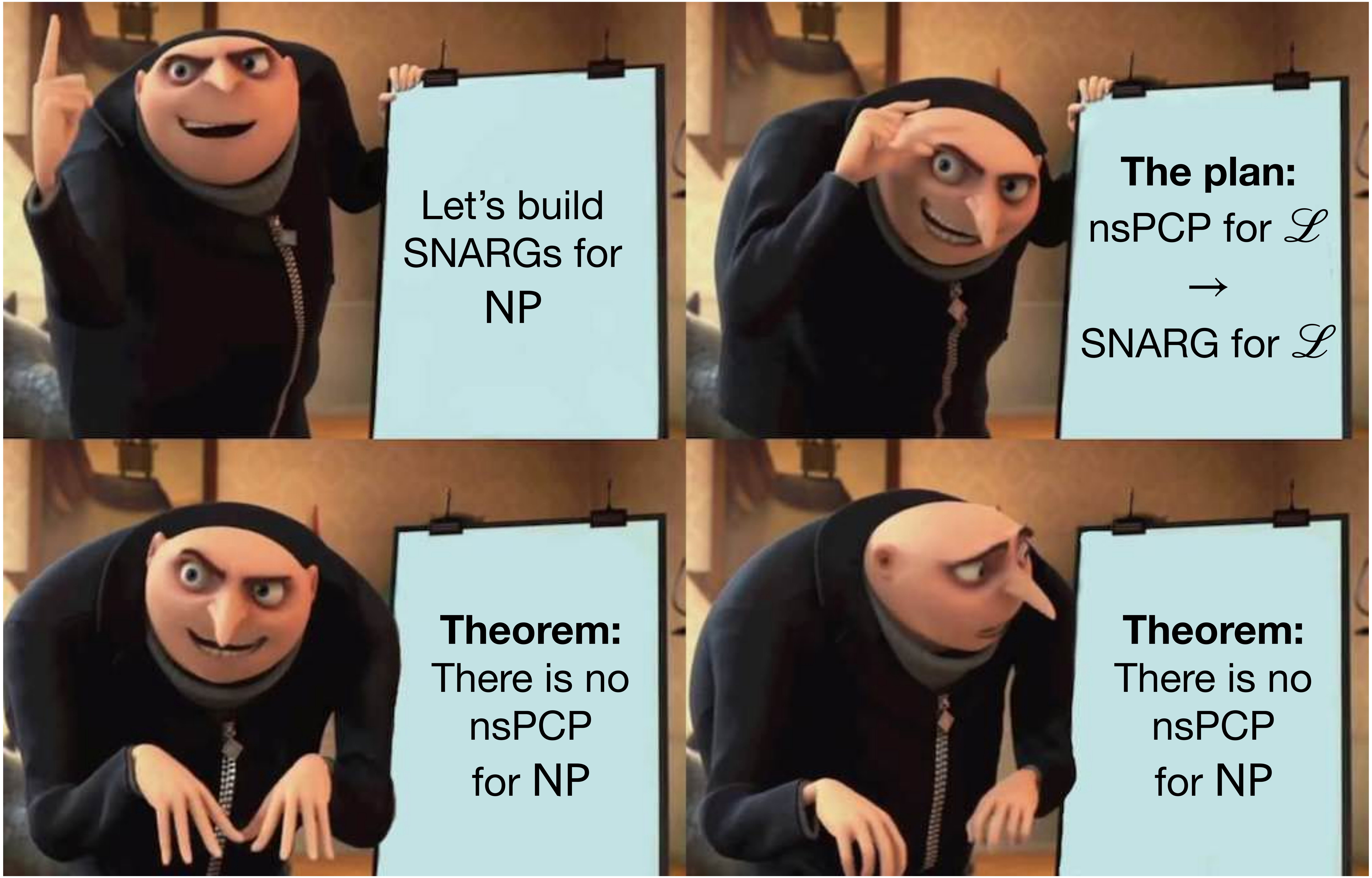
Any questions so far?

Any questions so far?

Any questions so far?


Let's build SNARGs for NP
The plan:
nsPCP for $\mathscr{L}$
$\rightarrow$
SNARG for $\mathscr{L}$


Let's build SNARGs for NP
The plan: nsPCP for $\mathscr{L}$ → SNARG for $\mathscr{L}$
Theorem: There is no nsPCP for NP



Let's build SNARGs for NP
The plan: nsPCP for $\mathscr{L}$ $\rightarrow$ SNARG for $\mathscr{L}$
Theorem: There is no nsPCP for NP
Theorem: There is no nsPCP for NP

The Non-Signaling Barrier

The Non-Signaling Barrier

- **Theorem.** Assuming SAT requires $2^{O(n)}$ time, there is **no efficient NS PCP for NP**.

The Non-Signaling Barrier

- **Theorem.** Assuming SAT requires $2^{O(n)}$ time, there is **no efficient NS PCP for NP**.

- **Proof:** Suppose there exists a PCP of length ℓ with locality k. Here is an algorithm for SAT:

The Non-Signaling Barrier

- **Theorem.** Assuming SAT requires $2^{O(n)}$ time, there is **no efficient NS PCP for NP**.
- **Proof:** Suppose there exists a PCP of length ℓ with locality k. Here is an algorithm for SAT:
  - Compute an PCP $\ell^{O(k)}$-size **linear program** to determine if the PCP has a successful non-signaling strategy for x. Output 1 if yes, 0 otherwise.

The Non-Signaling Barrier

- **Theorem.** Assuming SAT requires $2^{O(n)}$ time, there is **no efficient NS PCP for NP**.

- **Proof:** Suppose there exists a PCP of length ℓ with locality k. Here is an algorithm for SAT:

    - Compute an PCP $\ell^{O(k)}$-size **linear program** to determine if the PCP has a successful non-signaling strategy for x. Output 1 if yes, 0 otherwise.

#Variables in LP correspond (roughly) to all possible query sets Q to the PCP

The Non-Signaling Barrier

- **Theorem.** Assuming SAT requires $2^{O(n)}$ time, there is **no efficient NS PCP for NP**.

- **Proof:** Suppose there exists a PCP of length ℓ with locality k. Here is an algorithm for SAT:

  - Compute an PCP $\ell^{O(k)}$-size **linear program** to determine if the PCP has a successful non-signaling strategy for x. Output 1 if yes, 0 otherwise.

The Non-Signaling Barrier

- **Theorem.** Assuming SAT requires $2^{O(n)}$ time, there is **no efficient NS PCP for NP**.

- **Proof:** Suppose there exists a PCP of length ℓ with locality k. Here is an algorithm for SAT:

    - Compute an PCP $\ell^{O(k)}$-size **linear program** to determine if the PCP has a successful non-signaling strategy for x. Output 1 if yes, 0 otherwise.

- For $\ell = \text{poly(n)}$ and $k \ll n$, $\ell^{O(k)} \ll 2^{O(n)}$. Contradiction ∎

The Non-Signaling Barrier

- **Theorem.** Assuming SAT requires $2^{O(n)}$ time, there is **no efficient NS PCP for NP**.
- **Proof:** Suppose there exists a PCP of length ℓ with locality k. Here is an algorithm for SAT:
  - Compute an PCP $\ell^{O(k)}$-size **linear program** to determine if the PCP has a successful non-signaling strategy for x. Output 1 if yes, 0 otherwise.
- For ℓ = poly(n) and $k \ll n$, $\ell^{O(k)} \ll 2^{O(n)}$. Contradiction ∎

THE END

THE END

~~THE END~~

Our observation:

What if ℓ is $2^{O(n)}$? Then $\ell^k \geq 2^{O(n)}$.

There is no contradiction!

~~THE END~~

This work: There is more to be done!

This work: There is more to be done!

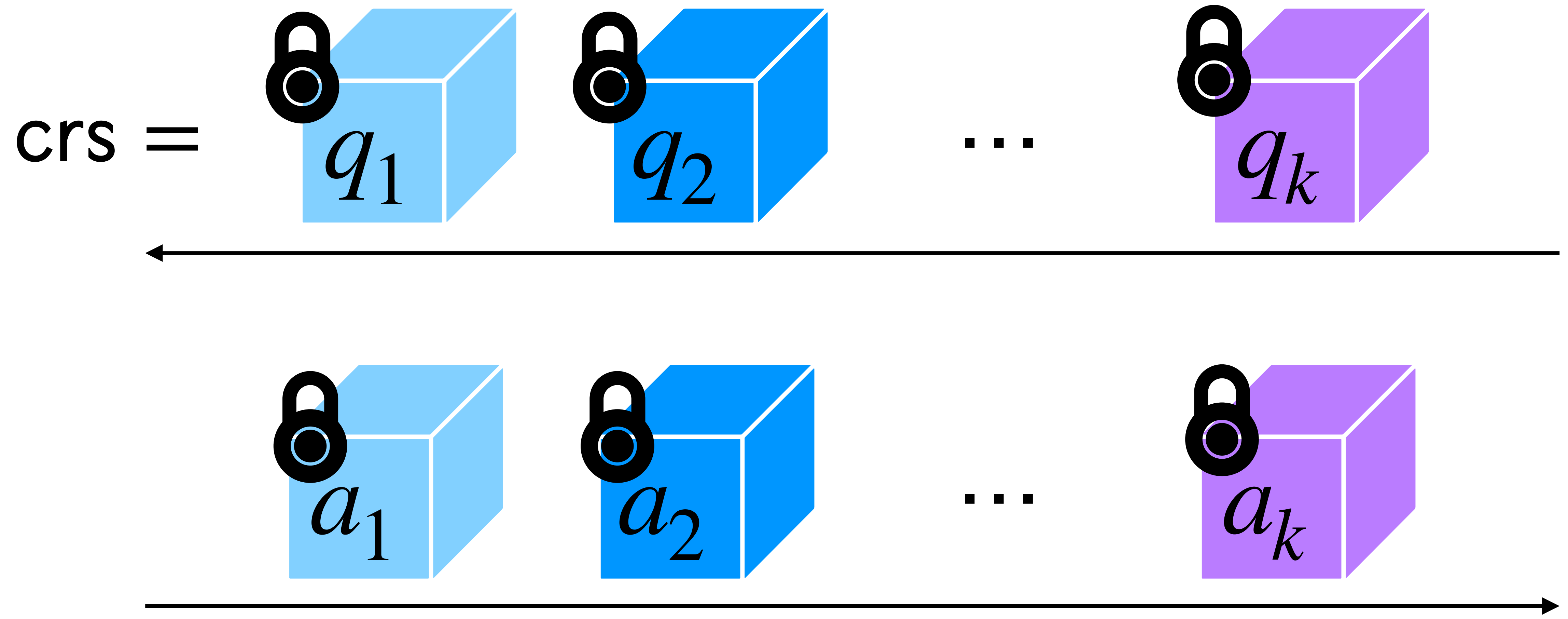
- What if… the PCP had length $\ell = 2^{O(n)}$? Impossibility goes away!

This work: There is more to be done!

- What if… the PCP had length $\ell = 2^{O(n)}$? Impossibility goes away!
- But… Prover needs to have the PCP in his hand.

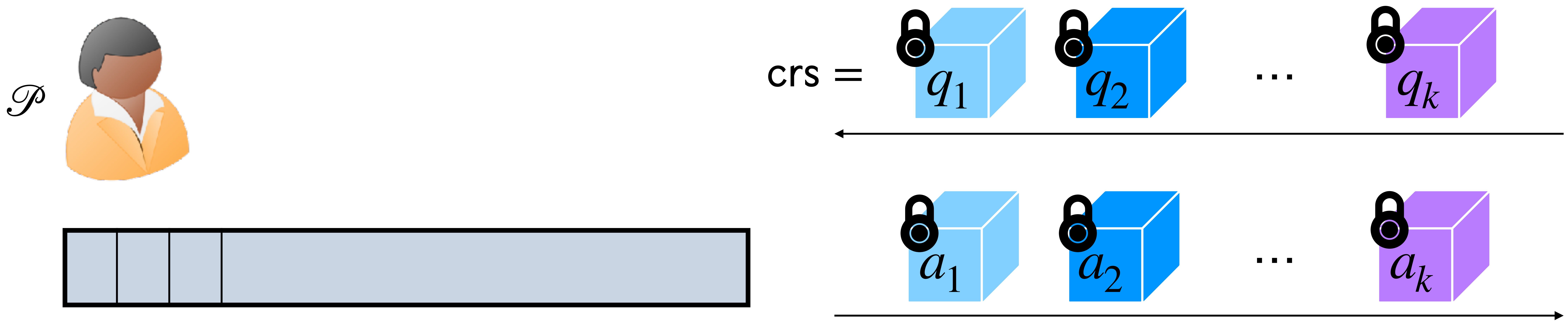
This work: There is more to be done!

- What if… the PCP had length $\ell = 2^{O(n)}$? Impossibility goes away!
- But… Prover needs to have the PCP in his hand.



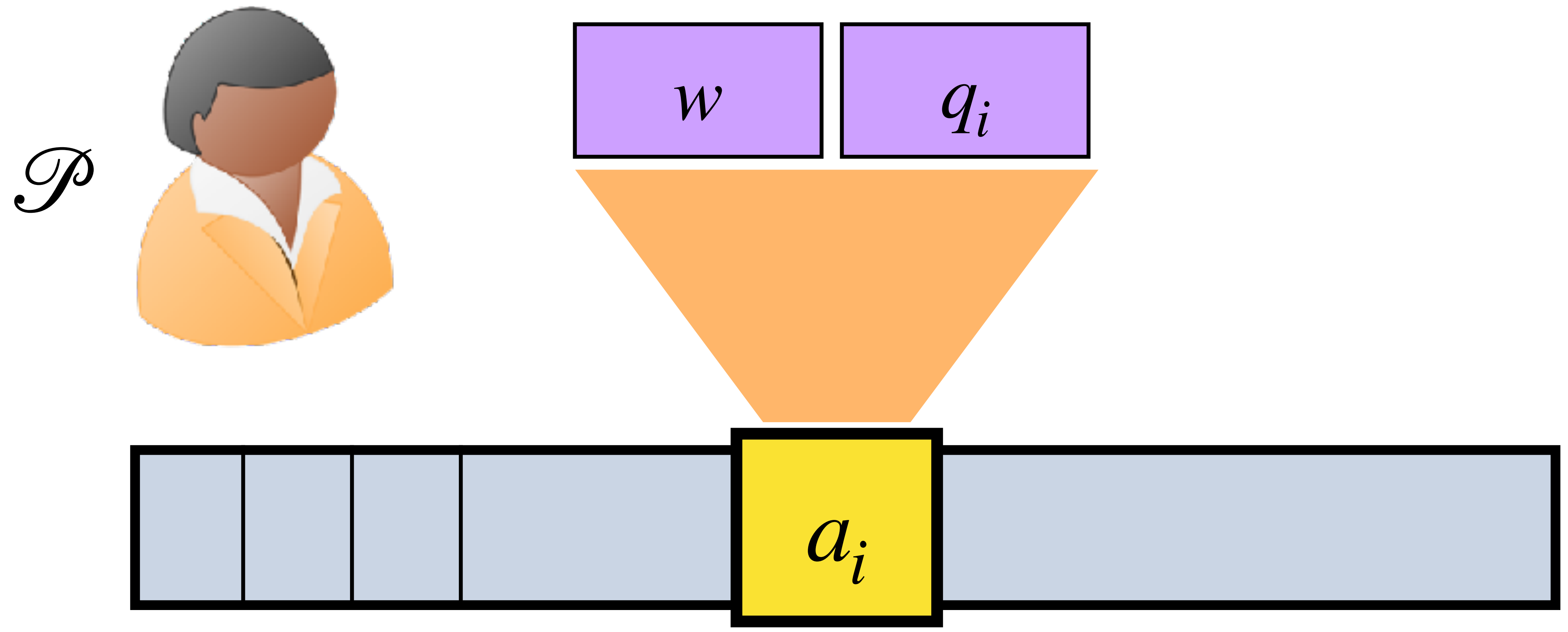
This work: There is more to be done!

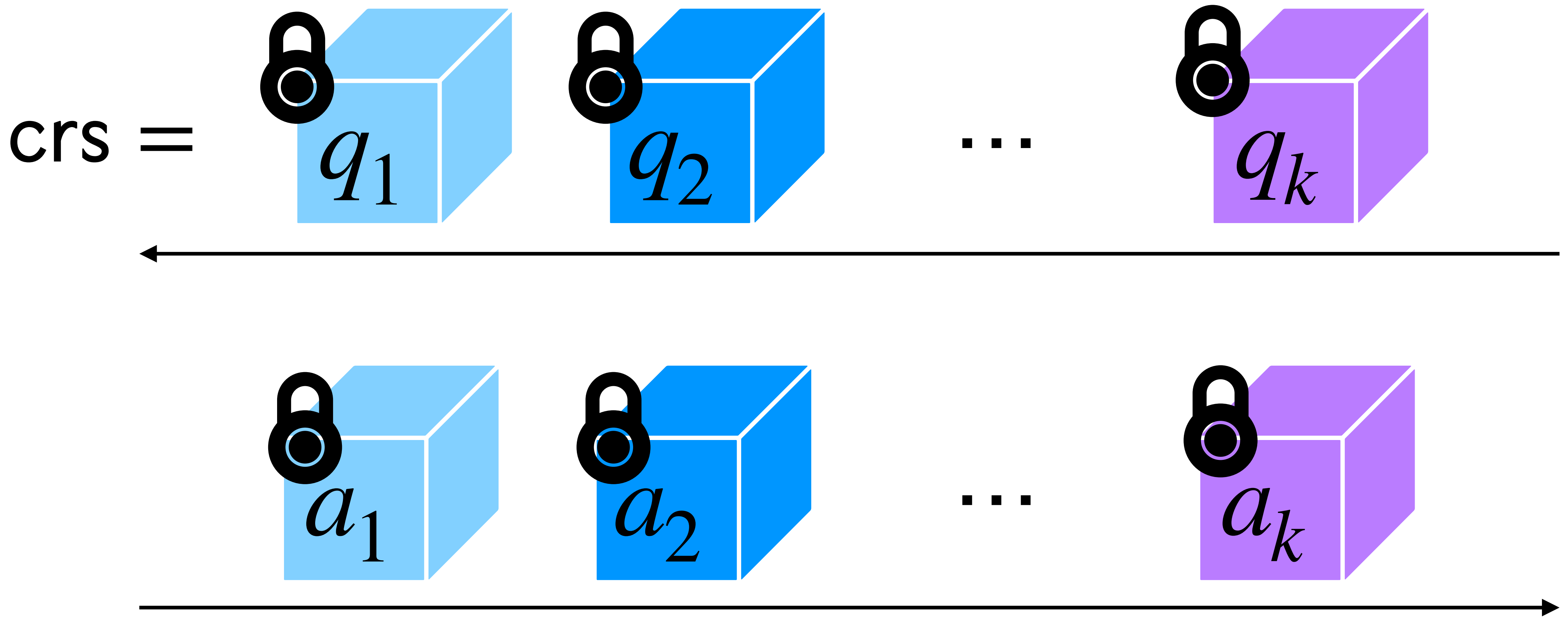
- What if… the PCP had length $\ell = 2^{O(n)}$? Impossibility goes away!
- But… Prover needs to have the PCP in his hand.



This work: There is more to be done!

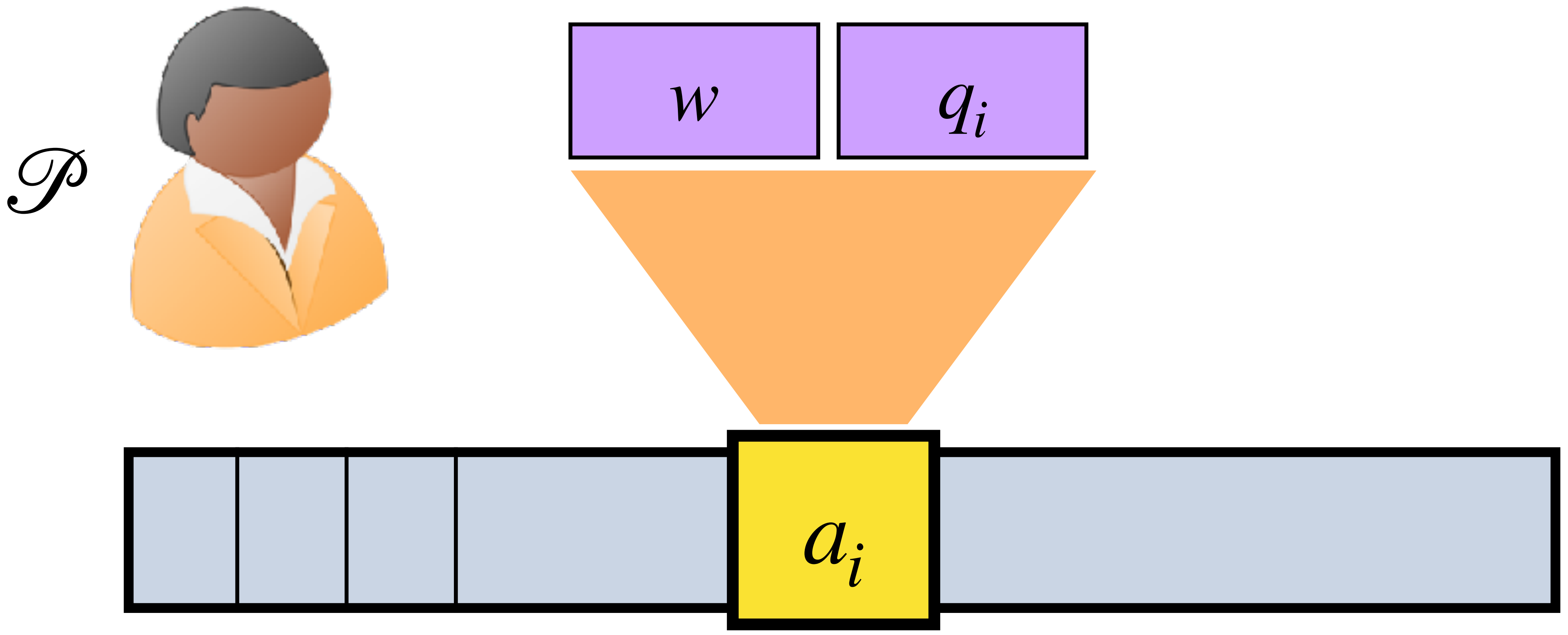
- What if… the PCP had length $\ell = 2^{O(n)}$? Impossibility goes away!
- But… Prover needs to have the PCP in his hand.

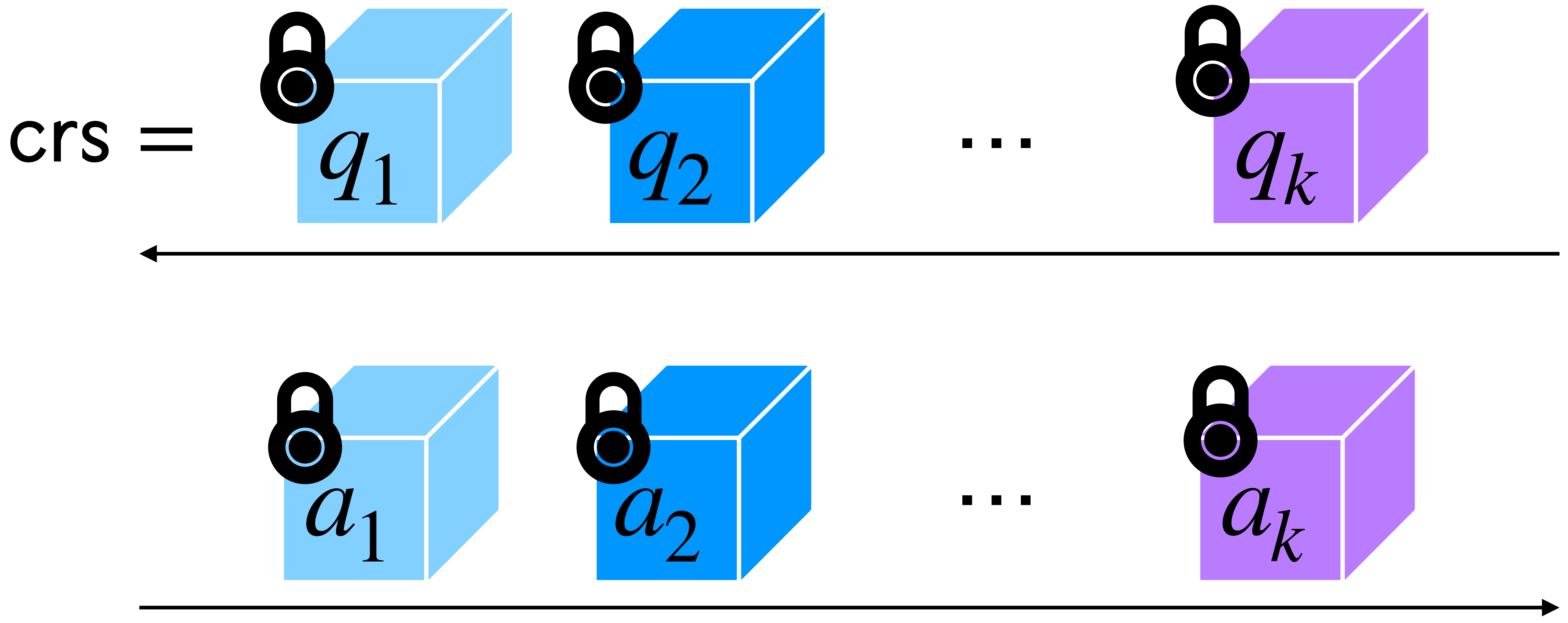




This work: There is more to be done!

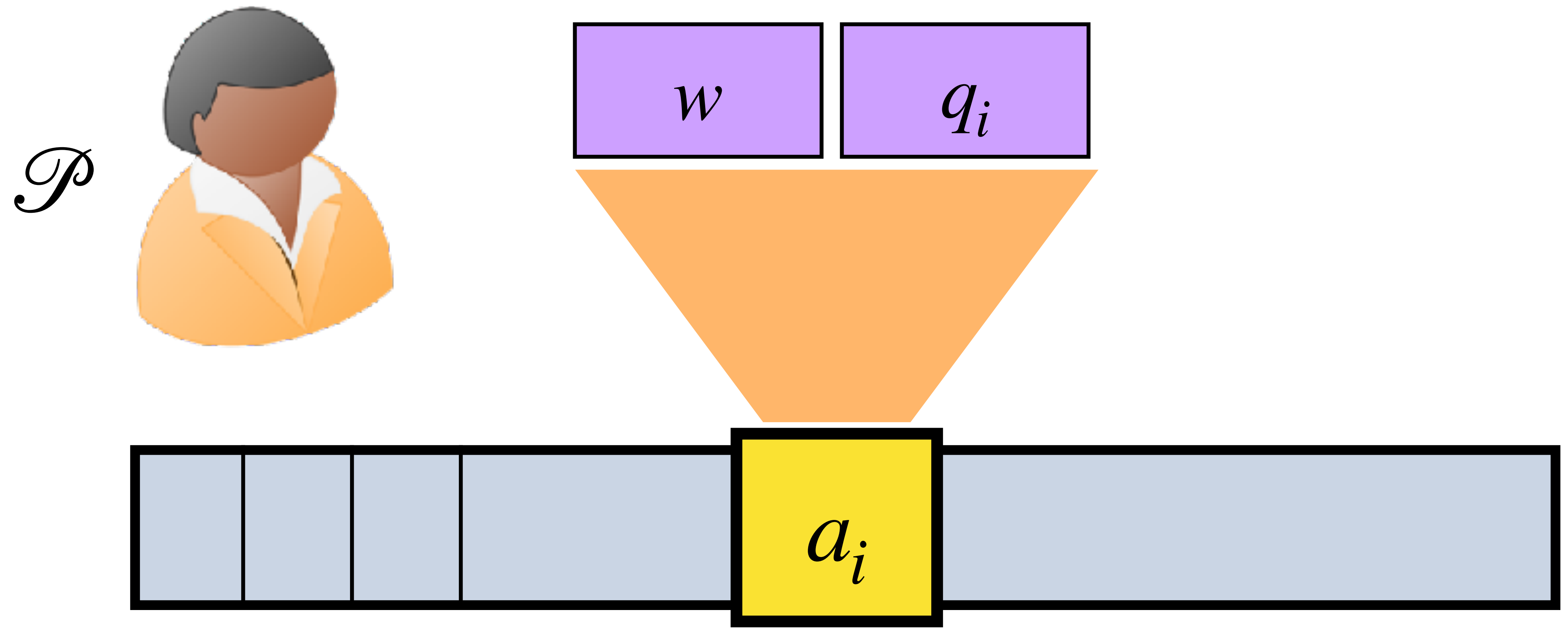
- What if… the PCP had length $\ell = 2^{O(n)}$? Impossibility goes away!
- But… Prover needs to have the PCP in his hand.
- Sufficient if each entry of the PCP can be computed locally!!!

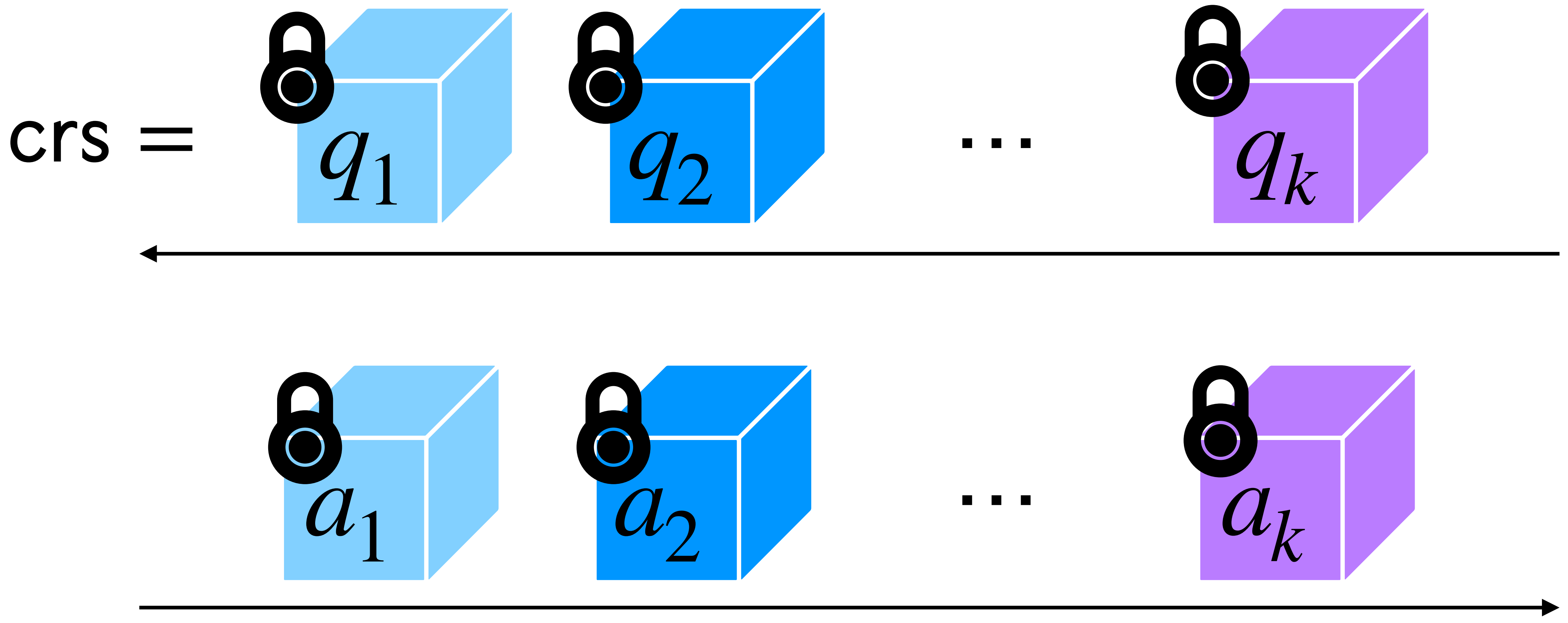




This work: There is more to be done!

- What if… the PCP had length $\ell = 2^{O(n)}$? Impossibility goes away!
- But… Prover needs to have the PCP in his hand.
- Sufficient if each entry of the PCP can be computed locally!!!





Each $|q_i| = O(n)$, $|a_i| = O(1)$

This work

This work

- **Theorem 1.** There exists a SNARG for NP assuming
  - Fully homomorphic encryption (LWE).
  - There **exists** an exponential-length *"nice"* non-signaling PCP for NP with **"weak soundness".**

This work

- **Theorem 1.** There exists a SNARG for NP assuming
  - Fully homomorphic encryption (LWE).
  - There **exists** an exponential-length *"nice"* non-signaling PCP for NP with **"weak soundness".**
- **"Nice"**: Correctness can be proven in propositional logic (Extended Frege).

This work

- **Theorem 1.** There exists a SNARG for NP assuming
  - Fully homomorphic encryption (LWE).
  - There **exists** an exponential-length *"nice"* non-signaling PCP for NP with **"weak soundness".**
- **"Nice"**: Correctness can be proven in propositional logic (Extended Frege).
  - Ask me later

This work

- **Theorem 1.** There exists a SNARG for NP assuming
  - Fully homomorphic encryption (LWE).
  - There **<u>exists</u>** an exponential-length *"nice"* non-signaling PCP for NP with **"weak soundness".**

- **"Nice"**: Correctness can be proven in propositional logic (Extended Frege).
  - Ask me later
- **Construction:** KRR14 + "Encrypt-Hash-and-BARG" ([JKLV24, JKL**M**25])

This work

- **Theorem 1.** There exists a SNARG for NP assuming
  - Fully homomorphic encryption (LWE).
  - There **exists** an exponential-length *"nice"* non-signaling PCP for NP with **"weak soundness".**

- **"Nice"**: Correctness can be proven in propositional logic (Extended Frege).
  - Ask me later
- **Construction:** KRR14 + "Encrypt-Hash-and-BARG" ([JKLV24, JKL**M**25])
  - [JKL**M**25]: Any "nice" designated-verifier SNARG can be boosted to be publicly verifiable.

Soundness vs. Weak Soundness

Soundness vs. Weak Soundness

- Suppose $x \notin \mathscr{L}$.

Soundness vs. Weak Soundness

- Suppose $x \notin \mathscr{L}$.
- **Usual NS Soundness:** For all NS strategies $\mathscr{D} = \{D_Q\}_Q$,

$$\Pr_Q[V(x, Q, A) = 1 \,|\, A \leftarrow D_Q] \leq \frac{1}{\mathsf{poly}(n)}.$$

Soundness vs. Weak Soundness

- Suppose $x \notin \mathscr{L}$.
- **Usual NS Soundness:** For all NS strategies $\mathscr{D} = \{D_Q\}_Q$,

$$\Pr_Q[V(x, Q, A) = 1 \,|\, A \leftarrow D_Q] \leq \frac{1}{\text{poly}(n)}.$$

- **Weak NS soundness:** For all NS strategies $\mathscr{D} = \{D_Q\}_Q$, **there exists any Q** such that

$$\Pr[V(x, Q, A) = 1 \,|\, A \leftarrow D_Q] \leq 1 - \frac{1}{\text{poly}(n)}.$$

Soundness vs. Weak Soundness

- Suppose $x \notin \mathscr{L}$.
- **Usual NS Soundness:** For all NS strategies $\mathscr{D} = \{D_Q\}_Q$,

  $$\Pr_{Q}[V(x, Q, A) = 1 \,|\, A \leftarrow D_Q] \leq \frac{1}{\text{poly}(n)}.$$

- **Weak NS soundness:** For all NS strategies $\mathscr{D} = \{D_Q\}_Q$, **there exists any Q** such that

  $$\Pr[V(x, Q, A) = 1 \,|\, A \leftarrow D_Q] \leq 1 - \frac{1}{\text{poly}(n)}.$$

Soundness vs. Weak Soundness

- Suppose $x \notin \mathscr{L}$.
- **Usual NS Soundness:** For all NS strategies $\mathscr{D} = \{D_Q\}_Q$,

$$\Pr_{Q}[V(x, Q, A) = 1 \,|\, A \leftarrow D_Q] \leq \frac{1}{\mathrm{poly}(n)}.$$

"Average-case soundness"

- **Weak NS soundness:** For all NS strategies $\mathscr{D} = \{D_Q\}_Q$, **there exists any Q** such that

$$\Pr[V(x, Q, A) = 1 \,|\, A \leftarrow D_Q] \leq 1 - \frac{1}{\mathrm{poly}(n)}.$$

Soundness vs. Weak Soundness

- Suppose $x \notin \mathscr{L}$.
- **Usual NS Soundness:** For all NS strategies $\mathscr{D} = \{D_Q\}_Q$,

$$\Pr_{Q}[V(x, Q, A) = 1 \,|\, A \leftarrow D_Q] \leq \frac{1}{\text{poly}(n)}.$$

“Average-case soundness”

- **Weak NS soundness:** For all NS strategies $\mathscr{D} = \{D_Q\}_Q$, there exists any Q such that

$$\Pr[V(x, Q, A) = 1 \,|\, A \leftarrow D_Q] \leq 1 - \frac{1}{\text{poly}(n)}.$$

“Worst-case soundness”

Soundness vs. Weak Soundness

- Suppose $x \notin \mathscr{L}$.
- **Usual NS Soundness:** For all NS strategies $\mathscr{D} = \{D_Q\}_Q$,

$$\Pr_Q[V(x, Q, A) = 1 \,|\, A \leftarrow D_Q] \leq \frac{1}{\mathsf{poly}(n)}.$$

"Average-case soundness"

- **Weak NS soundness:** For all NS strategies $\mathscr{D} = \{D_Q\}_Q$, there exists any Q such that

$$\Pr[V(x, Q, A) = 1 \,|\, A \leftarrow D_Q] \leq 1 - \frac{1}{\mathsf{poly}(n)}.$$

"Worst-case soundness"

Intuition: We show that the compiler from [JKLM25] is sound even if the underlying dvSNARG has "worst-case soundness"

Weakly sound nsPCP?

Weakly sound nsPCP?

- No unconditional result :(We invite you to help us :)

Weakly sound nsPCP?

- No unconditional result :(We invite you to help us :)
- **Theorem 2:** Hadamard PCP has weak NS soundness under Conjecture 3.

Weakly sound nsPCP?

- No unconditional result :(We invite you to help us :)
- **Theorem 2:** Hadamard PCP has weak NS soundness under Conjecture 3.
- Let $R = \mathbb{R}(x_1, \ldots, x_k) / \langle x_1^2 - 1, \ldots, x_k^2 - 1 \rangle$.

Weakly sound nsPCP?

- No unconditional result :(We invite you to help us :)
- **Theorem 2:** Hadamard PCP has weak NS soundness under Conjecture 3.
- Let $R = \mathbb{R}(x_1, \ldots, x_k)/\langle x_1^2 - 1, \ldots, x_k^2 - 1 \rangle$.

Conjecture 3 ("Low-Norm Nullstellensatz"): Consider a *special set* $P = \{p_1, \ldots, p_t\} \in R$, let $V(P) \subseteq \{\pm 1\}^k$ be the common zeros. For all $f(x_1, \ldots, x_k)$ that vanish on $V(P)$, there exists $\{q_i\}_i$ such that

Weakly sound nsPCP?

- No unconditional result :(We invite you to help us :)
- **Theorem 2:** Hadamard PCP has weak NS soundness under Conjecture 3.
- Let $R = \mathbb{R}(x_1, \ldots, x_k)/\langle x_1^2 - 1, \ldots, x_k^2 - 1 \rangle$.

Conjecture 3 ("Low-Norm Nullstellensatz"): Consider a *special set* $P = \{p_1, \ldots, p_t\} \in R$, let $V(P) \subseteq \{\pm 1\}^k$ be the common zeros. For all $f(x_1, \ldots, x_k)$ that vanish on $V(P)$, there exists $\{q_i\}_i$ such that

$$f = \sum_i p_i q_i \text{ and } \sum_i ||q_i||_1 \leq \text{poly}(n) \cdot ||f||_1 .$$

Weakly sound nsPCP?

- No unconditional result :(We invite you to help us :)
- **Theorem 2:** Hadamard PCP has weak NS soundness under Conjecture 3.
- Let $R = \mathbb{R}(x_1, \ldots, x_k)/\langle x_1^2 - 1, \ldots, x_k^2 - 1 \rangle$.

Conjecture 3 (Hilbert's Nullstellensatz”): Consider a *special set* $P = \{p_1, \ldots, p_t\} \in R$, let $V(P) \subseteq \{\pm 1\}^k$ be the common zeros. For all $f(x_1, \ldots, x_k)$ that vanish on $V(P)$, there exists $\{q_i\}_i$ such that

$$f = \sum_i p_i q_i \text{ and } \sum_i \cancel{||q_i||_1 \leq \text{poly}(n) \cdot ||f||_1}.$$

Weakly sound nsPCP?

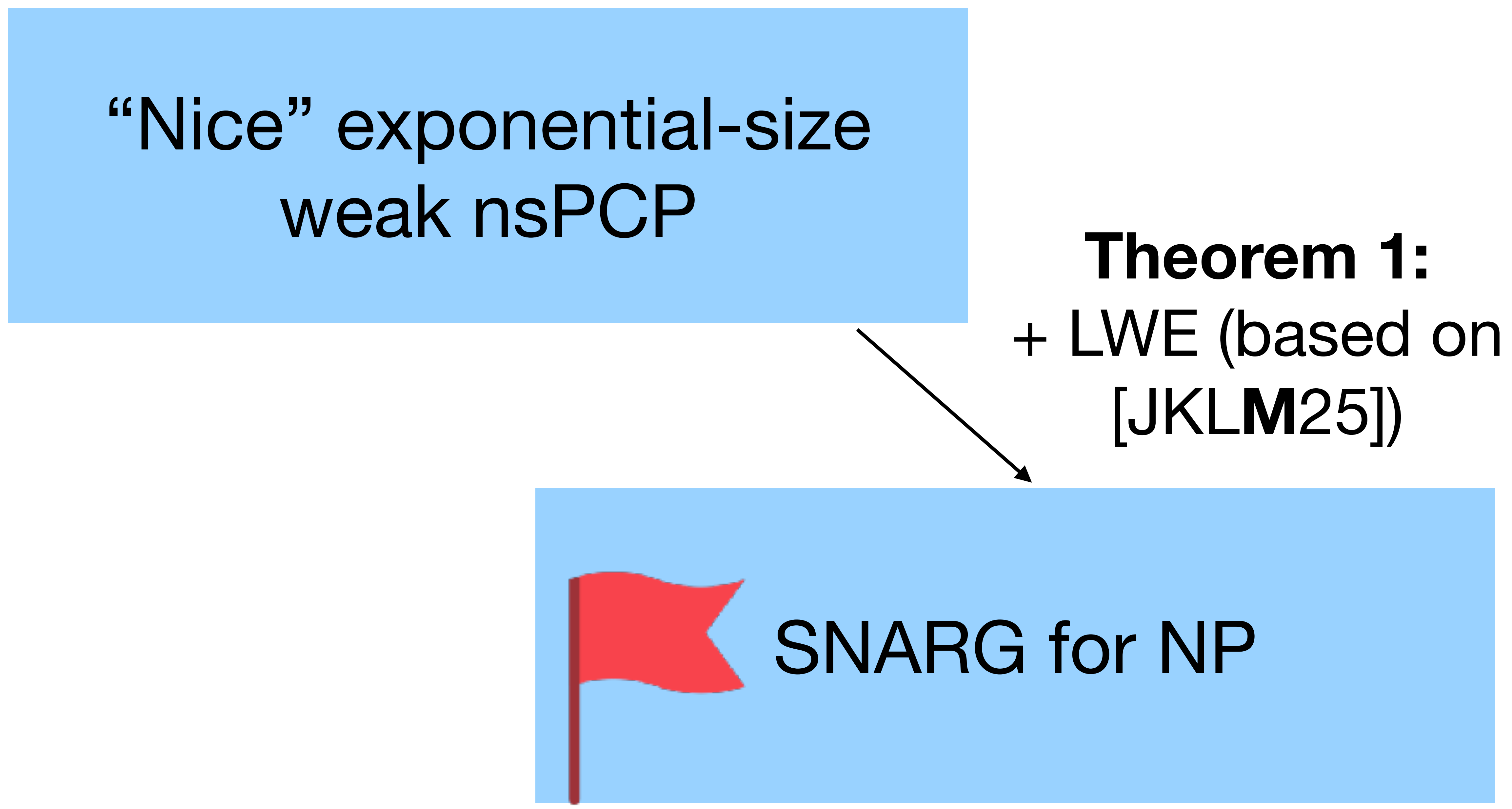
- No unconditional result :(We invite you to help us :)
- **Theorem 2:** Hadamard PCP has weak NS soundness under Conjecture 3.
- Let $R = \mathbb{R}(x_1, \ldots, x_k)/\langle x_1^2 - 1, \ldots, x_k^2 - 1 \rangle$.

Conjecture 3 ("Low-Norm Nullstellensatz"): Consider a *special set* $P = \{p_1, \ldots, p_t\} \in R$, let $V(P) \subseteq \{\pm 1\}^k$ be the common zeros. For all $f(x_1, \ldots, x_k)$ that vanish on $V(P)$, there exists $\{q_i\}_i$ such that

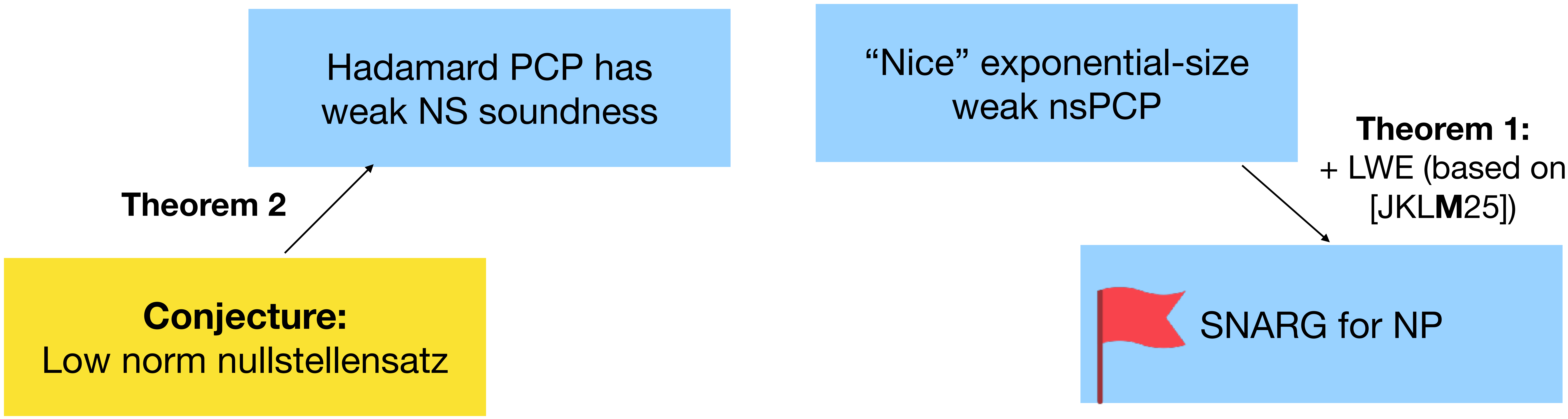
$$f = \sum_i p_i q_i \text{ and } \sum_i ||q_i||_1 \leq \text{poly}(n) \cdot ||f||_1 .$$

Roadmap of this work

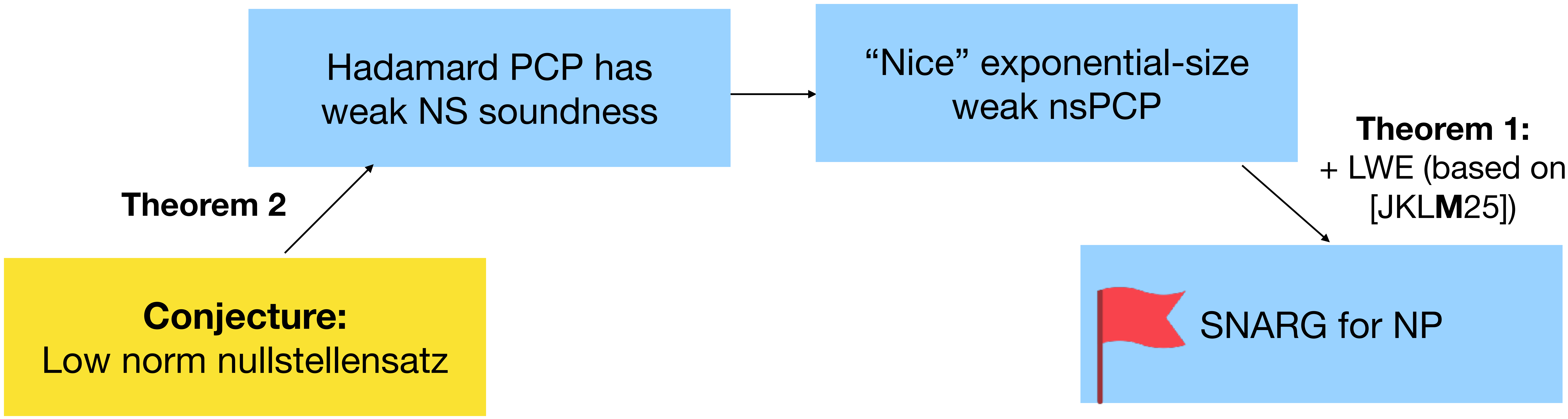
Roadmap of this work



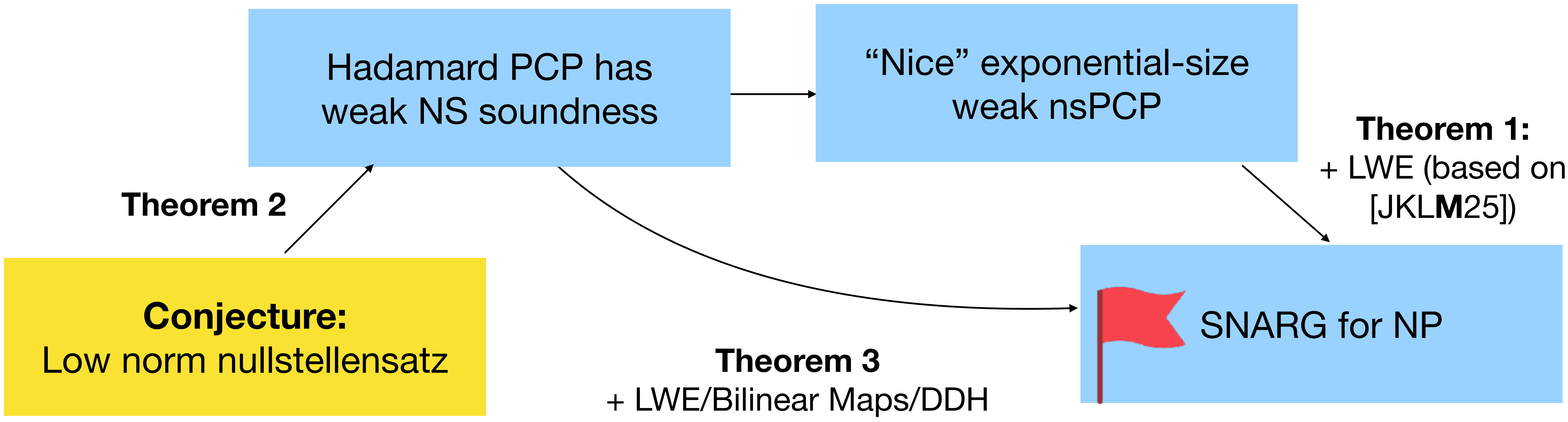
Roadmap of this work



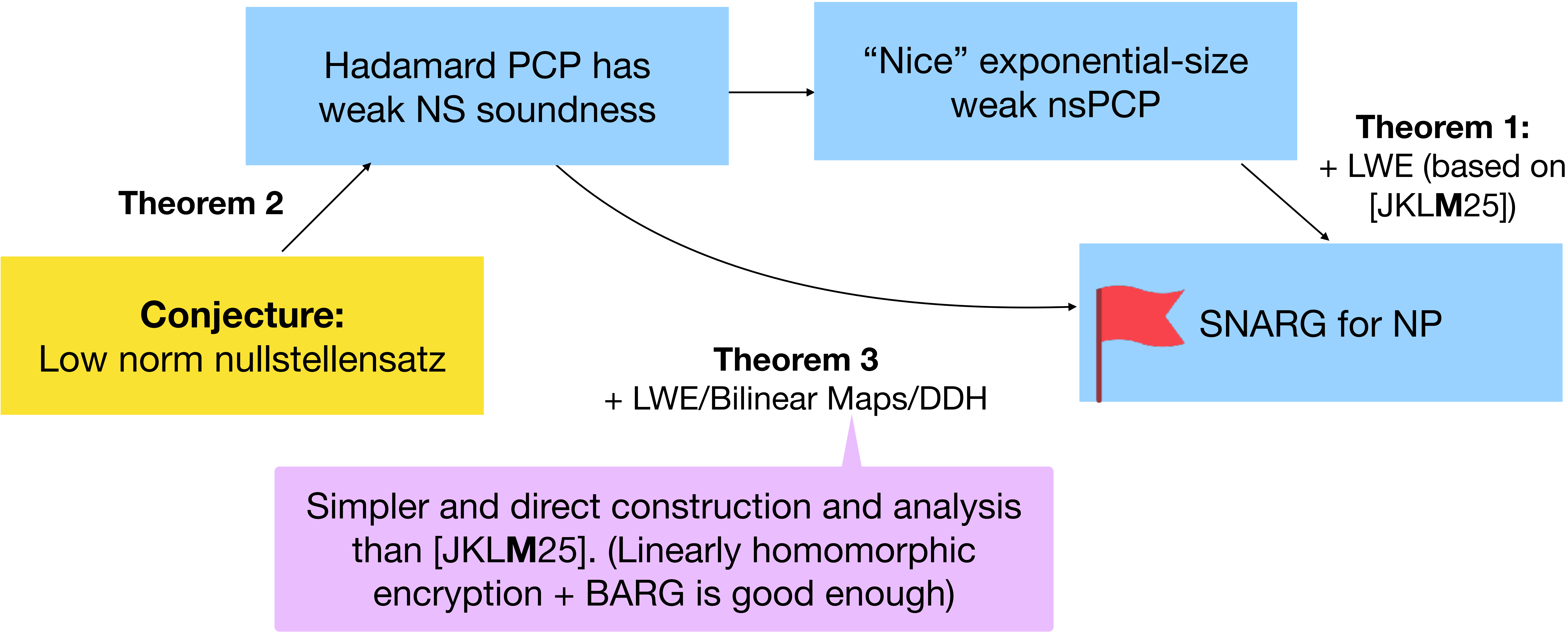
Roadmap of this work



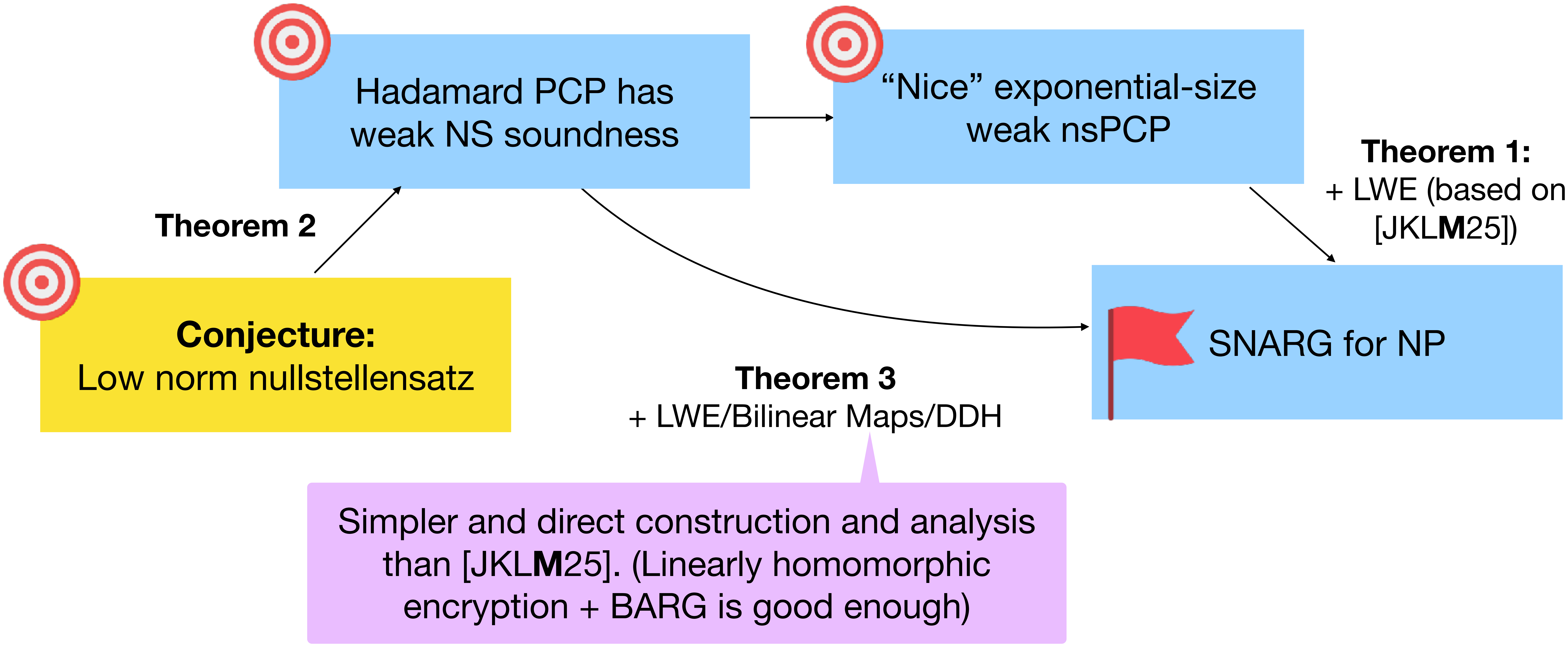
Roadmap of this work



Roadmap of this work



Roadmap of this work



Rest of the Talk

Rest of the Talk

Theorem 4 (Unconditional).
For Hadamard PCP, no NS strategy can **perfectly** satisfy every test.

Rest of the Talk

Theorem 4 (Unconditional).
For Hadamard PCP, no NS strategy can **perfectly** satisfy every test.

i.e. For any NS strategy $\mathscr{D} = \{D_Q\}_Q$, there exists Q such that

Rest of the Talk

Theorem 4 (Unconditional).
For Hadamard PCP, no NS strategy can **perfectly** satisfy every test.

i.e. For any NS strategy $\mathscr{D} = \{D_Q\}_Q$, there exists Q such that

$$\Pr[V(x, Q, A) = 1 \,|\, A \leftarrow D_Q] < 1.$$

Rest of the Talk

Theorem 4 (Unconditional).
For Hadamard PCP, no NS strategy can **perfectly** satisfy every test.

i.e. For any NS strategy $\mathscr{D} = \{D_Q\}_Q$, there exists Q such that

$$\Pr[V(x, Q, A) = 1 \,|\, A \leftarrow D_Q] < 1.$$

Recall Goal: $\Pr[V(x, Q, A) = 1 \,|\, A \leftarrow D_Q] \leq 1 - \frac{1}{\mathrm{poly}(n)}.$

Quadratic Equations

Quadratic Equations

- **NP Language:** QuadEq.

Quadratic Equations

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.

Quadratic Equations

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

Quadratic Equations

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

$$w_1 - w_2 + w_{34} = 0$$

$$\cdots$$

$$w_{56} + w_{67} + w_{78} = 1$$

Instance

Hadamard PCP, simplified

[Babai-Fortnow-Lund '91]

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

Hadamard PCP, simplified

[Babai-Fortnow-Lund '91]

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

Hadamard PCP, simplified

[Babai-Fortnow-Lund '91]

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:

Hadamard PCP, simplified

[Babai-Fortnow-Lund '91]

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:

Hadamard PCP, simplified

[Babai-Fortnow-Lund '91]

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:

Hadamard PCP, simplified

[Babai-Fortnow-Lund ’91]

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:

(Indicator vectors q)

Hadamard PCP, simplified

[Babai-Fortnow-Lund ’91]

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:

(Indicator vectors q)

$k = O(1)$

Hadamard PCP, simplified

[Babai-Fortnow-Lund '91]

- Verifier on $q_1, q_2, \ldots, q_k$ checks:

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:

(Indicator vectors q)

$k = O(1)$

Hadamard PCP, simplified

[Babai-Fortnow-Lund '91]

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:

(Indicator vectors q)

$k = O(1)$

Hadamard PCP, simplified

[Babai-Fortnow-Lund ’91]

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:

(Indicator vectors q)

$k = O(1)$

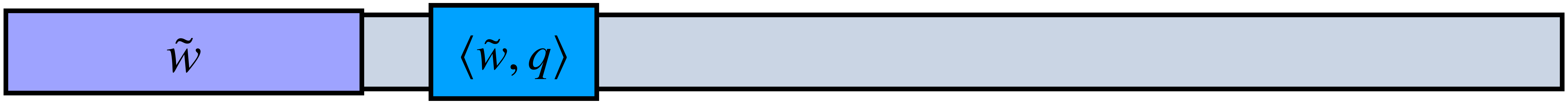
Hadamard PCP, simplified

[Babai-Fortnow-Lund '91]

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:



(Indicator vectors q)

$k = O(1)$

Hadamard PCP, simplified

[Babai-Fortnow-Lund '91]

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:

(Indicator vectors q)

$k = O(1)$

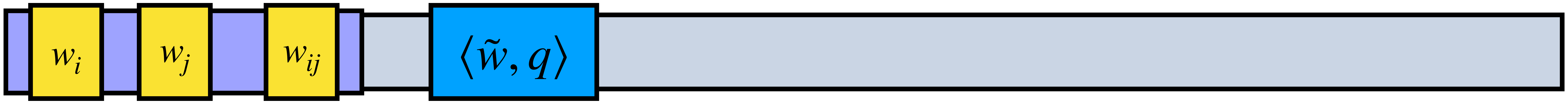
Hadamard PCP, simplified

[Babai-Fortnow-Lund ’91]

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:



(Indicator vectors q)

$k = O(1)$

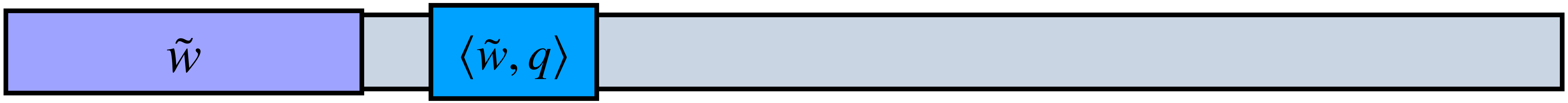
Hadamard PCP, simplified

[Babai-Fortnow-Lund '91]

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:



(Indicator vectors q)

$k = O(1)$

Hadamard PCP, simplified

[Babai-Fortnow-Lund '91]

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:

(Indicator vectors q)

$k = O(1)$

Hadamard PCP, simplified

[Babai-Fortnow-Lund '91]

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:

(Indicator vectors q)

$k = O(1)$

Hadamard PCP, simplified

[Babai-Fortnow-Lund '91]

E.g. Contains
$w_j - w_j + w_k = 0$

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:

(Indicator vectors q)

$k = O(1)$

Hadamard PCP, simplified

[Babai-Fortnow-Lund '91]

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

- **NP Language:** QuadEq.
  - **Instance:** Set of 3-local linear equations on $N = n + \binom{n}{2}$ variables.
  - **Witnesses:** $\tilde{w} = \{w_i\}_i \cup \{w_{ij}\}_{ij}$ satisfying above equations and $w_{ij} = w_i \cdot w_j$.

For $q \in \{0,1\}^N$:

(Indicator vectors q)

$k = O(1)$

Wishful Thinking (fake proof)

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

Wishful Thinking (fake proof)

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

Wishful Thinking (fake proof)

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

- **Step 1:** Any NS strategy satisfying linear and quadratic consistency can be written as probabilities over **"correct" encodings**, i.e. $\{\text{Had}(\tilde{w})\}_{w \in \{0,1\}^n}$.

Wishful Thinking (fake proof)

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

- **Step 1:** Any NS strategy satisfying linear and quadratic consistency can be written as probabilities over **"correct" encodings**, i.e. $\{\mathrm{Had}(\tilde{w})\}_{w \in \{0,1\}^n}$.

p_1 Had($\tilde{w}_1$)

p_2 Had($\tilde{w}_2$)

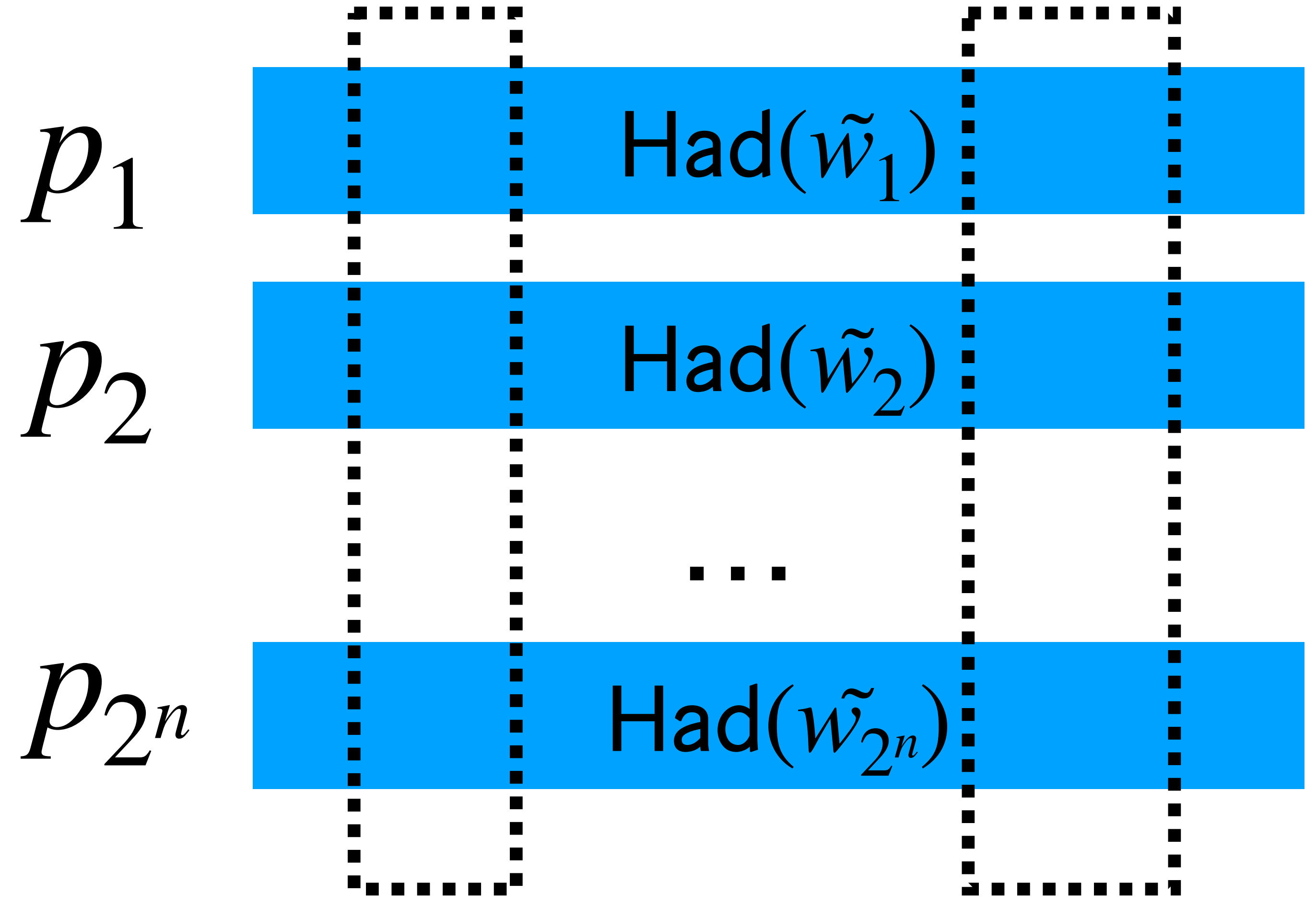
…

p_{2^n} Had($\tilde{w}_{2^n}$)

Wishful Thinking (fake proof)

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

- **Step 1:** Any NS strategy satisfying linear and quadratic consistency can be written as probabilities over **"correct" encodings**, i.e. $\{\text{Had}(\tilde{w})\}_{w \in \{0,1\}^n}$.



D_Q corresponds to the marginals.

Wishful Thinking (fake proof)

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

- **Step 1:** Any NS strategy satisfying linear and quadratic consistency can be written as probabilities over **"correct" encodings**, i.e. $\{\text{Had}(\tilde{w})\}_{w \in \{0,1\}^n}$.

Wishful Thinking (fake proof)

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

- **Step 1:** Any NS strategy satisfying linear and quadratic consistency can be written as probabilities over **"correct" encodings**, i.e. $\{\text{Had}(\tilde{w})\}_{w \in \{0,1\}^n}$.

Wishful Thinking (fake proof)

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

- **Step 1:** Any NS strategy satisfying linear and quadratic consistency can be written as probabilities over **"correct" encodings**, i.e. $\{\text{Had}(\tilde{w})\}_{w \in \{0,1\}^n}$.

- **Step 2:** If $x \notin \mathscr{L}$, every encoding **fails ≥ 1 one satisfiability test**. By PHP, some test that fails with probability $1/$#tests over the distribution

Real Proof Sketch

Extreme Bird's Eye View

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

Real Proof Sketch

Extreme Bird's Eye View

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

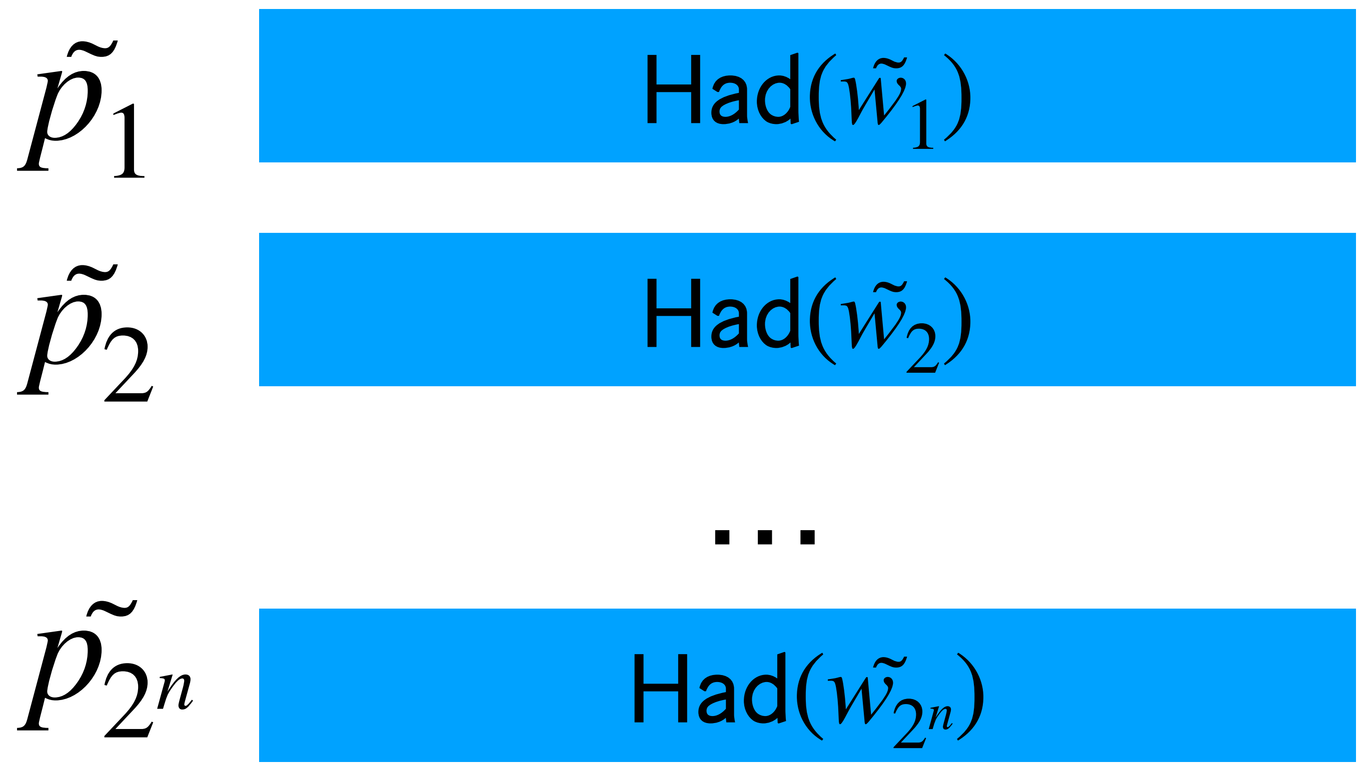
- **Step 1:** NS strategy satisfying linear and quadratic consistency can be written as "pseudo-probabilities" over "correct" encodings, i.e. $\{\text{Had}(\tilde{w})\}_{w \in \{0,1\}^n}$.

Real Proof Sketch

Extreme Bird's Eye View

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

- **Step 1:** NS strategy satisfying linear and quadratic consistency can be written as **"pseudo-probabilities"** over **"correct" encodings**, i.e. $\{\text{Had}(\tilde{w})\}_{w\in\{0,1\}^n}$.



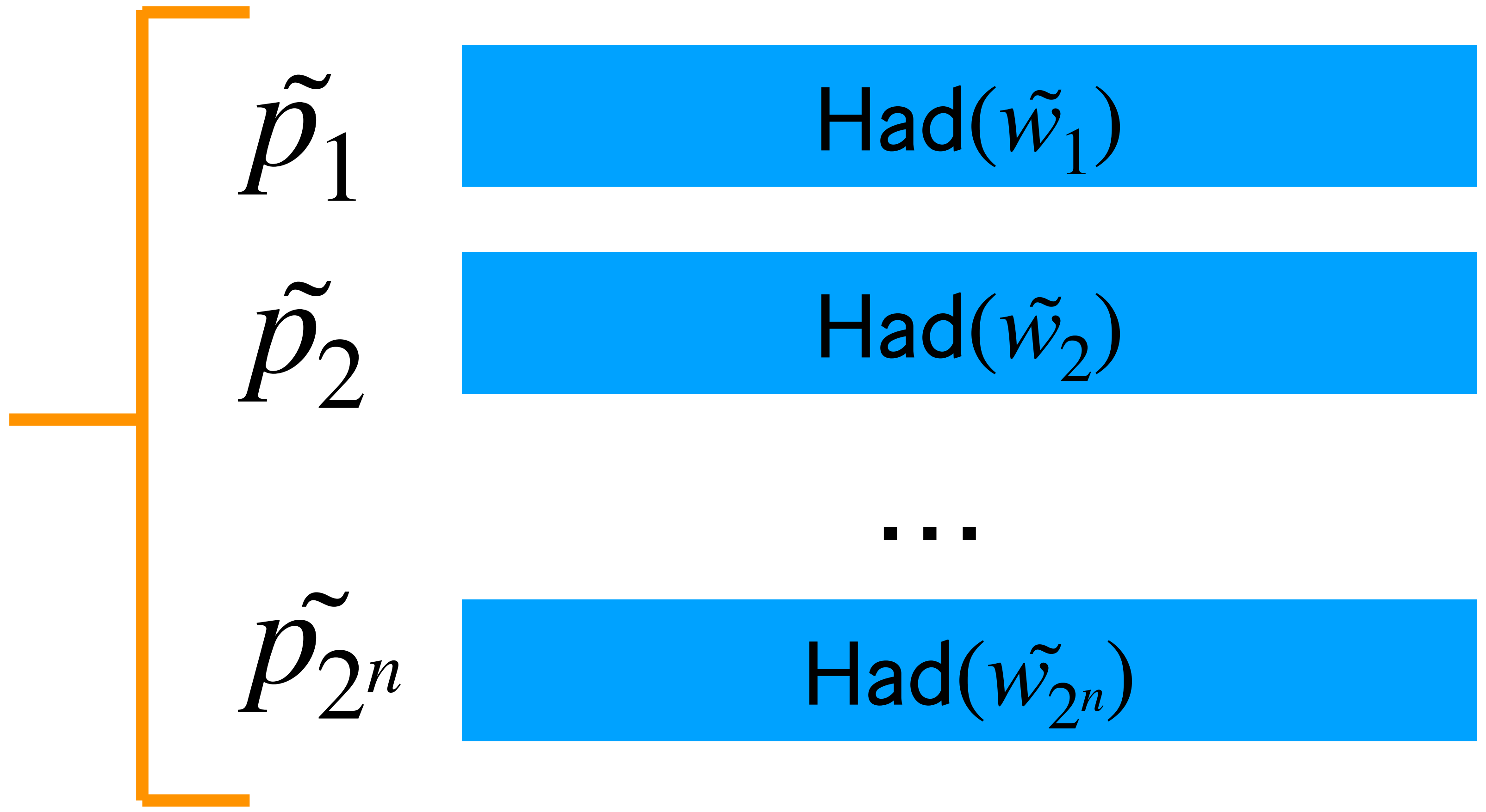
Real Proof Sketch

Extreme Bird's Eye View

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

- **Step 1:** NS strategy satisfying linear and quadratic consistency can be written as **"pseudo-probabilities"** over **"correct" encodings**, i.e. $\{\text{Had}(\tilde{w})\}_{w \in \{0,1\}^n}$.

$\tilde{p}_i$'s can be negative! Require $\sum_i \tilde{p}_i = 1.$



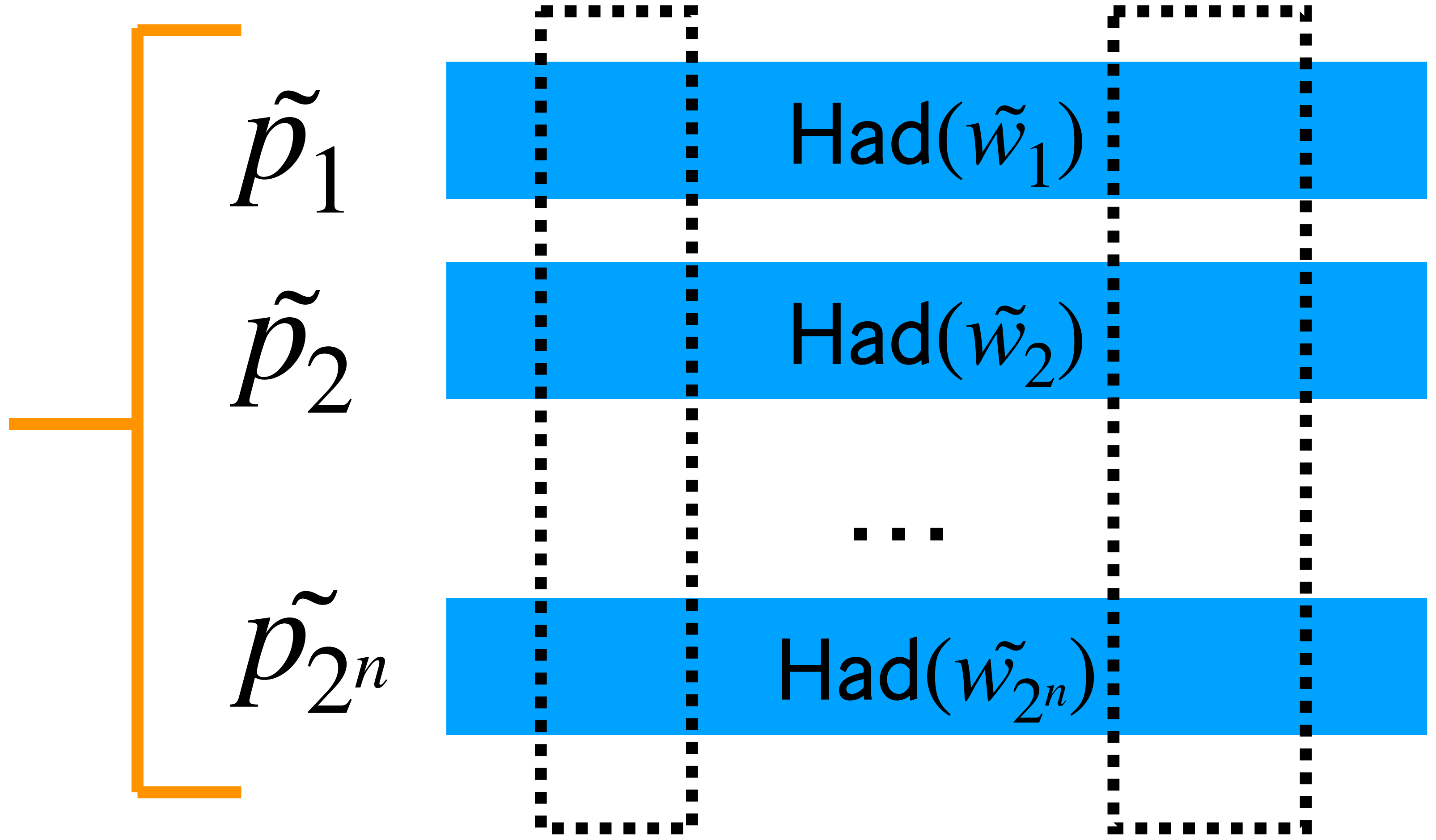
Real Proof Sketch

Extreme Bird's Eye View

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

- **Step 1:** NS strategy satisfying linear and quadratic consistency can be written as **"pseudo-probabilities"** over **"correct" encodings**, i.e. $\{\text{Had}(\tilde{w})\}_{w \in \{0,1\}^n}$.

$\tilde{p}_i$'s can be negative! Require $\sum_i \tilde{p}_i = 1$.



D_Q corresponds to the marginals.

Real Pro

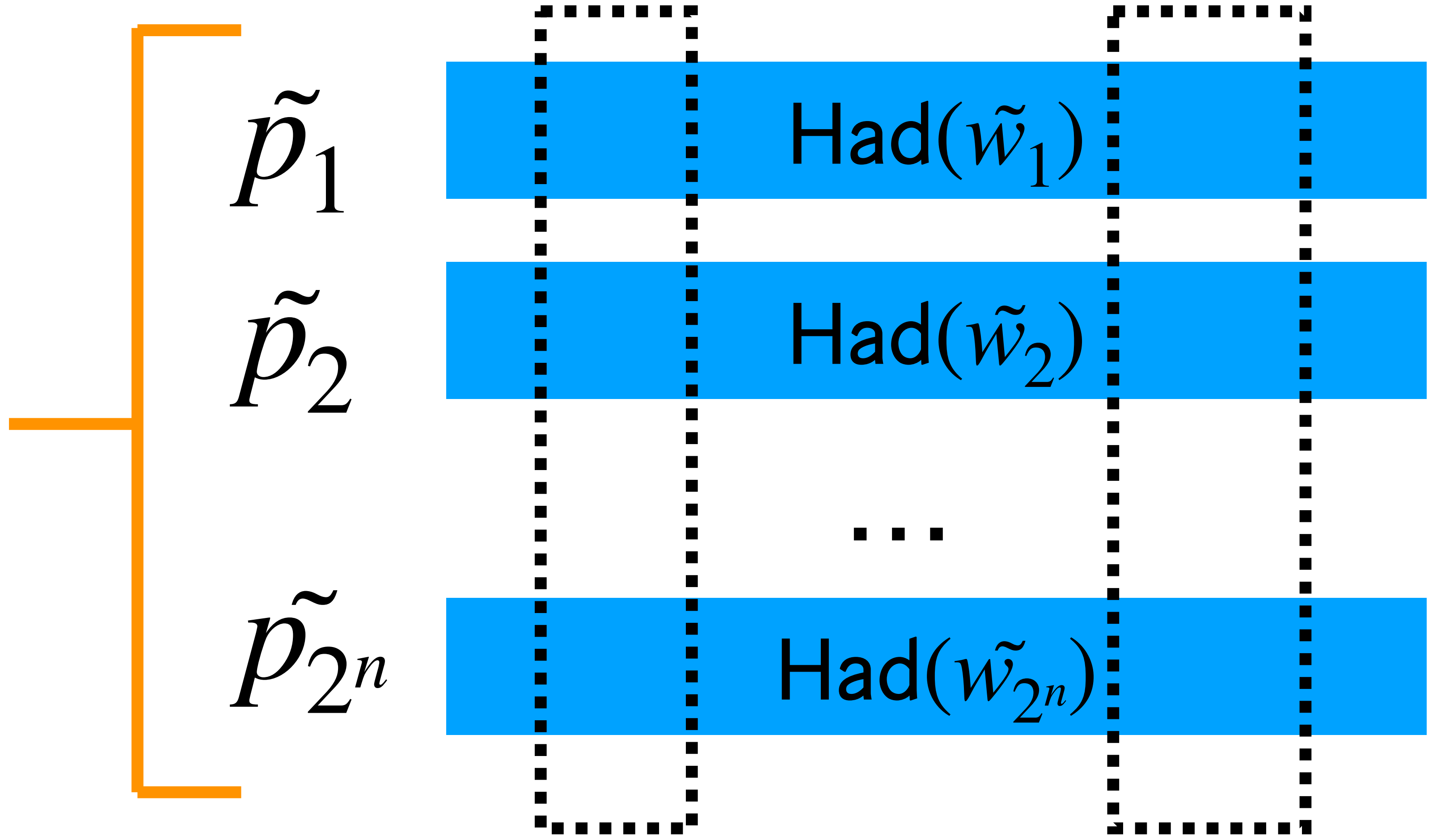
Extreme B

Uses Hilbert's Nullstellensatz
and Sherali-Adams pseudoexpectations.
Uses ideas from [CMS '18]

- Verifier on $q_1, q_2, \ldots, q_k$ checks:
  - **Linear consistency:** If $q_i \oplus q_j \oplus q_k = 0$, then $a_i \oplus a_j \oplus a_k = 0$.
  - **Quadratic consistency:** Check $w_{ij} = w_i \cdot w_j$.
  - **Satisfiability:** Check that 3-local linear equations are satisfied.

- **Step 1:** NS strategy satisfying linear and quadratic consistency can be written as **"pseudo-probabilities"** over **"correct" encodings**, i.e. $\{\text{Had}(\tilde{w})\}_{w \in \{0,1\}^n}$.

$\tilde{p}_i$'s can be negative! Require $\sum_i \tilde{p}_i = 1$.



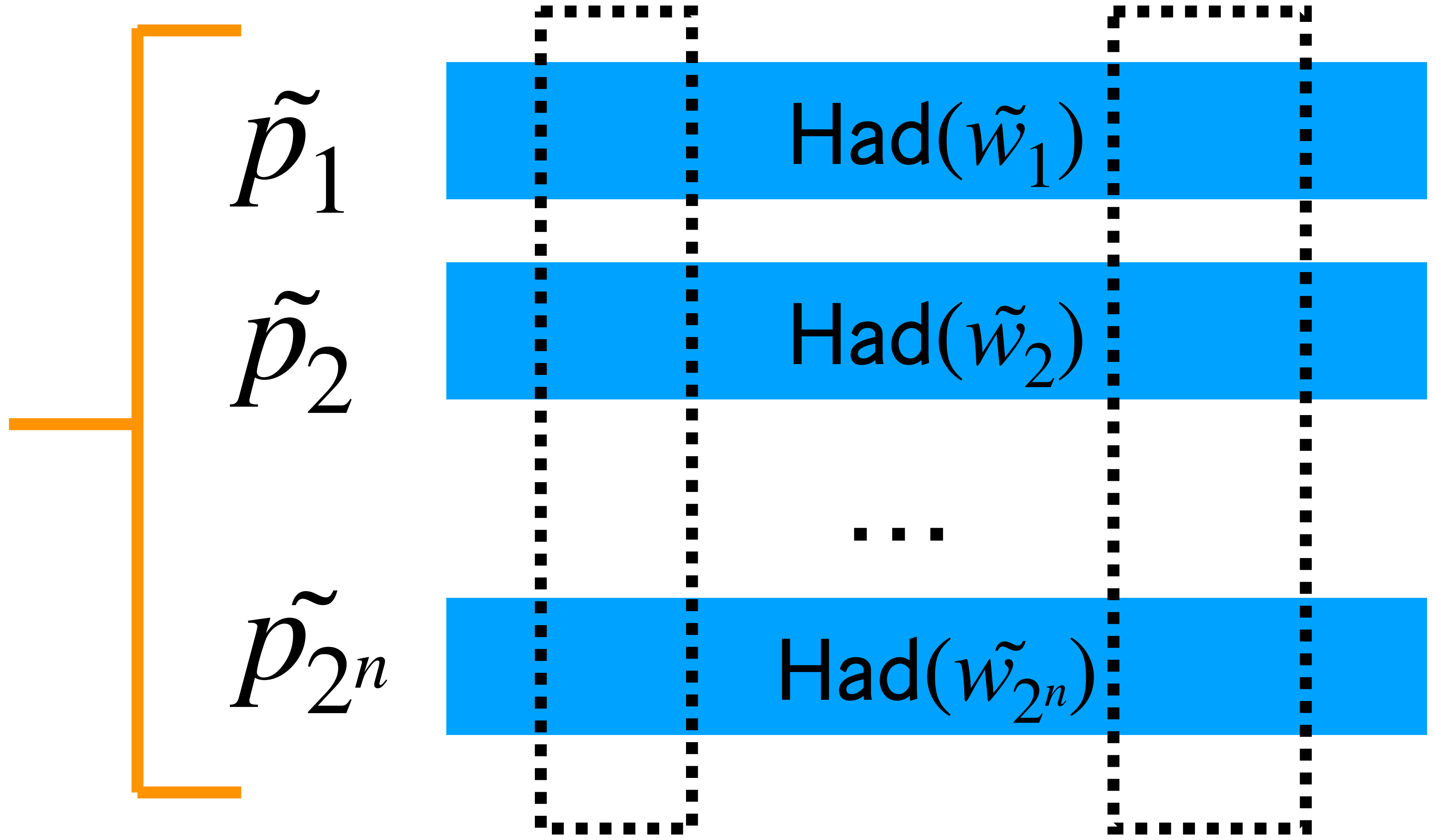
D_Q corresponds to the marginals.

Real Proof Sketch

Extreme Bird's Eye View

- **Step 1:** NS strategy satisfying linear and quadratic consistency can be written as **"pseudo-probabilities"** over **"correct" encodings**, i.e. $\{\text{Had}(\tilde{w})\}_{w \in \{0,1\}^n}$.

$\tilde{p}_i$'s can be negative! Require $\sum_i \tilde{p}_i = 1$.



D_Q corresponds to the marginals.

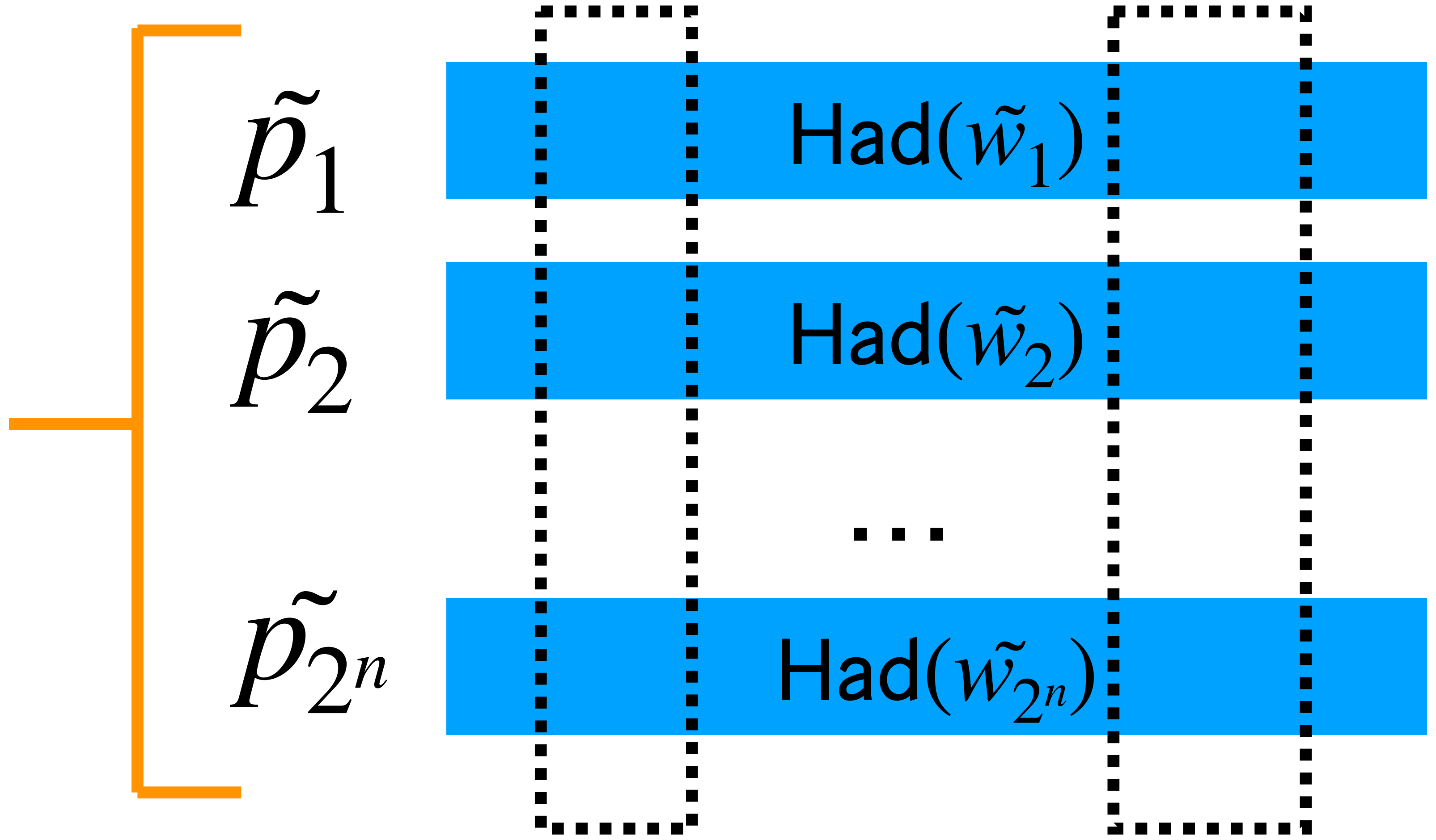
Real Proof Sketch

Extreme Bird's Eye View

- **Step 1:** NS strategy satisfying linear and quadratic consistency can be written as **"pseudo-probabilities"** over **"correct" encodings**, i.e. $\{\text{Had}(\tilde{w})\}_{w\in\{0,1\}^n}$.

Let $S_{Q,A}$ be the set of $i \in \{1,\ldots,2^n\}$ such that $\text{Had}(\tilde{w}_i)$ on Q is A.

$\tilde{p}_i$'s can be negative! Require $\sum_i \tilde{p}_i = 1$.



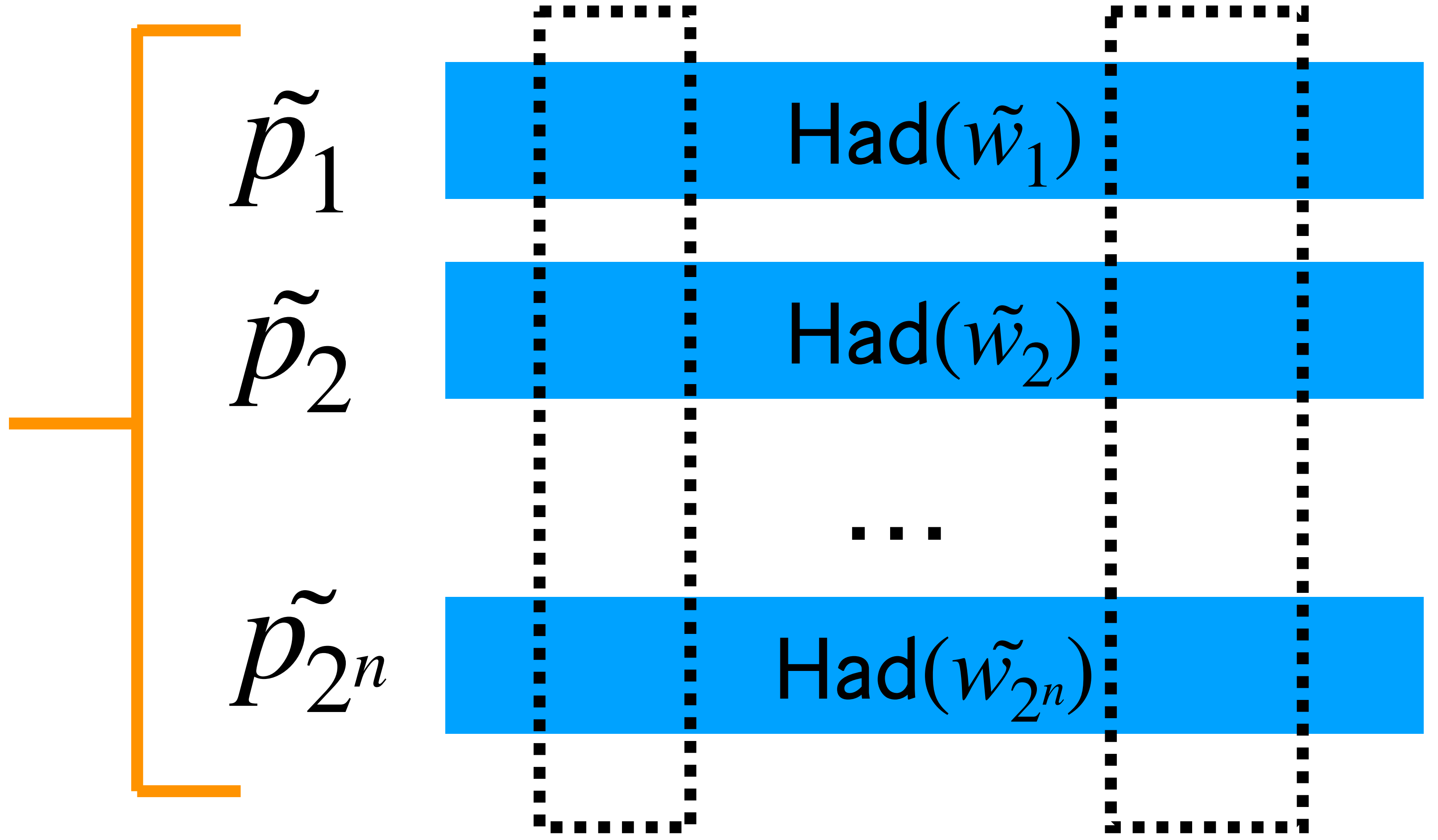
D_Q corresponds to the marginals.

Real Proof Sketch

Extreme Bird's Eye View

- **Step 1:** NS strategy satisfying linear and quadratic consistency can be written as **"pseudo-probabilities"** over **"correct" encodings**, i.e. $\{\text{Had}(\tilde{w})\}_{w \in \{0,1\}^n}$.

$\tilde{p}_i$'s can be negative! Require $\sum_i \tilde{p}_i = 1$.



D_Q corresponds to the marginals.

Let $S_{Q,A}$ be the set of $i \in \{1,\ldots,2^n\}$ such that Had($\tilde{w}_i$) on Q is A.

Then,

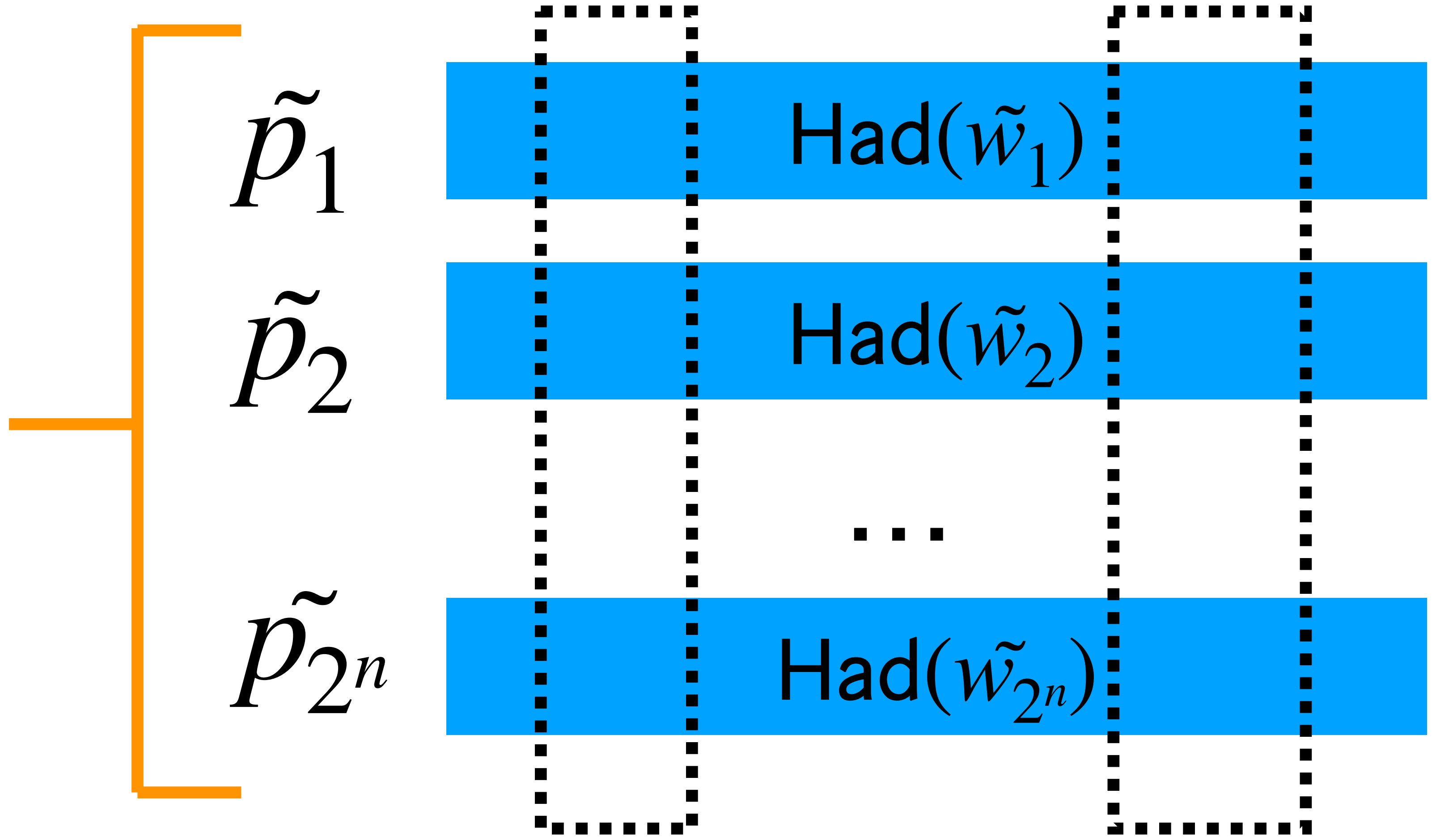
$$\Pr[A \leftarrow D_Q] = \sum_{i \in S_{Q,A_Q}} \tilde{p}_i.$$

Real Proof Sketch

Extreme Bird's Eye View

- **Step 1:** NS strategy satisfying linear and quadratic consistency can be written as **"pseudo-probabilities"** over **"correct" encodings**, i.e. $\{\text{Had}(\tilde{w})\}_{w \in \{0,1\}^n}$.

$\tilde{p}_i$'s can be negative! Require $\sum_i \tilde{p}_i = 1$.



D_Q corresponds to the marginals.

Let $S_{Q,A}$ be the set of $i \in \{1,\ldots,2^n\}$ such that Had($\tilde{w}_i$) on Q is A.

Then,

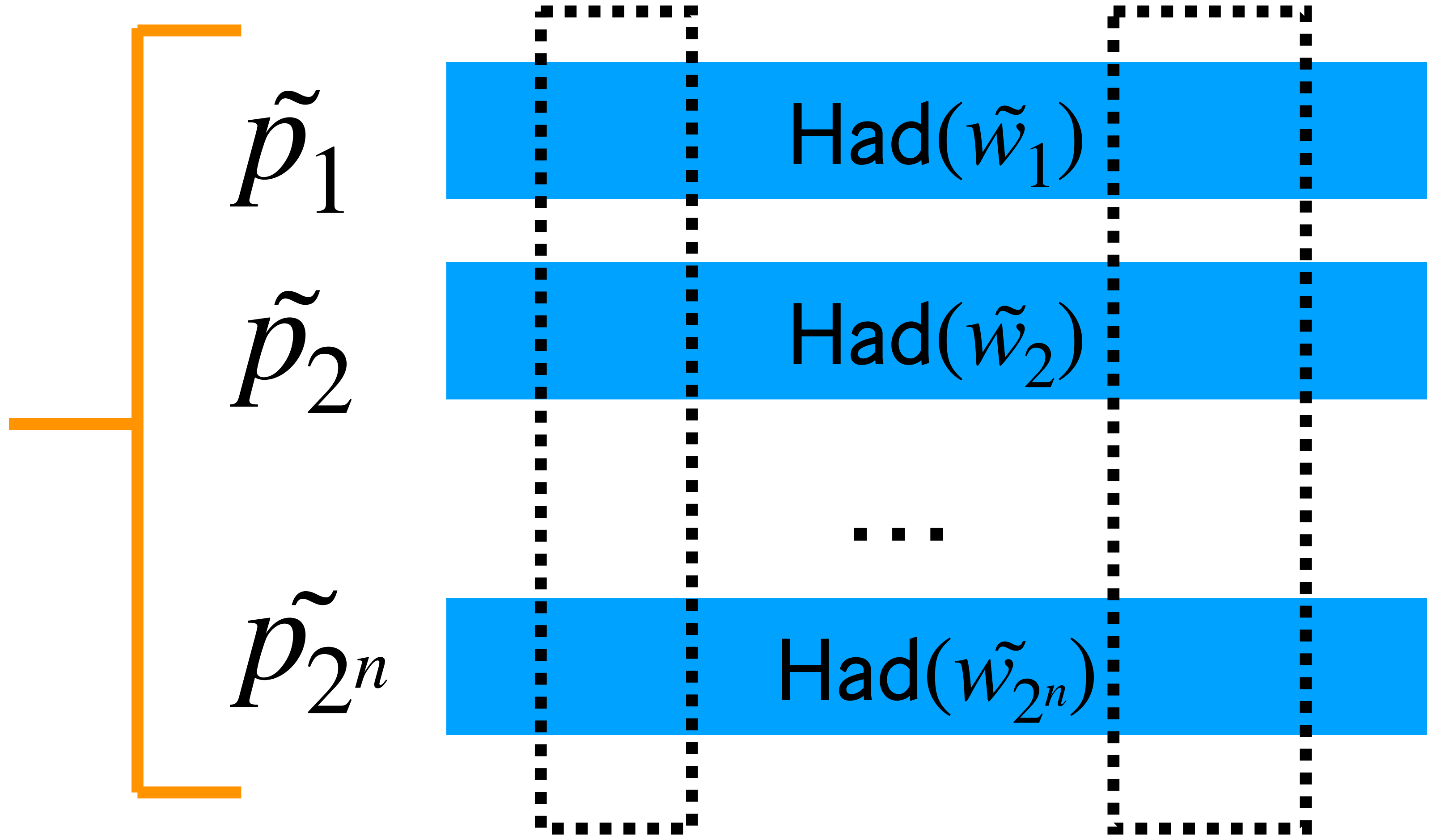
$$\Pr[A \leftarrow D_Q] = \sum_{i \in S_{Q,A_Q}} \tilde{p}_i.$$

Local views look "real": These probabilities will be in [0, 1]

Real Proof Sketch

Extreme Bird's Eye View

$\tilde{p}_i$'s can be negative! Require $\sum_i \tilde{p}_i = 1.$



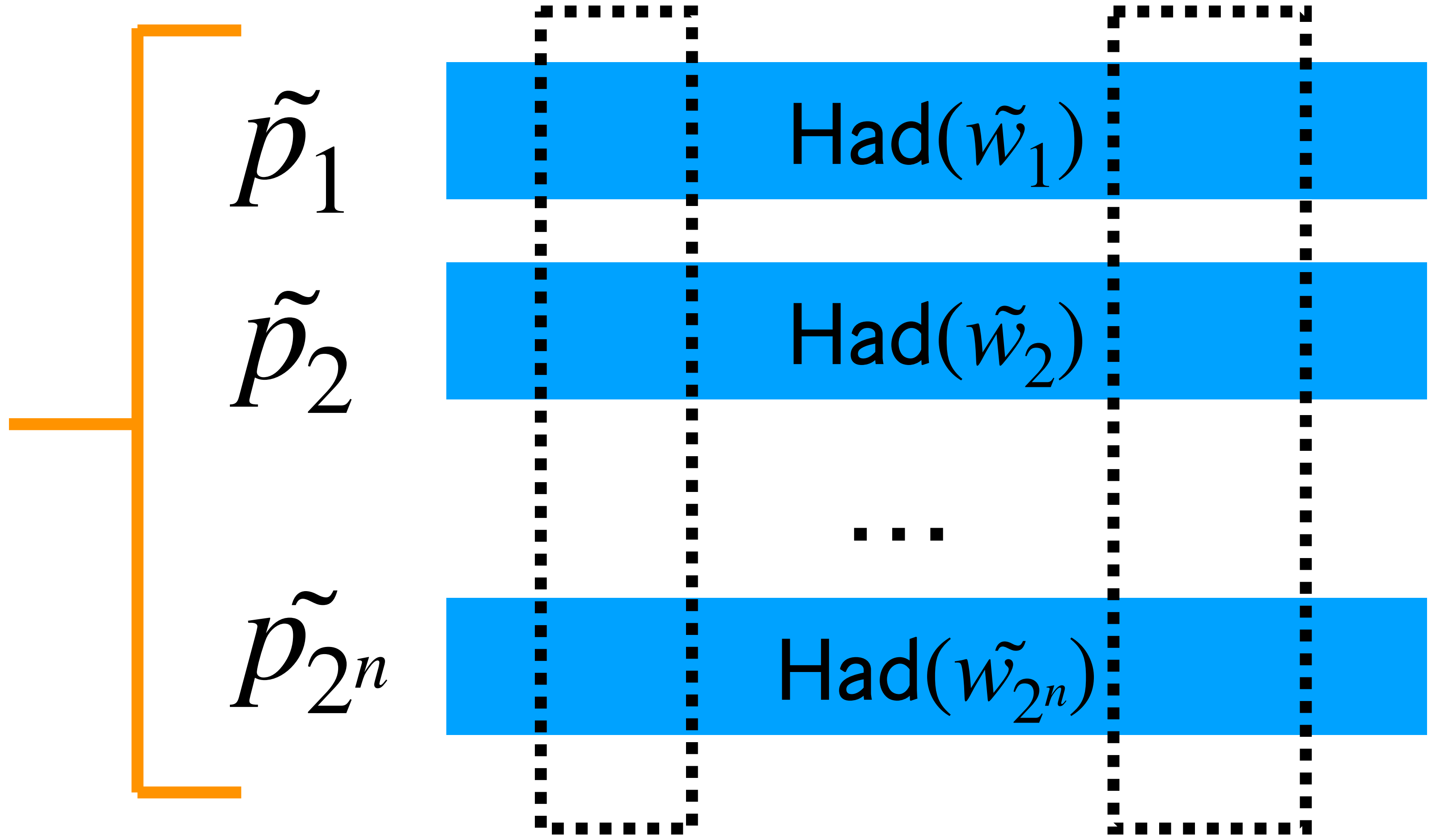
D_Q corresponds to the marginals.

Real Proof Sketch

Extreme Bird's Eye View

- **Step 2:** If $x \notin \mathscr{L}$, then every encoding fails *some* **satisfiability test.**

$\tilde{p}_i$'s can be negative! Require $\sum_i \tilde{p}_i = 1.$



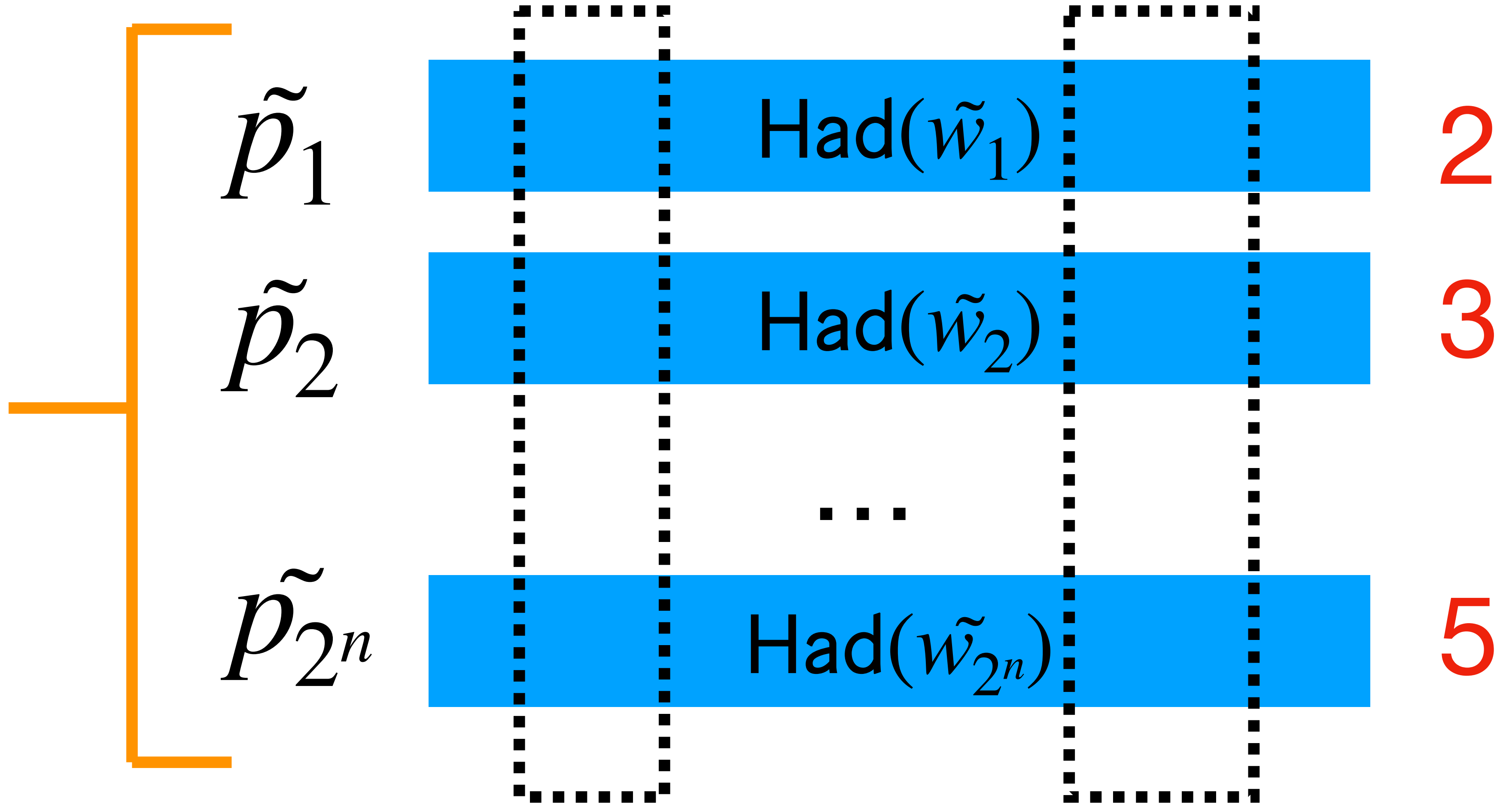
D_Q corresponds to the marginals.

Real Proof Sketch

Extreme Bird's Eye View

- **Step 2:** If $x \notin \mathscr{L}$, then every encoding fails *some* **satisfiability test.**

$\tilde{p}_i$'s can be negative! Require $\sum_i \tilde{p}_i = 1$.



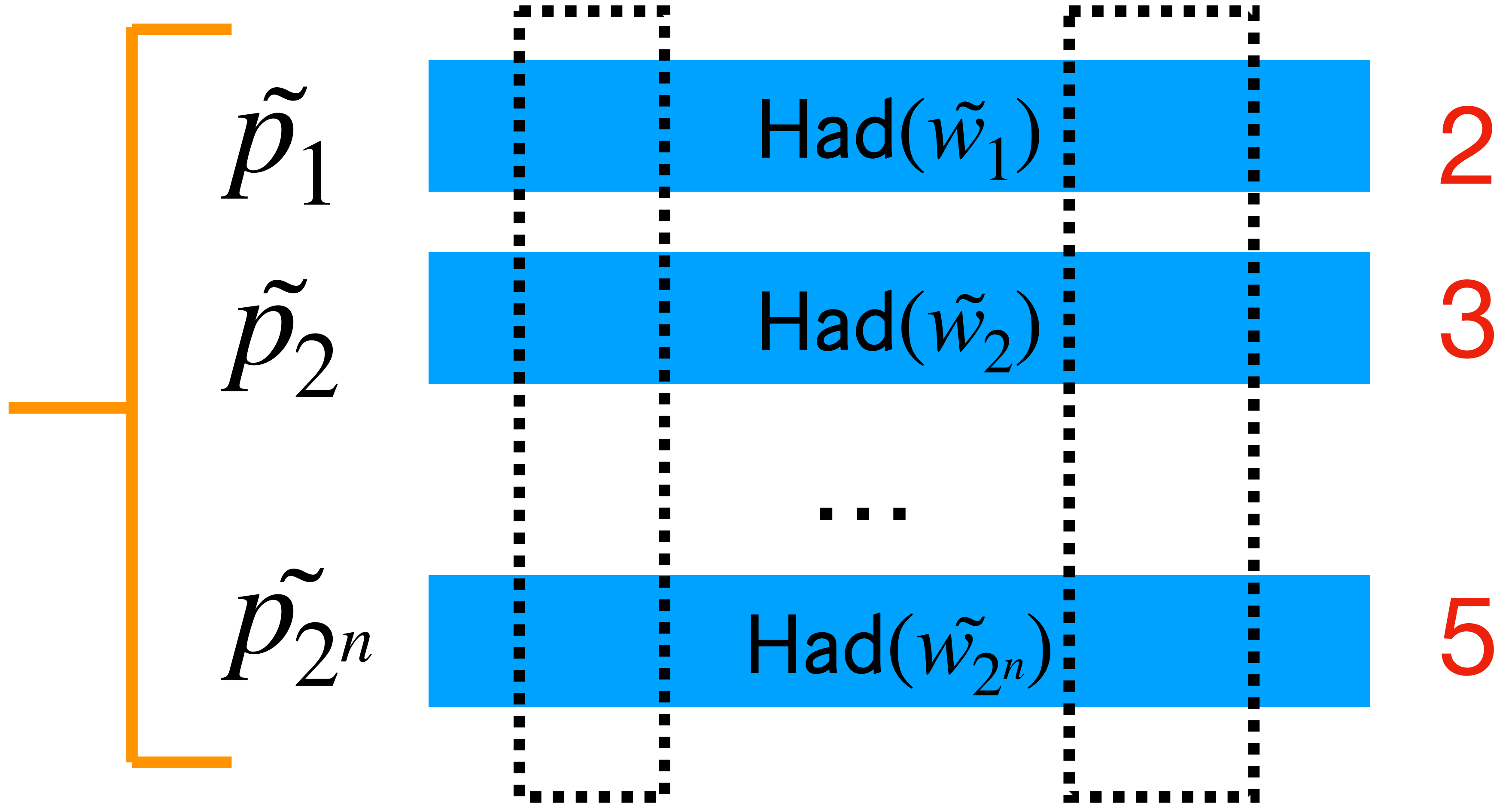
D_Q corresponds to the marginals.

Real Proof Sketch

Extreme Bird's Eye View

- **Step 2:** If $x \notin \mathscr{L}$, then every encoding fails *some* **satisfiability test.**
  - **Issue:** Might not be observable, because pseudo-probabilities can be **negative.**

$\tilde{p}_i$'s can be negative! Require $\sum_i \tilde{p}_i = 1$.



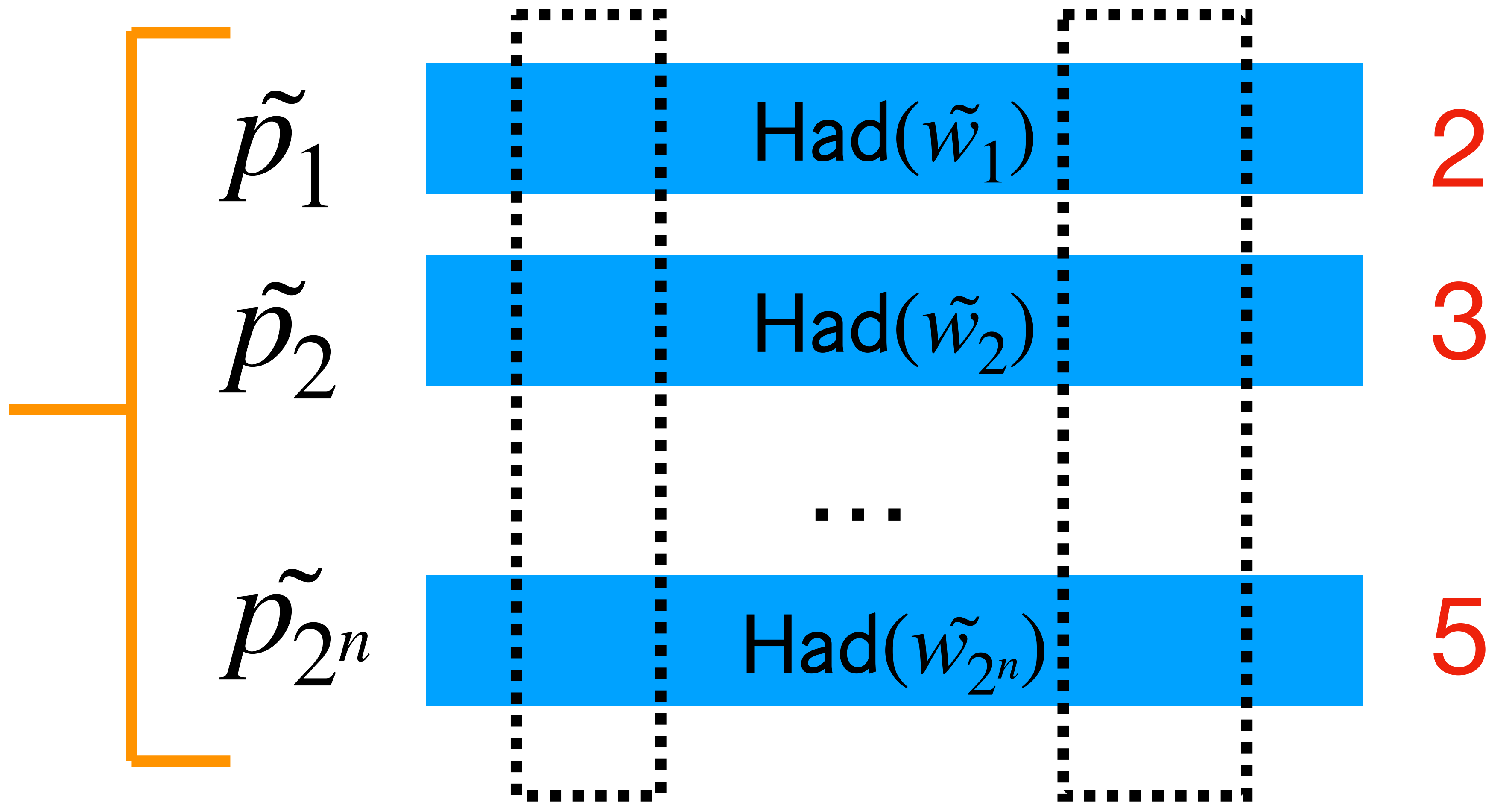
D_Q corresponds to the marginals.

Real Proof Sketch

Extreme Bird's Eye View

- **Step 2:** If $x \notin \mathscr{L}$, then every encoding fails *some* **satisfiability test.**
  - **Issue:** Might not be observable, because pseudo-probabilities can be **negative.**

$\tilde{p}_i$'s can be negative! Require $\sum_i \tilde{p}_i = 1.$



D_Q corresponds to the marginals.

Fix (high-level):
Use Hadamard encoding to read "**random** linear combinations of the **satisfiability tests**".

Use careful counting.

Summary

Summary

- **In this work:** We show a way around the “non-signaling barrier” and establish a new pathway towards constructing SNARGs for NP.

Summary

- **In this work:** We show a way around the "non-signaling barrier" and establish a new pathway towards constructing SNARGs for NP.

- We relax the requirement in two ways:

Summary

- **In this work:** We show a way around the "non-signaling barrier" and establish a new pathway towards constructing SNARGs for NP.
- We relax the requirement in two ways:
  - We allow for **exponential-length** PCPs.

Summary

- **In this work:** We show a way around the "non-signaling barrier" and establish a new pathway towards constructing SNARGs for NP.

- We relax the requirement in two ways:

  - We allow for **exponential-length** PCPs.

  - We relax to soundness requirement to **"weak soundness"**.

Summary

- **In this work:** We show a way around the “non-signaling barrier” and establish a new pathway towards constructing SNARGs for NP.
- We relax the requirement in two ways:
  - We allow for **exponential-length** PCPs.
  - We relax to soundness requirement to **“weak soundness”**.
- **Candidate construction** under a conjecture.

Summary

- **In this work:** We show a way around the “non-signaling barrier” and establish a new pathway towards constructing SNARGs for NP.
- We relax the requirement in two ways:
  - We allow for **exponential-length** PCPs.
  - We relax to soundness requirement to **“weak soundness”**.
- **Candidate construction** under a conjecture.
- **Open question:** Does there exist a NS PCP with weak soundness?

Thank you for your attention!

ePrint 2026/006

Thank you for your attention!

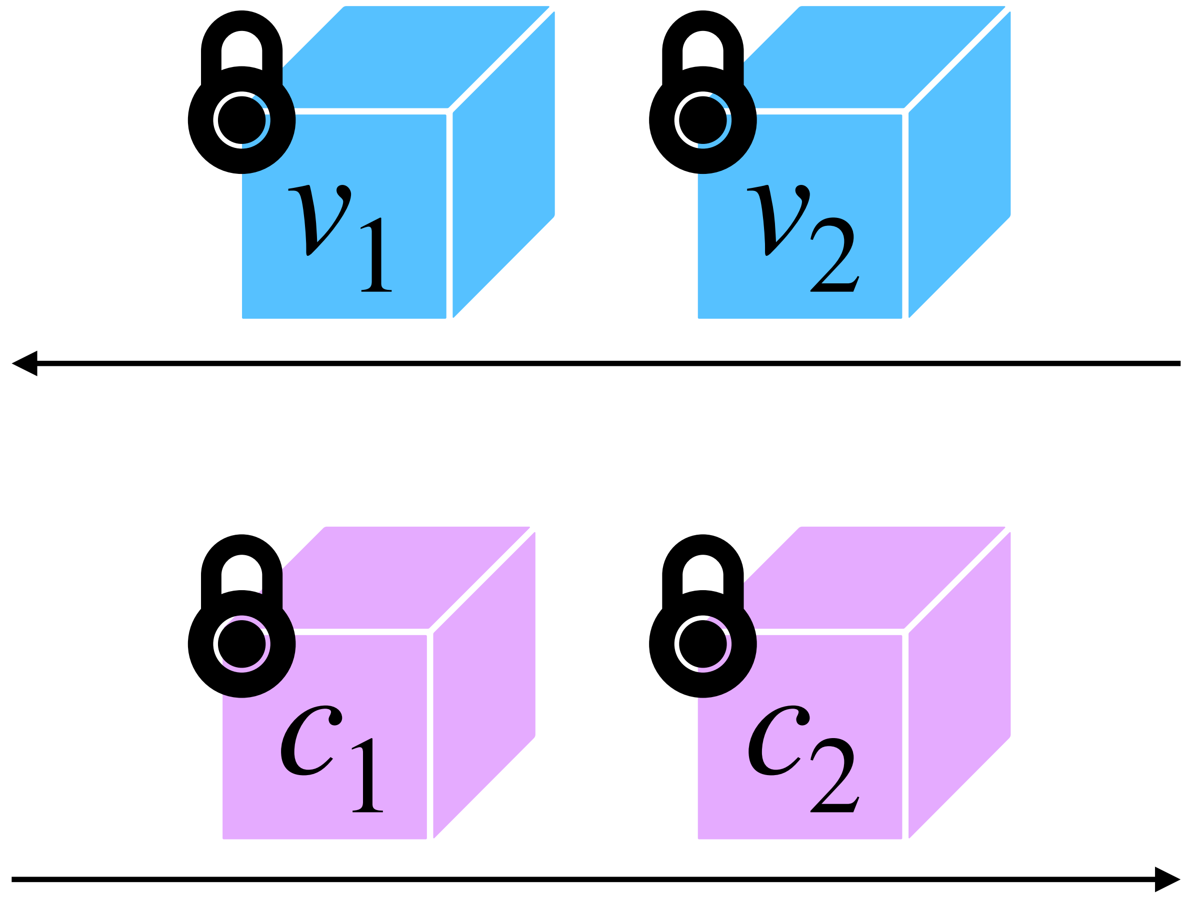
ePrint 2026/006

Images used are from flaticon.com, tikzpeople,

Oversimplified counterexample

[Dwork-Landberg-Naor-Nissim-Reingold '04]

- **Language:** Graph 3-Colouring
  - **PCP string:** 3-colouring of the graph
  - **Verifier:** Check a random edge (v1, v2). Catches with probability $1/|E|$.



Issue: Prover is not "committed" to any single PCP string!