How to Prove Post-Quantum Security for Succinct Non-Interactive Reductions

Alessandro Chiesa, Zijing Di, Zihan Hu, Yuxi Zheng

EPFL

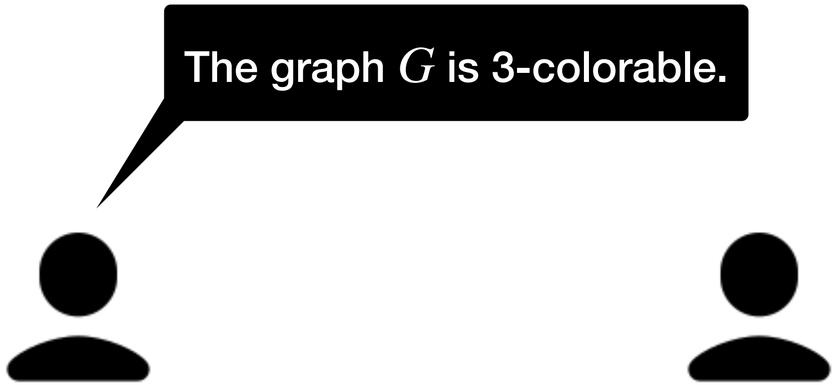
To appear in Eurocrypt 2026

What are succinct non-interactive reductions?

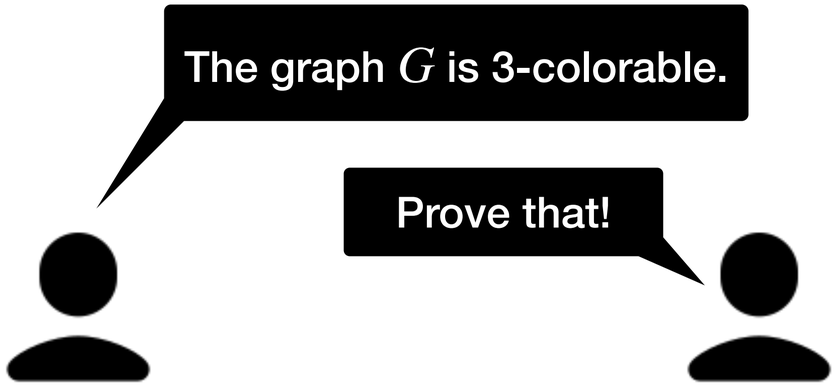
Succinct non-interactive arguments (SNARGs)

Succinct non-interactive arguments (SNARGs)

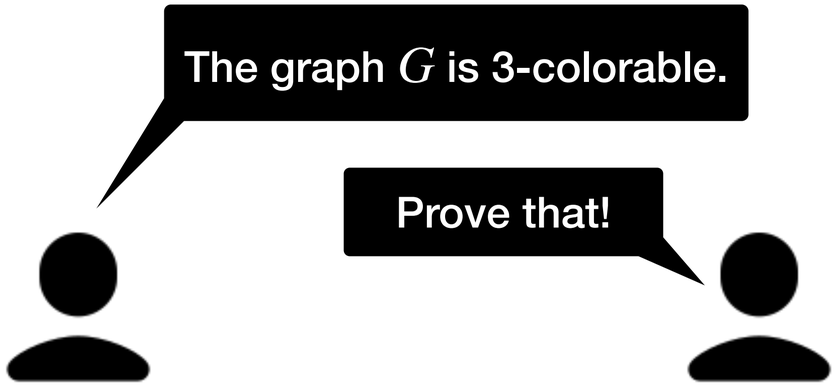
Succinct non-interactive arguments (SNARGs)



Succinct non-interactive arguments (SNARGs)

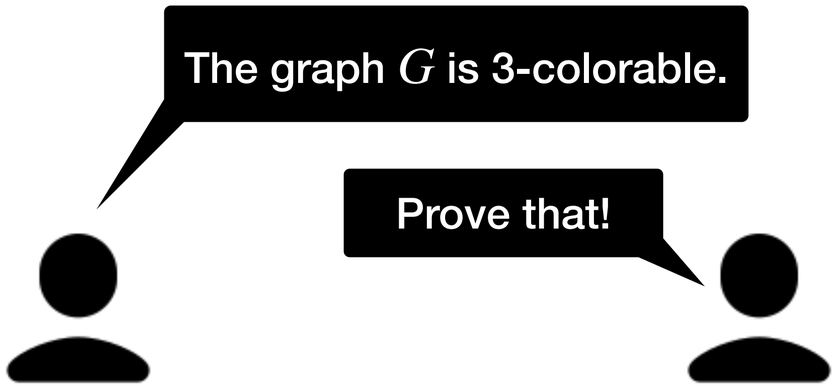


Succinct non-interactive arguments (SNARGs)



But a coloring of G is too long…

Succinct non-interactive arguments (SNARGs)

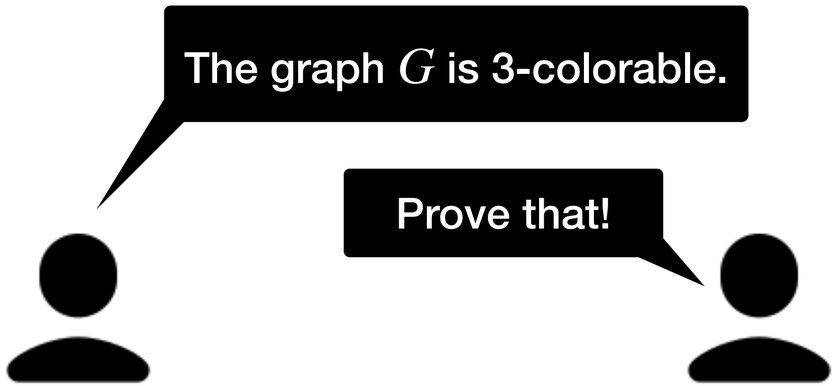


Prover

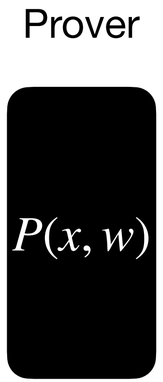
$P(x, w)$

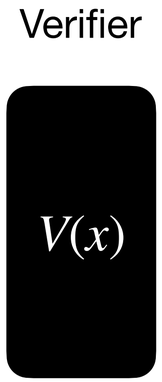
But a coloring of G is too long…

Succinct non-interactive arguments (SNARGs)

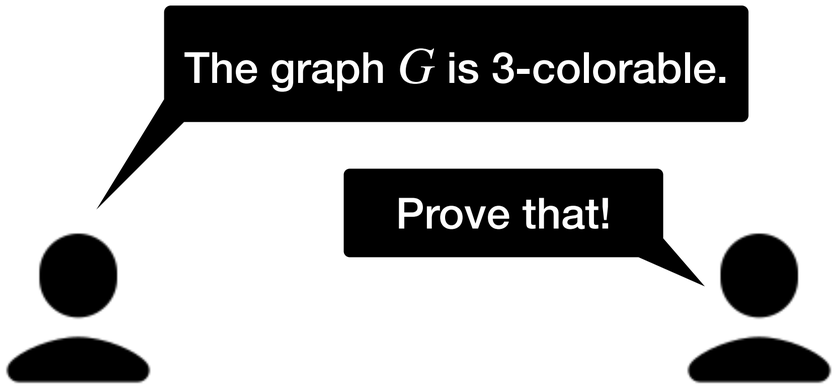


But a coloring of G is too long…

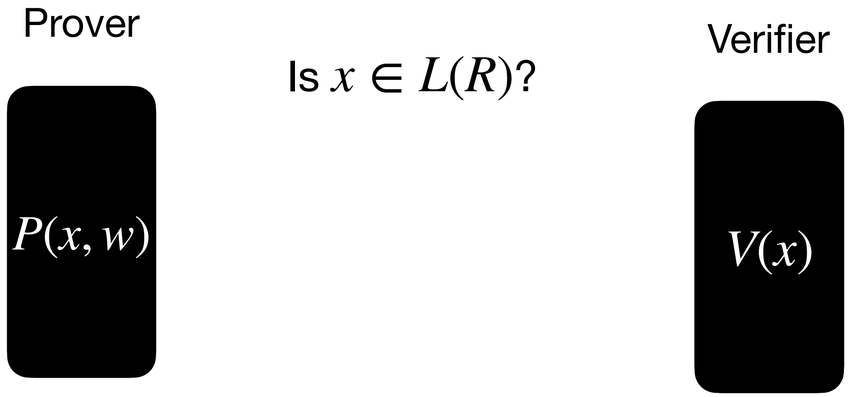




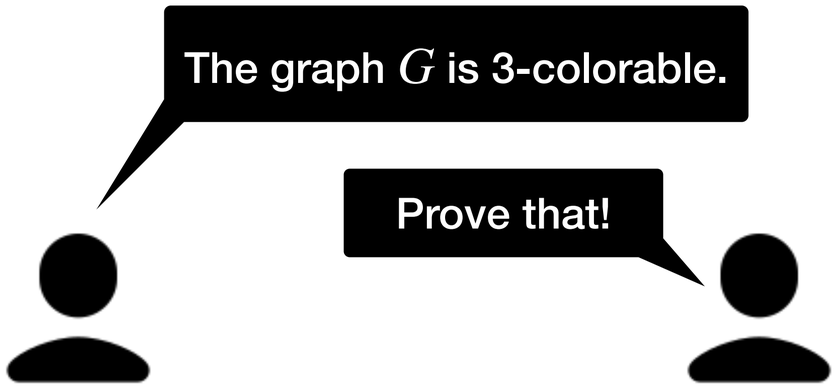
Succinct non-interactive arguments (SNARGs)



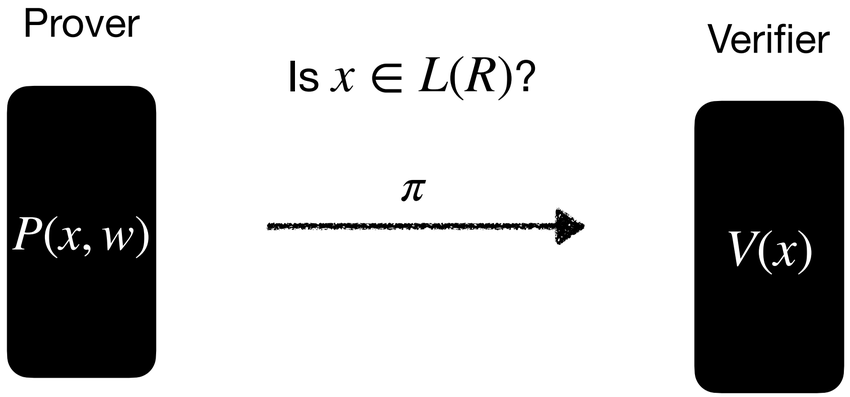
But a coloring of G is too long…



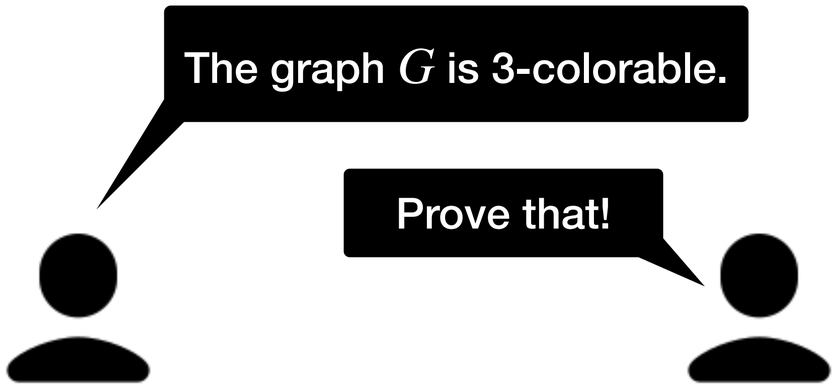
Succinct non-interactive arguments (SNARGs)



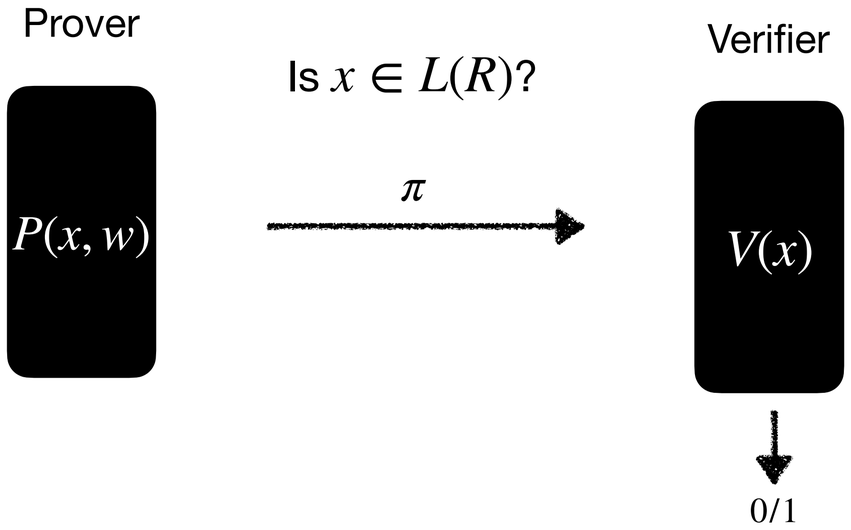
But a coloring of G is too long…



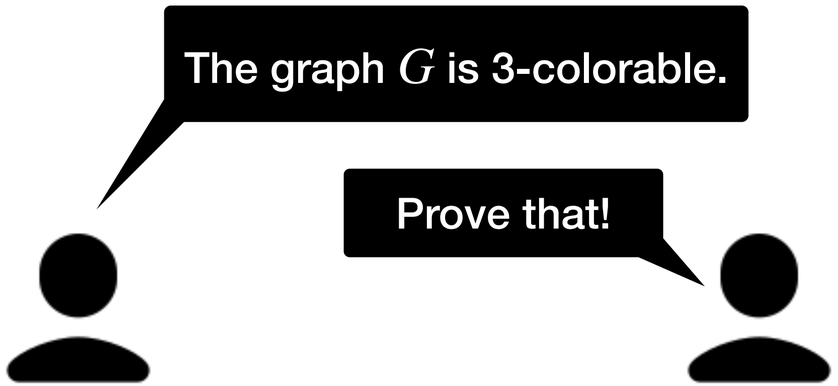
Succinct non-interactive arguments (SNARGs)



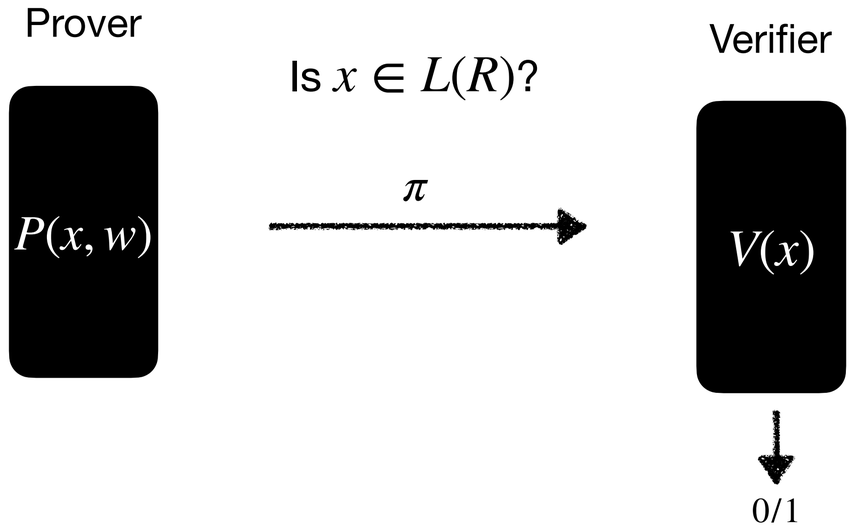
But a coloring of G is too long…



Succinct non-interactive arguments (SNARGs)

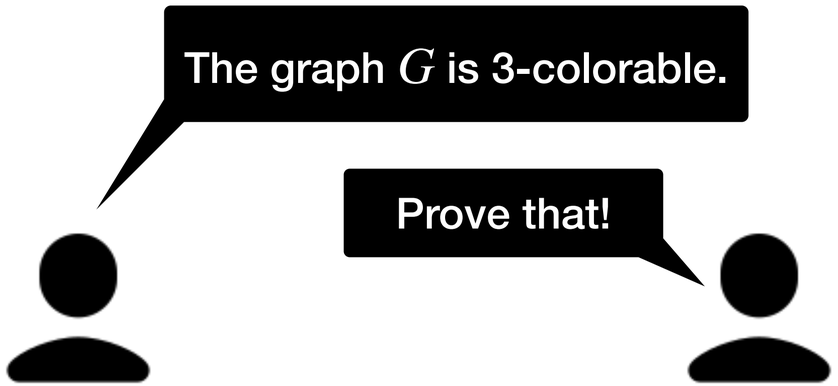


But a coloring of G is too long…

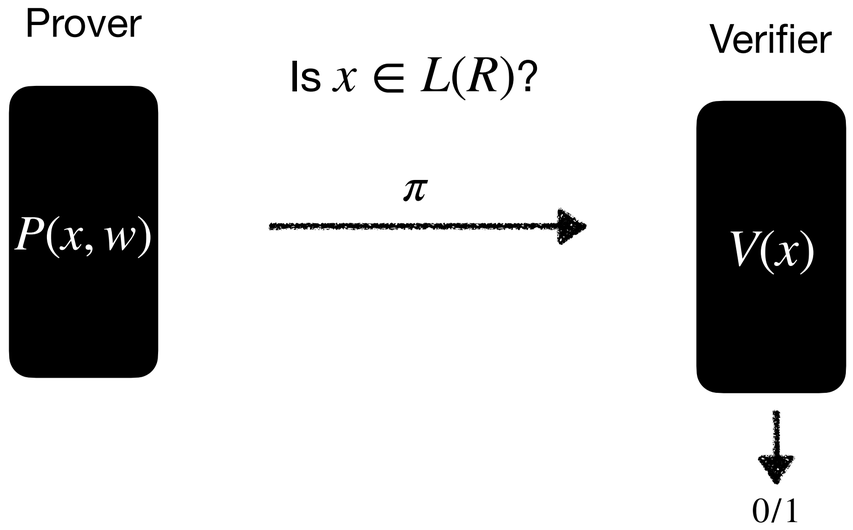


Completeness: $(x, w) \in R \rightarrow P(x, w)$ convinces $V(x)$.

Succinct non-interactive arguments (SNARGs)



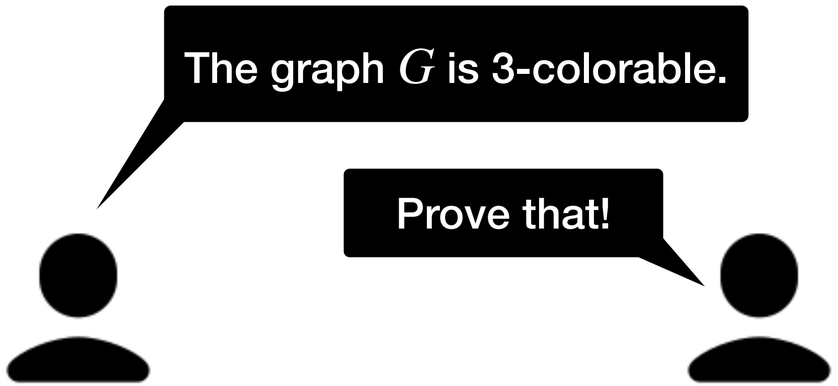
But a coloring of G is too long…



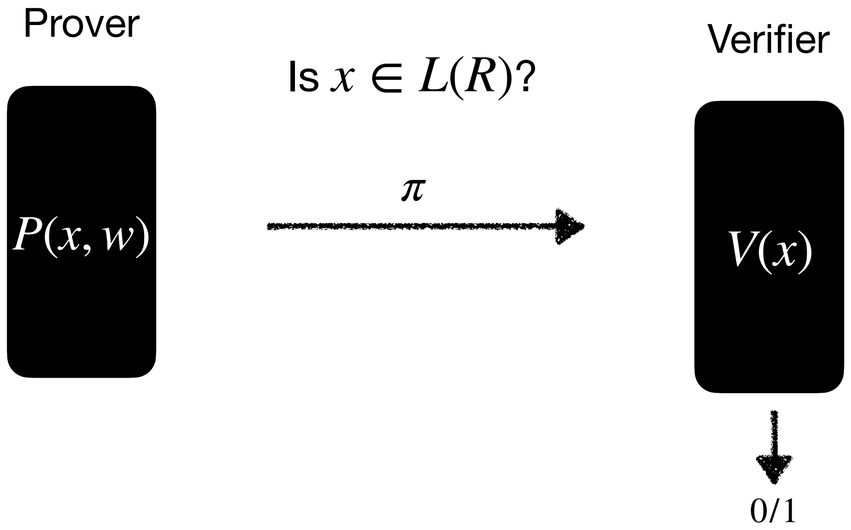
Completeness: $(x, w) \in R \rightarrow P(x, w)$ convinces $V(x)$.

Soundness: $x \notin L(R) \rightarrow$ every **efficient** $\tilde{P}$ convinces $V(x)$ with small probability ϵ.

Succinct non-interactive arguments (SNARGs)



But a coloring of G is too long…

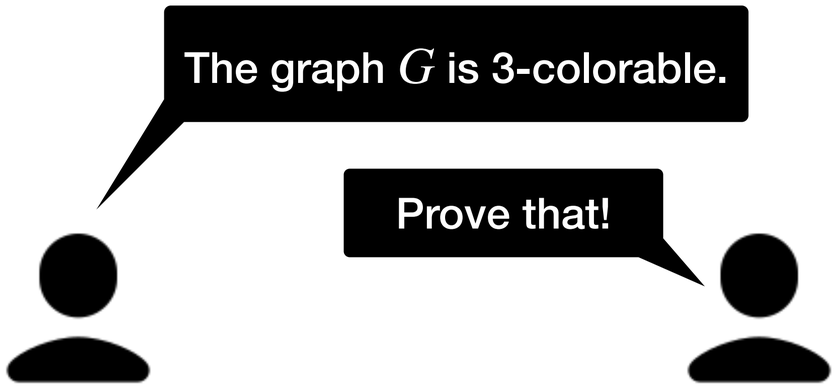


Completeness: $(x, w) \in R \rightarrow P(x, w)$ convinces $V(x)$.

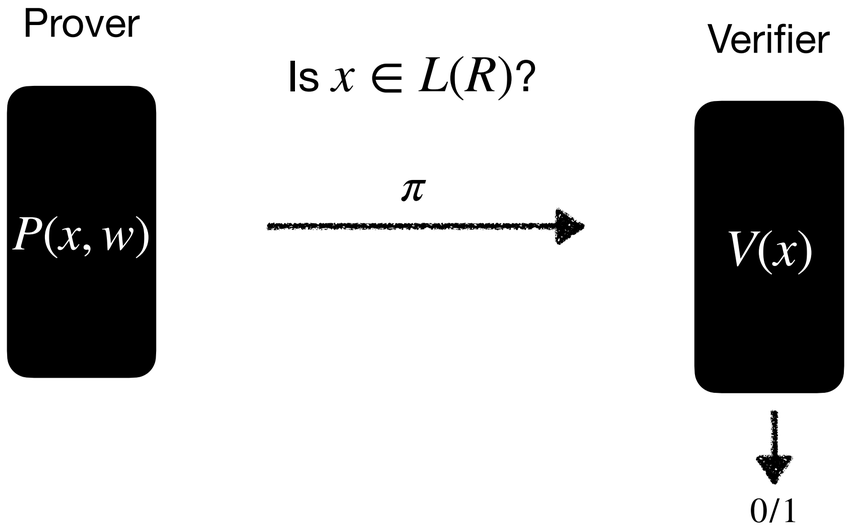
Soundness: $x \notin L(R) \rightarrow$ every **efficient** $\tilde{P}$ convinces $V(x)$ with small probability ϵ.

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

Succinct non-interactive arguments (SNARGs)



But a coloring of G is too long…



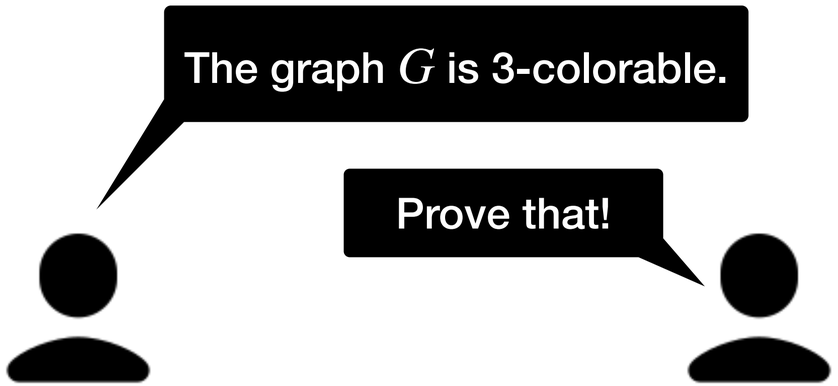
Completeness: $(x, w) \in R \rightarrow P(x, w)$ convinces $V(x)$.

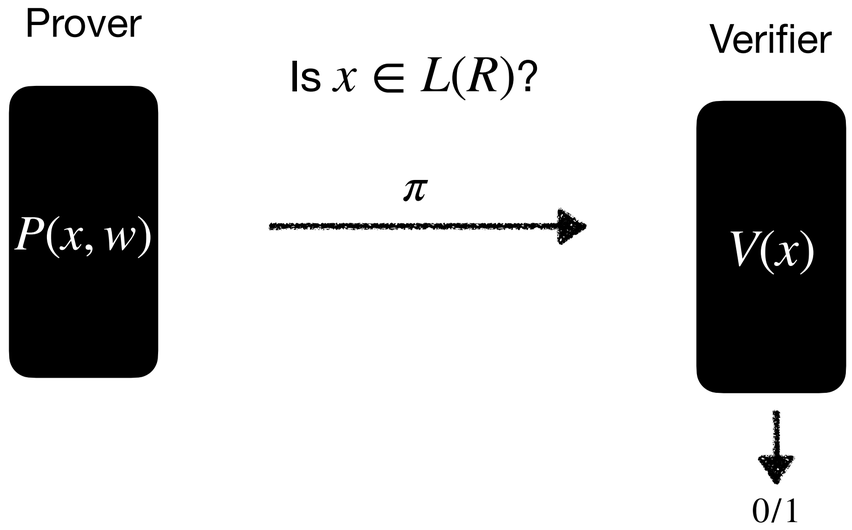
Soundness: $x \notin L(R) \rightarrow$ every **efficient** $\tilde{P}$ convinces $V(x)$ with small probability ϵ.

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

Knowledge soundness: every **efficient** $\tilde{P}$ that convinces $V(x)$ must "know" a witness w s.t. $(x, w) \in R$ (up to a small error κ).

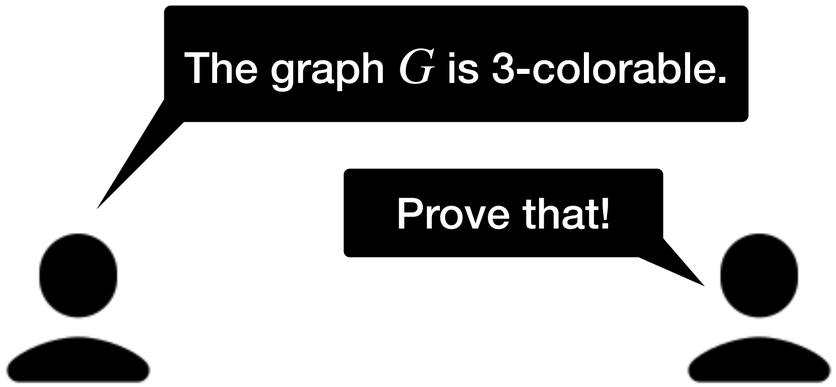
Succinct non-interactive arguments (SNARGs)

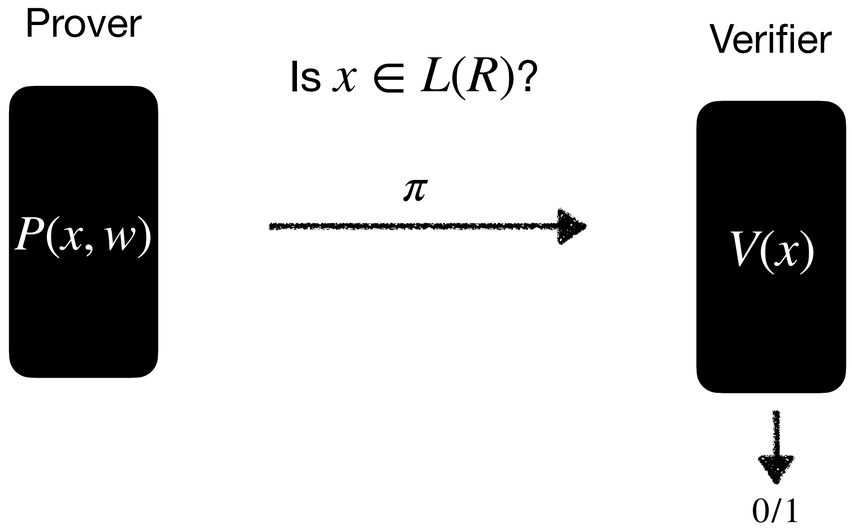




SNARGs have **numerous real-world applications**.

Succinct non-interactive arguments (SNARGs)



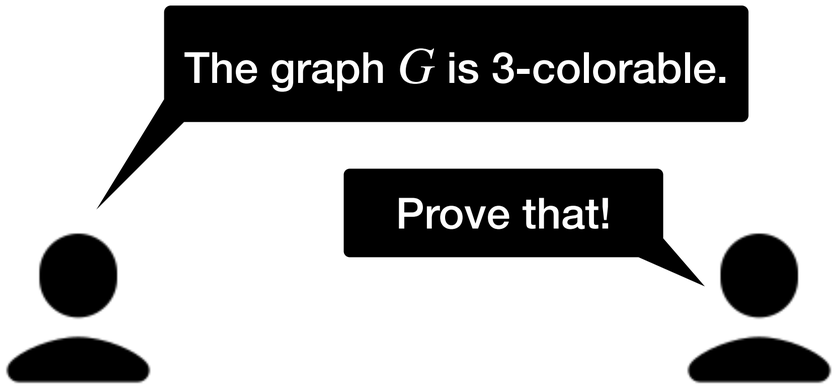


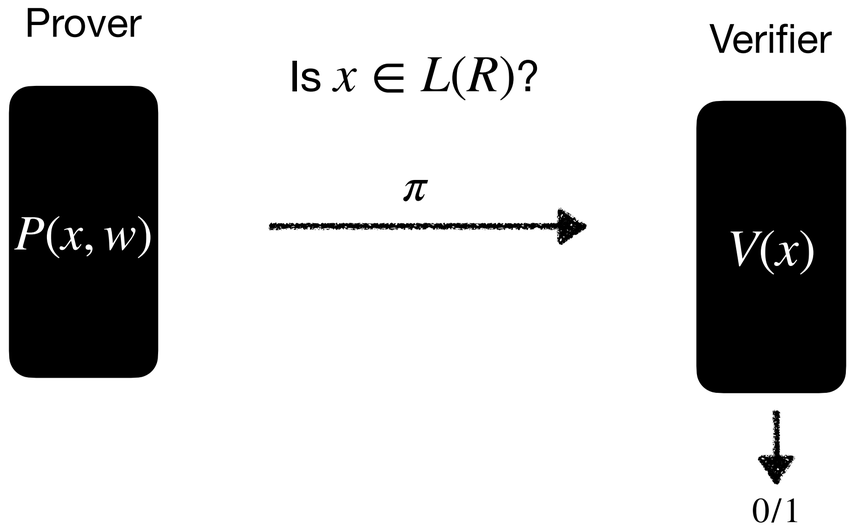
SNARGs have **numerous real-world applications**.

Succinct · RISC ZERO · Aztec · VALIDA · Irreducible

STARKWARE · polygon · NEXUS · Ligero · MatterLabs · ...

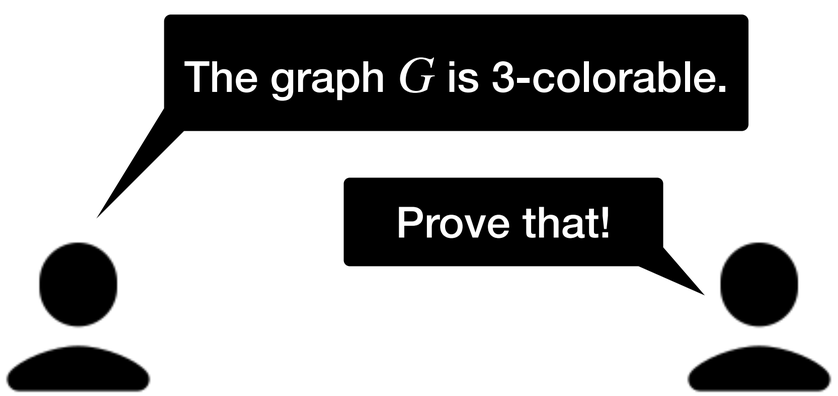
Succinct non-interactive arguments (SNARGs)

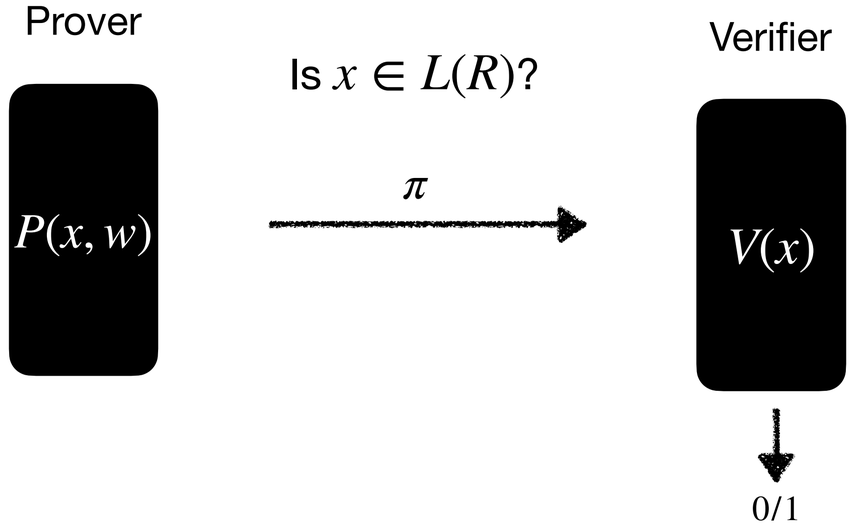




SNARGs have **numerous real-world applications**.

Succinct non-interactive arguments (SNARGs)

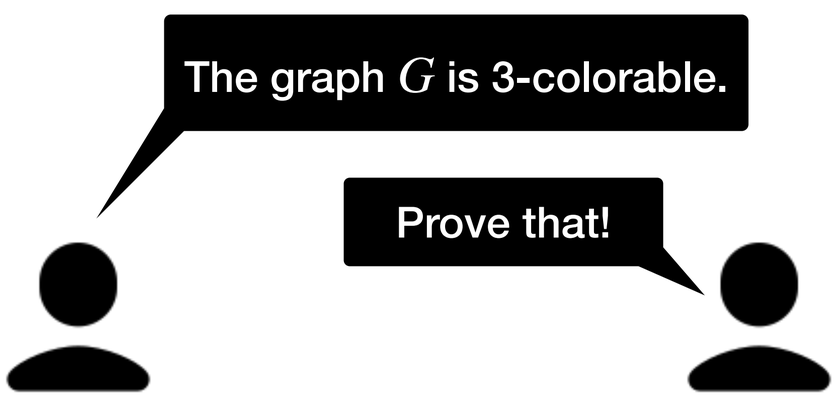


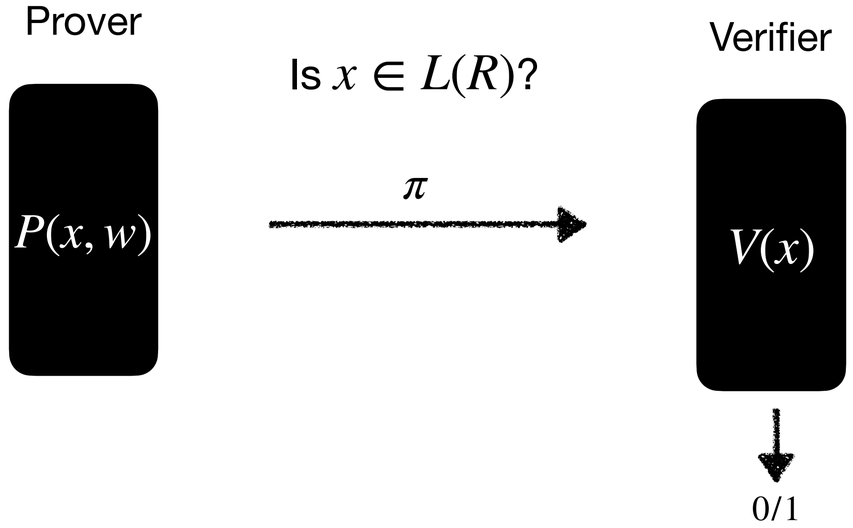


SNARGs have **numerous real-world applications**.

SNARGs are powerful, but sometimes more than needed.

Succinct non-interactive arguments (SNARGs)





SNARGs have **numerous real-world applications**.

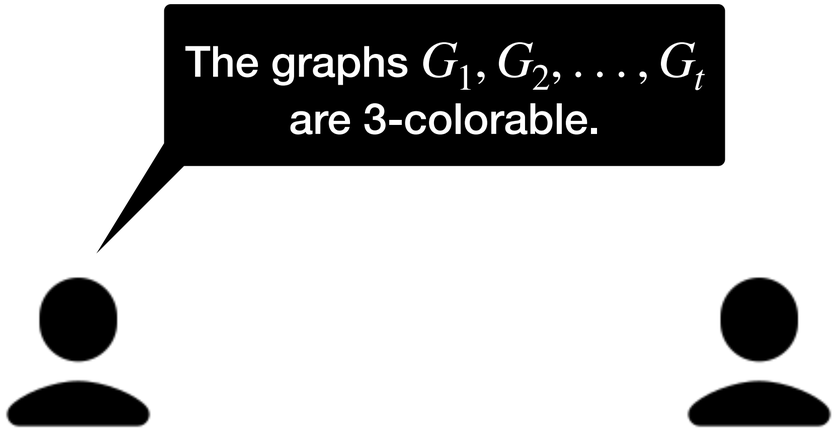
SNARGs are powerful, but sometimes more than needed.

Recent work shows for certain applications, a **more lightweight** primitive called **SNRDXs** suffices.

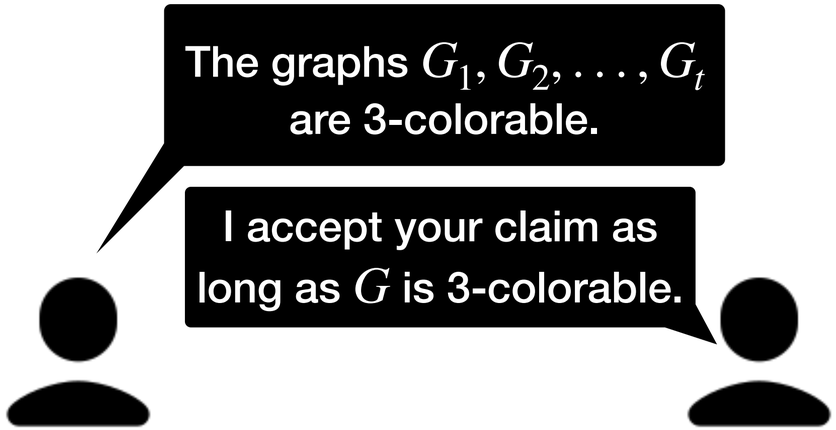
Succinct non-interactive reductions (SNRDXs)

Succinct non-interactive reductions (SNRDXs)

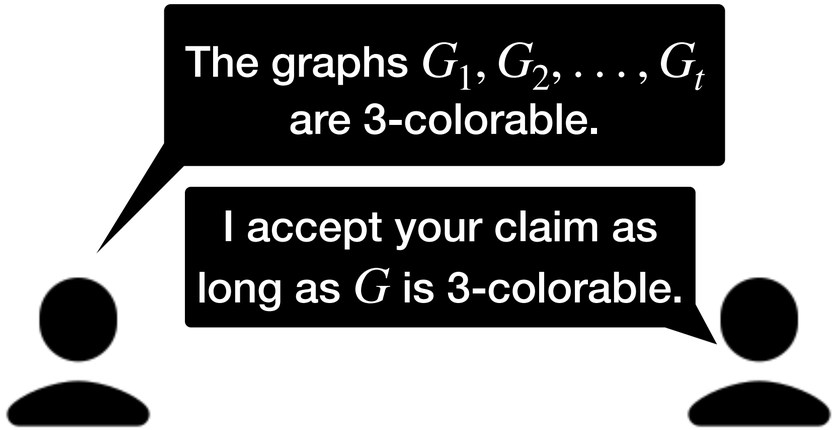
Succinct non-interactive reductions (SNRDXs)



Succinct non-interactive reductions (SNRDXs)

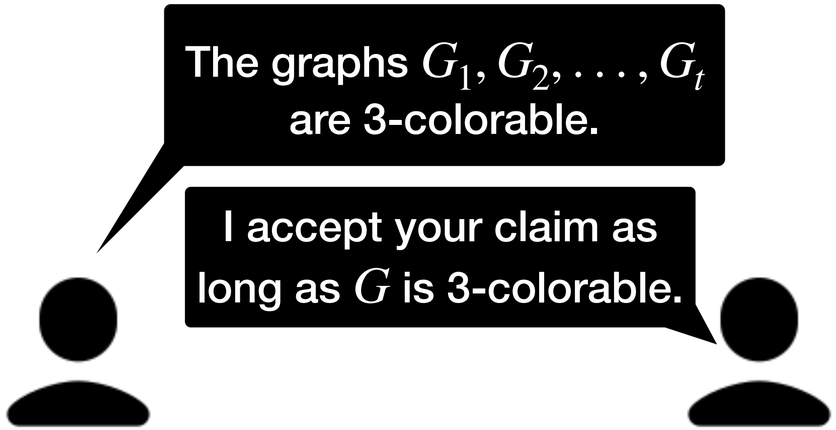


Succinct non-interactive reductions (SNRDXs)



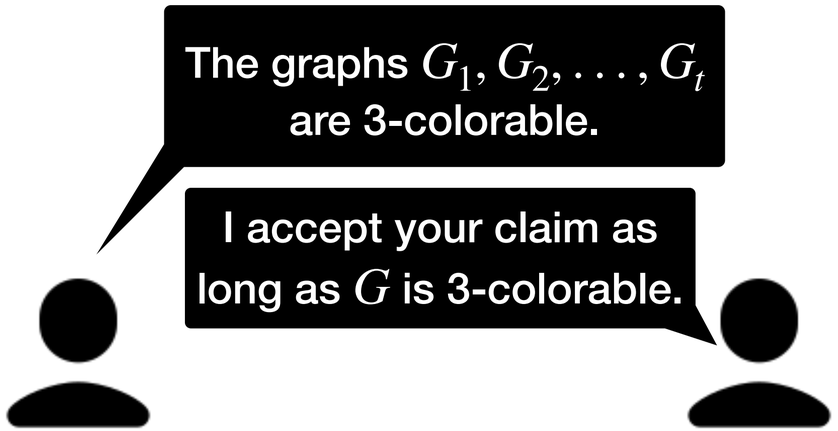
Then G is checked via other protocols

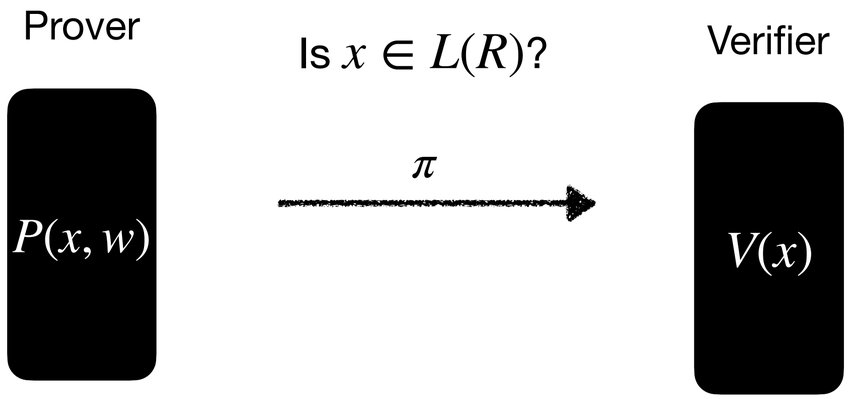
Succinct non-interactive reductions (SNRDXs)



Then G is checked via other protocols

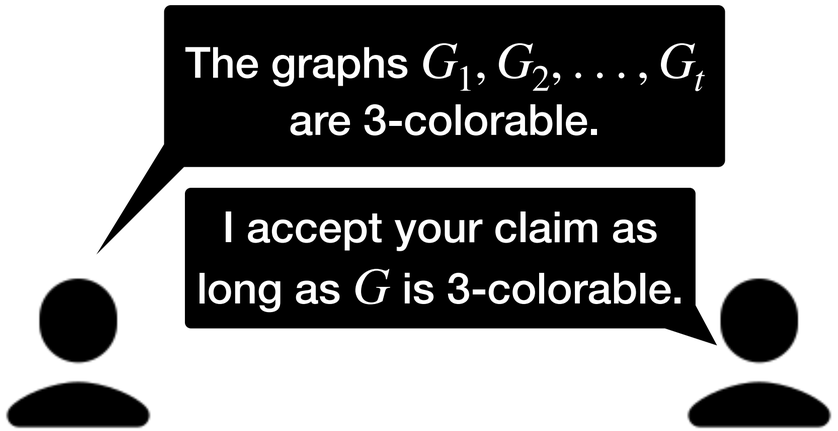
Succinct non-interactive reductions (SNRDXs)



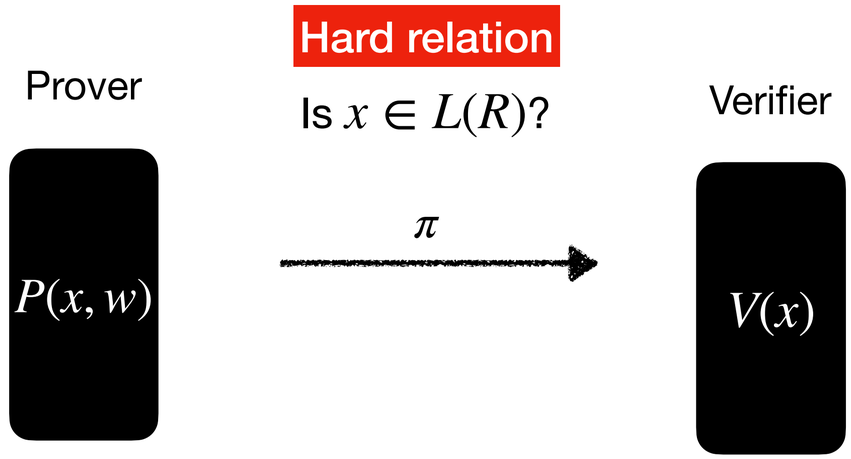


Then G is checked via other protocols

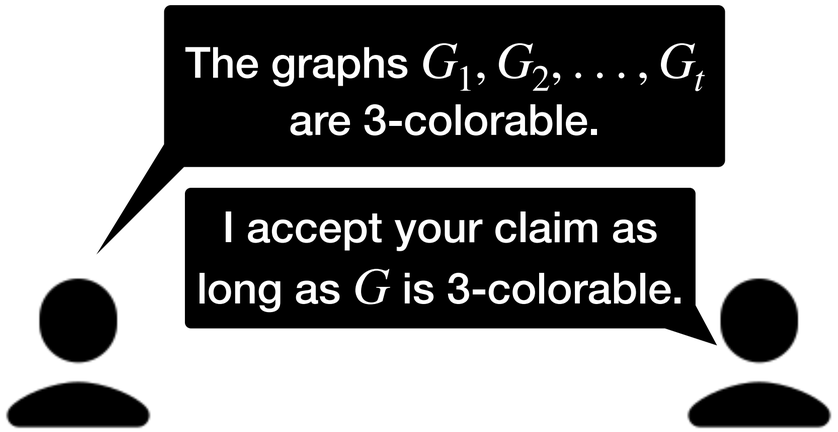
Succinct non-interactive reductions (SNRDXs)



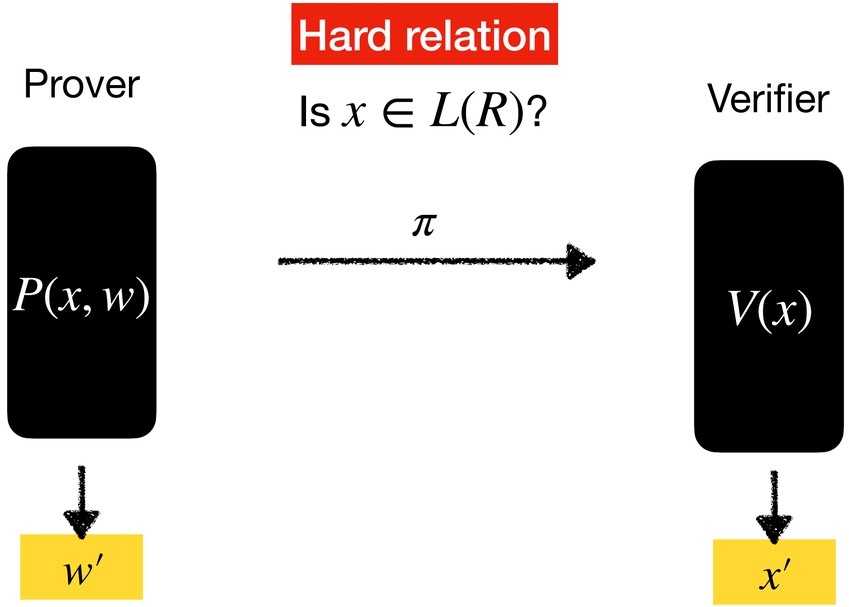
Then G is checked via other protocols



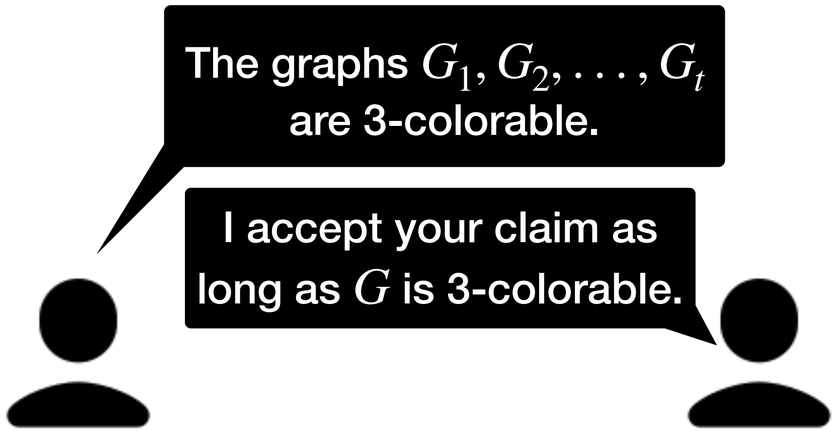
Succinct non-interactive reductions (SNRDXs)



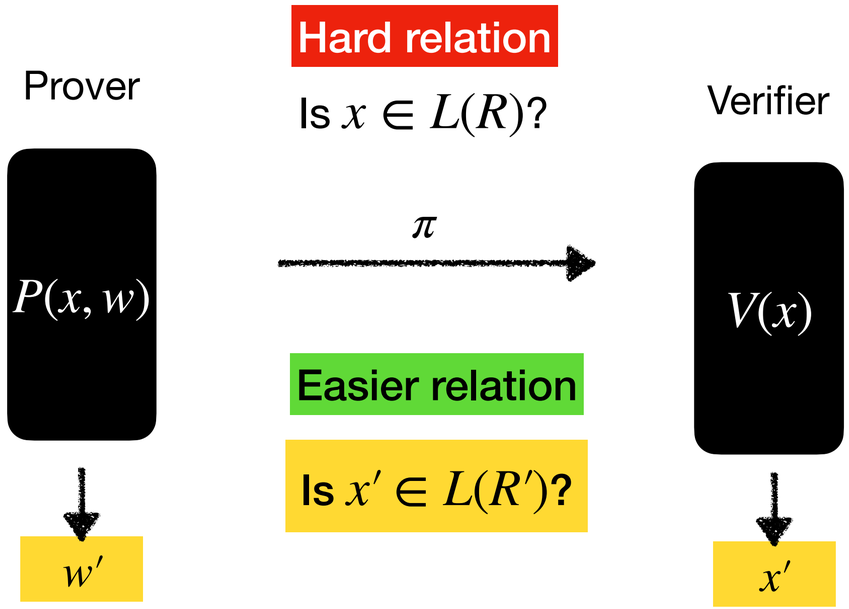
Then G is checked via other protocols



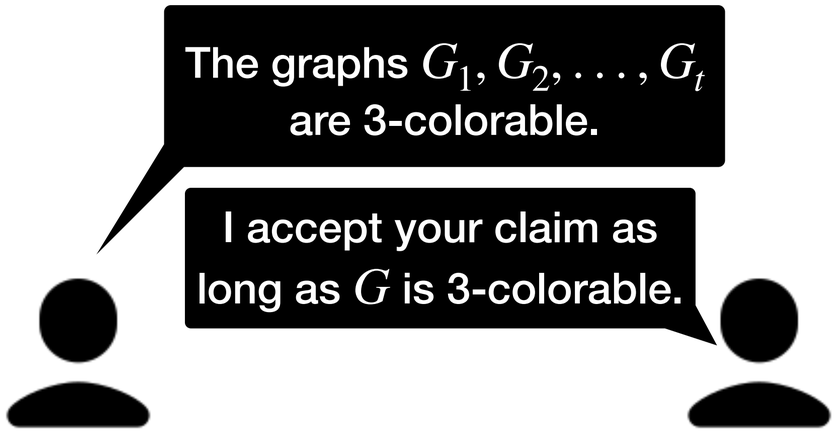
Succinct non-interactive reductions (SNRDXs)



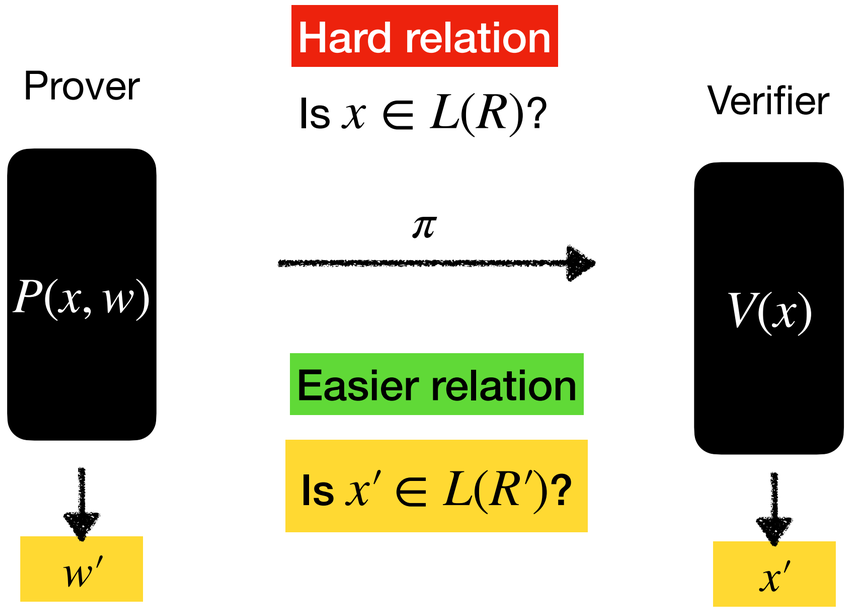
Then G is checked via other protocols



Succinct non-interactive reductions (SNRDXs)

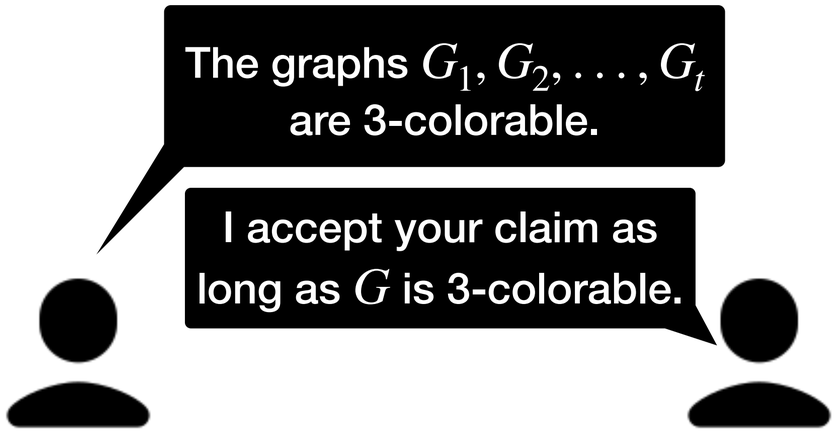


Then G is checked via other protocols

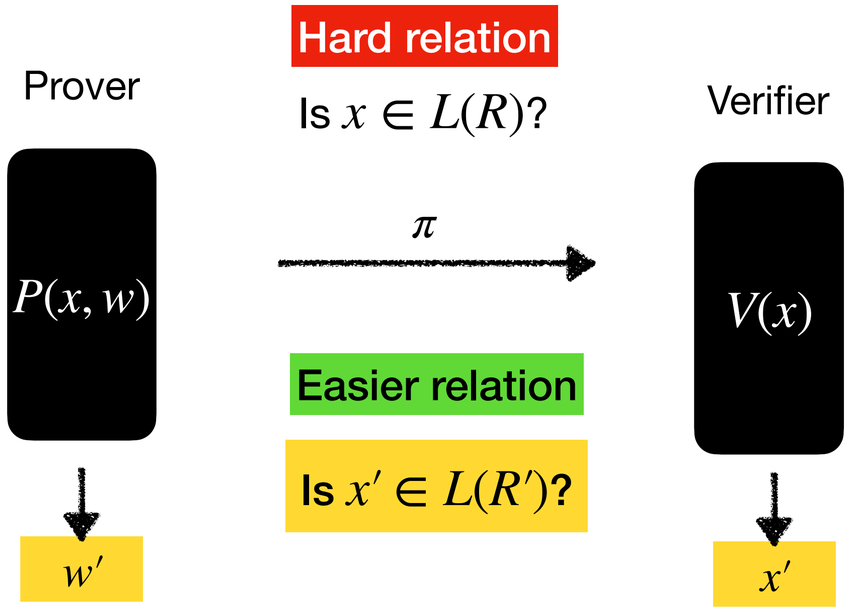


Completeness: $(x, w) \in R \to P(x, w)$ outputs (π, w') and $V(x, \pi)$ outputs x' such that $(x', w') \in R'$.

Succinct non-interactive reductions (SNRDXs)



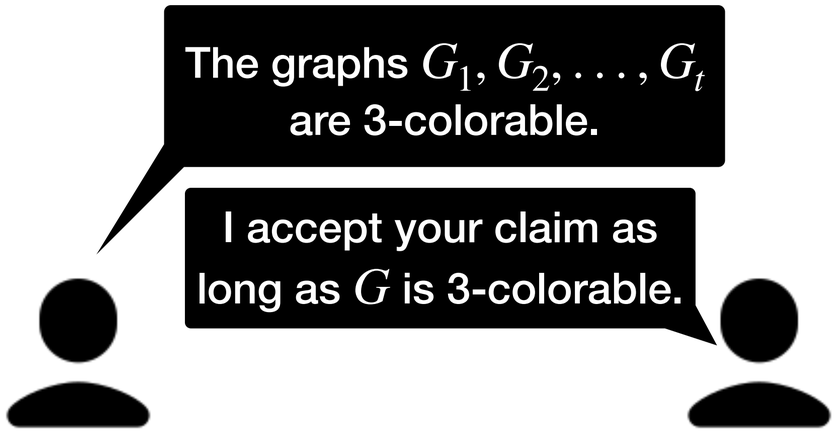
Then G is checked via other protocols



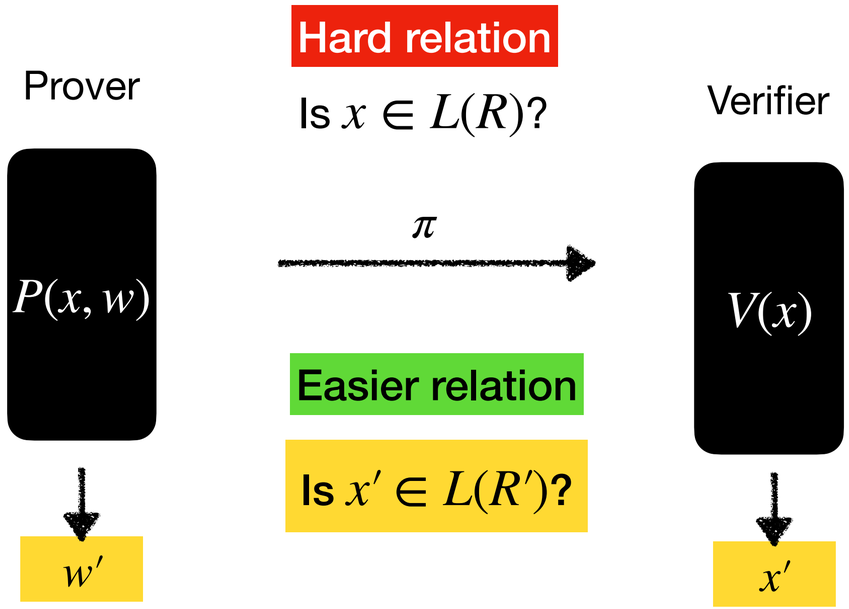
Completeness: $(x, w) \in R \rightarrow P(x, w)$ outputs (π, w') and $V(x, \pi)$ outputs x' such that $(x', w') \in R'$.

Soundness: $x \notin L(R) \rightarrow$ every **efficient** $\tilde{P}$ makes $V(x)$ output x' s.t. $x' \notin L(R')$ (up to a small error ϵ).

Succinct non-interactive reductions (SNRDXs)



Then G is checked via other protocols

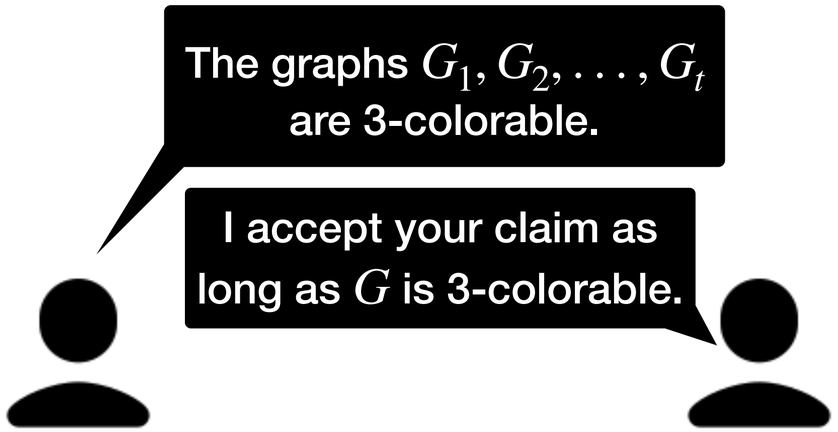


Completeness: $(x, w) \in R \rightarrow P(x, w)$ outputs (π, w') and $V(x, \pi)$ outputs x' such that $(x', w') \in R'$.

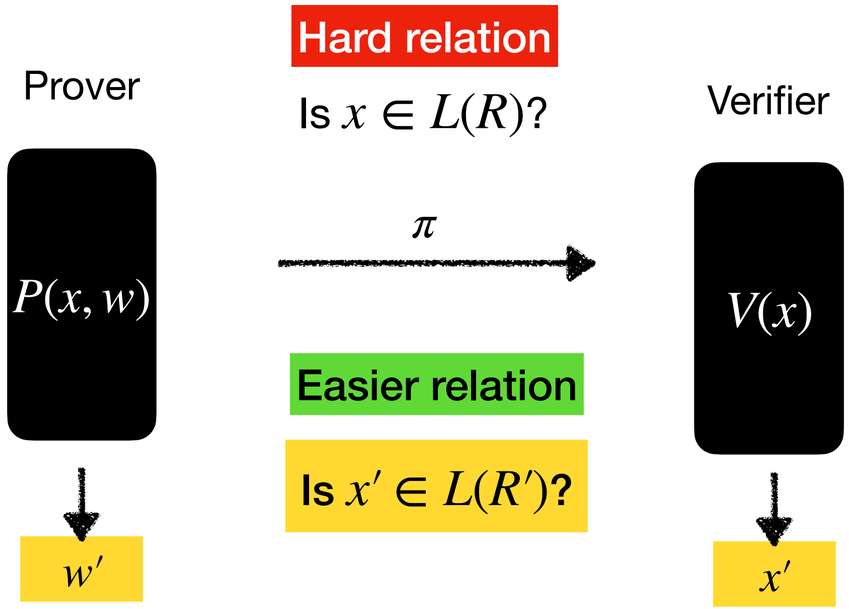
Soundness: $x \notin L(R) \rightarrow$ every **efficient** $\tilde{P}$ makes $V(x)$ output x' s.t. $x' \notin L(R')$ (up to a small error ϵ).

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

Succinct non-interactive reductions (SNRDXs)



Then G is checked via other protocols



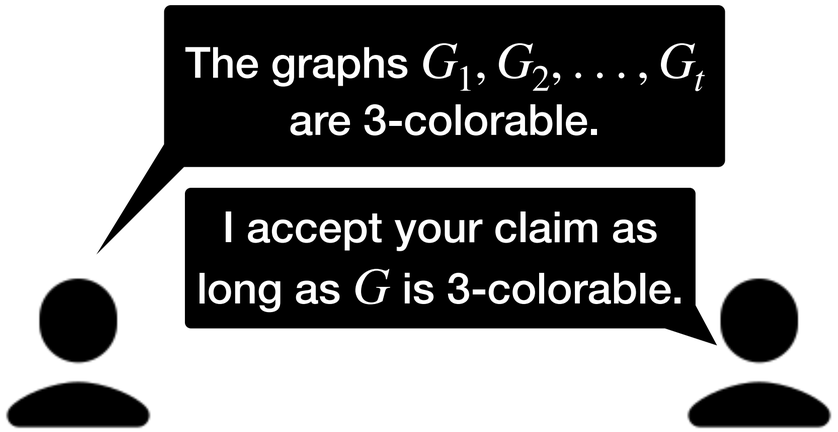
Completeness: $(x, w) \in R \rightarrow P(x, w)$ outputs (π, w') and $V(x, \pi)$ outputs x' such that $(x', w') \in R'$.

Soundness: $x \notin L(R) \rightarrow$ every **efficient** $\tilde{P}$ makes $V(x)$ output x' s.t. $x' \notin L(R')$ (up to a small error ϵ).

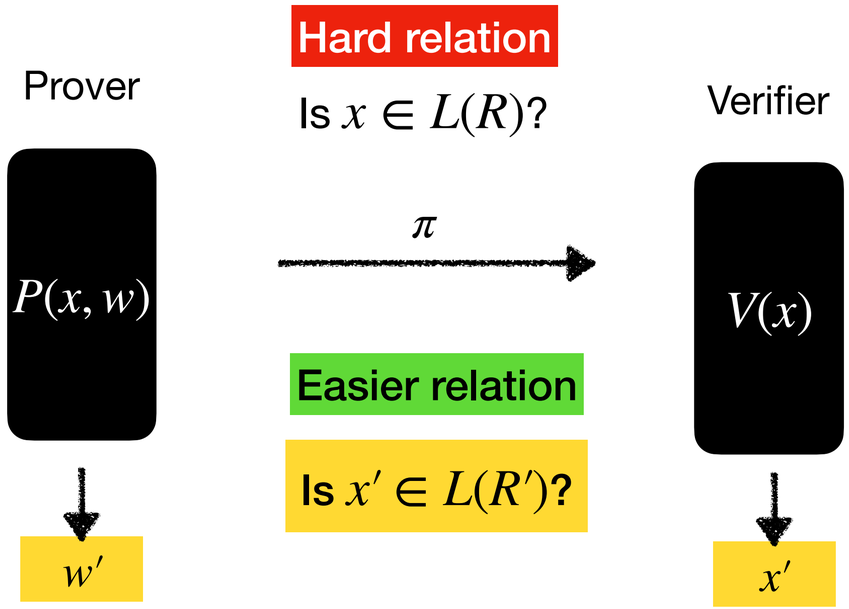
Succinctness: $|\pi| \ll |w|$.

Knowledge soundness: every **efficient** $\tilde{P}$ that outputs a witness w' s.t. $(x', w') \in R'$, must "know" w s.t. $(x, w) \in R$ (up to a small error κ).

Succinct non-interactive reductions (SNRDXs)



Then G is checked via other protocols



Completeness: $(x, w) \in R \rightarrow P(x, w)$ outputs (π, w') and $V(x, \pi)$ outputs x' such that $(x', w') \in R'$.

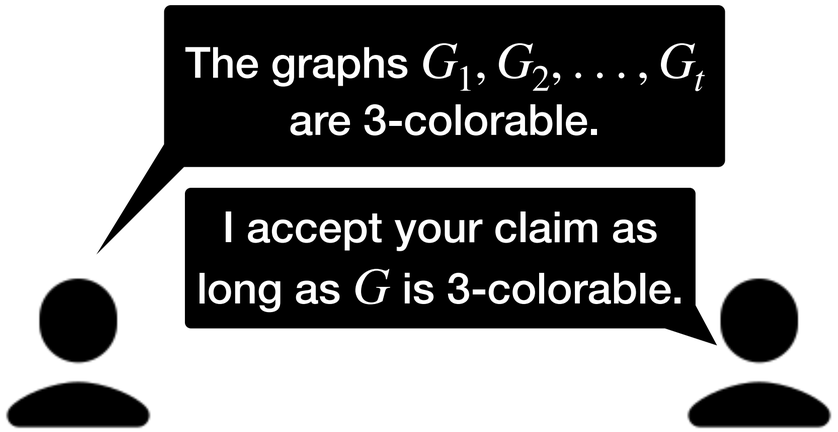
Soundness: $x \notin L(R) \rightarrow$ every **efficient** $\tilde{P}$ makes $V(x)$ output x' s.t. $x' \notin L(R')$ (up to a small error ϵ).

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

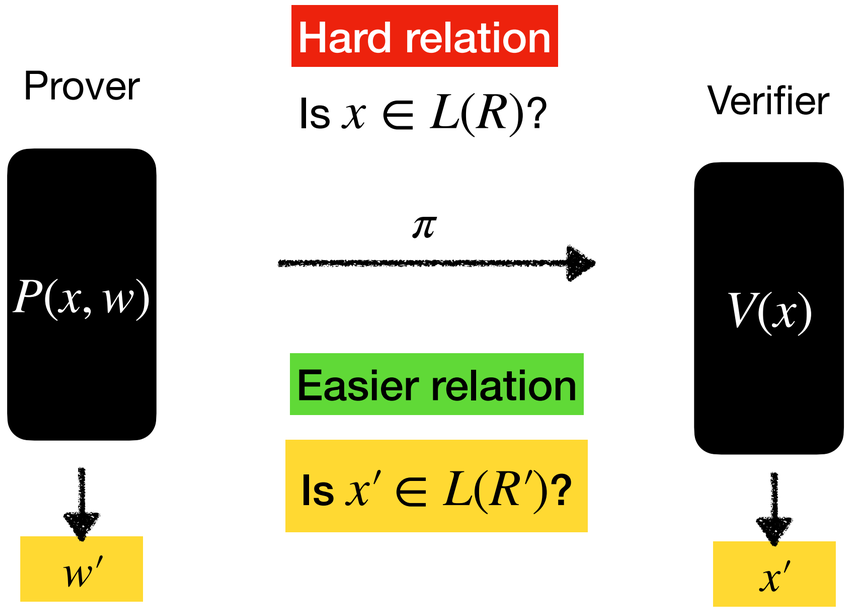
Knowledge soundness: every **efficient** $\tilde{P}$ that outputs a witness w' s.t. $(x', w') \in R'$, must "know" w s.t. $(x, w) \in R$ (up to a small error κ).

Why are SNRDXs useful?

Succinct non-interactive reductions (SNRDXs)



Then G is checked via other protocols



Completeness: $(x, w) \in R \rightarrow P(x, w)$ outputs (π, w') and $V(x, \pi)$ outputs x' such that $(x', w') \in R'$.

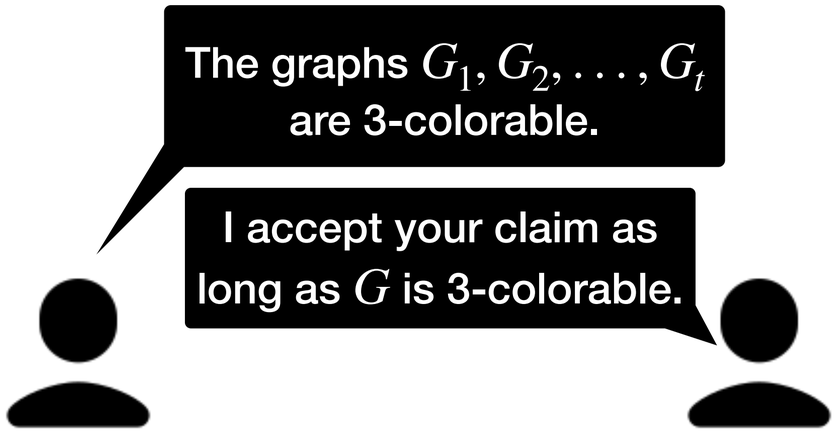
Soundness: $x \notin L(R) \rightarrow$ every **efficient** $\tilde{P}$ makes $V(x)$ output x' s.t. $x' \notin L(R')$ (up to a small error ϵ).

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

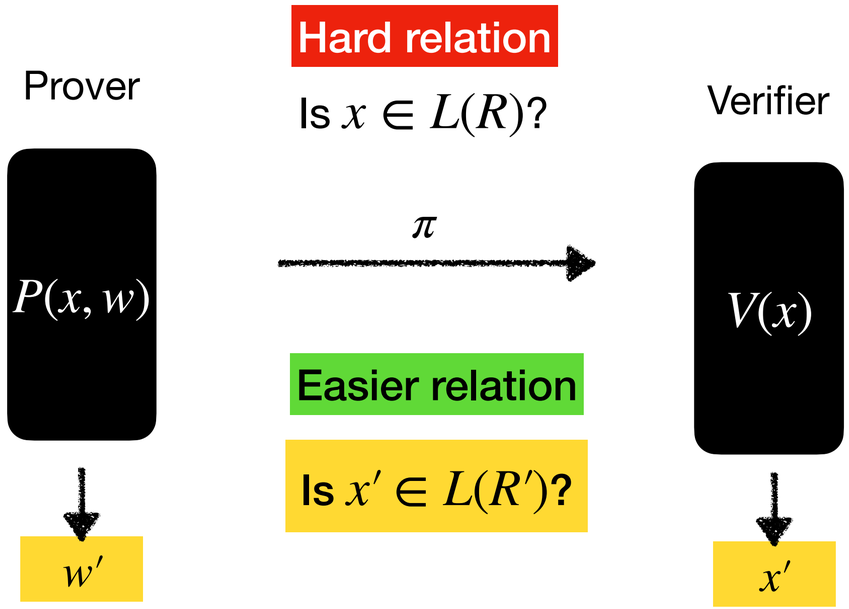
Knowledge soundness: every **efficient** $\tilde{P}$ that outputs a witness w' s.t. $(x', w') \in R'$, must "know" w s.t. $(x, w) \in R$ (up to a small error κ).

Why are SNRDXs useful? **(+) generalization of SNARGs**: SNARG for R = SNRDX from R to trivial relation $R' = \{(x', w') : x' = 1\}$.

Succinct non-interactive reductions (SNRDXs)



Then G is checked via other protocols



Completeness: $(x, w) \in R \rightarrow P(x, w)$ outputs (π, w') and $V(x, \pi)$ outputs x' such that $(x', w') \in R'$.

Soundness: $x \notin L(R) \rightarrow$ every **efficient** $\tilde{P}$ makes $V(x)$ output x' s.t. $x' \notin L(R')$ (up to a small error ϵ).

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

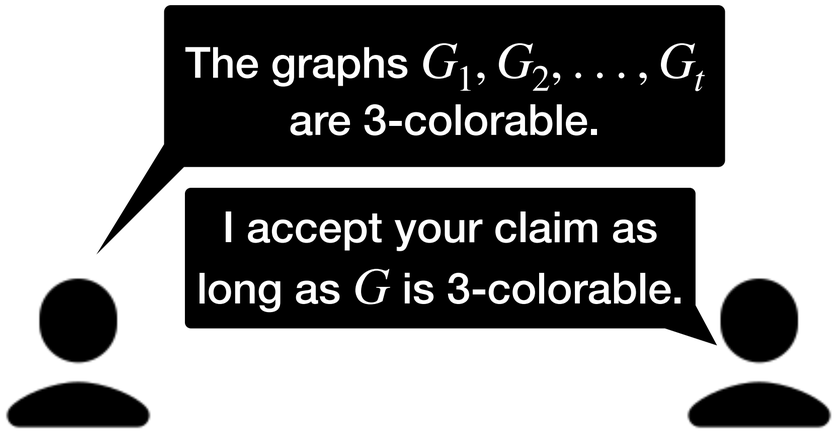
Knowledge soundness: every **efficient** $\tilde{P}$ that outputs a witness w' s.t. $(x', w') \in R'$, must "know" w s.t. $(x, w) \in R$ (up to a small error κ).

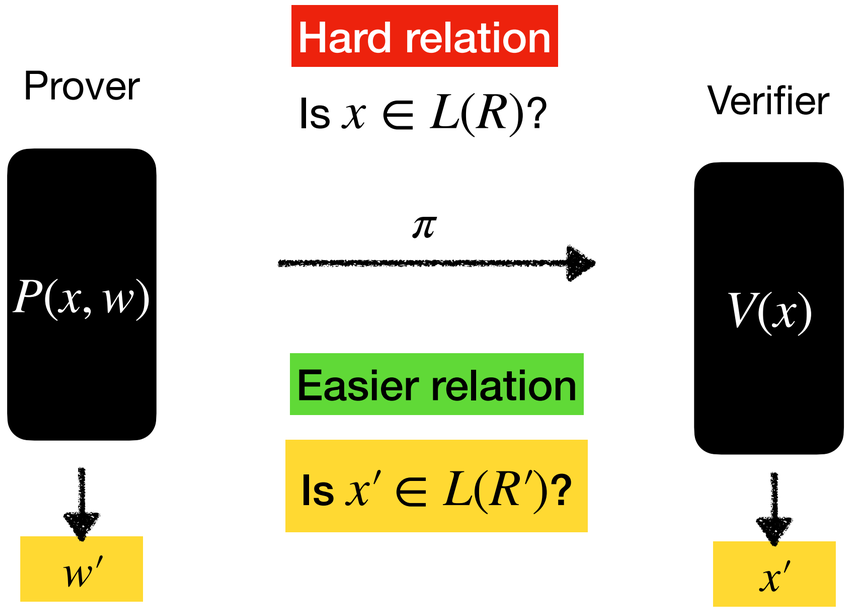
Why are SNRDXs useful?

(+) generalization of SNARGs: SNARG for R = SNRDX from R to trivial relation $R' = \{(x', w') : x' = 1\}$.

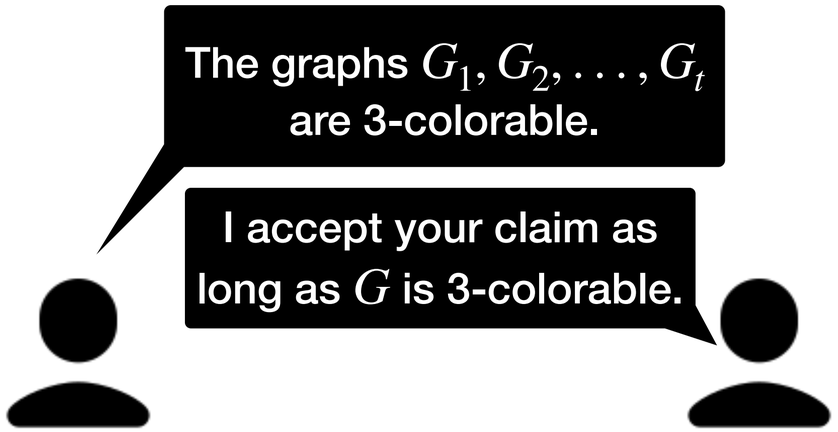
(+) cheaper to construct than SNARGs for some relations R'.

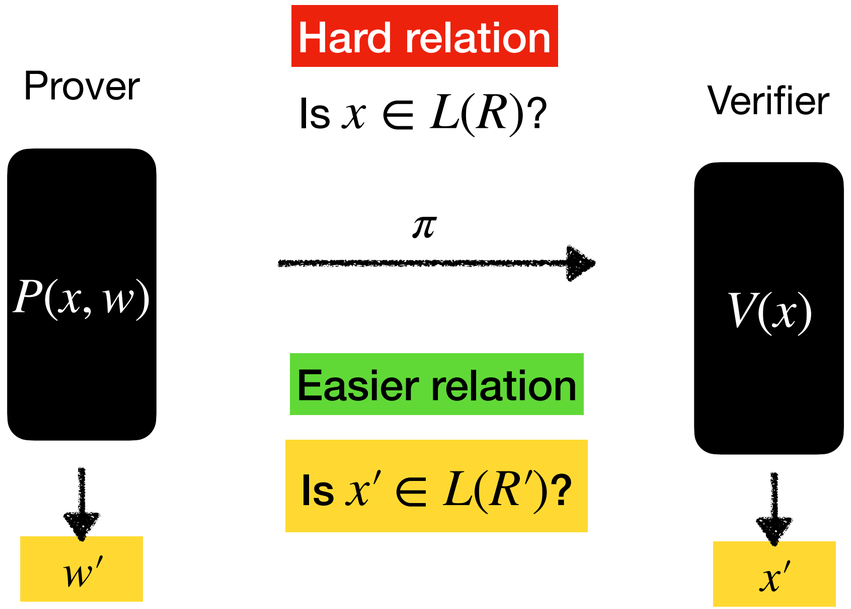
Succinct non-interactive reductions (SNRDXs)





Succinct non-interactive reductions (SNRDXs)

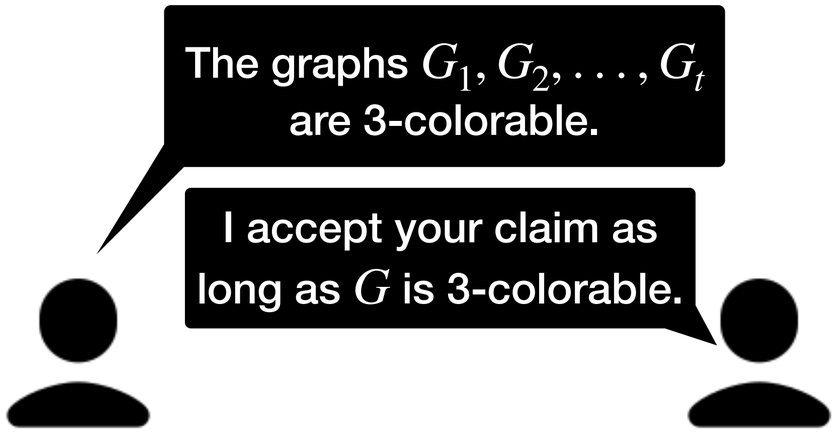


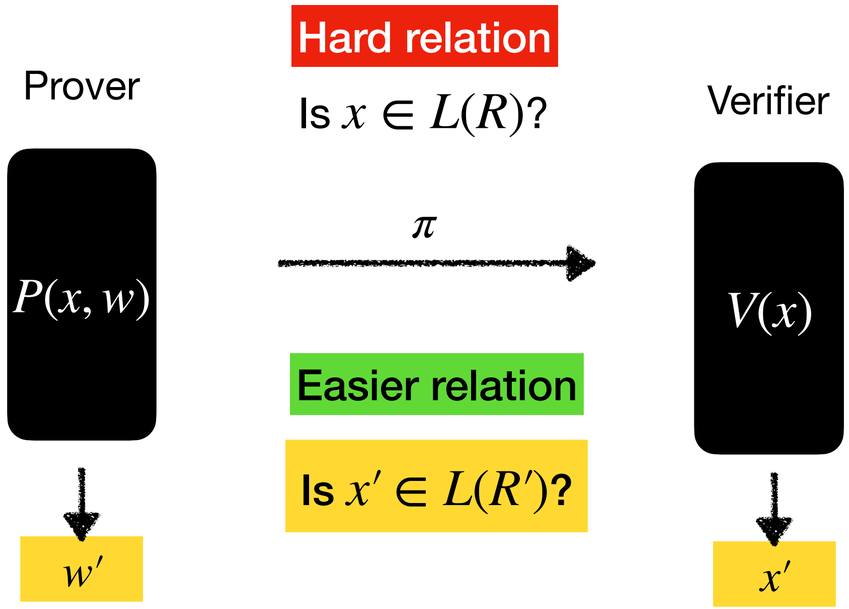


SNRDXs have **numerous real-world applications**.

SNRDXs (packaged as accumulation schemes or folding schemes) yield **proof-carrying data**, **incrementally verifiable computation**, etc.

Succinct non-interactive reductions (SNRDXs)





SNRDXs have **numerous real-world applications**.

SNRDXs (packaged as accumulation schemes or folding schemes) yield **proof-carrying data**, **incrementally verifiable computation**, etc.

NEXUS

RISC ZERO

...

Where do SNARGs/SNRDXs come from?

Where do SNARGs/SNRDXs come from?

A few places.

Where do SNARGs/SNRDXs come from?

A few places. Our focus:

Where do SNARGs/SNRDXs come from?

A few places. Our focus:

Hash-based SNARGs/SNRDXs

Where do SNARGs/SNRDXs come from?

A few places. Our focus:

Hash-based SNARGs/SNRDXs

✅ Efficient

Where do SNARGs/SNRDXs come from?

A few places. Our focus:

Hash-based SNARGs/SNRDXs

✅ Efficient ✅ Public (transparent) setup

Where do SNARGs/SNRDXs come from?

A few places. Our focus:

Hash-based SNARGs/SNRDXs

✅ Efficient ✅ Public (transparent) setup ✅ Plausibly post-quantum

Where do SNARGs/SNRDXs come from?

A few places. Our focus:

Hash-based SNARGs/SNRDXs

✅ Efficient

✅ Public (transparent) setup

✅ Plausibly post-quantum

today

Recall: SNARG BCS[IOP, MT]

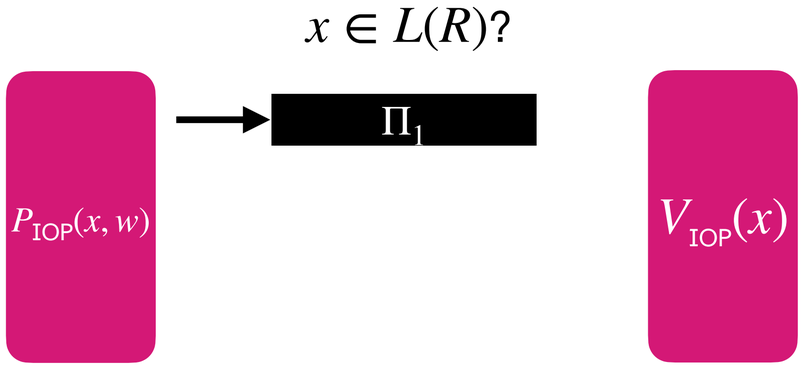
Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)

$x \in L(R)$?

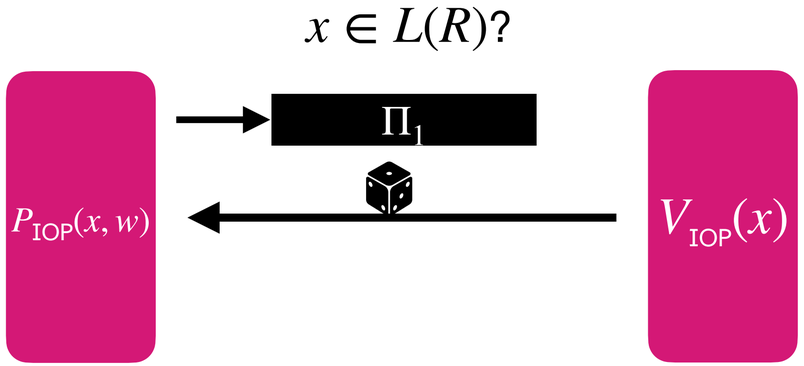
Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)



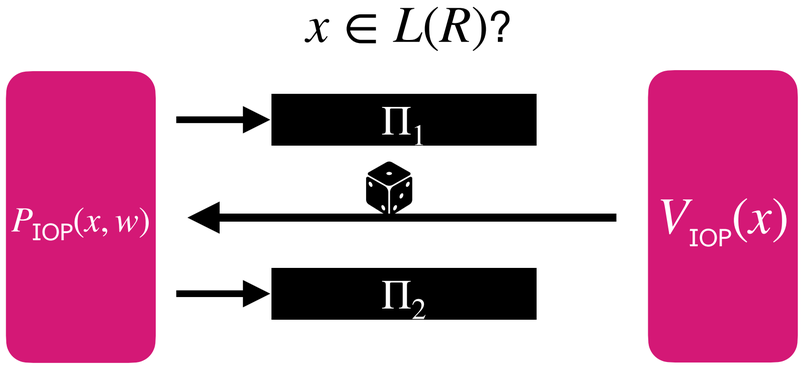
Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)



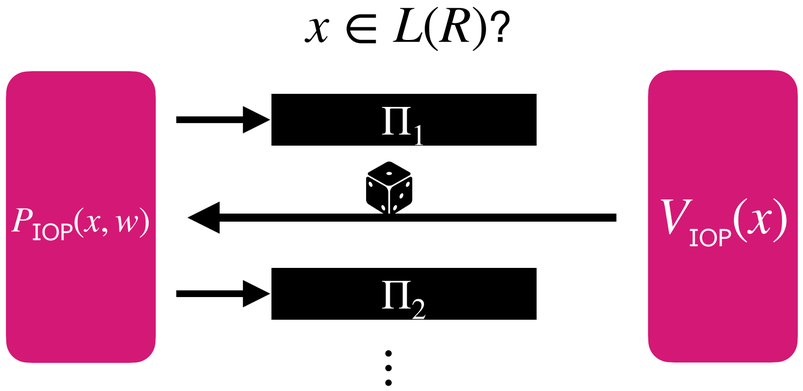
Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)



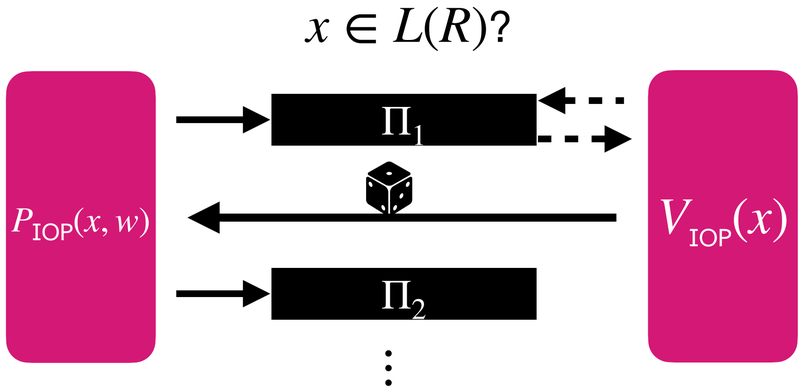
Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)



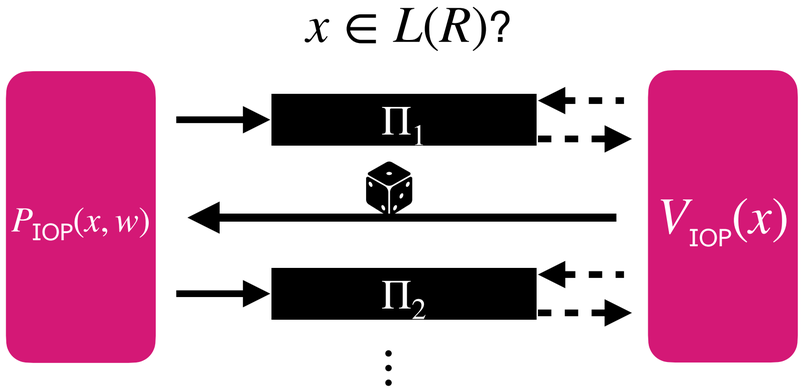
Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)



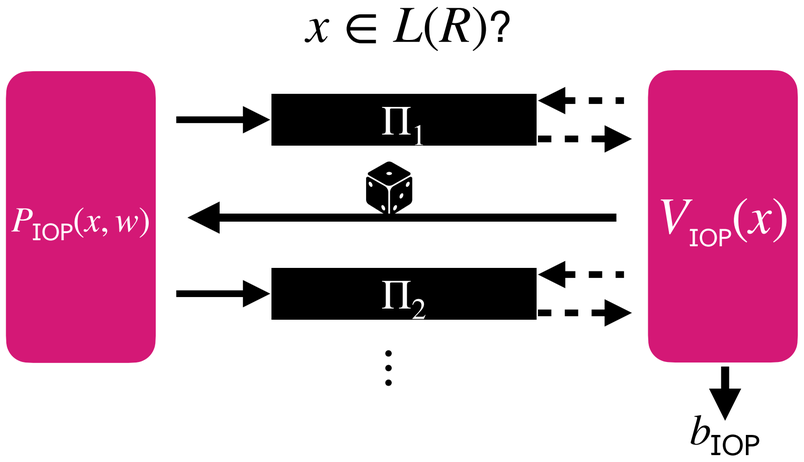
Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)



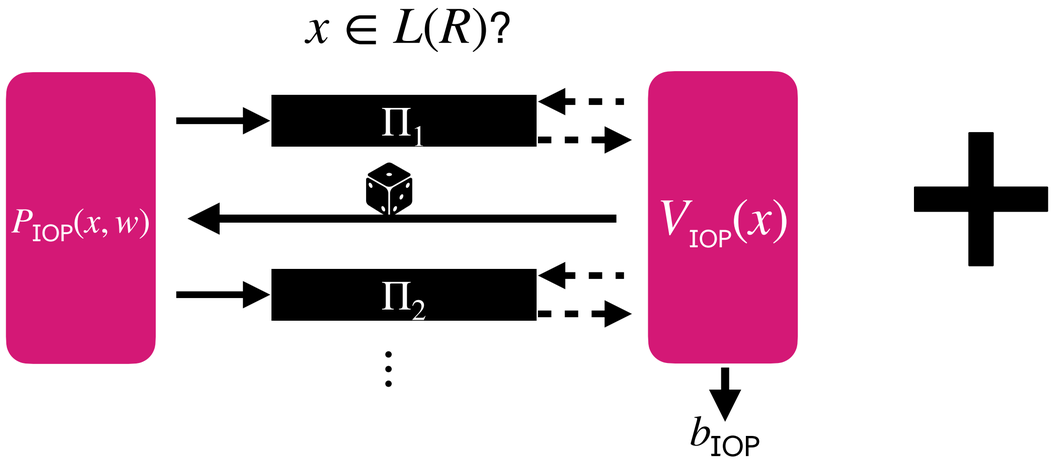
Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)



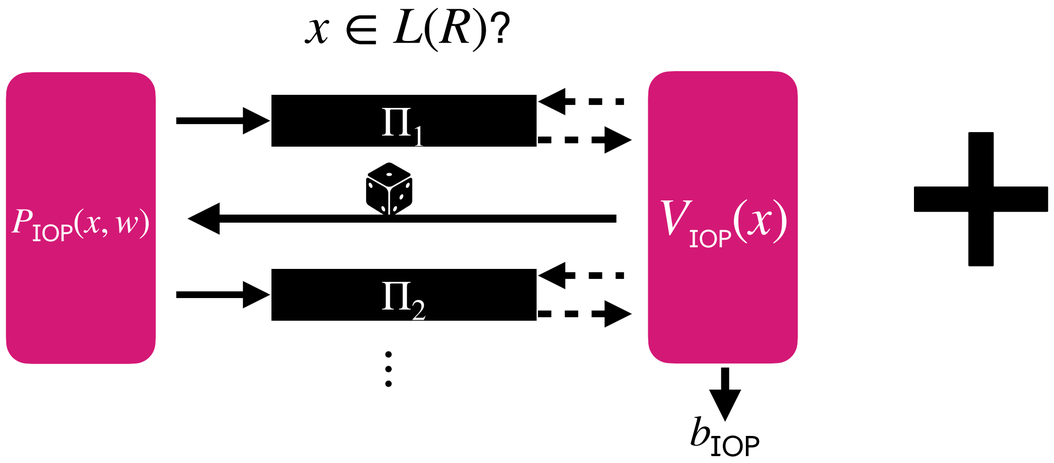
Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)



Recall: SNARG BCS[IOP, MT]

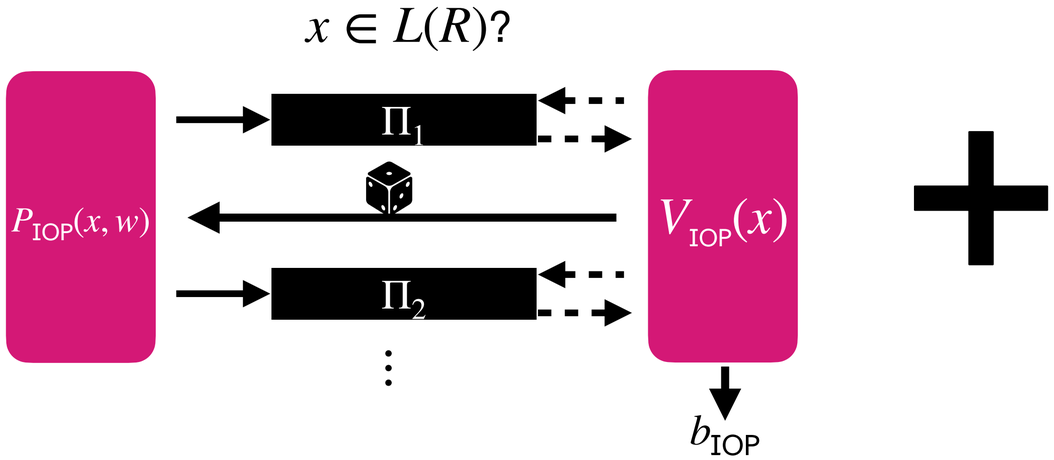
Ingredient #1: Interactive oracle proof (IOP)



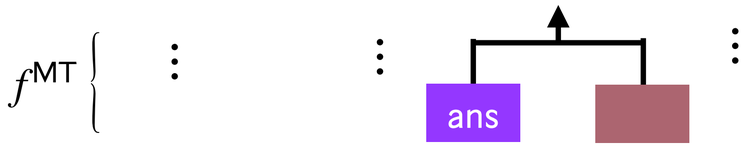
Ingredient #2: Merkle commitment scheme (MT)

Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)

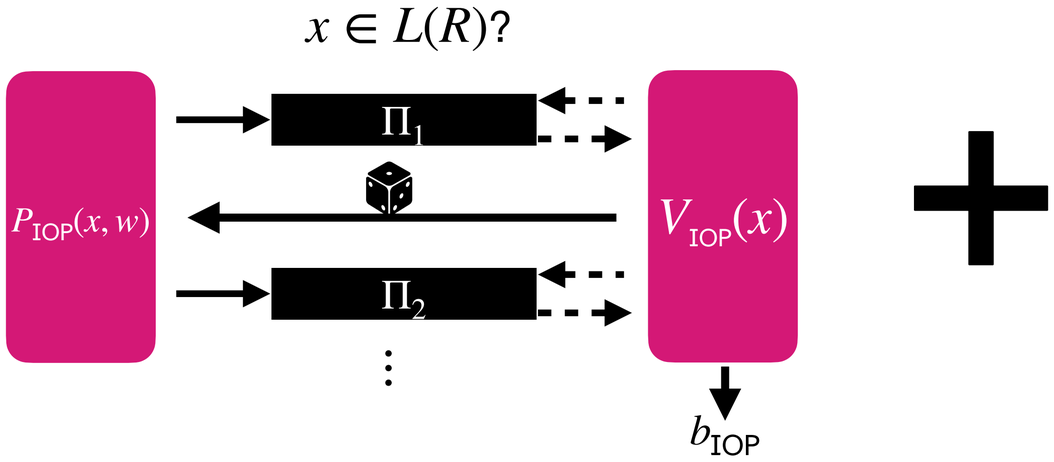


Ingredient #2: Merkle commitment scheme (MT)

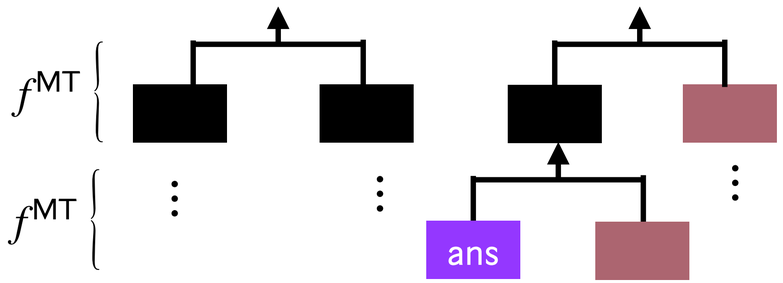


Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)

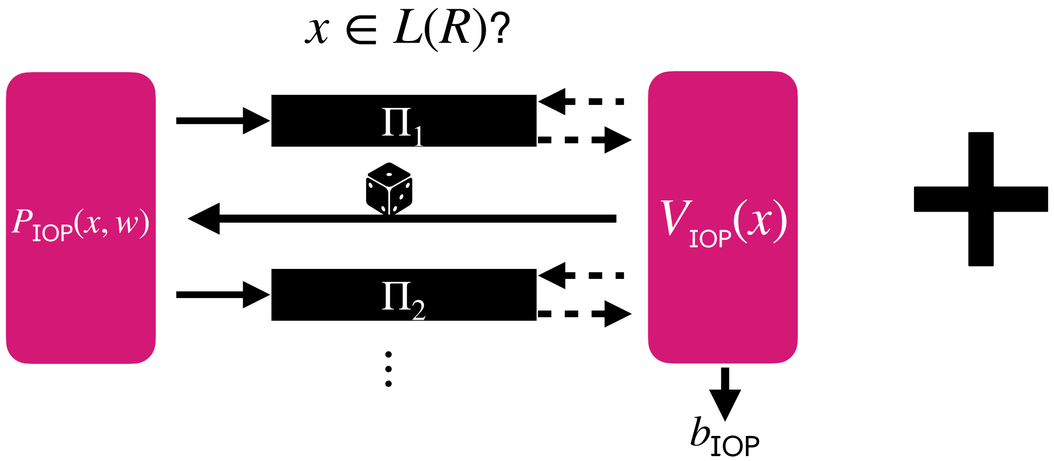


Ingredient #2: Merkle commitment scheme (MT)

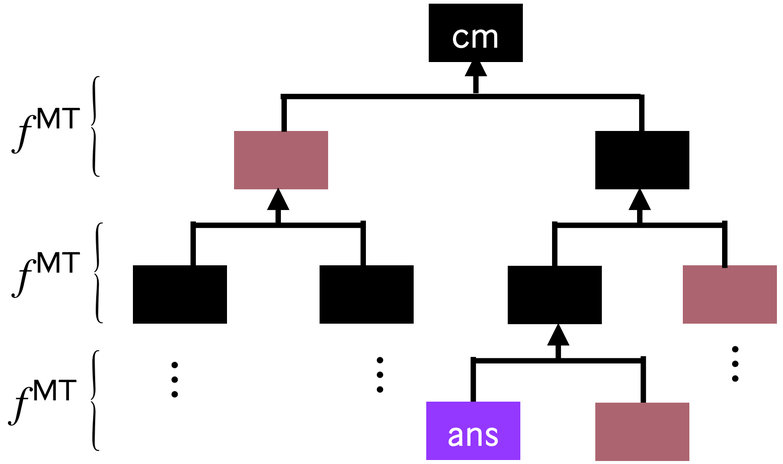


Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)

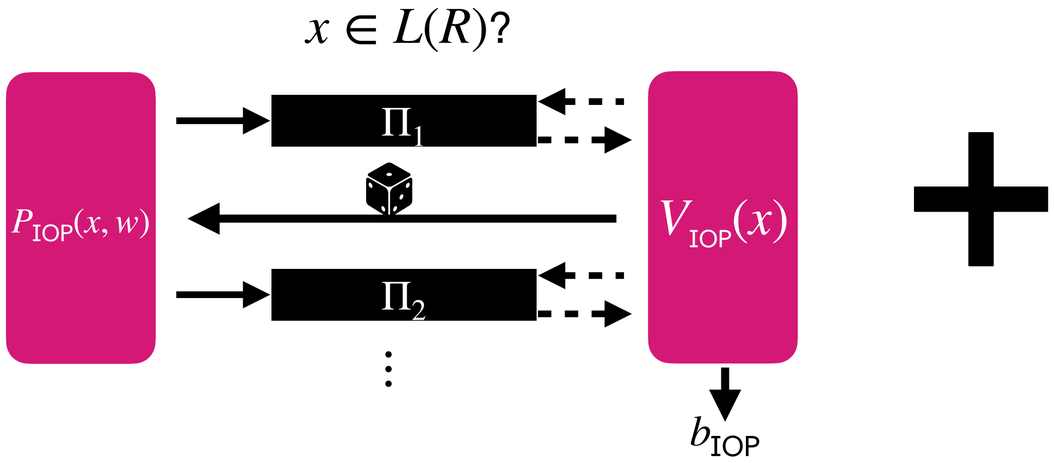


Ingredient #2: Merkle commitment scheme (MT)

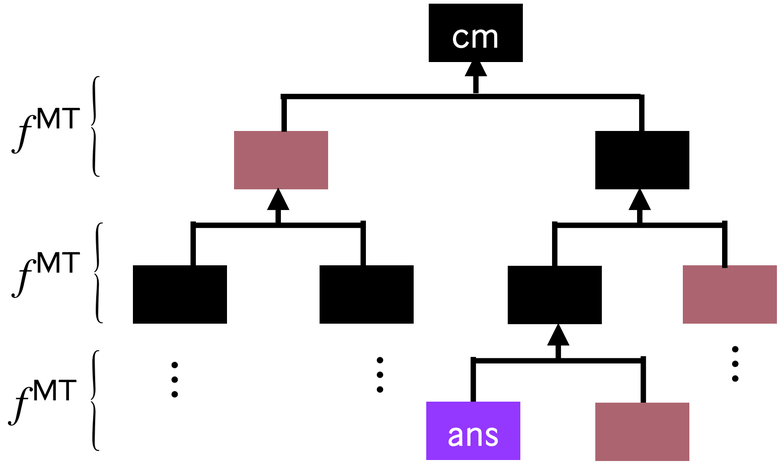


Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)

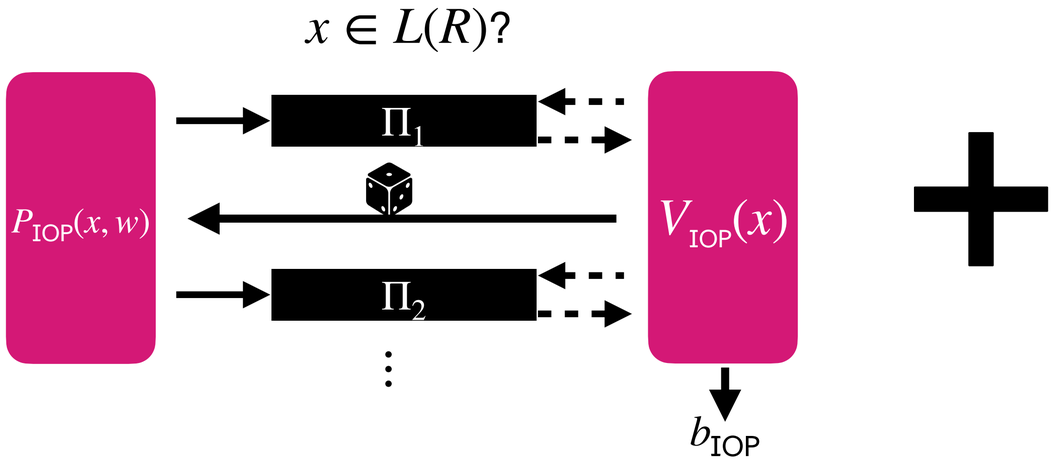


Ingredient #2: Merkle commitment scheme (MT)

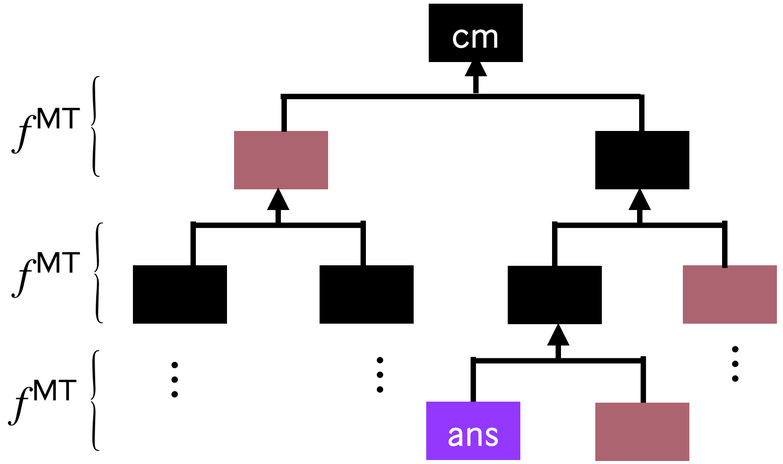


Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)

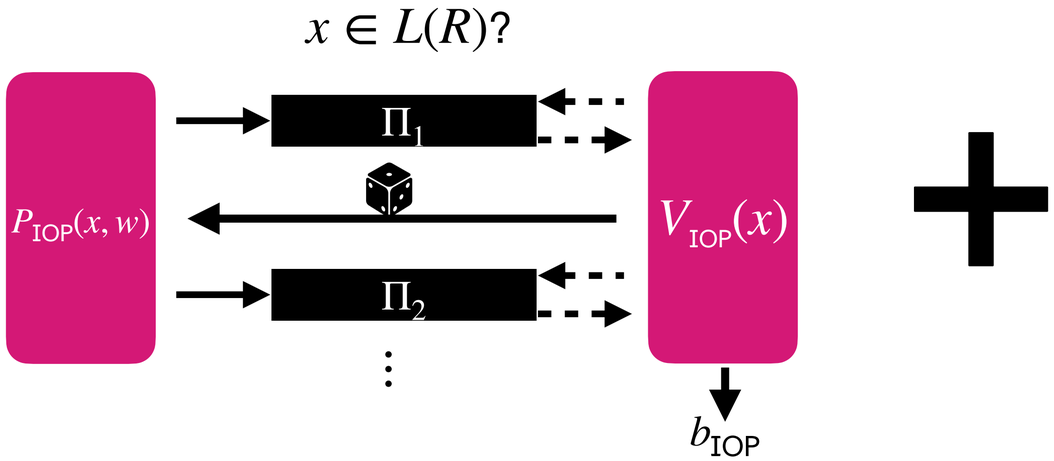


Ingredient #2: Merkle commitment scheme (MT)

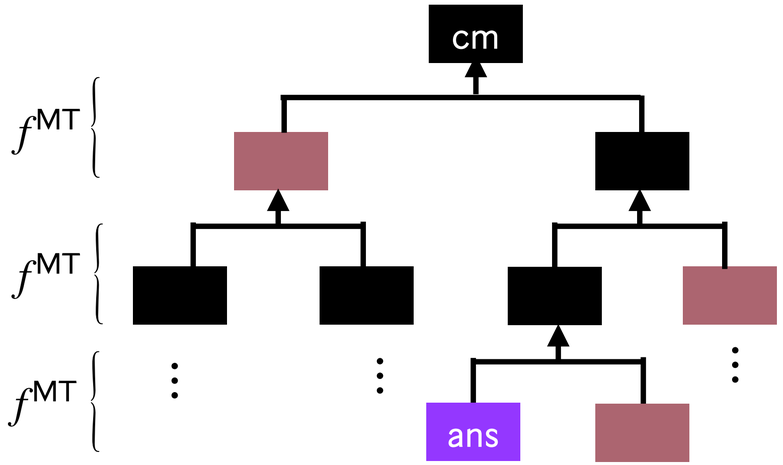


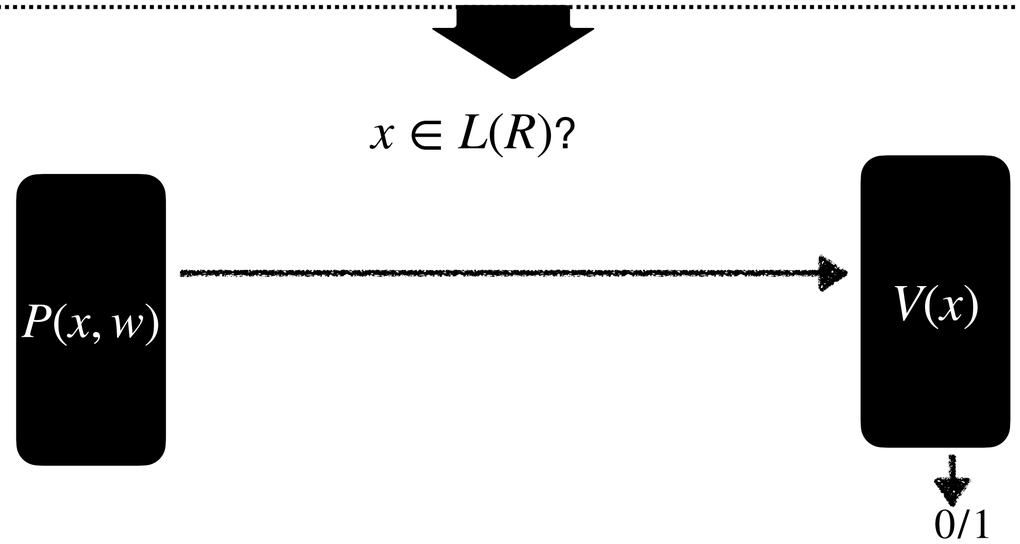
Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)



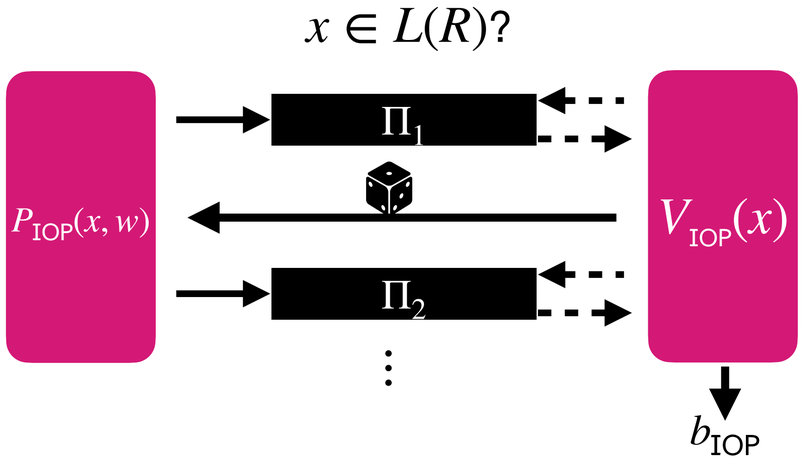
Ingredient #2: Merkle commitment scheme (MT)



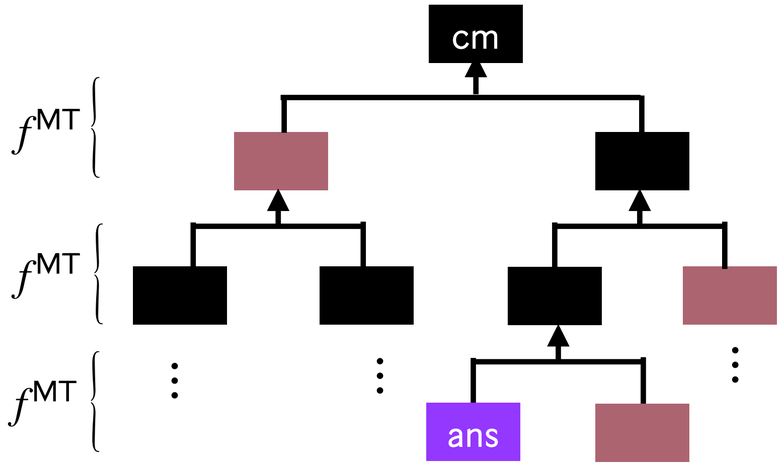


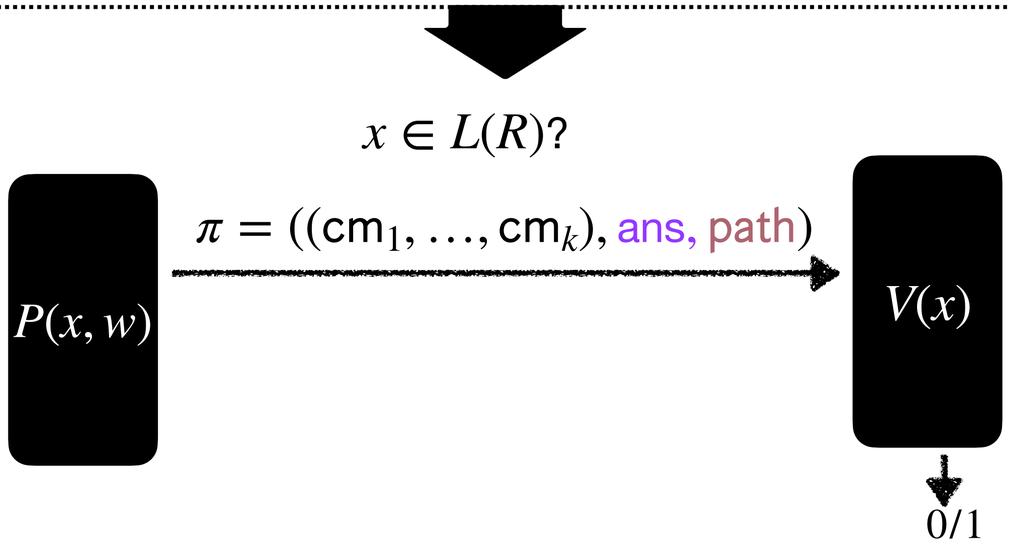
Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)



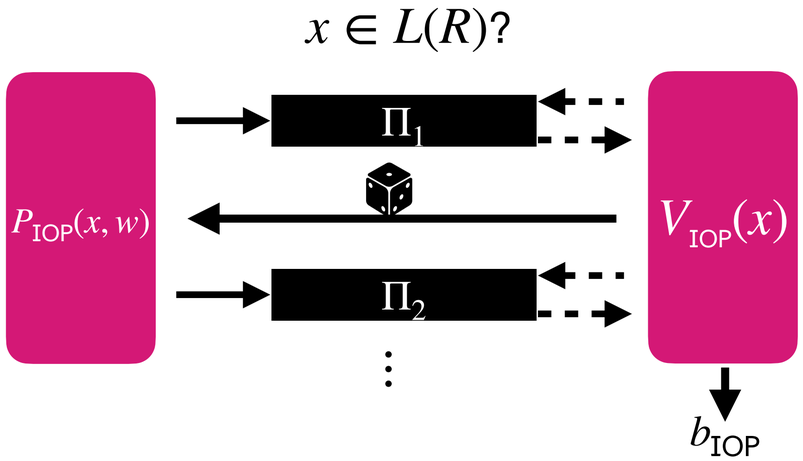
Ingredient #2: Merkle commitment scheme (MT)





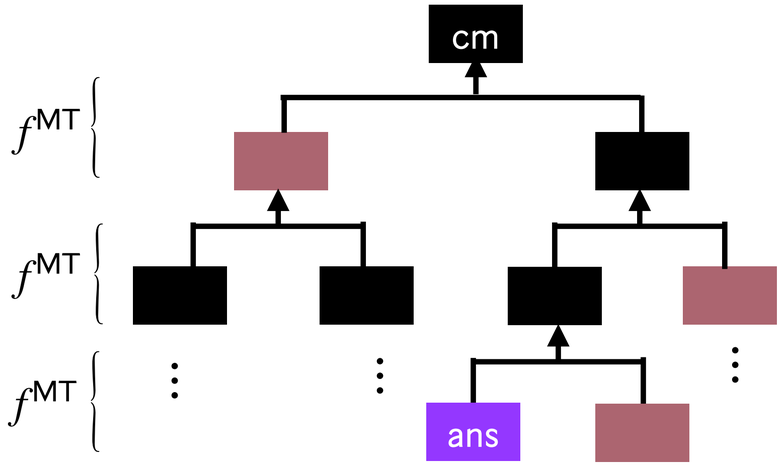
Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)

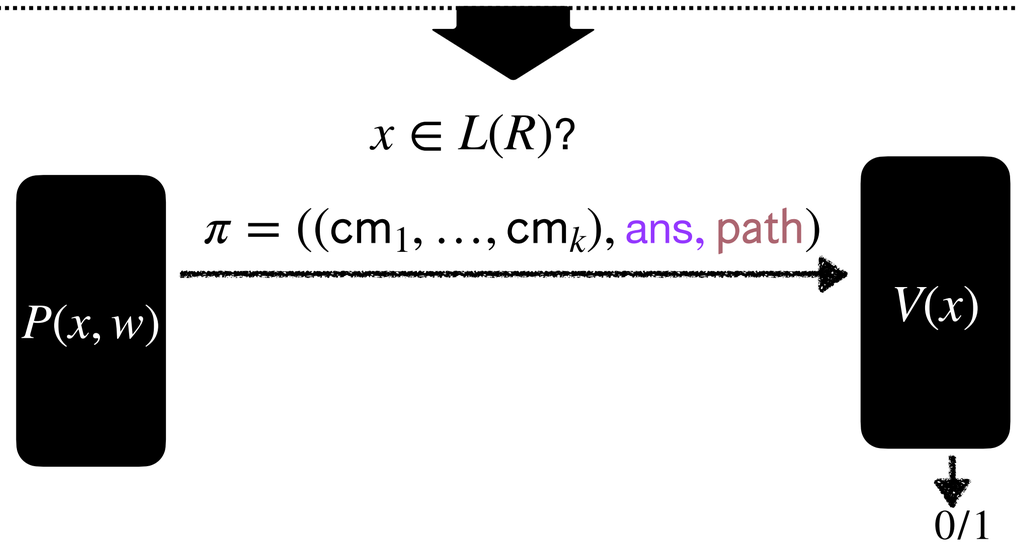


+

Ingredient #2: Merkle commitment scheme (MT)

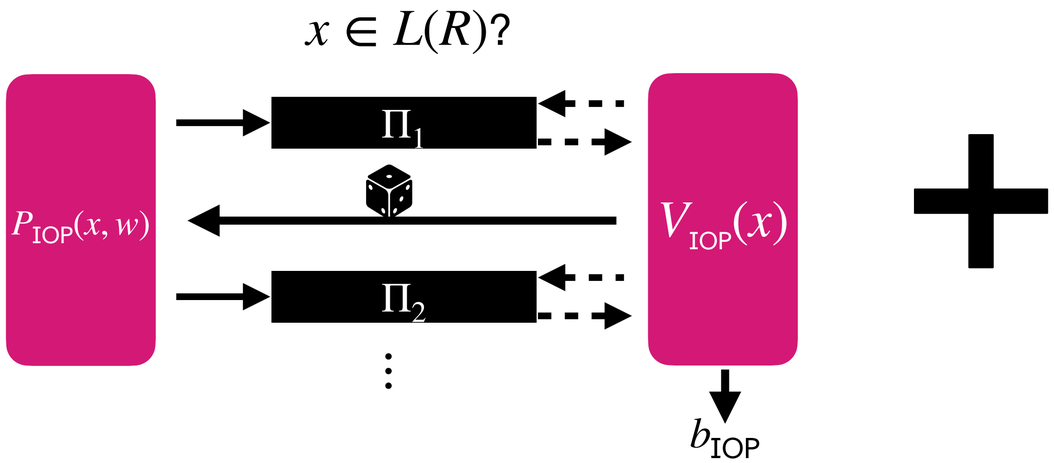


The BCS protocol is widely-used in practice.

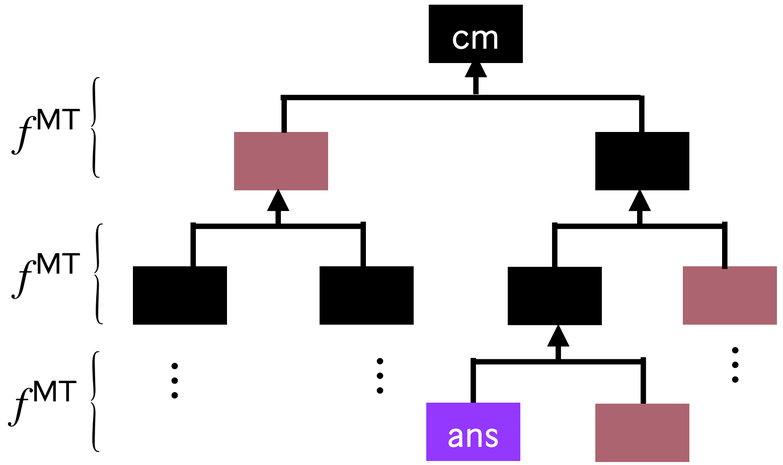


Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)

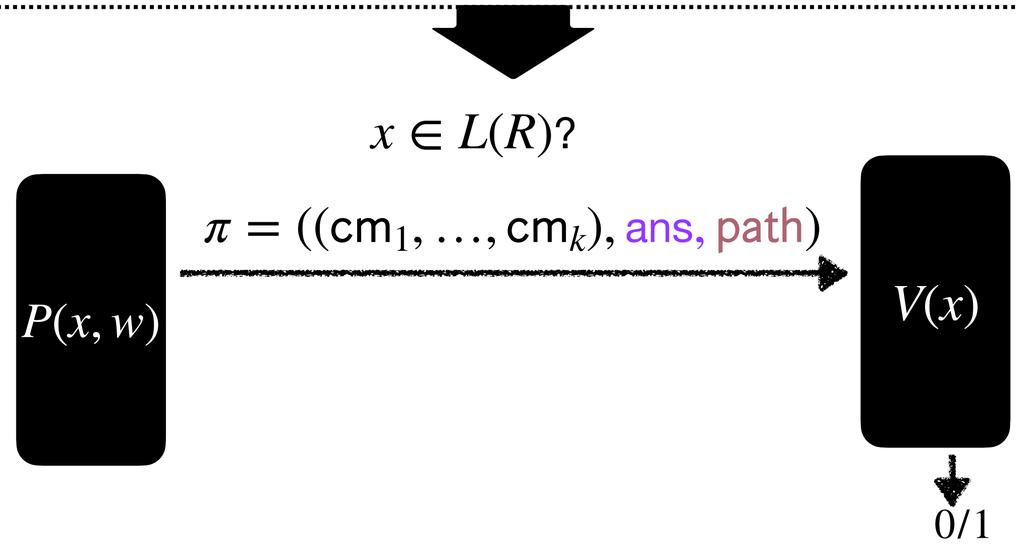


Ingredient #2: Merkle commitment scheme (MT)



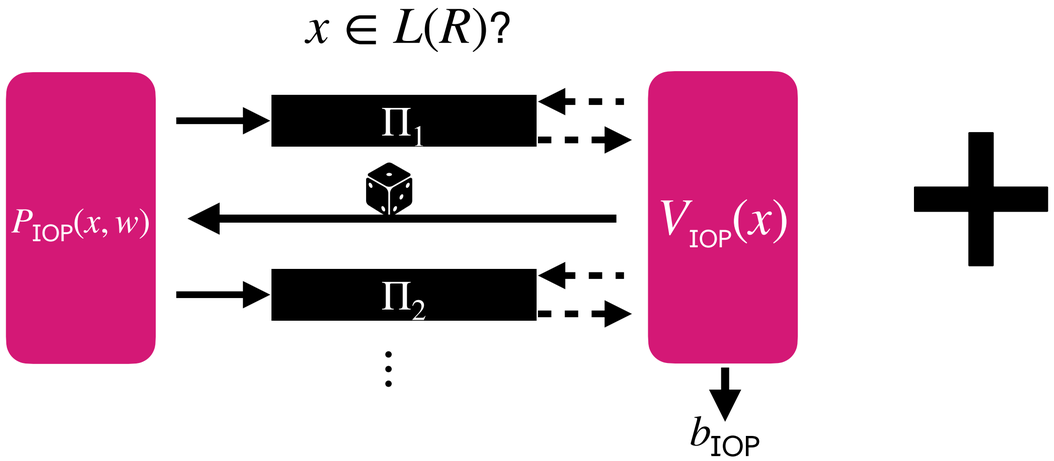
The BCS protocol is widely-used in practice.

Security is analyzed in an ideal model: **random oracle model**.

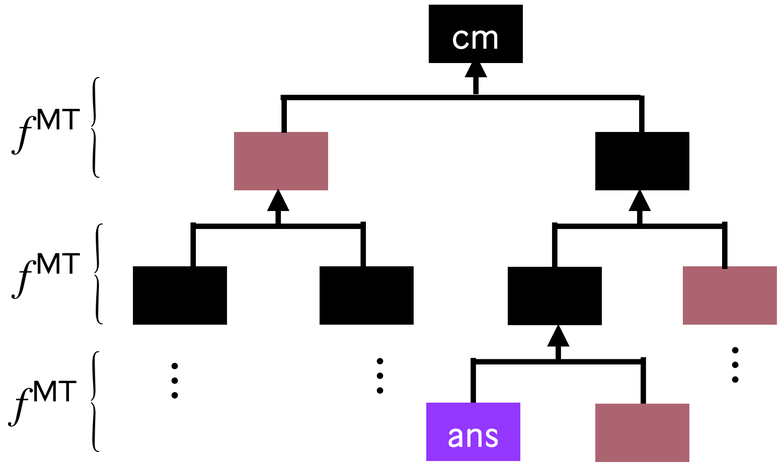


Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)



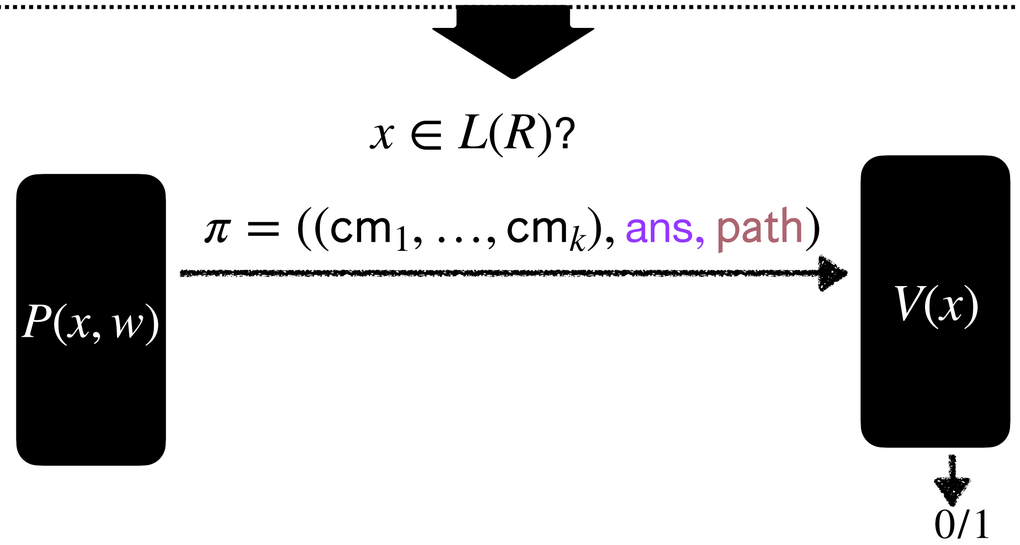
Ingredient #2: Merkle commitment scheme (MT)



The BCS protocol is widely-used in practice.

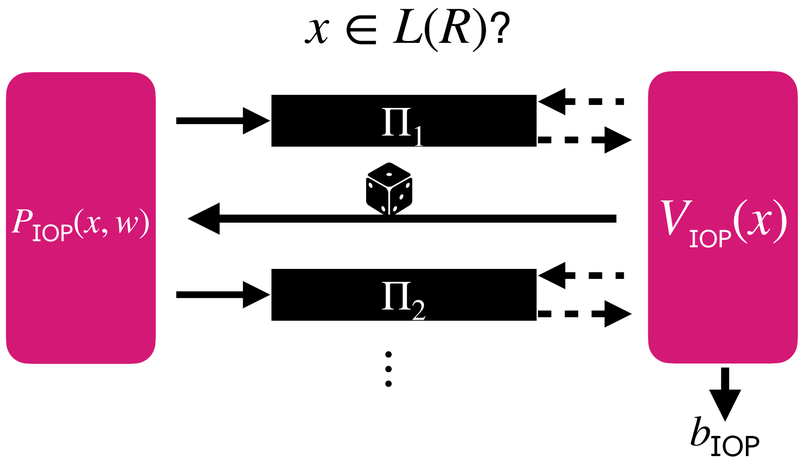
Security is analyzed in an ideal model: **random oracle model**.

Security holds even against **quantum** attackers:

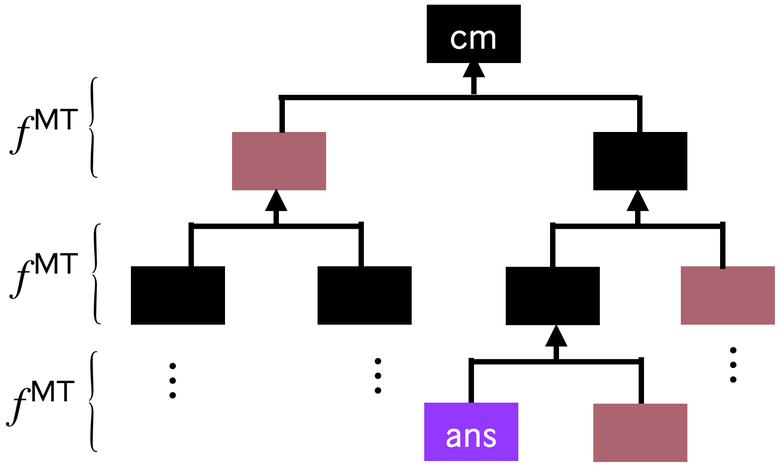


Recall: SNARG BCS[IOP, MT]

Ingredient #1: Interactive oracle proof (IOP)



Ingredient #2: Merkle commitment scheme (MT)

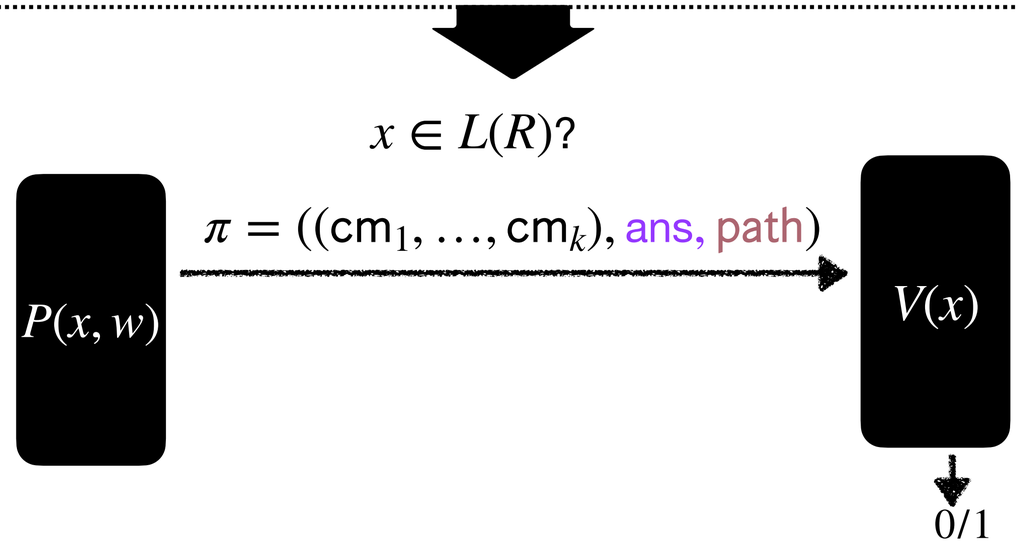


The BCS protocol is widely-used in practice.

Security is analyzed in an ideal model: **random oracle model**.

Security holds even against **quantum** attackers:

[CMS19]:
the BCS protocol is secure in the
quantum random oracle model

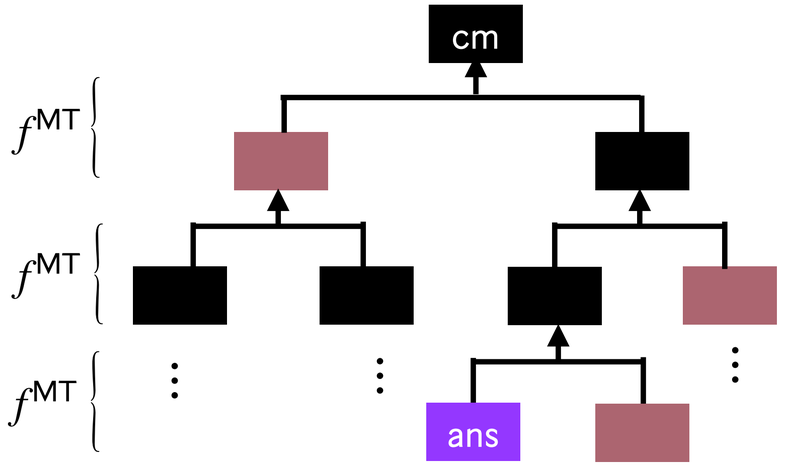


[BMNW25]: SNRDX BCS[IOR, MT]

[BMNW25]: SNRDX BCS[IOR, MT]

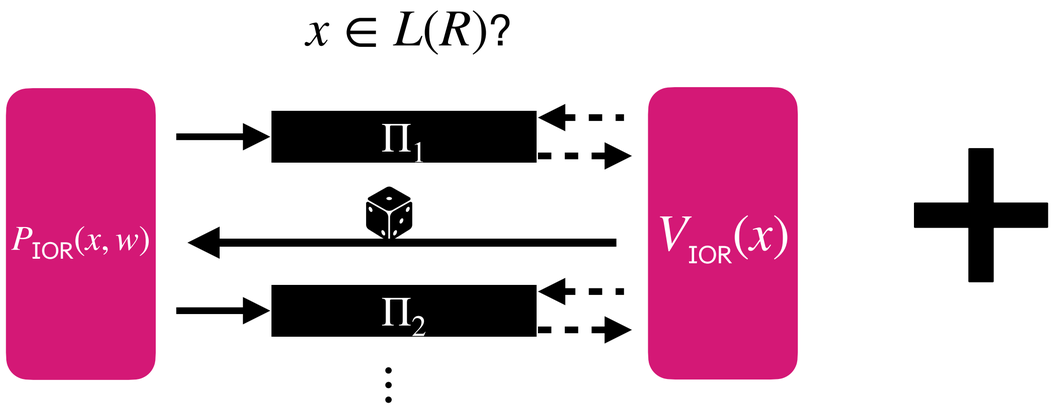
+

Ingredient #2: Merkle commitment scheme (MT)

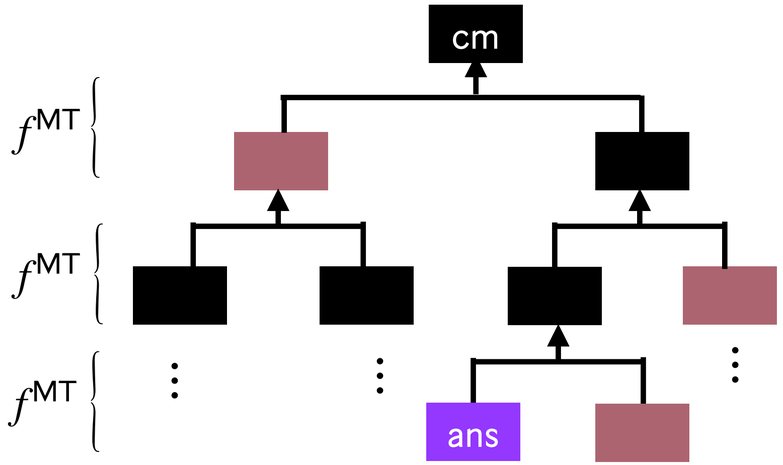


[BMNW25]: SNRDX BCS[IOR, MT]

Ingredient #1: Interactive oracle reduction (IOR)

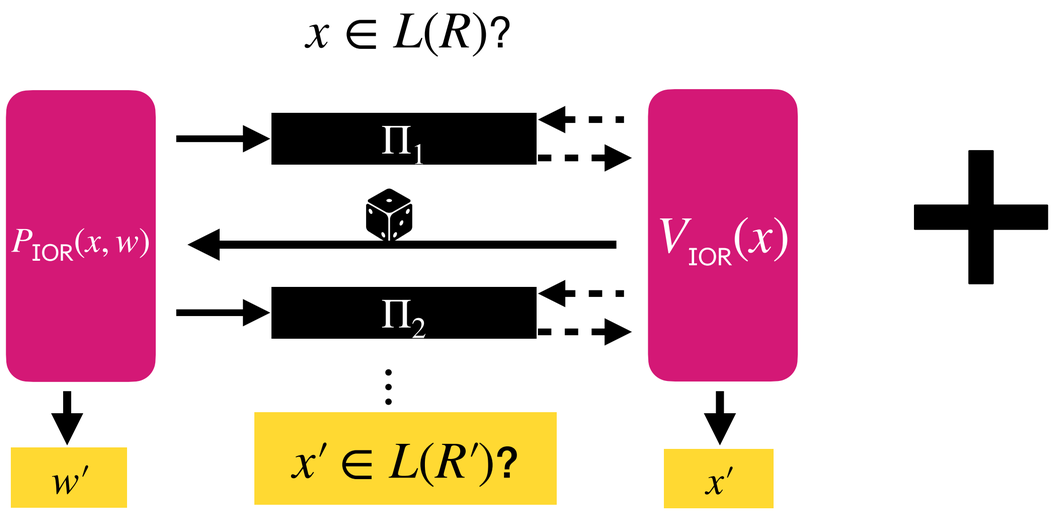


Ingredient #2: Merkle commitment scheme (MT)

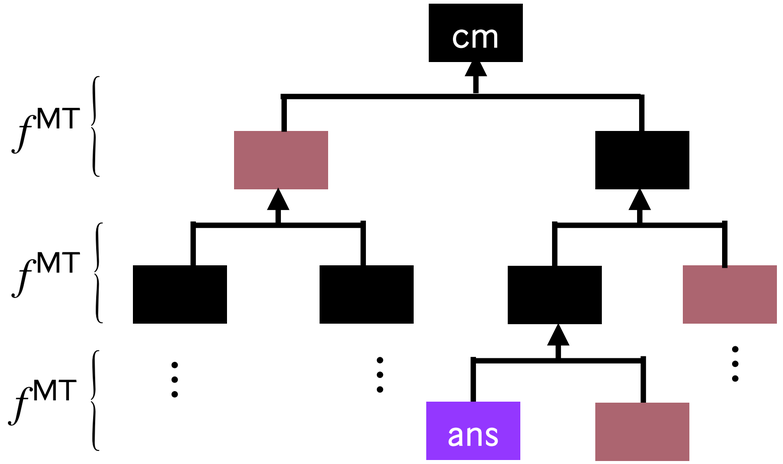


[BMNW25]: SNRDX BCS[IOR, MT]

Ingredient #1: Interactive oracle reduction (IOR)

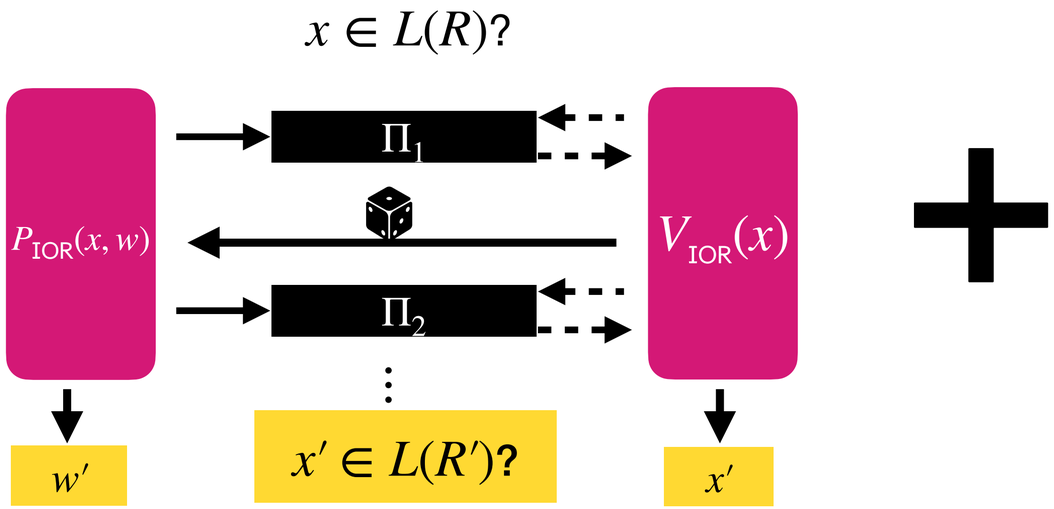


Ingredient #2: Merkle commitment scheme (MT)

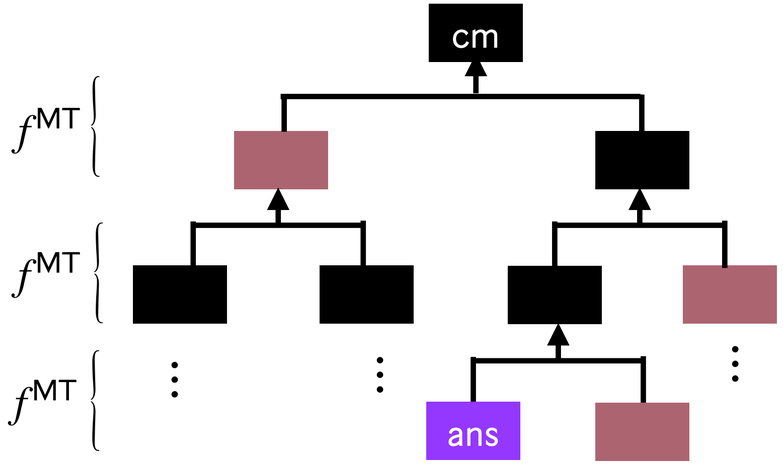


[BMNW25]: SNRDX BCS[IOR, MT]

Ingredient #1: Interactive oracle reduction (IOR)

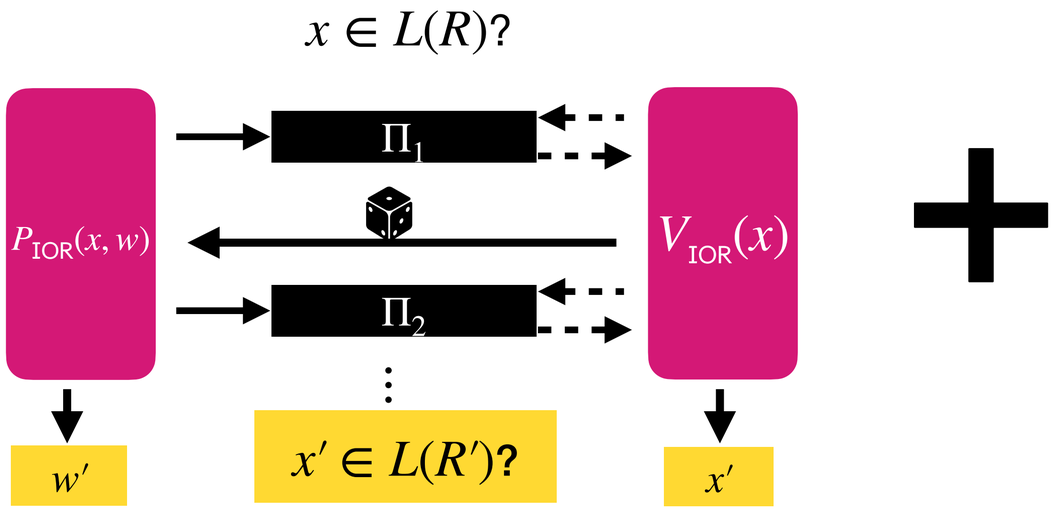


Ingredient #2: Merkle commitment scheme (MT)

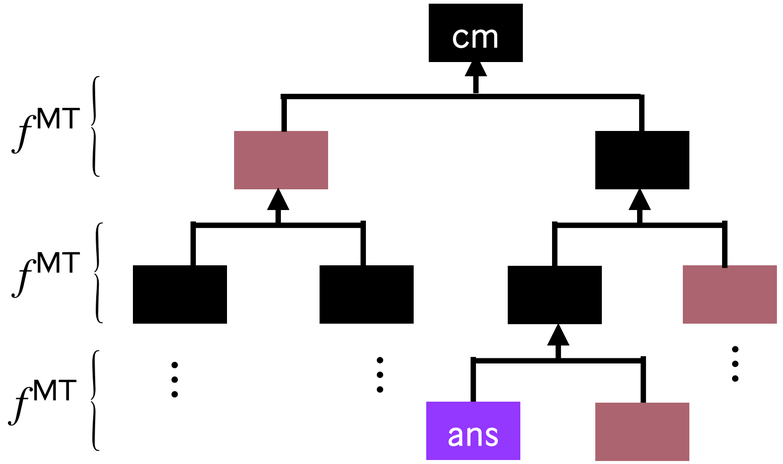


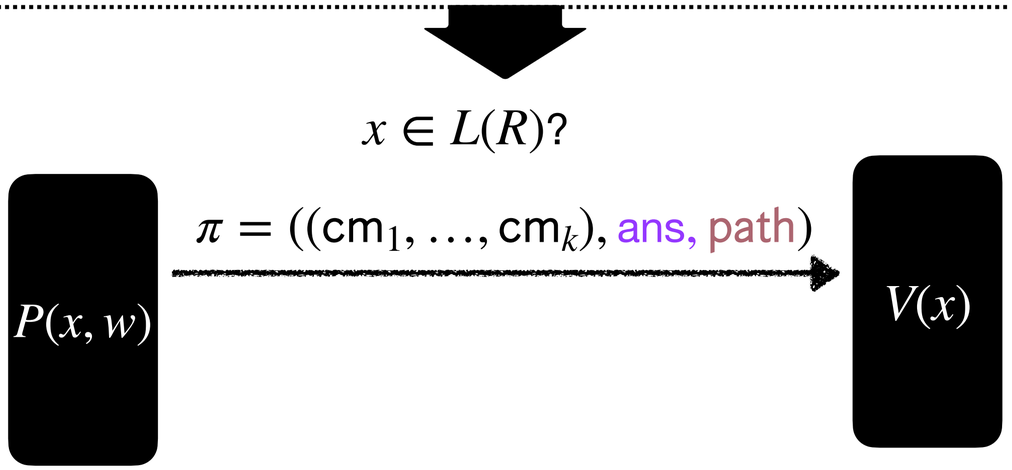
[BMNW25]: SNRDX BCS[IOR, MT]

Ingredient #1: Interactive oracle reduction (IOR)



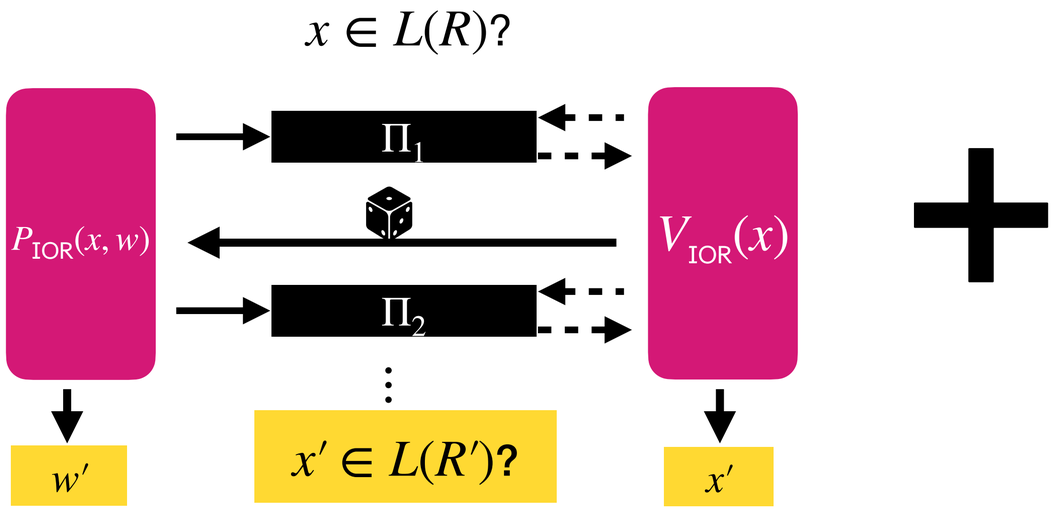
Ingredient #2: Merkle commitment scheme (MT)



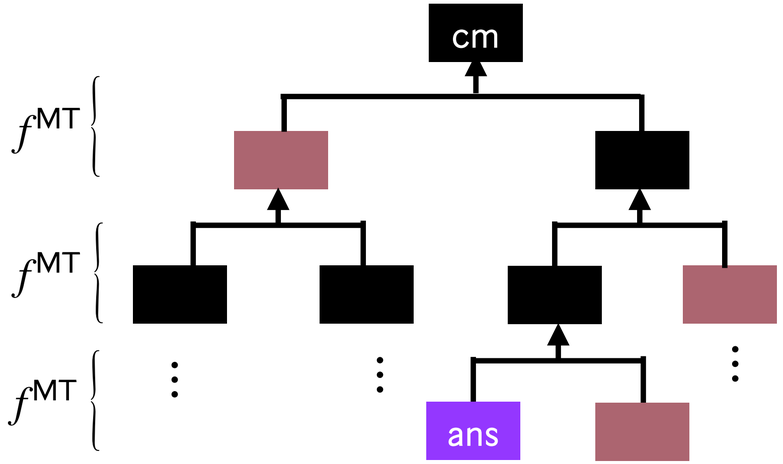


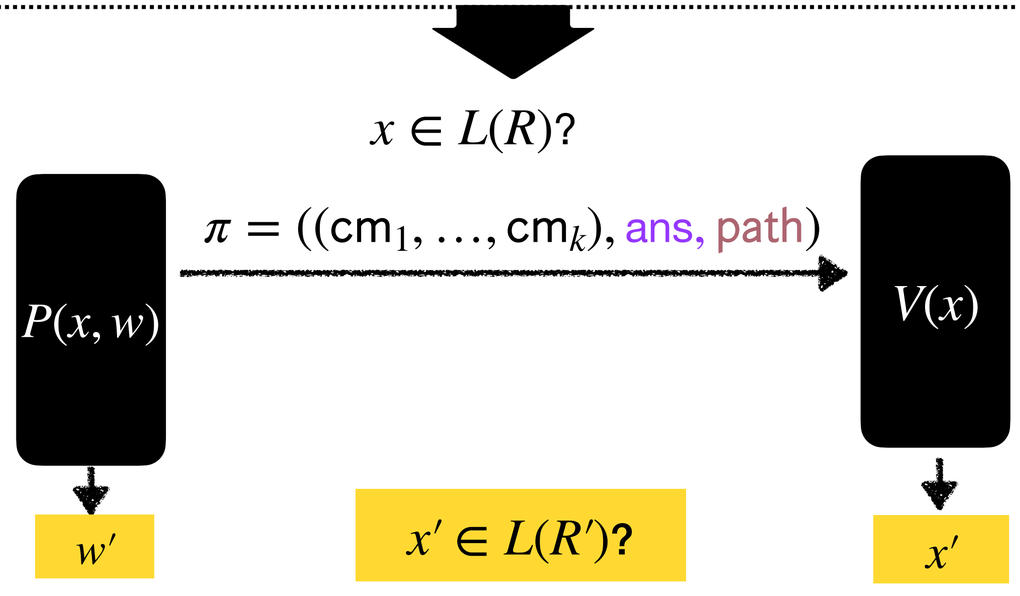
[BMNW25]: SNRDX BCS[IOR, MT]

Ingredient #1: Interactive oracle reduction (IOR)



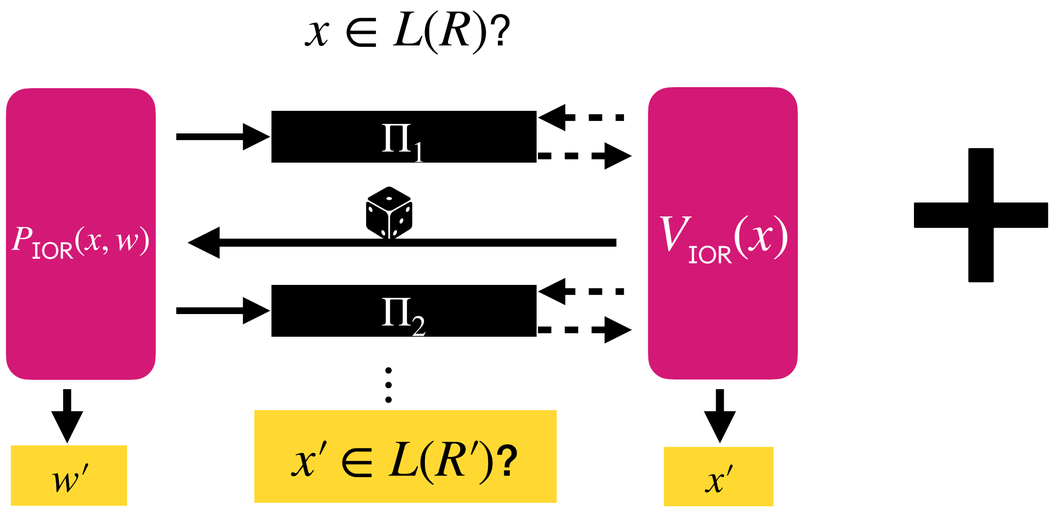
Ingredient #2: Merkle commitment scheme (MT)



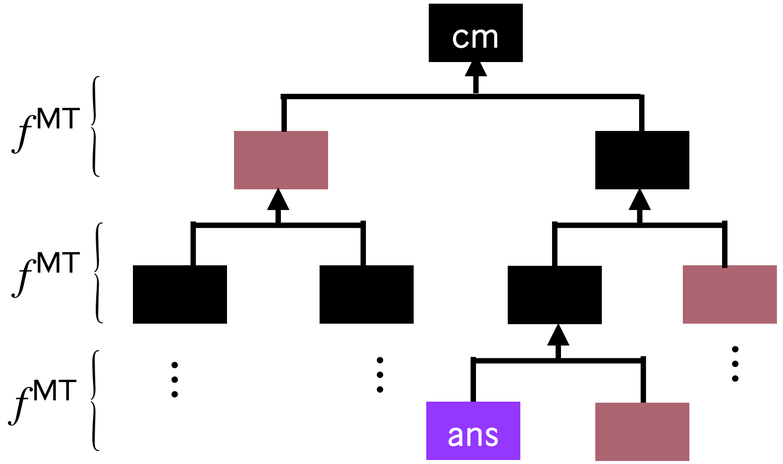


[BMNW25]: SNRDX BCS[IOR, MT]

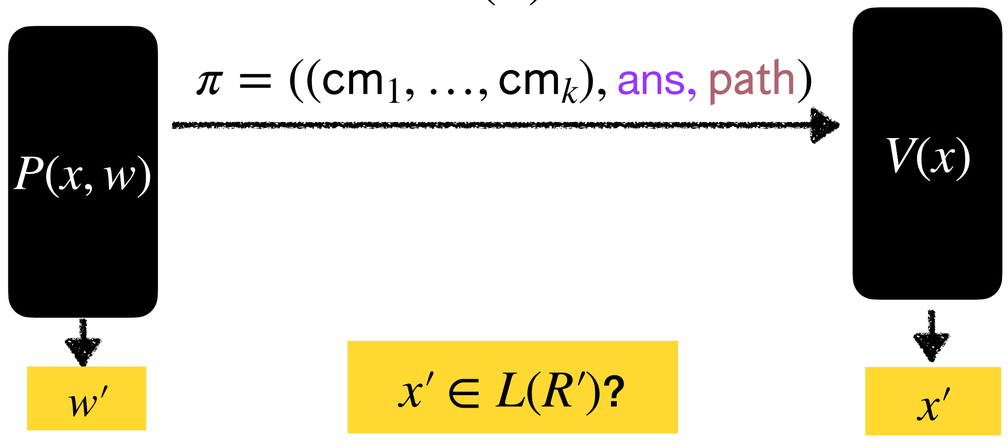
Ingredient #1: Interactive oracle reduction (IOR)



Ingredient #2: Merkle commitment scheme (MT)

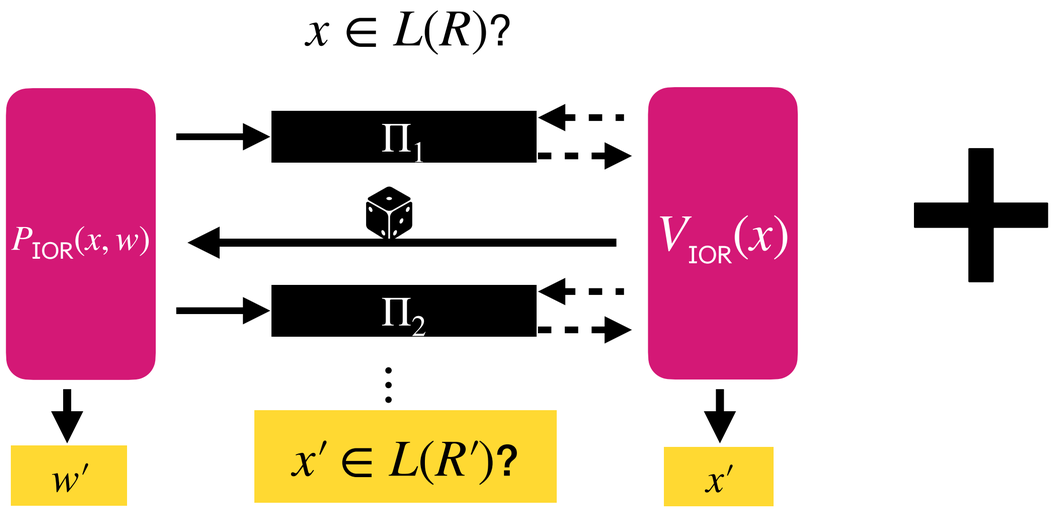


Simple and efficient hash-based SNRDXs [BMNW25; BCFW25].

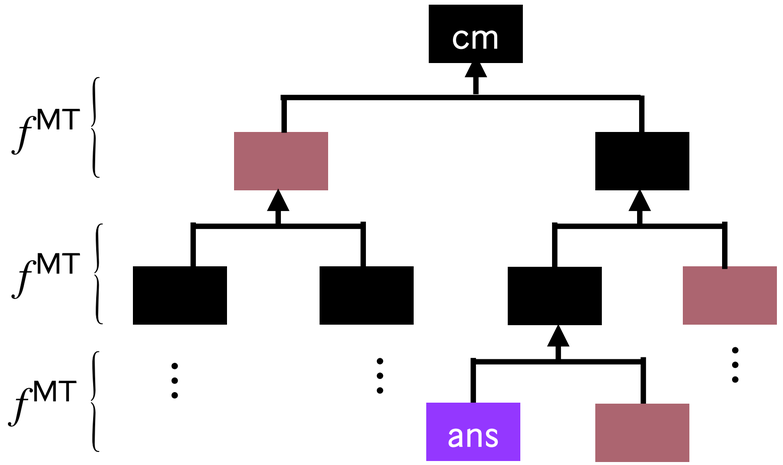


[BMNW25]: SNRDX BCS[IOR, MT]

Ingredient #1: Interactive oracle reduction (IOR)

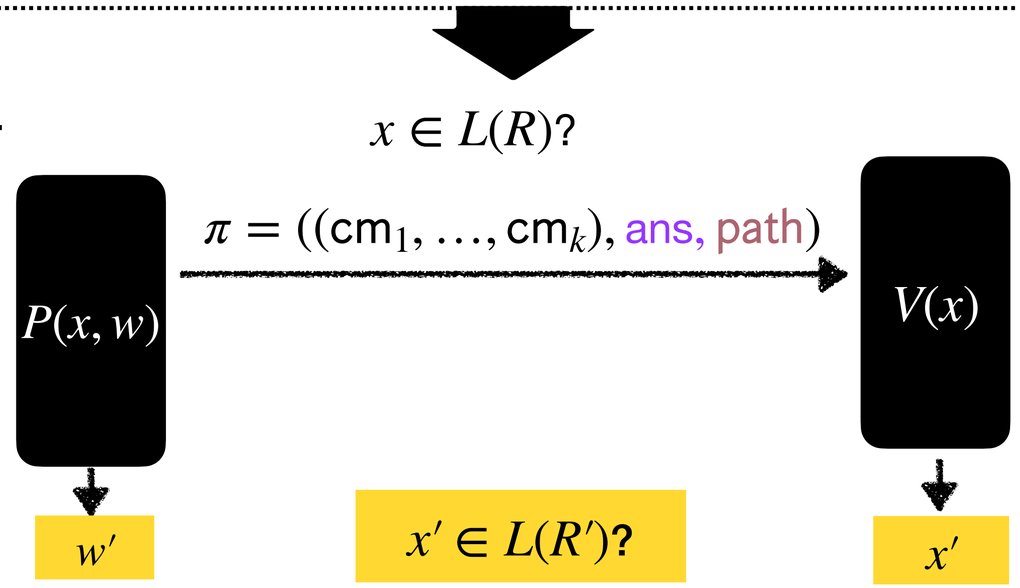


Ingredient #2: Merkle commitment scheme (MT)



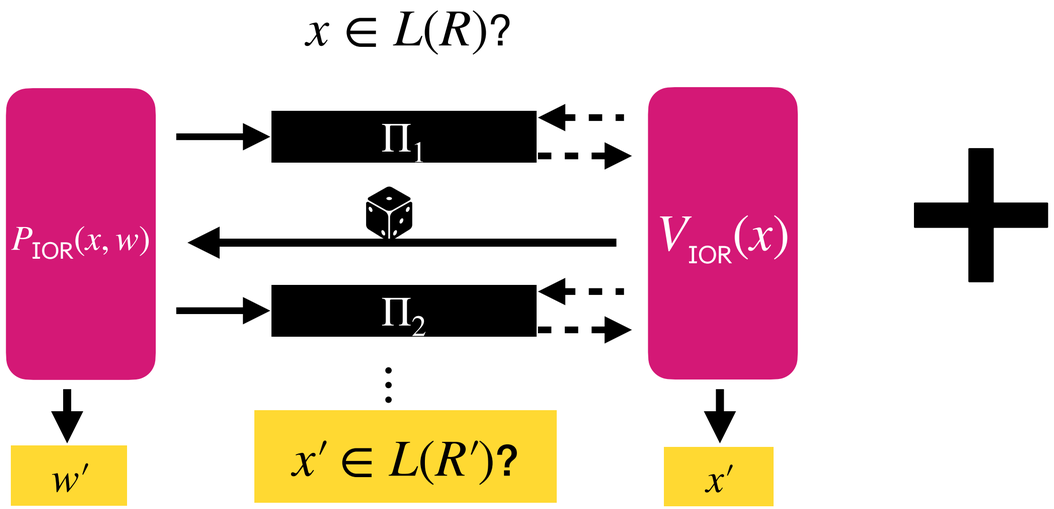
Simple and efficient hash-based SNRDXs [BMNW25; BCFW25].

Secure in the ROM against classical attackers [BMNW25].

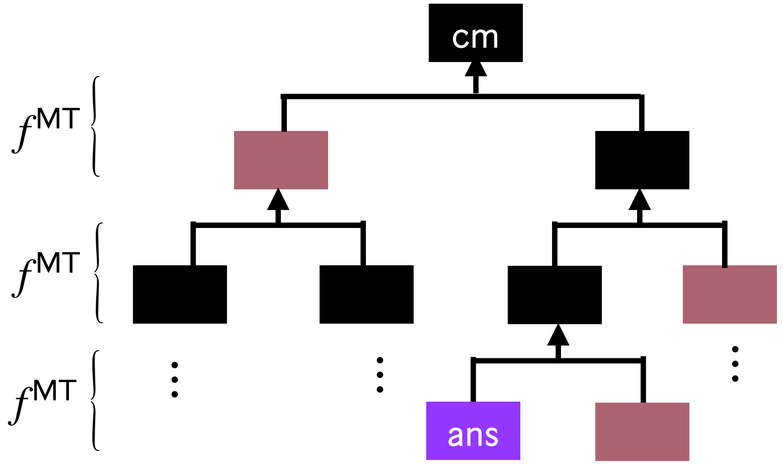


[BMNW25]: SNRDX BCS[IOR, MT]

Ingredient #1: Interactive oracle reduction (IOR)



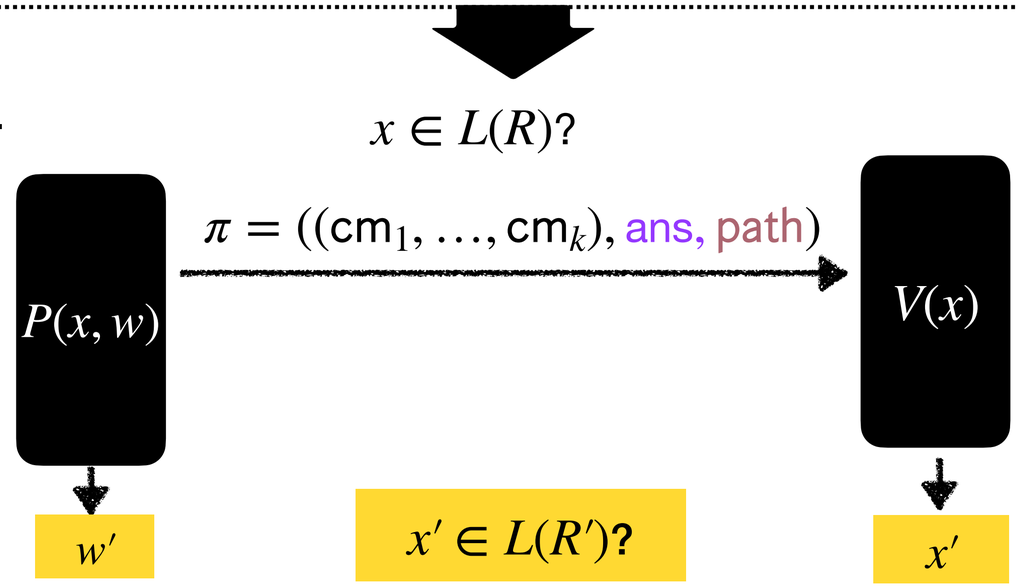
Ingredient #2: Merkle commitment scheme (MT)



Simple and efficient hash-based SNRDXs [BMNW25; BCFW25].

Secure in the ROM against classical attackers [BMNW25].

OUR QUESTION:
Are these hash-based SNRDXs secure in the QROM?



Why post-quantum security matters for hash-based SNRDXs?

Why post-quantum security matters for hash-based SNRDXs?

Hash-based SNRDXs
(packaged as hash-based accumulation/folding schemes),

Why post-quantum security matters for hash-based SNRDXs?

Hash-based SNRDXs
(packaged as hash-based accumulation/folding schemes),

are likely to be an important building block
for post-quantum redesigns of Ethereum.

Why not use [CMS19]?

Why not use [CMS19]?

We cannot. Also, we should not.

Problem 1: requires the IOP to satisfy **RBR knowledge soundness**

Why not use [CMS19]?

We cannot. Also, we should not.

Problem 1: requires the IOP to satisfy **RBR knowledge soundness**

State-of-the-art IOPs/IORs satisfy a **weaker** variant.

Why not use [CMS19]?

We cannot. Also, we should not.

Problem 1: requires the IOP to satisfy **RBR knowledge soundness**

State-of-the-art IOPs/IORs satisfy a **weaker** variant.

Problem 2: applies to BCS[IOP, MT], not BCS[IOR, MT]

Why not use [CMS19]?

We cannot. Also, we should not.

Problem 1: requires the IOP to satisfy **RBR knowledge soundness**

State-of-the-art IOPs/IORs satisfy a **weaker** variant.

Problem 2: applies to BCS[IOP, MT], not BCS[IOR, MT]

Classically, BCS[IOR, MT] and BCS[IOP, MT]
require **different** proofs.
Quantumly, even **larger gap**.

Why not use [CMS19]?

We cannot. Also, we should not.

Problem 1: requires the IOP to satisfy **RBR knowledge soundness**

State-of-the-art IOPs/IORs satisfy a **weaker** variant.

Problem 2: applies to BCS[IOP, MT], not BCS[IOR, MT]

Classically, BCS[IOR, MT] and BCS[IOP, MT]
require **different** proofs.
Quantumly, even **larger gap**.

Why not use [CMS19]?

We cannot. Also, we should not.

Problem 3: proves **non-adaptive** security of BCS[IOP, MT]

Problem 1: requires the IOP to satisfy **RBR knowledge soundness**

State-of-the-art IOPs/IORs satisfy a **weaker** variant.

Problem 2: applies to BCS[IOP, MT], not BCS[IOR, MT]

Classically, BCS[IOR, MT] and BCS[IOP, MT]
require **different** proofs.
Quantumly, even **larger gap**.

Why not use [CMS19]?

We cannot. Also, we should not.

Problem 3: proves **non-adaptive** security of BCS[IOP, MT]

We target **adaptive** security of BCS[IOR, MT].

Problem 1: requires the IOP to satisfy **RBR knowledge soundness**

State-of-the-art IOPs/IORs satisfy a **weaker** variant.

Problem 2: applies to BCS[IOP, MT], not BCS[IOR, MT]

Classically, BCS[IOR, MT] and BCS[IOP, MT]
require **different** proofs.
Quantumly, even **larger gap**.

Why not use [CMS19]?

We cannot. Also, we should not.

Problem 3: proves **non-adaptive** security of BCS[IOP, MT]

We target **adaptive** security of BCS[IOR, MT].

Problem 4: adopts a **"monolithic"** proof approach

Problem 1: requires the IOP to satisfy **RBR knowledge soundness**

State-of-the-art IOPs/IORs satisfy a **weaker** variant.

Problem 2: applies to BCS[IOP, MT], not BCS[IOR, MT]

Classically, BCS[IOR, MT] and BCS[IOP, MT]
require **different** proofs.
Quantumly, even **larger gap**.

Why not use [CMS19]?

We cannot. Also, we should not.

Problem 3: proves **non-adaptive** security of BCS[IOP, MT]

We target **adaptive** security of BCS[IOR, MT].

Problem 4: adopts a **"monolithic"** proof approach

We want a quantum proof of BCS[IOR, MT]
that aligns with the classical one (we want **the "right" one**!).

Problem 1: requires the IOP to satisfy **RBR knowledge soundness**

State-of-the-art IOPs/IORs satisfy a **weaker** variant.

Problem 2: applies to BCS[IOP, MT], not BCS[IOR, MT]

Classically, BCS[IOR, MT] and BCS[IOP, MT]
require **different** proofs.
Quantumly, even **larger gap**.

Why not use [CMS

We cannot. Also, we shou

Back to the drawing board!

Problem 3: proves **non-adaptive** security of BCS[IOP, MT]

We target **adaptive** security of BCS[IOR, MT].

Problem 4: adopts a **"monolithic"** proof approach

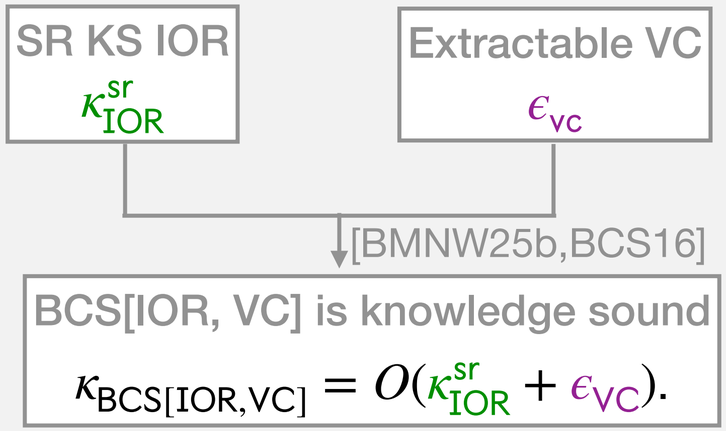
We want a quantum proof of BCS[IOR, MT]
that aligns with the classical one (we want **the "right" one**!).

Our results

Theorem 1:

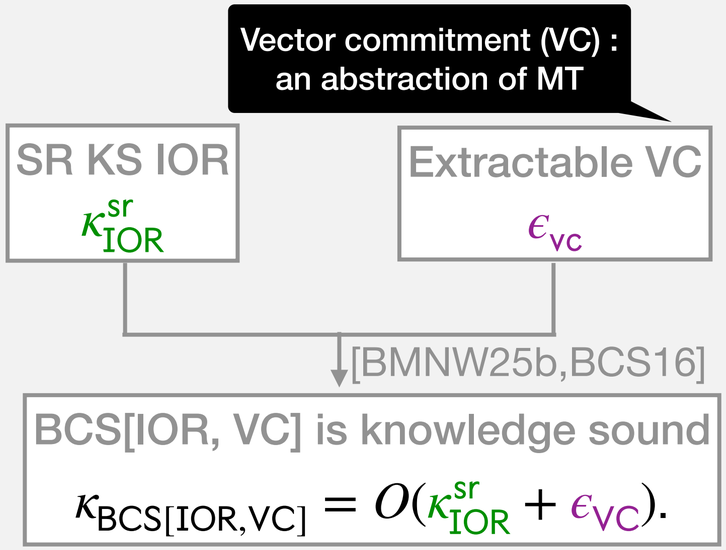
Theorem 1:

Classical case



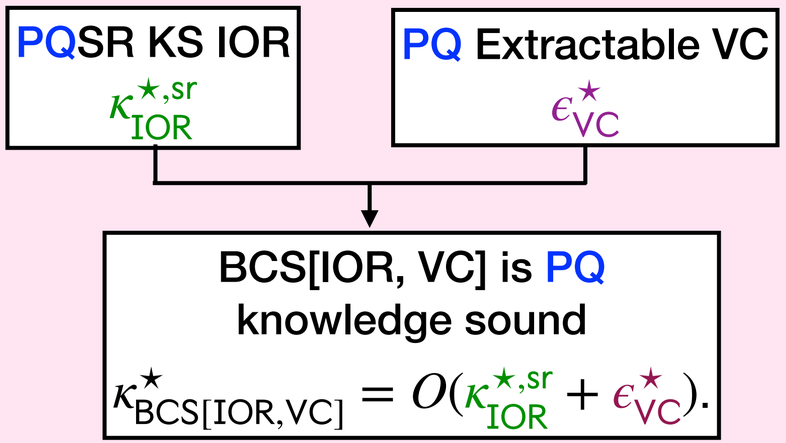
Theorem 1:

Classical case

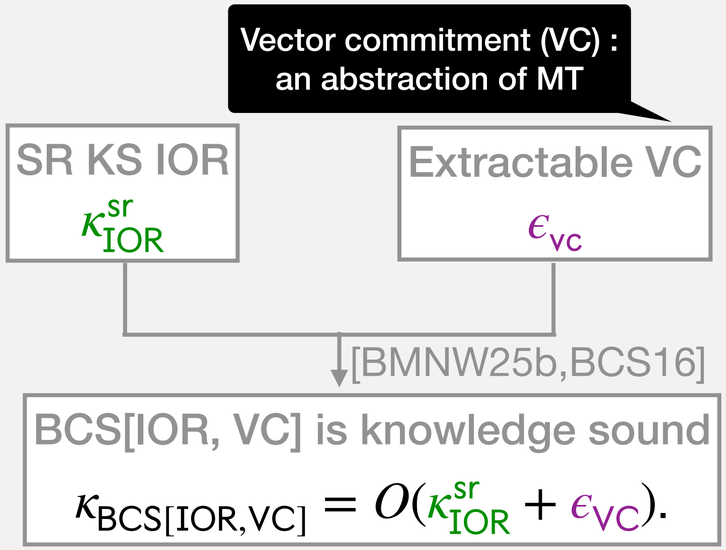


Theorem 1:

Post-quantum case



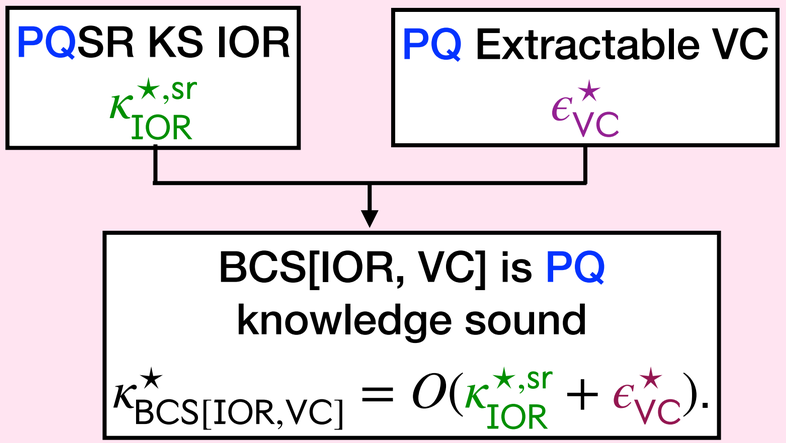
Classical case



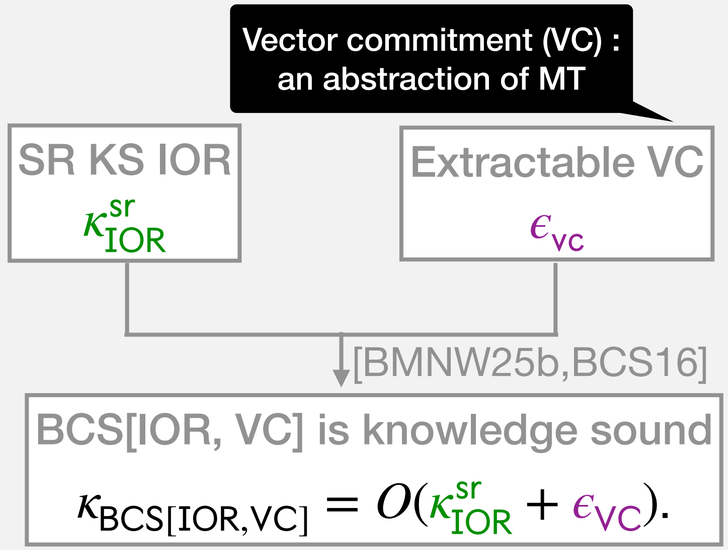
Theorem 2:

Theorem 1:

Post-quantum case



Classical case



Theorem 2:

Theorem 1:

Post-quantum case

PQSR KS IOR

$\kappa_{\text{IOR}}^{\star,\text{sr}}$

PQ Extractable VC

$\epsilon_{\text{VC}}^{\star}$

BCS[IOR, VC] is PQ knowledge sound

$\kappa_{\text{BCS[IOR,VC]}}^{\star} = O(\kappa_{\text{IOR}}^{\star,\text{sr}} + \epsilon_{\text{VC}}^{\star}).$

Classical case

RBR KS IOR

relaxed RBR KS IOR

$\kappa_{\text{IOR}}^{\text{rrbr}}$

[BCFW25]

SR KS IOR

$\kappa_{\text{IOR}}^{\text{sr}}(t) = O((t+k) \cdot \kappa_{\text{IOR}}^{\text{rrbr}})$

Vector commitment (VC) : an abstraction of MT

SR KS IOR

$\kappa_{\text{IOR}}^{\text{sr}}$

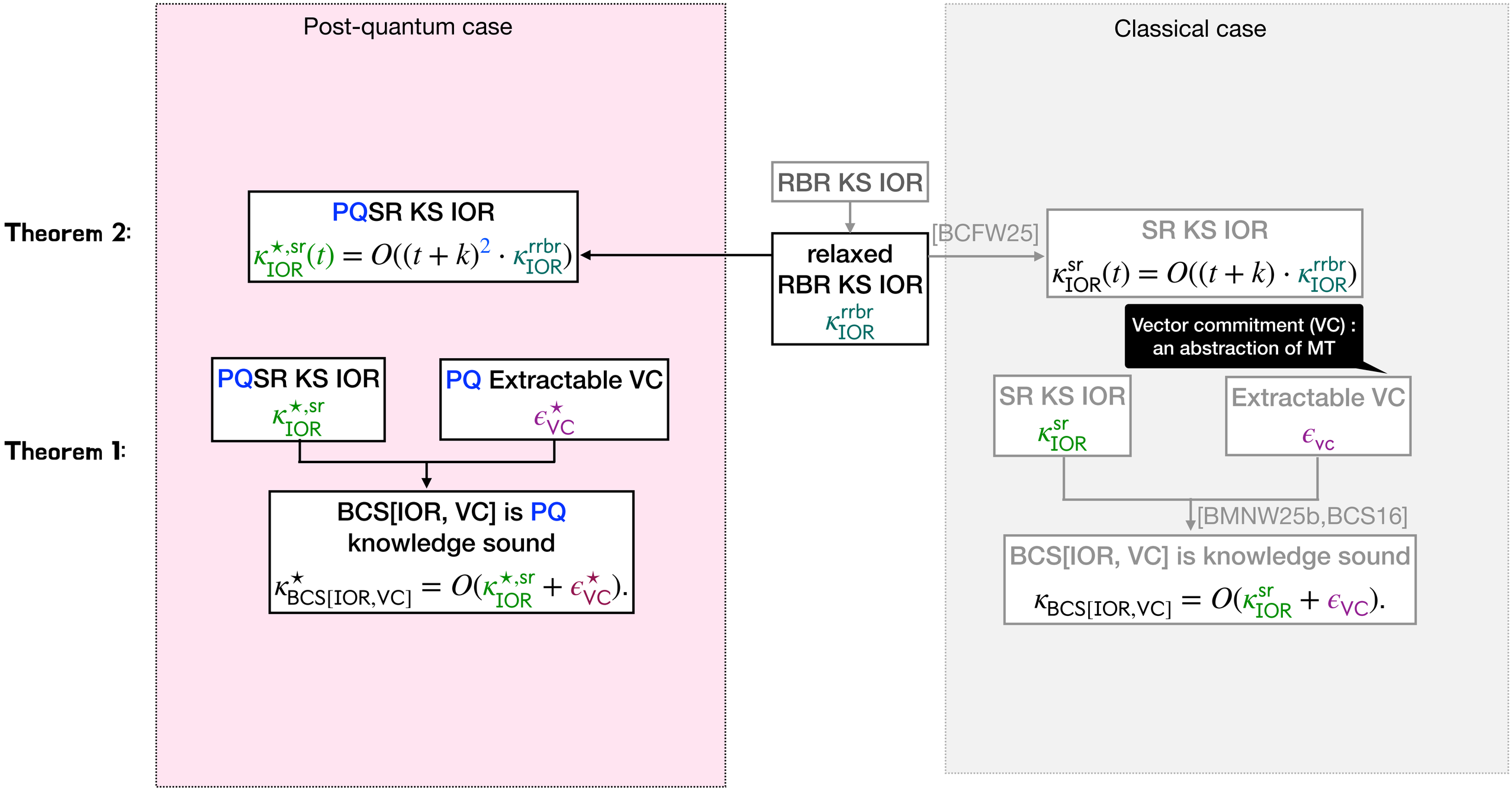
Extractable VC

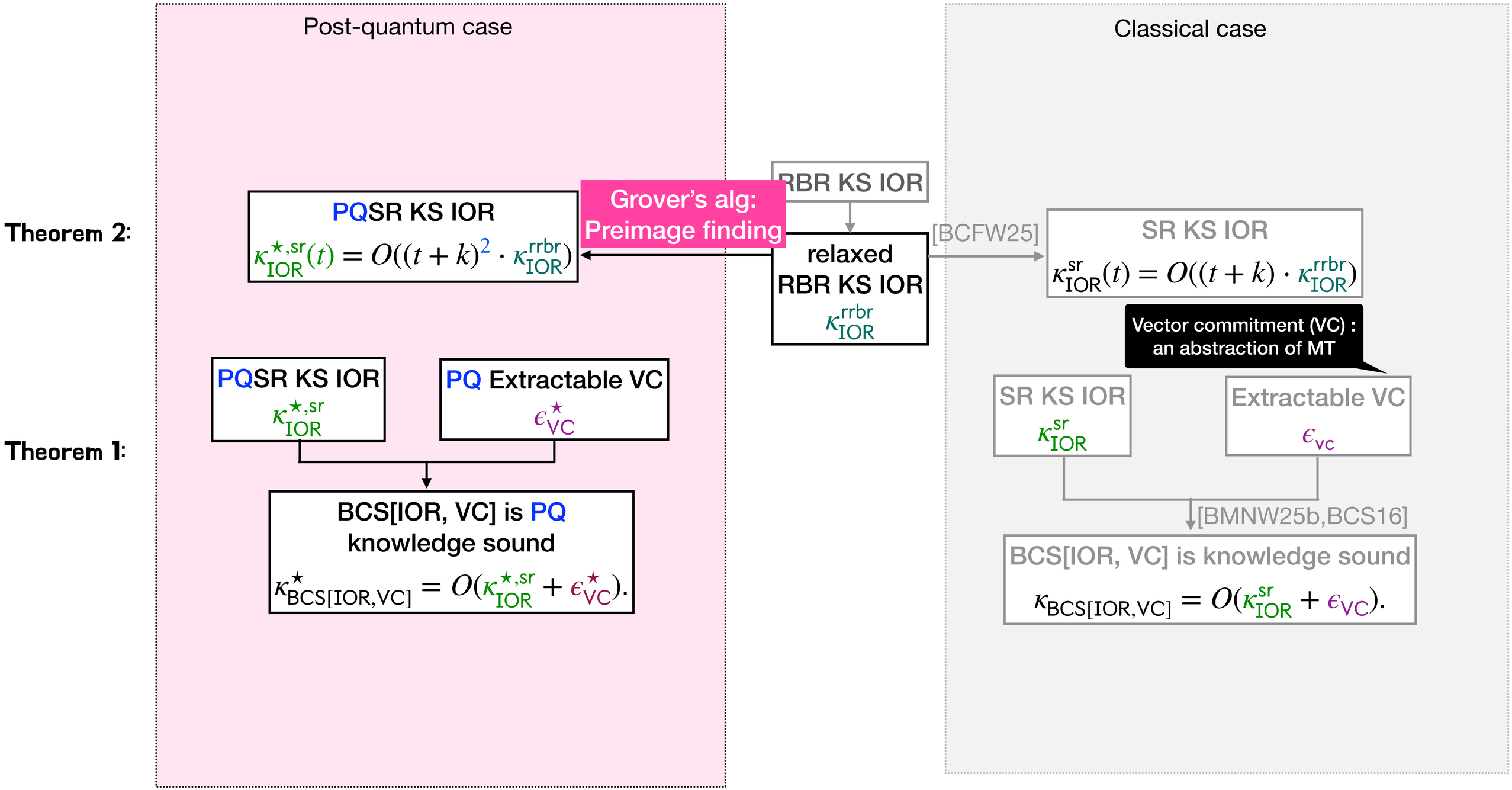
ϵ_{vc}

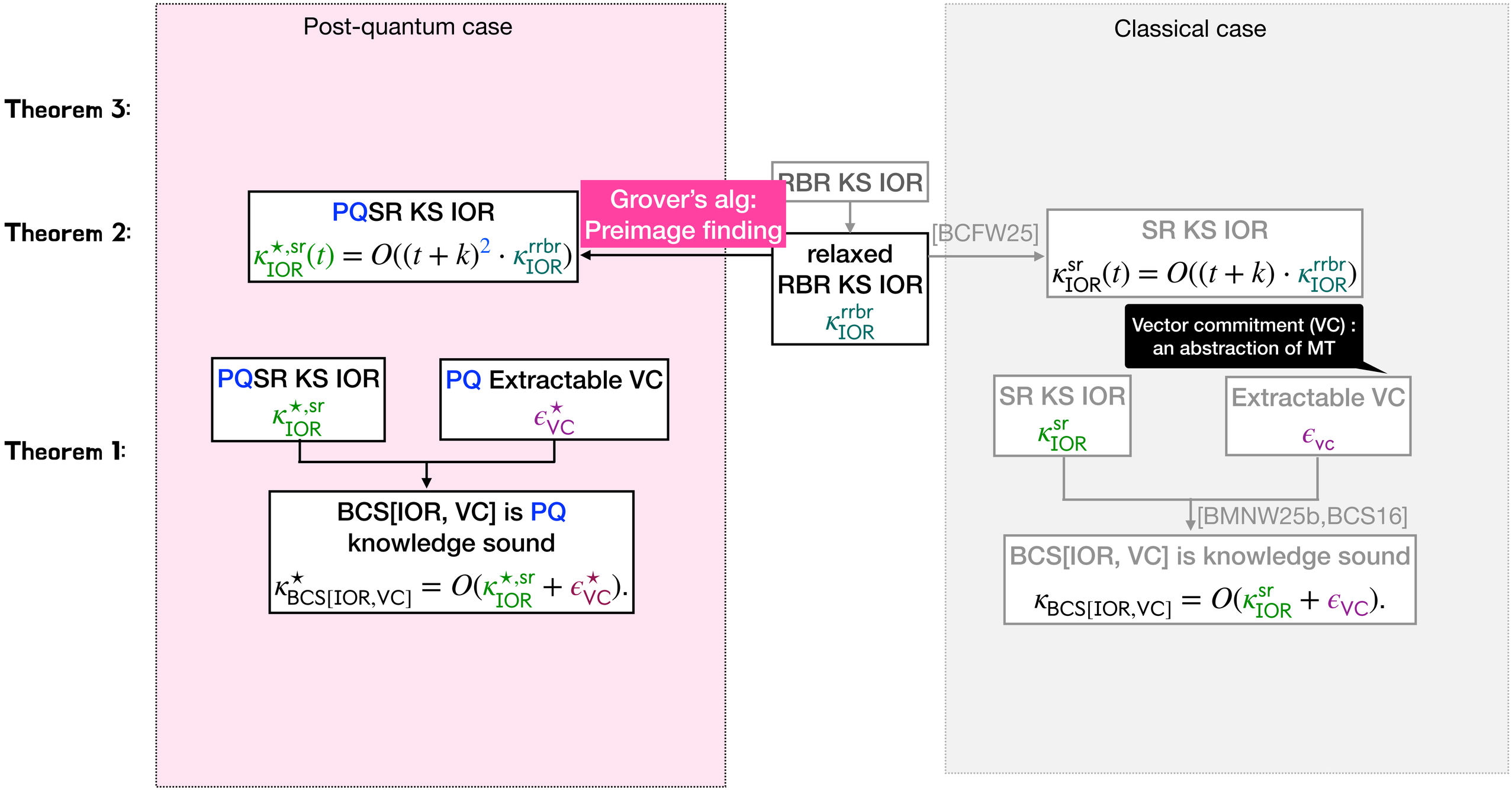
[BMNW25b,BCS16]

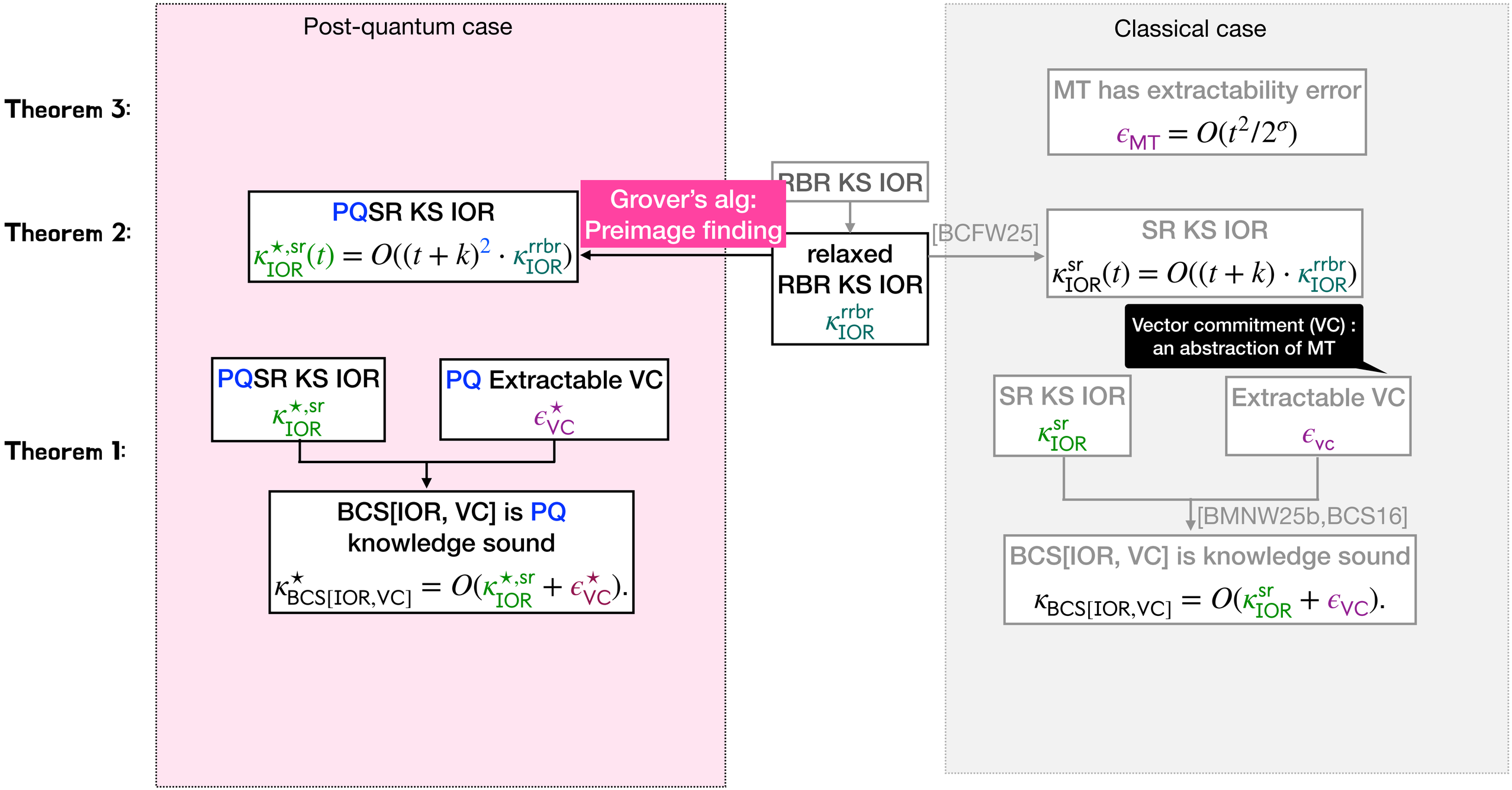
BCS[IOR, VC] is knowledge sound

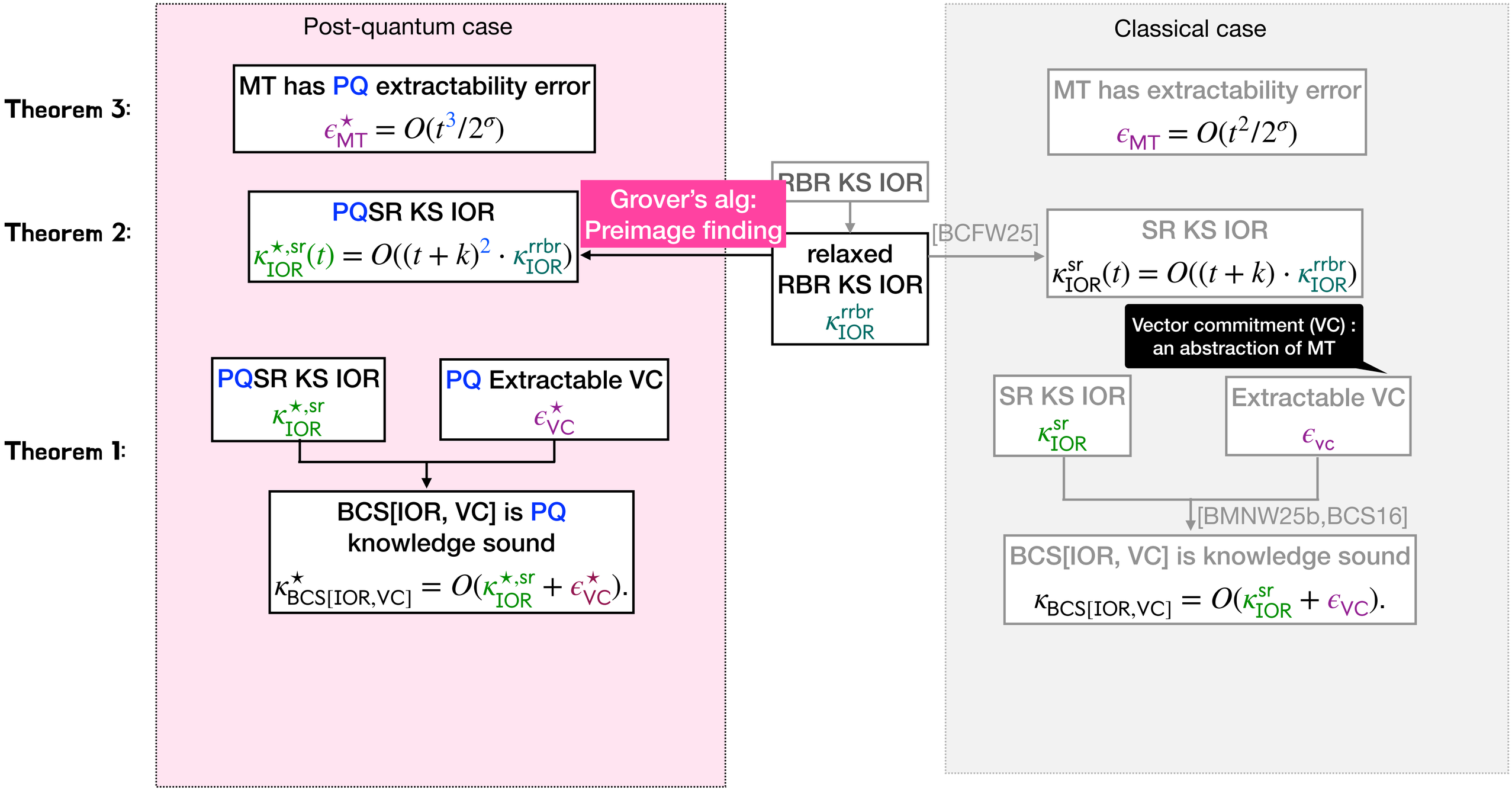
$\kappa_{\text{BCS[IOR,VC]}} = O(\kappa_{\text{IOR}}^{\text{sr}} + \epsilon_{\text{VC}}).$

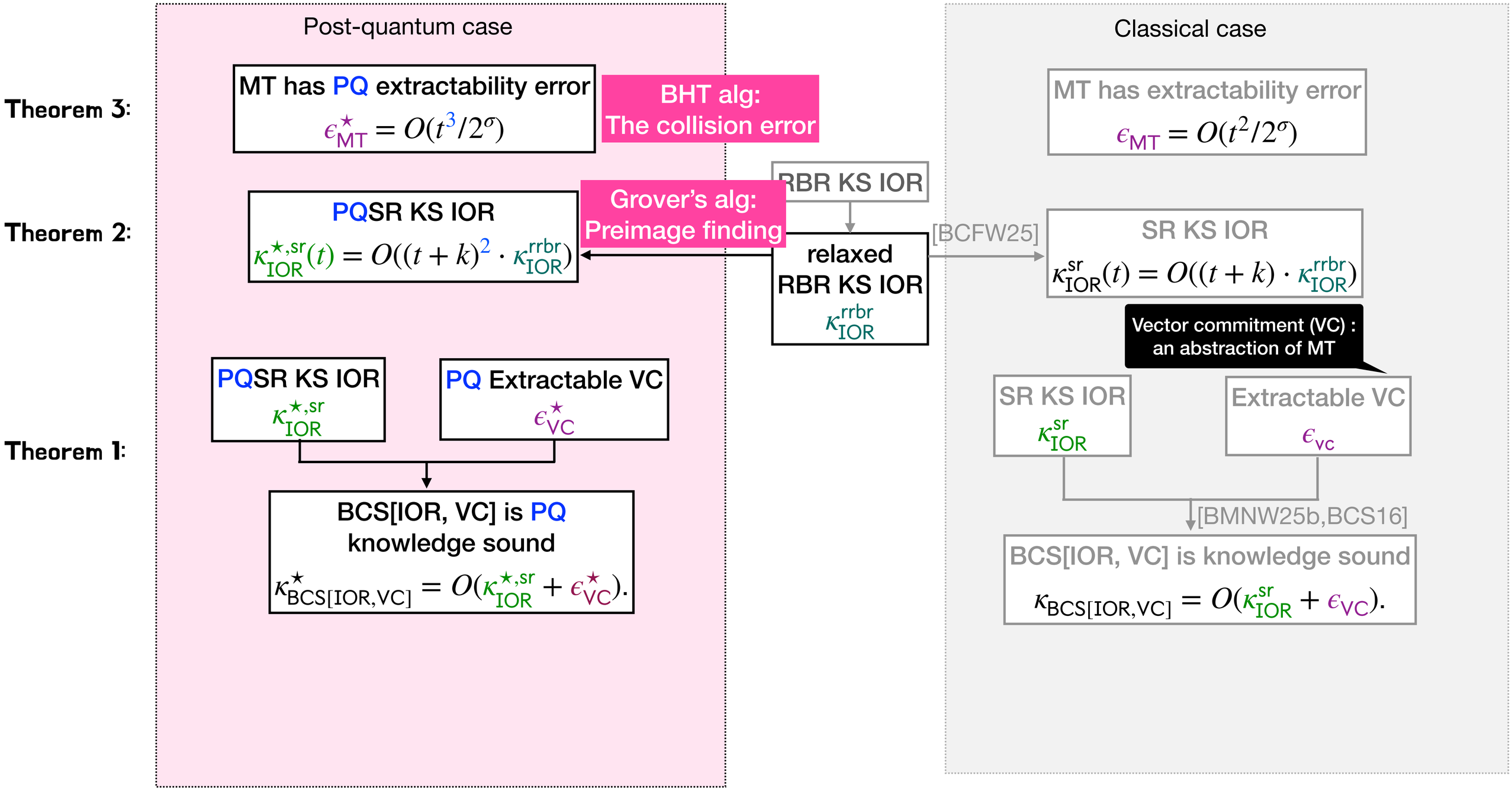

Post-quantum case
Classical case
Theorem 2:
Theorem 1:
PQSR KS IOR
$\kappa^{\star,\mathrm{sr}}_{\mathrm{IOR}}(t) = O((t+k)^2 \cdot \kappa^{\mathrm{rrbr}}_{\mathrm{IOR}})$
RBR KS IOR
relaxed RBR KS IOR
$\kappa^{\mathrm{rrbr}}_{\mathrm{IOR}}$
[BCFW25]
SR KS IOR
$\kappa^{\mathrm{sr}}_{\mathrm{IOR}}(t) = O((t+k) \cdot \kappa^{\mathrm{rrbr}}_{\mathrm{IOR}})$
Vector commitment (VC) : an abstraction of MT
PQSR KS IOR
$\kappa^{\star,\mathrm{sr}}_{\mathrm{IOR}}$
PQ Extractable VC
$\epsilon^{\star}_{\mathrm{VC}}$
SR KS IOR
$\kappa^{\mathrm{sr}}_{\mathrm{IOR}}$
Extractable VC
ϵ_{vc}
[BMNW25b,BCS16]
BCS[IOR, VC] is PQ knowledge sound
$\kappa^{\star}_{\mathrm{BCS[IOR,VC]}} = O(\kappa^{\star,\mathrm{sr}}_{\mathrm{IOR}} + \epsilon^{\star}_{\mathrm{VC}})$.
BCS[IOR, VC] is knowledge sound
$\kappa_{\mathrm{BCS[IOR,VC]}} = O(\kappa^{\mathrm{sr}}_{\mathrm{IOR}} + \epsilon_{\mathrm{VC}})$.

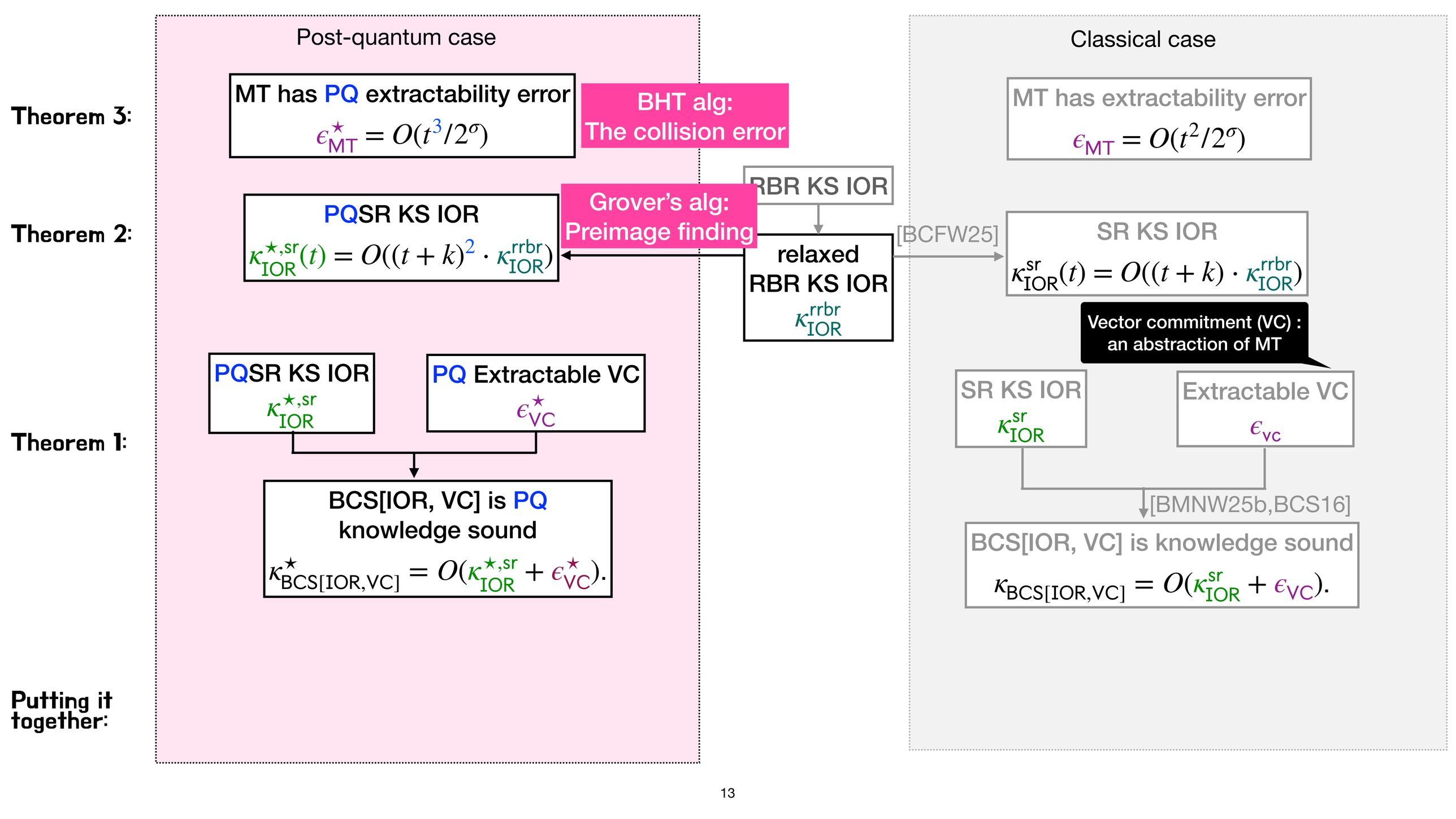

Post-quantum case
Classical case
Theorem 2:
Theorem 1:
PQSR KS IOR
$\kappa^{\star,\mathrm{sr}}_{\mathrm{IOR}}(t) = O((t+k)^2 \cdot \kappa^{\mathrm{rrbr}}_{\mathrm{IOR}})$
Grover's alg:
Preimage finding
RBR KS IOR
relaxed
RBR KS IOR
$\kappa^{\mathrm{rrbr}}_{\mathrm{IOR}}$
[BCFW25]
SR KS IOR
$\kappa^{\mathrm{sr}}_{\mathrm{IOR}}(t) = O((t+k) \cdot \kappa^{\mathrm{rrbr}}_{\mathrm{IOR}})$
Vector commitment (VC) :
an abstraction of MT
PQSR KS IOR
$\kappa^{\star,\mathrm{sr}}_{\mathrm{IOR}}$
PQ Extractable VC
$\epsilon^{\star}_{\mathrm{VC}}$
SR KS IOR
$\kappa^{\mathrm{sr}}_{\mathrm{IOR}}$
Extractable VC
ϵ_{vc}
[BMNW25b,BCS16]
BCS[IOR, VC] is PQ
knowledge sound
$\kappa^{\star}_{\mathrm{BCS[IOR,VC]}} = O(\kappa^{\star,\mathrm{sr}}_{\mathrm{IOR}} + \epsilon^{\star}_{\mathrm{VC}})$.
BCS[IOR, VC] is knowledge sound
$\kappa_{\mathrm{BCS[IOR,VC]}} = O(\kappa^{\mathrm{sr}}_{\mathrm{IOR}} + \epsilon_{\mathrm{VC}})$.

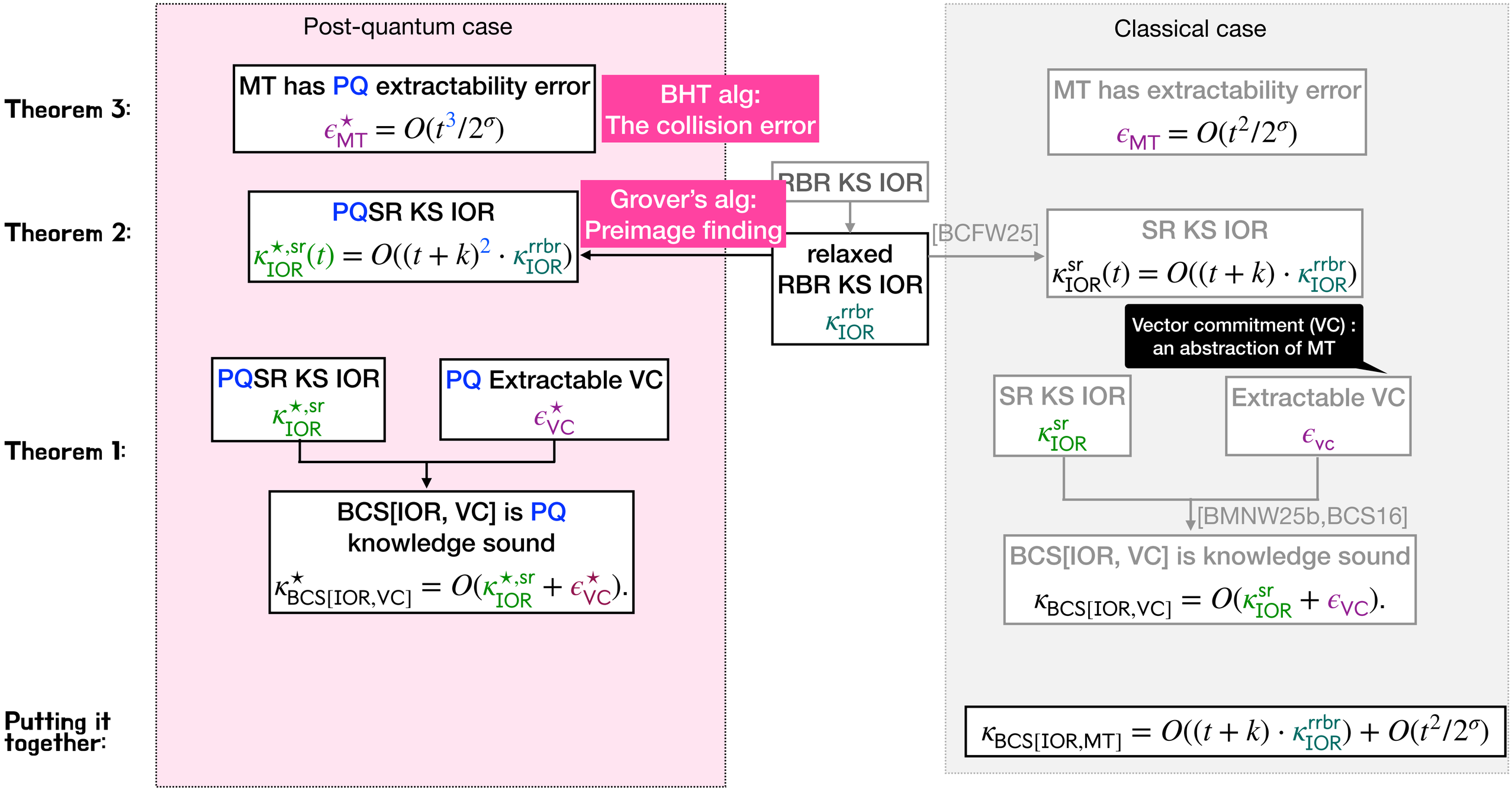

Post-quantum case
Classical case
Theorem 3:
Theorem 2:
Theorem 1:
PQSR KS IOR
$\kappa^{\star,\mathrm{sr}}_{\mathrm{IOR}}(t) = O((t+k)^2 \cdot \kappa^{\mathrm{rrbr}}_{\mathrm{IOR}})$
Grover's alg:
Preimage finding
RBR KS IOR
relaxed
RBR KS IOR
$\kappa^{\mathrm{rrbr}}_{\mathrm{IOR}}$
[BCFW25]
SR KS IOR
$\kappa^{\mathrm{sr}}_{\mathrm{IOR}}(t) = O((t+k) \cdot \kappa^{\mathrm{rrbr}}_{\mathrm{IOR}})$
Vector commitment (VC) :
an abstraction of MT
PQSR KS IOR
$\kappa^{\star,\mathrm{sr}}_{\mathrm{IOR}}$
PQ Extractable VC
$\epsilon^{\star}_{\mathrm{VC}}$
SR KS IOR
$\kappa^{\mathrm{sr}}_{\mathrm{IOR}}$
Extractable VC
ϵ_{vc}
[BMNW25b,BCS16]
BCS[IOR, VC] is PQ
knowledge sound
$\kappa^{\star}_{\mathrm{BCS[IOR,VC]}} = O(\kappa^{\star,\mathrm{sr}}_{\mathrm{IOR}} + \epsilon^{\star}_{\mathrm{VC}})$.
BCS[IOR, VC] is knowledge sound
$\kappa_{\mathrm{BCS[IOR,VC]}} = O(\kappa^{\mathrm{sr}}_{\mathrm{IOR}} + \epsilon_{\mathrm{VC}})$.

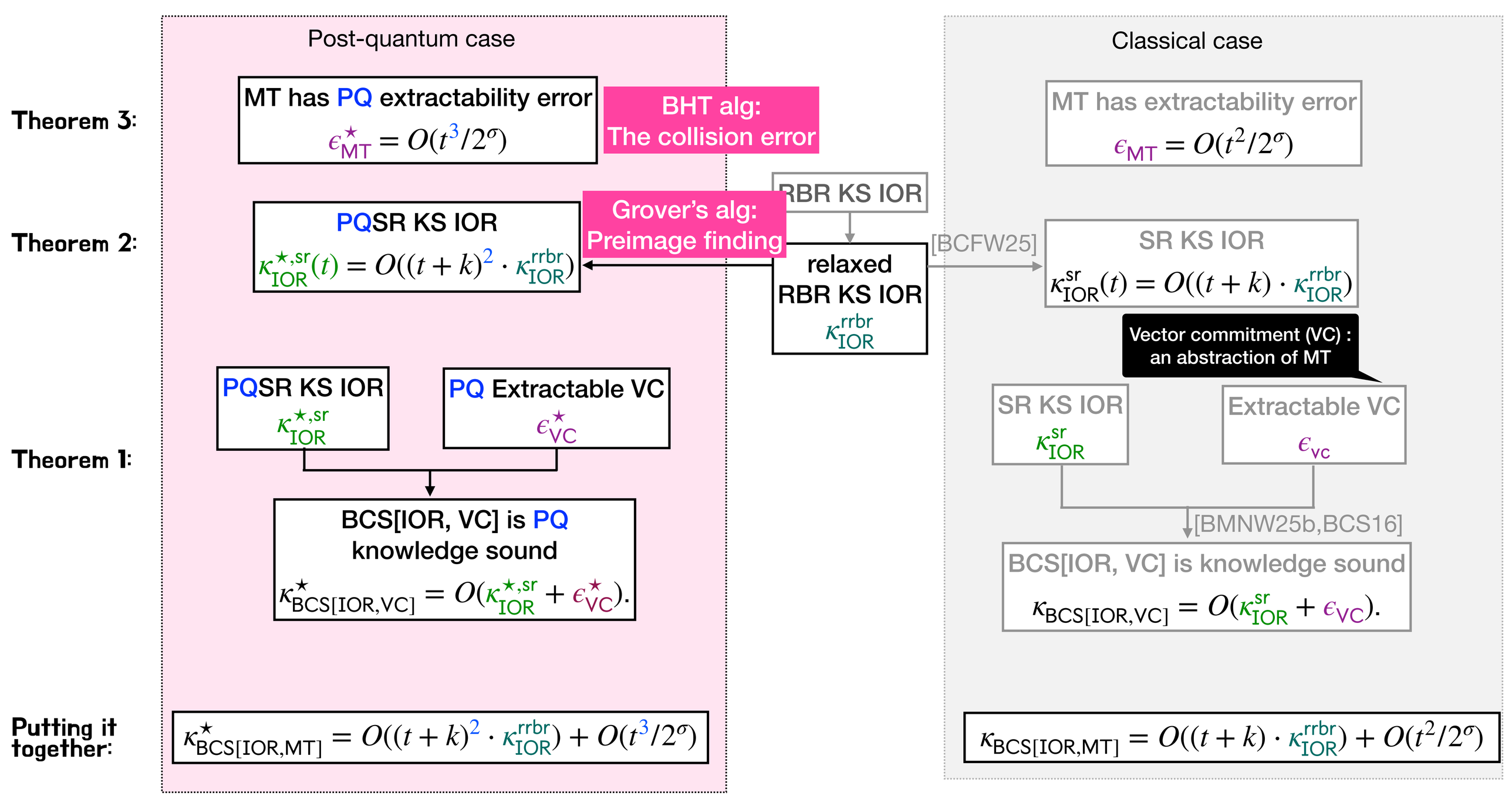

Post-quantum case
Classical case
Theorem 3:
Theorem 2:
Theorem 1:
MT has extractability error
$\epsilon_{\mathsf{MT}} = O(t^2/2^\sigma)$
RBR KS IOR
Grover's alg:
Preimage finding
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k)^2 \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
relaxed
RBR KS IOR
$\kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}$
[BCFW25]
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k) \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Vector commitment (VC) :
an abstraction of MT
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}$
PQ Extractable VC
$\epsilon^{\star}_{\mathsf{VC}}$
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}$
Extractable VC
ϵ_{vc}
[BMNW25b,BCS16]
BCS[IOR, VC] is PQ
knowledge sound
$\kappa^{\star}_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}} + \epsilon^{\star}_{\mathsf{VC}})$.
BCS[IOR, VC] is knowledge sound
$\kappa_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\mathsf{sr}}_{\mathsf{IOR}} + \epsilon_{\mathsf{VC}})$.

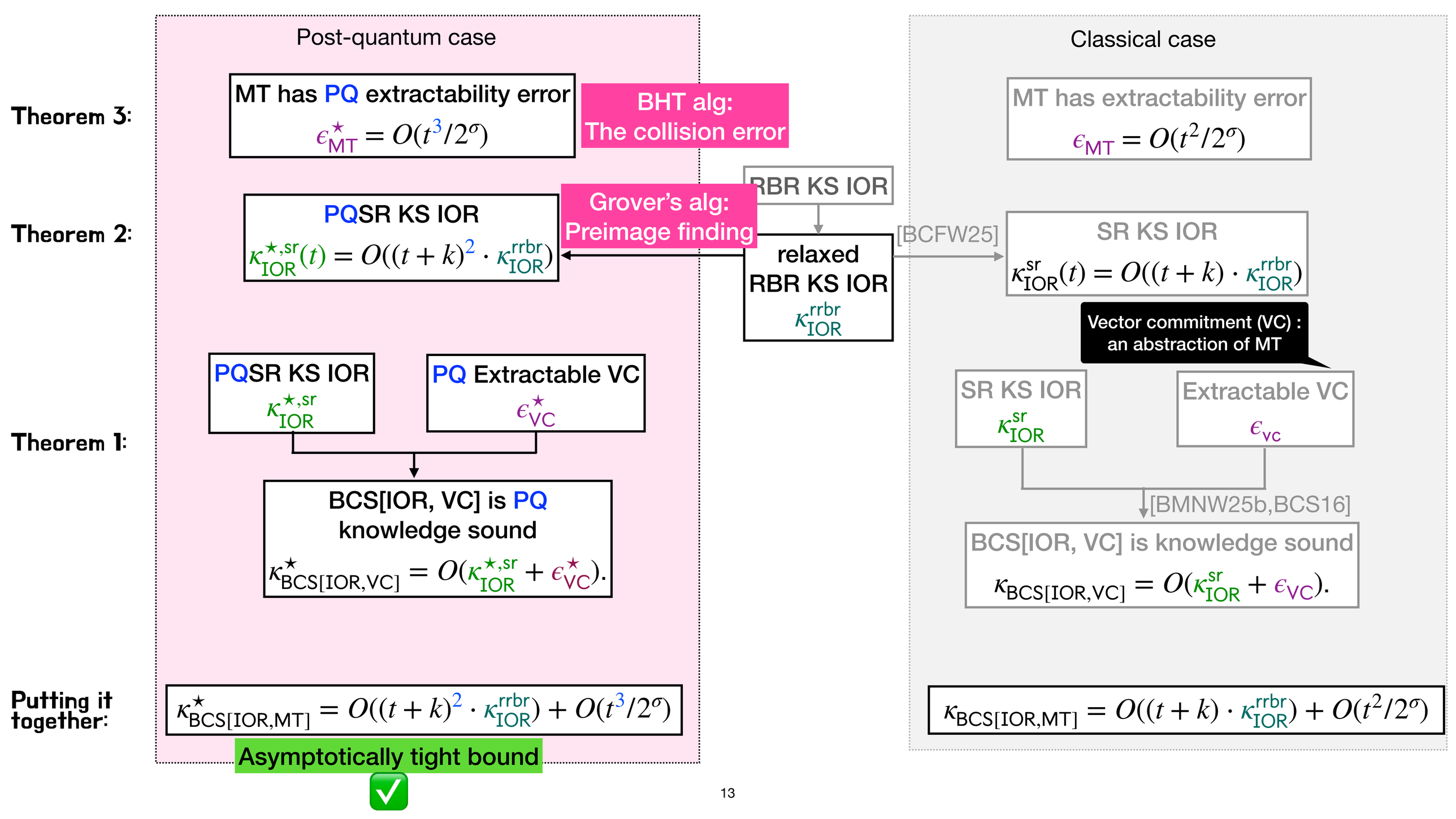

Post-quantum case
Classical case
Theorem 3:
Theorem 2:
Theorem 1:
MT has PQ extractability error
$\epsilon^{\star}_{\mathsf{MT}} = O(t^3/2^{\sigma})$
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k)^2 \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Grover's alg:
Preimage finding
RBR KS IOR
relaxed
RBR KS IOR
$\kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}$
[BCFW25]
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k) \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Vector commitment (VC) :
an abstraction of MT
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}$
PQ Extractable VC
$\epsilon^{\star}_{\mathsf{VC}}$
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}$
Extractable VC
ϵ_{vc}
[BMNW25b,BCS16]
BCS[IOR, VC] is PQ
knowledge sound
$\kappa^{\star}_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}} + \epsilon^{\star}_{\mathsf{VC}})$.
BCS[IOR, VC] is knowledge sound
$\kappa_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\mathsf{sr}}_{\mathsf{IOR}} + \epsilon_{\mathsf{VC}})$.

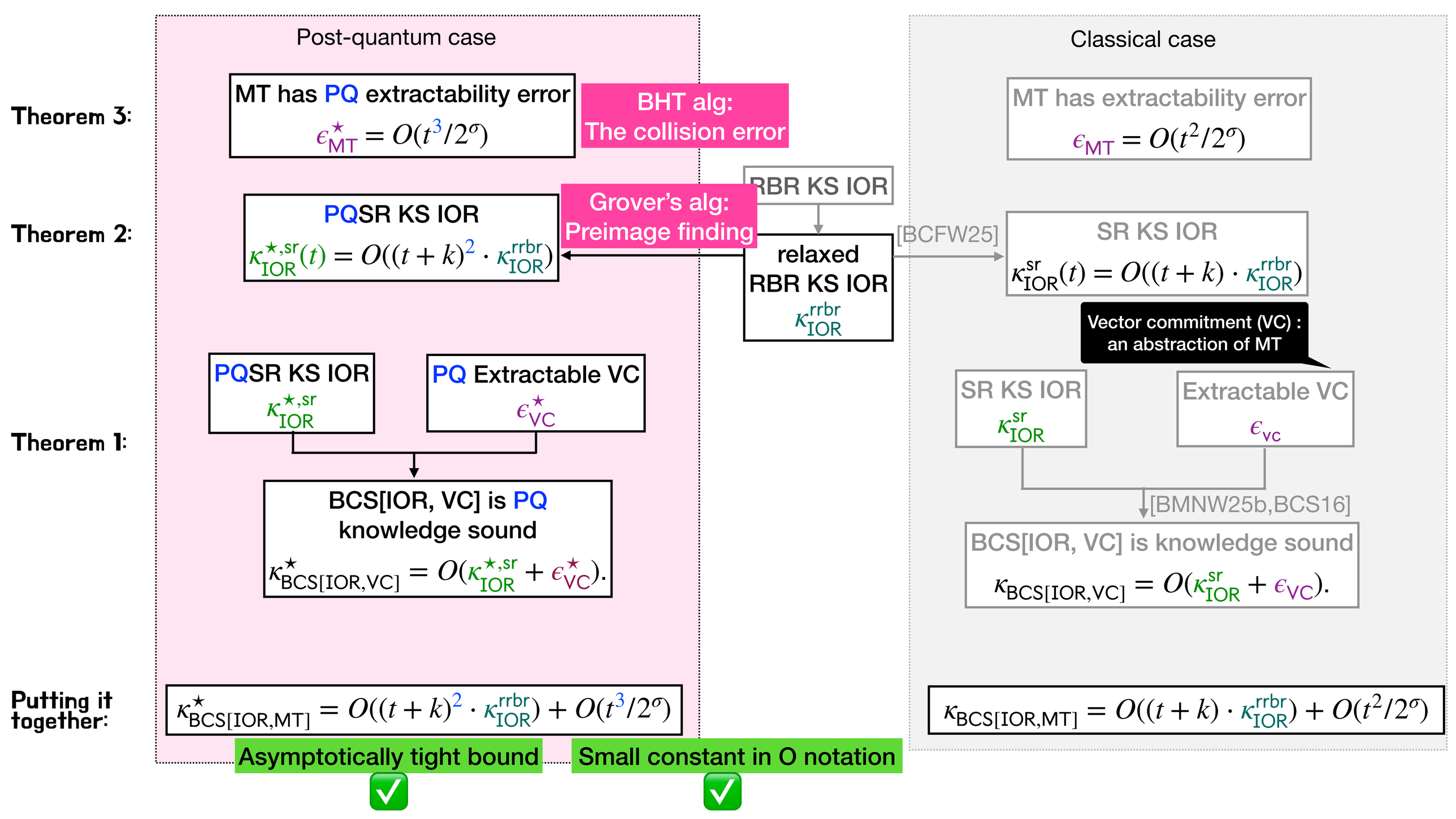

Post-quantum case
Classical case
Theorem 3:
Theorem 2:
Theorem 1:
MT has PQ extractability error
$\epsilon^{\star}_{\mathsf{MT}} = O(t^3/2^{\sigma})$
BHT alg:
The collision error
MT has extractability error
$\epsilon_{\mathsf{MT}} = O(t^2/2^{\sigma})$
RBR KS IOR
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k)^2 \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Grover's alg:
Preimage finding
relaxed
RBR KS IOR
$\kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}$
[BCFW25]
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k) \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Vector commitment (VC) :
an abstraction of MT
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}$
PQ Extractable VC
$\epsilon^{\star}_{\mathsf{VC}}$
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}$
Extractable VC
ϵ_{vc}
[BMNW25b,BCS16]
BCS[IOR, VC] is PQ
knowledge sound
$\kappa^{\star}_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}} + \epsilon^{\star}_{\mathsf{VC}})$.
BCS[IOR, VC] is knowledge sound
$\kappa_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\mathsf{sr}}_{\mathsf{IOR}} + \epsilon_{\mathsf{VC}})$.


Post-quantum case
Classical case
Theorem 3:
Theorem 2:
Theorem 1:
Putting it together:
MT has PQ extractability error
$\epsilon^{\star}_{\mathsf{MT}} = O(t^3/2^{\sigma})$
BHT alg:
The collision error
MT has extractability error
$\epsilon_{\mathsf{MT}} = O(t^2/2^{\sigma})$
RBR KS IOR
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k)^2 \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Grover's alg:
Preimage finding
relaxed
RBR KS IOR
$\kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}$
[BCFW25]
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k) \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Vector commitment (VC) :
an abstraction of MT
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}$
PQ Extractable VC
$\epsilon^{\star}_{\mathsf{VC}}$
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}$
Extractable VC
ϵ_{vc}
[BMNW25b,BCS16]
BCS[IOR, VC] is PQ
knowledge sound
$\kappa^{\star}_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}} + \epsilon^{\star}_{\mathsf{VC}})$.
BCS[IOR, VC] is knowledge sound
$\kappa_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\mathsf{sr}}_{\mathsf{IOR}} + \epsilon_{\mathsf{VC}})$.


Post-quantum case
Classical case
Theorem 3:
Theorem 2:
Theorem 1:
Putting it together:
MT has PQ extractability error
$\epsilon^{\star}_{\mathsf{MT}} = O(t^3/2^{\sigma})$
BHT alg:
The collision error
MT has extractability error
$\epsilon_{\mathsf{MT}} = O(t^2/2^{\sigma})$
RBR KS IOR
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k)^2 \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Grover's alg:
Preimage finding
relaxed
RBR KS IOR
$\kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}$
[BCFW25]
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k) \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Vector commitment (VC) :
an abstraction of MT
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}$
PQ Extractable VC
$\epsilon^{\star}_{\mathsf{VC}}$
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}$
Extractable VC
ϵ_{vc}
[BMNW25b,BCS16]
BCS[IOR, VC] is PQ
knowledge sound
$\kappa^{\star}_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}} + \epsilon^{\star}_{\mathsf{VC}})$.
BCS[IOR, VC] is knowledge sound
$\kappa_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\mathsf{sr}}_{\mathsf{IOR}} + \epsilon_{\mathsf{VC}})$.
$\kappa_{\mathsf{BCS[IOR,MT]}} = O((t+k) \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}) + O(t^2/2^{\sigma})$


Post-quantum case
Classical case
Theorem 3:
Theorem 2:
Theorem 1:
Putting it together:
MT has PQ extractability error
$\epsilon^{\star}_{\mathsf{MT}} = O(t^3/2^{\sigma})$
BHT alg:
The collision error
MT has extractability error
$\epsilon_{\mathsf{MT}} = O(t^2/2^{\sigma})$
RBR KS IOR
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k)^2 \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Grover's alg:
Preimage finding
relaxed
RBR KS IOR
$\kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}$
[BCFW25]
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k) \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Vector commitment (VC) :
an abstraction of MT
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}$
PQ Extractable VC
$\epsilon^{\star}_{\mathsf{VC}}$
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}$
Extractable VC
ϵ_{vc}
[BMNW25b,BCS16]
BCS[IOR, VC] is PQ
knowledge sound
$\kappa^{\star}_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}} + \epsilon^{\star}_{\mathsf{VC}})$.
BCS[IOR, VC] is knowledge sound
$\kappa_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\mathsf{sr}}_{\mathsf{IOR}} + \epsilon_{\mathsf{VC}})$.
$\kappa^{\star}_{\mathsf{BCS[IOR,MT]}} = O((t+k)^2 \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}) + O(t^3/2^{\sigma})$
$\kappa_{\mathsf{BCS[IOR,MT]}} = O((t+k) \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}) + O(t^2/2^{\sigma})$


Post-quantum case
Classical case
Theorem 3:
MT has PQ extractability error
$\epsilon^{\star}_{\mathsf{MT}} = O(t^3/2^{\sigma})$
BHT alg:
The collision error
MT has extractability error
$\epsilon_{\mathsf{MT}} = O(t^2/2^{\sigma})$
Theorem 2:
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k)^2 \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Grover's alg:
Preimage finding
RBR KS IOR
relaxed
RBR KS IOR
$\kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}$
[BCFW25]
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k) \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Vector commitment (VC) :
an abstraction of MT
Theorem 1:
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}$
PQ Extractable VC
$\epsilon^{\star}_{\mathsf{VC}}$
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}$
Extractable VC
ϵ_{vc}
BCS[IOR, VC] is PQ
knowledge sound
$\kappa^{\star}_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}} + \epsilon^{\star}_{\mathsf{VC}})$.
[BMNW25b,BCS16]
BCS[IOR, VC] is knowledge sound
$\kappa_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\mathsf{sr}}_{\mathsf{IOR}} + \epsilon_{\mathsf{VC}})$.
Putting it together:
$\kappa^{\star}_{\mathsf{BCS[IOR,MT]}} = O((t+k)^2 \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}) + O(t^3/2^{\sigma})$
$\kappa_{\mathsf{BCS[IOR,MT]}} = O((t+k) \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}) + O(t^2/2^{\sigma})$
Asymptotically tight bound
✅


Post-quantum case
Classical case
Theorem 3:
MT has PQ extractability error
$\epsilon^{\star}_{\mathsf{MT}} = O(t^3/2^{\sigma})$
BHT alg:
The collision error
MT has extractability error
$\epsilon_{\mathsf{MT}} = O(t^2/2^{\sigma})$
Theorem 2:
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k)^2 \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Grover's alg:
Preimage finding
RBR KS IOR
relaxed
RBR KS IOR
$\kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}$
[BCFW25]
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}(t) = O((t+k) \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}})$
Vector commitment (VC) :
an abstraction of MT
Theorem 1:
PQSR KS IOR
$\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}}$
PQ Extractable VC
$\epsilon^{\star}_{\mathsf{VC}}$
SR KS IOR
$\kappa^{\mathsf{sr}}_{\mathsf{IOR}}$
Extractable VC
ϵ_{vc}
BCS[IOR, VC] is PQ
knowledge sound
$\kappa^{\star}_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\star,\mathsf{sr}}_{\mathsf{IOR}} + \epsilon^{\star}_{\mathsf{VC}})$.
[BMNW25b,BCS16]
BCS[IOR, VC] is knowledge sound
$\kappa_{\mathsf{BCS[IOR,VC]}} = O(\kappa^{\mathsf{sr}}_{\mathsf{IOR}} + \epsilon_{\mathsf{VC}})$.
Putting it together:
$\kappa^{\star}_{\mathsf{BCS[IOR,MT]}} = O((t+k)^2 \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}) + O(t^3/2^{\sigma})$
$\kappa_{\mathsf{BCS[IOR,MT]}} = O((t+k) \cdot \kappa^{\mathsf{rrbr}}_{\mathsf{IOR}}) + O(t^2/2^{\sigma})$
Asymptotically tight bound
Small constant in O notation

Technical Overview

Ideal model for hash functions

Ideal model for hash functions

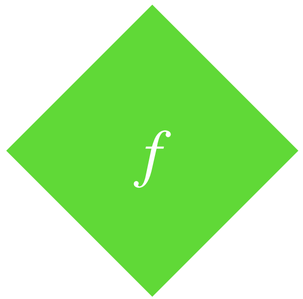
Random oracle $f \leftarrow (\{0,1\}^* \rightarrow \{0,1\}^\sigma)$

Ideal model for hash functions

Random oracle $f \leftarrow (\{0,1\}^* \rightarrow \{0,1\}^\sigma)$

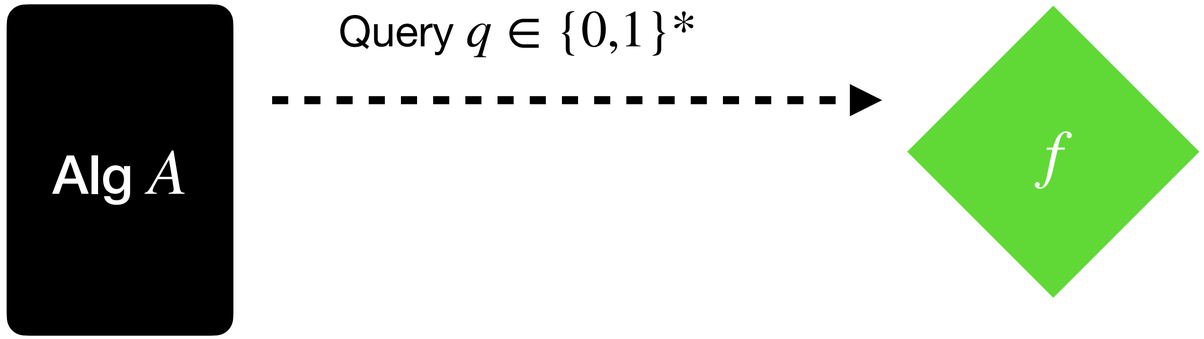
Ideal model for hash functions

Random oracle $f \leftarrow (\{0,1\}^* \rightarrow \{0,1\}^\sigma)$



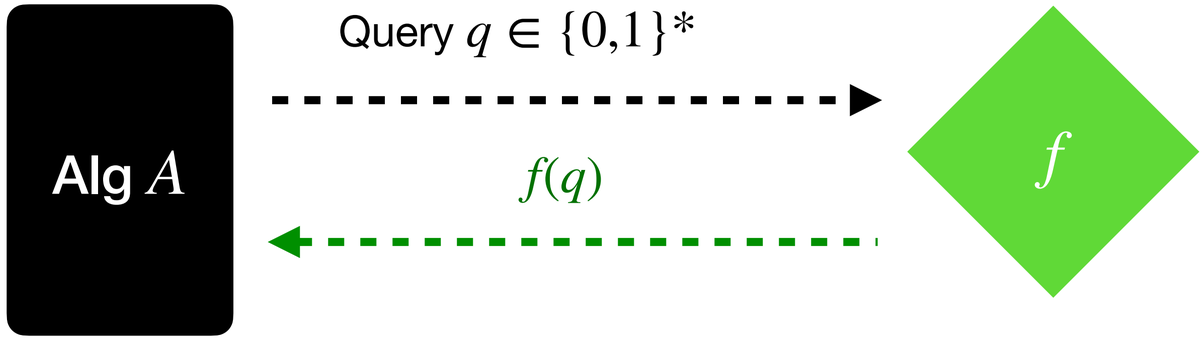
Ideal model for hash functions

Random oracle $f \leftarrow (\{0,1\}^* \rightarrow \{0,1\}^\sigma)$



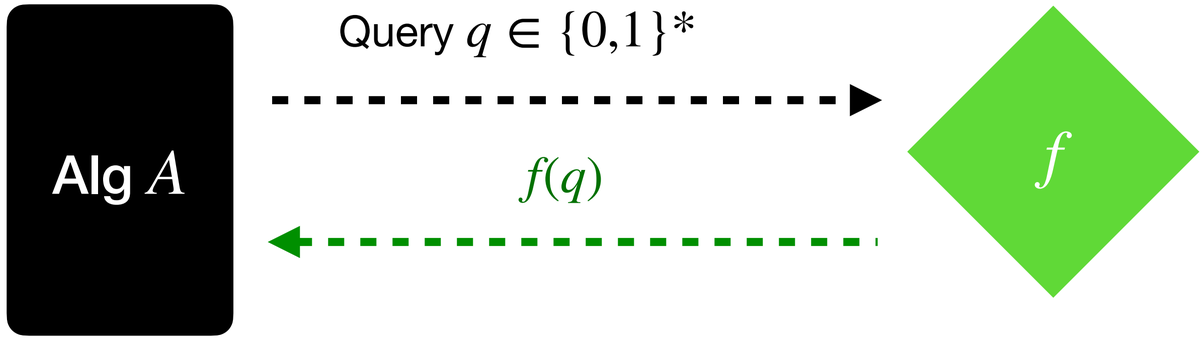
Ideal model for hash functions

Random oracle $f \leftarrow (\{0,1\}^* \rightarrow \{0,1\}^\sigma)$



Ideal model for hash functions

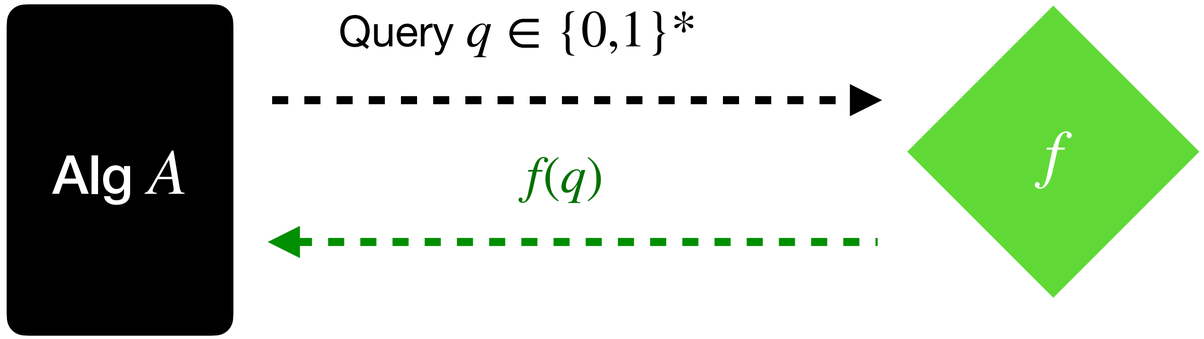
Random oracle $f \leftarrow (\{0,1\}^* \rightarrow \{0,1\}^\sigma)$



Quantum random oracle $f \leftarrow (\{0,1\}^* \rightarrow \{0,1\}^\sigma)$

Ideal model for hash functions

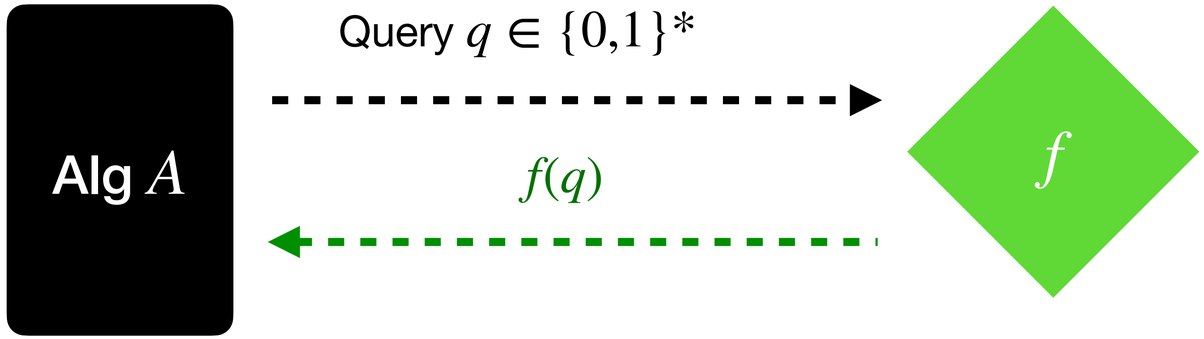
Random oracle $f \leftarrow (\{0,1\}^* \rightarrow \{0,1\}^\sigma)$



Quantum random oracle $f \leftarrow (\{0,1\}^* \rightarrow \{0,1\}^\sigma)$

Ideal model for hash functions

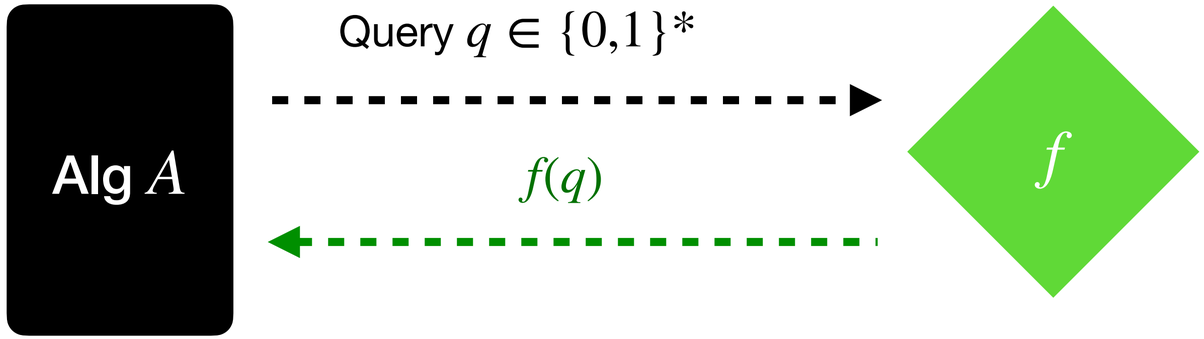
Random oracle $f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma)$



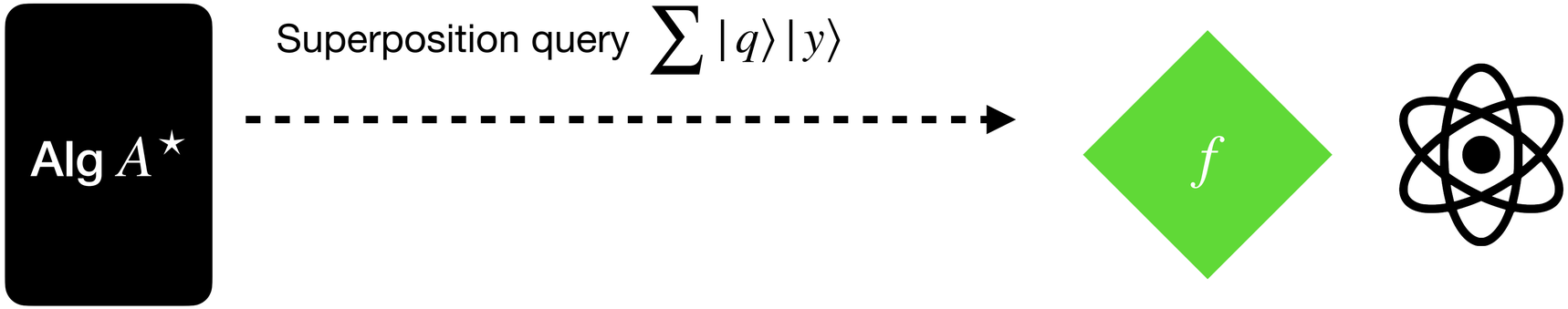
Quantum random oracle $f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma)$

Ideal model for hash functions

Random oracle $f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma)$

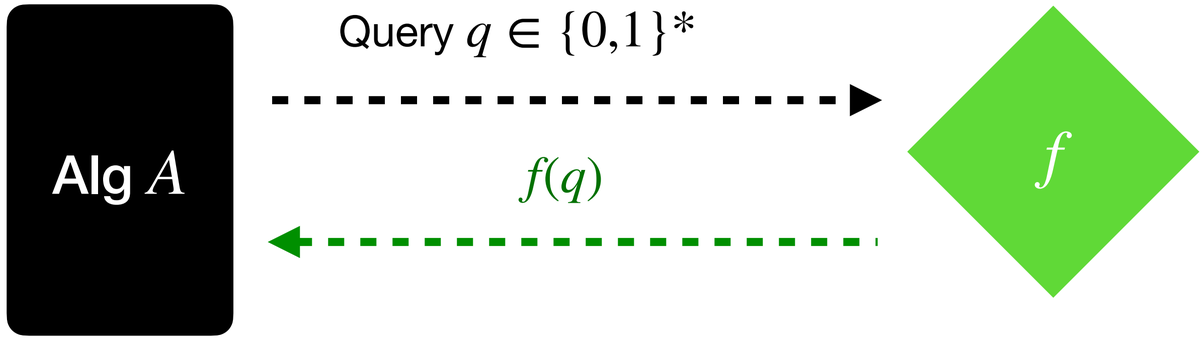


Quantum random oracle $f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma)$

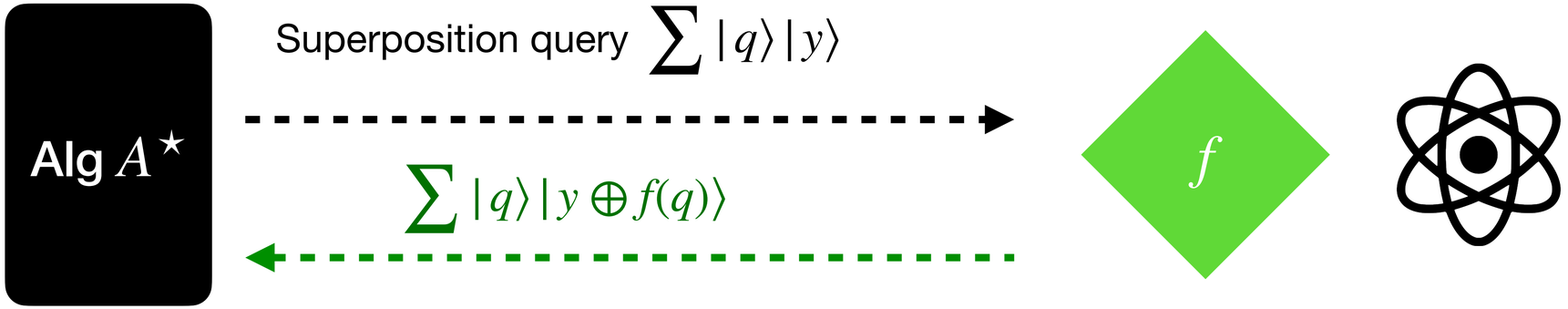


Ideal model for hash functions

Random oracle $f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma)$



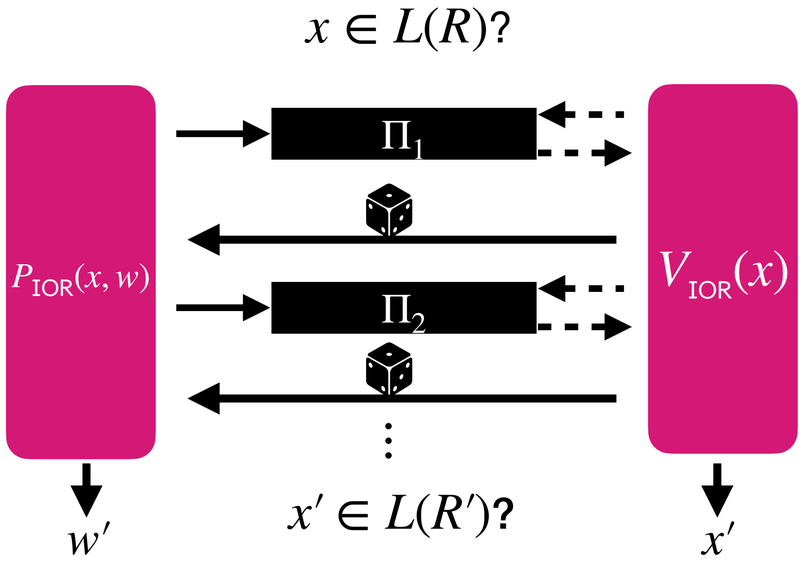
Quantum random oracle $f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma)$



How to remove interaction?

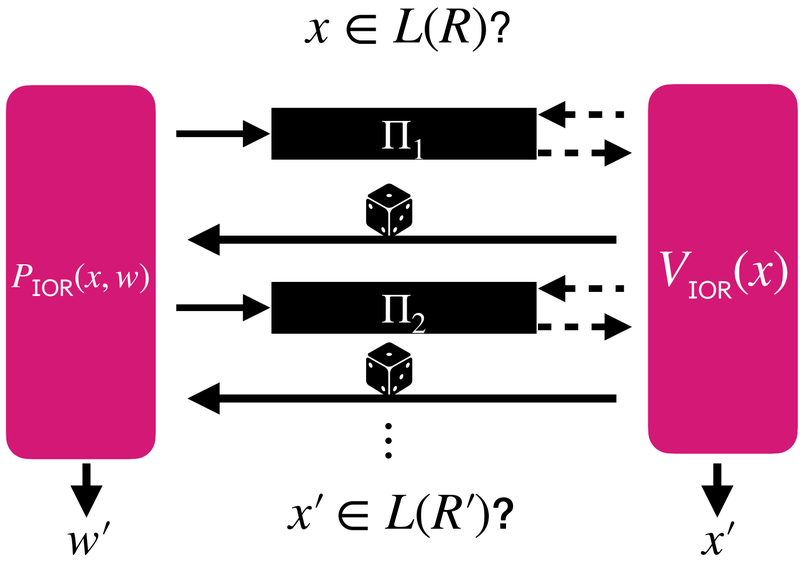
How to remove interaction?

Interactive oracle reduction (IOR)



How to remove interaction?

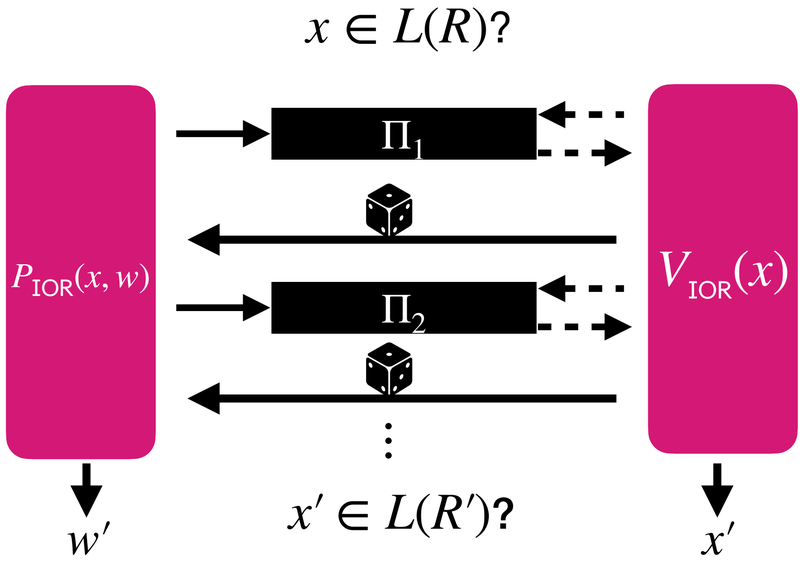
Interactive oracle reduction (IOR)



Omitted: instances x, x' can also include oracles.

How to remove interaction?

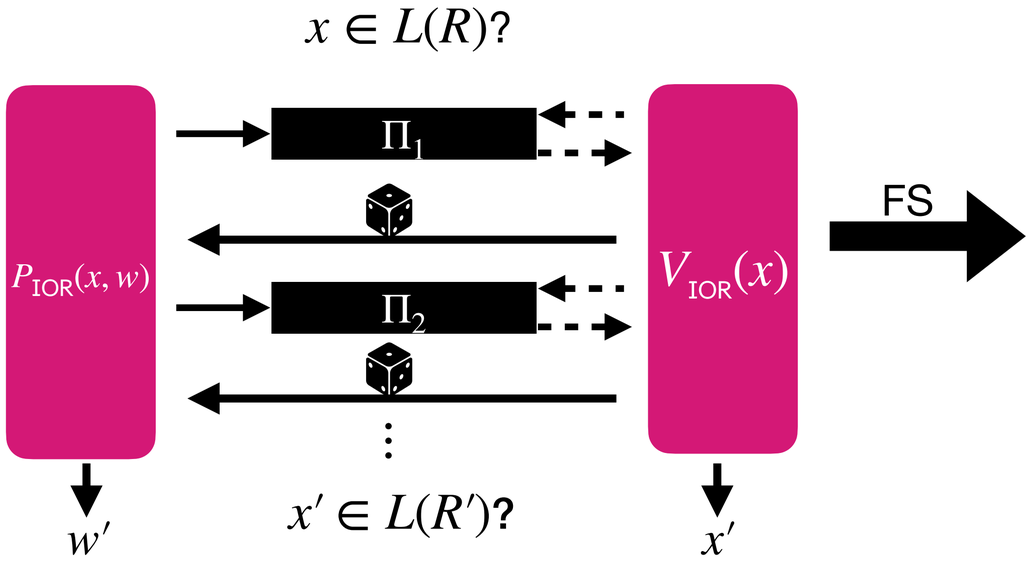
Interactive oracle reduction (IOR) **Interactive**



Omitted: instances x, x' can also include oracles.

How to remove interaction?

Interactive oracle reduction (IOR) **Interactive**

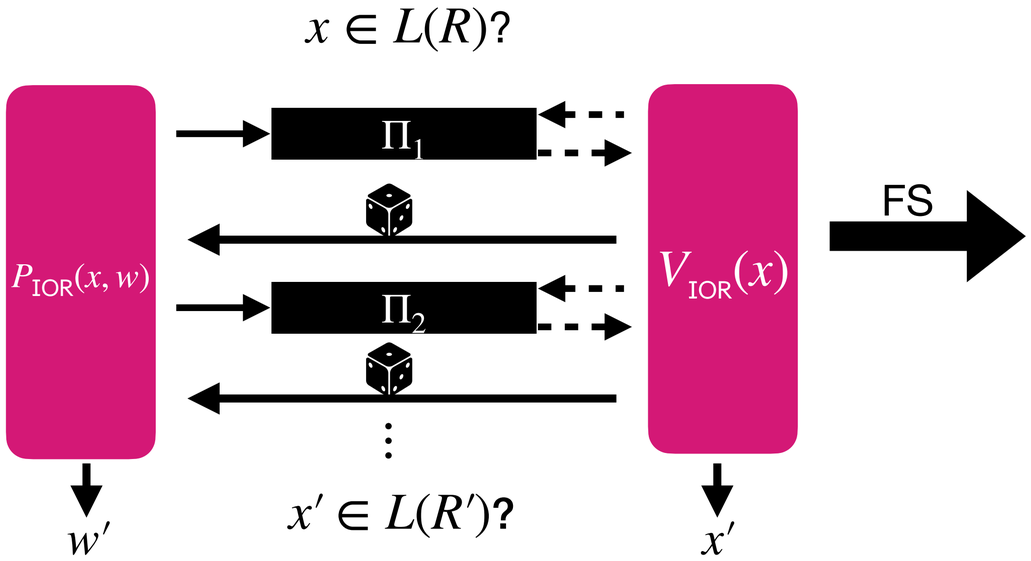


Omitted: instances x, x' can also include oracles.

How to remove interaction?

Interactive oracle reduction (IOR) Interactive

Non-interactive



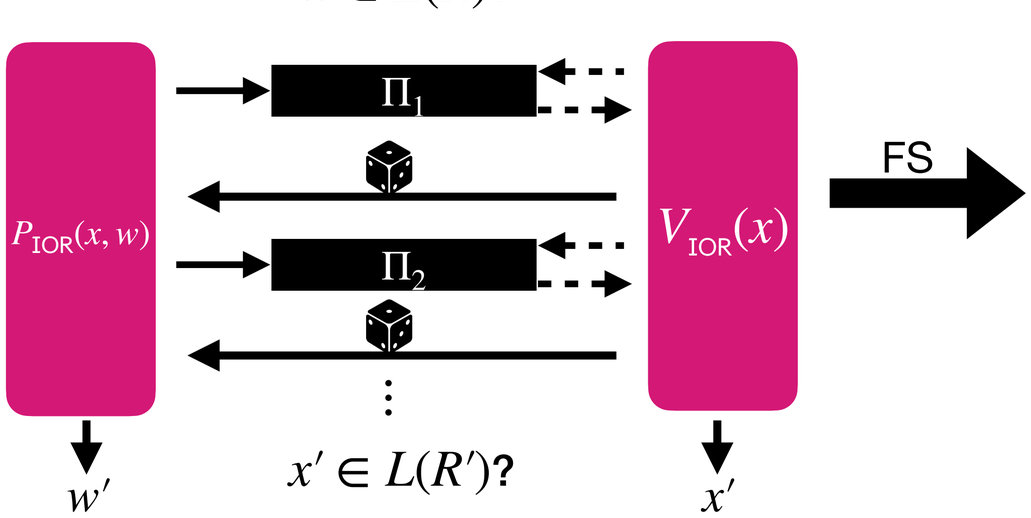
Omitted: instances x, x' can also include oracles.

How to remove interaction?

Interactive oracle reduction (IOR) Interactive

Non-interactive

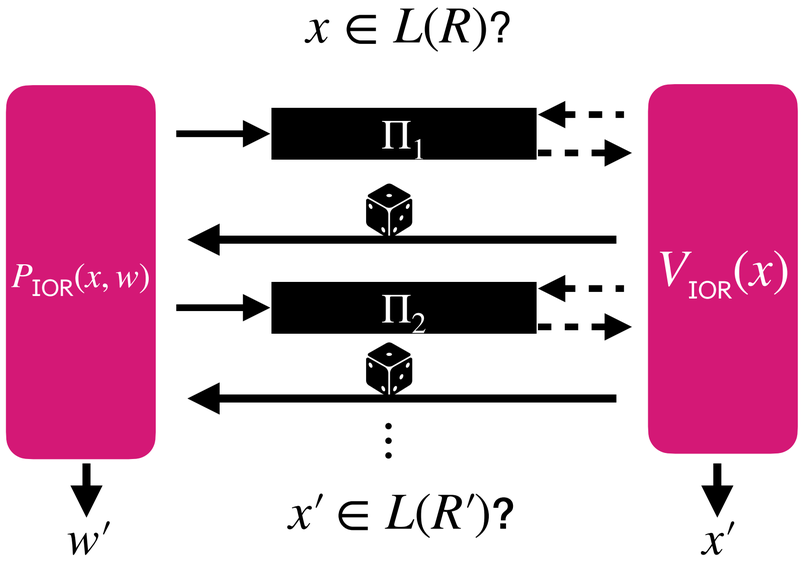
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

Interactive oracle reduction (IOR) Interactive



FS

Non-interactive

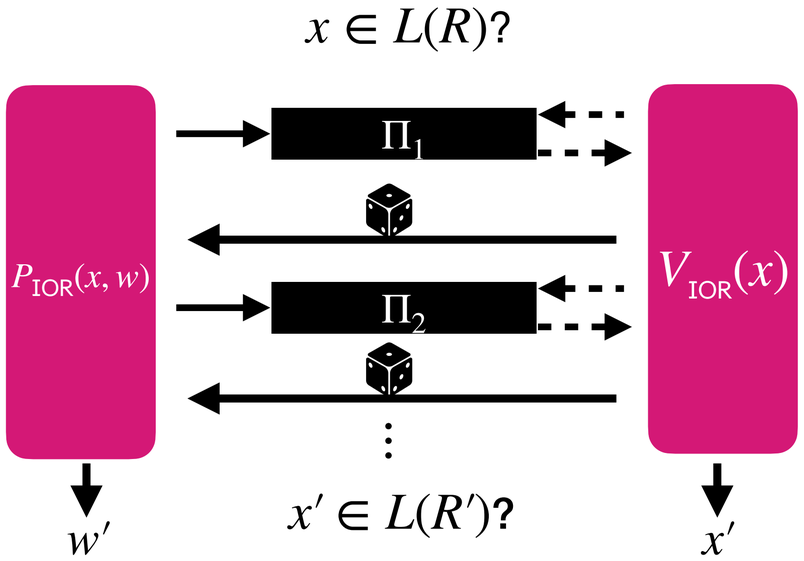
Use random function to derive randomness

P_{FS}

Omitted: instances x, x' can also include oracles.

How to remove interaction?

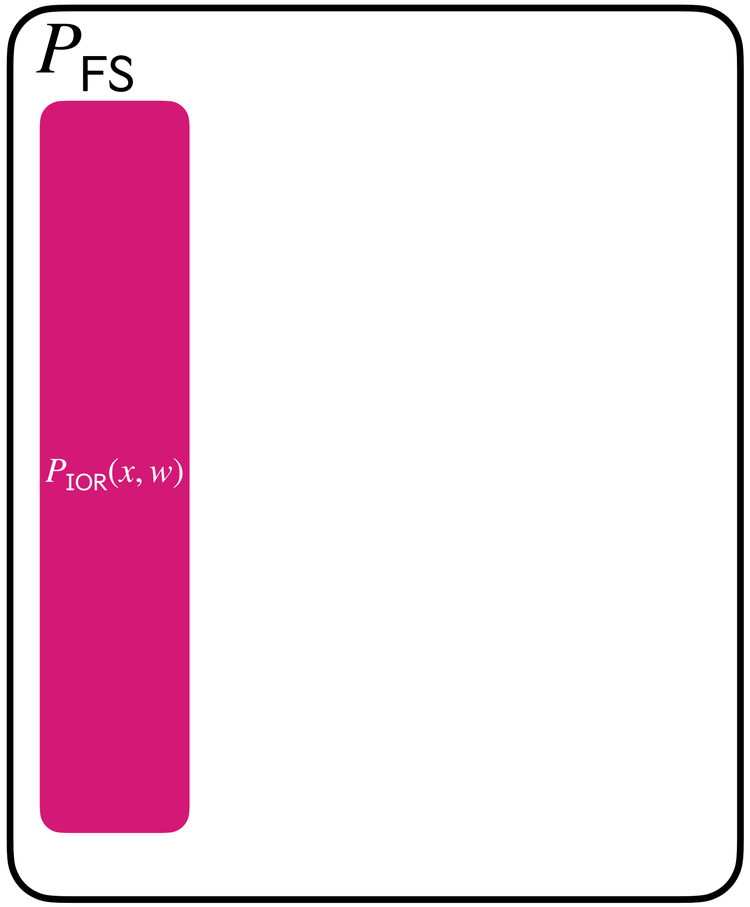
Interactive oracle reduction (IOR) **Interactive**



FS

Non-interactive

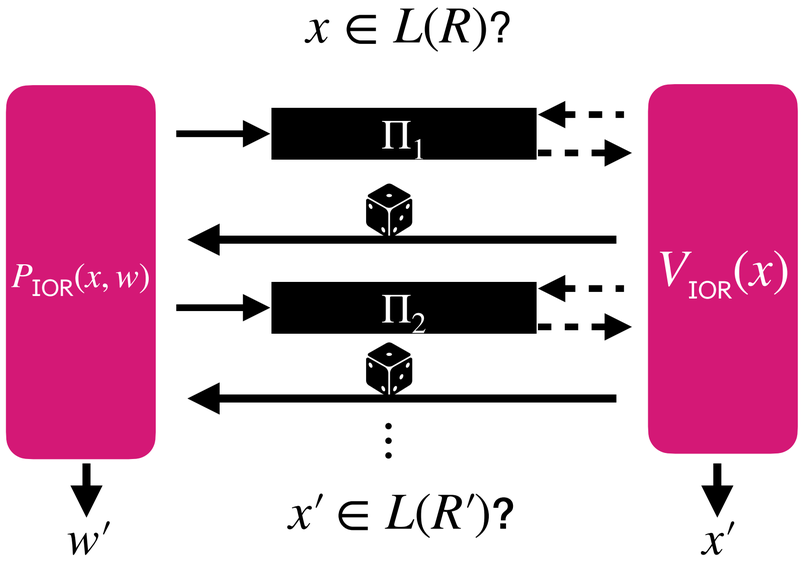
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

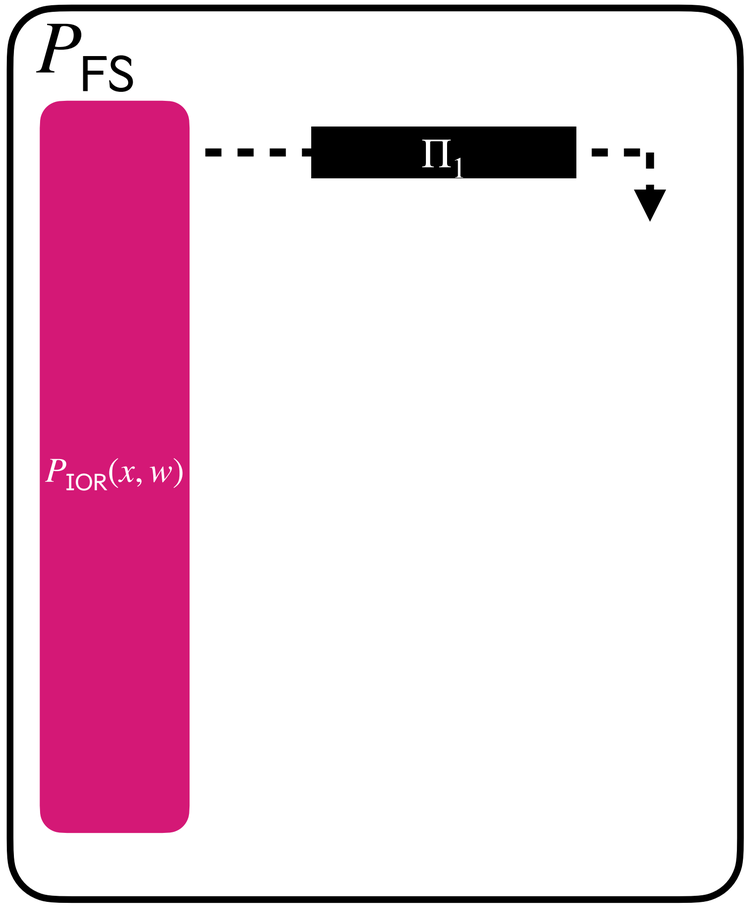
Interactive oracle reduction (IOR) **Interactive**



FS

Non-interactive

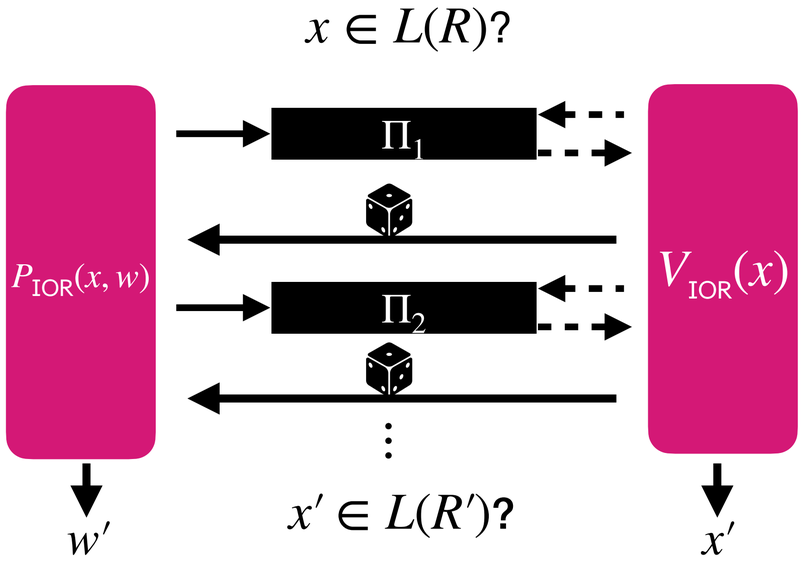
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

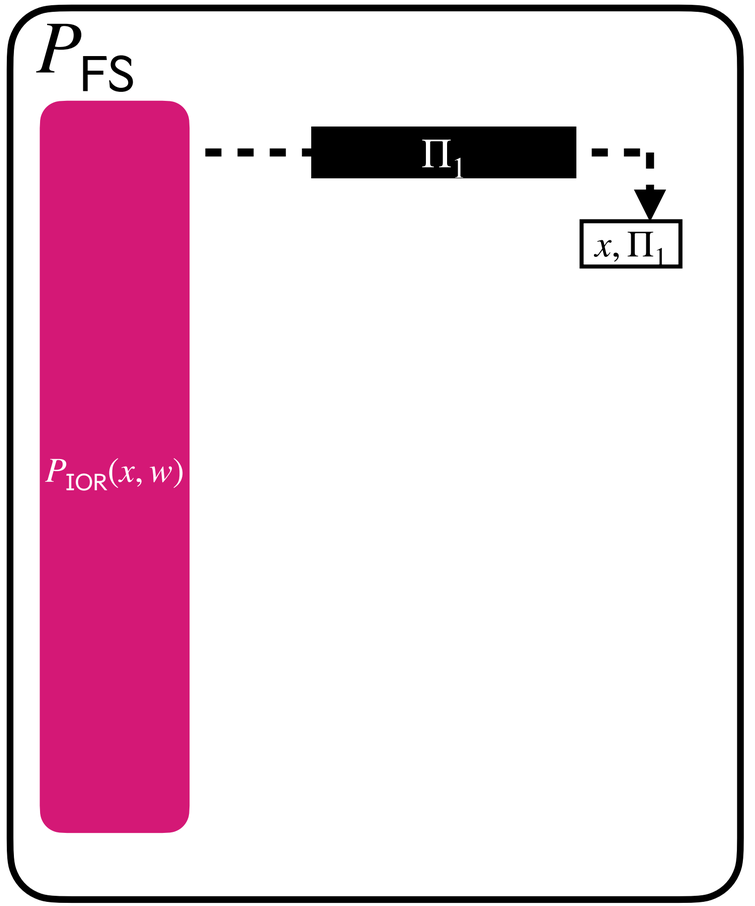
Interactive oracle reduction (IOR) **Interactive**



FS

Non-interactive

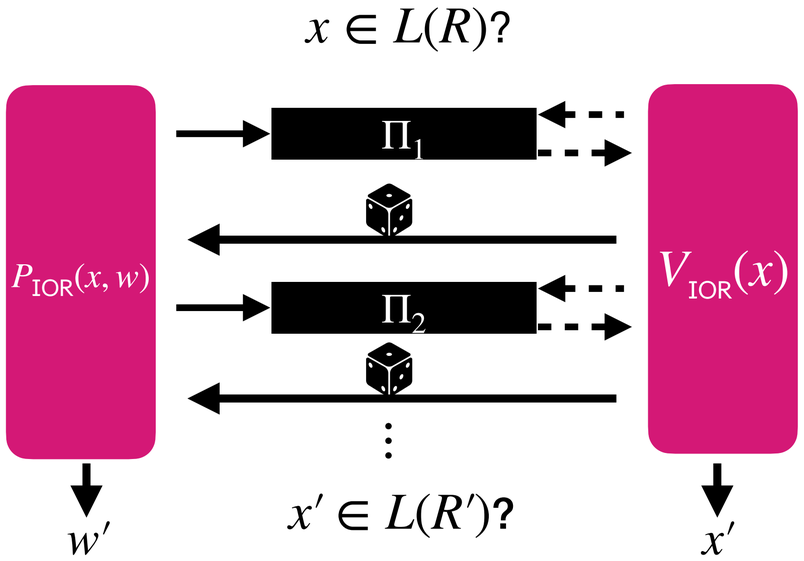
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

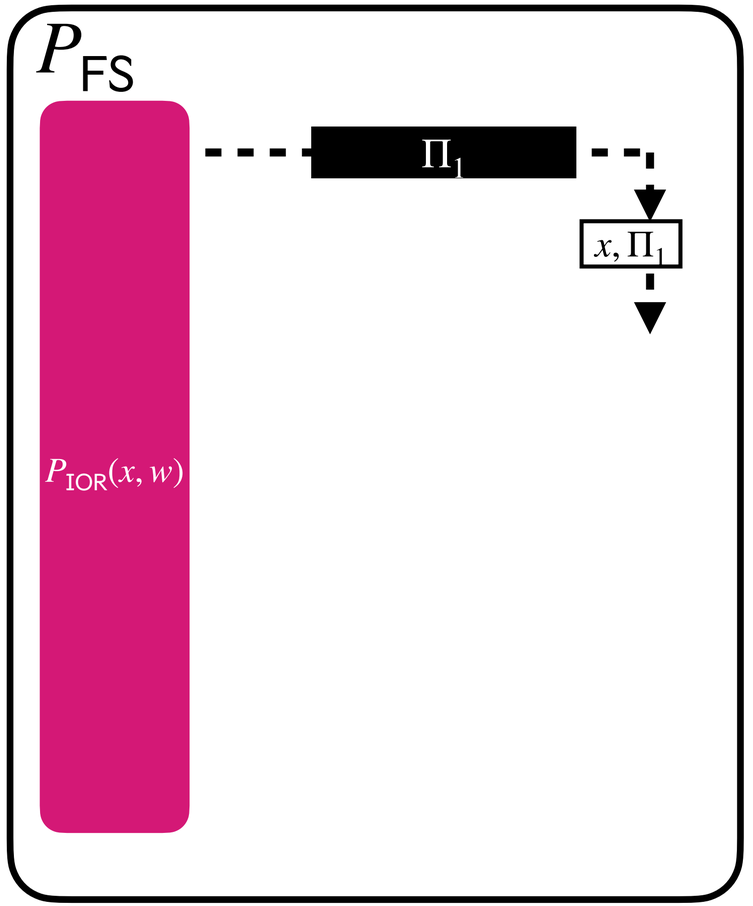
Interactive oracle reduction (IOR) **Interactive**



FS

Non-interactive

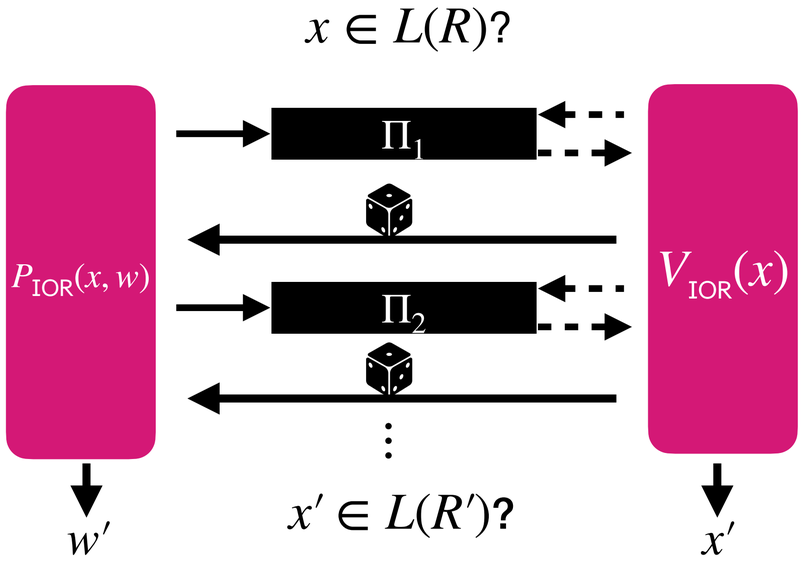
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

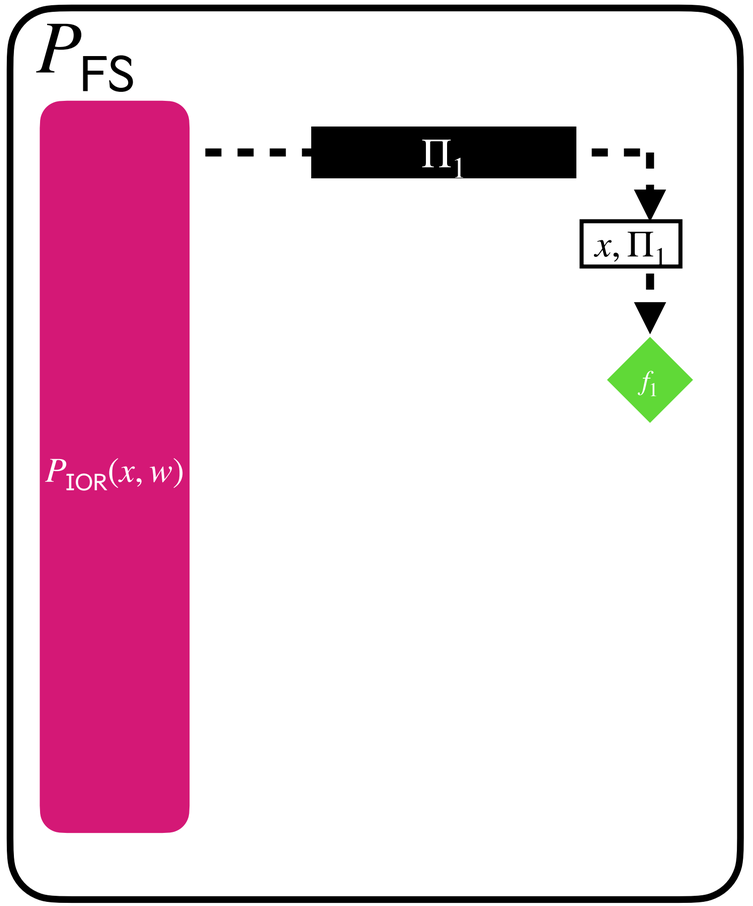
Interactive oracle reduction (IOR) Interactive



FS

Non-interactive

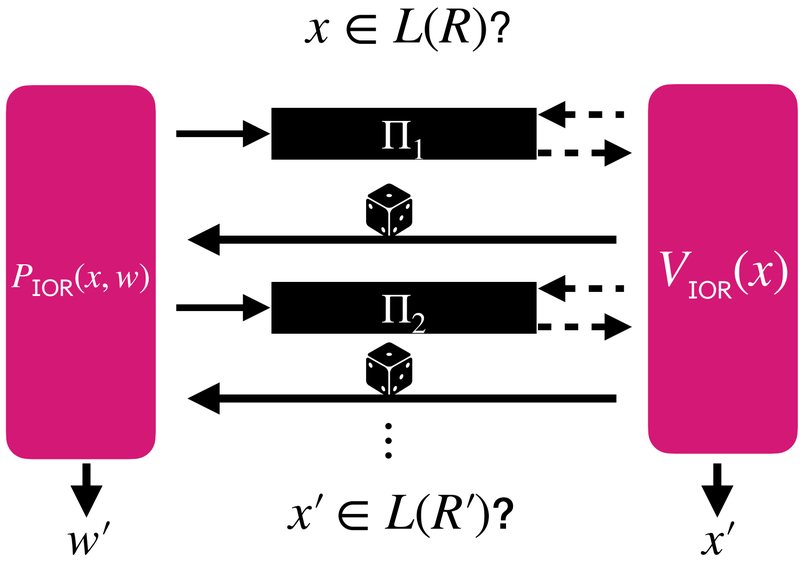
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

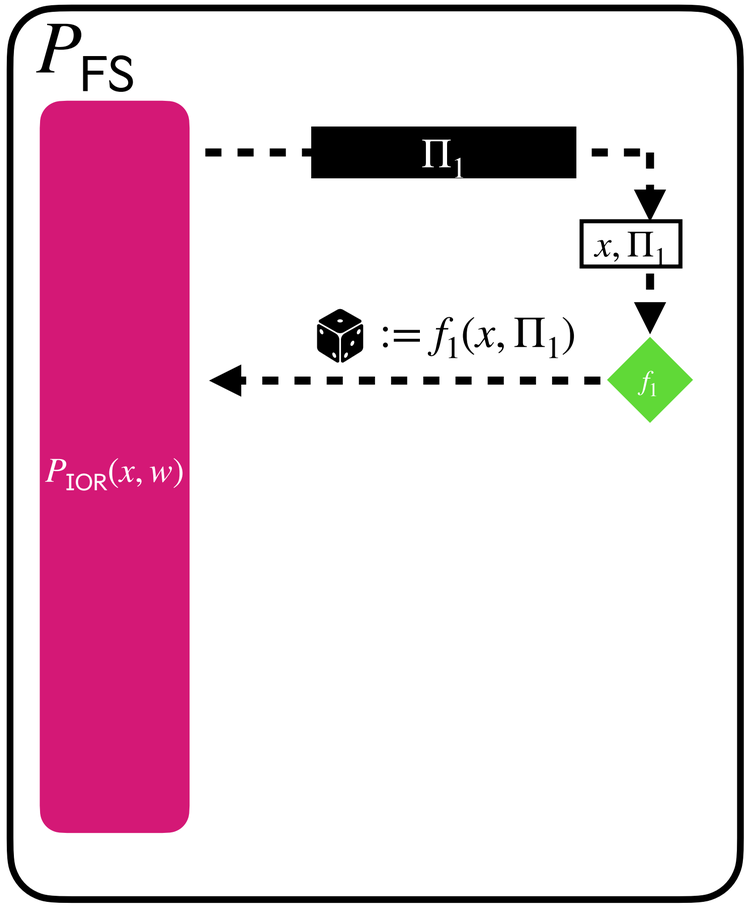
Interactive oracle reduction (IOR) **Interactive**



FS

Non-interactive

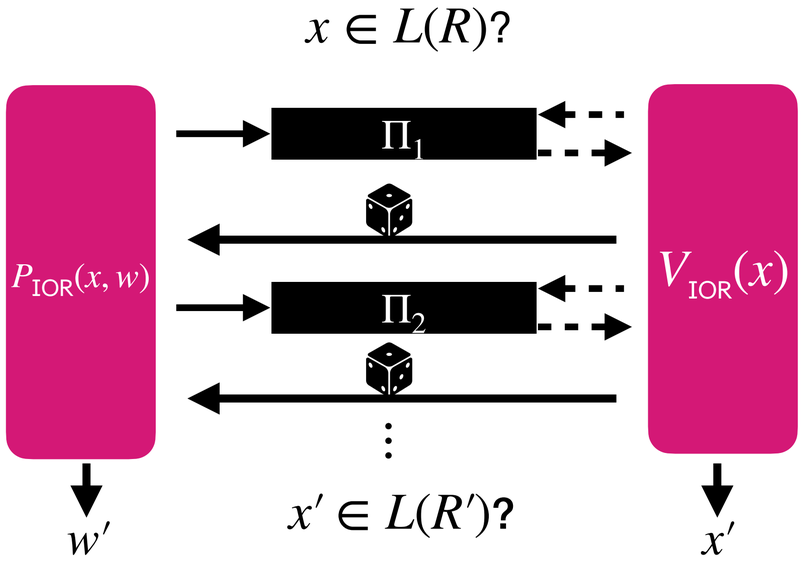
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

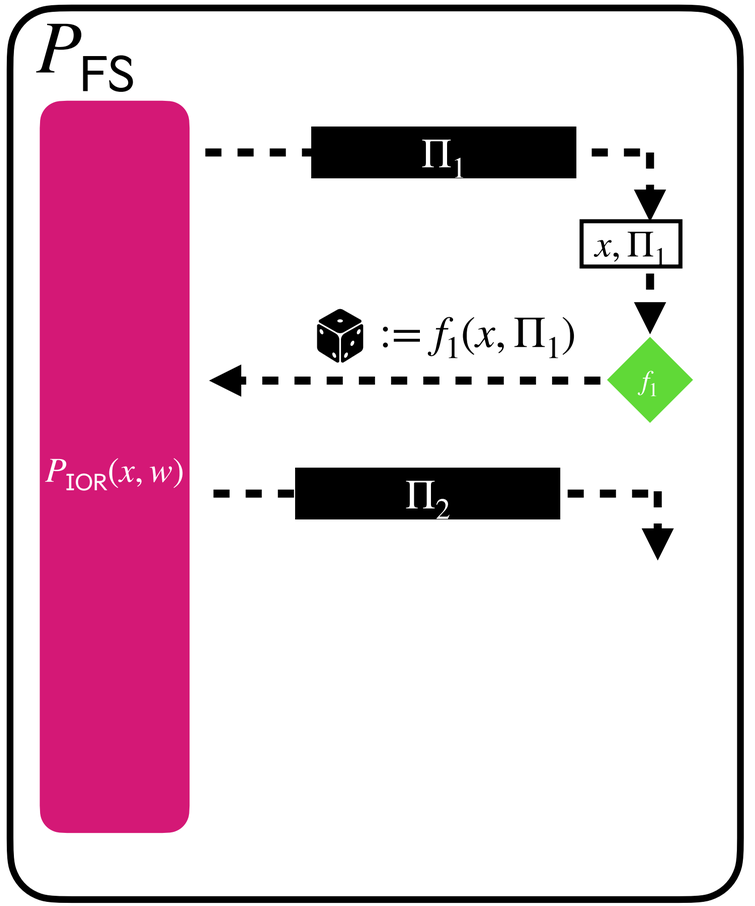
Interactive oracle reduction (IOR) **Interactive**



FS

Non-interactive

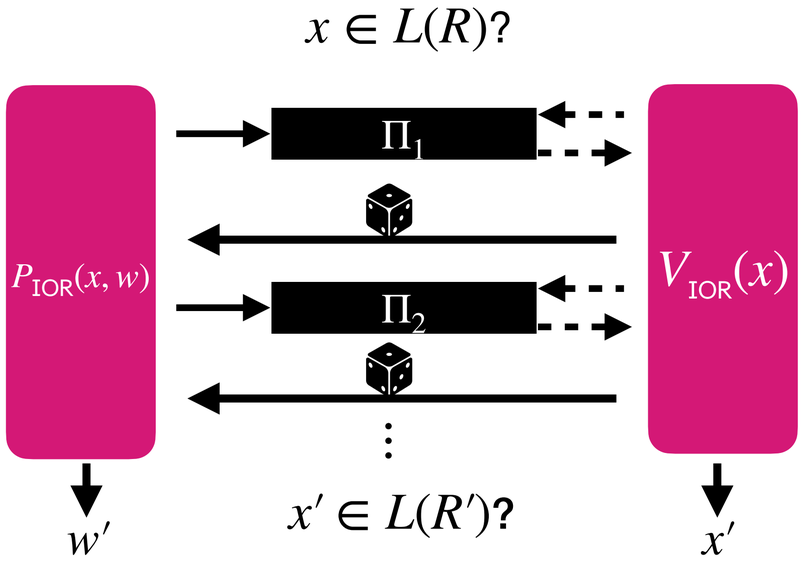
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

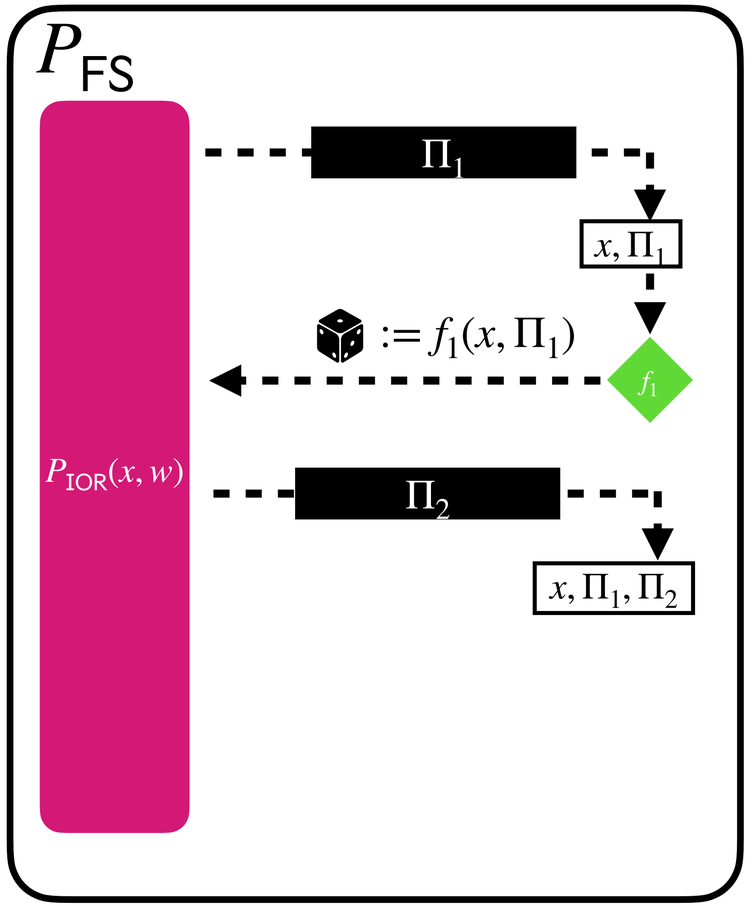
Interactive oracle reduction (IOR) **Interactive**



FS

Non-interactive

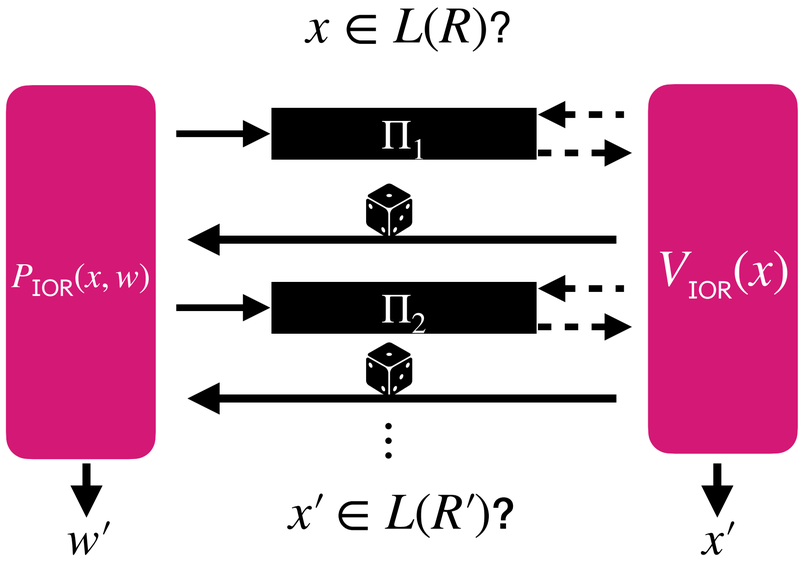
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

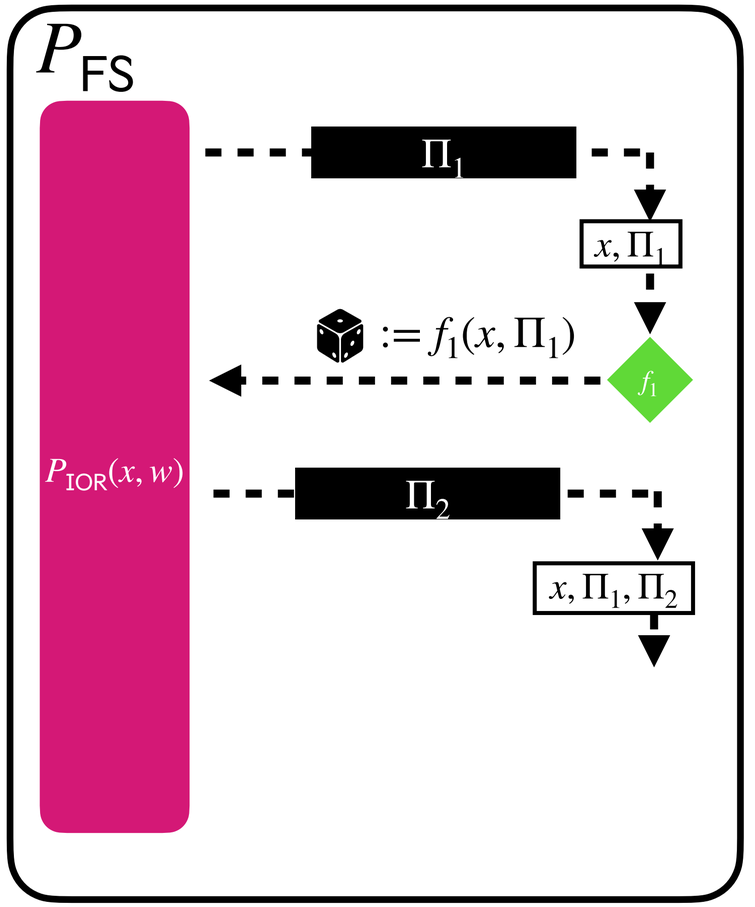
Interactive oracle reduction (IOR) **Interactive**



FS

Non-interactive

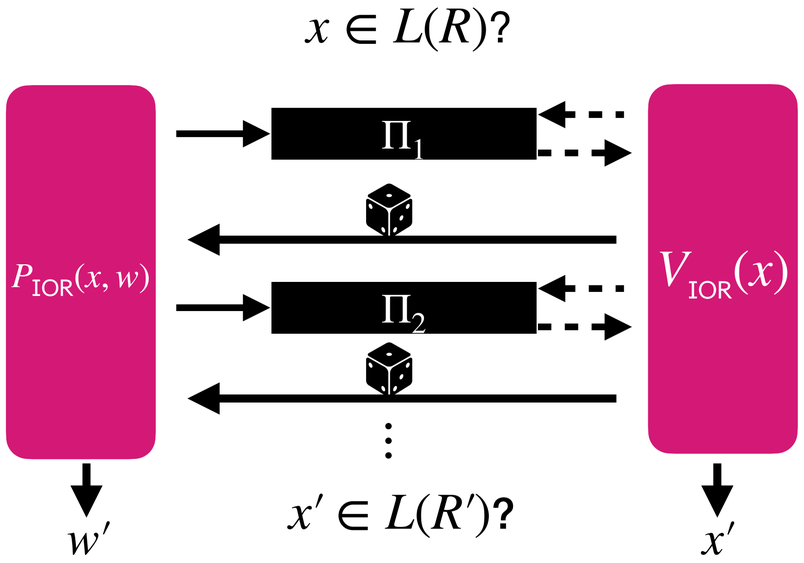
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

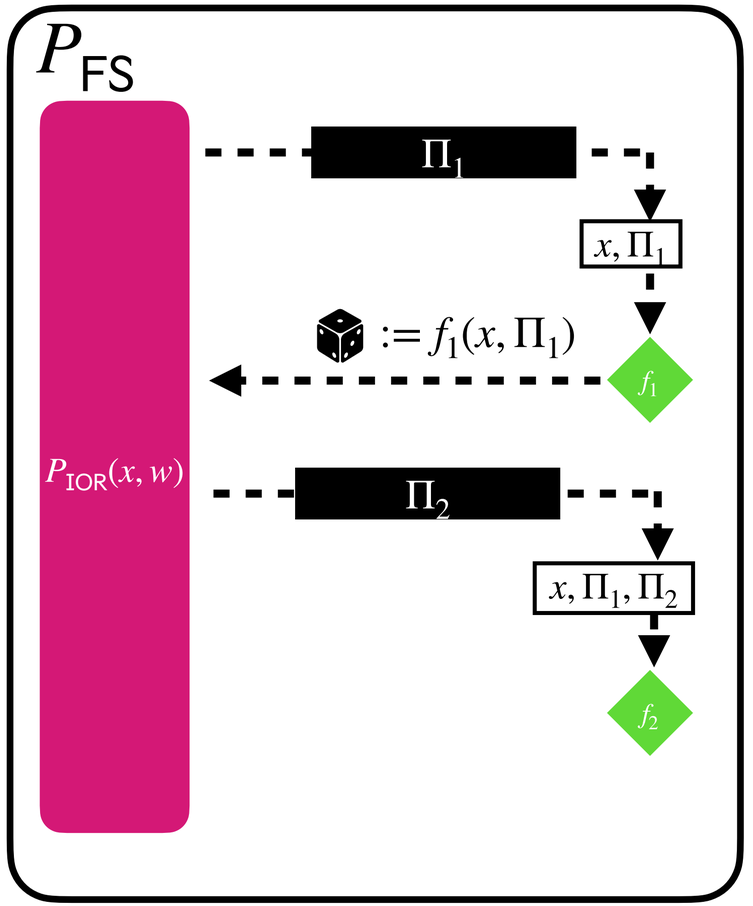
Interactive oracle reduction (IOR) Interactive



FS

Non-interactive

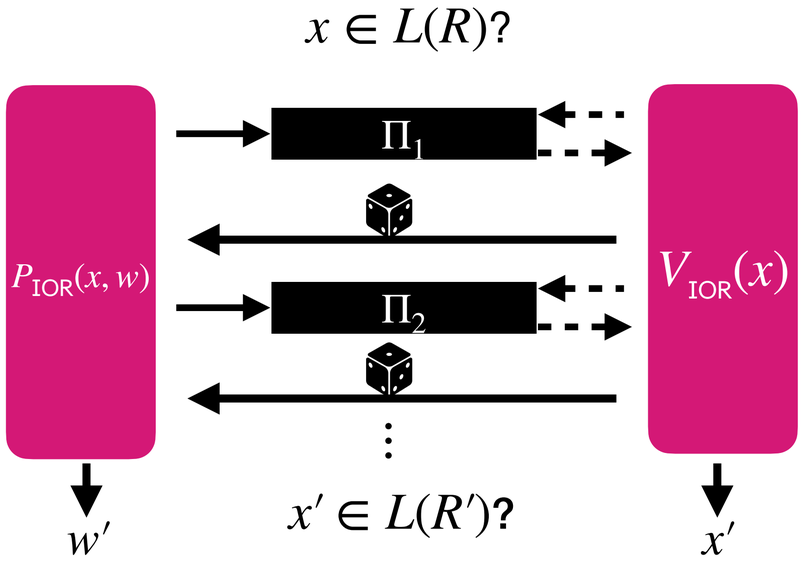
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

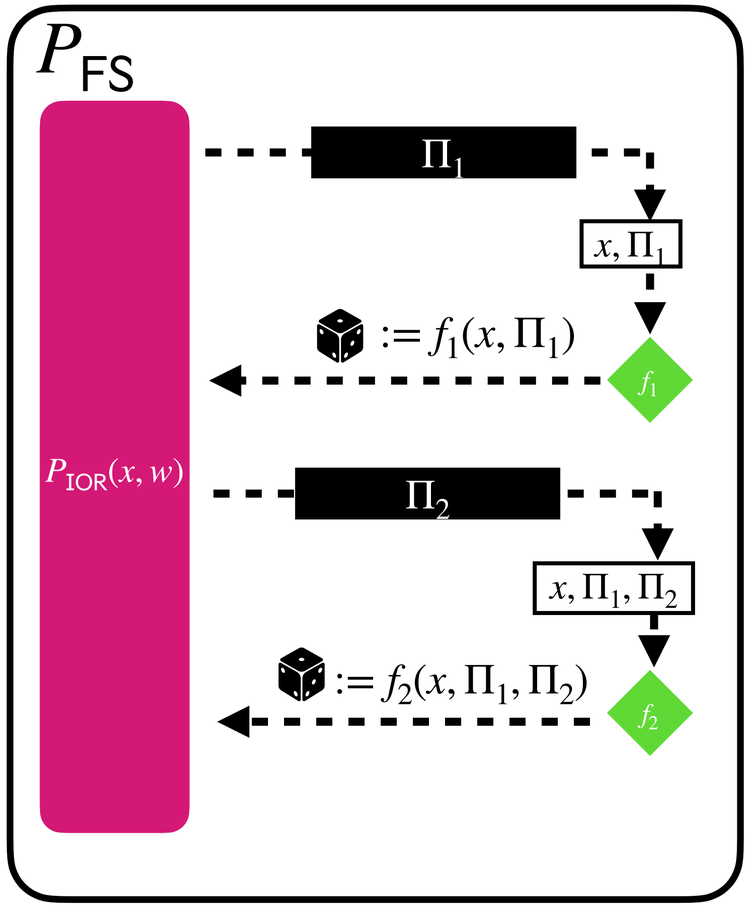
Interactive oracle reduction (IOR) **Interactive**



FS

Non-interactive

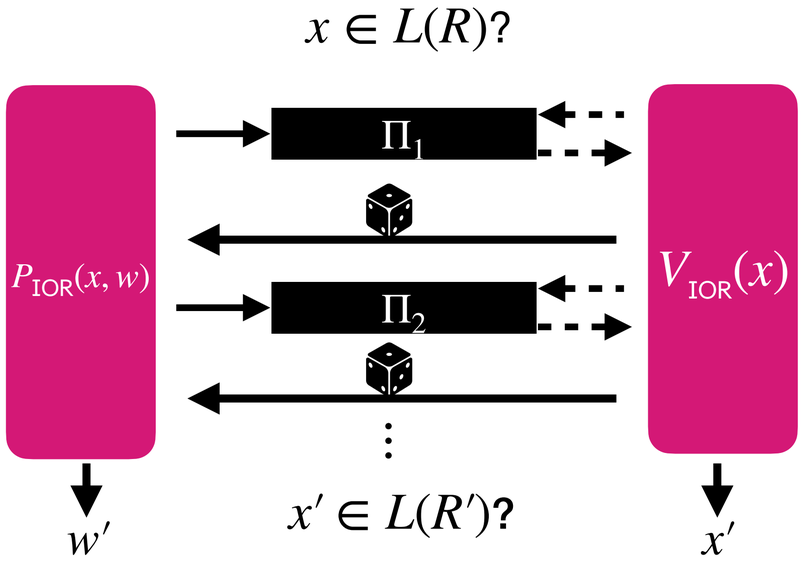
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

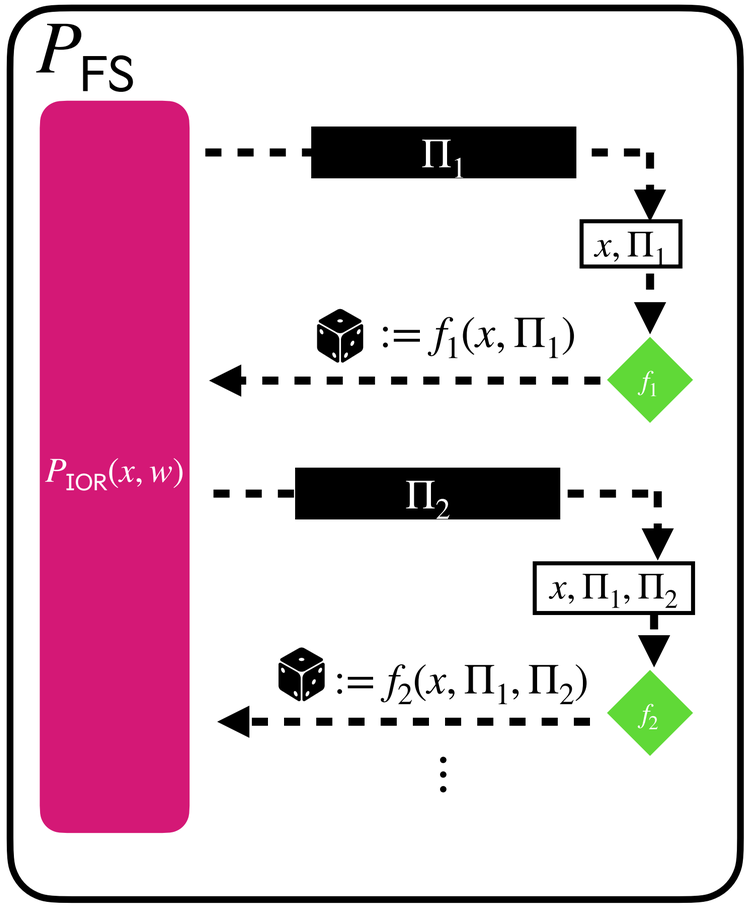
Interactive oracle reduction (IOR) **Interactive**



FS

Non-interactive

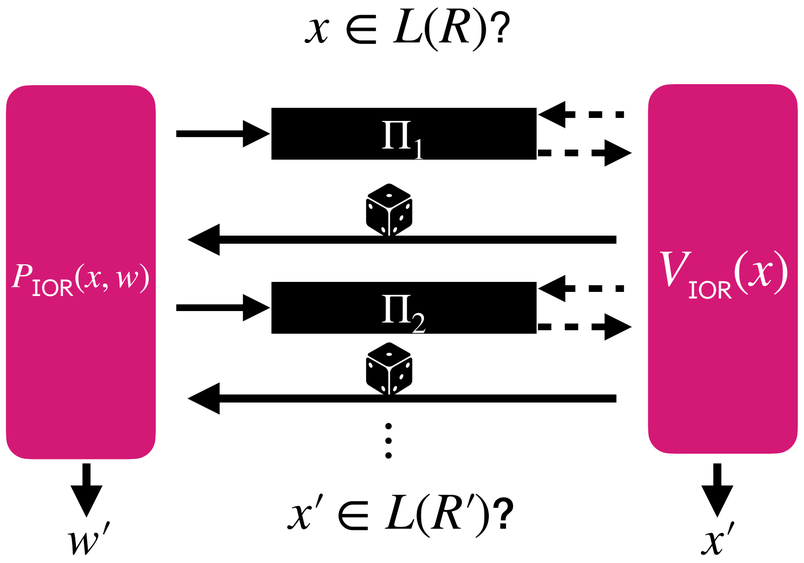
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

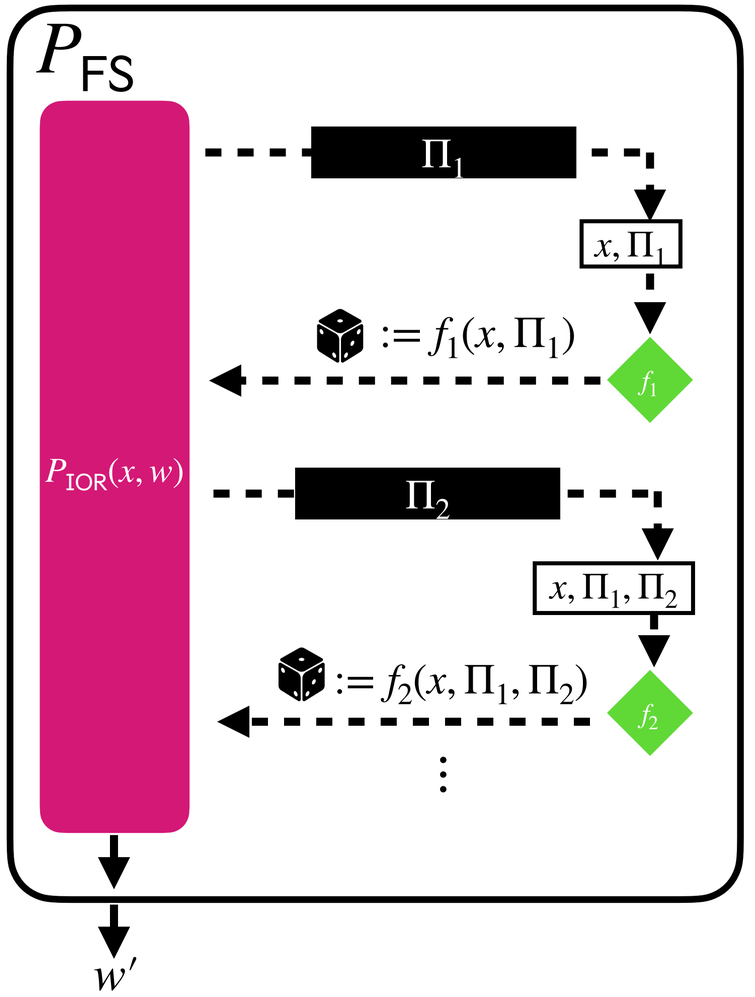
Interactive oracle reduction (IOR) **Interactive**



FS

Non-interactive

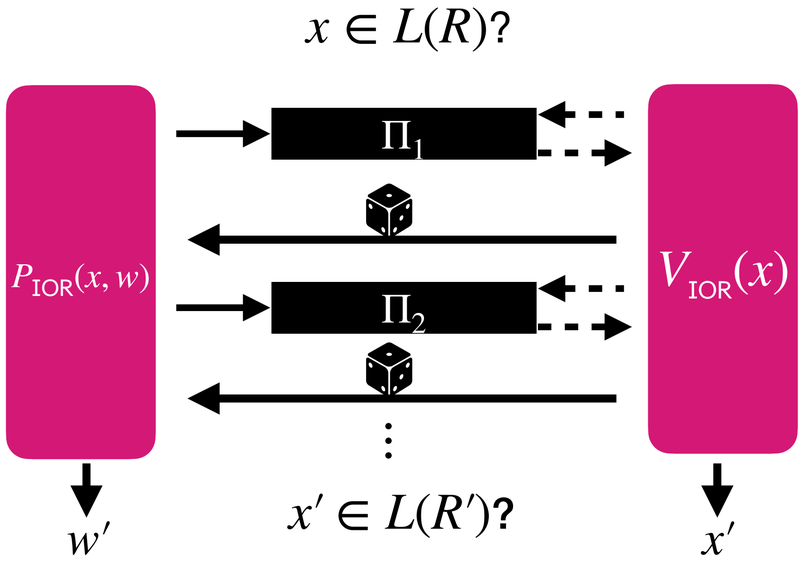
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

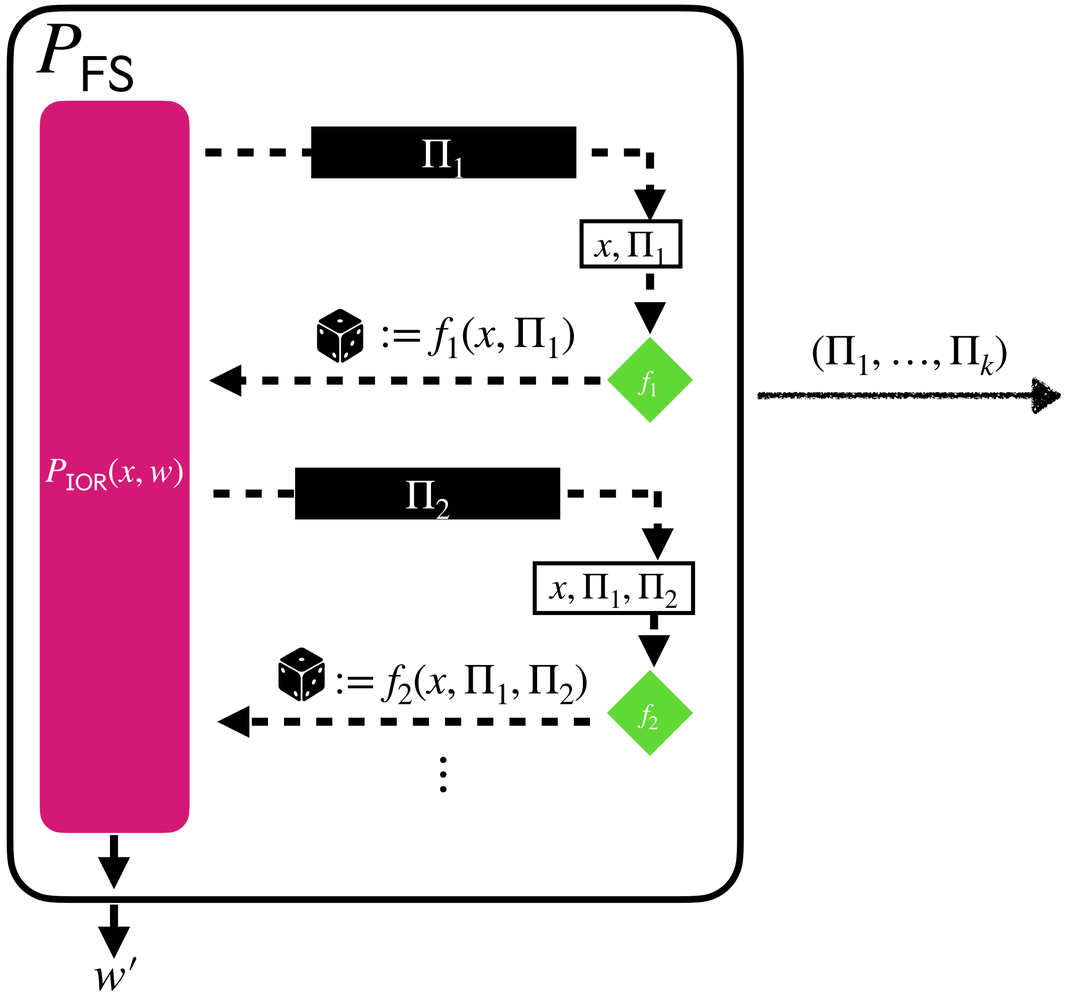
Interactive oracle reduction (IOR) **Interactive**



FS

Non-interactive

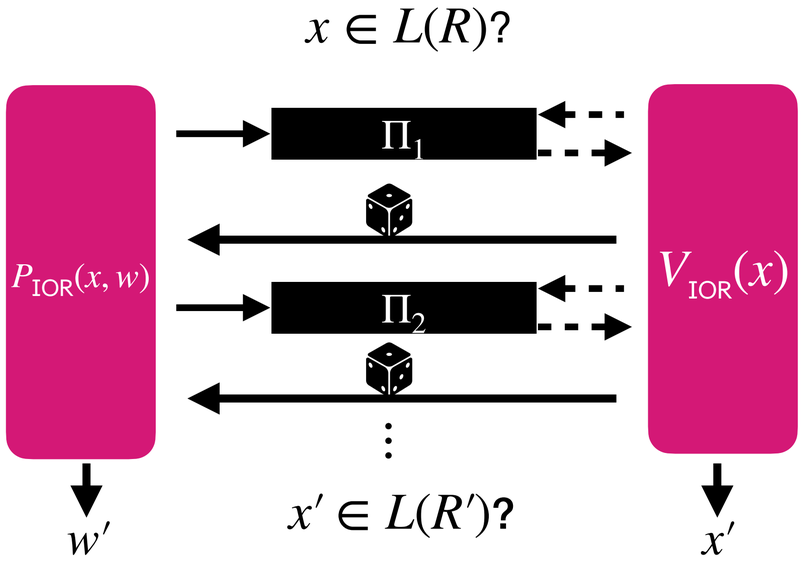
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

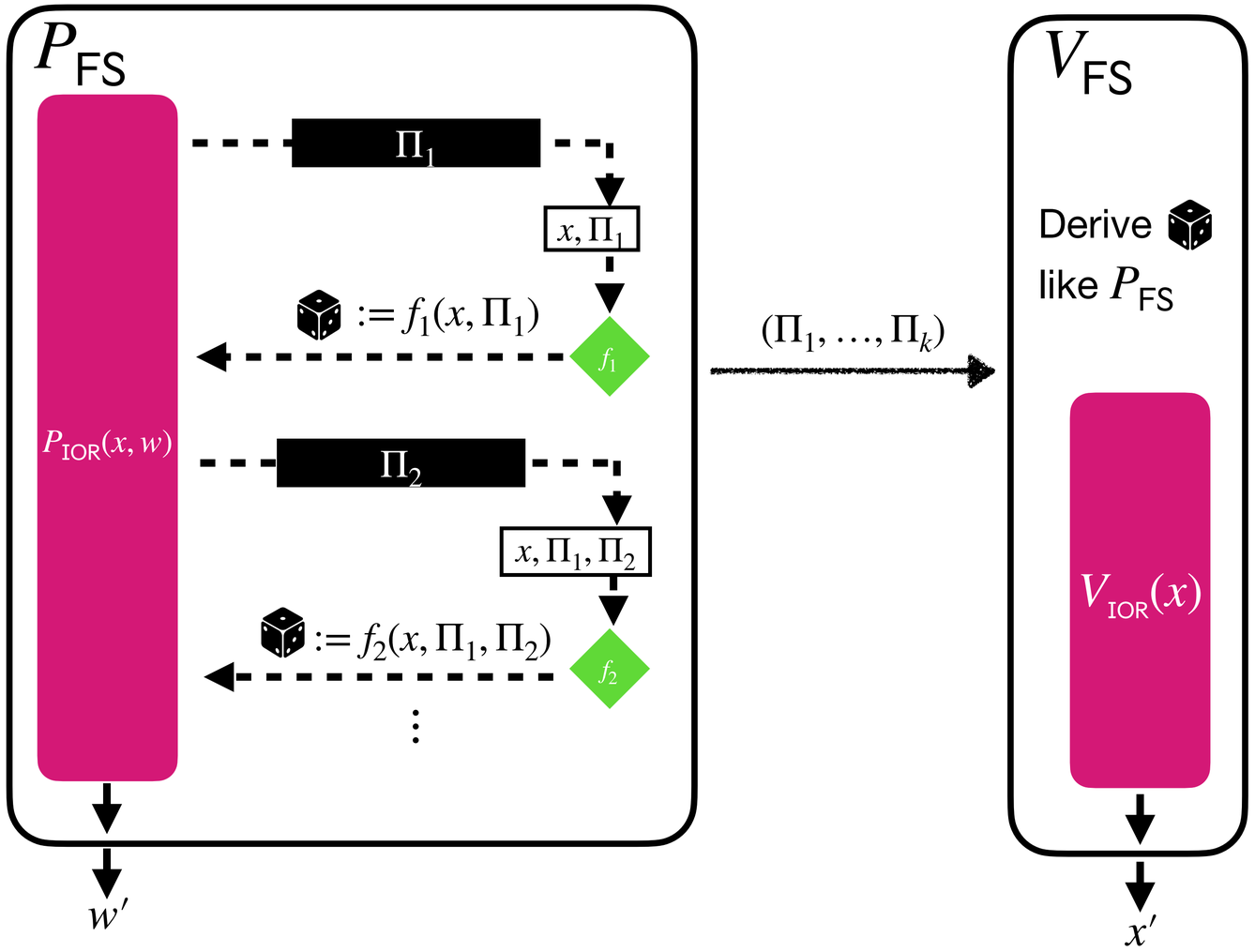
Interactive oracle reduction (IOR) **Interactive**



FS

Non-interactive

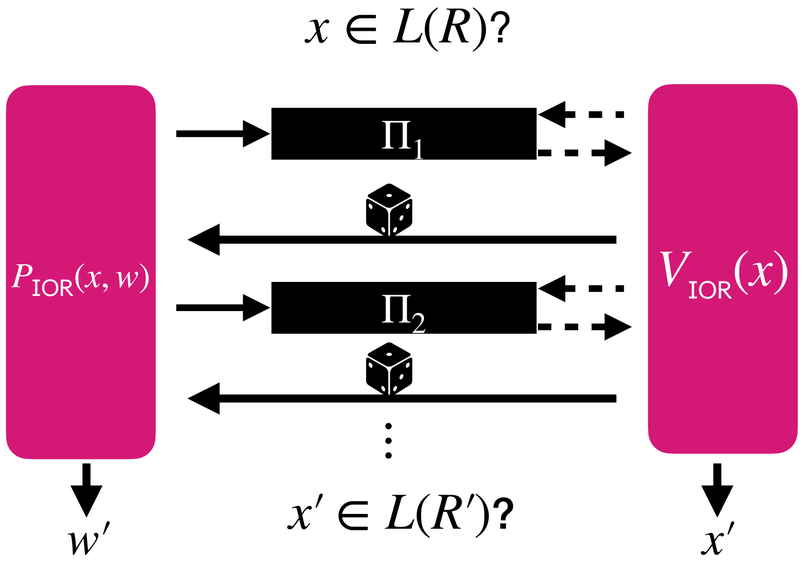
Use random function to derive randomness



Omitted: instances x, x' can also include oracles.

How to remove interaction?

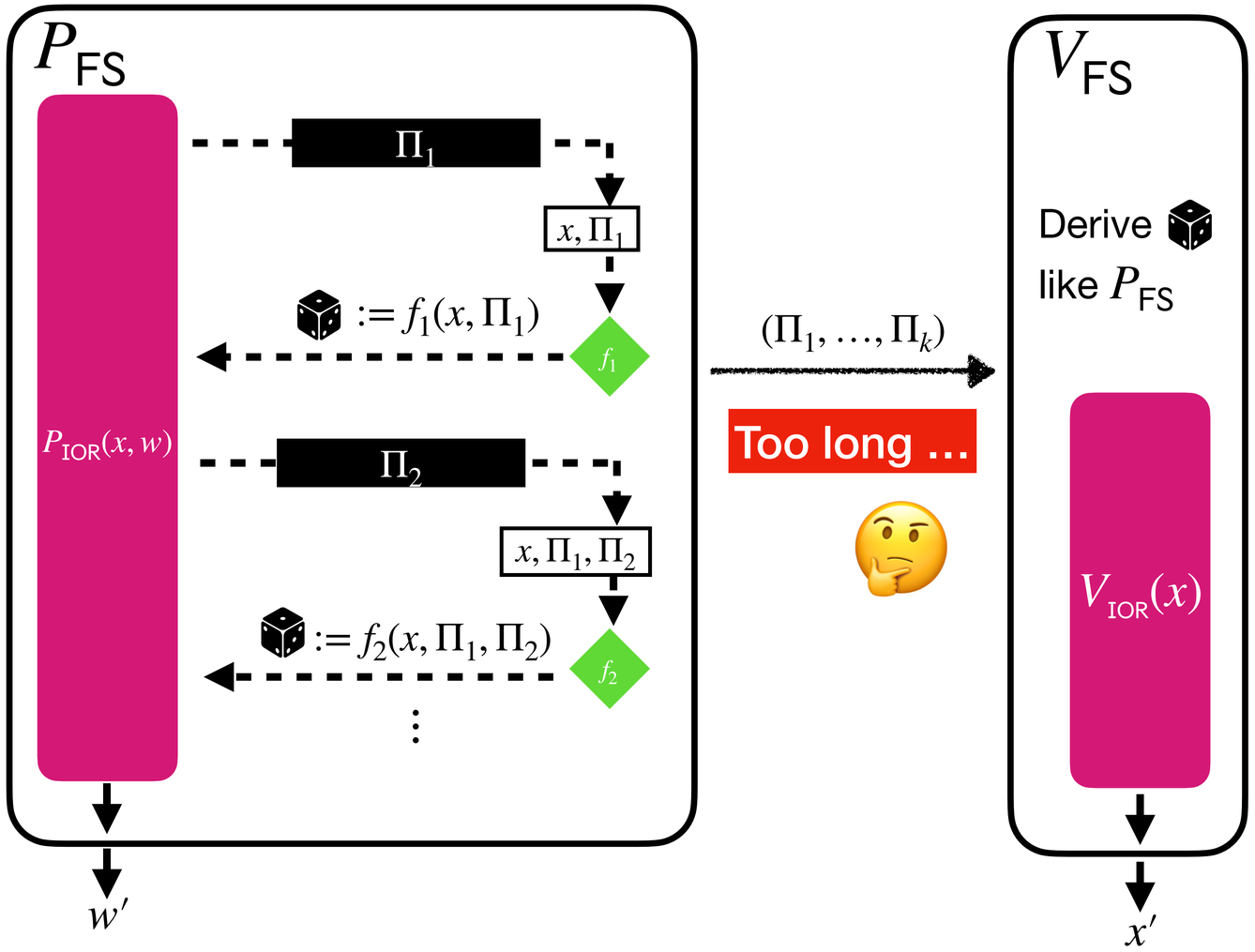
Interactive oracle reduction (IOR) **Interactive**



FS

Non-interactive

Use random function to derive randomness

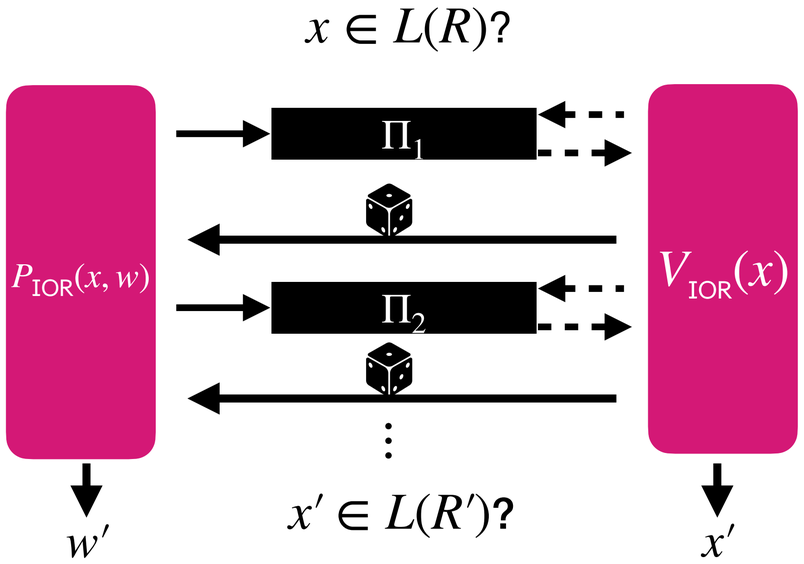


Omitted: instances x, x' can also include oracles.

[BCS16,BMNW25]

Review: the BCS protocol for IOR

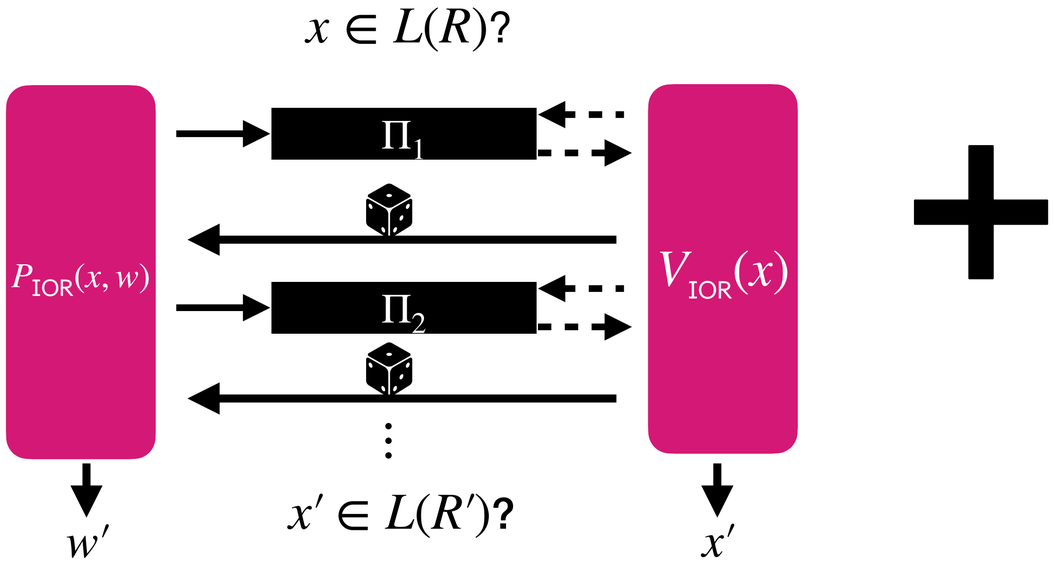
Ingredient #1: Interactive oracle reduction (IOR)



[BCS16,BMNW25]

Review: the BCS protocol for IOR

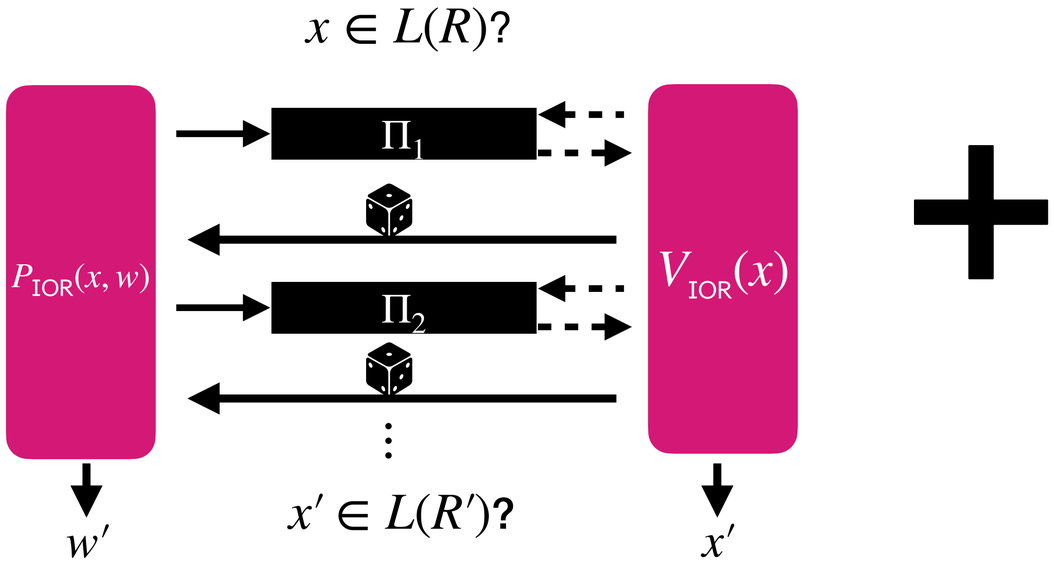
Ingredient #1: Interactive oracle reduction (IOR)



[BCS16,BMNW25]

Review: the BCS protocol for IOR

Ingredient #1: Interactive oracle reduction (IOR)



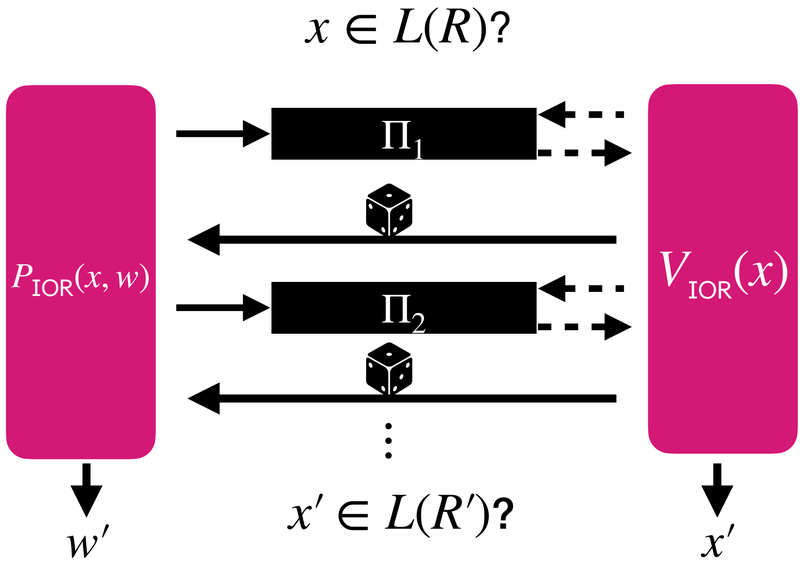
Ingredient #2: Vector commitment scheme (VC)

[BCS16,BMNW25]

Review: the BCS protocol for IOR

an abstraction of MT

Ingredient #1: Interactive oracle reduction (IOR)



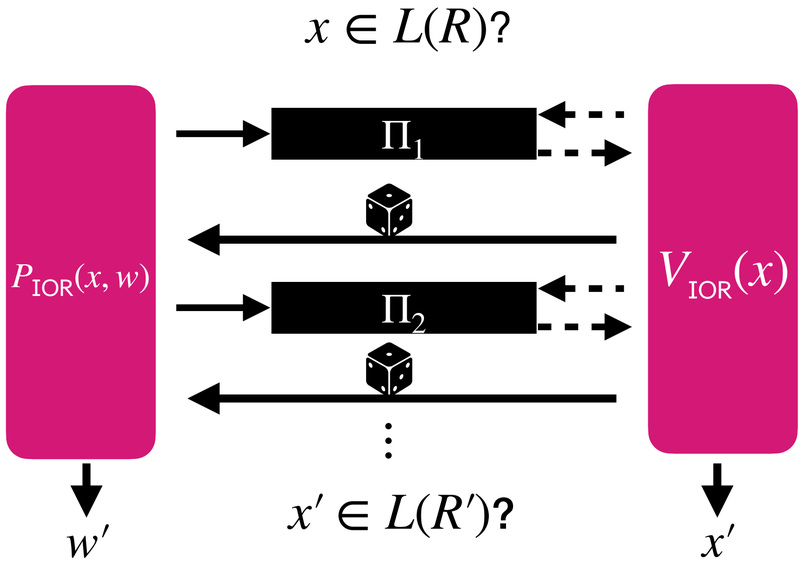
Ingredient #2: Vector commitment scheme (VC)

[BCS16,BMNW25]

Review: the BCS protocol for IOR

an abstraction of MT

Ingredient #1: Interactive oracle reduction (IOR)



+

Ingredient #2: Vector commitment scheme (VC)

VC . Commit — Outputs a commitment cm for Π.

VC . Open — Provides a proof pf for ans $= \Pi[I]$.

VC . Check — Checks if pf is correct. f_{VC}

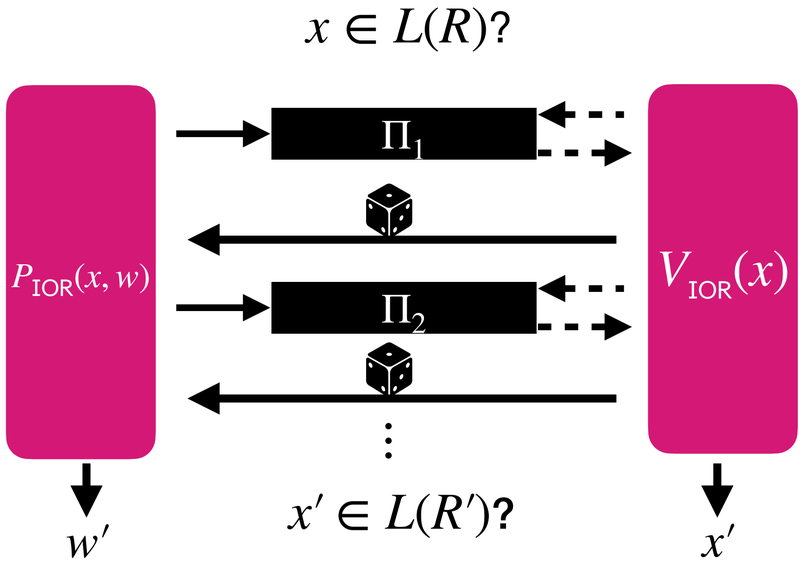
Π :

[BCS16,BMNW25]

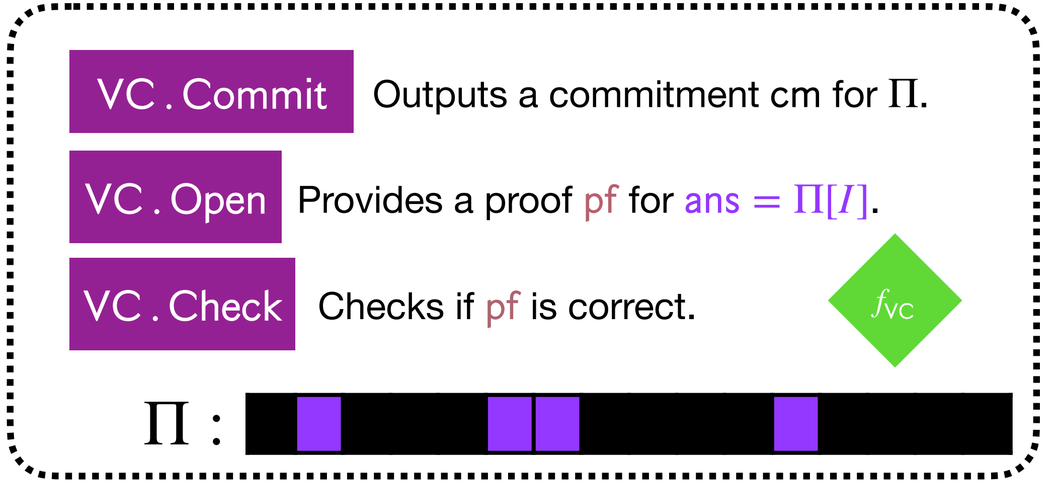
Review: the BCS protocol for IOR

an abstraction of MT

Ingredient #1: Interactive oracle reduction (IOR)



Ingredient #2: Vector commitment scheme (VC)

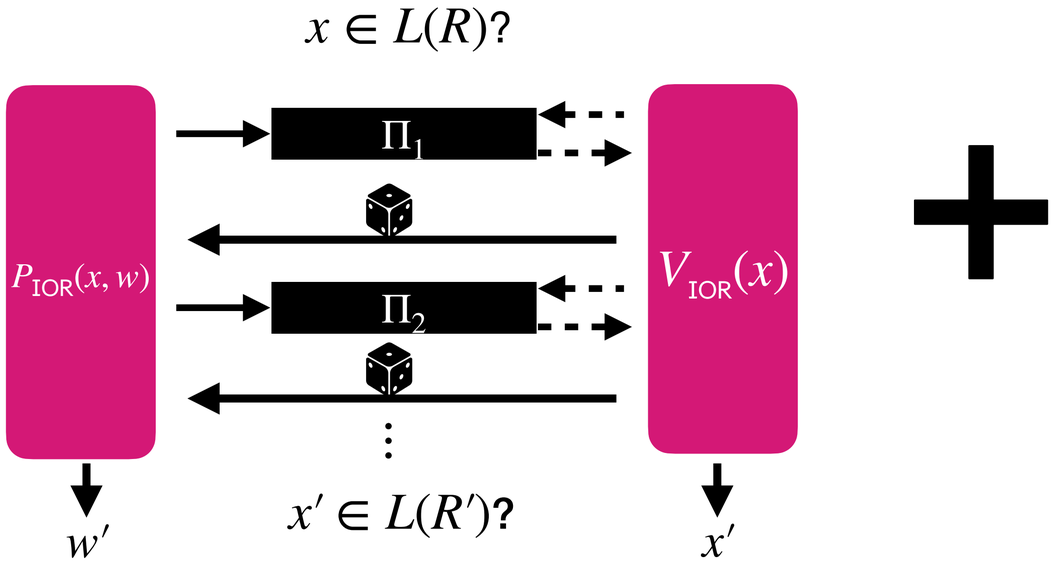


[BCS16,BMNW25]

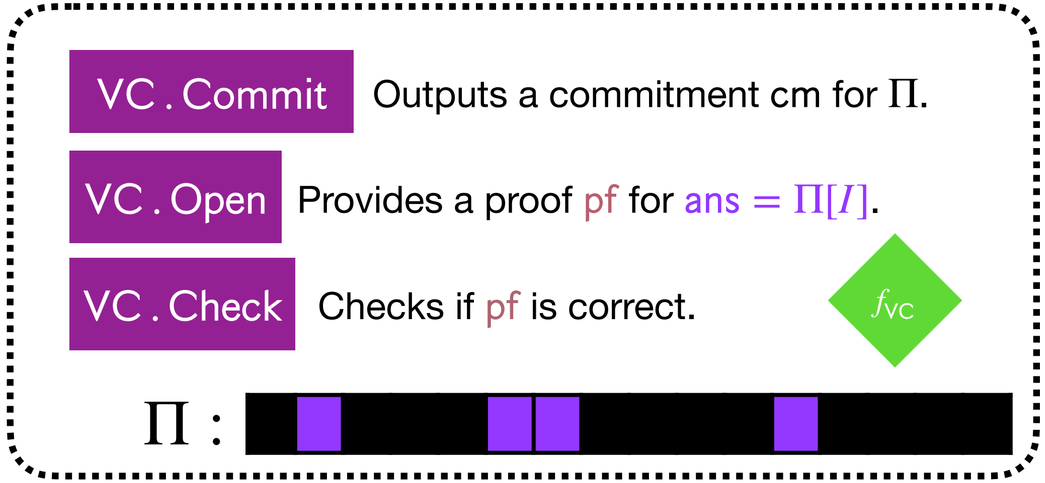
Review: the BCS protocol for IOR

an abstraction of MT

Ingredient #1: Interactive oracle reduction (IOR)



Ingredient #2: Vector commitment scheme (VC)

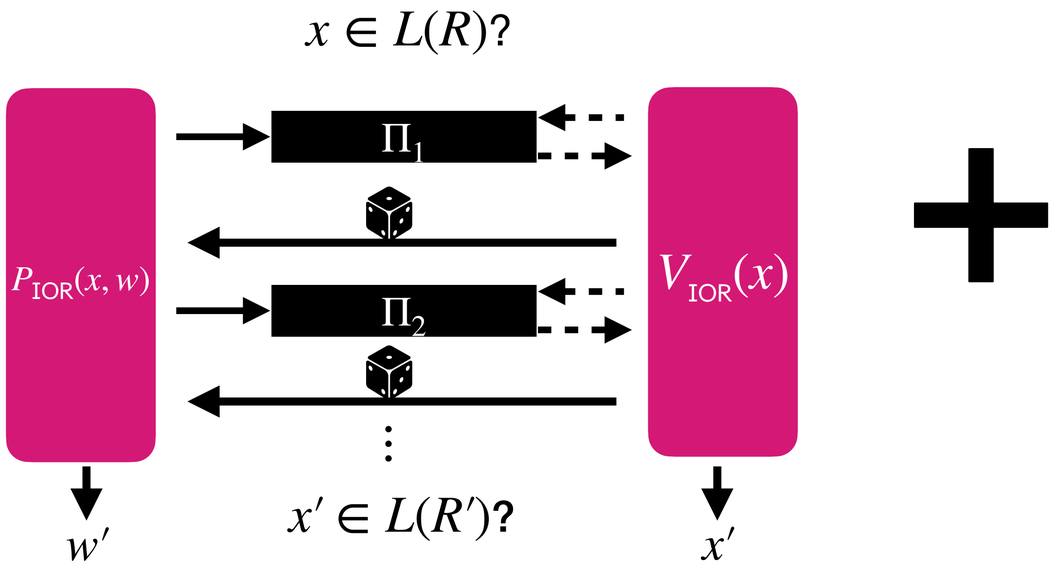


[BCS16,BMNW25]

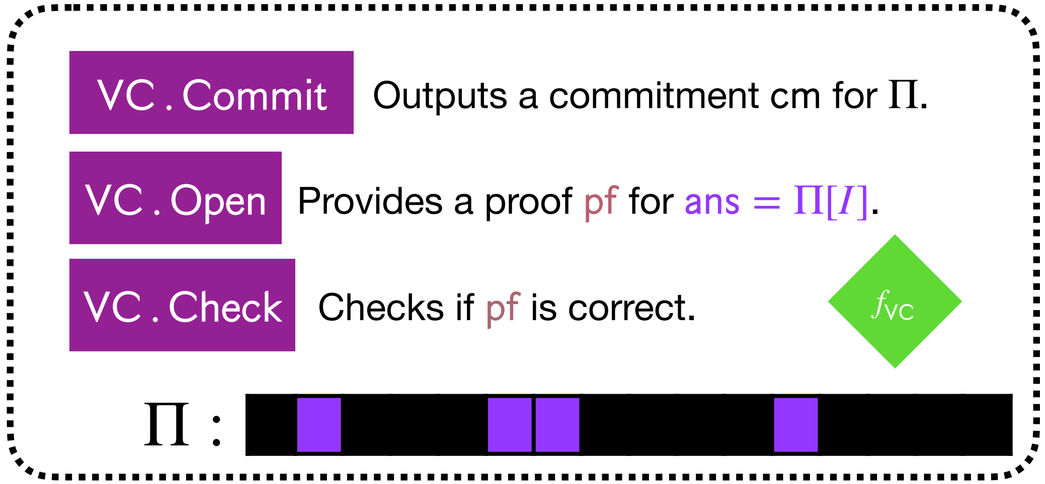
Review: the BCS protocol for IOR

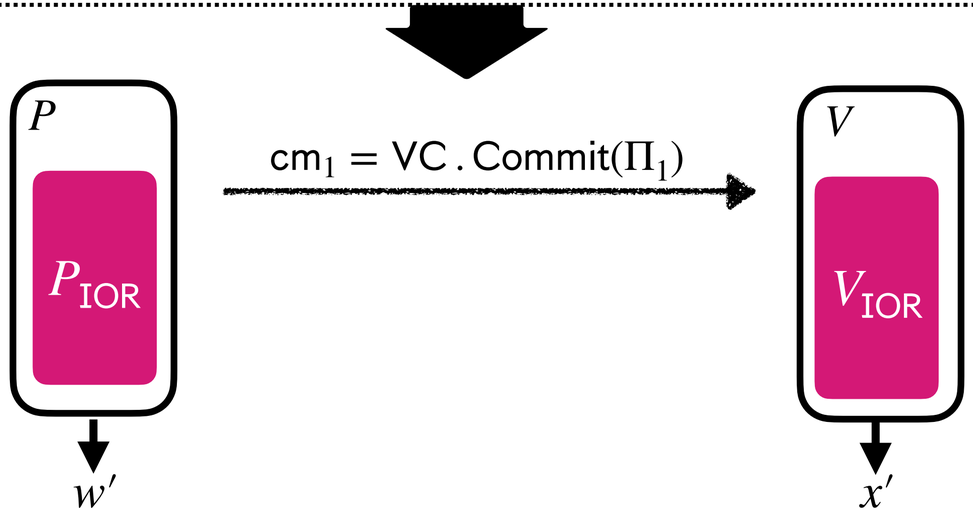
an abstraction of MT

Ingredient #1: Interactive oracle reduction (IOR)



Ingredient #2: Vector commitment scheme (VC)



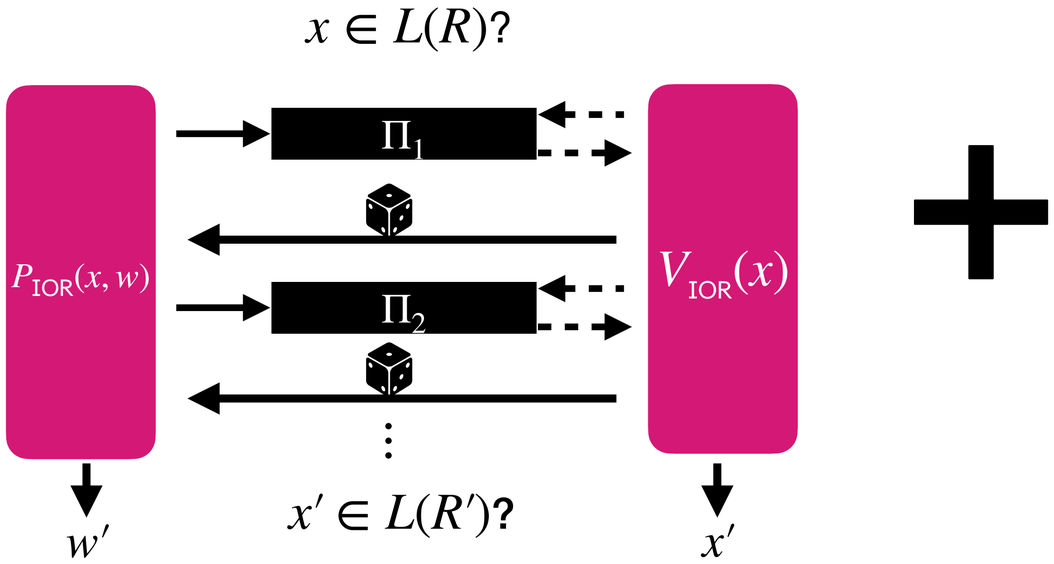


[BCS16,BMNW25]

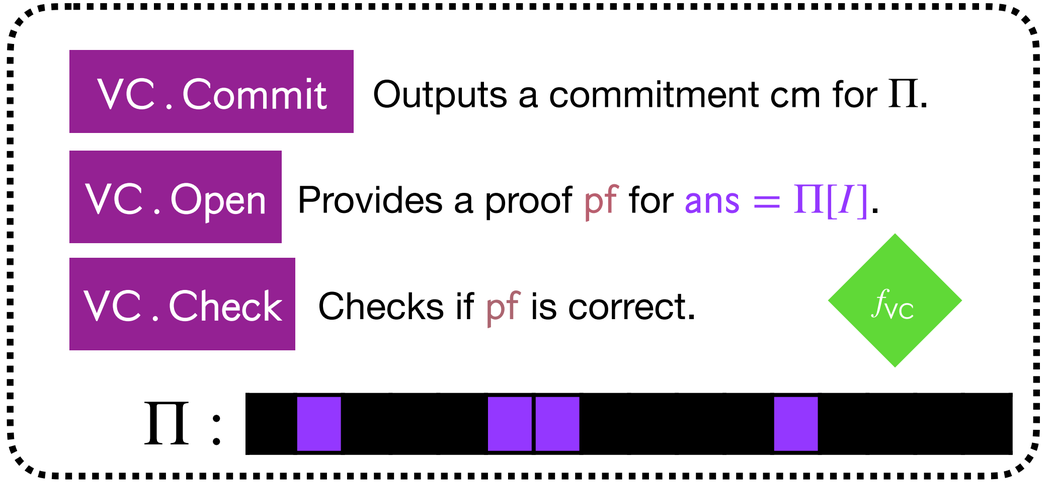
Review: the BCS protocol for IOR

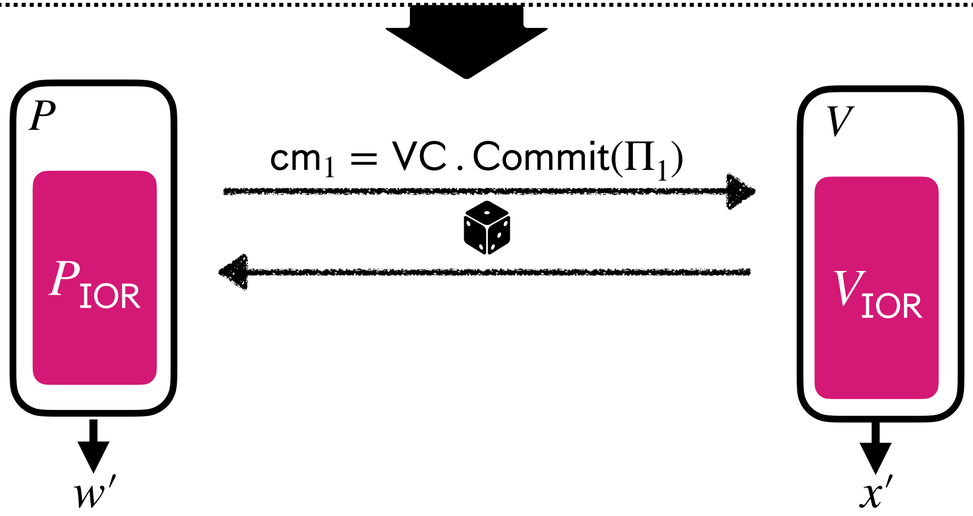
an abstraction of MT

Ingredient #1: Interactive oracle reduction (IOR)



Ingredient #2: Vector commitment scheme (VC)



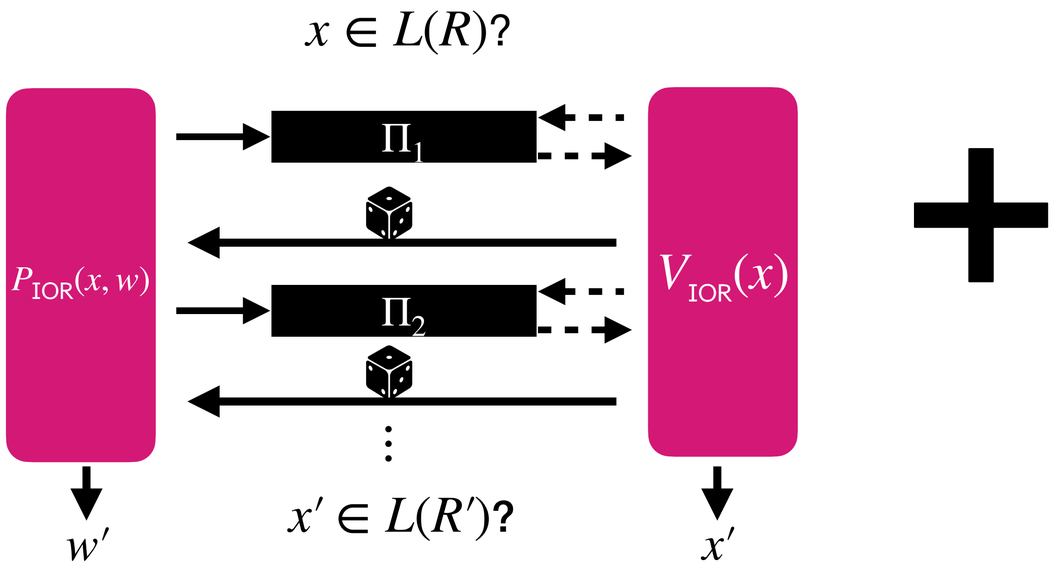


[BCS16,BMNW25]

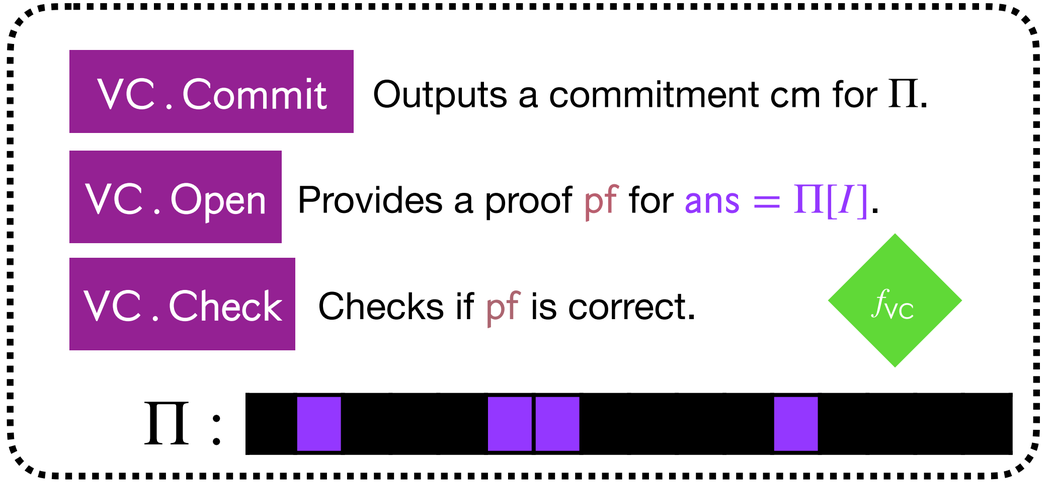
Review: the BCS protocol for IOR

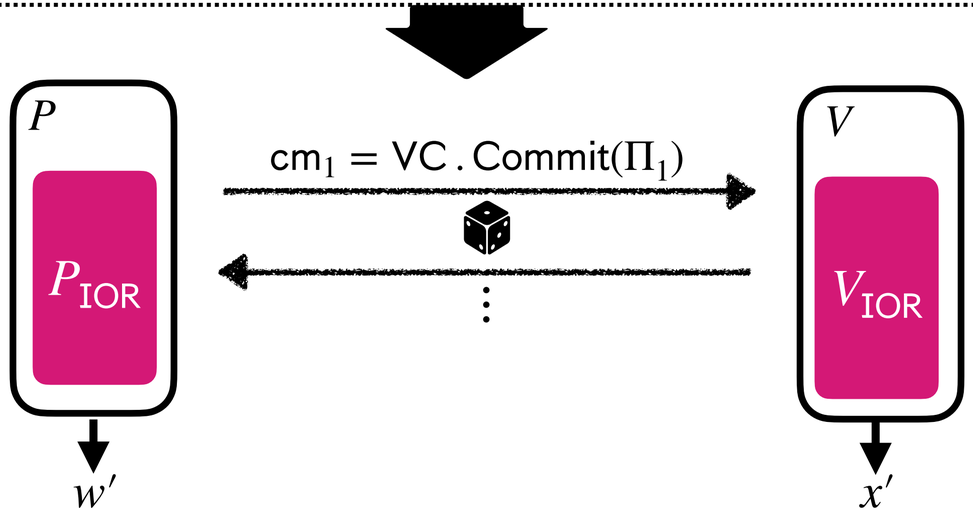
an abstraction of MT

Ingredient #1: Interactive oracle reduction (IOR)



Ingredient #2: Vector commitment scheme (VC)



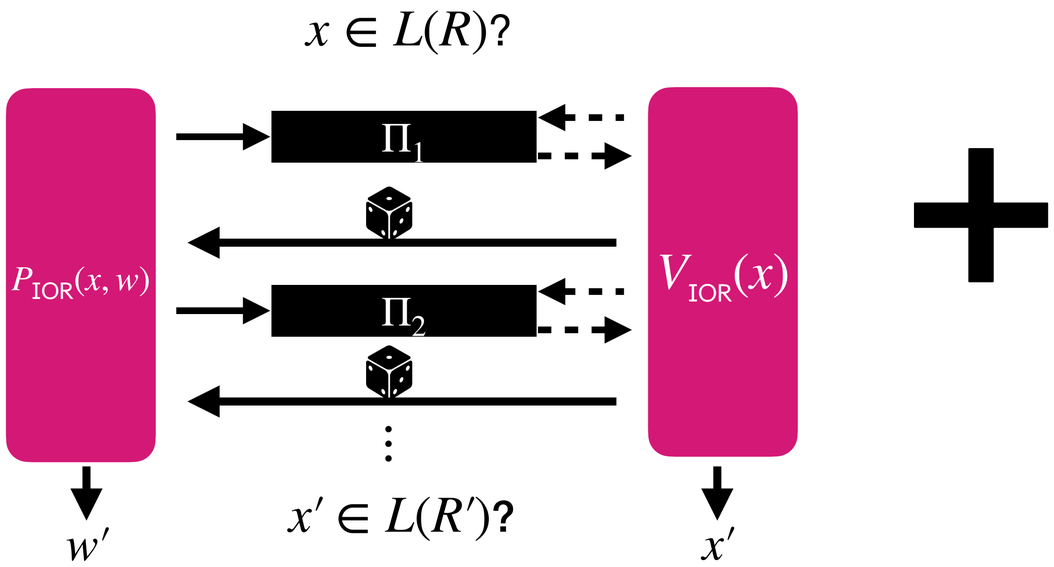


[BCS16,BMNW25]

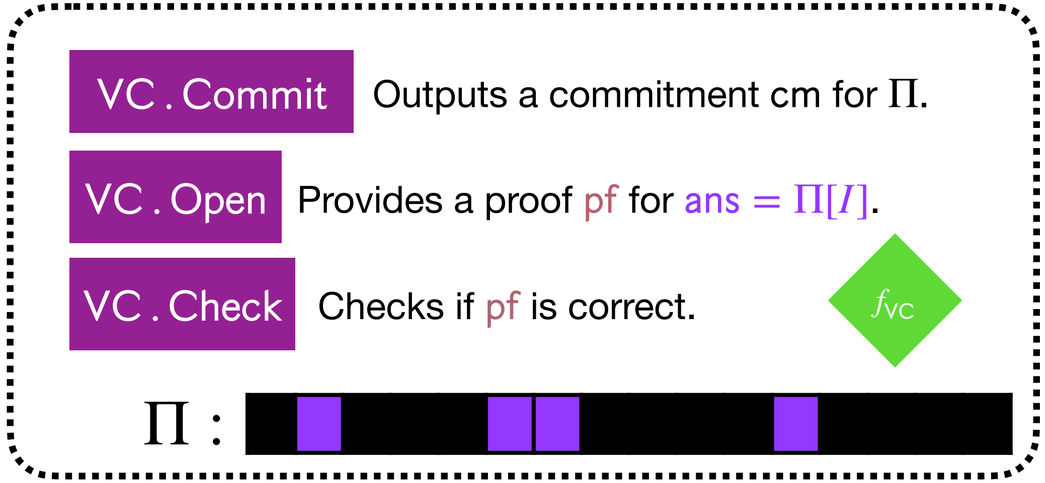
Review: the BCS protocol for IOR

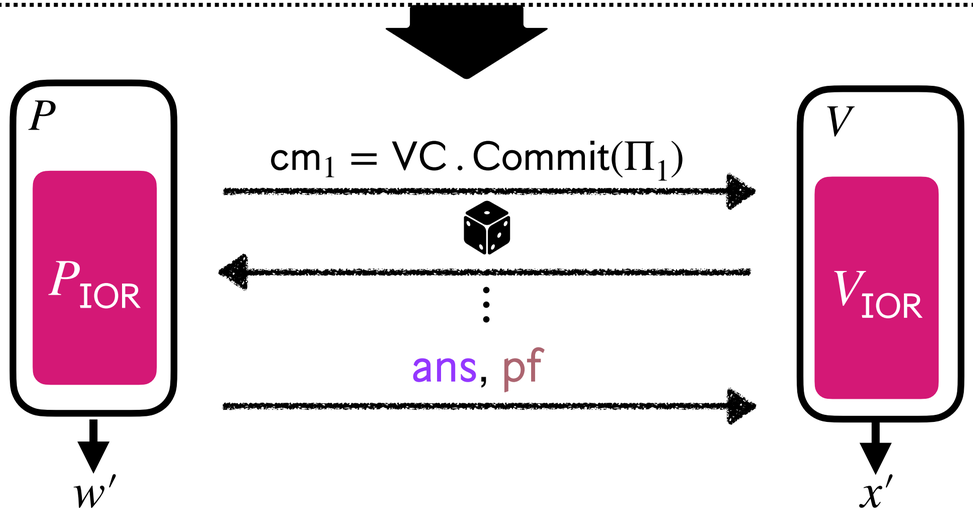
an abstraction of MT

Ingredient #1: Interactive oracle reduction (IOR)



Ingredient #2: Vector commitment scheme (VC)



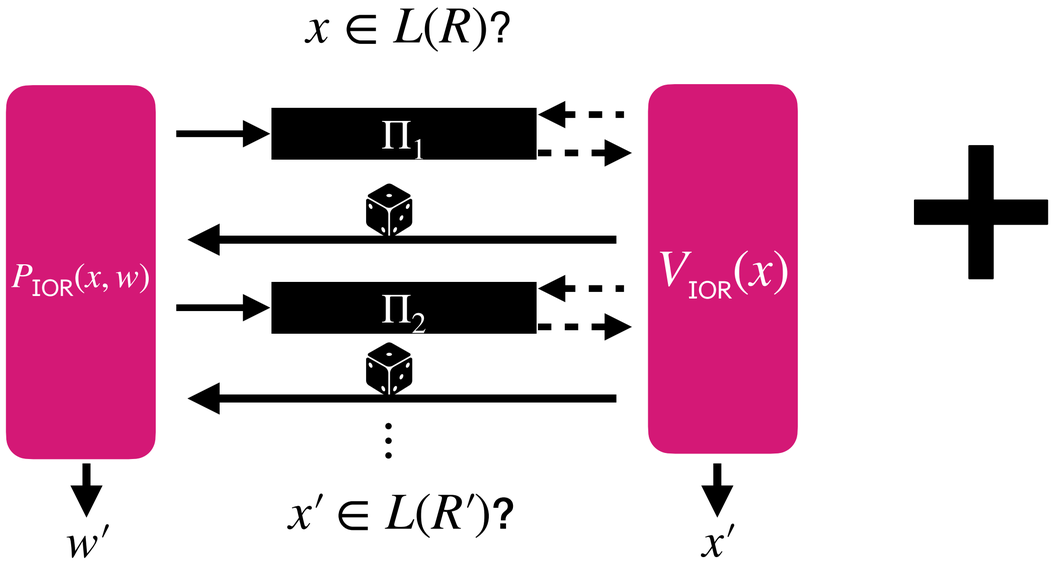


[BCS16,BMNW25]

Review: the BCS protocol for IOR

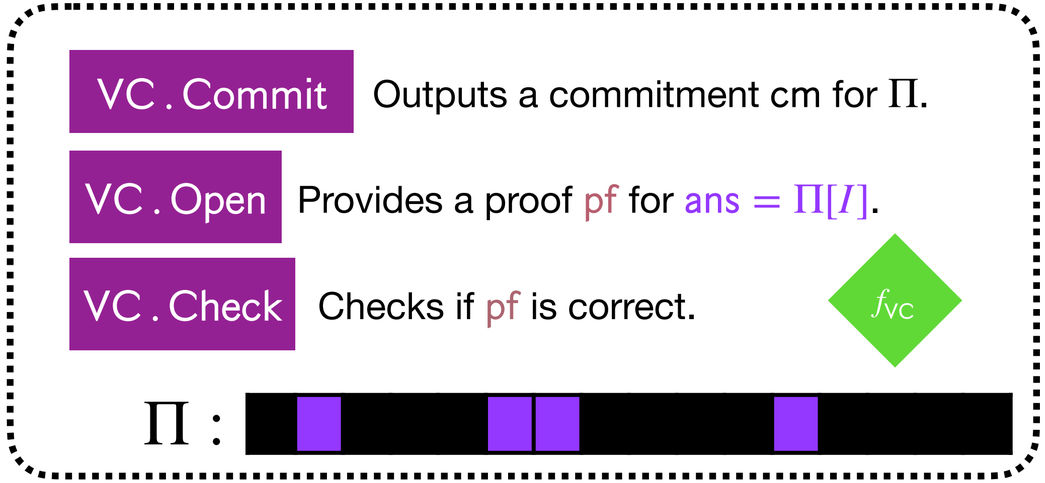
an abstraction of MT

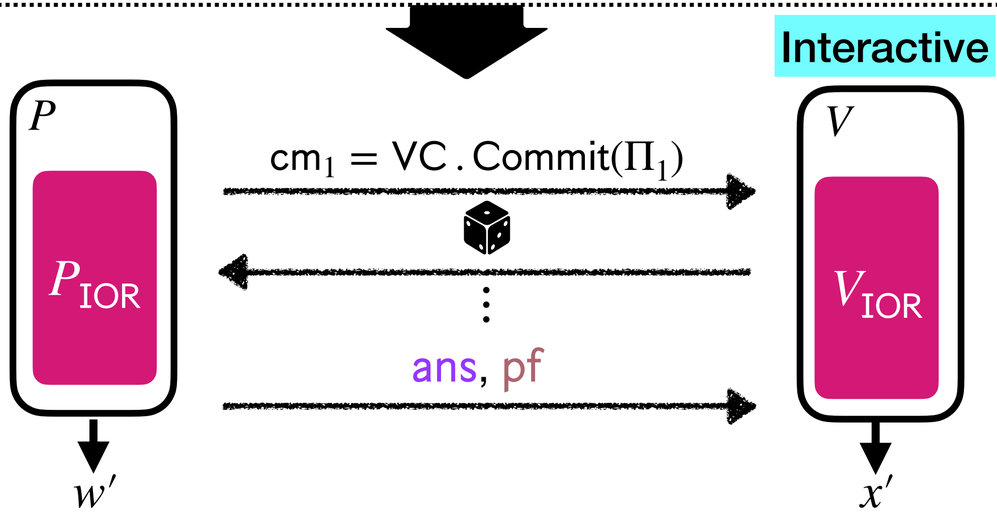
Ingredient #1: Interactive oracle reduction (IOR)



+

Ingredient #2: Vector commitment scheme (VC)



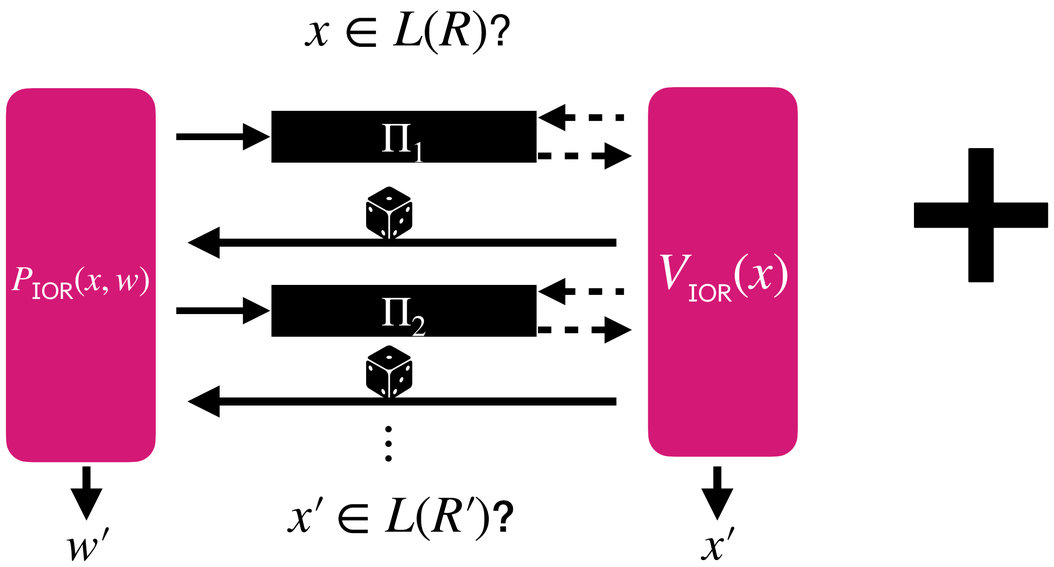


[BCS16,BMNW25]

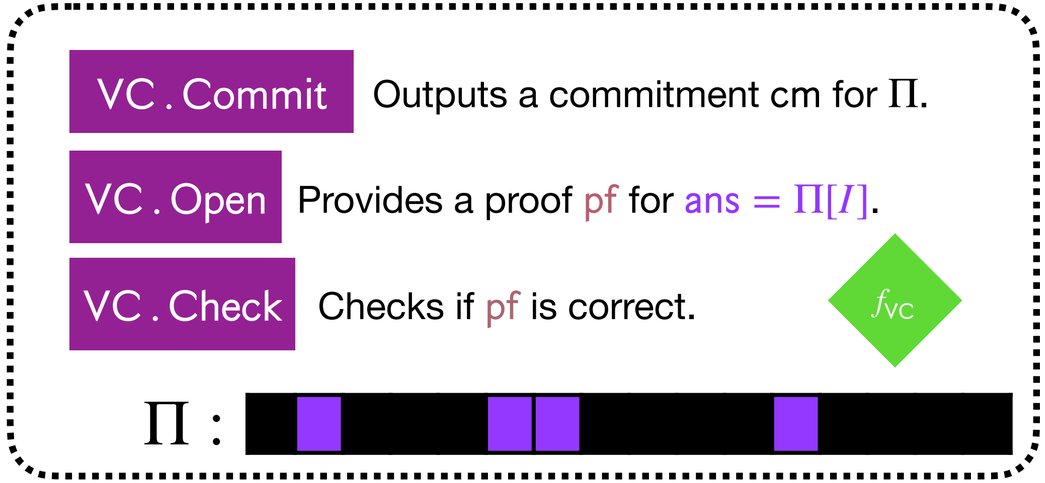
Review: the BCS protocol for IOR

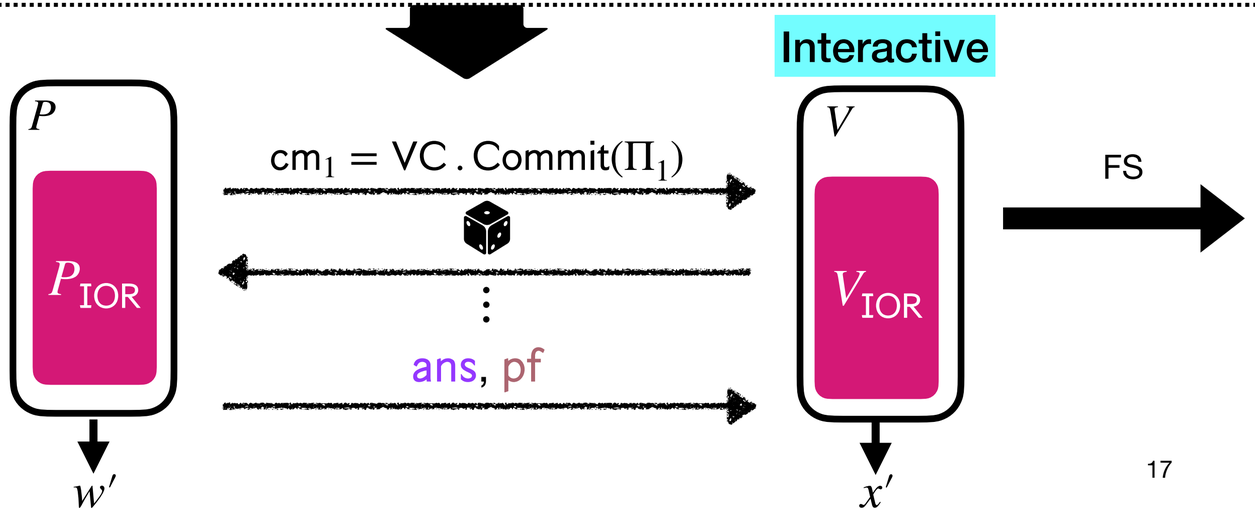
an abstraction of MT

Ingredient #1: Interactive oracle reduction (IOR)



Ingredient #2: Vector commitment scheme (VC)





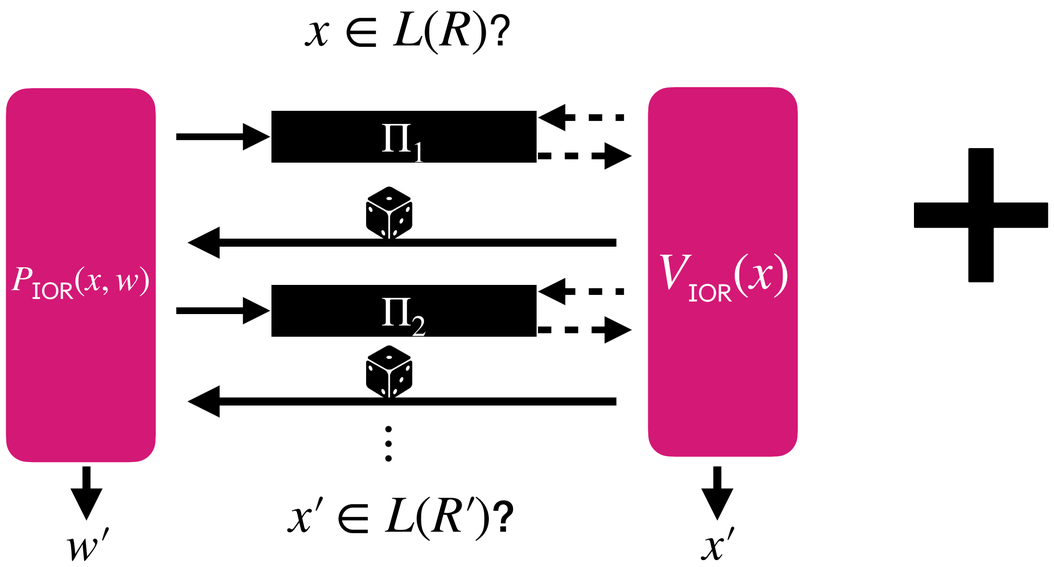
FS

[BCS16,BMNW25]

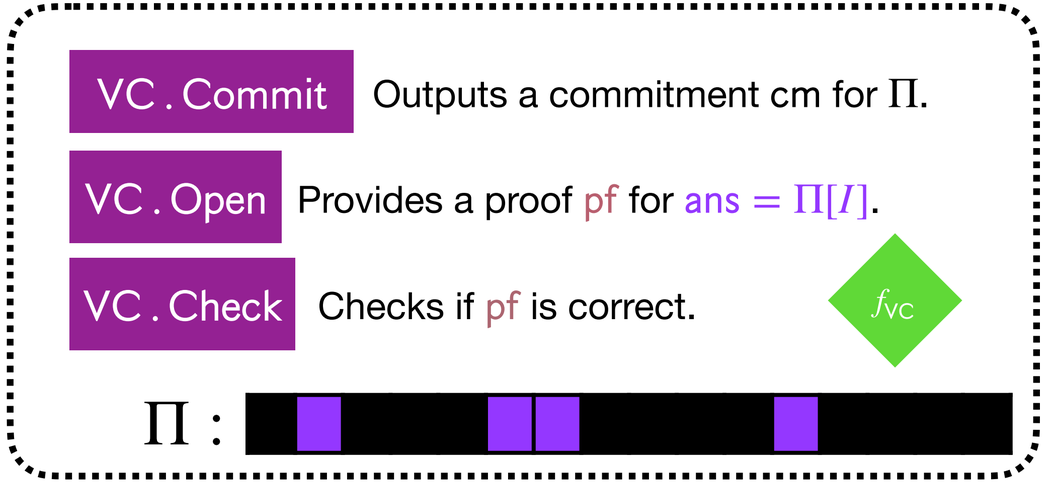
Review: the BCS protocol for IOR

an abstraction of MT

Ingredient #1: Interactive oracle reduction (IOR)

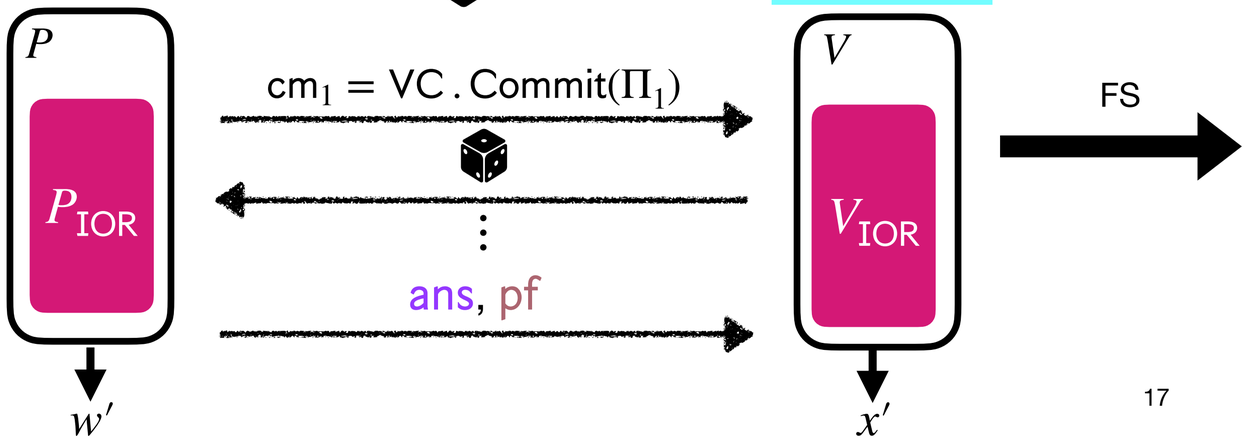


Ingredient #2: Vector commitment scheme (VC)



Interactive

Non-interactive



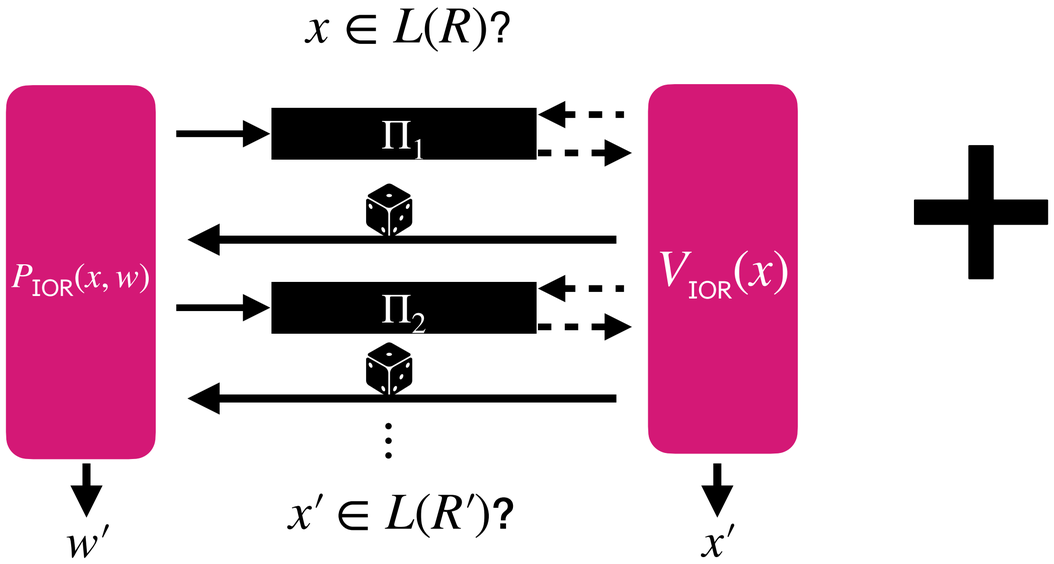
FS

[BCS16,BMNW25]

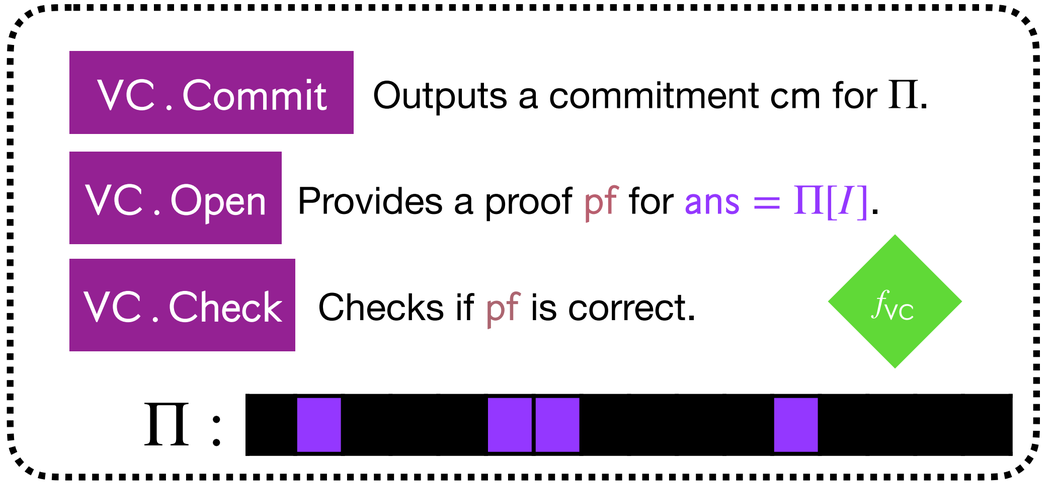
Review: the BCS protocol for IOR

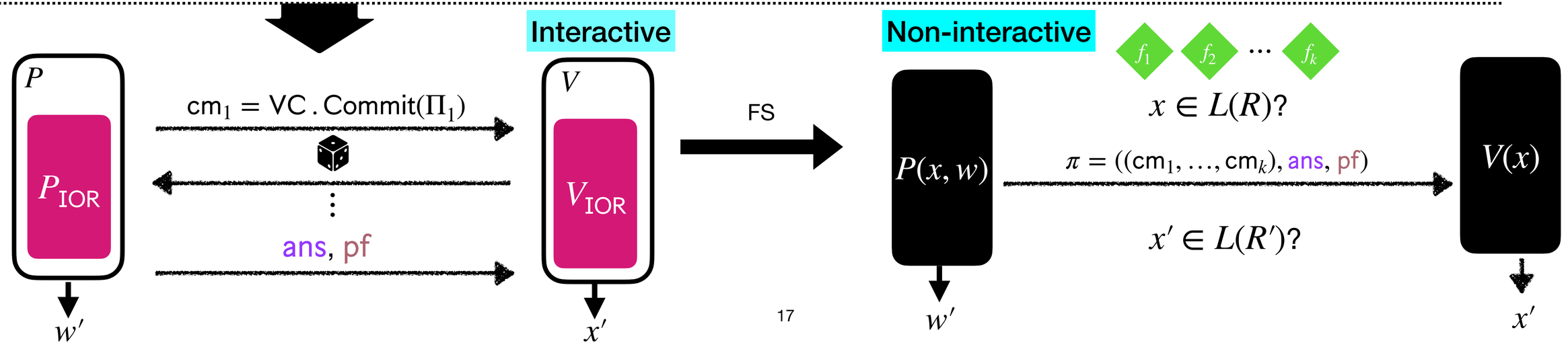
an abstraction of MT

Ingredient #1: Interactive oracle reduction (IOR)



Ingredient #2: Vector commitment scheme (VC)



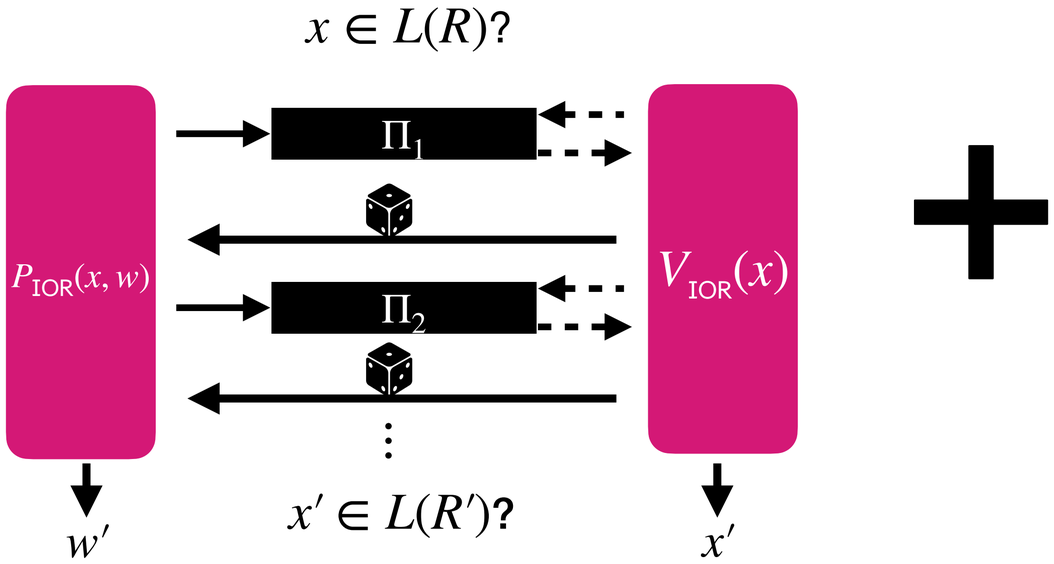


[BCS16,BMNW25]

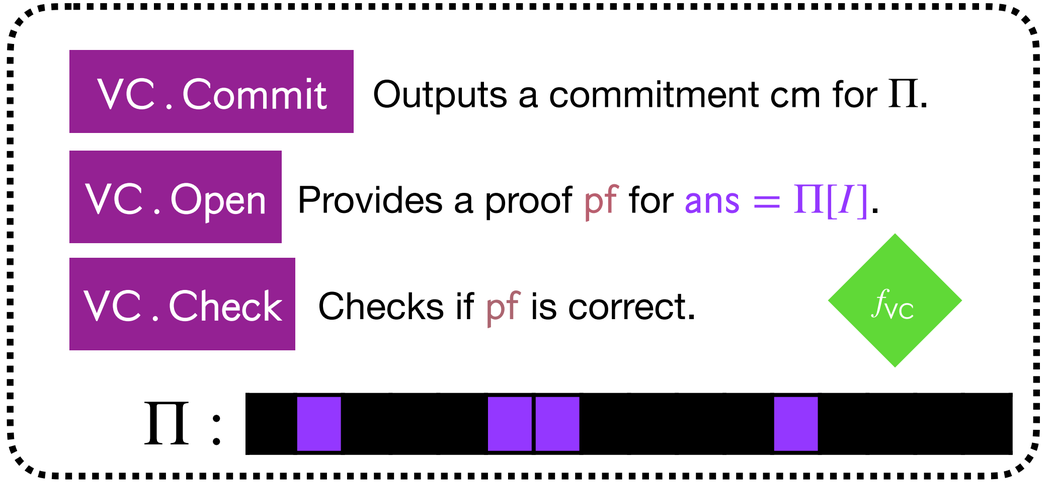
Review: the BCS protocol for IOR

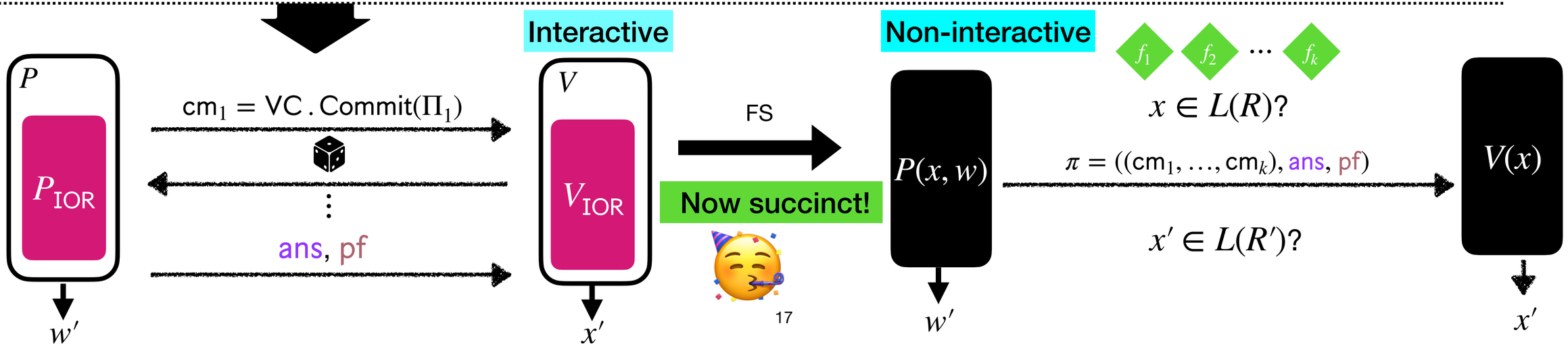
an abstraction of MT

Ingredient #1: Interactive oracle reduction (IOR)



Ingredient #2: Vector commitment scheme (VC)

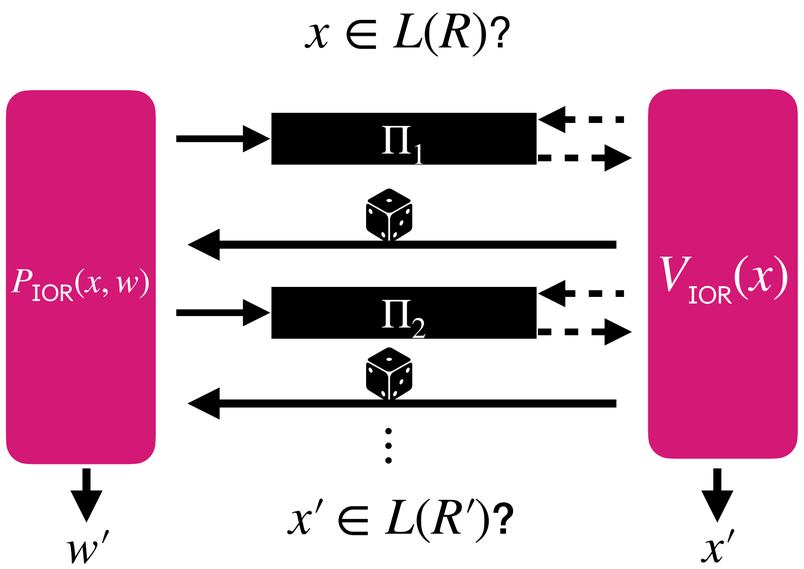




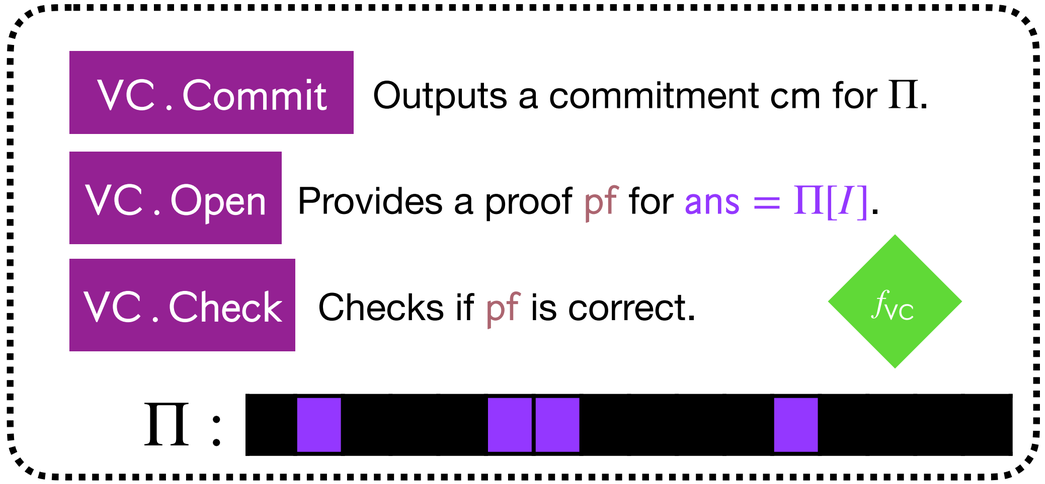
[BCS16,BMNW25]

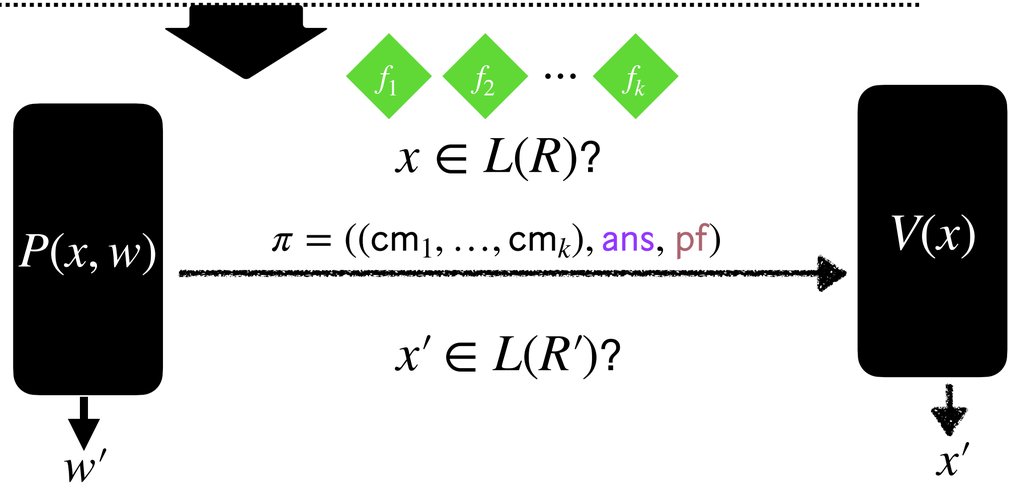
BCS[IOR, VC]

Ingredient #1: Interactive oracle reduction (IOR)



Ingredient #2: Vector commitment scheme (VC)

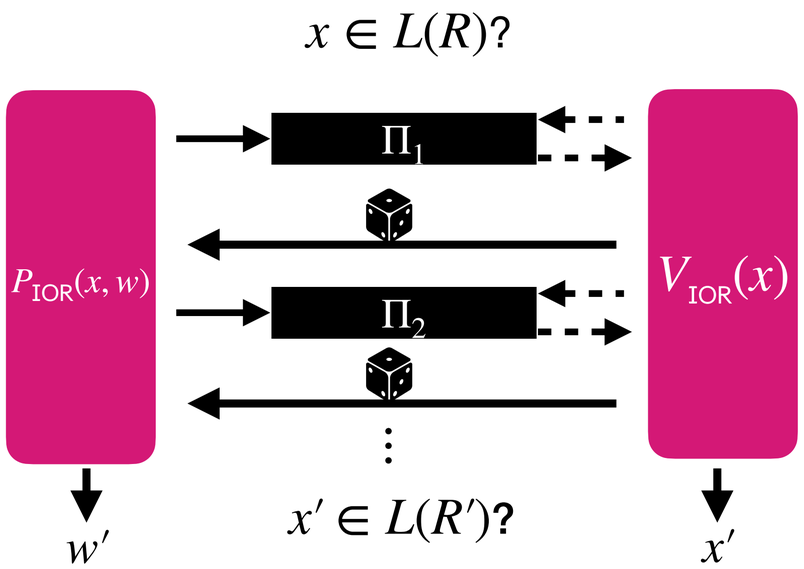




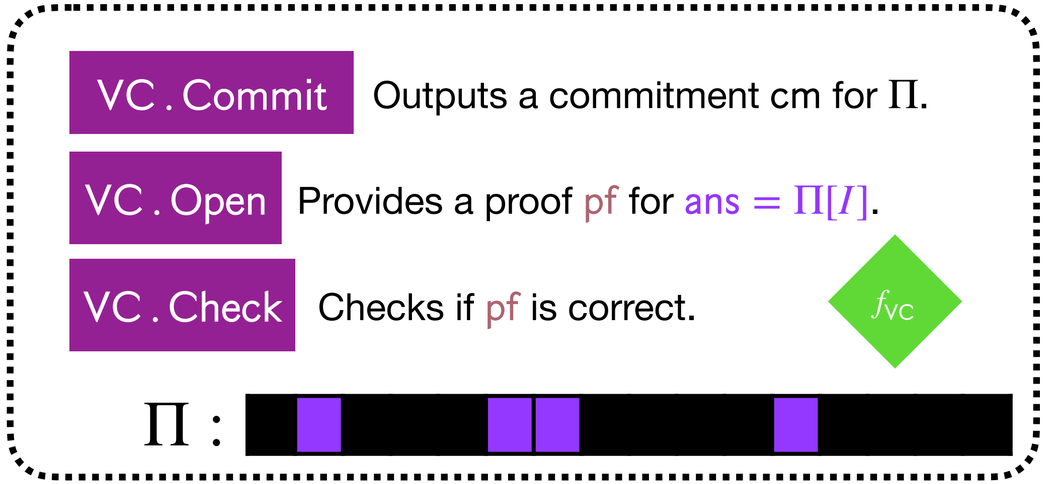
[BCS16,BMNW25]

BCS[IOR, VC]

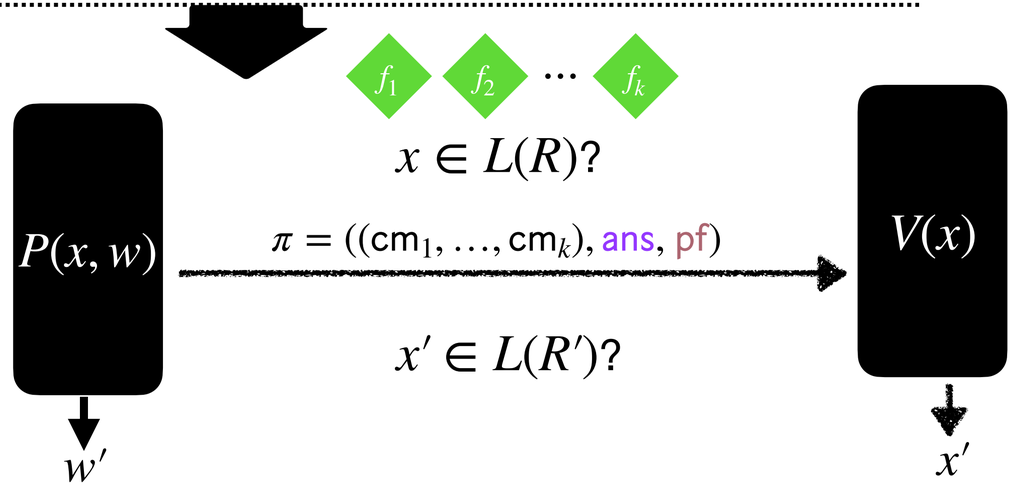
Ingredient #1: Interactive oracle reduction (IOR)



Ingredient #2: Vector commitment scheme (VC)



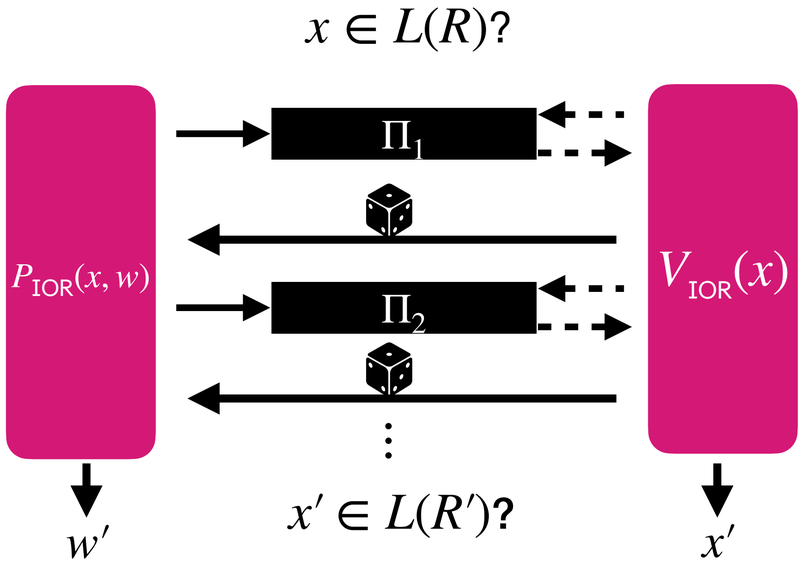
Two potential attacks to **BCS[IOR, VC]**:



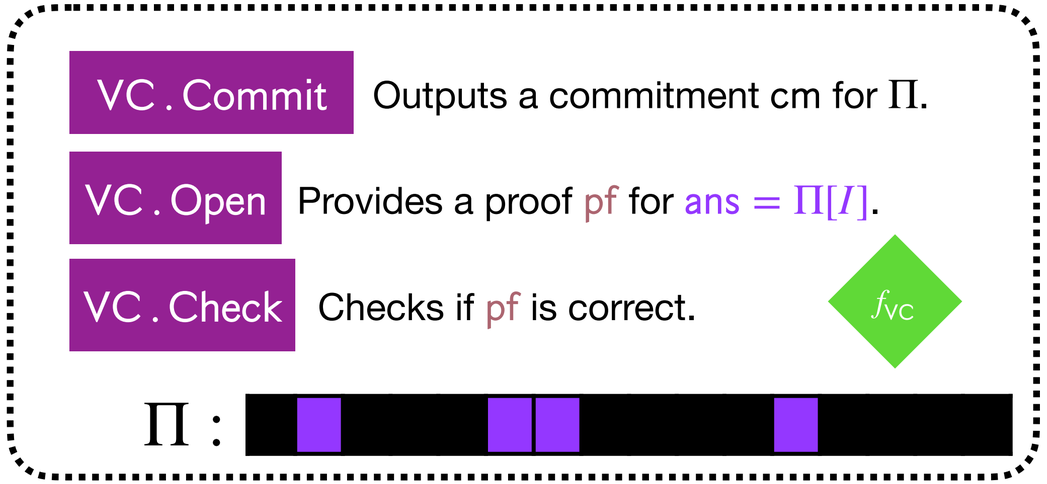
[BCS16,BMNW25]

BCS[IOR, VC]

Ingredient #1: Interactive oracle reduction (IOR)

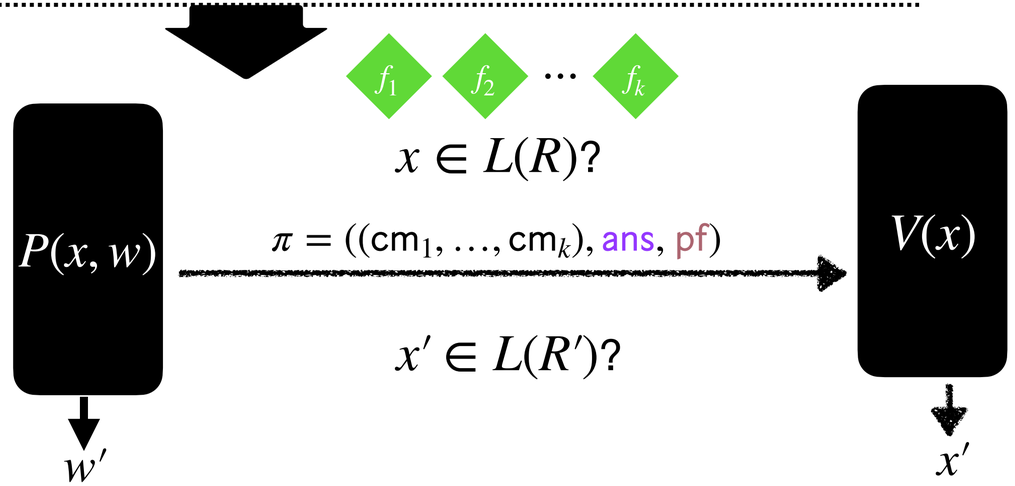


Ingredient #2: Vector commitment scheme (VC)



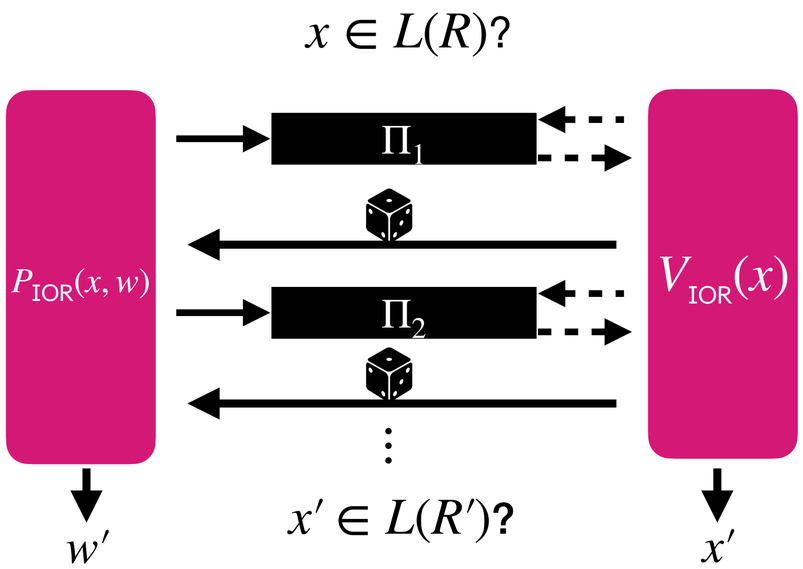
Two potential attacks to **BCS[IOR, VC]**:

1. $\tilde{P}$ can query f_1 f_2 $\cdots$ f_k many times to get 🎲 that makes V_{IOR} output $x' \in L(R')$.

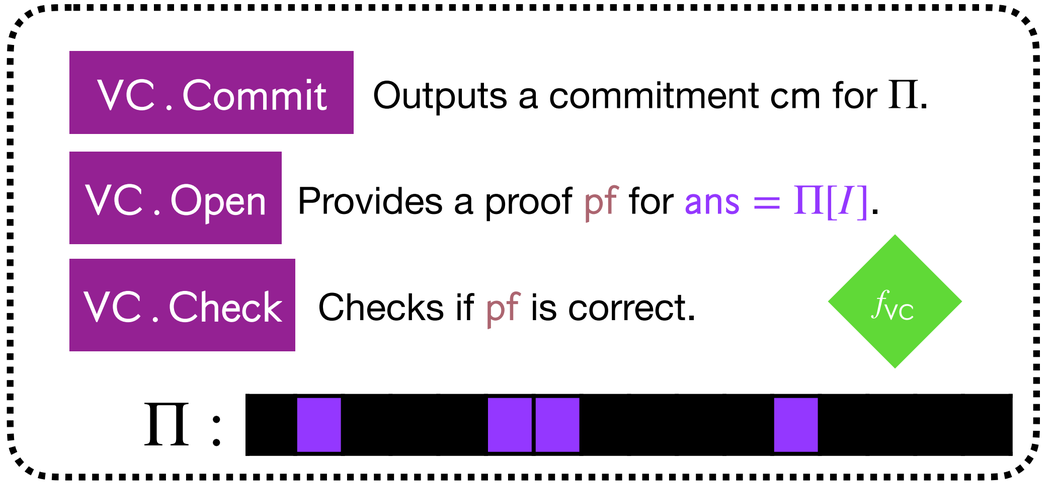


BCS[IOR, VC]

Ingredient #1: Interactive oracle reduction (IOR)

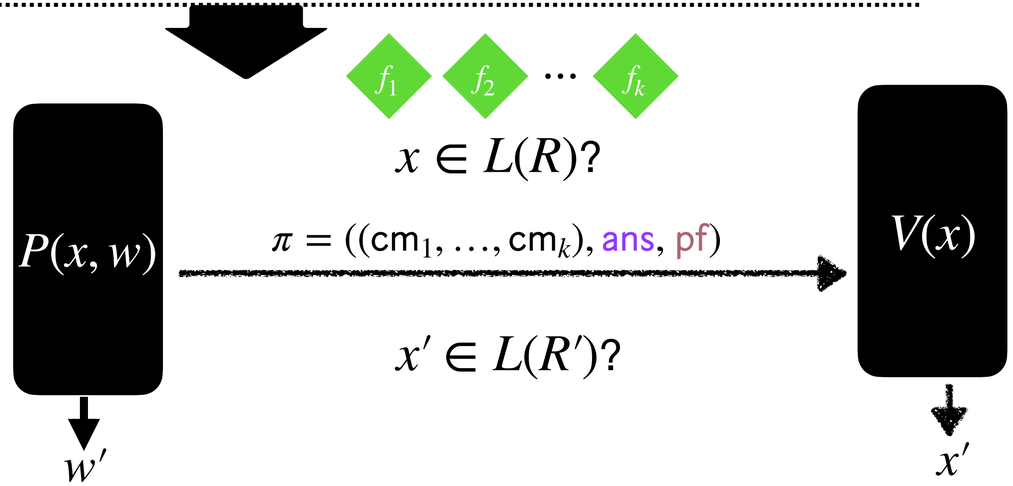


Ingredient #2: Vector commitment scheme (VC)



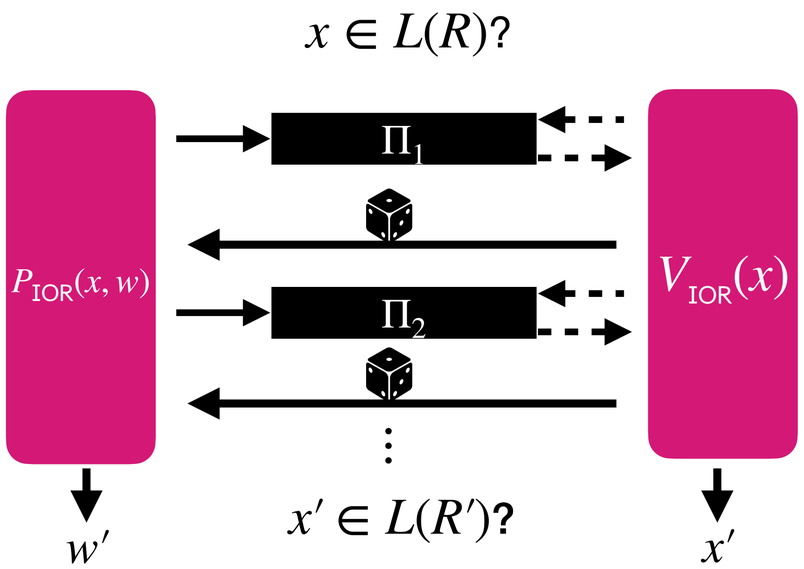
Two potential attacks to **BCS[IOR, VC]**:

1. $\tilde{P}$ can query f_1 f_2 $\cdots$ f_k many times to get 🎲 that makes V_{IOR} output $x' \in L(R')$.
2. $\tilde{P}$ can attack VC (e.g. use inconsistent ans).

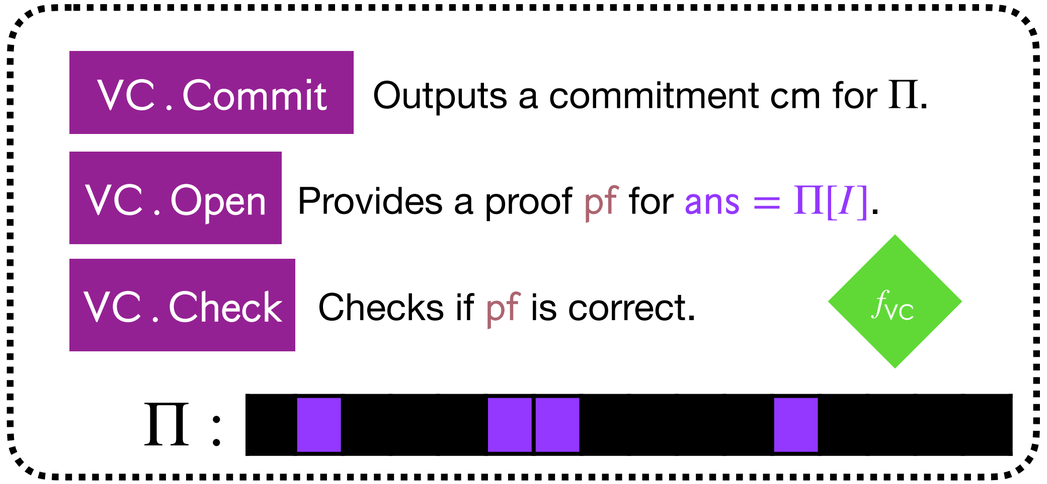


BCS[IOR, VC]

Ingredient #1: Interactive oracle reduction (IOR)



Ingredient #2: Vector commitment scheme (VC)

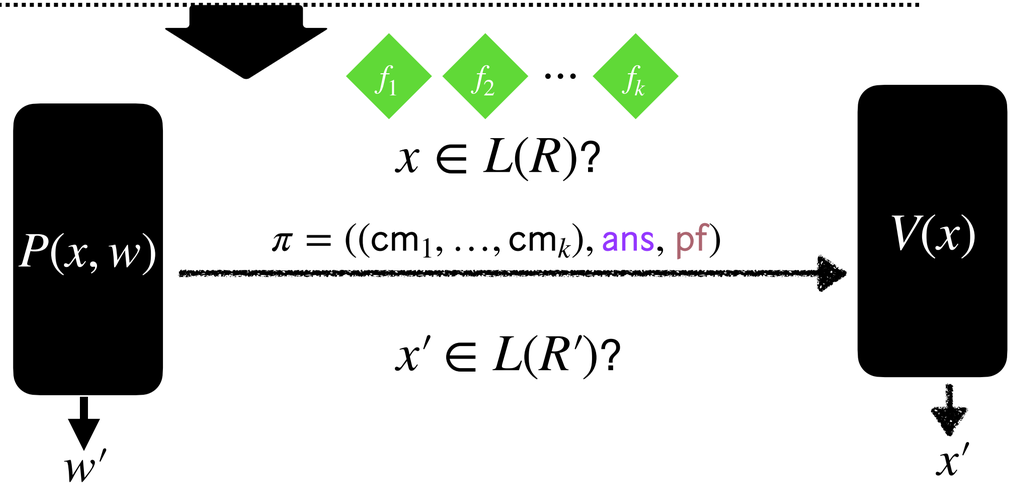


Two potential attacks to **BCS[IOR, VC]**:

1. $\tilde{P}$ can query f_1 f_2 $\cdots$ f_k many times to get 🎲 that makes V_{IOR} output $x' \in L(R')$.

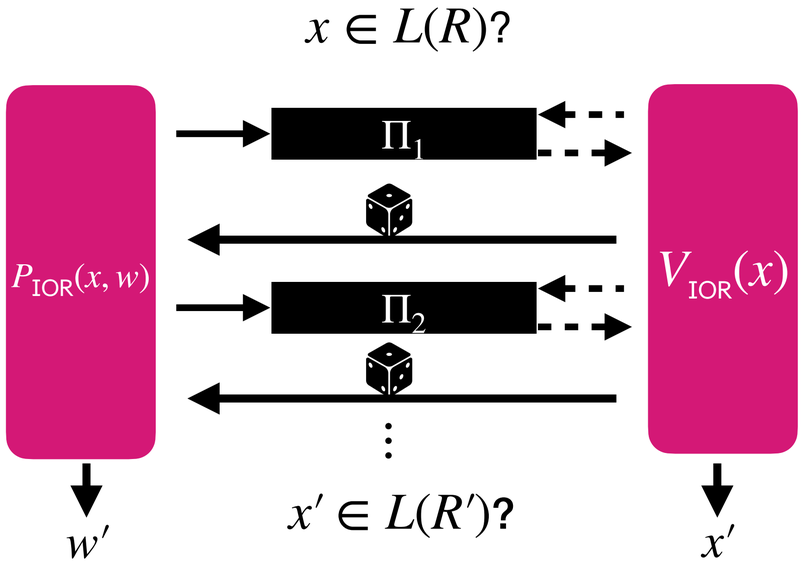
   FS error of V_{IOR}

2. $\tilde{P}$ can attack VC (e.g. use inconsistent ans).

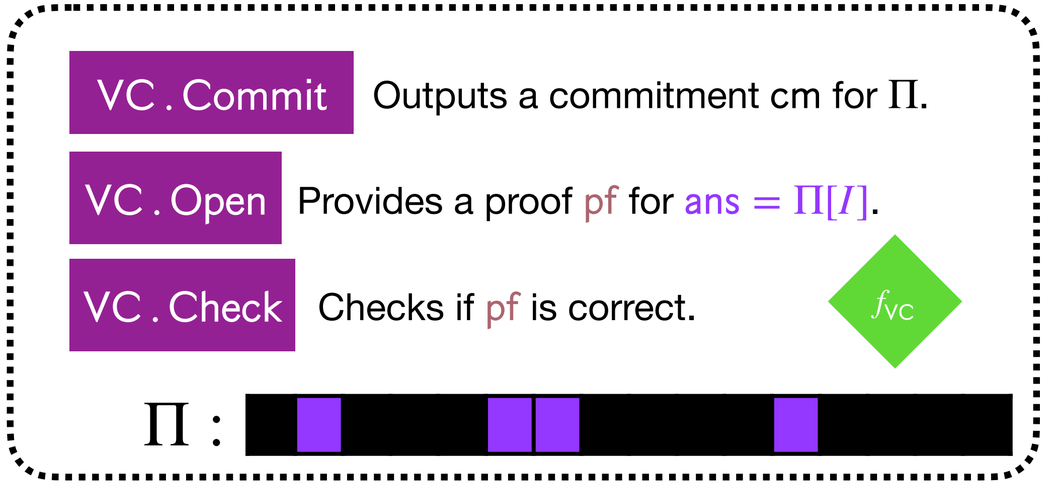


BCS[IOR, VC]

Ingredient #1: Interactive oracle reduction (IOR)



Ingredient #2: Vector commitment scheme (VC)



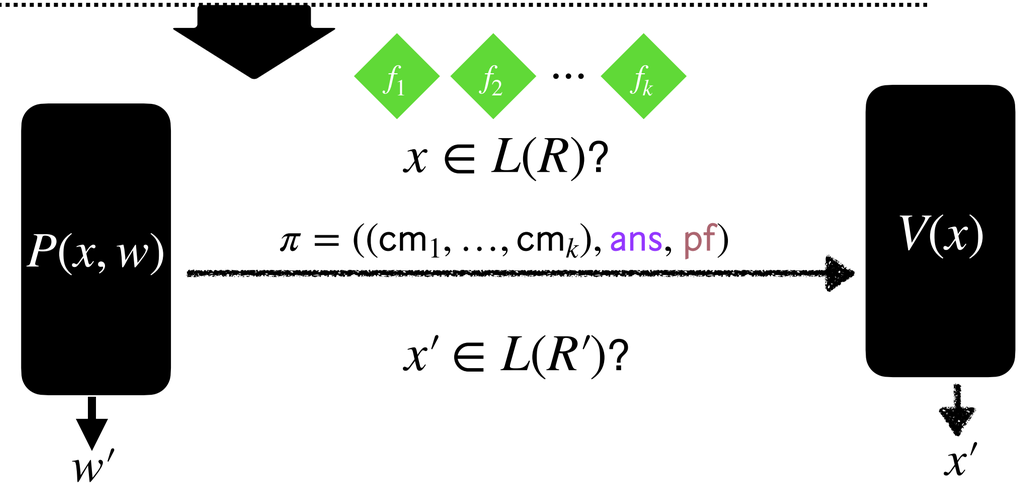
Two potential attacks to **BCS[IOR, VC]**:

1. $\tilde{P}$ can query f_1 f_2 $\cdots$ f_k many times to get 🎲 that makes V_{IOR} output $x' \in L(R')$.

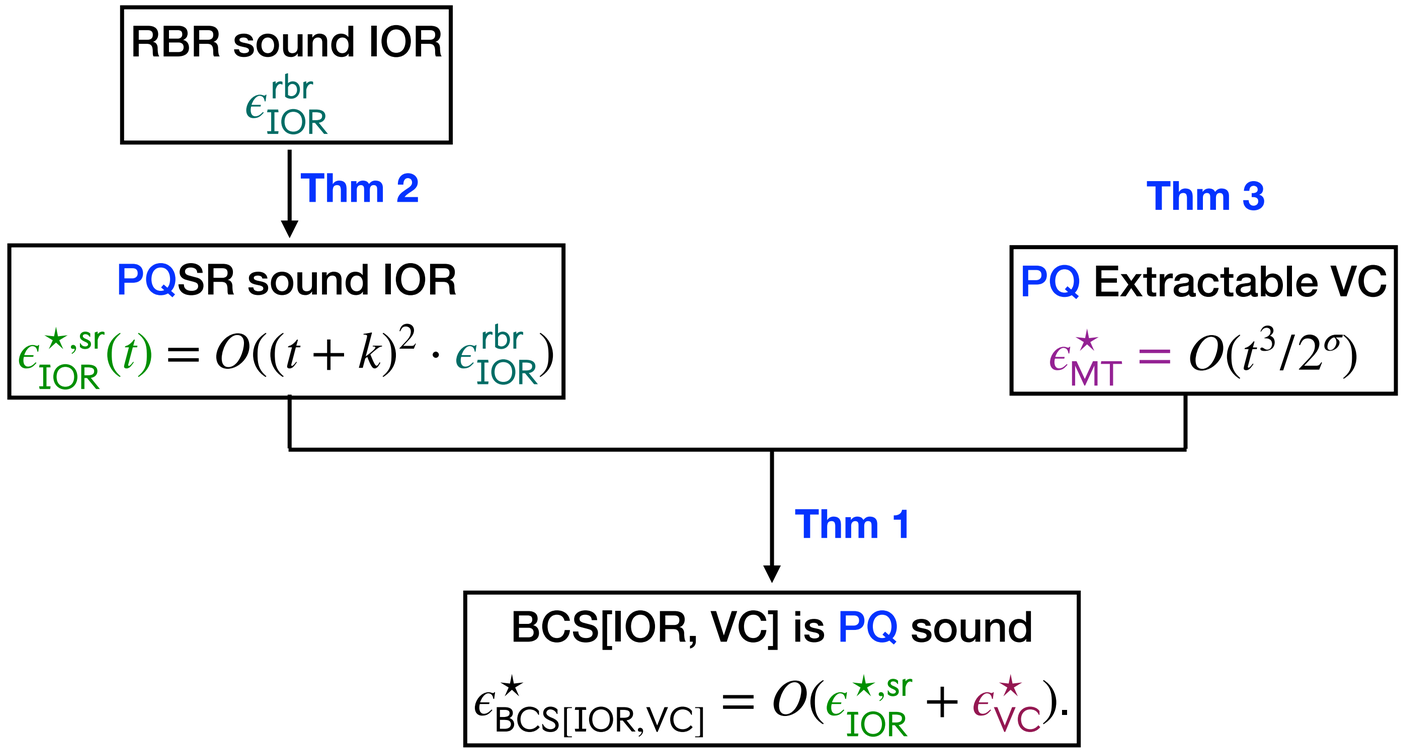
   FS error of V_{IOR}

2. $\tilde{P}$ can attack VC (e.g. use inconsistent ans).

   VC error

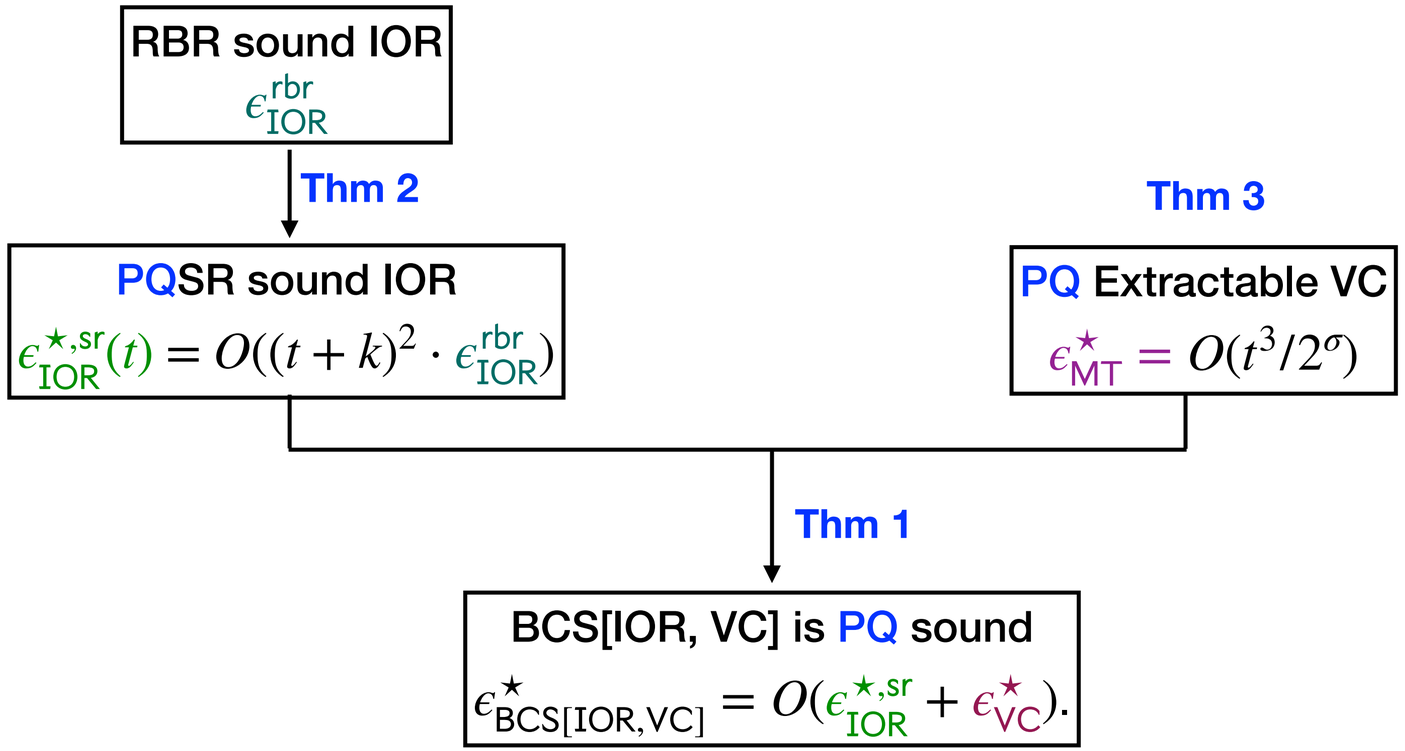


The role of state-restoration



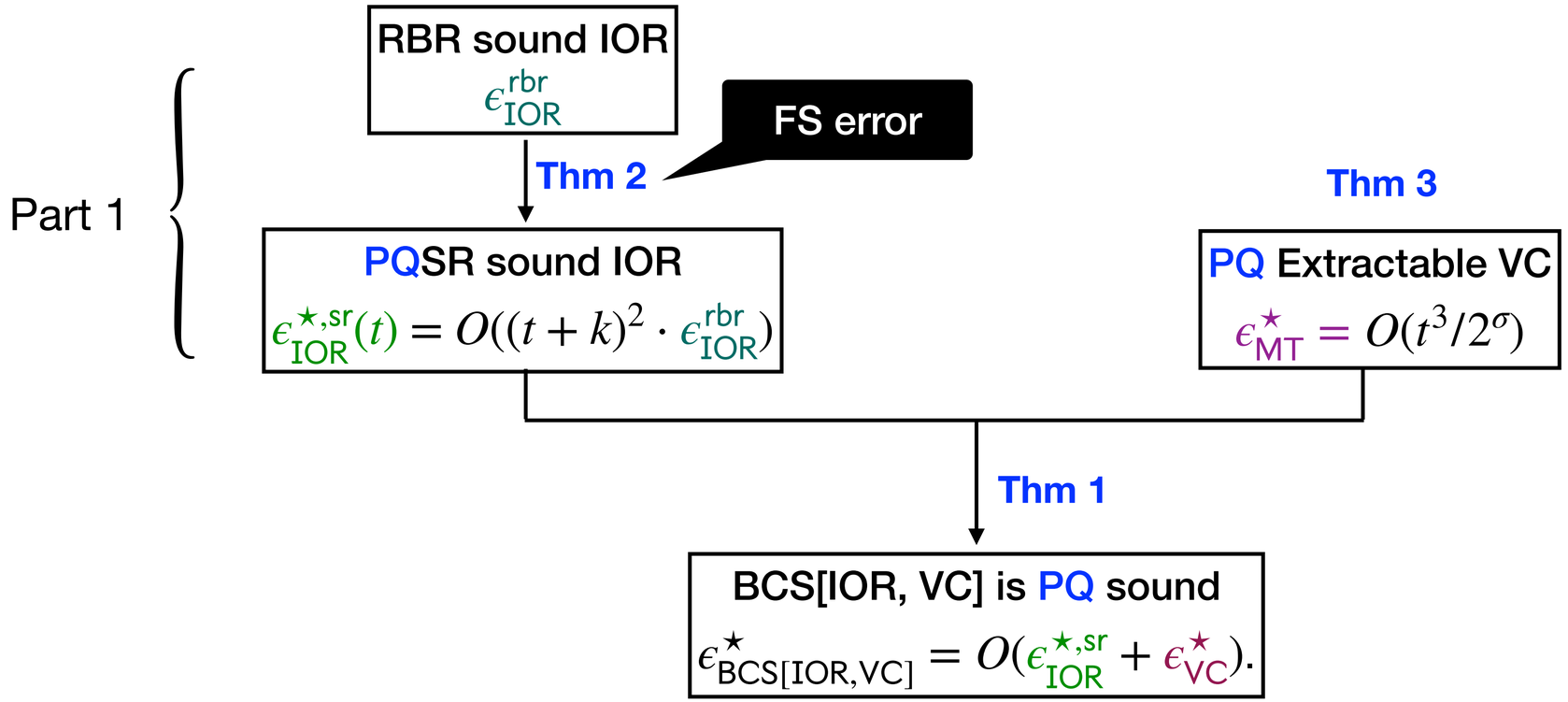
The role of state-restoration

Today's focus: soundness



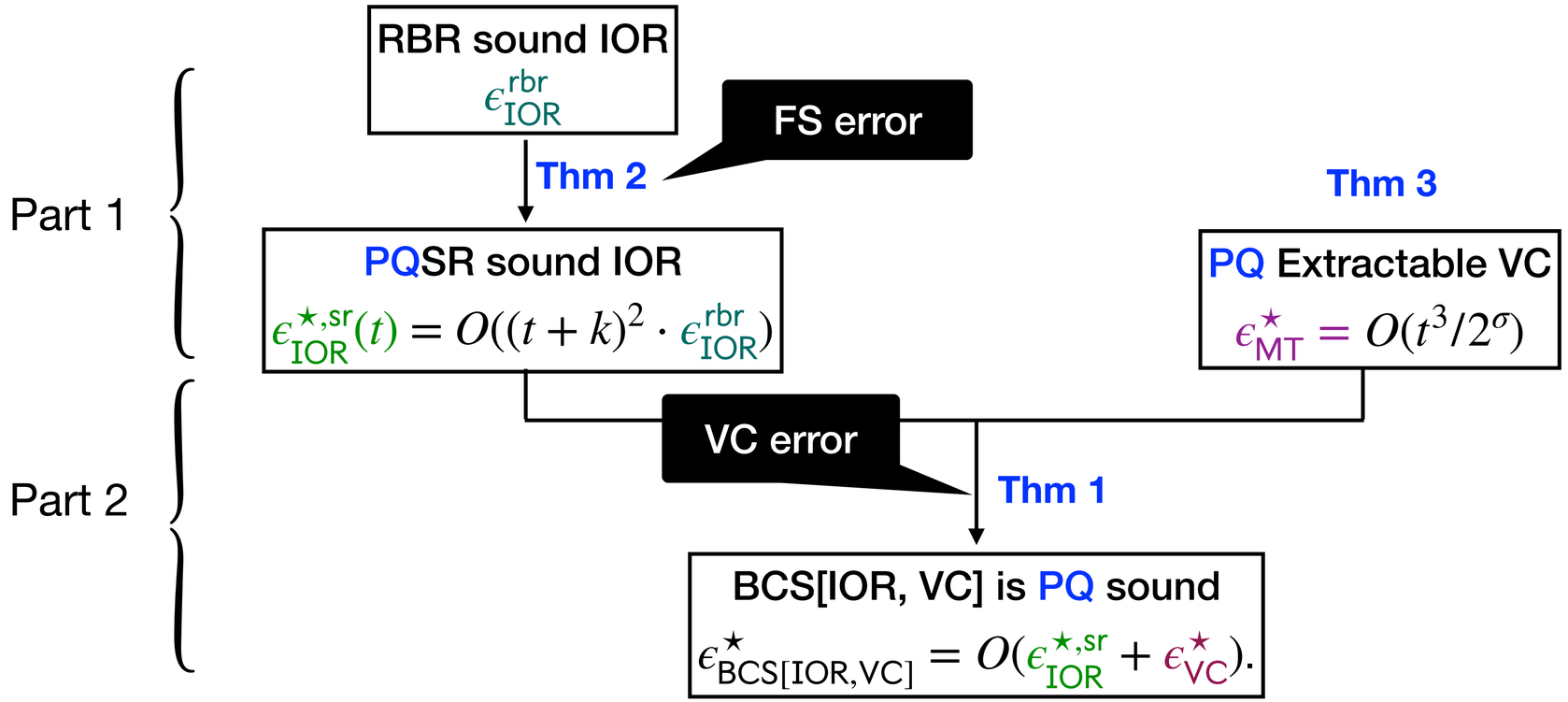
The role of state-restoration

Today's focus: soundness



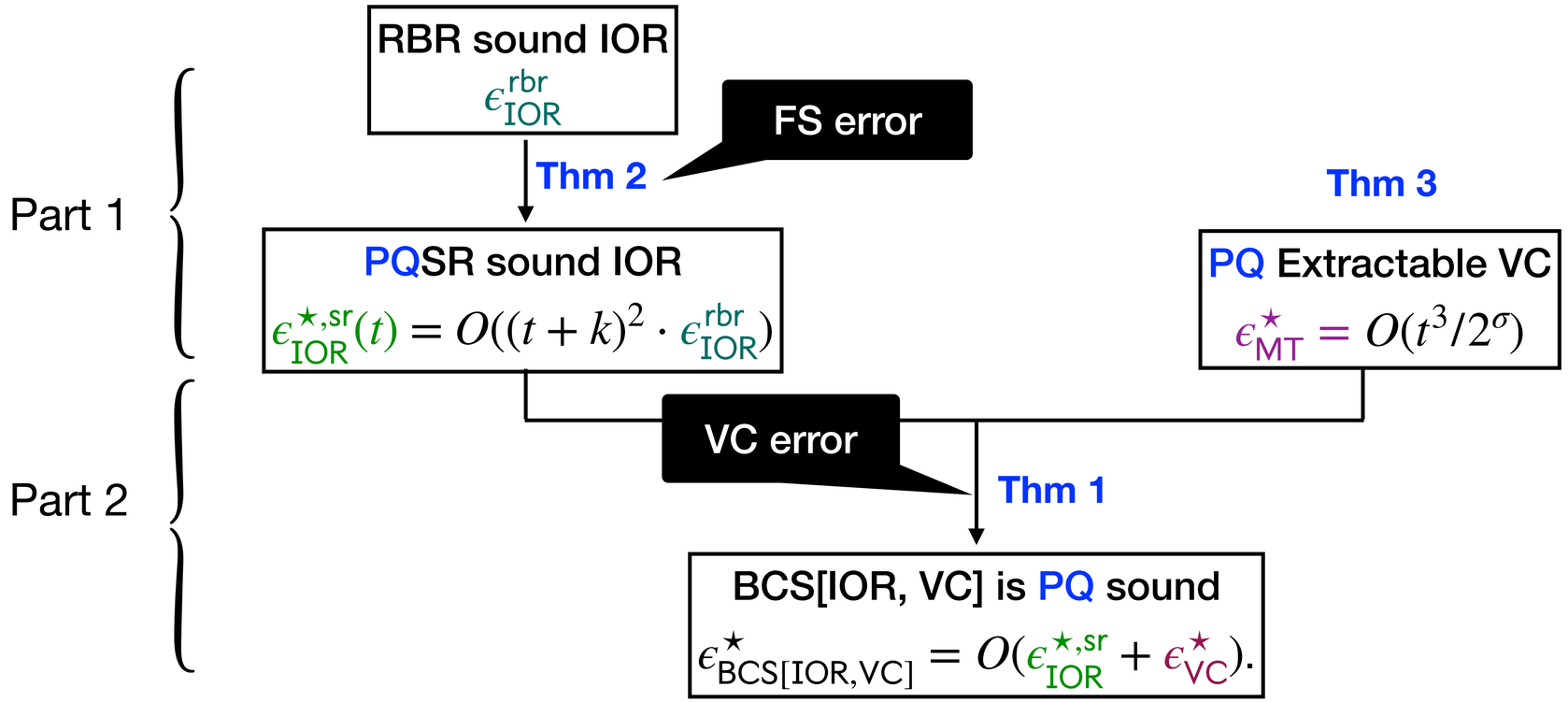
The role of state-restoration

Today's focus: soundness



The role of state-restoration

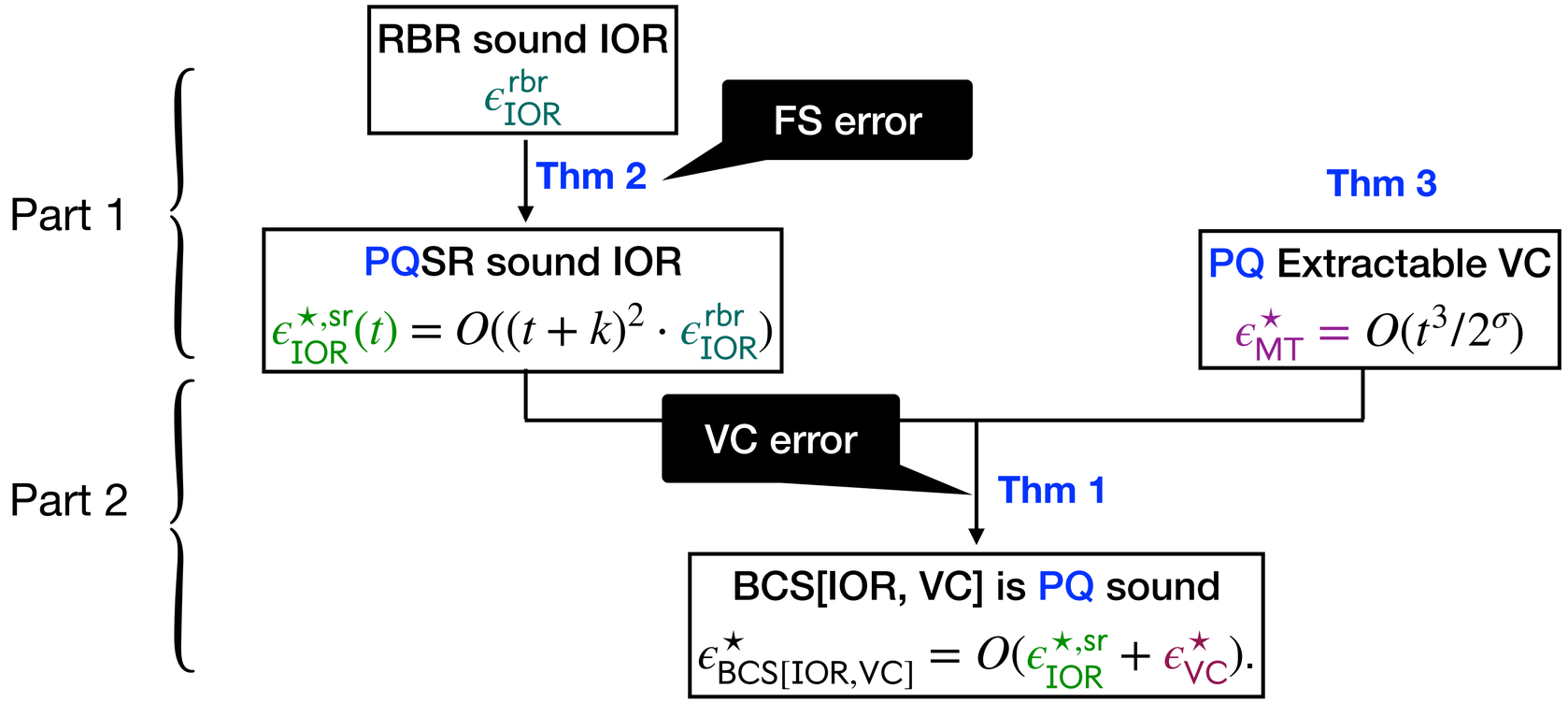
Today's focus: soundness



Putting it together: $\epsilon^{\star}_{\mathsf{BCS[IOR,MT]}} = O((t+k)^2 \cdot \epsilon^{\mathsf{rbr}}_{\mathsf{IOR}}) + O(t^3/2^{\sigma})$

The role of state-restoration

Today's focus: soundness



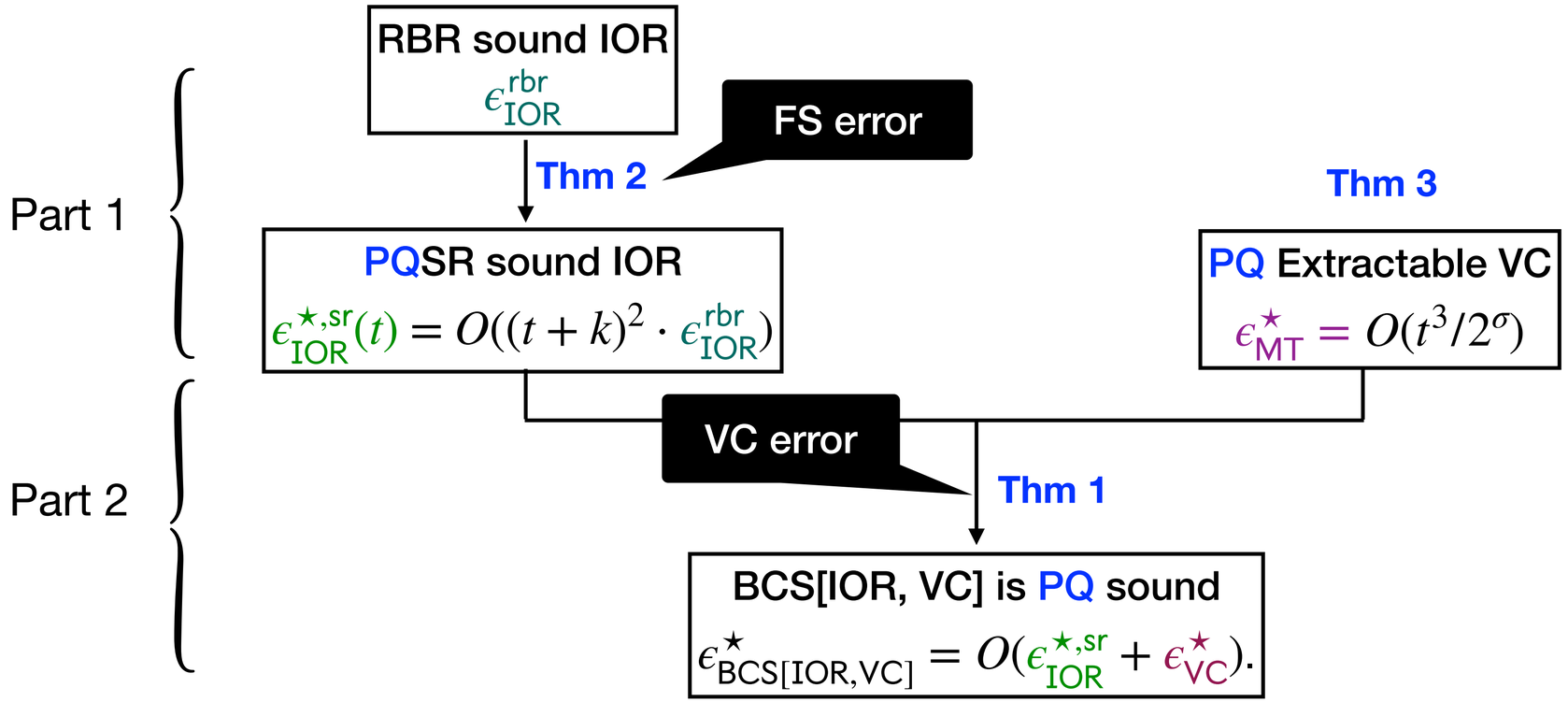
Putting it together: $\epsilon^{\star}_{\mathsf{BCS[IOR,MT]}} = O((t+k)^2 \cdot \epsilon^{\mathsf{rbr}}_{\mathsf{IOR}}) + O(t^3/2^{\sigma})$

BCS error = FS error + VC error for PQ case!

The role of state-restoration

PQSR is weak enough
s.t. it only captures the FS error
and is implied by a classical property

Today's focus: soundness



Putting it together: $\epsilon^{\star}_{\mathsf{BCS[IOR,MT]}} = O((t+k)^2 \cdot \epsilon^{\mathsf{rbr}}_{\mathsf{IOR}}) + O(t^3/2^{\sigma})$

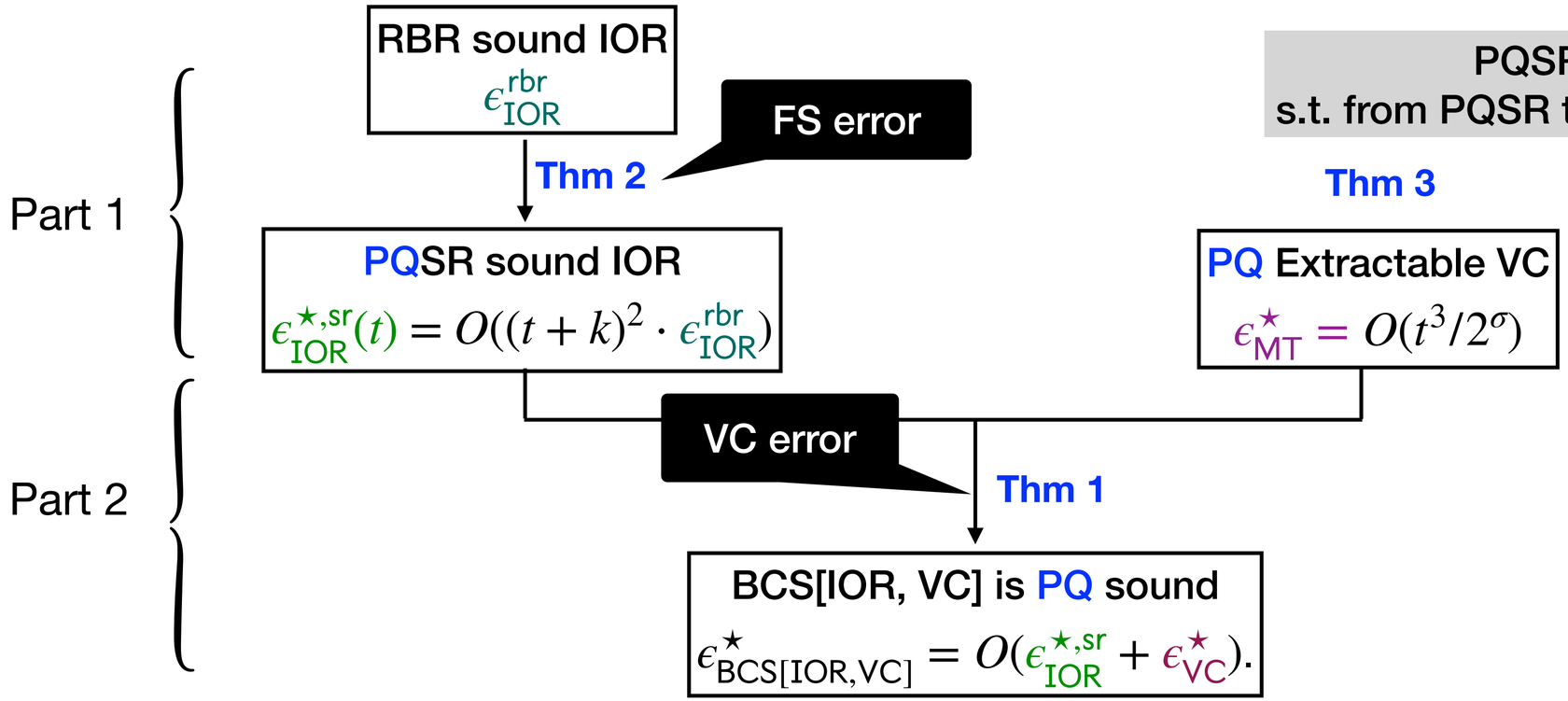
BCS error = FS error + VC error for PQ case!

The role of state-restoration

Today's focus: soundness

PQSR is weak enough
s.t. it only captures the FS error
and is implied by a classical property

PQSR is strong enough
s.t. from PQSR to BCS, there is only VC error

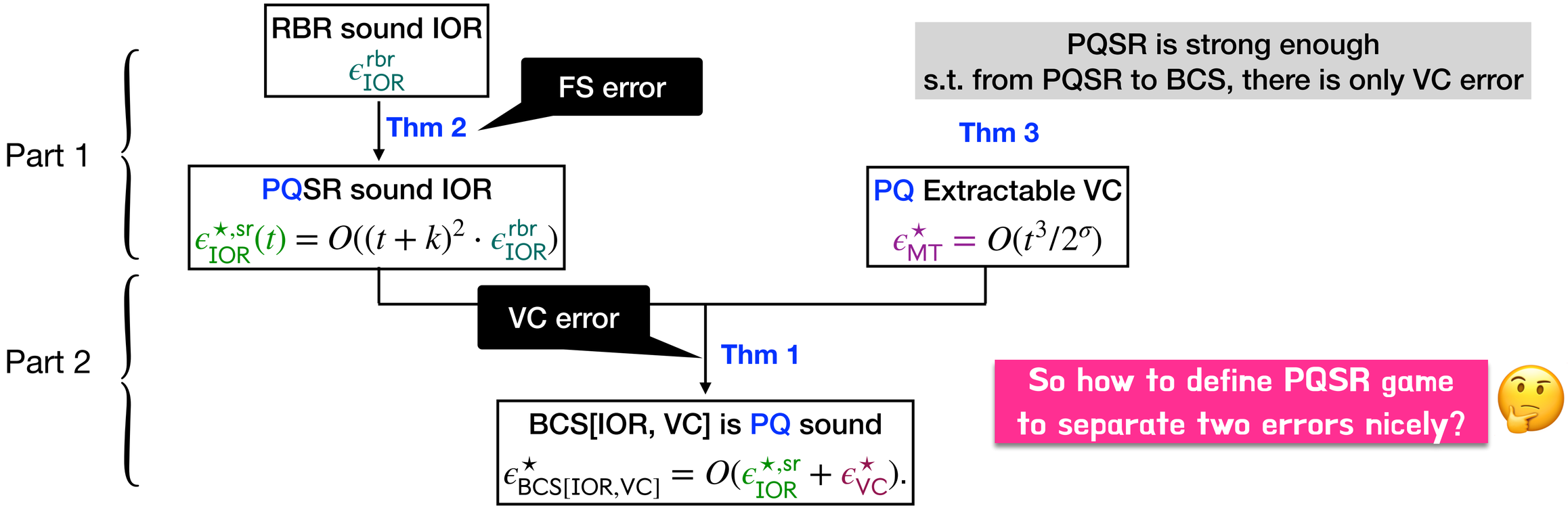


Putting it together: $\epsilon^{\star}_{\mathsf{BCS[IOR,MT]}} = O((t+k)^2 \cdot \epsilon^{\mathsf{rbr}}_{\mathsf{IOR}}) + O(t^3/2^{\sigma})$

BCS error = FS error + VC error for PQ case!

The role of state-restoration

Today's focus: soundness



PQSR is weak enough
s.t. it only captures the FS error
and is implied by a classical property

PQSR is strong enough
s.t. from PQSR to BCS, there is only VC error

Thm 3

So how to define PQSR game
to separate two errors nicely?

Putting it together: $\epsilon^{\star}_{\mathsf{BCS[IOR,MT]}} = O((t+k)^2 \cdot \epsilon^{\mathsf{rbr}}_{\mathsf{IOR}}) + O(t^3/2^\sigma)$

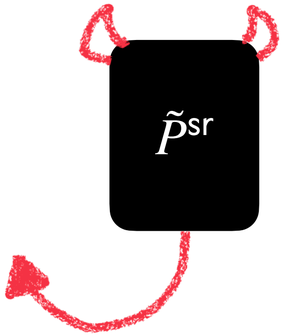
BCS error = FS error + VC error for PQ case!

Part 1:
PQSR soundness is implied by RBR soundness

State-restoration captures the classical FS error

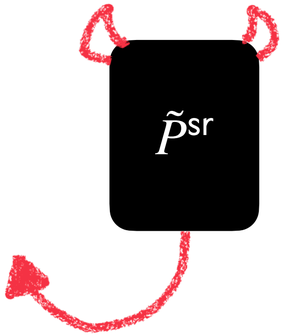
State-restoration captures the classical FS error

Classical adversary



State-restoration captures the classical FS error

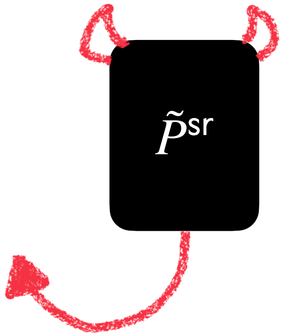
Classical adversary



$x, (\Pi_1, \ldots, \Pi_i)$

State-restoration captures the classical FS error

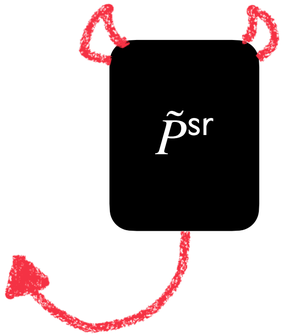
Classical adversary

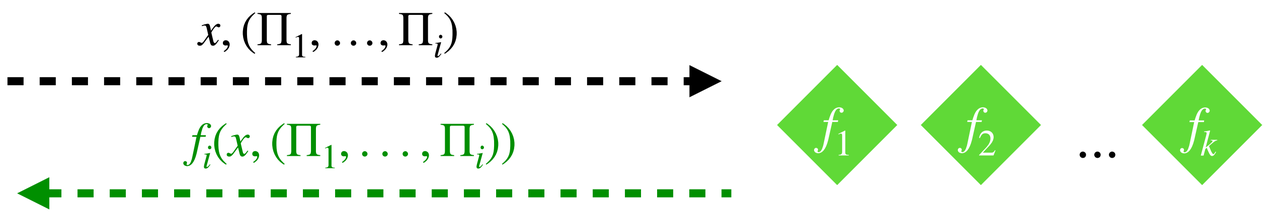


$x, (\Pi_1, \ldots, \Pi_i)$

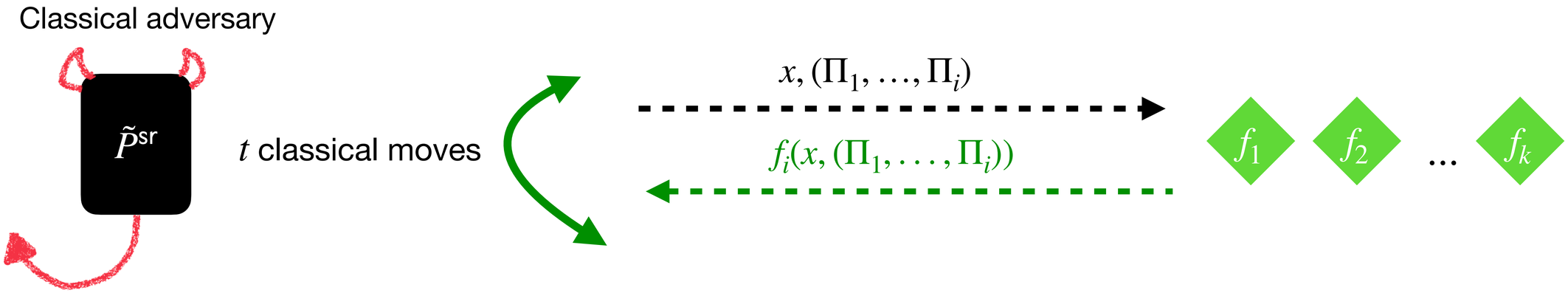
State-restoration captures the classical FS error

Classical adversary

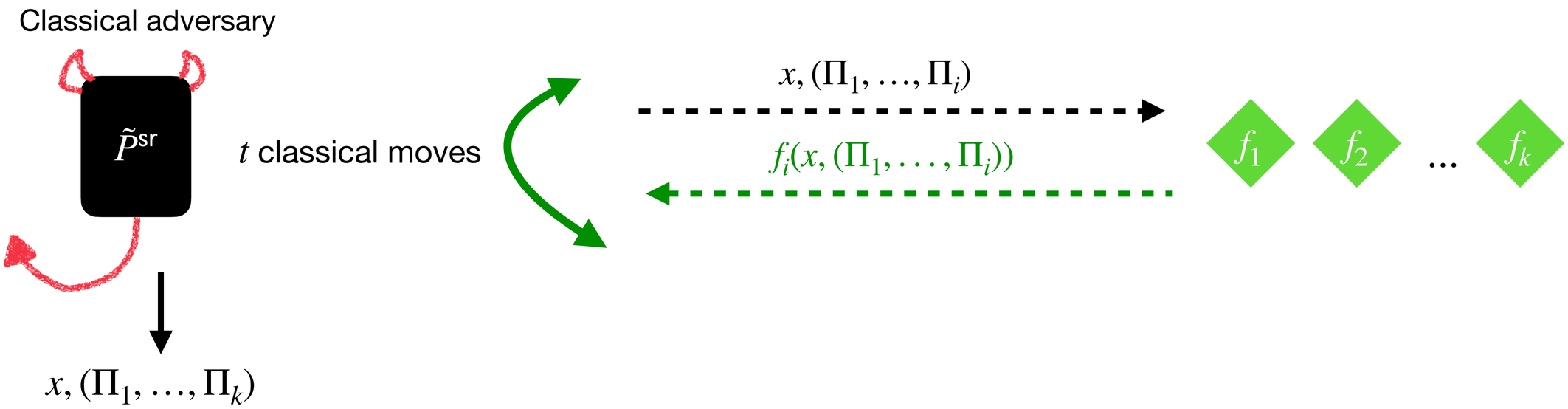




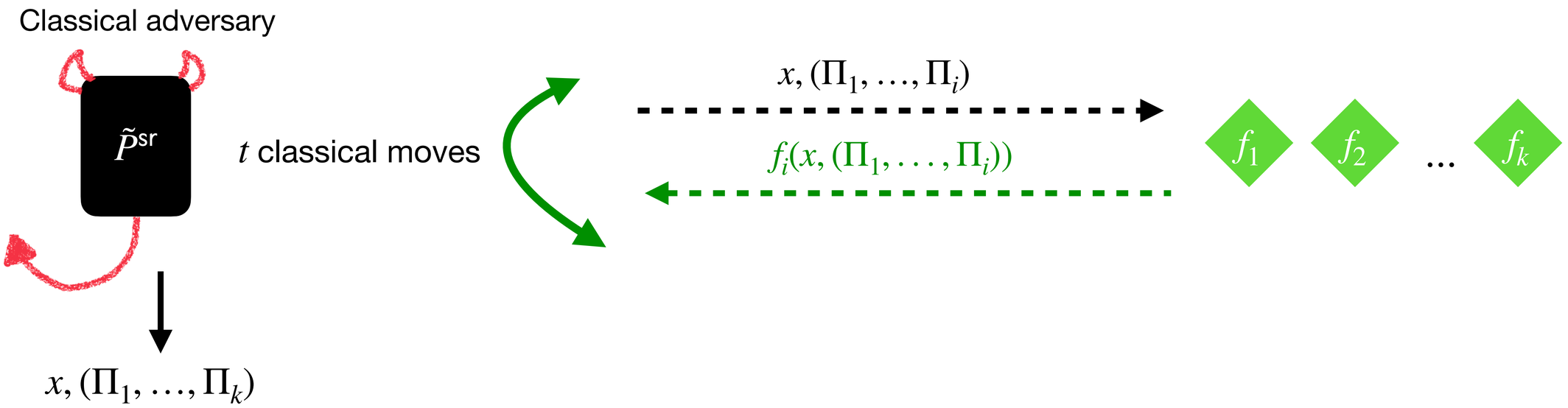
State-restoration captures the classical FS error

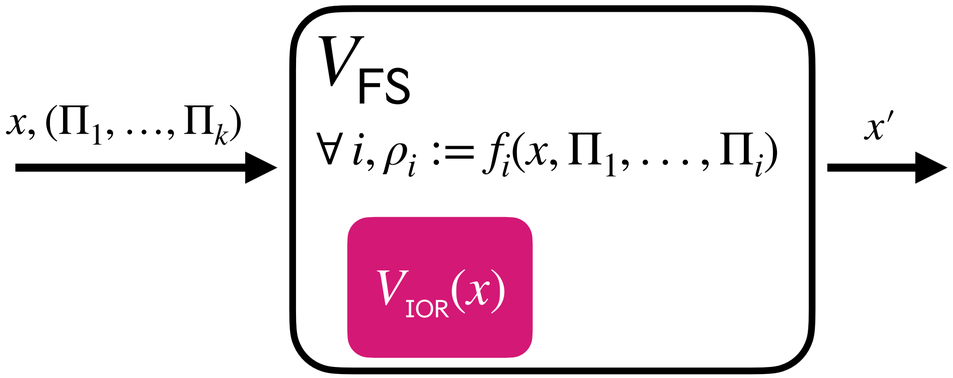


State-restoration captures the classical FS error

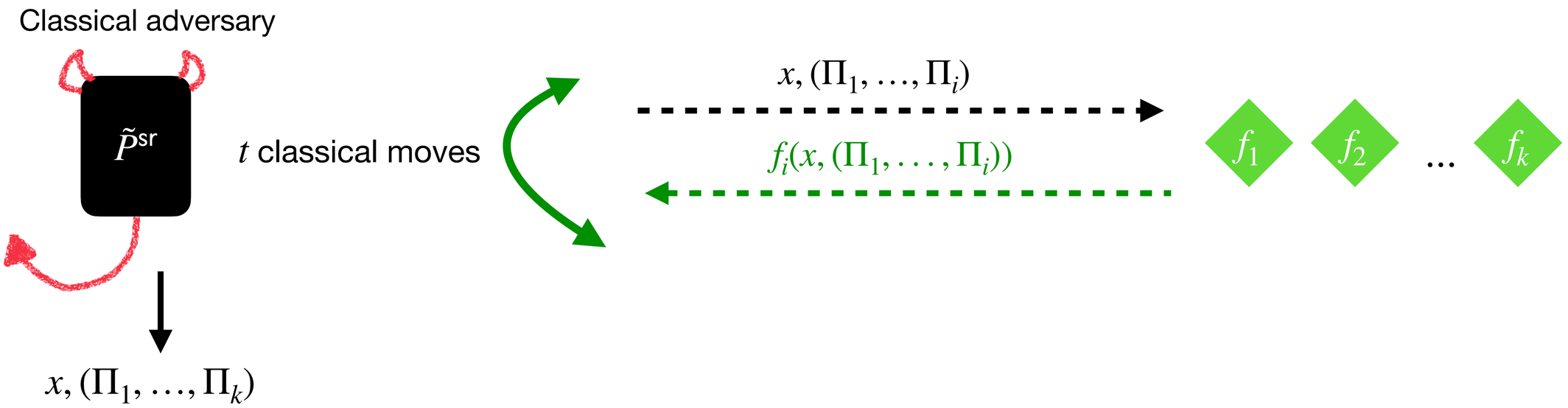


State-restoration captures the classical FS error





State-restoration captures the classical FS error

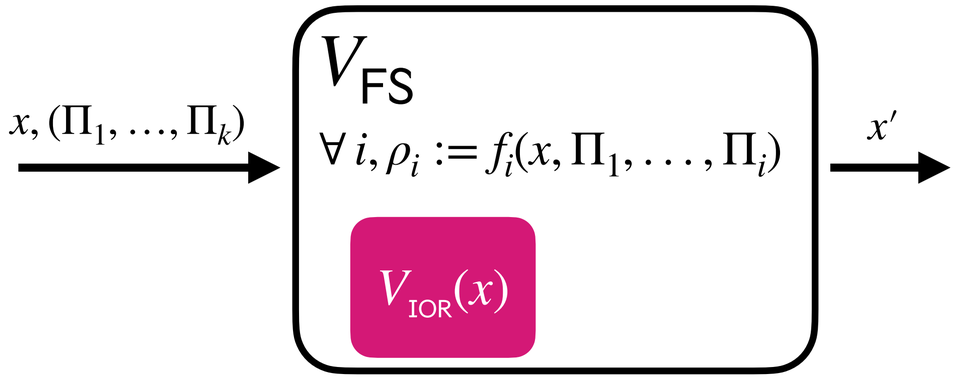


Soundness:

t-move $\tilde{P}^{\mathsf{sr}}$ cannot output $x, (\Pi_1, \ldots, \Pi_k)$

s.t. it reduces a no instance $x \notin L(R)$

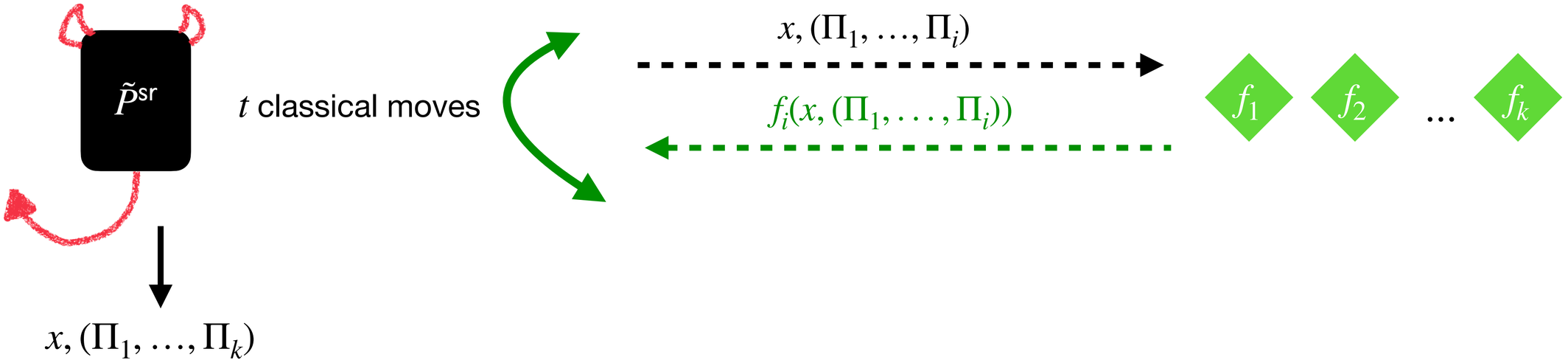
to a yes instance $x' \in L(R')$, except with error $\epsilon^{\mathsf{sr}}_{\mathsf{IOR}}(t)$



State-restoration captures the classical FS error

$\epsilon^{\text{sr}}_{\text{IOR}}$ = the (classical) soundness error of FS[IOR]

Classical adversary

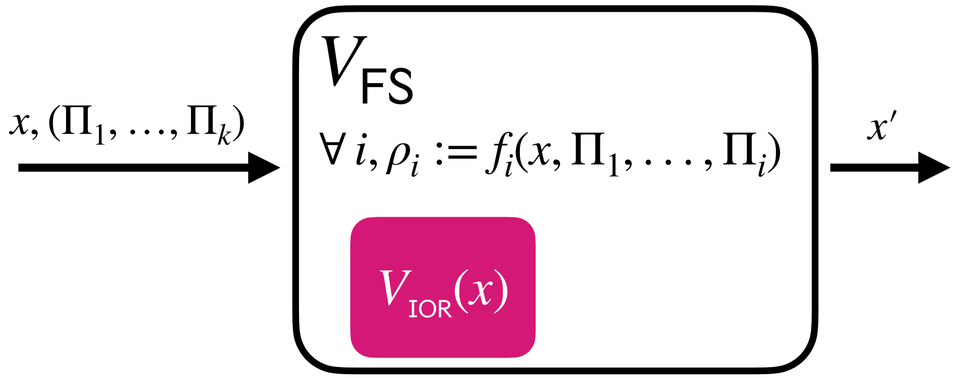


Soundness:

t-move $\tilde{P}^{\text{sr}}$ cannot output $x, (\Pi_1, \ldots, \Pi_k)$

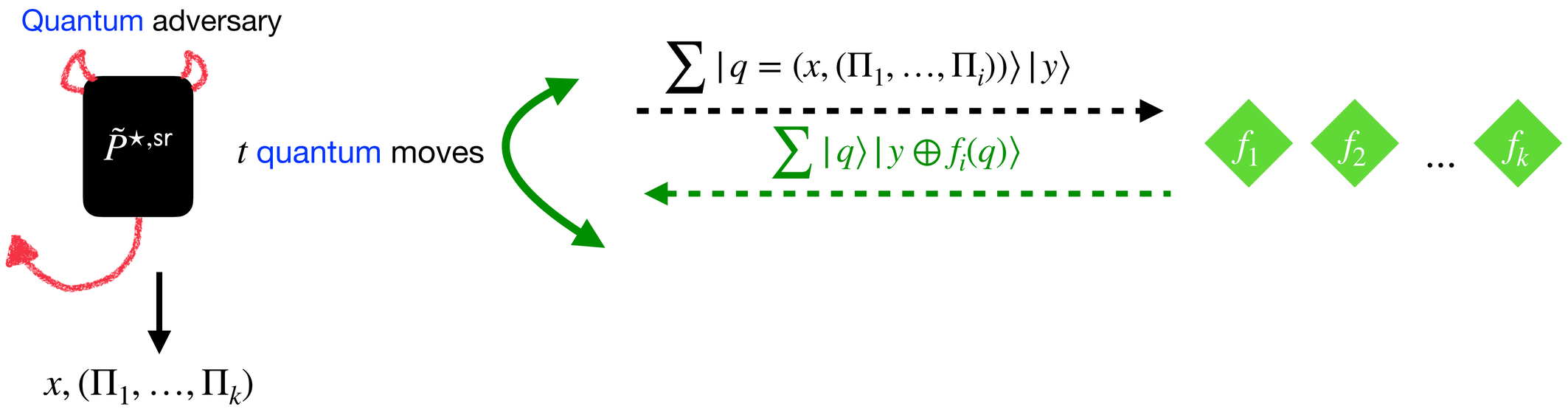
s.t. it reduces a no instance $x \notin L(R)$

to a yes instance $x' \in L(R')$, except with error $\epsilon^{\text{sr}}_{\text{IOR}}(t)$



Our PQ state-restoration captures the PQ FS error

$\epsilon_{\text{IOR}}^{\star,\text{sr}}$ = the PQ soundness error of FS[IOR]

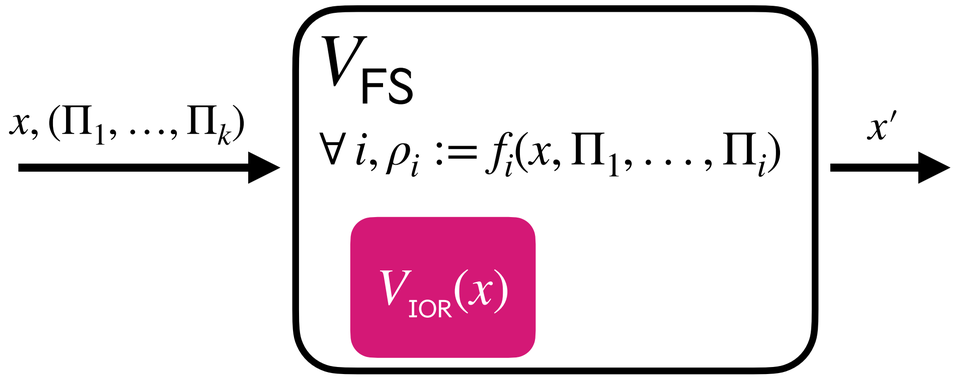


Soundness:

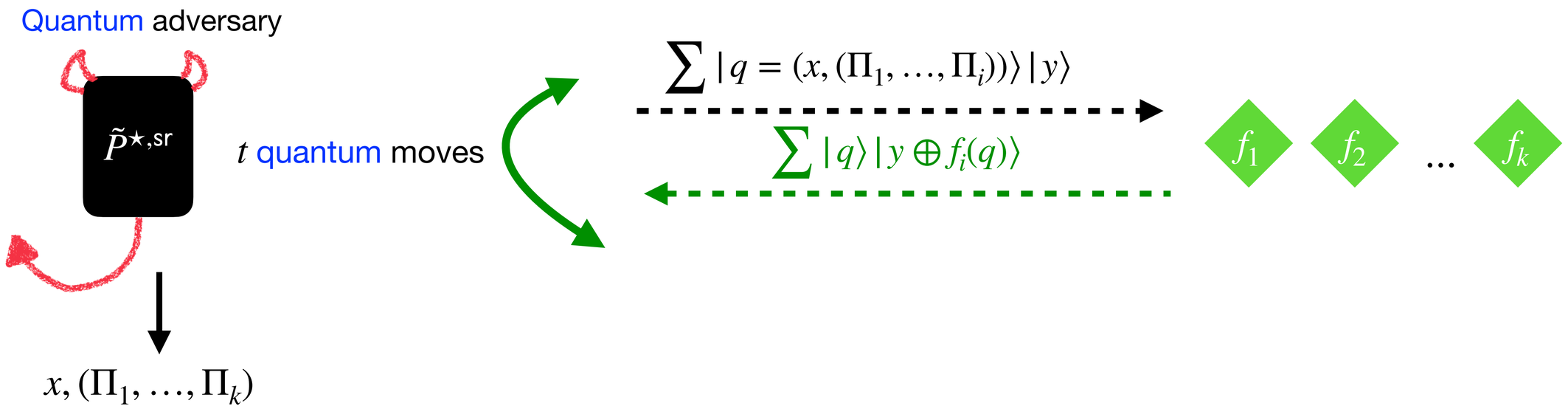
t-move $\tilde{P}^{\star,\text{sr}}$ cannot output $x, (\Pi_1, \ldots, \Pi_k)$

s.t. it reduces a no instance $x \notin L(R)$

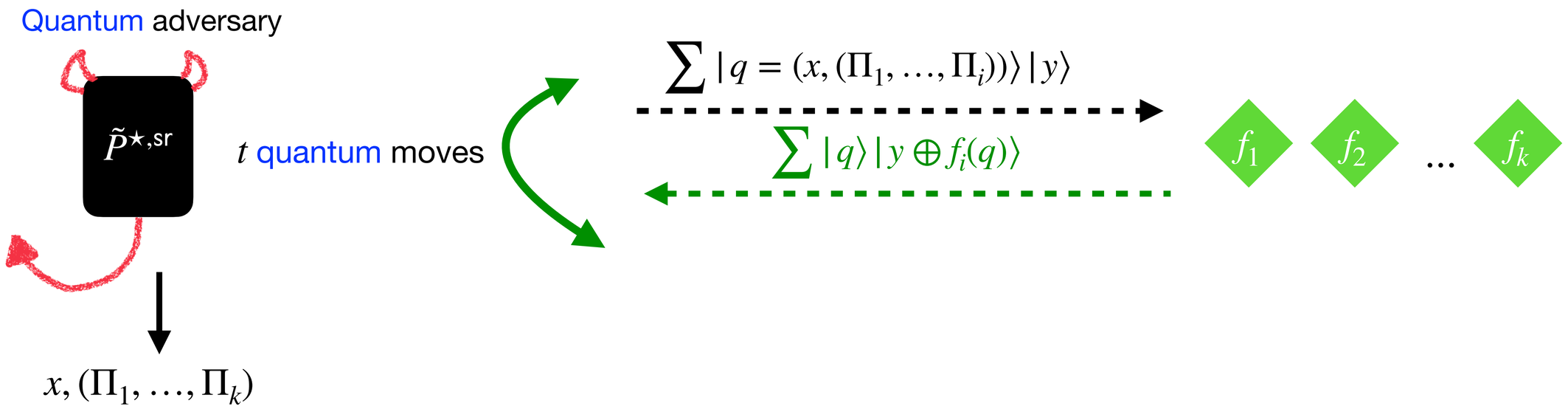
to a yes instance $x' \in L(R')$, except with error $\epsilon_{\text{IOR}}^{\star,\text{sr}}(t)$



Our PQ state-restoration captures the PQ FS error

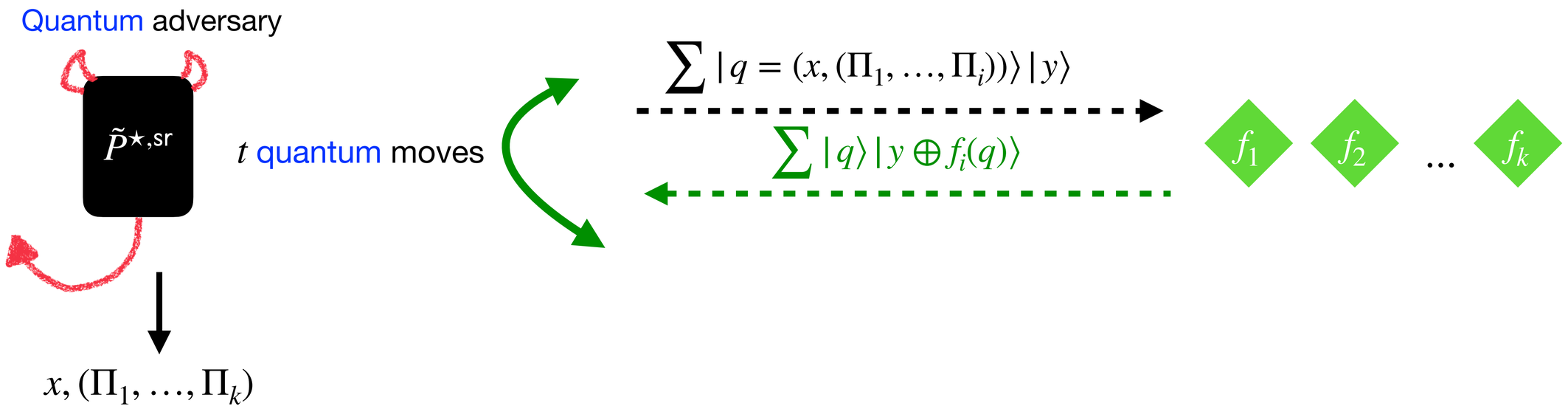


Our PQ state-restoration captures the PQ FS error



$\tilde{P}^{\star,\mathrm{sr}}$ has quantum power.
What if it queries multiple oracles at once?

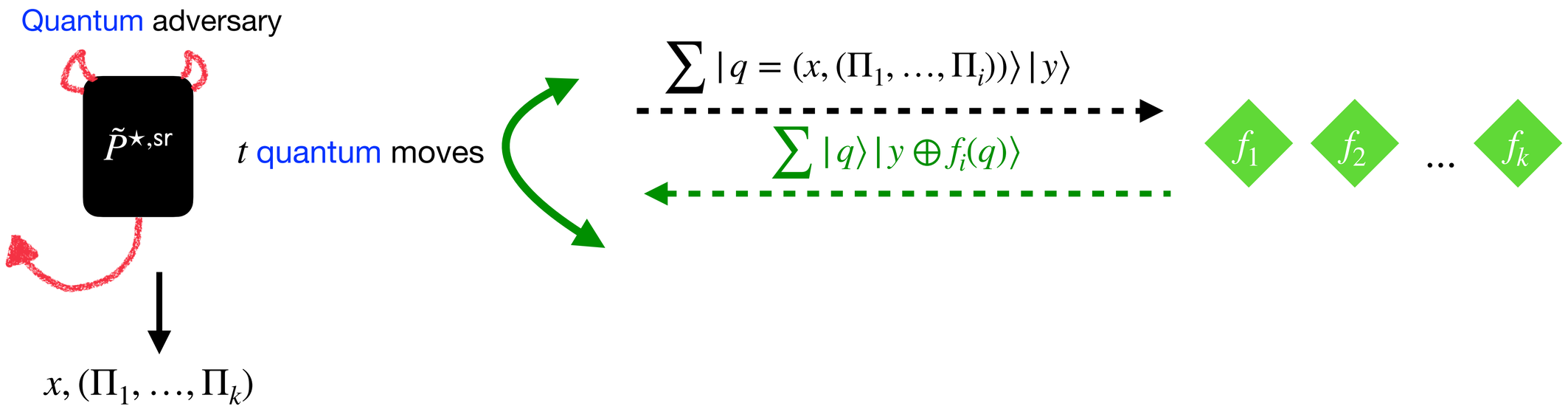
Our PQ state-restoration captures the PQ FS error



$\tilde{P}^{\star,\mathrm{sr}}$ has quantum power.
What if it queries multiple oracles at once?

Our final definition captures this!

Our PQ state-restoration captures the PQ FS error

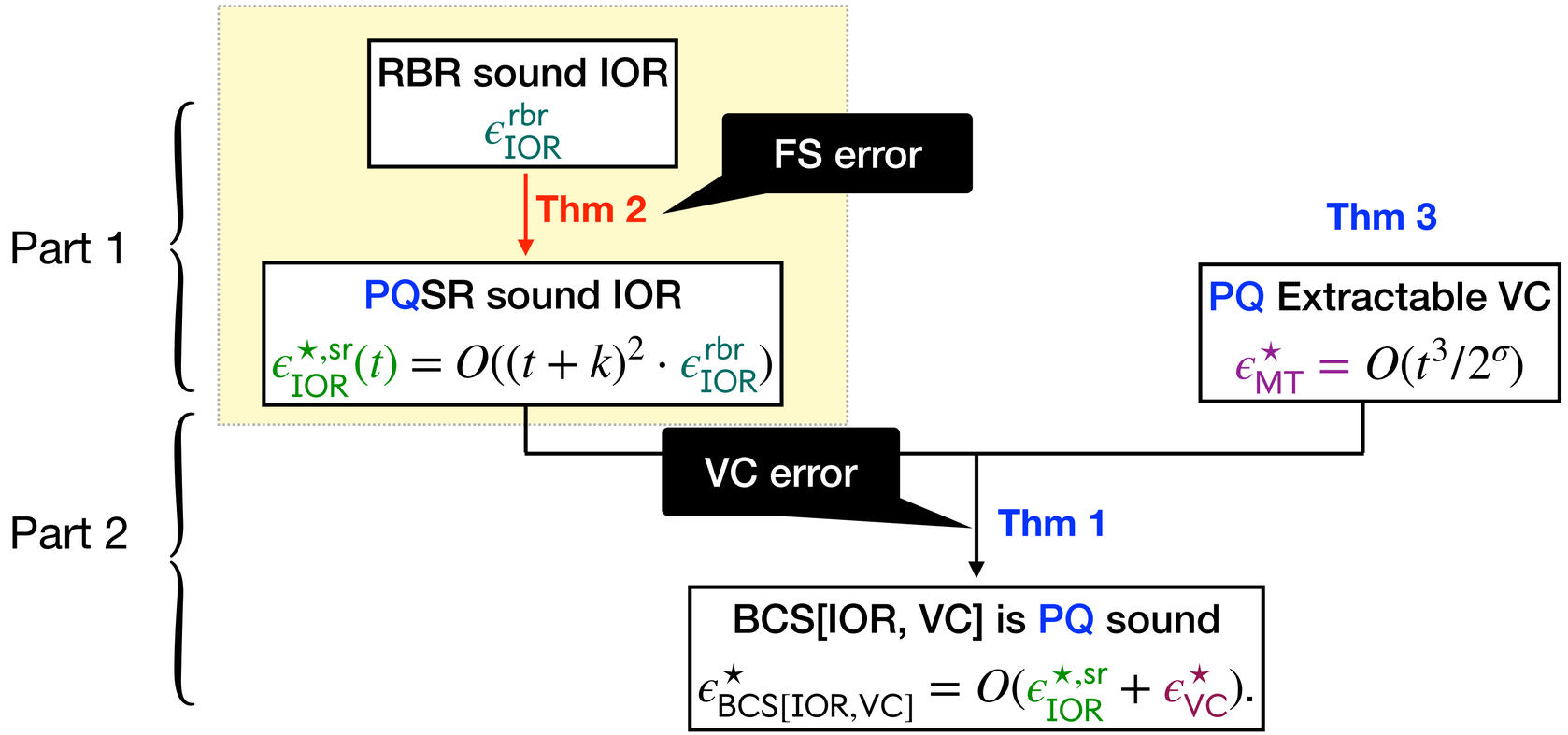


$\tilde{P}^{\star,\mathrm{sr}}$ has quantum power.
What if it queries multiple oracles at once?

Our final definition captures this!

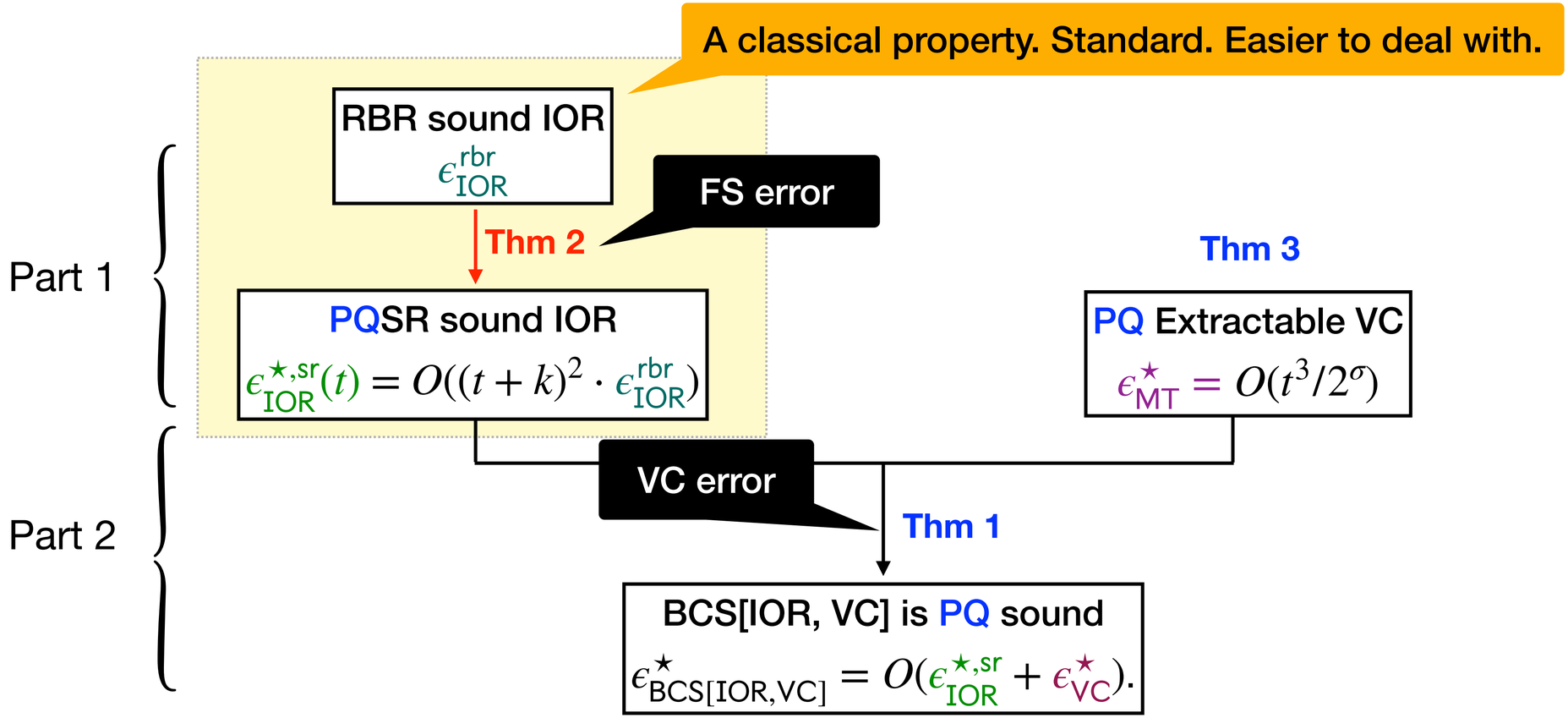
PQSR is a quantum property (too difficult).
Can we connect it with an easy classical property?

PQSR soundness is implied by RBR soundness



Putting it together: $\epsilon^{\star}_{\mathsf{BCS[IOR,MT]}} = O((t+k)^2 \cdot \epsilon^{\mathsf{rbr}}_{\mathsf{IOR}}) + O(t^3/2^{\sigma})$

PQSR soundness is implied by RBR soundness



Putting it together: $\epsilon^{\star}_{\mathsf{BCS[IOR,MT]}} = O((t+k)^2 \cdot \epsilon^{\mathsf{rbr}}_{\mathsf{IOR}}) + O(t^3/2^{\sigma})$

Definition of RBR soundness $\epsilon_{\text{IOR}}^{\text{rbr}}$:

Each partial transcript is labeled either

doomed

or not doomed

Definition of RBR soundness $\epsilon_{\text{IOR}}^{\text{rbr}}$:

Each partial transcript is labeled either

doomed **Almost impossible to make V output $x' \in L'$**

or not doomed

Definition of RBR soundness $\epsilon_{\text{IOR}}^{\text{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$

Definition of RBR soundness $\epsilon_{\text{IOR}}^{\text{rbr}}$:

doomed $x, \Pi_1, \rho_1, \ldots, \rho_{i-1}, \Pi_i$

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

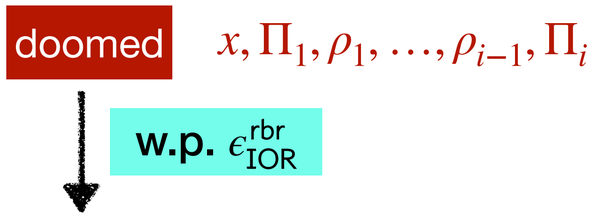
or not doomed — Promising to make V output $x' \in L'$

Definition of RBR soundness $\epsilon_{\text{IOR}}^{\text{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$

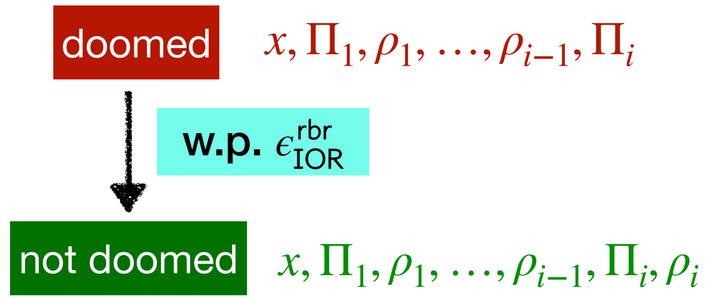


Definition of RBR soundness $\epsilon_{\text{IOR}}^{\text{rbr}}$:

Each partial transcript is labeled either

doomed — **Almost impossible to make V output $x' \in L'$**

or not doomed — **Promising to make V output $x' \in L'$**

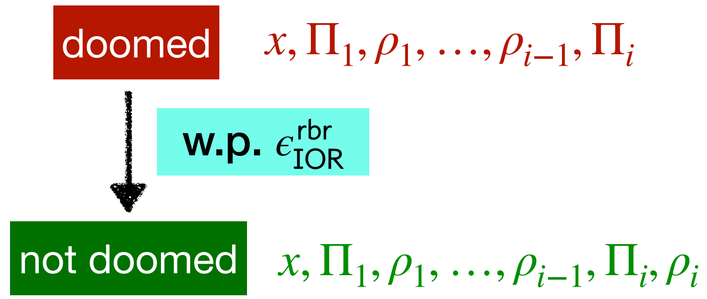


Definition of RBR soundness $\epsilon_{\mathsf{IOR}}^{\mathsf{rbr}}$:

Each partial transcript is labeled either

doomed **Almost impossible to make V output $x' \in L'$**

or not doomed **Promising to make V output $x' \in L'$**



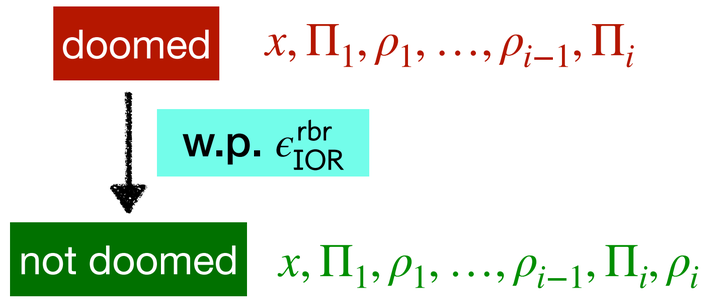
To win SR game, $\tilde{P}^{\mathsf{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

Definition of RBR soundness $\epsilon_{\text{IOR}}^{\text{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\text{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

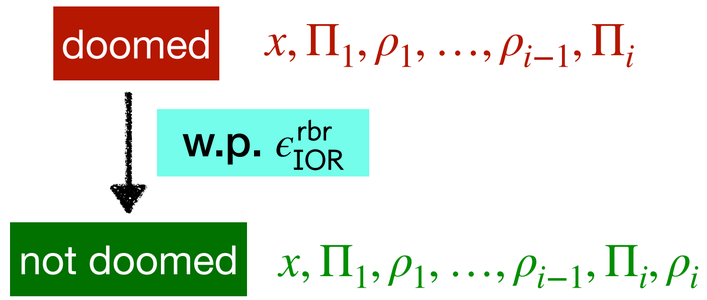
x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

Definition of RBR soundness $\epsilon_{\mathsf{IOR}}^{\mathsf{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\mathsf{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

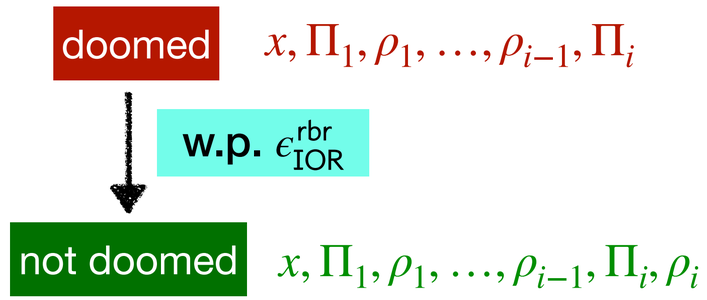
x

Definition of RBR soundness $\epsilon^{\text{rbr}}_{\text{IOR}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\text{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

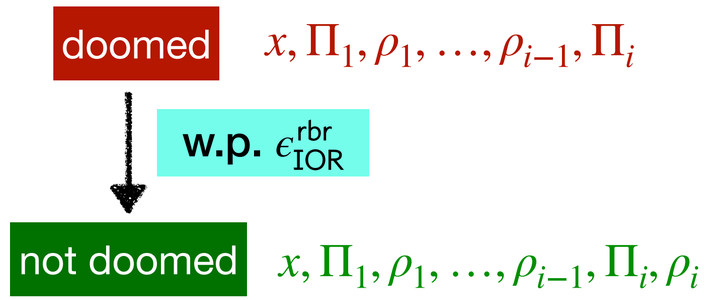
x ⟶

Definition of RBR soundness $\epsilon_{\text{IOR}}^{\text{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\text{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

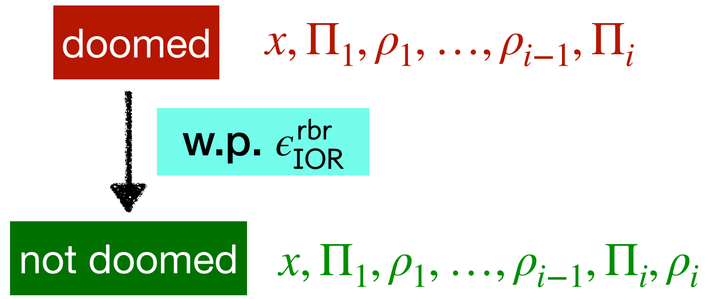
$x \longrightarrow x, \Pi_1$

Definition of RBR soundness $\epsilon_{\mathsf{IOR}}^{\mathsf{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\mathsf{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

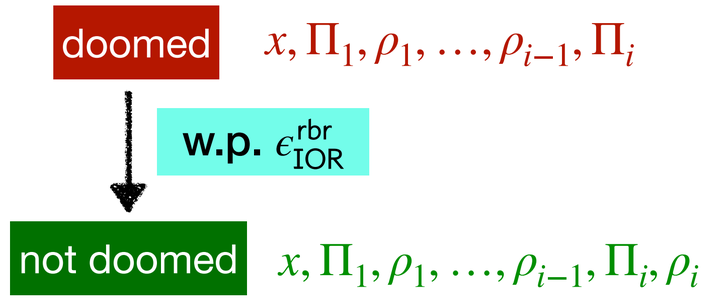
$x \longrightarrow x, \Pi_1 \longrightarrow$

Definition of RBR soundness $\epsilon^{\text{rbr}}_{\text{IOR}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\text{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

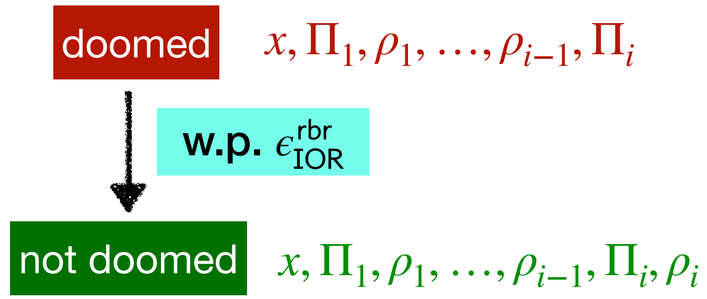
$$x \longrightarrow x, \Pi_1 \longrightarrow x, \Pi_1, \rho_1$$

Definition of RBR soundness $\epsilon_{\mathsf{IOR}}^{\mathsf{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\mathsf{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

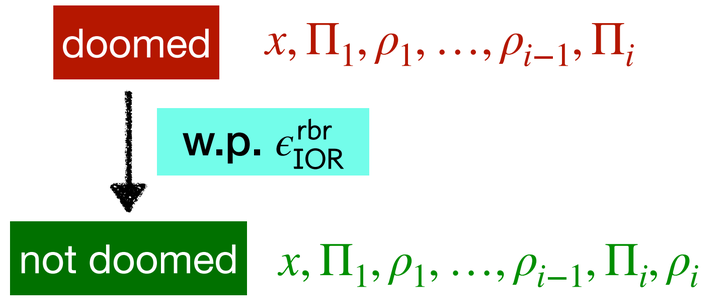
$$x \longrightarrow x, \Pi_1 \longrightarrow x, \Pi_1, \rho_1 \longrightarrow$$

Definition of RBR soundness $\epsilon_{\text{IOR}}^{\text{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\text{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

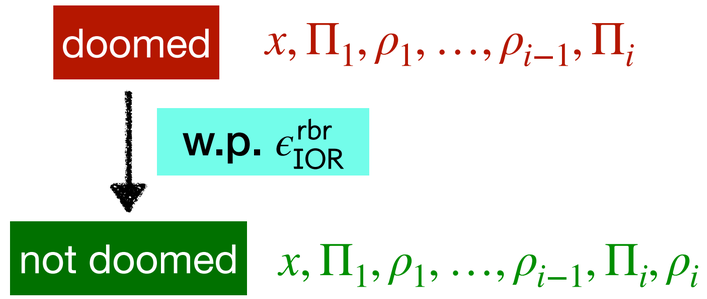
$$x \longrightarrow x, \Pi_1 \longrightarrow x, \Pi_1, \rho_1 \longrightarrow x, \Pi_1, \rho_1, \Pi_2$$

Definition of RBR soundness $\epsilon_{\text{IOR}}^{\text{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\text{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

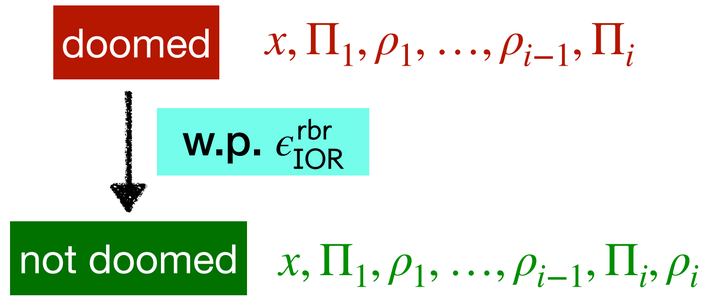
$$x \longrightarrow x, \Pi_1 \longrightarrow x, \Pi_1, \rho_1 \longrightarrow x, \Pi_1, \rho_1, \Pi_2 \longrightarrow$$

Definition of RBR soundness $\epsilon_{\mathsf{IOR}}^{\mathsf{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\mathsf{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

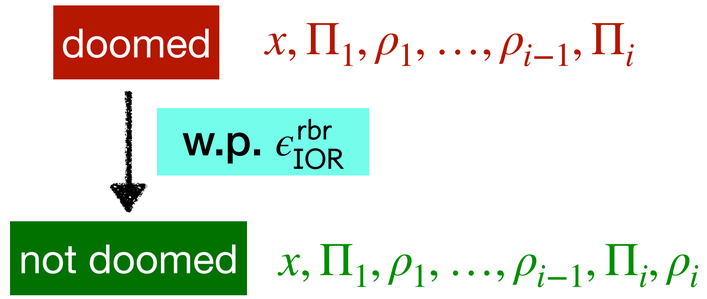
$$x \longrightarrow x, \Pi_1 \longrightarrow x, \Pi_1, \rho_1 \longrightarrow x, \Pi_1, \rho_1, \Pi_2 \longrightarrow \cdots$$

Definition of RBR soundness $\epsilon_{\text{IOR}}^{\text{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\text{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

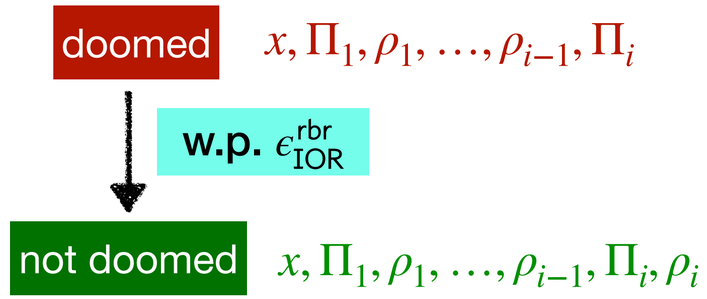
$$x \longrightarrow x, \Pi_1 \longrightarrow x, \Pi_1, \rho_1 \longrightarrow x, \Pi_1, \rho_1, \Pi_2 \longrightarrow \cdots \longrightarrow$$

Definition of RBR soundness $\epsilon_{\text{IOR}}^{\text{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\text{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

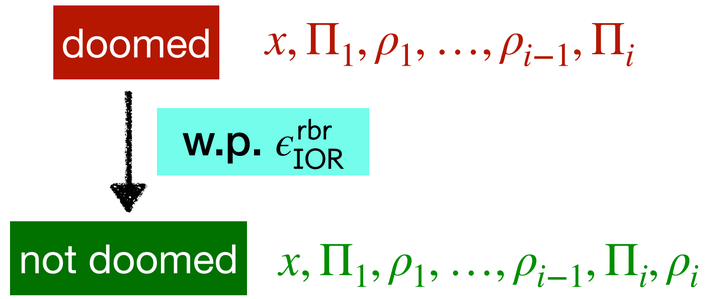
$$x \longrightarrow x, \Pi_1 \longrightarrow x, \Pi_1, \rho_1 \longrightarrow x, \Pi_1, \rho_1, \Pi_2 \longrightarrow \cdots \longrightarrow x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$$

Definition of RBR soundness $\epsilon^{\text{rbr}}_{\text{IOR}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\text{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

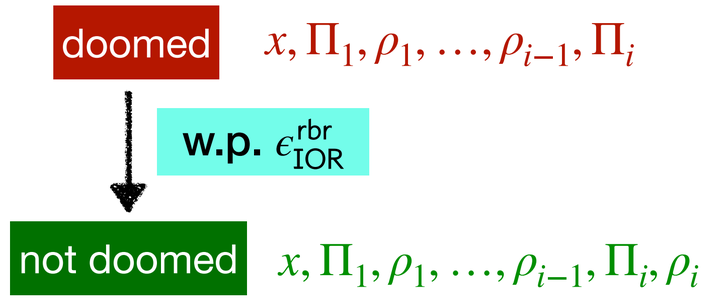
$$x \longrightarrow x, \Pi_1 \longrightarrow x, \Pi_1, \rho_1 \longrightarrow x, \Pi_1, \rho_1, \Pi_2 \longrightarrow \cdots \longrightarrow x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$$

Definition of RBR soundness $\epsilon^{\text{rbr}}_{\text{IOR}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\text{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

$$x \longrightarrow x, \Pi_1 \longrightarrow x, \Pi_1, \rho_1 \longrightarrow x, \Pi_1, \rho_1, \Pi_2 \longrightarrow \cdots \longrightarrow x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$$

$\mathscr{A}$: run $\tilde{P}^{\text{sr}}$ and compute $\rho_1, \cdots, \rho_k$ (at most $t + k$ classical queries);

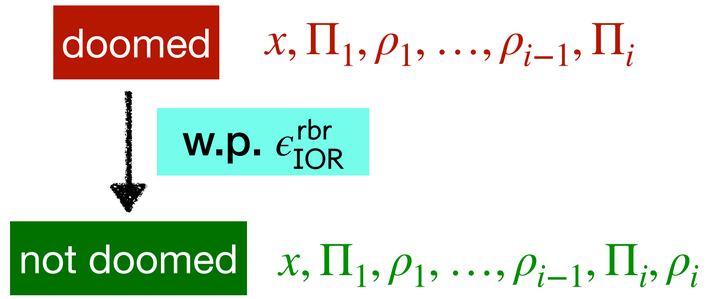
then $\tilde{P}^{\text{sr}}$ wins $\Rightarrow$ $\mathscr{A}$ can **find** $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i that jumps to not doomed.

Definition of RBR soundness $\epsilon_{\mathsf{IOR}}^{\mathsf{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\mathsf{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

$$x \longrightarrow x, \Pi_1 \longrightarrow x, \Pi_1, \rho_1 \longrightarrow x, \Pi_1, \rho_1, \Pi_2 \longrightarrow \cdots \longrightarrow x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$$

$\mathscr{A}$: run $\tilde{P}^{\mathsf{sr}}$ and compute $\rho_1, \cdots, \rho_k$ (at most $t + k$ classical queries);

then $\tilde{P}^{\mathsf{sr}}$ wins $\Rightarrow$ $\mathscr{A}$ can **find** $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i that jumps to not doomed.

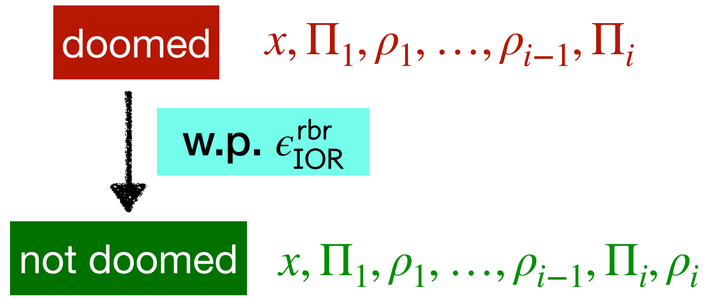
$\epsilon_{\mathsf{IOR}}^{\mathsf{sr}}(t) \leq \Pr[\mathscr{A}$ finds such $x, \Pi_1, \ldots, \rho_i]$ for a $(t + k)$-query $\mathscr{A}$

Definition of RBR soundness $\epsilon^{\text{rbr}}_{\text{IOR}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\text{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

$$x \longrightarrow x, \Pi_1 \longrightarrow x, \Pi_1, \rho_1 \longrightarrow x, \Pi_1, \rho_1, \Pi_2 \longrightarrow \cdots \longrightarrow x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$$

$\mathcal{A}$: run $\tilde{P}^{\text{sr}}$ and compute $\rho_1, \cdots, \rho_k$ (at most $t + k$ classical queries);

then $\tilde{P}^{\text{sr}}$ wins $\Rightarrow$ $\mathcal{A}$ can **find** $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i that jumps to not doomed.

$\epsilon^{\text{rbr}}_{\text{IOR}}$-sparse

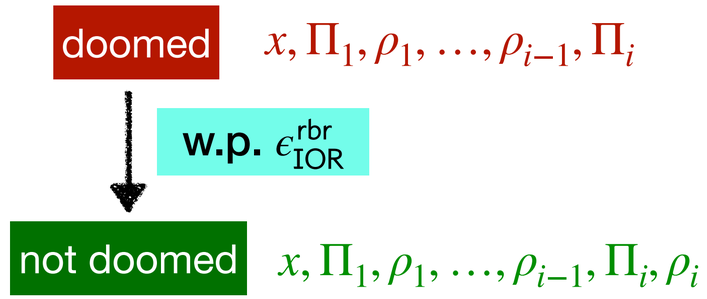
$\epsilon^{\text{sr}}_{\text{IOR}}(t) \leq \Pr[\mathcal{A}$ finds such $x, \Pi_1, \ldots, \rho_i]$ for a $(t + k)$-query $\mathcal{A}$

Definition of RBR soundness $\epsilon^{\text{rbr}}_{\text{IOR}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\text{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

$$x \longrightarrow x, \Pi_1 \longrightarrow x, \Pi_1, \rho_1 \longrightarrow x, \Pi_1, \rho_1, \Pi_2 \longrightarrow \cdots \longrightarrow x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$$

Search problem for some sparse set!

$\mathcal{A}$: run $\tilde{P}^{\text{sr}}$ and compute $\rho_1, \cdots, \rho_k$ (at most $t + k$ classical queries);

then $\tilde{P}^{\text{sr}}$ wins $\Rightarrow$ $\mathcal{A}$ can **find** $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i that jumps to not doomed.

$\epsilon^{\text{rbr}}_{\text{IOR}}$-sparse

$\epsilon^{\text{sr}}_{\text{IOR}}(t) \leq \Pr[\mathcal{A}$ finds such $x, \Pi_1, \ldots, \rho_i]$ for a $(t + k)$-query $\mathcal{A}$

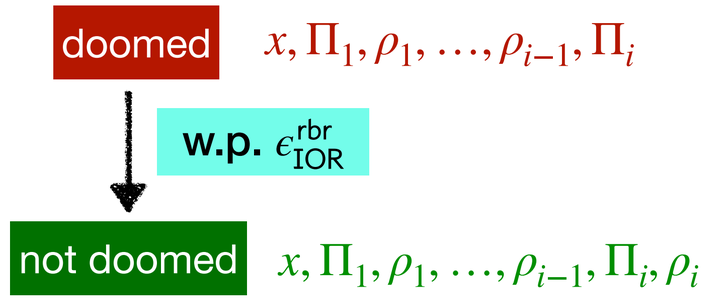
RBR soundness induces a search problem in the SR game

Definition of RBR soundness $\epsilon_{\mathsf{IOR}}^{\mathsf{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\mathsf{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

$$x \longrightarrow x, \Pi_1 \longrightarrow x, \Pi_1, \rho_1 \longrightarrow x, \Pi_1, \rho_1, \Pi_2 \longrightarrow \cdots \longrightarrow x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$$

Search problem for some sparse set!

$\mathcal{A}$: run $\tilde{P}^{\mathsf{sr}}$ and compute $\rho_1, \cdots, \rho_k$ (at most $t + k$ classical queries);

then $\tilde{P}^{\mathsf{sr}}$ wins $\Rightarrow$ $\mathcal{A}$ can **find** $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i that jumps to not doomed.

$\epsilon_{\mathsf{IOR}}^{\mathsf{rbr}}$-sparse

$\epsilon_{\mathsf{IOR}}^{\mathsf{sr}}(t) \leq \Pr[\mathcal{A}$ finds such $x, \Pi_1, \ldots, \rho_i]$ for a $(t + k)$-query $\mathcal{A}$

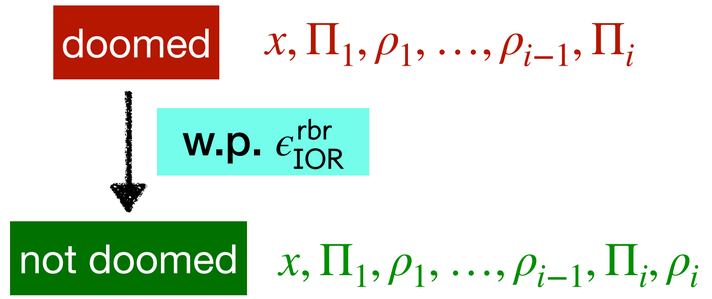
RBR soundness induces a search problem in the SR game

Definition of RBR soundness $\epsilon_{\mathsf{IOR}}^{\mathsf{rbr}}$:

Each partial transcript is labeled either

doomed — Almost impossible to make V output $x' \in L'$

or not doomed — Promising to make V output $x' \in L'$



To win SR game, $\tilde{P}^{\mathsf{sr}}$ needs to find $x, (\Pi_1, \ldots, \Pi_k)$ such that $x \notin L$ but V_{FS} outputs $x' \in L'$.

x is doomed, but $x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$ is not doomed.

$$x \longrightarrow x, \Pi_1 \longrightarrow x, \Pi_1, \rho_1 \longrightarrow x, \Pi_1, \rho_1, \Pi_2 \longrightarrow \cdots \longrightarrow x, \Pi_1, \rho_1, \ldots, \Pi_k, \rho_k$$

Search problem for some sparse set!

$\mathscr{A}$: run $\tilde{P}^{\mathsf{sr}}$ and compute $\rho_1, \cdots, \rho_k$ (at most $t + k$ classical queries);

then $\tilde{P}^{\mathsf{sr}}$ wins $\Rightarrow$ $\mathscr{A}$ can **find** $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i that jumps to not doomed.

$\epsilon_{\mathsf{IOR}}^{\mathsf{rbr}}$-sparse

$\epsilon_{\mathsf{IOR}}^{\mathsf{sr}}(t) \leq \Pr[\mathscr{A}$ finds such $x, \Pi_1, \ldots, \rho_i]$ for a $(t+k)$-query $\mathscr{A}$

$= O((t+k) \cdot \epsilon_{\mathsf{IOR}}^{\mathsf{rbr}})$

What happens in the quantum case?

What happens in the quantum case?

find $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i

$\epsilon_{\text{IOR}}^{\text{rbr}}$-sparse

Quantumly, the same search problem, but with quantum queries!

What happens in the quantum case?

find $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i

$\epsilon_{\text{IOR}}^{\text{rbr}}$-sparse

Quantumly, the same search problem, but with quantum queries!

$\epsilon_{\text{IOR}}^{\star,\text{sr}}(t) \leq \Pr[\mathscr{A}^{\star}$ finds such $x, \Pi_1, \ldots, \rho_i]$ for a $(t+k)$-query quantum $\mathscr{A}^{\star}$

What happens in the quantum case?

find $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i — $\epsilon_{\text{IOR}}^{\text{rbr}}$-sparse

Quantumly, the same search problem, but with quantum queries!

$\epsilon_{\text{IOR}}^{\star,\text{sr}}(t) \leq \Pr[\mathscr{A}^{\star}$ finds such $x, \Pi_1, \ldots, \rho_i]$ for a $(t+k)$-query quantum $\mathscr{A}^{\star}$

Search is faster in the quantum world: Grover's algorithm for multiple oracles

What happens in the quantum case?

find $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i — $\epsilon_{\text{IOR}}^{\text{rbr}}$-sparse

Quantumly, the same search problem, but with quantum queries!

$\epsilon_{\text{IOR}}^{\star,\text{sr}}(t) \leq \Pr[\mathscr{A}^\star \text{ finds such } x, \Pi_1, \ldots, \rho_i] \text{ for a } (t+k)\text{-query quantum } \mathscr{A}^\star$

Search is faster in the quantum world: Grover's algorithm for multiple oracles

We can find a preimage of a set S for some f_i w.p. $\Omega(T^2 \cdot \text{sparsity of } S)$ with T queries to $f_1, \ldots, f_k$.

What happens in the quantum case?

find $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i — $\epsilon_{\text{IOR}}^{\text{rbr}}$-sparse

Quantumly, the same search problem, but with quantum queries!

$\epsilon_{\text{IOR}}^{\star,\text{sr}}(t) \leq \Pr[\mathscr{A}^\star \text{ finds such } x, \Pi_1, \ldots, \rho_i] \text{ for a } (t+k)\text{-query quantum } \mathscr{A}^\star$

Search is faster in the quantum world: Grover's algorithm for multiple oracles

We can find a preimage of a set S for some f_i w.p. $\Omega(T^2 \cdot \text{sparsity of } S)$ with T queries to $f_1, \ldots, f_k$.

There's a limit to the speed up: Grover's optimality

What happens in the quantum case?

find $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i — $\epsilon_{\text{IOR}}^{\text{rbr}}$-sparse

Quantumly, the same search problem, but with quantum queries!

$\epsilon_{\text{IOR}}^{\star,\text{sr}}(t) \leq \Pr[\mathscr{A}^{\star}$ finds such $x, \Pi_1, \ldots, \rho_i]$ for a $(t+k)$-query quantum $\mathscr{A}^{\star}$

Search is faster in the quantum world: Grover's algorithm for multiple oracles

We can find a preimage of a set S for some f_i w.p. $\Omega(T^2 \cdot$ sparsity of $S)$ with T queries to $f_1, \ldots, f_k$.

There's a limit to the speed up: Grover's optimality

Every T-query $\mathscr{A}^{\star}$ can find a preimage of a set S for some f_i w.p. $O(T^2 \cdot$ sparsity of $S)$.

What happens in the quantum case?

find $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i

$\epsilon_{\text{IOR}}^{\text{rbr}}$-sparse

Quantumly, the same search problem, but with quantum queries!

$\epsilon_{\text{IOR}}^{\star,\text{sr}}(t) \leq \Pr[\mathscr{A}^\star \text{ finds such } x, \Pi_1, \ldots, \rho_i] \text{ for a } (t+k)\text{-query quantum } \mathscr{A}^\star$

Search is faster in the quantum world: Grover's algorithm for multiple oracles

We can find a preimage of a set S for some f_i w.p. $\Omega(T^2 \cdot \text{sparsity of } S)$ with T queries to $f_1, \ldots, f_k$.

There's a limit to the speed up: Grover's optimality

Every T-query $\mathscr{A}^\star$ can find a preimage of a set S for some f_i w.p. $O(T^2 \cdot \text{sparsity of } S)$.

Almost there…
But we are not searching ρ_i in a set S

What happens in the quantum case?

find $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i — $\epsilon^{\text{rbr}}_{\text{IOR}}$-sparse

Quantumly, the same search problem, but with quantum queries!

$\epsilon^{\star,\text{sr}}_{\text{IOR}}(t) \leq \Pr[\mathscr{A}^\star \text{ finds such } x, \Pi_1, \ldots, \rho_i] \text{ for a } (t+k)\text{-query quantum } \mathscr{A}^\star$

Search is faster in the quantum world: Grover's algorithm for multiple oracles

We can find a preimage of a set S for some f_i w.p. $\Omega(T^2 \cdot \text{sparsity of } S)$ with T queries to $f_1, \ldots, f_k$.

There's a limit to the speed up: Grover's optimality

Every T-query $\mathscr{A}^\star$ can find a preimage of a set S for some f_i w.p. $O(T^2 \cdot \text{sparsity of } S)$.

Almost there…
But we are not searching ρ_i in a set S

$\mathscr{A}^\star$ **needs** to find $q = (x, \Pi_1, \ldots, \Pi_i)$ and $\rho_i = f_i(q)$ s.t. $((x, \Pi_1, \rho_1, \ldots, \Pi_i), \rho_i)$ in a **relation**

What happens in the quantum case?

find $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i — $\epsilon_{\text{IOR}}^{\text{rbr}}$-sparse

Quantumly, the same search problem, but with quantum queries!

$\epsilon_{\text{IOR}}^{\star,\text{sr}}(t) \leq \Pr[\mathscr{A}^\star \text{ finds such } x, \Pi_1, \ldots, \rho_i] \text{ for a } (t+k)\text{-query quantum } \mathscr{A}^\star$

Search is faster in the quantum world: Grover's algorithm for multiple oracles

We can find a preimage of a set S for some f_i w.p. $\Omega(T^2 \cdot \text{sparsity of } S)$ with T queries to $f_1, \ldots, f_k$.

There's a limit to the speed up: Grover's optimality

Every T-query $\mathscr{A}^\star$ can find a preimage of a set S for some f_i w.p. $O(T^2 \cdot \text{sparsity of } S)$.

Almost there…
But we are not searching ρ_i in a set S

$\mathscr{A}^\star$ **needs** to find $q = (x, \Pi_1, \ldots, \Pi_i)$ and $\rho_i = f_i(q)$ s.t. $((x, \Pi_1, \rho_1, \ldots, \Pi_i), \rho_i)$ in a **relation**

Grover's optimality is also true for finding $(q, f_i(q)) \in R$ for sparse relation R.

What happens in the quantum case?

find $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i — $\epsilon^{\text{rbr}}_{\text{IOR}}$-sparse

Quantumly, the same search problem, but with quantum queries!

$\epsilon^{\star,\text{sr}}_{\text{IOR}}(t) \leq \Pr[\mathscr{A}^\star \text{ finds such } x, \Pi_1, \ldots, \rho_i] \text{ for a } (t+k)\text{-query quantum } \mathscr{A}^\star$

Search is faster in the quantum world: Grover's algorithm for multiple oracles

We can find a preimage of a set S for some f_i w.p. $\Omega(T^2 \cdot \text{sparsity of } S)$ with T queries to $f_1, \ldots, f_k$.

There's a limit to the speed up: Grover's optimality

Every T-query $\mathscr{A}^\star$ can find a preimage of a set S for some f_i w.p. $O(T^2 \cdot \text{sparsity of } S)$.

Almost there…
But we are not searching ρ_i in a set S

$\mathscr{A}^\star$ **needs** to find $q = (x, \Pi_1, \ldots, \Pi_i)$ and $\rho_i = f_i(q)$ s.t. $((x, \Pi_1, \rho_1, \ldots, \Pi_i), \rho_i)$ in a **relation**

Grover's optimality is also true for finding $(q, f_i(q)) \in R$ for sparse relation R.

But wait, we have $\rho_1, \ldots, \rho_{i-1}$ in the relation.

What happens in the quantum case?

find $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i — $\epsilon_{\text{IOR}}^{\text{rbr}}$-sparse

Quantumly, the same search problem, but with quantum queries!

$\epsilon_{\text{IOR}}^{\star,\text{sr}}(t) \leq \Pr[\mathscr{A}^\star \text{ finds such } x, \Pi_1, \ldots, \rho_i] \text{ for a } (t+k)\text{-query quantum } \mathscr{A}^\star$

Search is faster in the quantum world: Grover's algorithm for multiple oracles

We can find a preimage of a set S for some f_i w.p. $\Omega(T^2 \cdot \text{sparsity of } S)$ with T queries to $f_1, \ldots, f_k$.

There's a limit to the speed up: Grover's optimality

Every T-query $\mathscr{A}^\star$ can find a preimage of a set S for some f_i w.p. $O(T^2 \cdot \text{sparsity of } S)$.

Almost there…
But we are not searching ρ_i in a set S

$\mathscr{A}^\star$ **needs** to find $q = (x, \Pi_1, \ldots, \Pi_i)$ and $\rho_i = f_i(q)$ s.t. $((x, \Pi_1, \rho_1, \ldots, \Pi_i), \rho_i)$ in a **relation**

Grover's optimality is also true for finding $(q, f_i(q)) \in R$ for sparse relation R.

But wait, we have $\rho_1, \ldots, \rho_{i-1}$ in the relation.

Our solution: fix $f_1, \ldots, f_{i-1}$ when analyzing for f_i, then it's searching $(q, f_i(q)) \in R'$ for $\epsilon_{\text{IOR}}^{\text{rbr}}$-sparse R'.

What happens in the quantum case?

find $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i — $\epsilon_{\text{IOR}}^{\text{rbr}}$-sparse

Quantumly, the same search problem, but with quantum queries!

$\epsilon_{\text{IOR}}^{\star,\text{sr}}(t) \leq \Pr[\mathscr{A}^\star \text{ finds such } x, \Pi_1, \ldots, \rho_i] \text{ for a } (t+k)\text{-query quantum } \mathscr{A}^\star$

Search is faster in the quantum world: Grover's algorithm for multiple oracles

We can find a preimage of a set S for some f_i w.p. $\Omega(T^2 \cdot \text{sparsity of } S)$ with T queries to $f_1, \ldots, f_k$.

There's a limit to the speed up: Grover's optimality

Every T-query $\mathscr{A}^\star$ can find a preimage of a set S for some f_i w.p. $O(T^2 \cdot \text{sparsity of } S)$.

Almost there…
But we are not searching ρ_i in a set S

$\mathscr{A}^\star$ **needs** to find $q = (x, \Pi_1, \ldots, \Pi_i)$ and $\rho_i = f_i(q)$ s.t. $((x, \Pi_1, \rho_1, \ldots, \Pi_i), \rho_i)$ in a **relation**

Grover's optimality is also true for finding $(q, f_i(q)) \in R$ for sparse relation R.

But wait, we have $\rho_1, \ldots, \rho_{i-1}$ in the relation.

Proof uses instability lemma from [CMS19].

Our solution: fix $f_1, \ldots, f_{i-1}$ when analyzing for f_i, then it's searching $(q, f_i(q)) \in R'$ for $\epsilon_{\text{IOR}}^{\text{rbr}}$-sparse R'.

What happens in the quantum case?

find $x, \Pi_1, \rho_1, \ldots, \Pi_i$ and ρ_i — $\epsilon^{\text{rbr}}_{\text{IOR}}$-sparse

Quantumly, the same search problem, but with quantum queries!

$\epsilon^{\star,\text{sr}}_{\text{IOR}}(t) \leq \Pr[\mathscr{A}^\star \text{ finds such } x, \Pi_1, \ldots, \rho_i] \text{ for a } (t+k)\text{-query quantum } \mathscr{A}^\star$

$= O((t+k)^2\, \epsilon^{\text{rbr}}_{\text{IOR}})$

Search is faster in the quantum world: Grover's algorithm for multiple oracles

We can find a preimage of a set S for some f_i w.p. $\Omega(T^2 \cdot \text{sparsity of } S)$ with T queries to $f_1, \ldots, f_k$.

There's a limit to the speed up: Grover's optimality

Every T-query $\mathscr{A}^\star$ can find a preimage of a set S for some f_i w.p. $O(T^2 \cdot \text{sparsity of } S)$.

Almost there…
But we are not searching ρ_i in a set S

$\mathscr{A}^\star$ **needs** to find $q = (x, \Pi_1, \ldots, \Pi_i)$ and $\rho_i = f_i(q)$ s.t. $((x, \Pi_1, \rho_1, \ldots, \Pi_i), \rho_i)$ in a **relation**

Grover's optimality is also true for finding $(q, f_i(q)) \in R$ for sparse relation R.

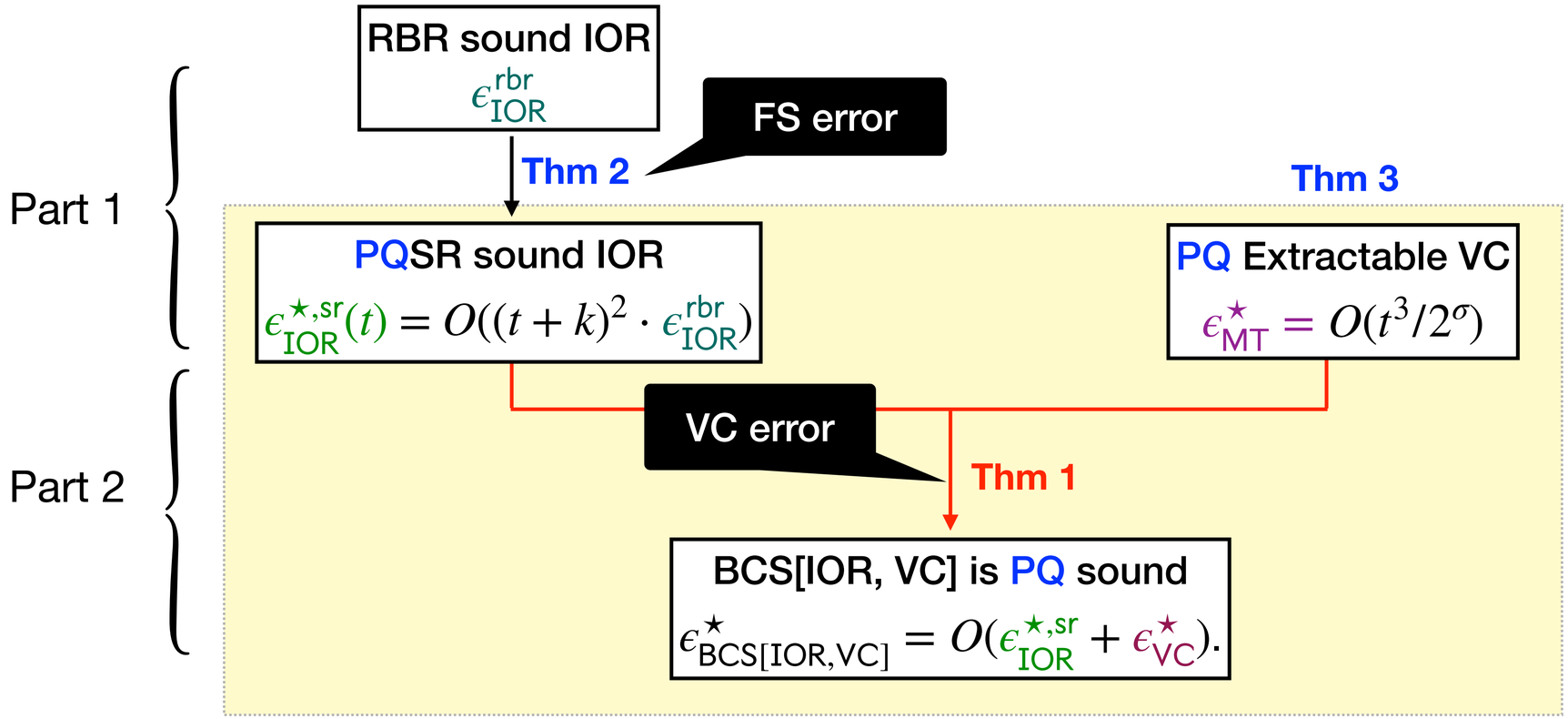
But wait, we have $\rho_1, \ldots, \rho_{i-1}$ in the relation.

Proof uses instability lemma from [CMS19].

Our solution: fix $f_1, \ldots, f_{i-1}$ when analyzing for f_i, then it's searching $(q, f_i(q)) \in R'$ for $\epsilon^{\text{rbr}}_{\text{IOR}}$-sparse R'.

Part 2: From PQSR IOR to PQ NRDX

BCS PQ soundness = PQSR soundness + VC PQ error



Putting it together: $\epsilon^{\star}_{\text{BCS[IOR,MT]}} = O((t+k)^2 \cdot \epsilon^{\text{rbr}}_{\text{IOR}}) + O(t^3/2^{\sigma})$

What happens in the classical case?

Classical case

Goal: we want to construct a SR prover $\tilde{P}^{\text{sr}}$ such that

$$\Pr[\tilde{P}^{\text{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\text{VC}}$$

Goal: we want to construct a SR prover $\tilde{P}^{\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

A construction: $\tilde{P}^{\mathsf{sr}}$ **simulates** $\tilde{P}$.

Goal: we want to construct a SR prover $\tilde{P}^{\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

A construction: $\tilde{P}^{\mathsf{sr}}$ simulates $\tilde{P}$.

Malicious BCS prover

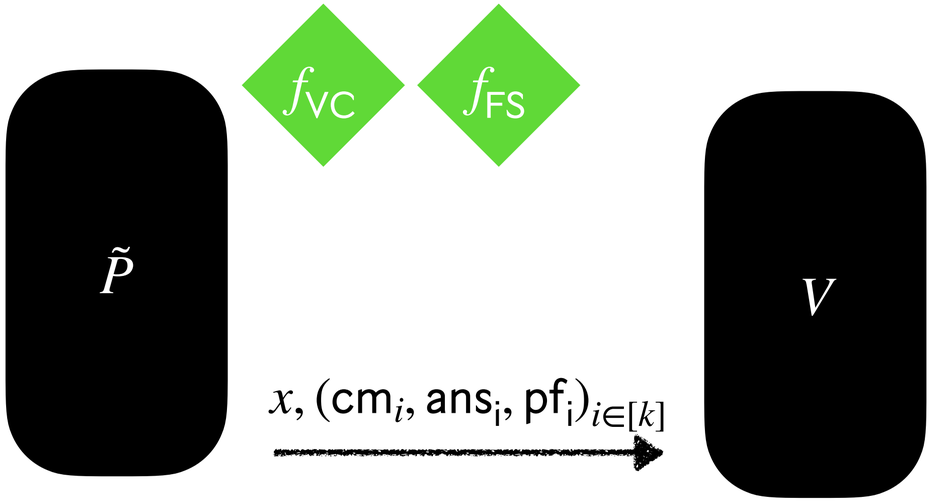
Classical case

Goal: we want to construct a SR prover $\tilde{P}^{\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

A construction: $\tilde{P}^{\mathsf{sr}}$ simulates $\tilde{P}$.

Malicious BCS prover

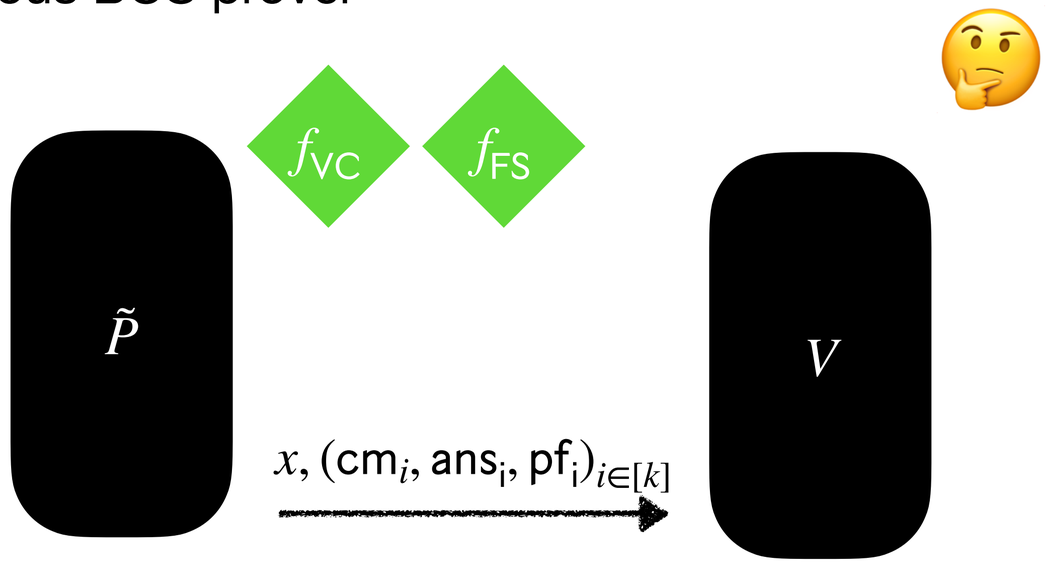


Goal: we want to construct a SR prover $\tilde{P}^{\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

A construction: $\tilde{P}^{\mathsf{sr}}$ simulates $\tilde{P}$.

Malicious BCS prover

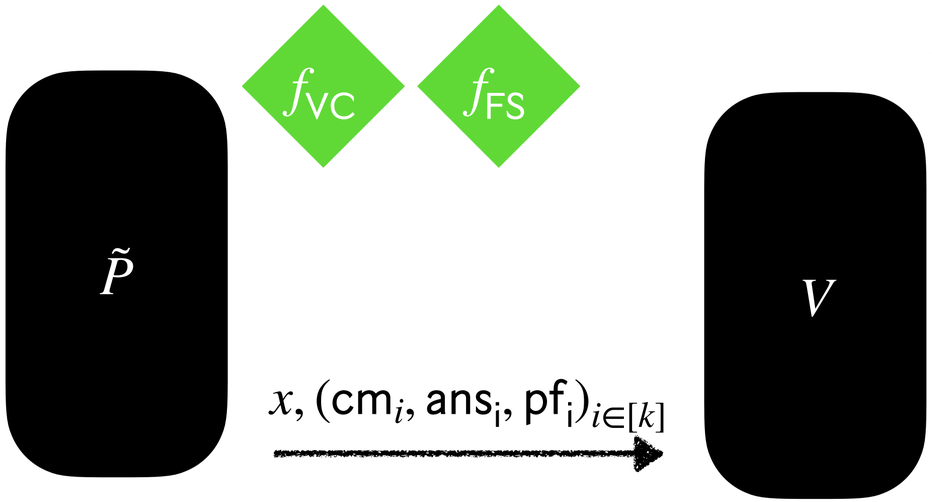


Goal: we want to construct a SR prover $\tilde{P}^{\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

A construction: $\tilde{P}^{\mathsf{sr}}$ simulates $\tilde{P}$.

Malicious BCS prover



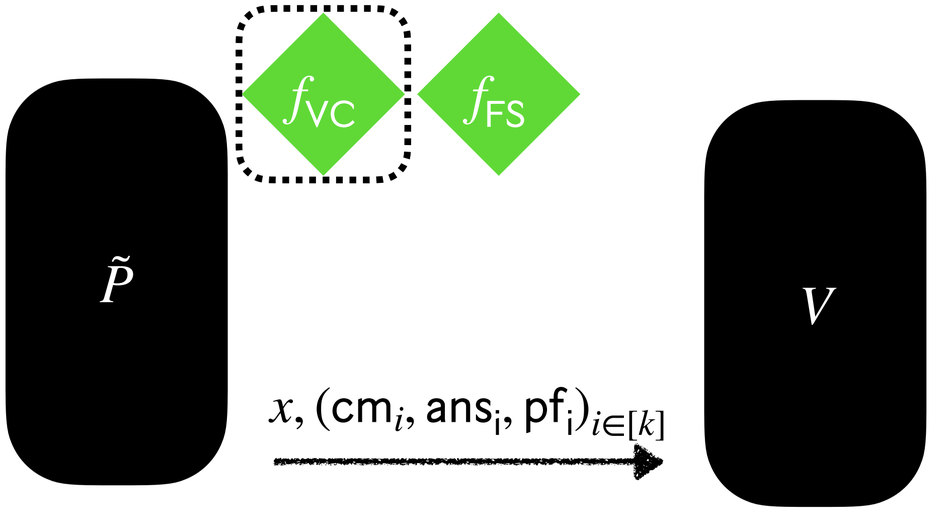
🤔 **How to…**

Goal: we want to construct a SR prover $\tilde{P}^{\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

A construction: $\tilde{P}^{\mathsf{sr}}$ simulates $\tilde{P}$.

Malicious BCS prover



🤔 **How to…**

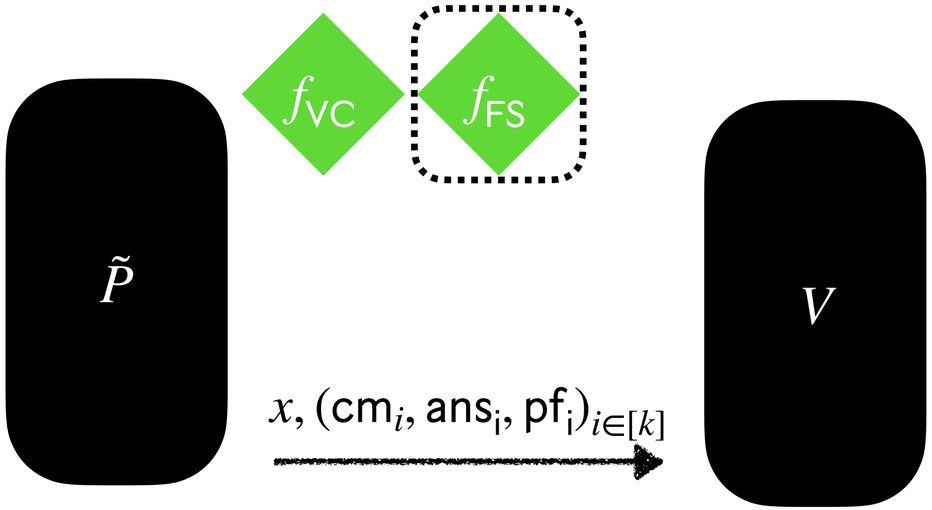
1. Answer f_{VC} queries?

Goal: we want to construct a SR prover $\tilde{P}^{\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

A construction: $\tilde{P}^{\mathsf{sr}}$ simulates $\tilde{P}$.

Malicious BCS prover



🤔 **How to...**

1. Answer f_{VC} queries?
2. Answer f_{FS} queries?

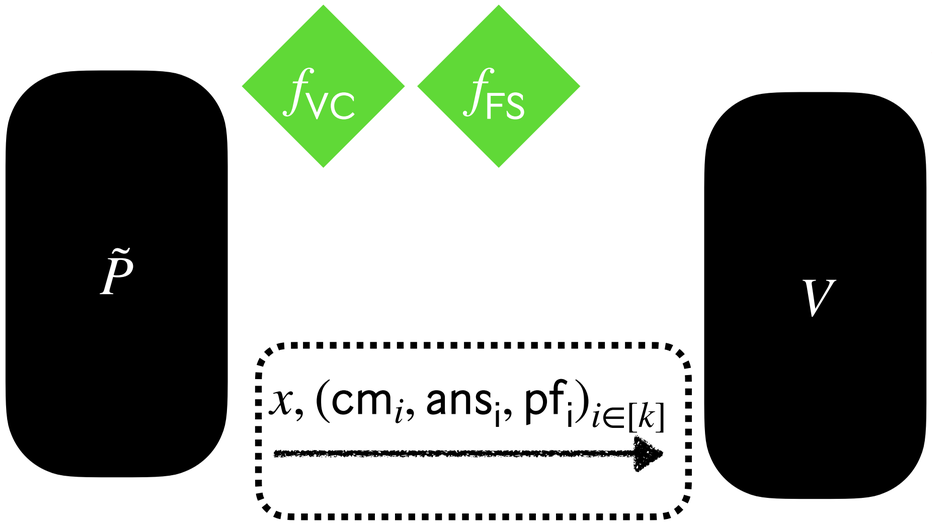
Classical case

Goal: we want to construct a SR prover $\tilde{P}^{\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

A construction: $\tilde{P}^{\mathsf{sr}}$ simulates $\tilde{P}$.

Malicious BCS prover



🤔 **How to...**

1. Answer f_{VC} queries?
2. Answer f_{FS} queries?
3. Derive the output of $\tilde{P}^{\mathsf{sr}}$ from the output of $\tilde{P}$?

Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

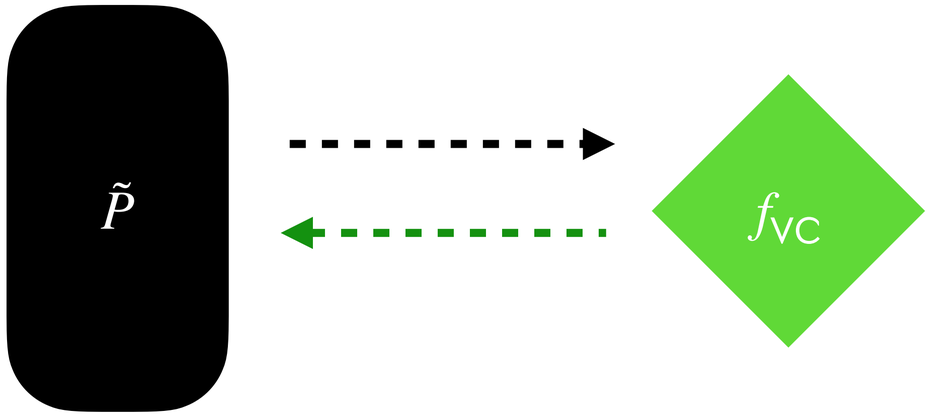
Step 1: how to answer f_{VC} queries?

Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 1: how to answer f_{VC} queries?

Malicious BCS prover

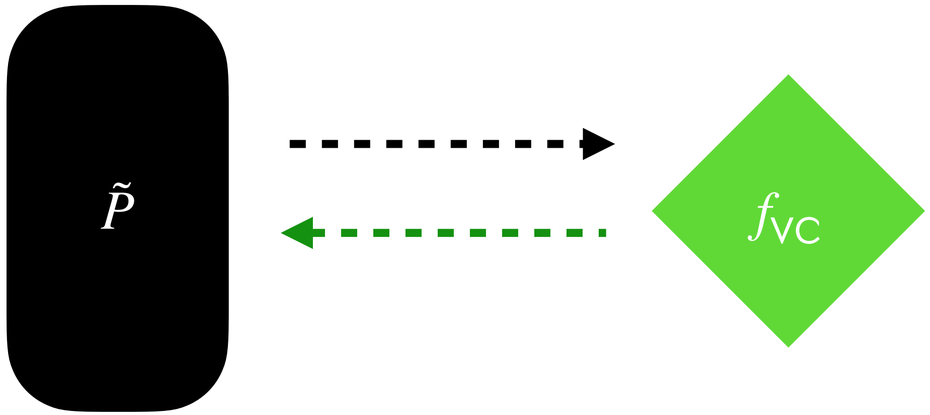


Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 1: how to answer f_{VC} queries?

Malicious BCS prover



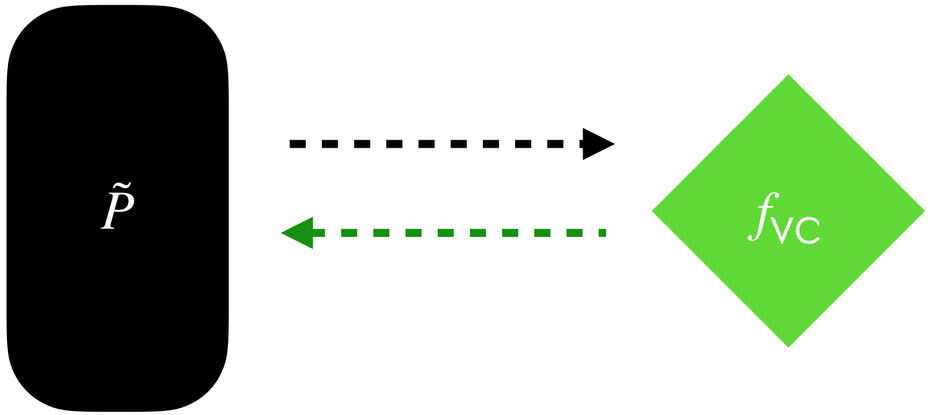
Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 1: how to answer f_{VC} queries?

$\tilde{P}^{\mathsf{sr}}$ does not have oracle access to f_{VC}

Malicious BCS prover

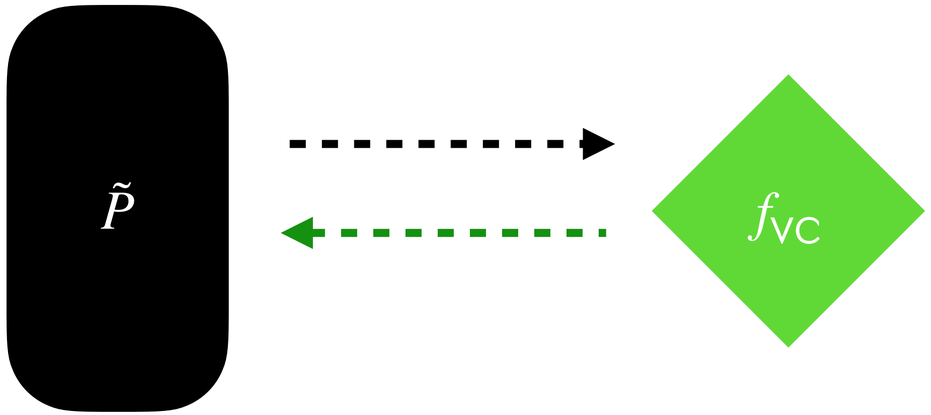


Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

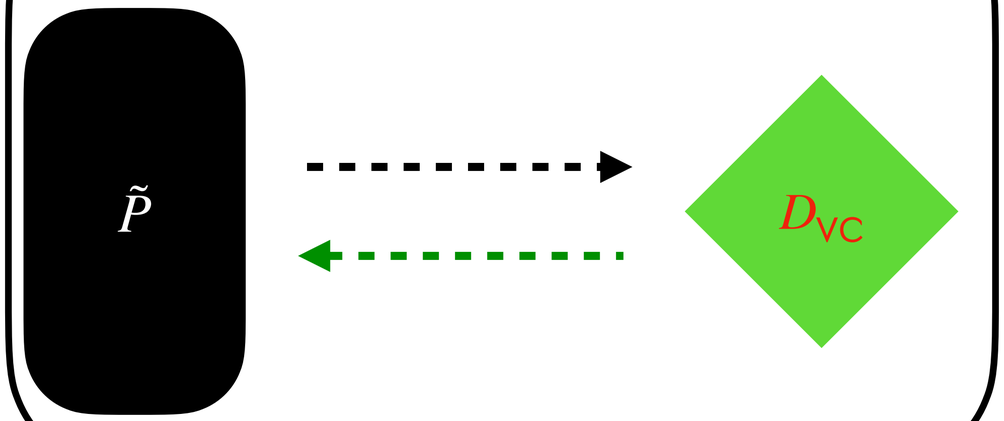
Step 1: how to answer f_{VC} queries?

Malicious BCS prover



$\tilde{P}^{\mathsf{sr}}$ does not have oracle access to f_{VC}

Malicious SR prover $\tilde{P}^{\mathsf{sr}}$



Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 2: how to answer f_{FS} queries?

Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 2: how to answer f_{FS} queries?

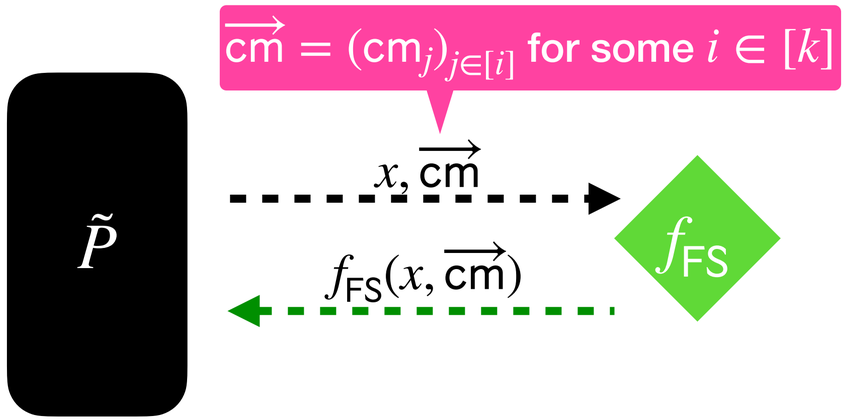
A natural attempt

Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 2: how to answer f_{FS} queries?

A natural attempt

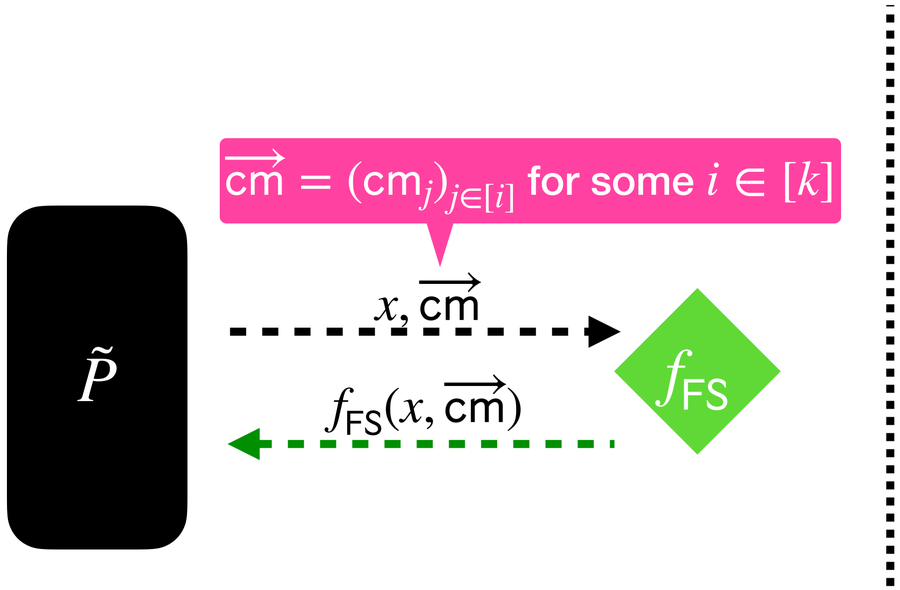


Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 2: how to answer f_{FS} queries?

A natural attempt



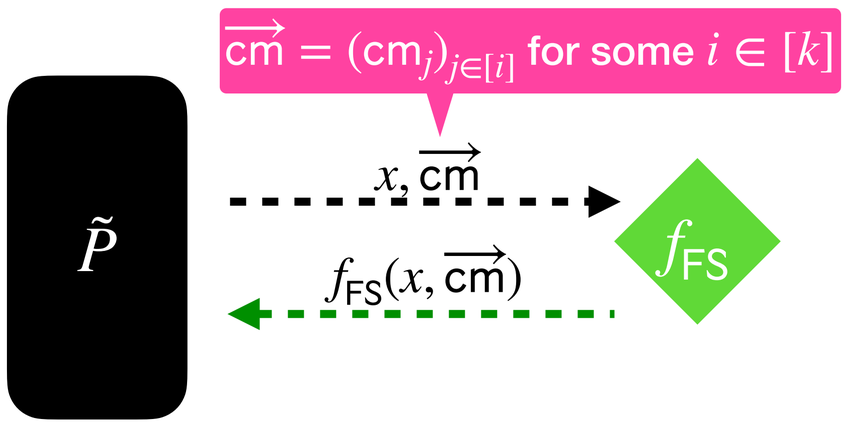
Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 2: how to answer f_{FS} queries?

A natural attempt

$\tilde{P}^{\mathsf{sr}}$ needs to query f_{FS} on IOR strings



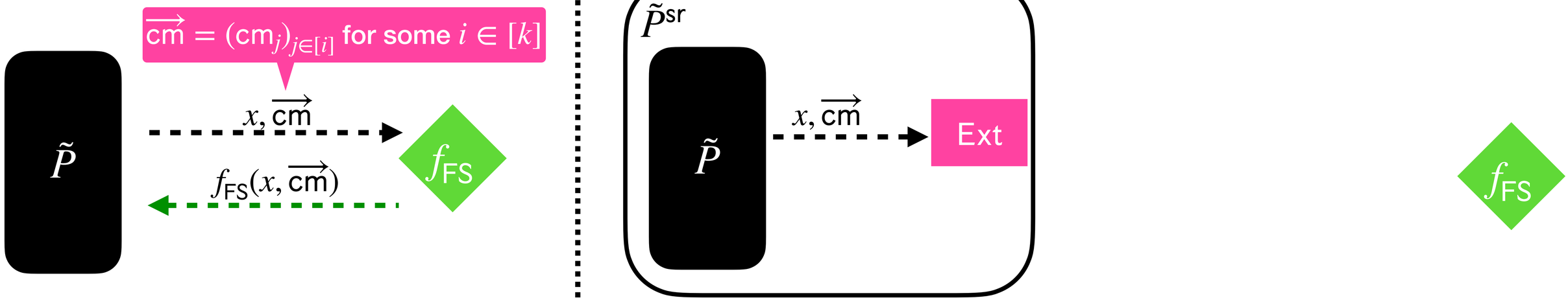
Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 2: how to answer f_{FS} queries?

A natural attempt

$\tilde{P}^{\mathsf{sr}}$ needs to query f_{FS} on IOR strings



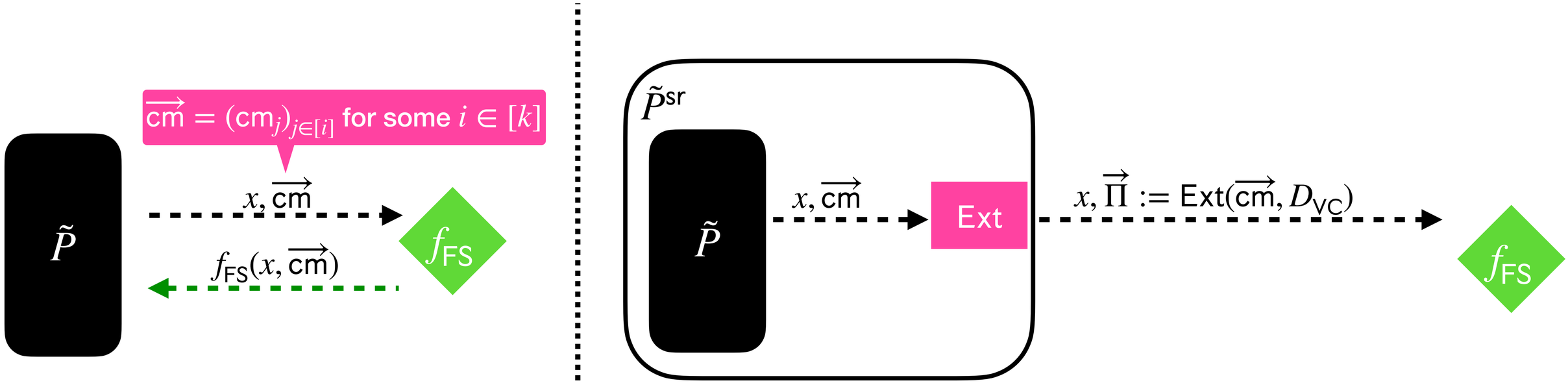
Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 2: how to answer f_{FS} queries?

A natural attempt

$\tilde{P}^{\mathsf{sr}}$ needs to query f_{FS} on IOR strings



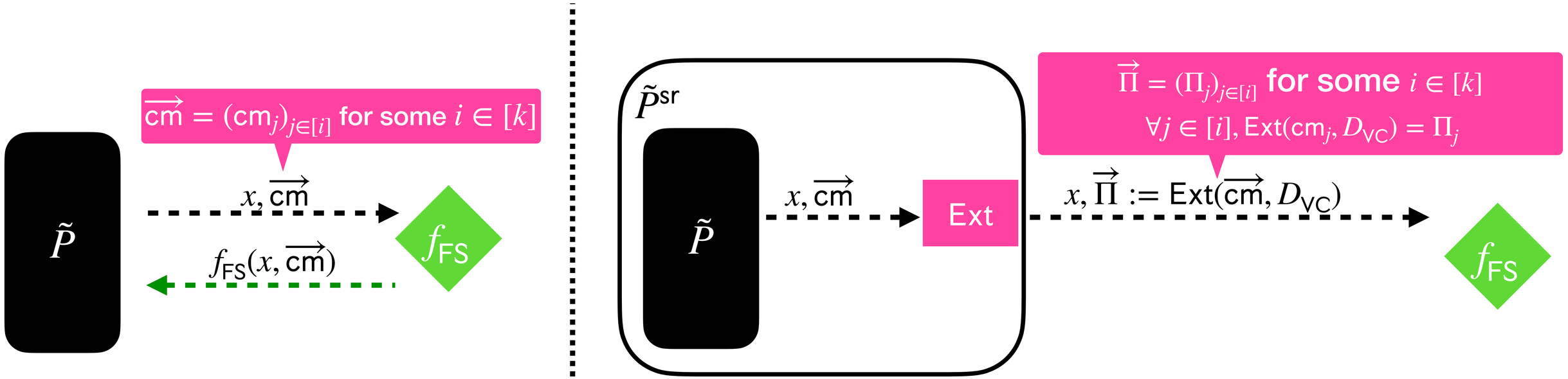
Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 2: how to answer f_{FS} queries?

A natural attempt

$\tilde{P}^{\mathsf{sr}}$ needs to query f_{FS} on IOR strings



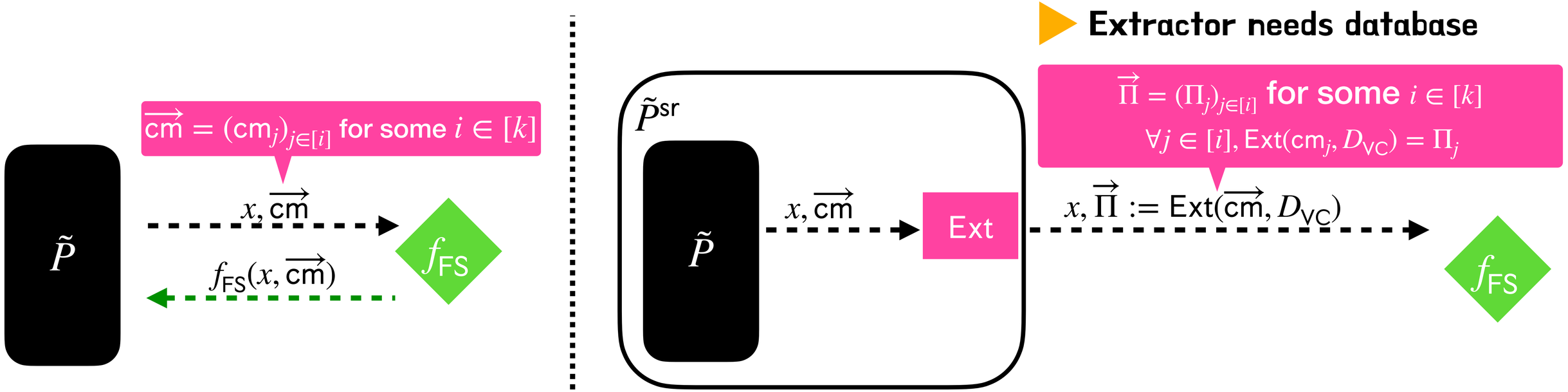
Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 2: how to answer f_{FS} queries?

A natural attempt

$\tilde{P}^{\mathsf{sr}}$ needs to query f_{FS} on IOR strings



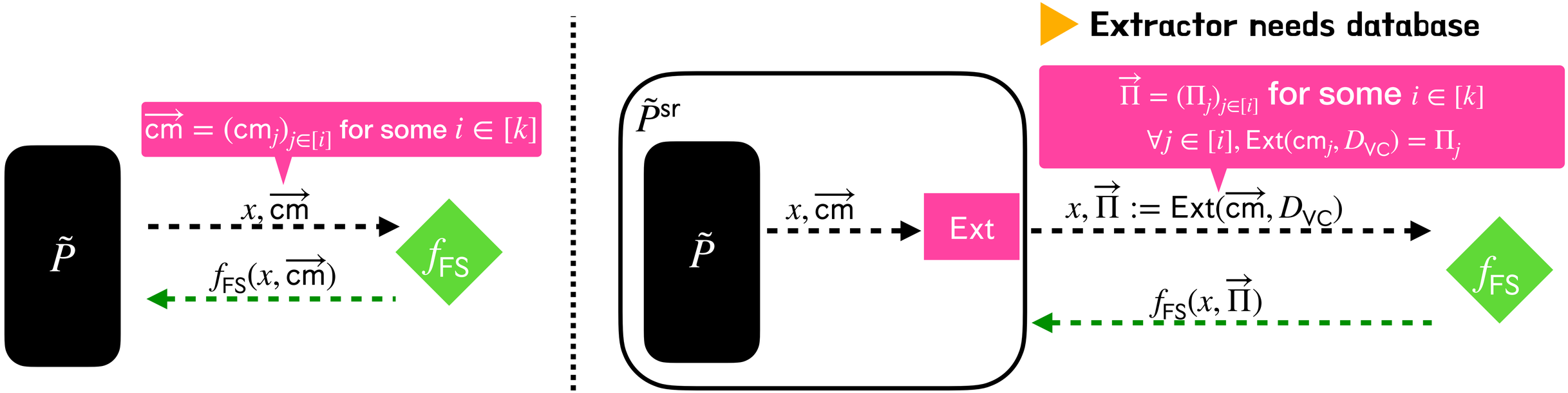
Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 2: how to answer f_{FS} queries?

A natural attempt

$\tilde{P}^{\mathsf{sr}}$ needs to query f_{FS} on IOR strings



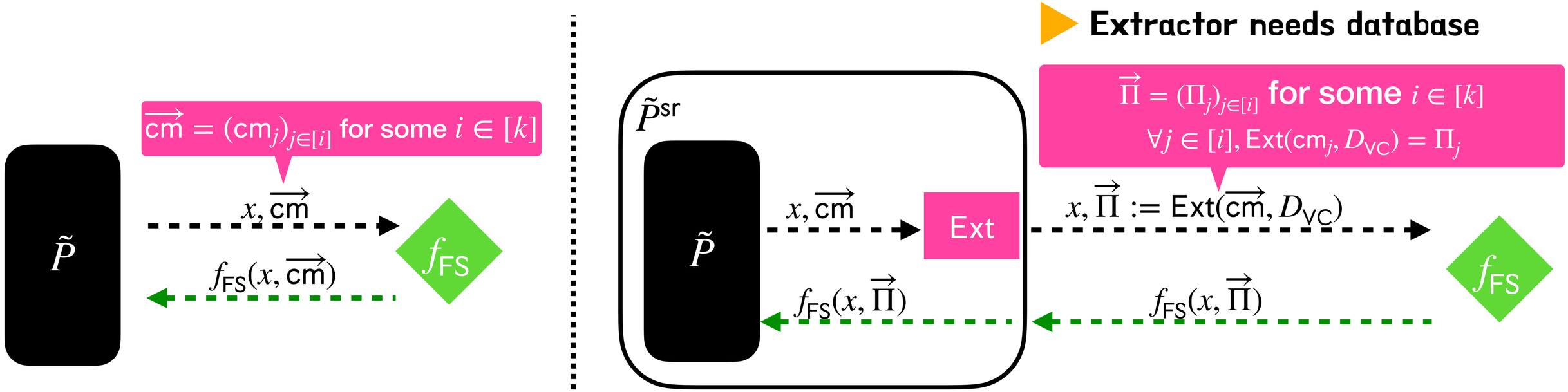
Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 2: how to answer f_{FS} queries?

A natural attempt

$\tilde{P}^{\mathsf{sr}}$ needs to query f_{FS} on IOR strings



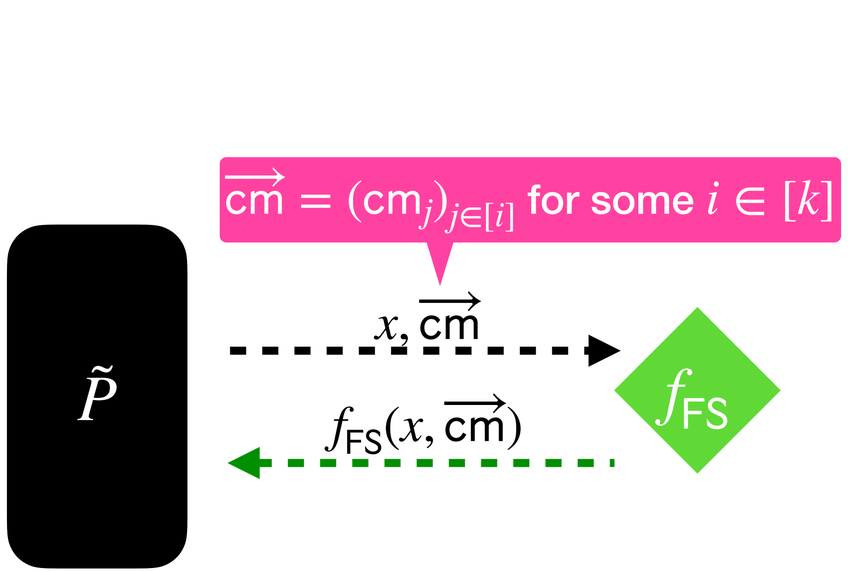
Construction of $\tilde{P}^{\mathsf{sr}}$

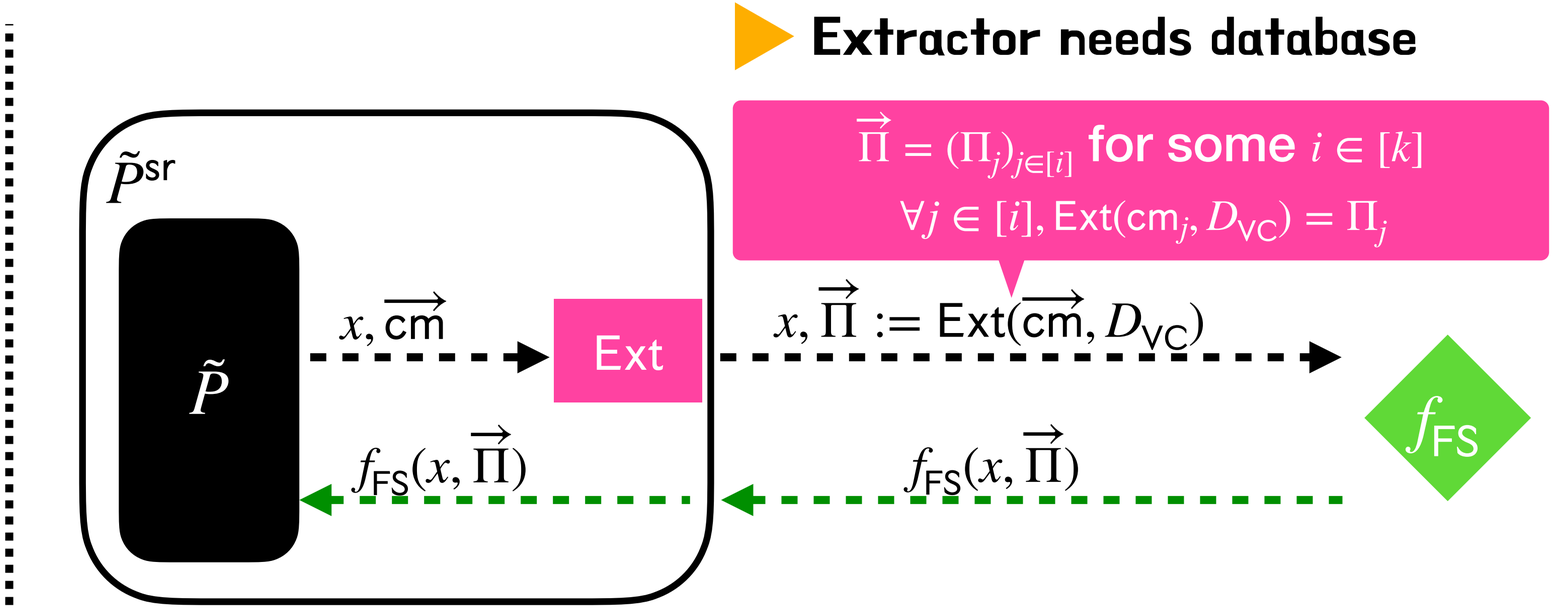
Classical case

Step 2: how to answer f_{FS} queries?

A natural attempt

$\tilde{P}^{\mathsf{sr}}$ needs to query f_{FS} on IOR strings





But $\tilde{P}$ can query $\mathsf{cm}_1 \neq \mathsf{cm}_1'$, with the same underlying message Π_1.

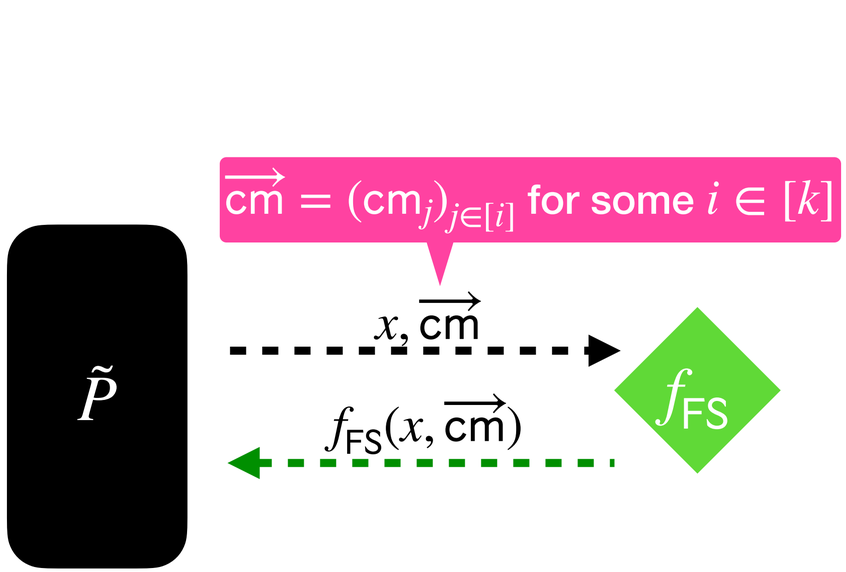
Construction of $\tilde{P}^{\mathsf{sr}}$

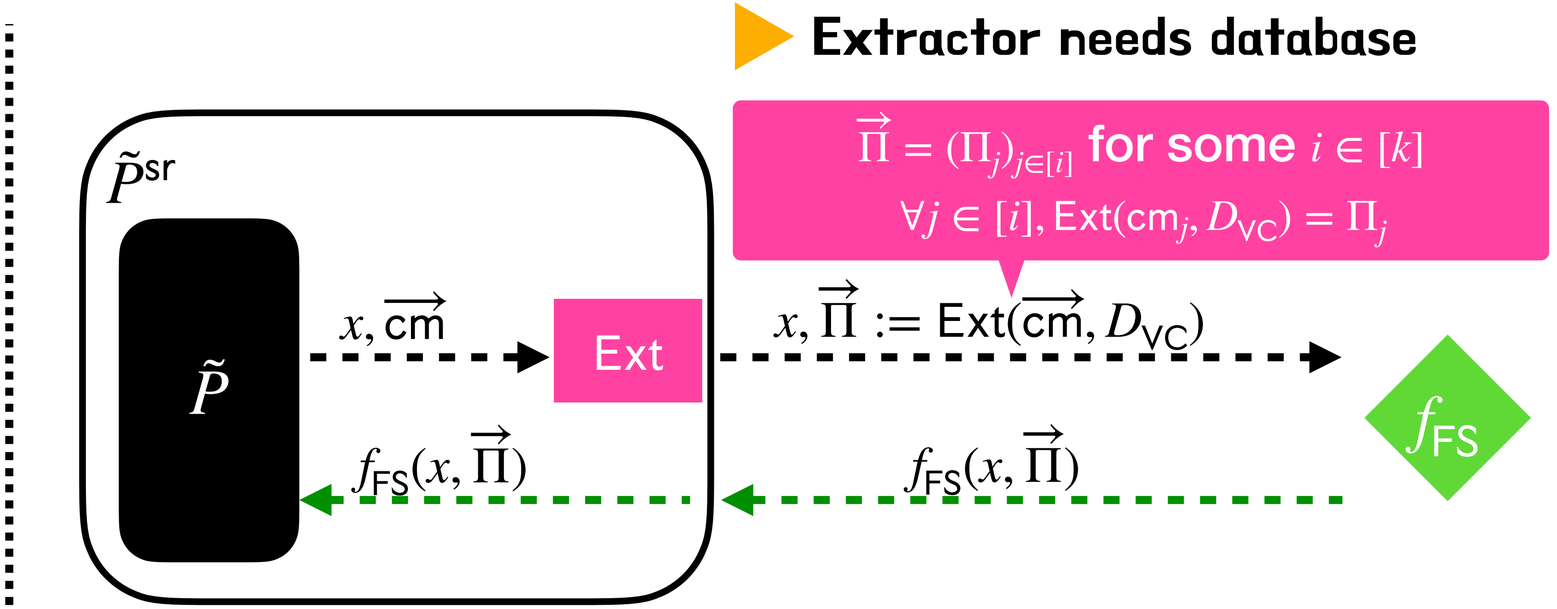
Classical case

Step 2: how to answer f_{FS} queries?

Instead...

$\tilde{P}^{\mathsf{sr}}$ needs to query f_{FS} on IOR strings





But $\tilde{P}$ can query $\mathsf{cm}_1 \neq \mathsf{cm}_1'$, with the same underlying message Π_1.

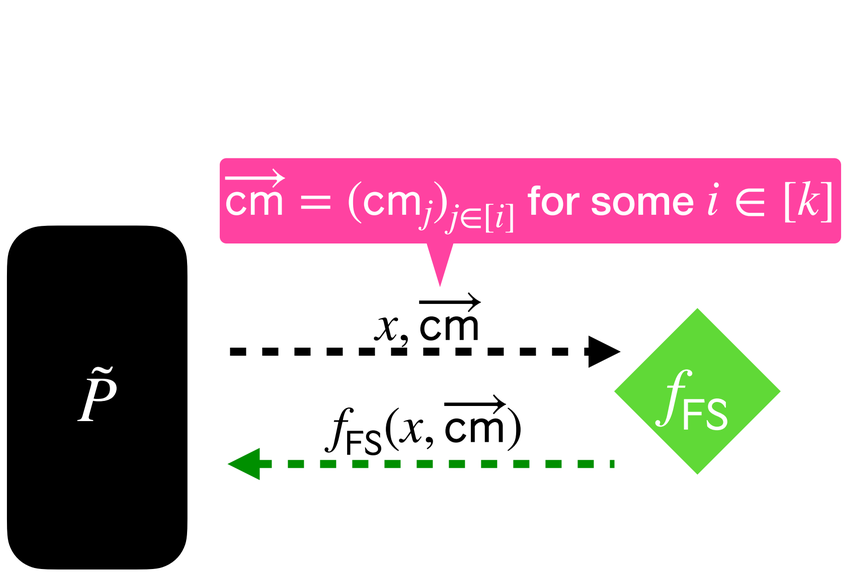
Construction of $\tilde{P}^{\mathsf{sr}}$

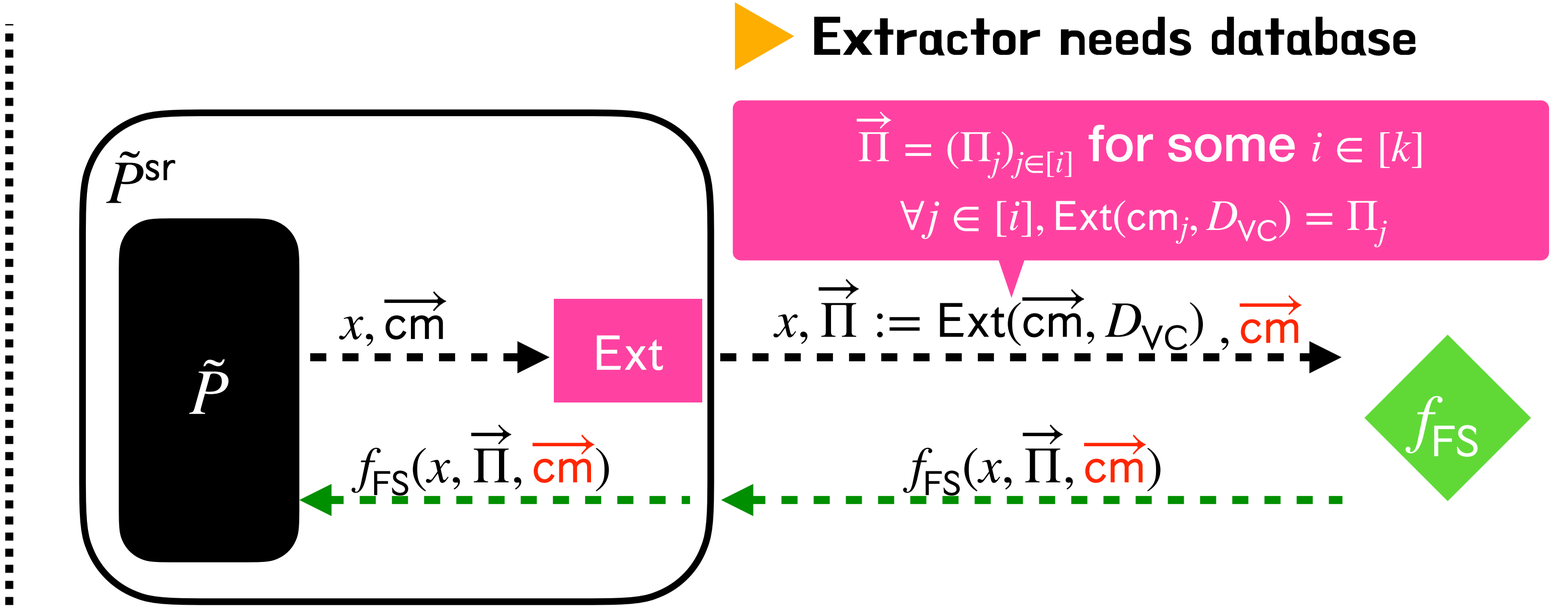
Classical case

Step 2: how to answer f_{FS} queries?

Instead...

$\tilde{P}^{\mathsf{sr}}$ needs to query f_{FS} on IOR strings





But $\tilde{P}$ can query $\mathsf{cm}_1 \neq \mathsf{cm}_1'$, with the same underlying message Π_1.

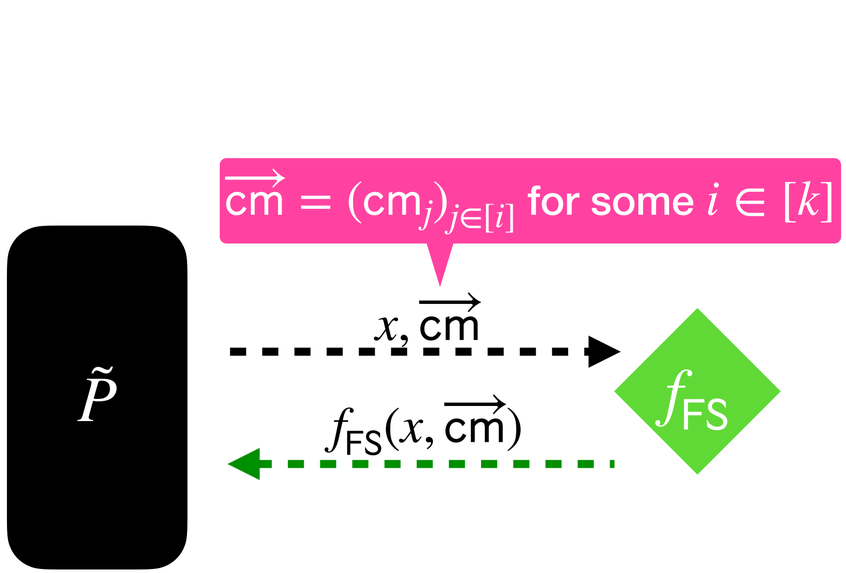
Construction of $\tilde{P}^{\mathsf{sr}}$

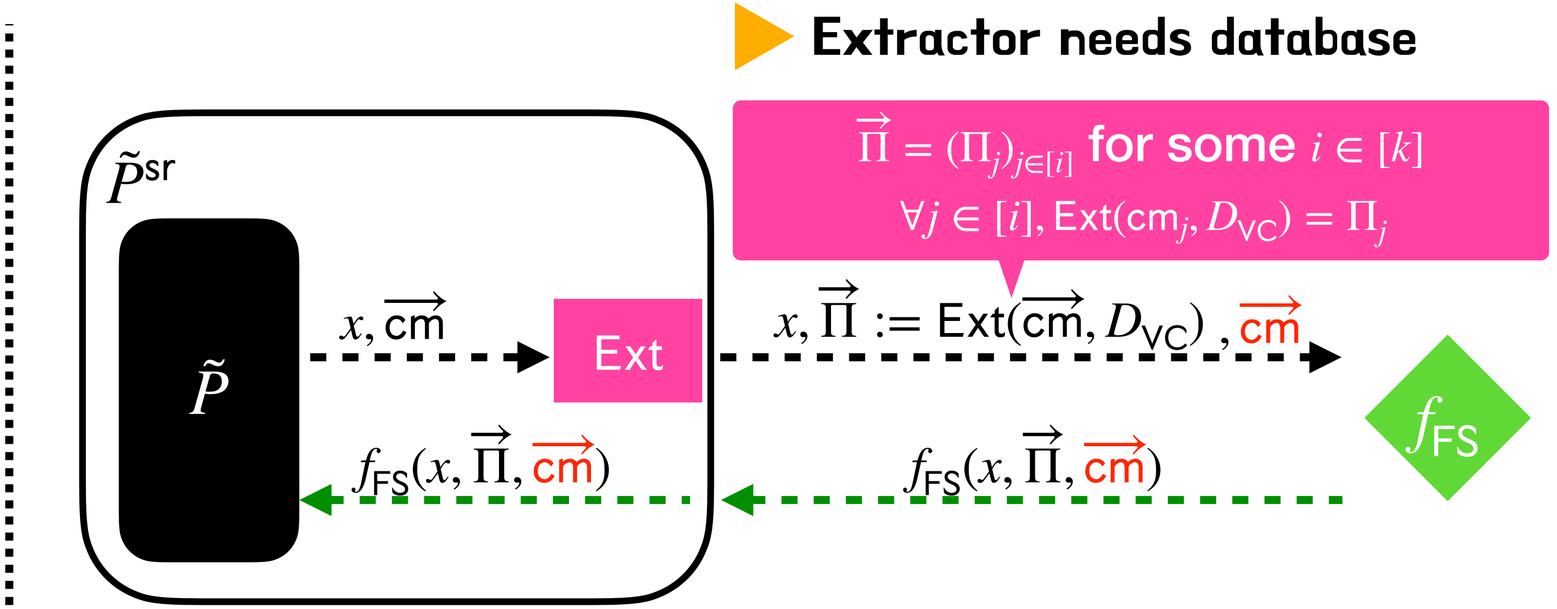
Classical case

Step 2: how to answer f_{FS} queries?

Instead...

$\tilde{P}^{\mathsf{sr}}$ needs to query f_{FS} on IOR strings





But $\tilde{P}$ can query $\mathsf{cm}_1 \neq \mathsf{cm}_1'$, with the same underlying message Π_1.

Omitted: actual PQSR definition includes salt

Construction of $\tilde{P}^{\mathsf{sr}}$

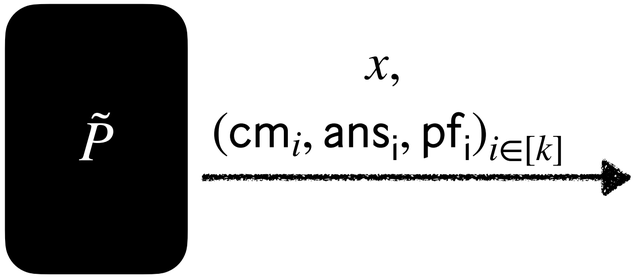
Classical case

Step 3: how to derive the output

Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

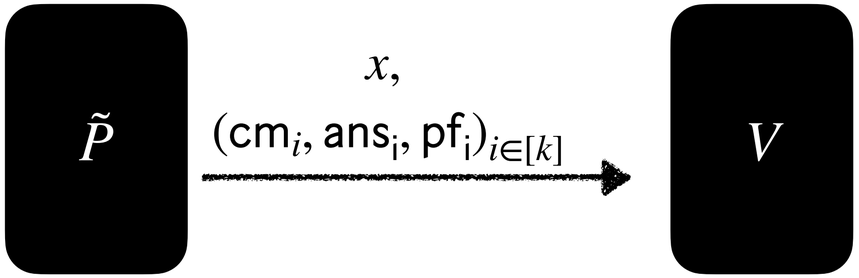
Step 3: how to derive the output



Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

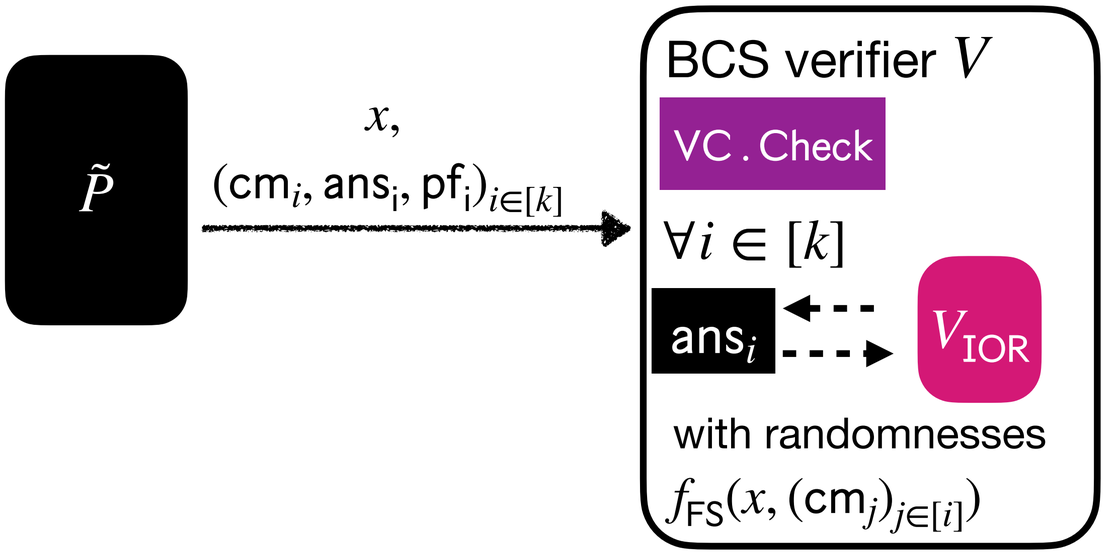
Step 3: how to derive the output



Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

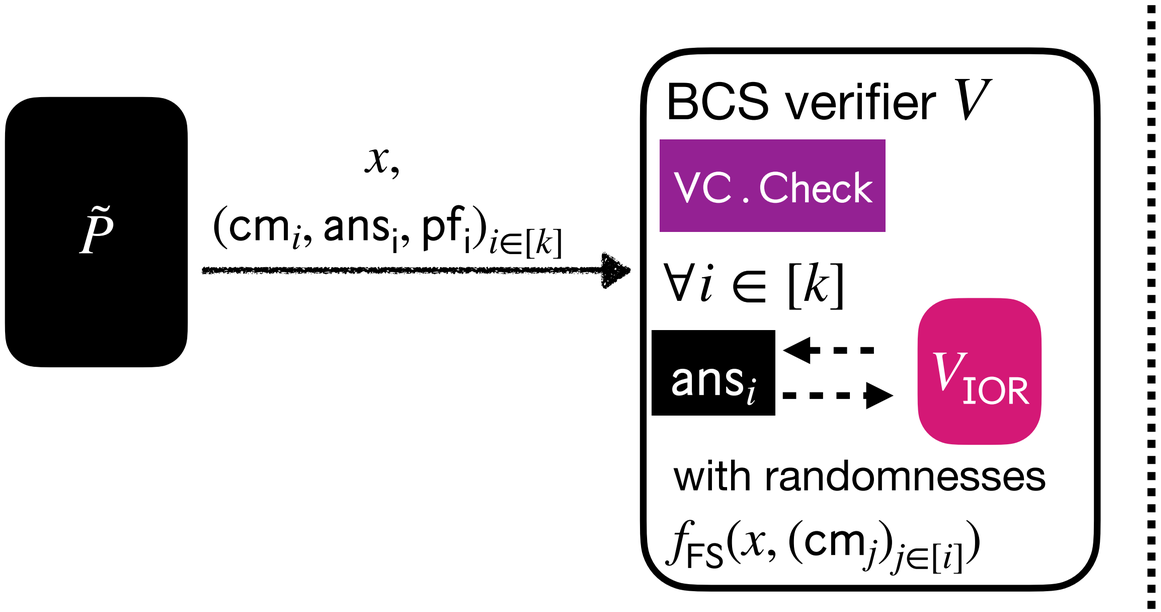
Step 3: how to derive the output



Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

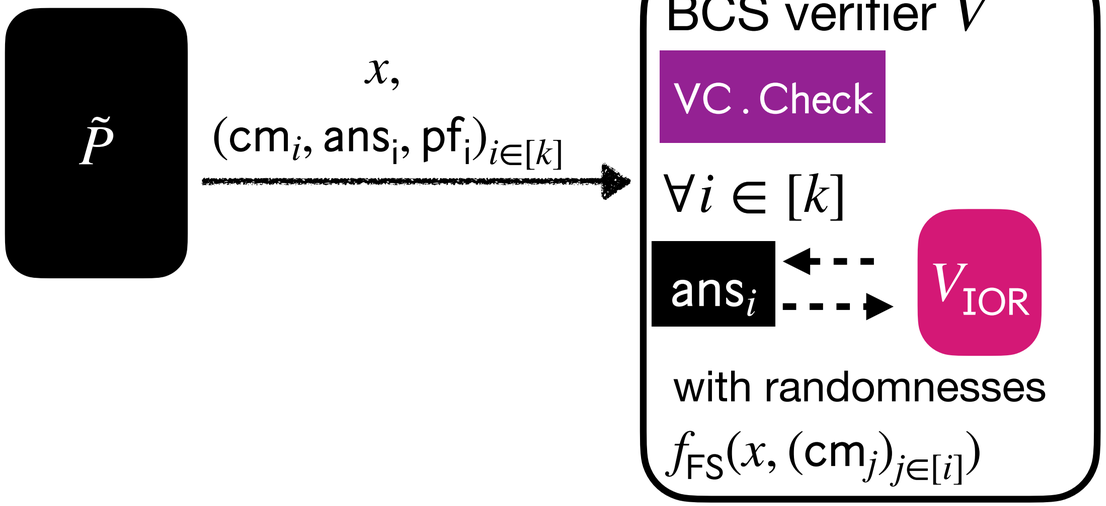
Step 3: how to derive the output



Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 3: how to derive the output



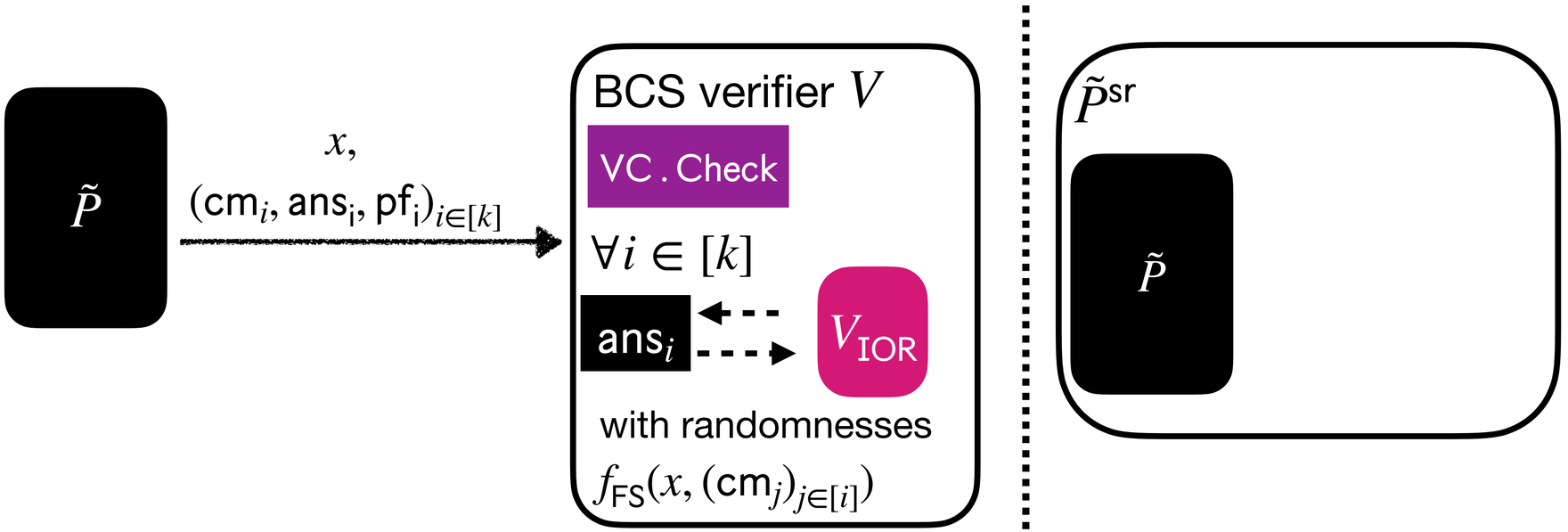
$\tilde{P}^{\mathsf{sr}}$ needs to output the chosen IOR strings

Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 3: how to derive the output

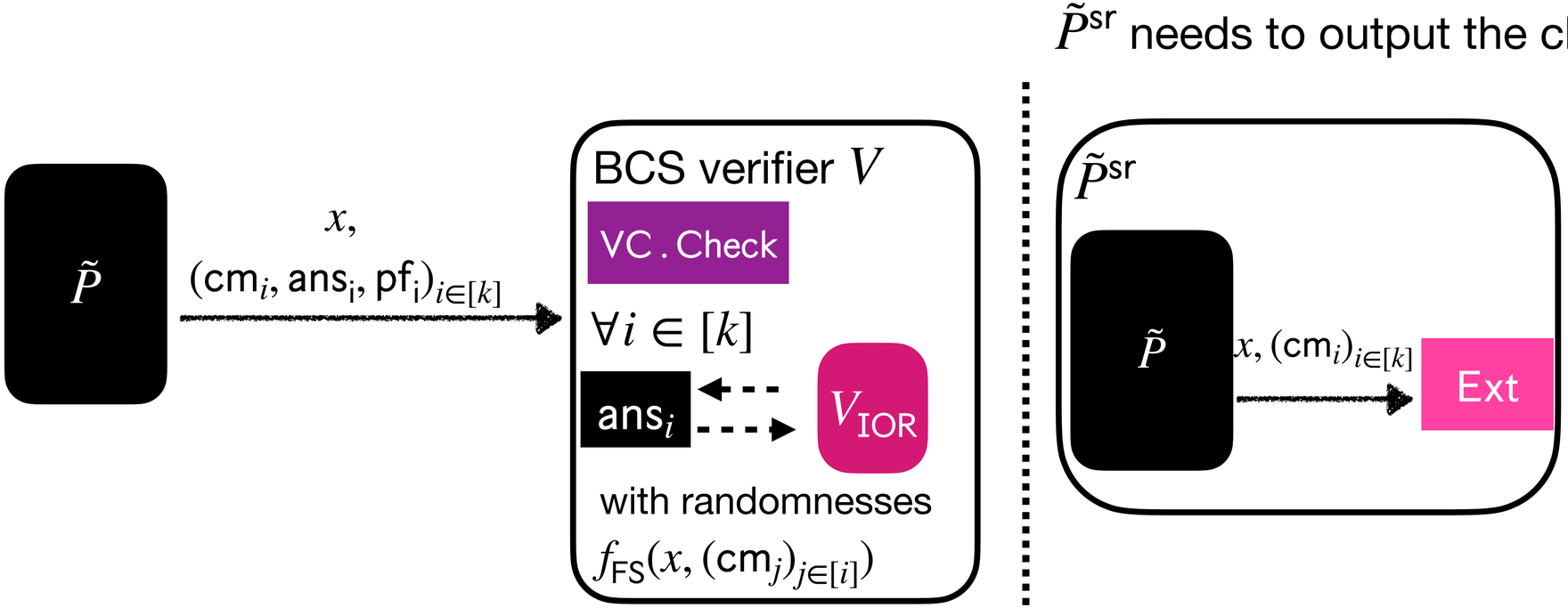
$\tilde{P}^{\mathsf{sr}}$ needs to output the chosen IOR strings



Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 3: how to derive the output



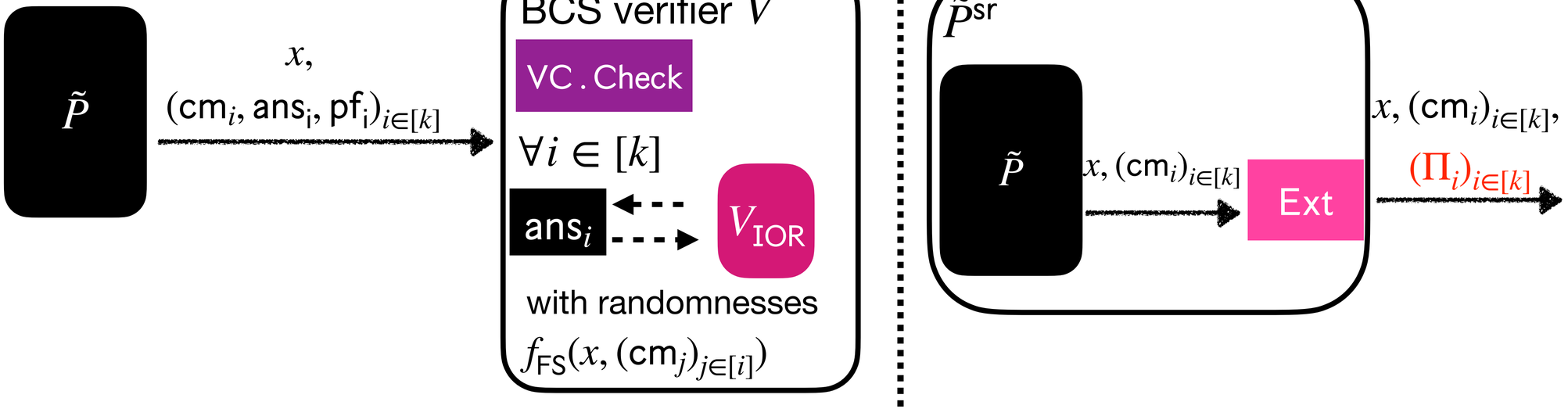
$\tilde{P}^{\mathsf{sr}}$ needs to output the chosen IOR strings

Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 3: how to derive the output

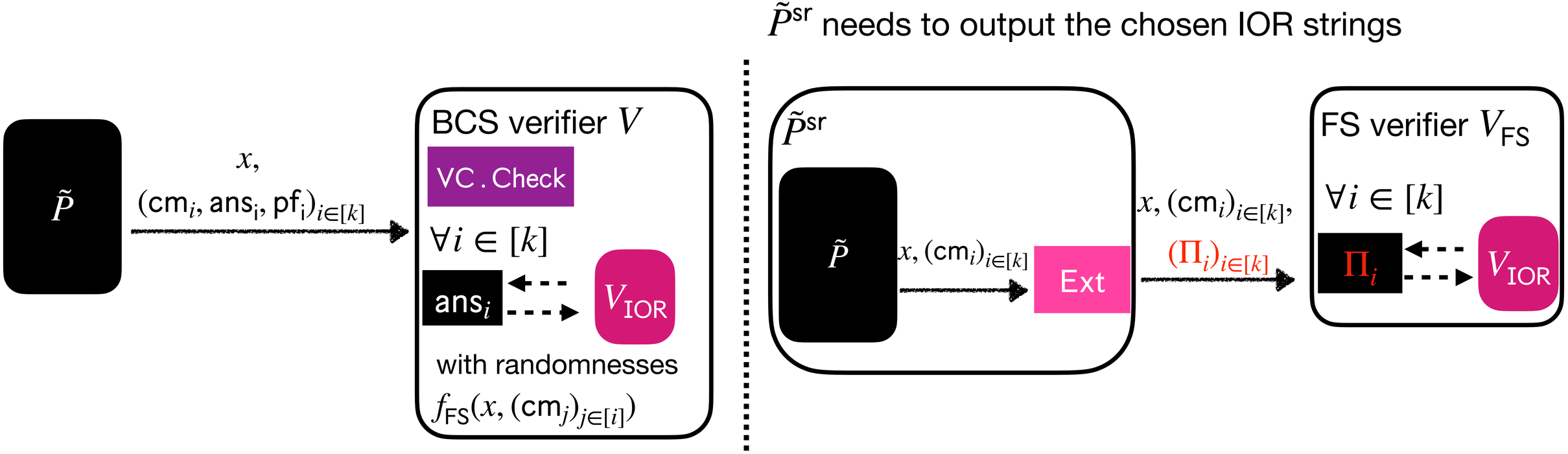
$\tilde{P}^{\mathsf{sr}}$ needs to output the chosen IOR strings



Construction of $\tilde{P}^{\mathsf{sr}}$

Classical case

Step 3: how to derive the output



Classical case

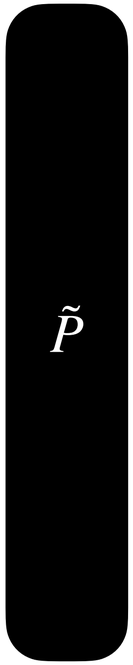
The construction in summary: $\tilde{P}^{\mathrm{sr}}$ simulates $\tilde{P}$.

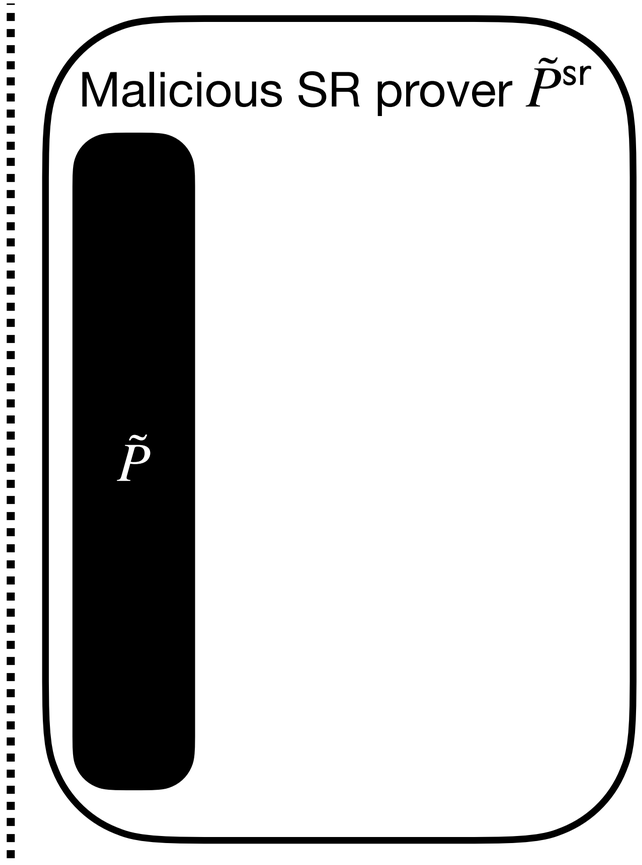
Classical case

The construction in summary: $\tilde{P}^{\mathsf{sr}}$ simulates $\tilde{P}$.

Classical case

Malicious BCS prover



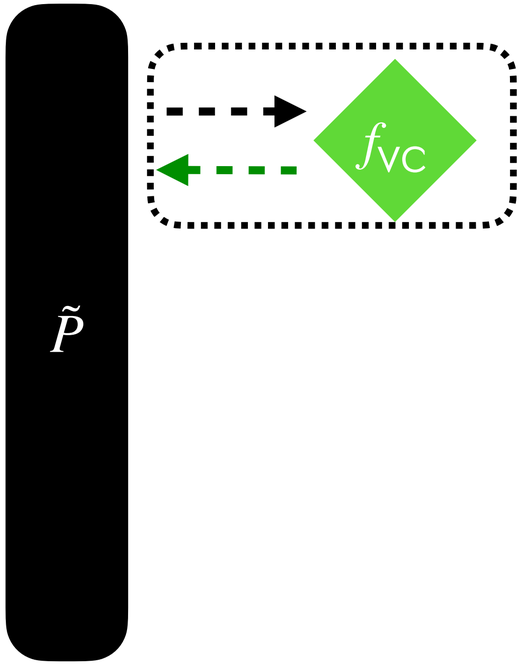


The construction in summary: $\tilde{P}^{\mathsf{sr}}$ simulates $\tilde{P}$.

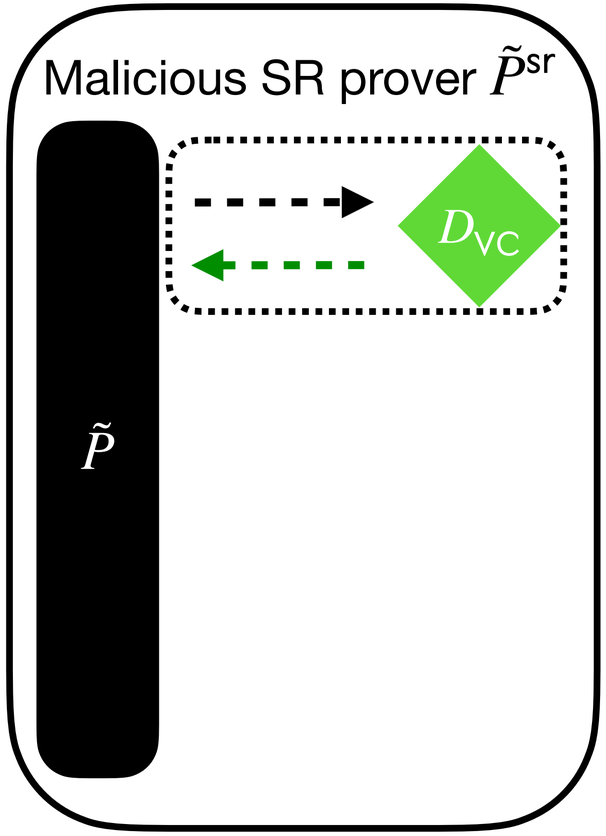
Classical case

How to answer f_{VC} queries?

Malicious BCS prover



Malicious SR prover $\tilde{P}^{\mathsf{sr}}$

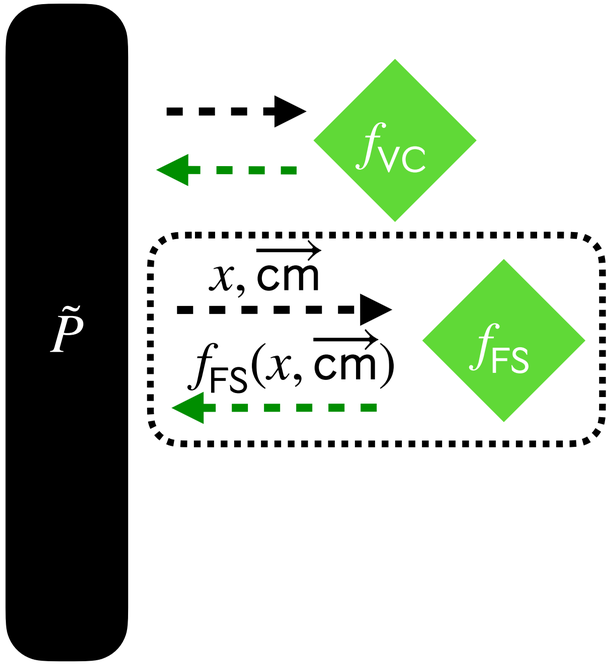


The construction in summary: $\tilde{P}^{\mathsf{sr}}$ simulates $\tilde{P}$.

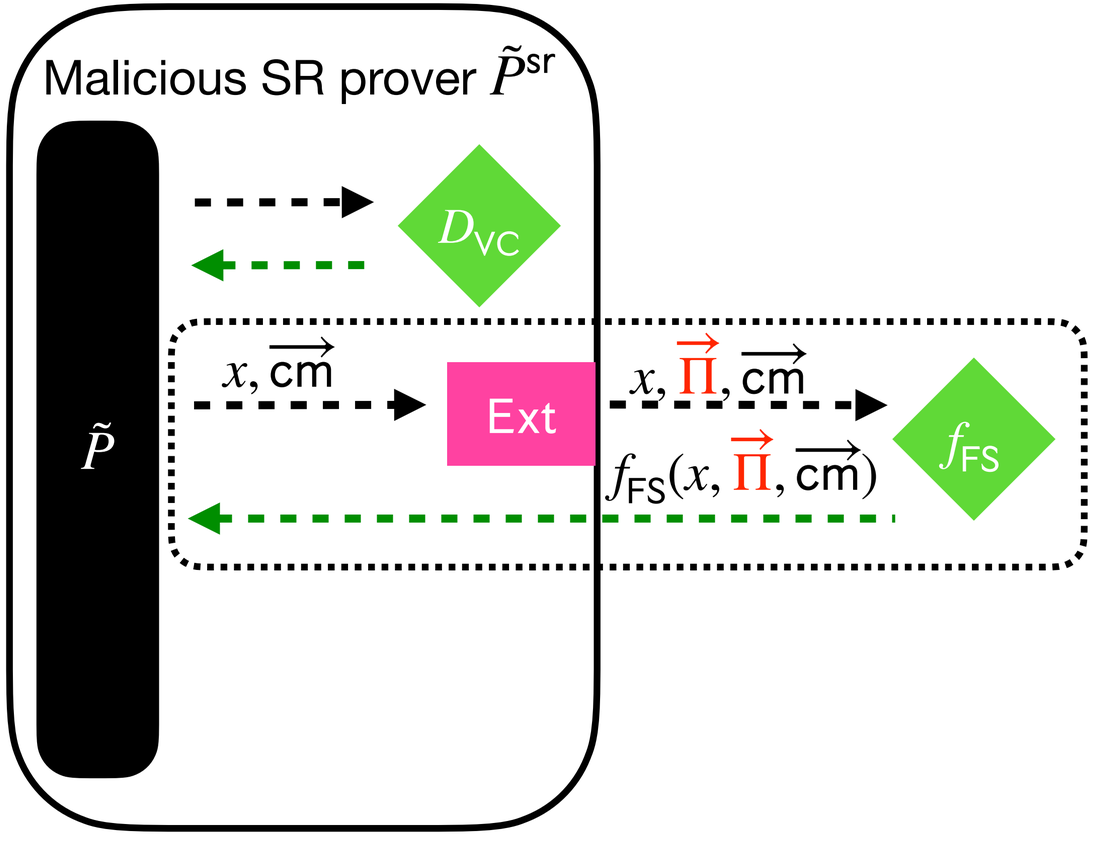
Classical case

How to answer f_{FS} queries?

Malicious BCS prover



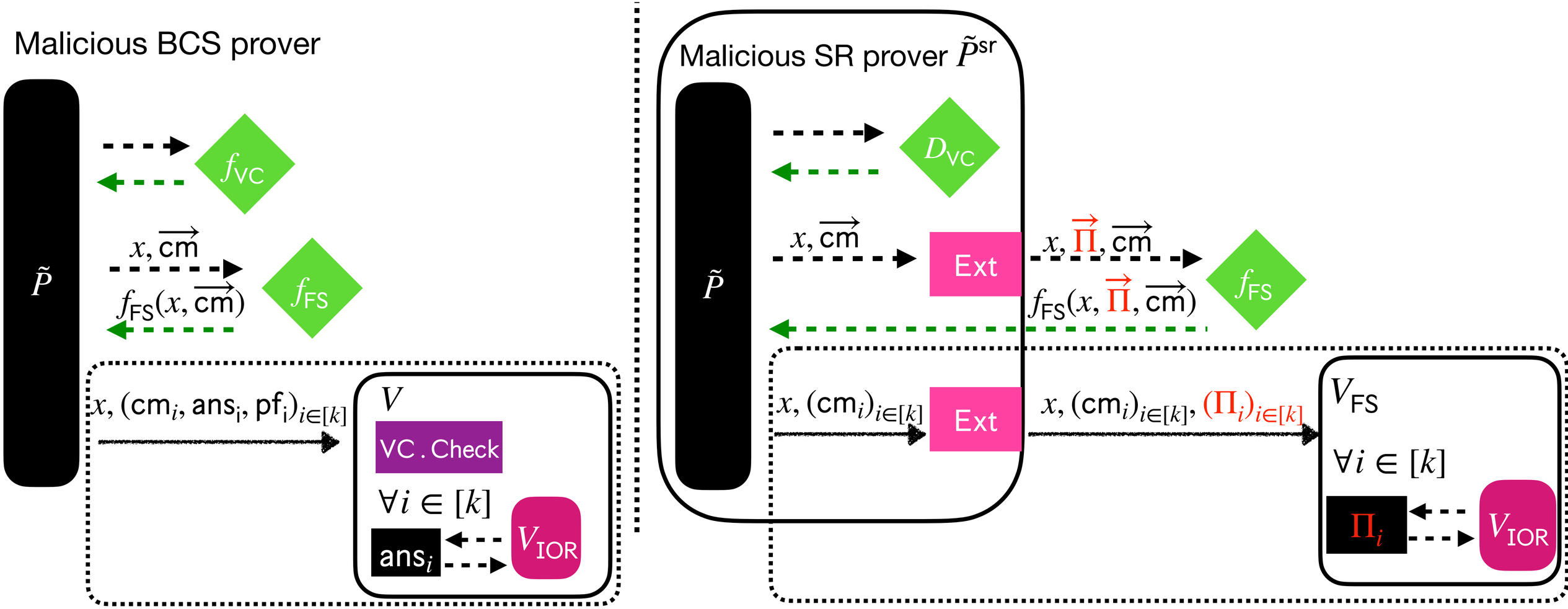
Malicious SR prover $\tilde{P}^{\mathsf{sr}}$



The construction in summary: $\tilde{P}^{\mathsf{sr}}$ simulates $\tilde{P}$.

Classical case

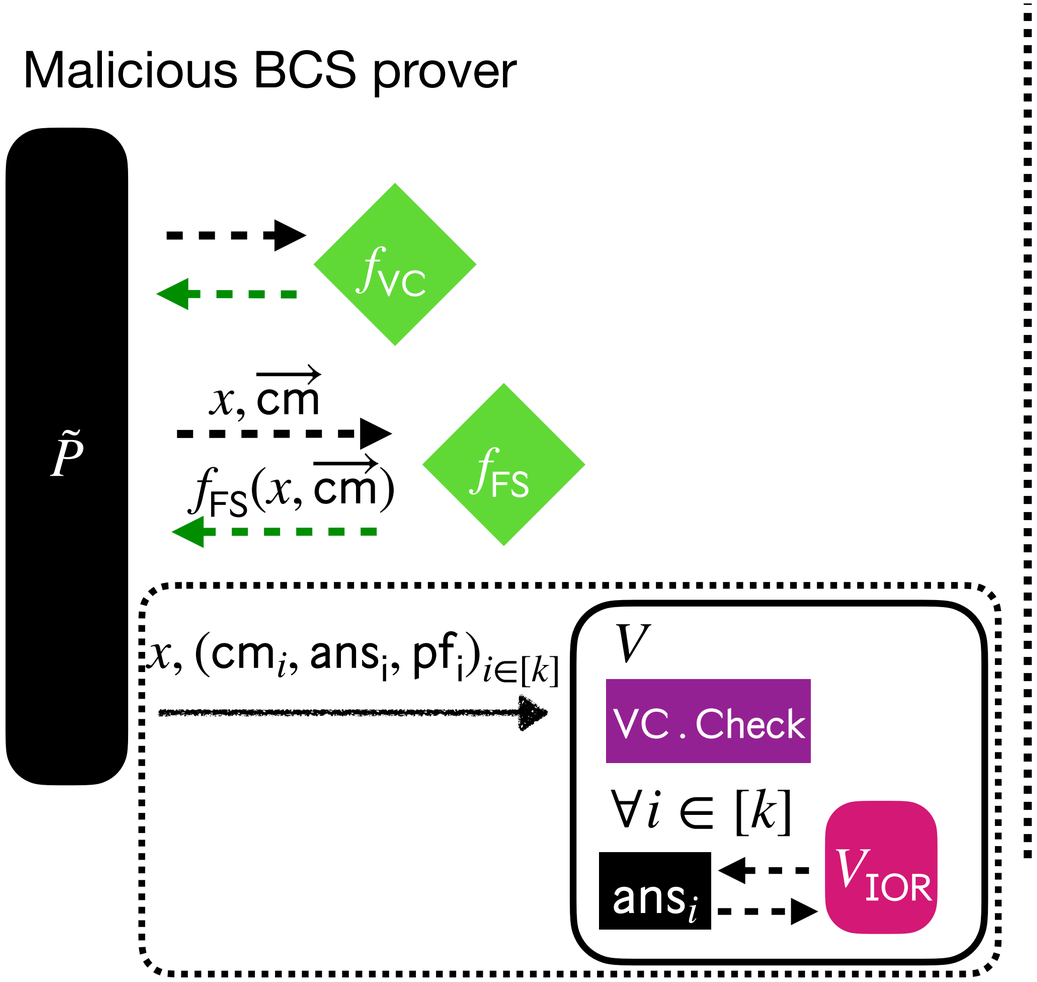
How to derive the output of $\tilde{P}^{\mathsf{sr}}$ from the output of $\tilde{P}$?

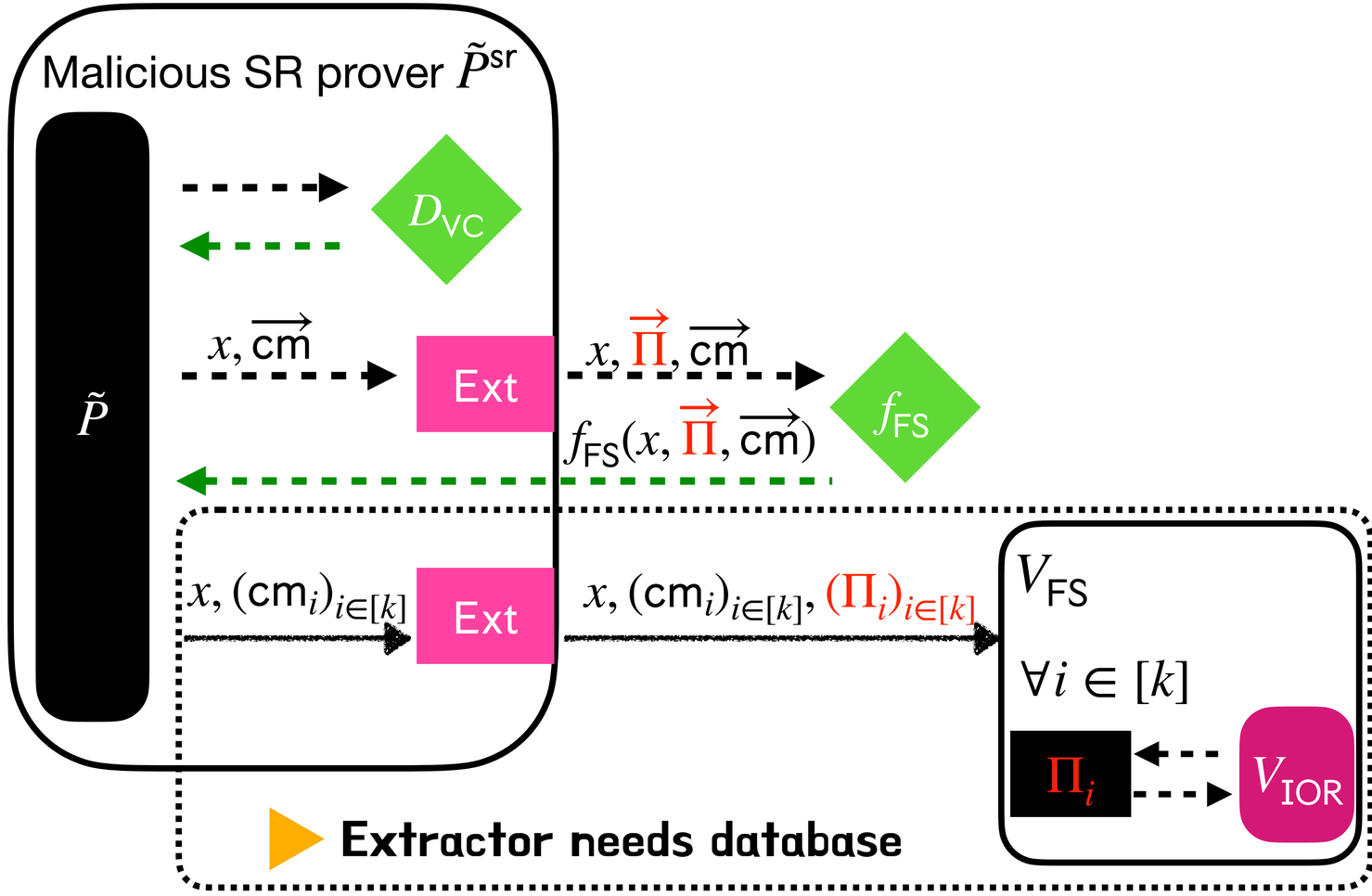


The construction in summary: $\tilde{P}^{\mathsf{sr}}$ simulates $\tilde{P}$.

Classical case

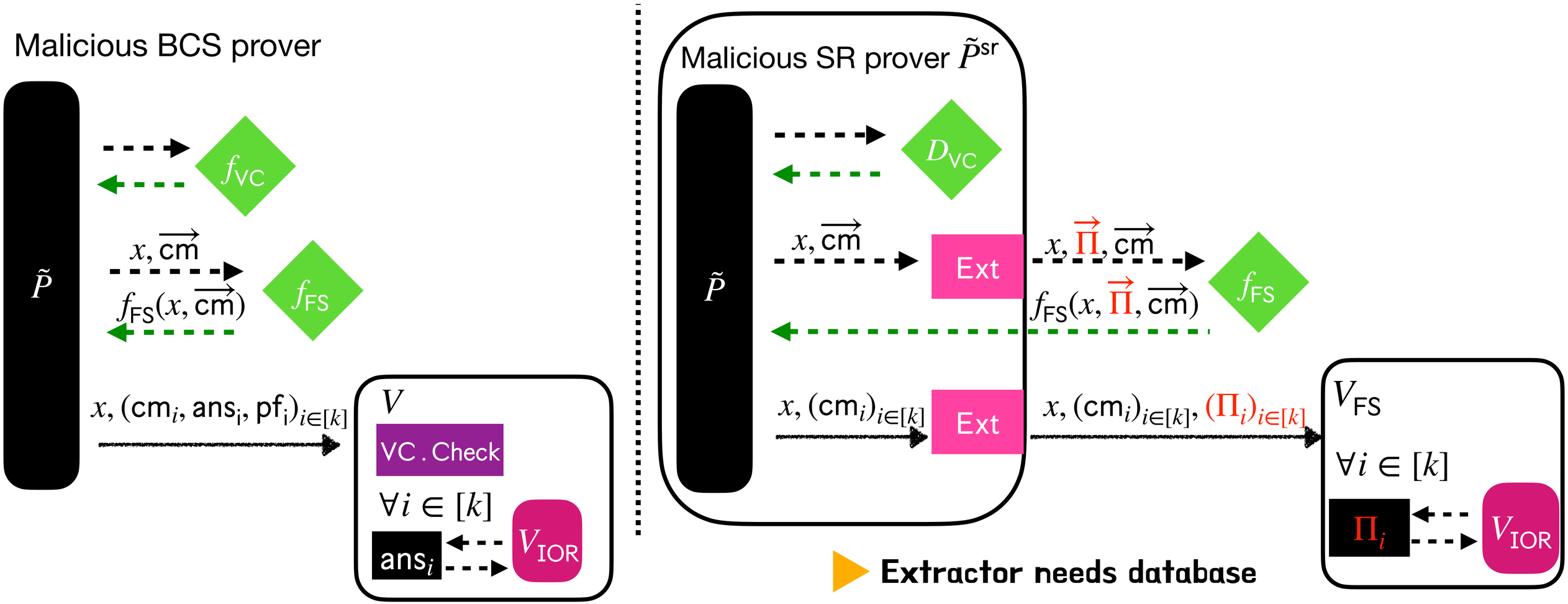
How to derive the output of $\tilde{P}^{\mathsf{sr}}$ from the output of $\tilde{P}$?





The construction in summary: $\tilde{P}^{\mathsf{sr}}$ simulates $\tilde{P}$.

Classical case



Goal: we want to show $\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$

Classical case

Goal: we want to show

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

Classical case

Goal: we want to show

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

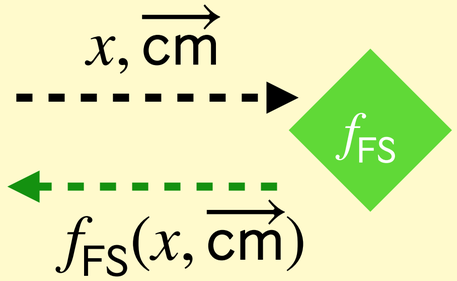
Difference 1

Classical case

Goal: we want to show

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

Difference 1

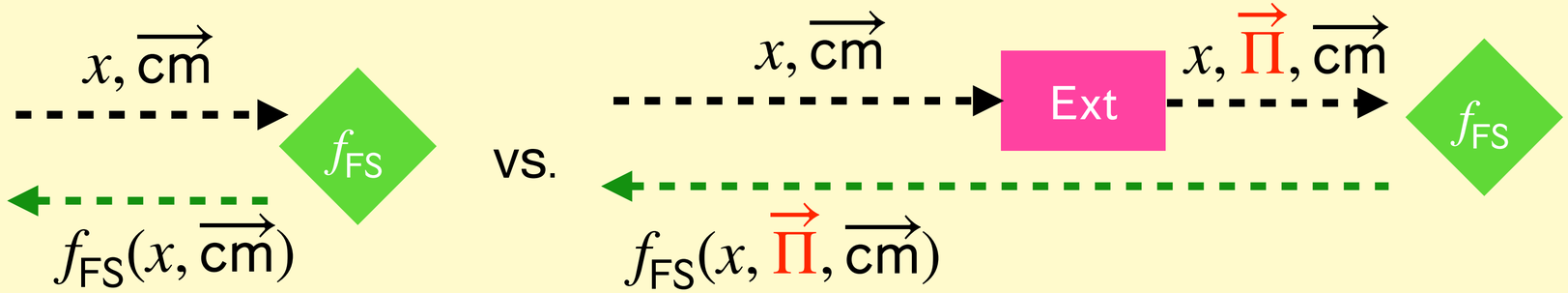


Classical case

Goal: we want to show

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

Difference 1

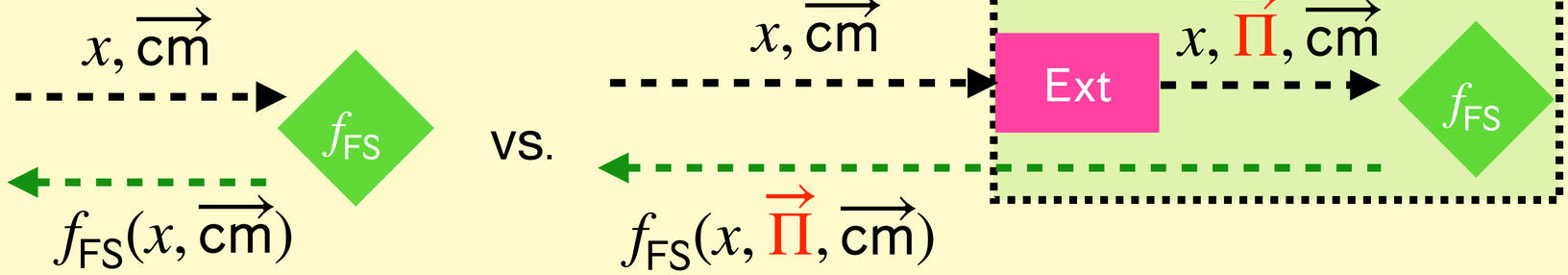


Classical case

Goal: we want to show

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

Difference 1

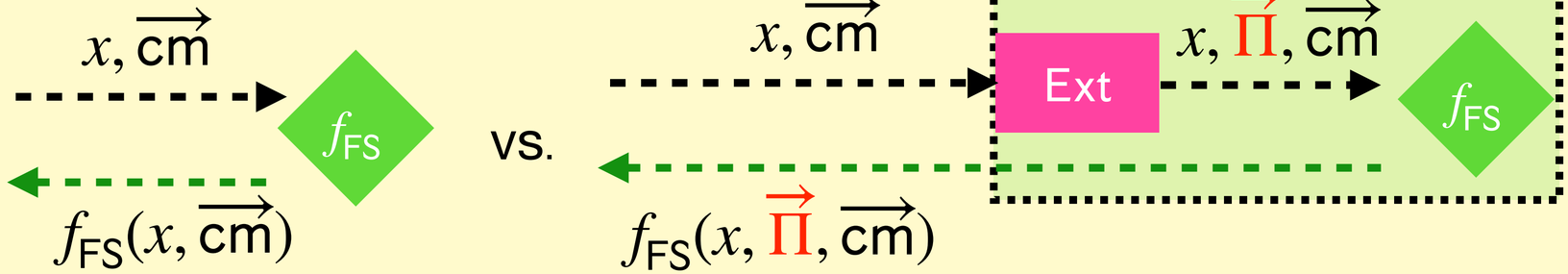


Classical case

Goal: we want to show

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

Difference 1

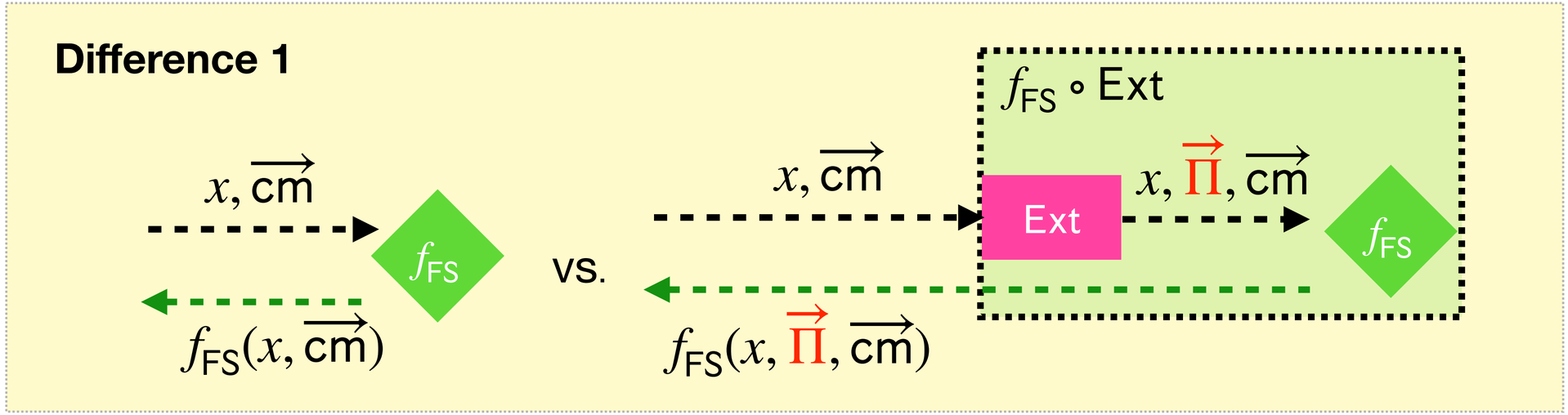


VC Property 1: Online consistency

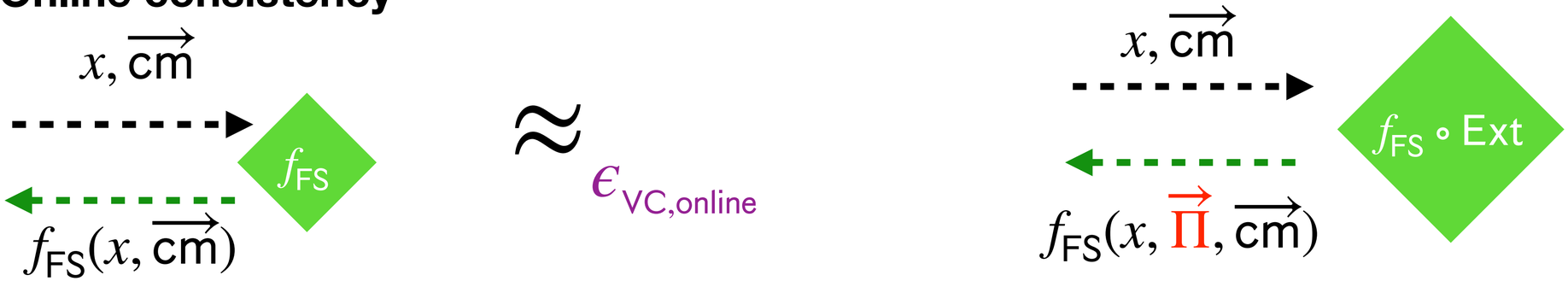
Classical case

Goal: we want to show

$$\Pr[\tilde{P}^{\text{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\text{VC}}$$



VC Property 1: Online consistency



Classical case

Goal: we want to show

$$\Pr[\tilde{P}^{\text{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\text{VC}}$$

Classical case

Goal: we want to show

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

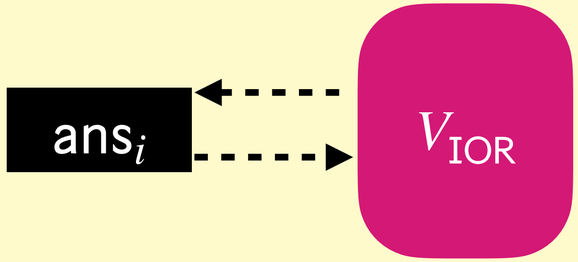
Difference 2

Classical case

Goal: we want to show

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

Difference 2

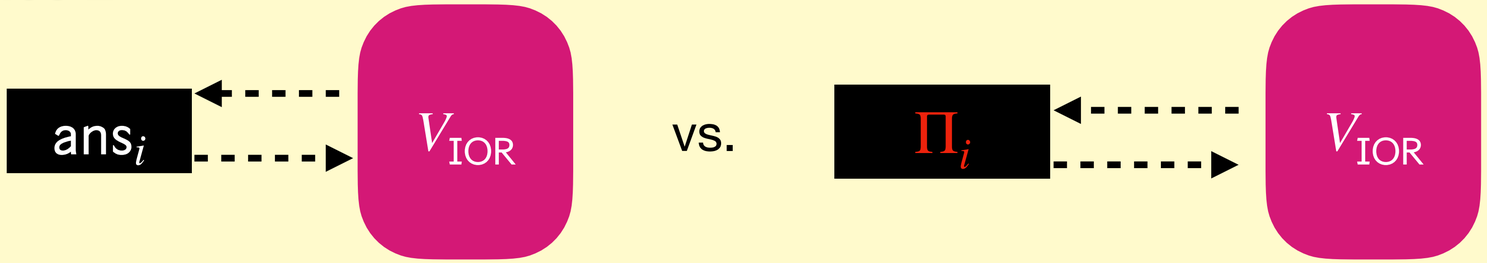


Classical case

Goal: we want to show

$$\Pr[\tilde{P}^{\mathsf{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\mathsf{VC}}$$

Difference 2

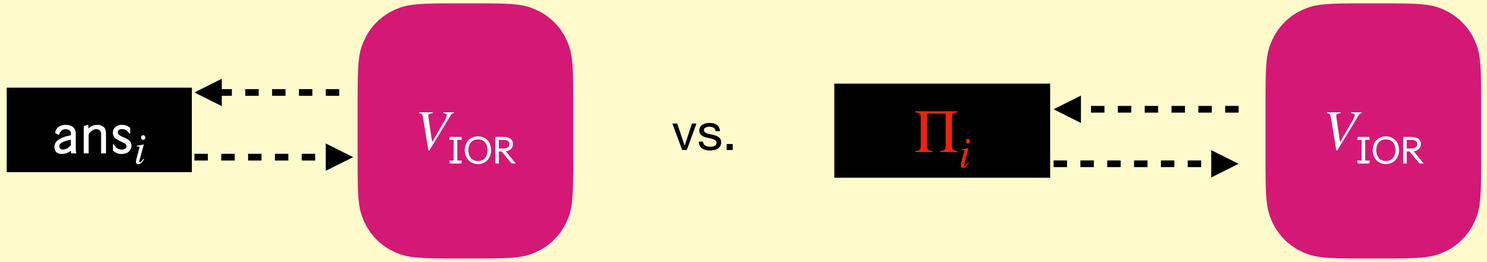


Classical case

Goal: we want to show

$$\Pr[\tilde{P}^{\text{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\text{VC}}$$

Difference 2



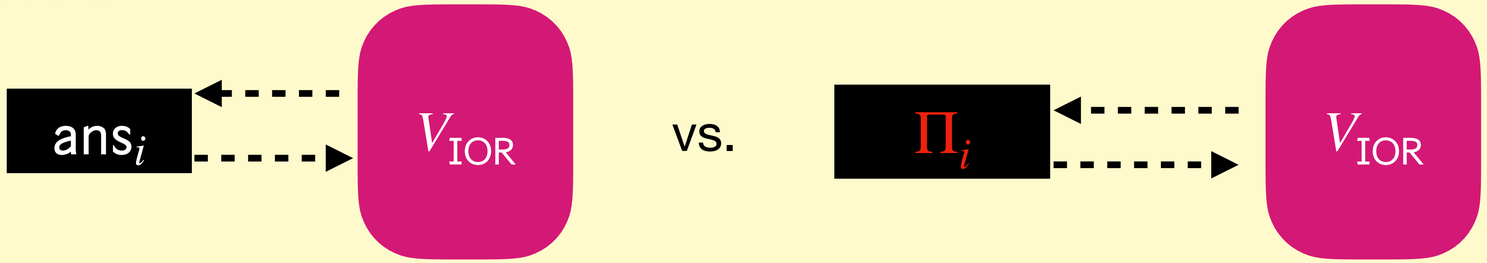
VC Property 2: Offline extractability

Classical case

Goal: we want to show

$$\Pr[\tilde{P}^{\text{sr}} \text{ wins SR game}] \geq \Pr[\tilde{P} \text{ fools } V] - \epsilon_{\text{VC}}$$

Difference 2



VC Property 2: Offline extractability

if VC . Check $= 1$

What happens in the quantum case?

Quantum case

Goal: we want to construct a PQSR prover $\tilde{P}^{\star,\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon^{\star}_{\mathsf{VC}}$$

Quantum case

Goal: we want to construct a PQSR prover $\tilde{P}^{\star,\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon^{\star}_{\mathsf{VC}}$$

Our construction: $\tilde{P}^{\star,\mathsf{sr}}$ **simulates** $\tilde{P}^{\star}$.

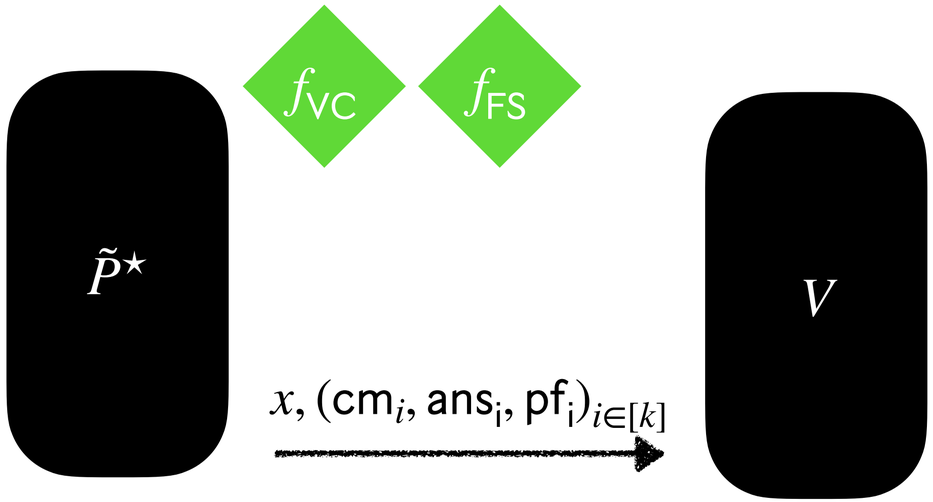
Quantum case

Goal: we want to construct a PQSR prover $\tilde{P}^{\star,\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon^{\star}_{\mathsf{VC}}$$

Our construction: $\tilde{P}^{\star,\mathsf{sr}}$ simulates $\tilde{P}^{\star}$.

Malicious BCS prover

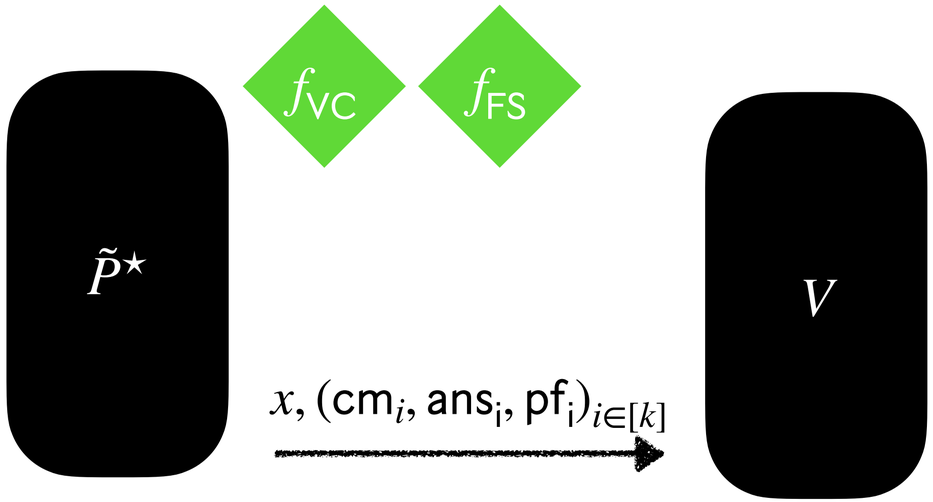


Goal: we want to construct a PQSR prover $\tilde{P}^{\star,\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon_{\mathsf{VC}}^{\star}$$

Our construction: $\tilde{P}^{\star,\mathsf{sr}}$ simulates $\tilde{P}^{\star}$.

Malicious BCS prover



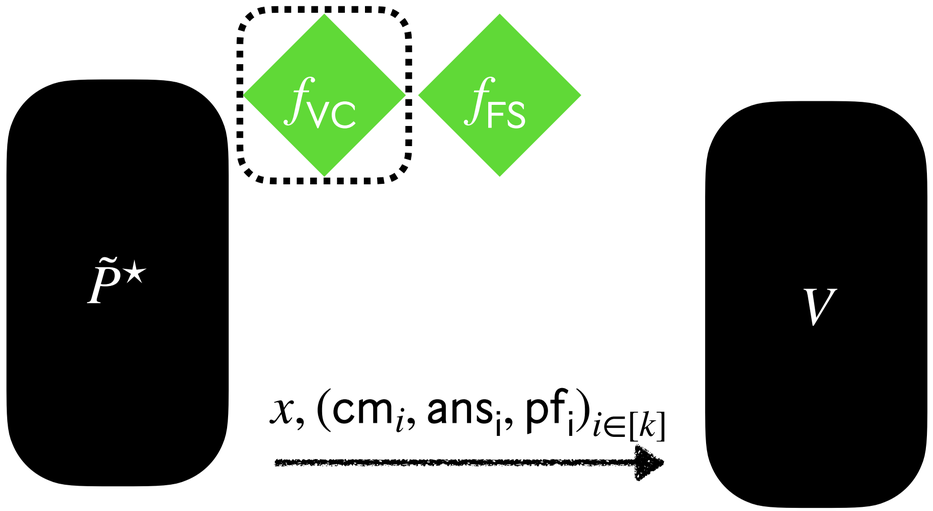
🤔 **How to…**

Goal: we want to construct a PQSR prover $\tilde{P}^{\star,\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon_{\mathsf{VC}}^{\star}$$

Our construction: $\tilde{P}^{\star,\mathsf{sr}}$ simulates $\tilde{P}^{\star}$.

Malicious BCS prover



🤔 **How to…**

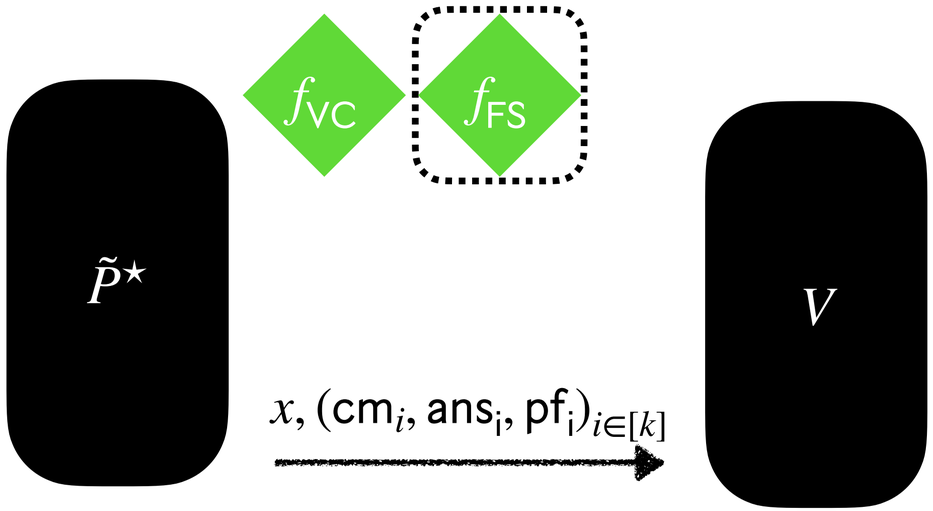
1. Answer quantum f_{VC} queries?

Goal: we want to construct a PQSR prover $\tilde{P}^{\star,\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon^{\star}_{\mathsf{VC}}$$

Our construction: $\tilde{P}^{\star,\mathsf{sr}}$ **simulates** $\tilde{P}^{\star}$.

Malicious BCS prover



🤔 **How to…**

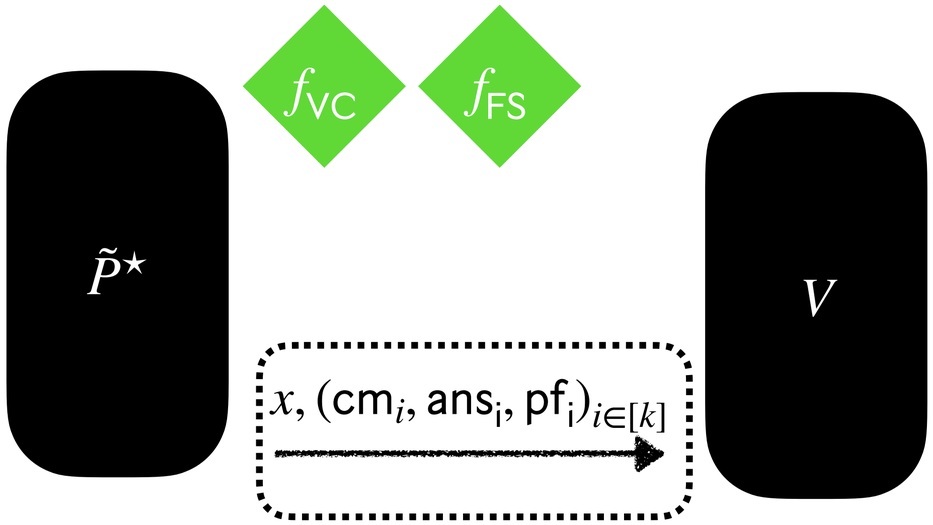
1. Answer quantum f_{VC} queries?
2. Answer quantum f_{FS} queries?

Goal: we want to construct a PQSR prover $\tilde{P}^{\star,\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon^{\star}_{\mathsf{VC}}$$

Our construction: $\tilde{P}^{\star,\mathsf{sr}}$ **simulates** $\tilde{P}^{\star}$.

Malicious BCS prover



🤔 **How to…**

1. Answer quantum f_{VC} queries?
2. Answer quantum f_{FS} queries?
3. Derive the output of $\tilde{P}^{\star,\mathsf{sr}}$ from the output of $\tilde{P}^{\star}$?

Quantum case

Goal: we want to construct a PQSR prover $\tilde{P}^{\star,\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon^{\star}_{\mathsf{VC}}$$

Quantum case

Goal: we want to construct a PQSR prover $\tilde{P}^{\star,\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon^{\star}_{\mathsf{VC}}$$

The VC extractor needs some trapdoor information about adversary's queries.

Goal: we want to construct a PQSR prover $\tilde{P}^{\star,\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon^{\star}_{\mathsf{VC}}$$

The VC extractor needs some trapdoor information about adversary's queries.

Starting point: Use compressed oracle!

Goal: we want to construct a PQSR prover $\tilde{P}^{\star,\mathsf{sr}}$ such that

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon^{\star}_{\mathsf{VC}}$$

The VC extractor needs some trapdoor information about adversary's queries.

Starting point: Use compressed oracle!

It gives you "Quantum Database" $\mathscr{D}_{VC}$,

but additional care is required to simulate $\tilde{P}^{\star}$ without much disturbance.

Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

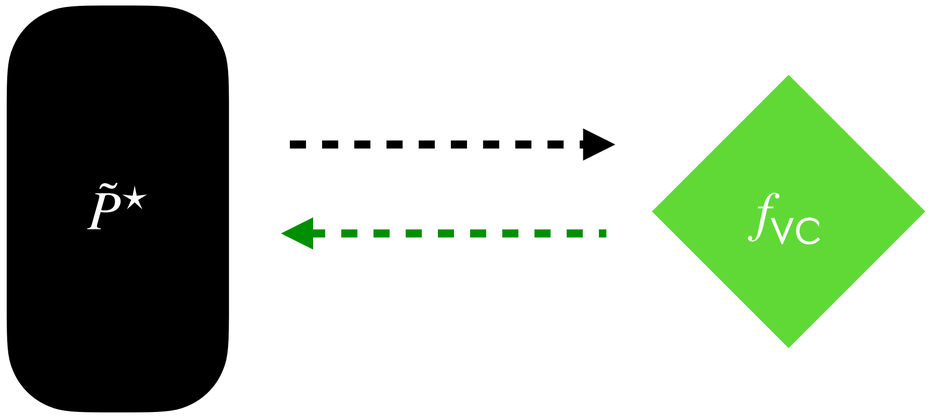
Step 1: how to answer quantum f_{VC} queries?

Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

Step 1: how to answer quantum f_{VC} queries?

Malicious BCS prover

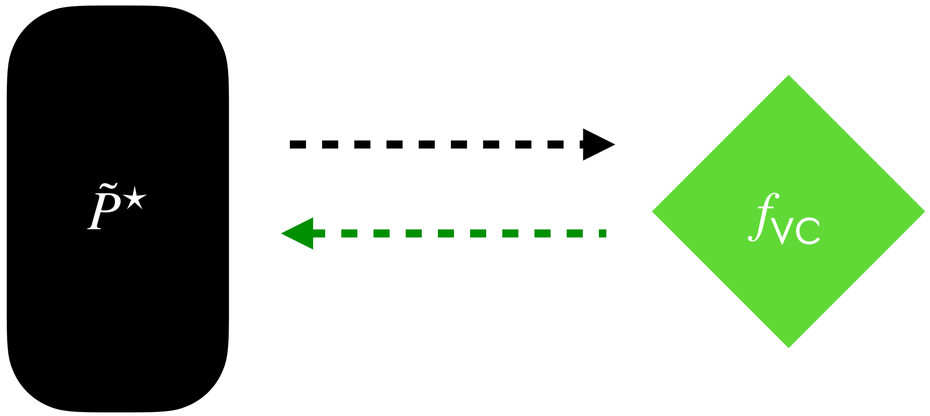


Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

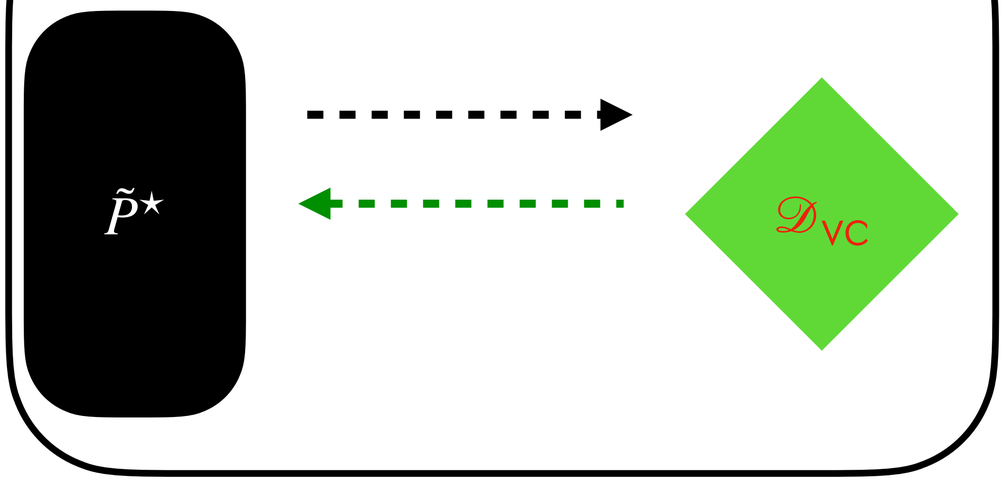
Quantum case

Step 1: how to answer quantum f_{VC} queries?

Malicious BCS prover



Malicious SR prover $\tilde{P}^{\star,\mathsf{sr}}$



Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

Step 2: how to answer quantum f_{FS} queries?

Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

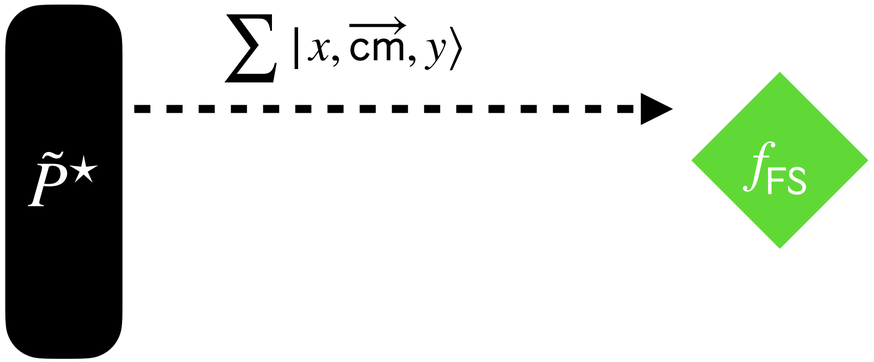
Quantum case

Step 2: how to answer quantum f_{FS} queries?

Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

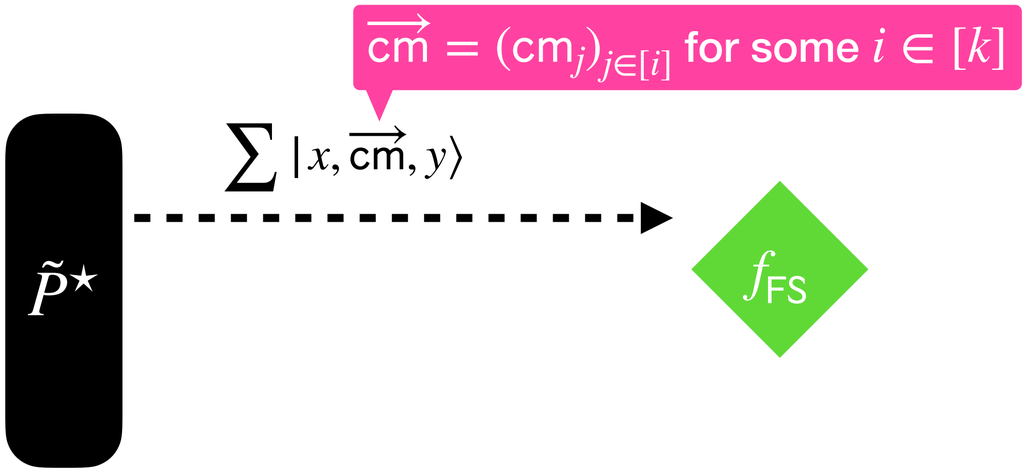
Step 2: how to answer quantum f_{FS} queries?



Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

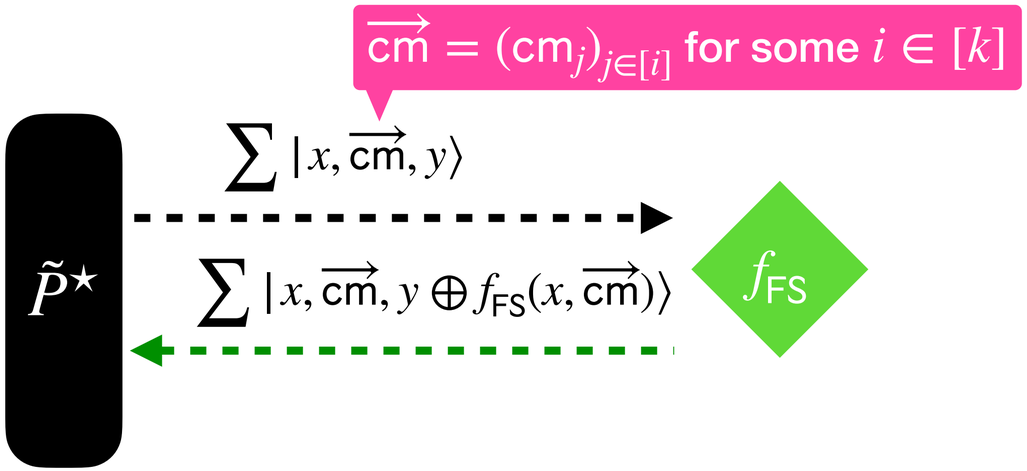
Step 2: how to answer quantum f_{FS} queries?



Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

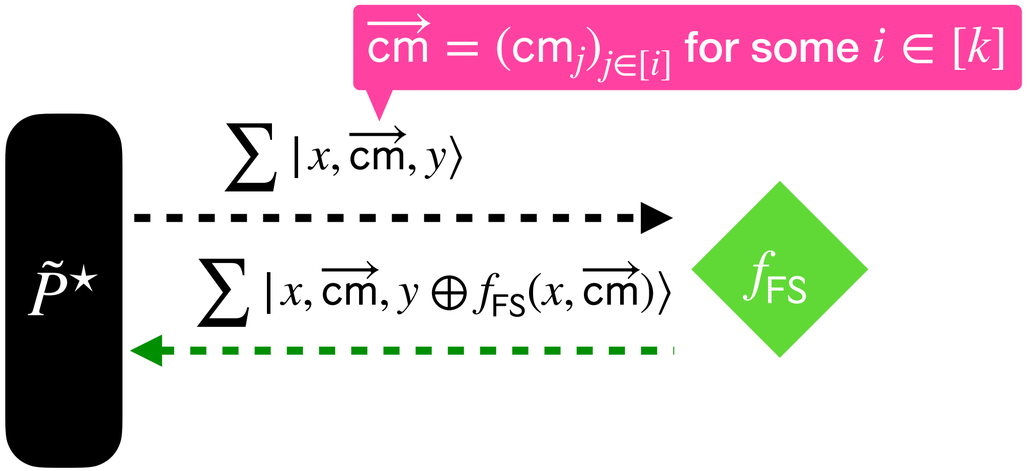
Step 2: how to answer quantum f_{FS} queries?



Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

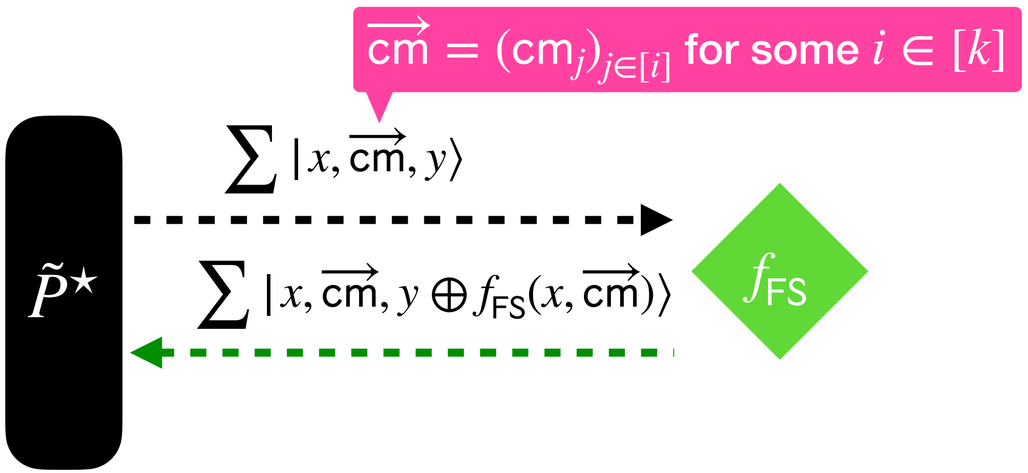
Step 2: how to answer quantum f_{FS} queries?



Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

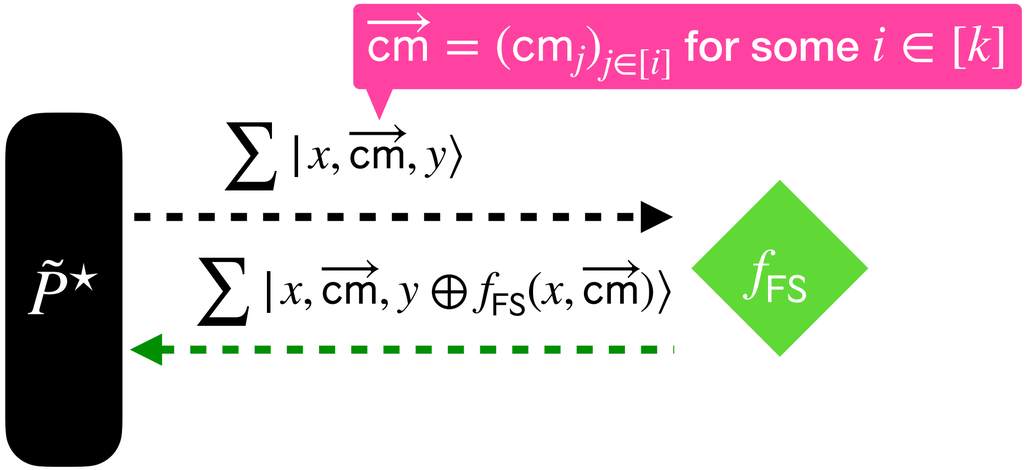
Step 2: how to answer quantum f_{FS} queries?



Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

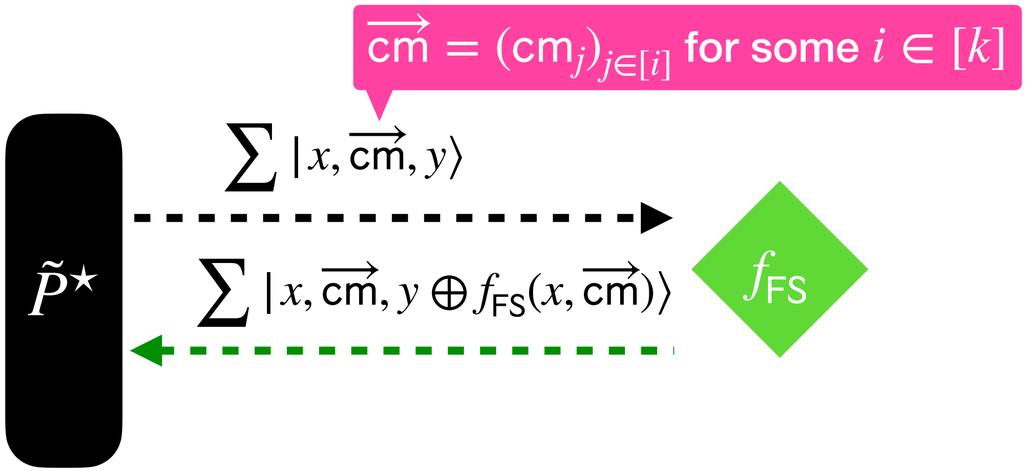
Step 2: how to answer quantum f_{FS} queries?

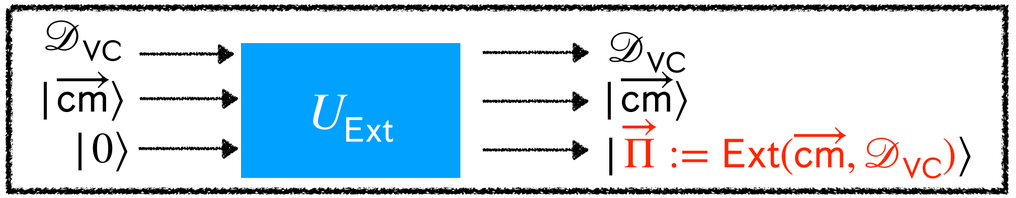


Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

Step 2: how to answer quantum f_{FS} queries?

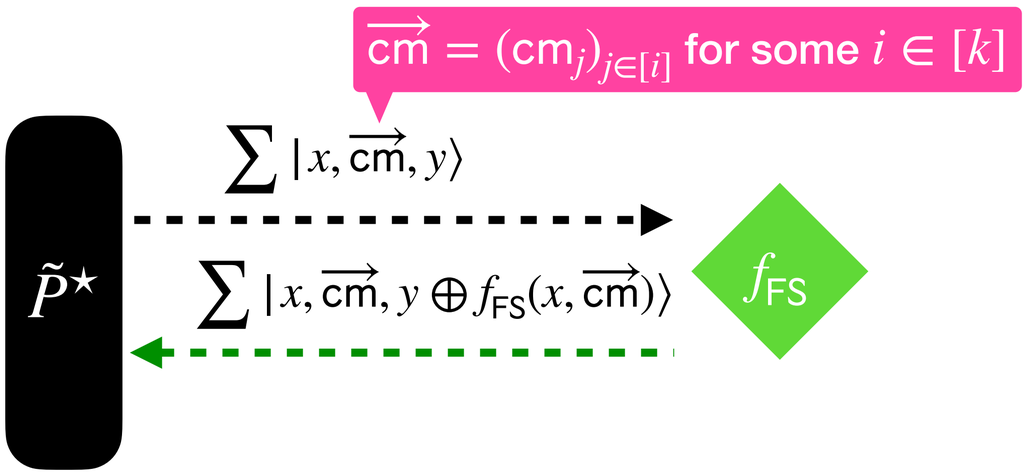


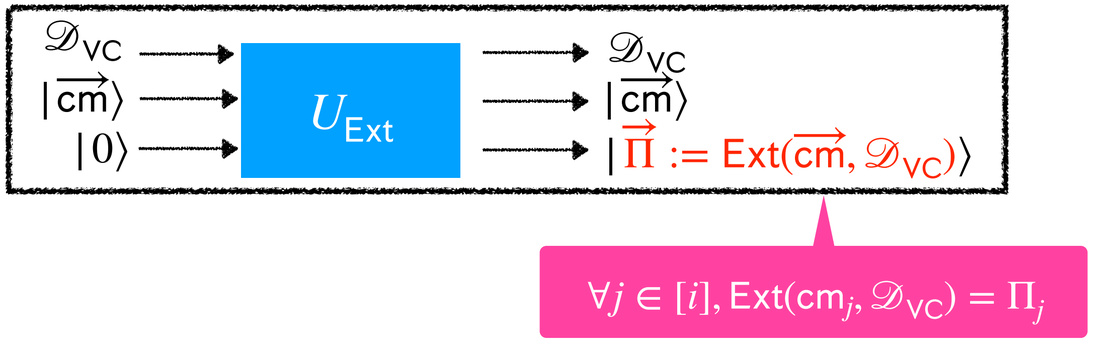


Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

Step 2: how to answer quantum f_{FS} queries?

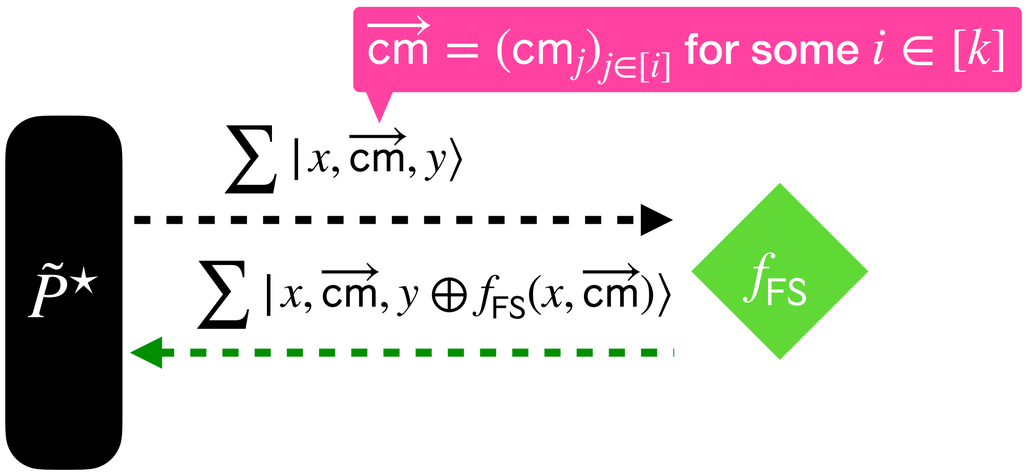




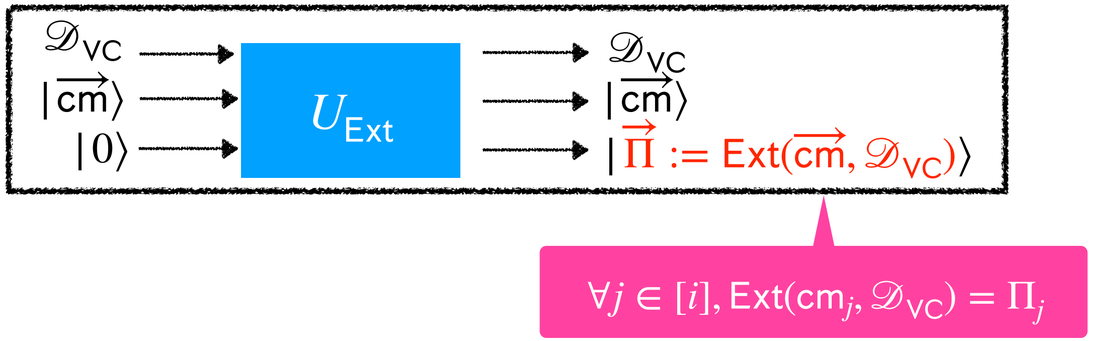
Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

Step 2: how to answer quantum f_{FS} queries?



Extractor needs database

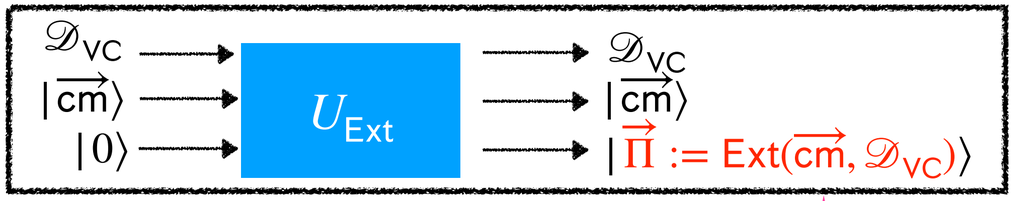


Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

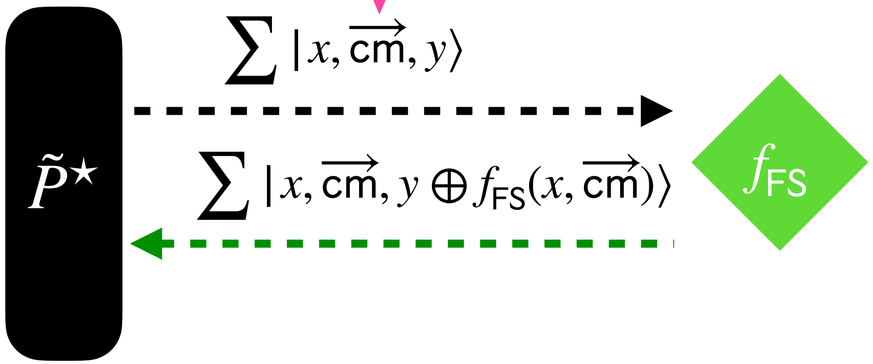
Step 2: how to answer quantum f_{FS} queries?

Extractor needs database



$\forall j \in [i], \mathsf{Ext}(\mathsf{cm}_j, \mathscr{D}_{\mathsf{VC}}) = \Pi_j$

$\overrightarrow{\mathsf{cm}} = (\mathsf{cm}_j)_{j\in[i]}$ for some $i \in [k]$

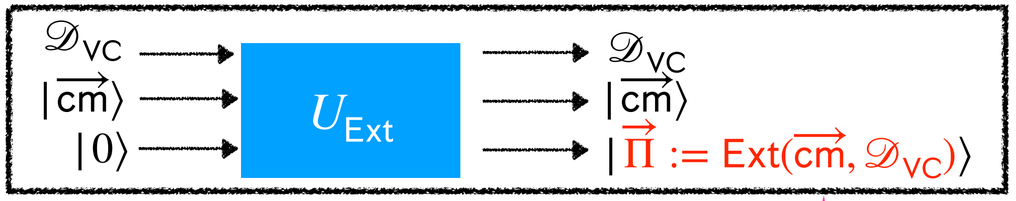


Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

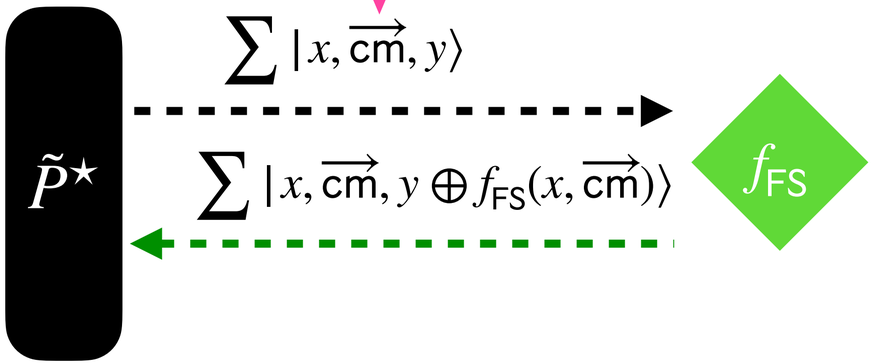
Step 2: how to answer quantum f_{FS} queries?

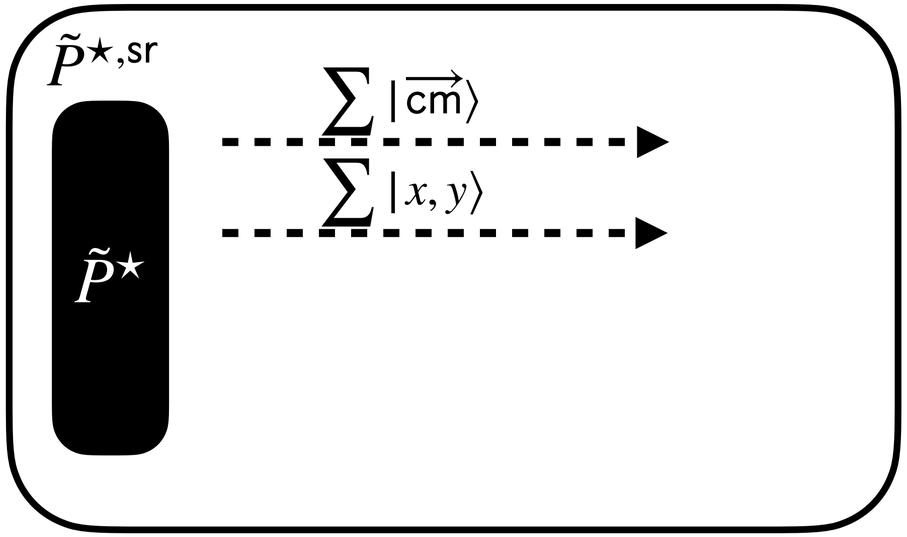
Extractor needs database



$\forall j \in [i], \mathsf{Ext}(\mathsf{cm}_j, \mathscr{D}_{\mathsf{VC}}) = \Pi_j$

$\overrightarrow{\mathsf{cm}} = (\mathsf{cm}_j)_{j\in[i]}$ for some $i \in [k]$

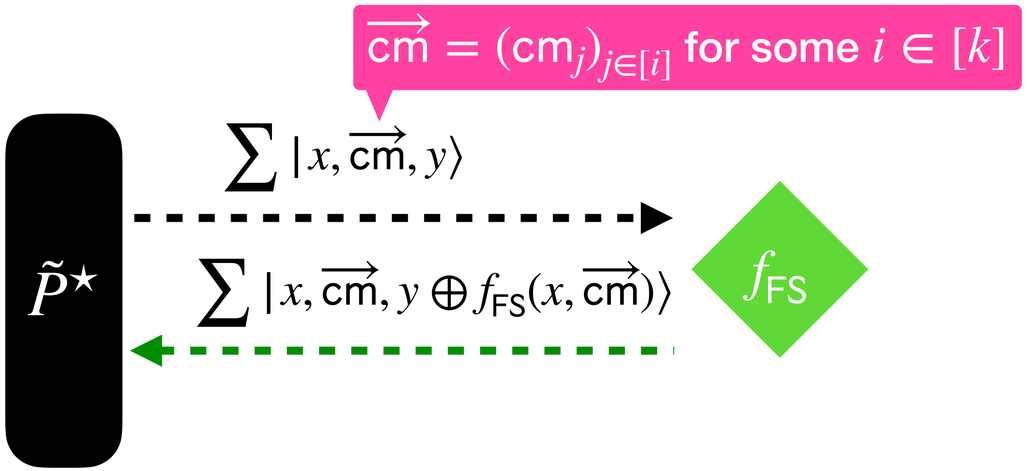




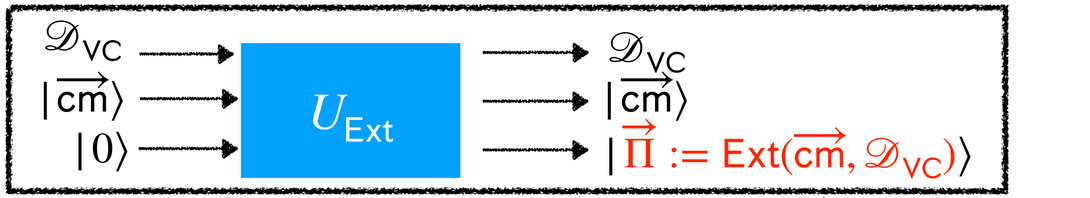
Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

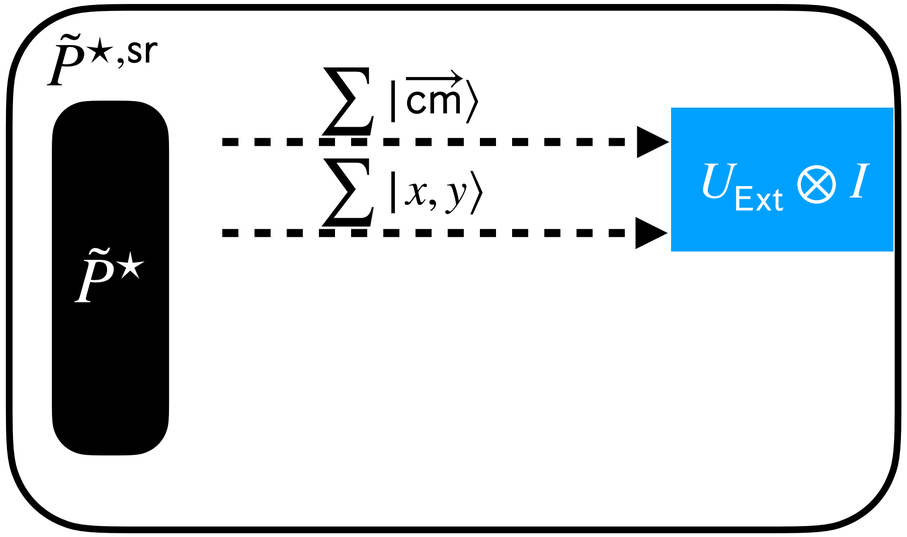
Step 2: how to answer quantum f_{FS} queries?



Extractor needs database



$\forall j \in [i], \mathsf{Ext}(\mathsf{cm}_j, \mathscr{D}_{\mathsf{VC}}) = \Pi_j$

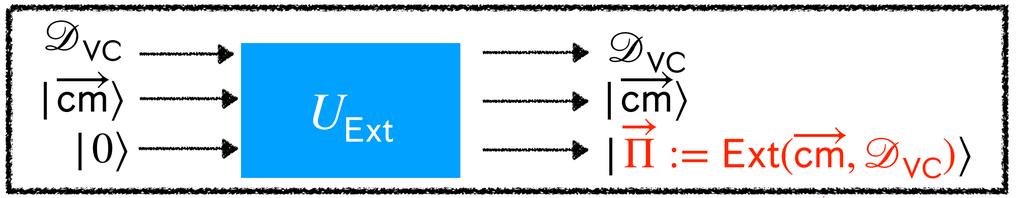


Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

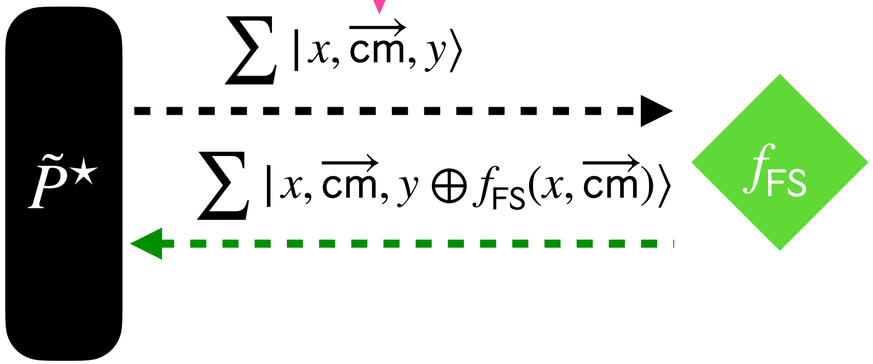
Step 2: how to answer quantum f_{FS} queries?

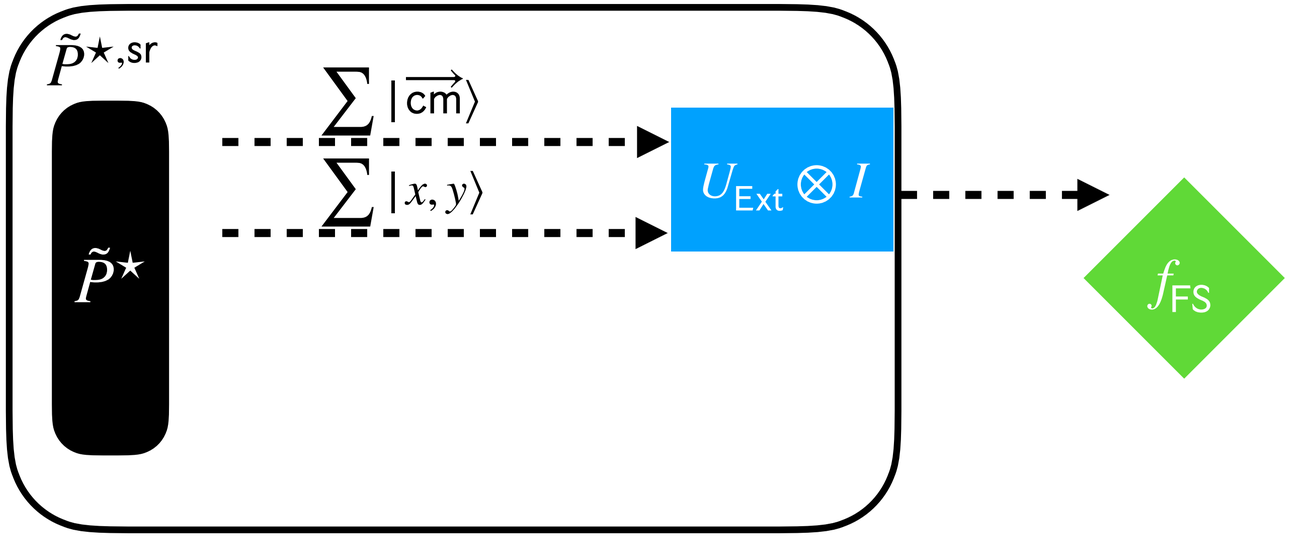
Extractor needs database



$\forall j \in [i], \mathsf{Ext}(\mathsf{cm}_j, \mathscr{D}_{\mathsf{VC}}) = \Pi_j$

$\overrightarrow{\mathsf{cm}} = (\mathsf{cm}_j)_{j\in[i]}$ for some $i \in [k]$



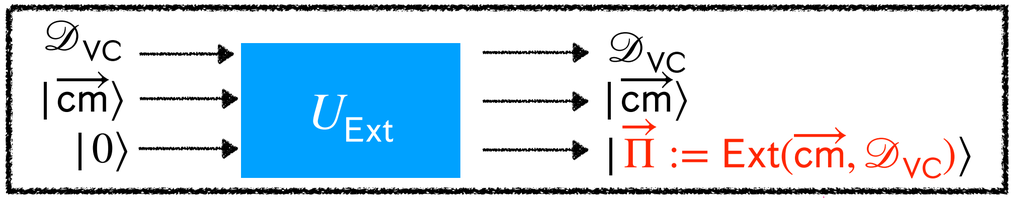


Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

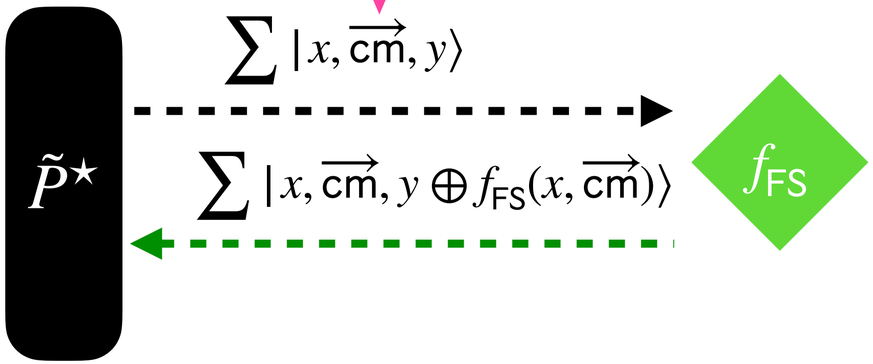
Step 2: how to answer quantum f_{FS} queries?

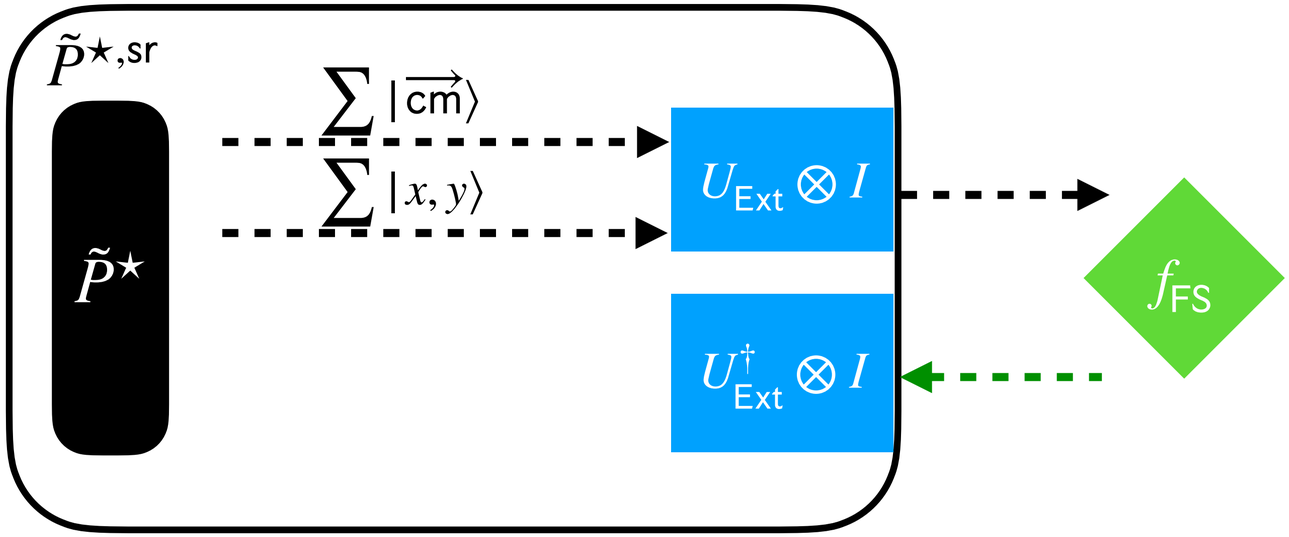
Extractor needs database



$\forall j \in [i], \mathsf{Ext}(\mathsf{cm}_j, \mathscr{D}_{\mathsf{VC}}) = \Pi_j$

$\overrightarrow{\mathsf{cm}} = (\mathsf{cm}_j)_{j\in[i]}$ for some $i \in [k]$



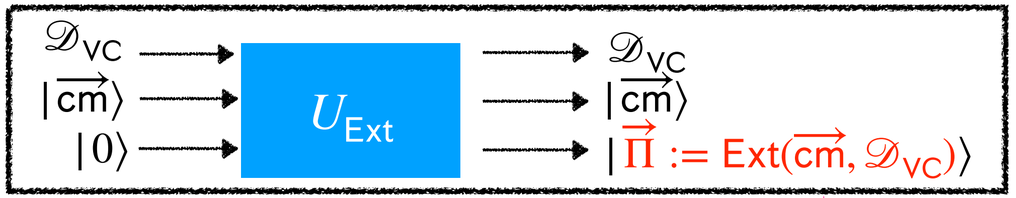


Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Quantum case

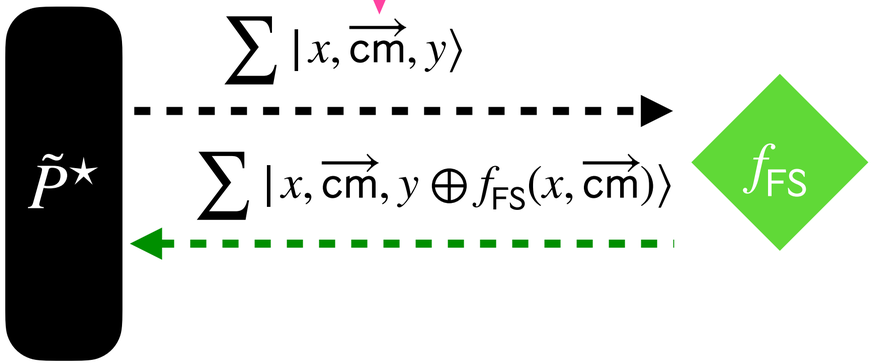
Step 2: how to answer quantum f_{FS} queries?

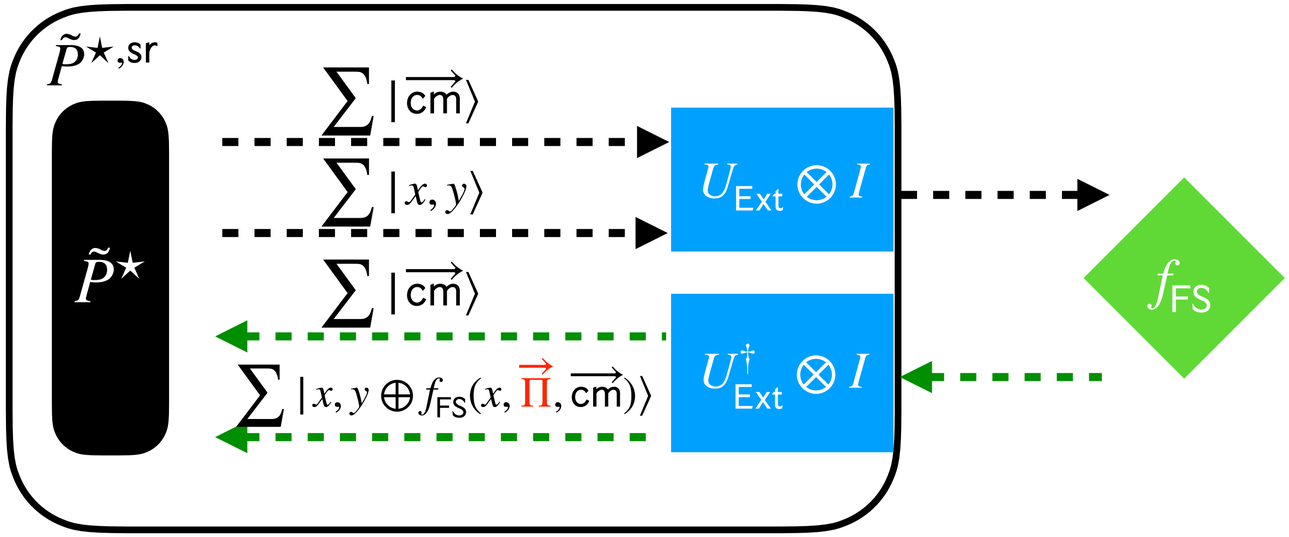
Extractor needs database



$\forall j \in [i], \mathsf{Ext}(\mathsf{cm}_j, \mathscr{D}_{\mathsf{VC}}) = \Pi_j$

$\overrightarrow{\mathsf{cm}} = (\mathsf{cm}_j)_{j\in[i]}$ for some $i \in [k]$





Our construction of $\tilde{P}^{\star,\mathrm{sr}}$

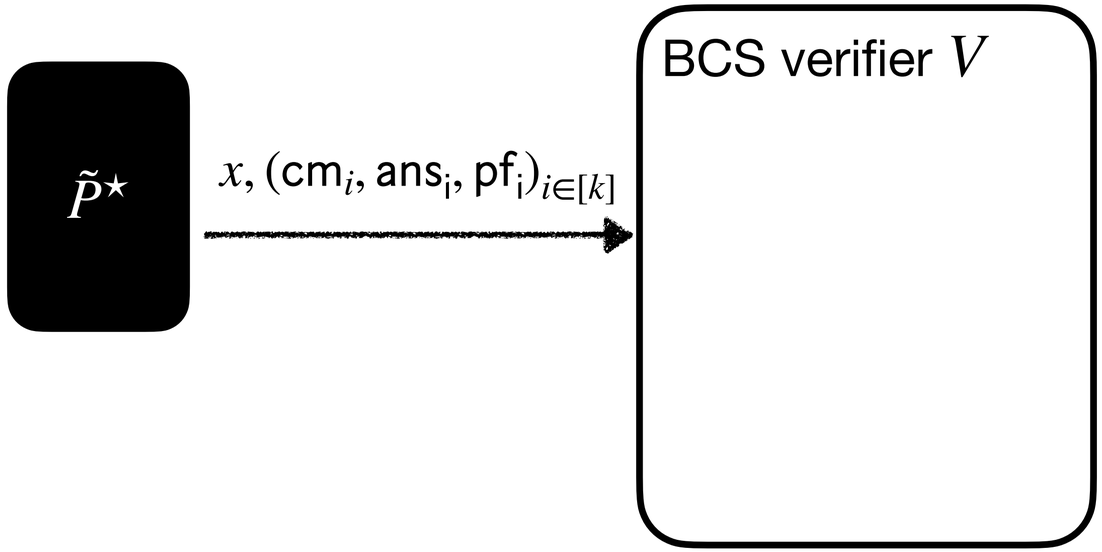
Step 3: how to derive the output

Quantum case

Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Step 3: how to derive the output

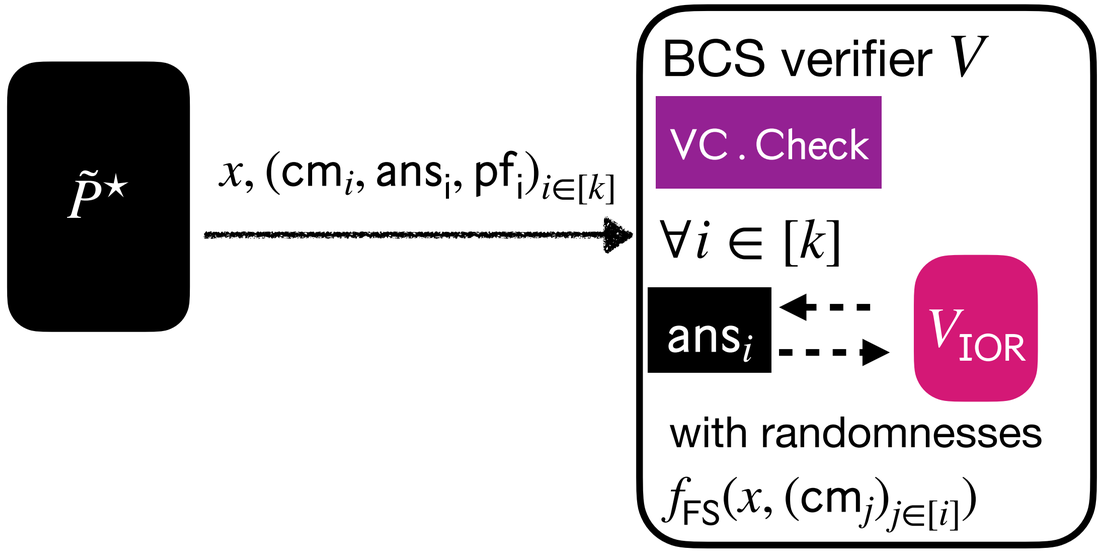
Quantum case



Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Step 3: how to derive the output

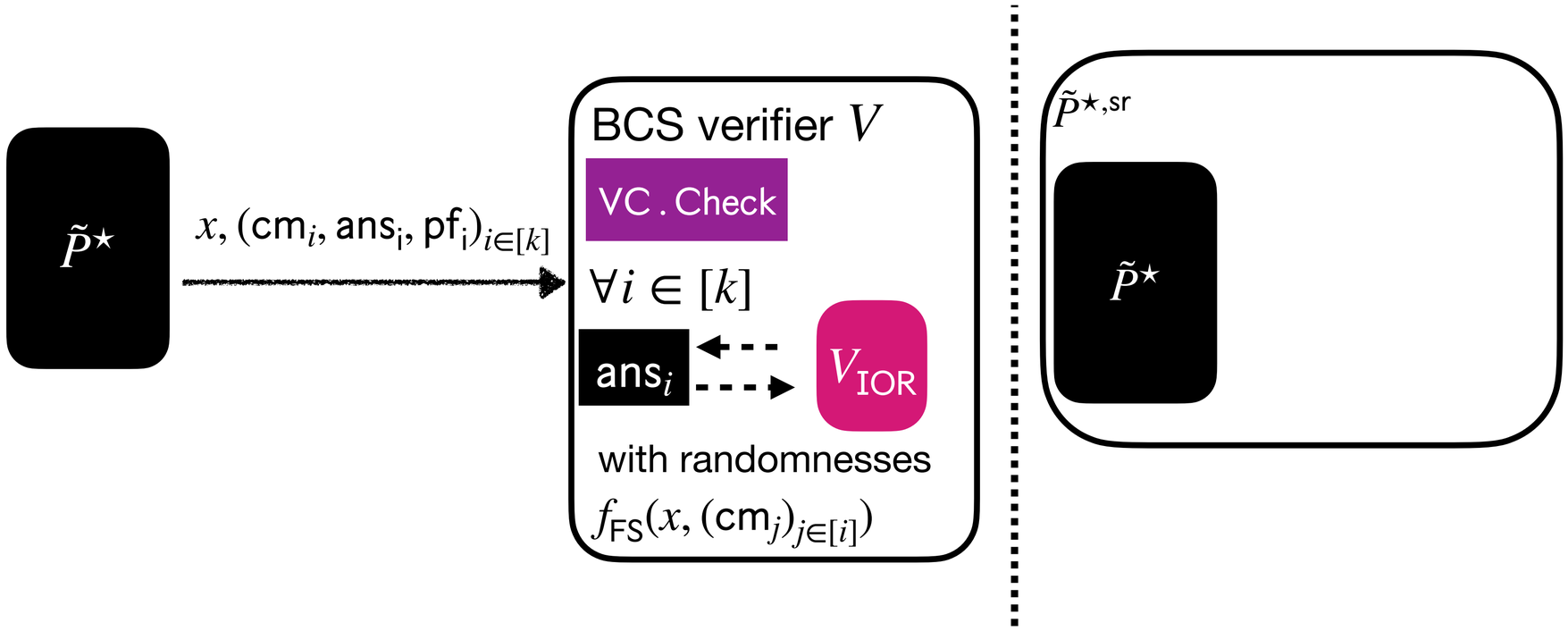
Quantum case



Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Step 3: how to derive the output

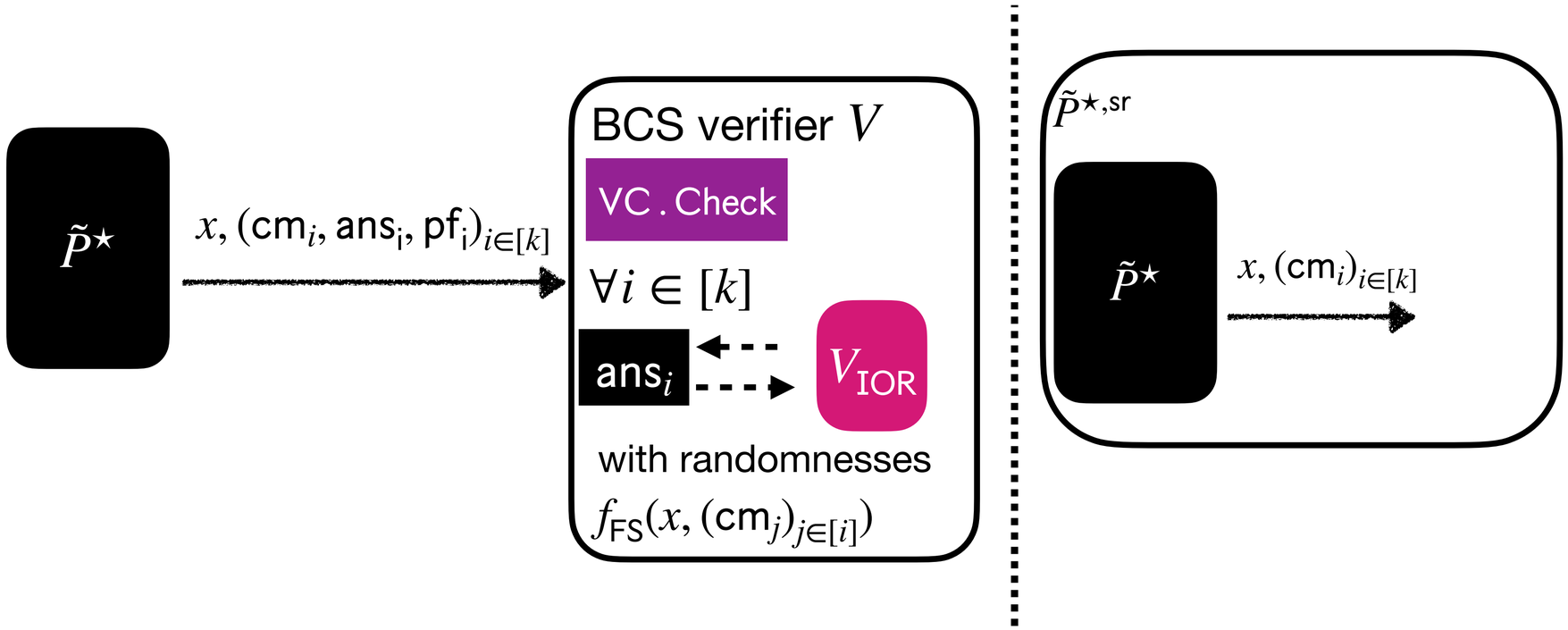
Quantum case



Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Step 3: how to derive the output

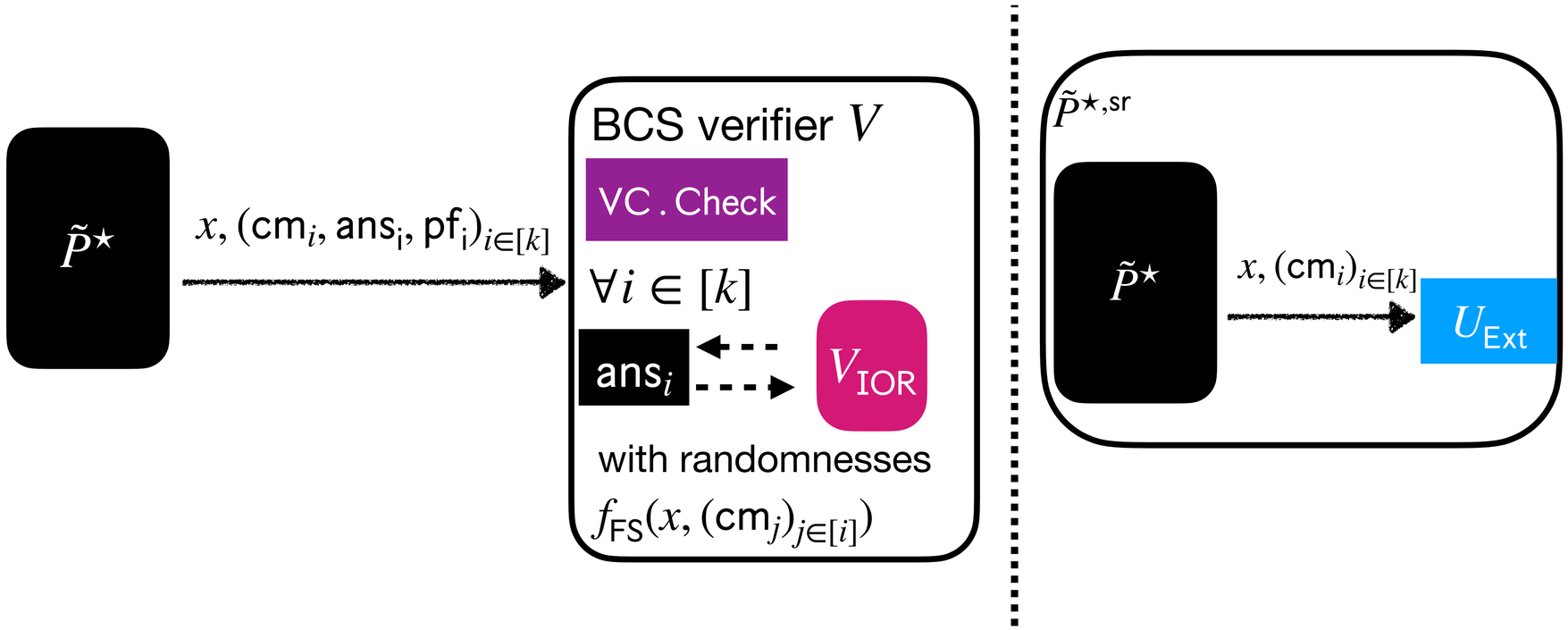
Quantum case



Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Step 3: how to derive the output

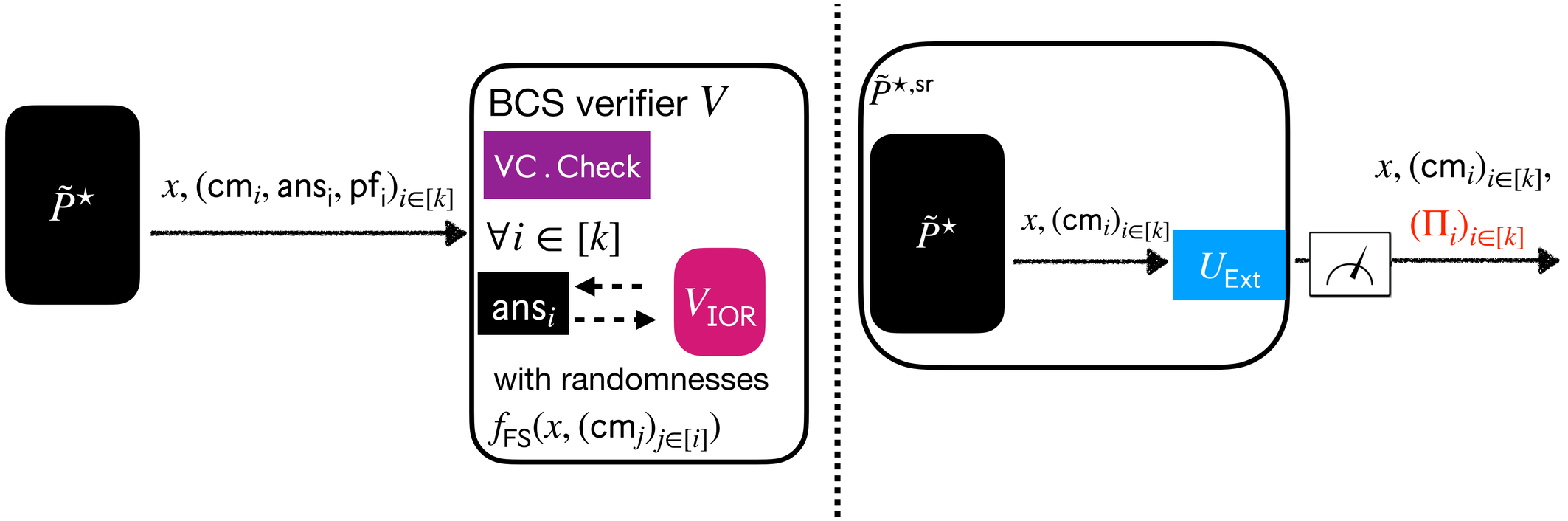
Quantum case



Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Step 3: how to derive the output

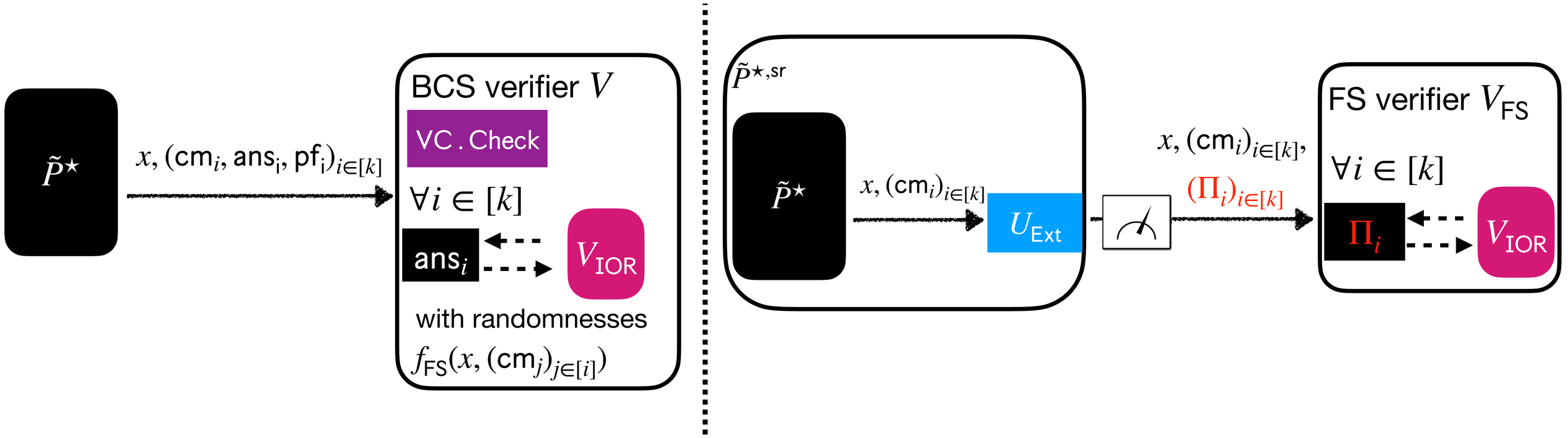
Quantum case



Our construction of $\tilde{P}^{\star,\mathsf{sr}}$

Step 3: how to derive the output

Quantum case



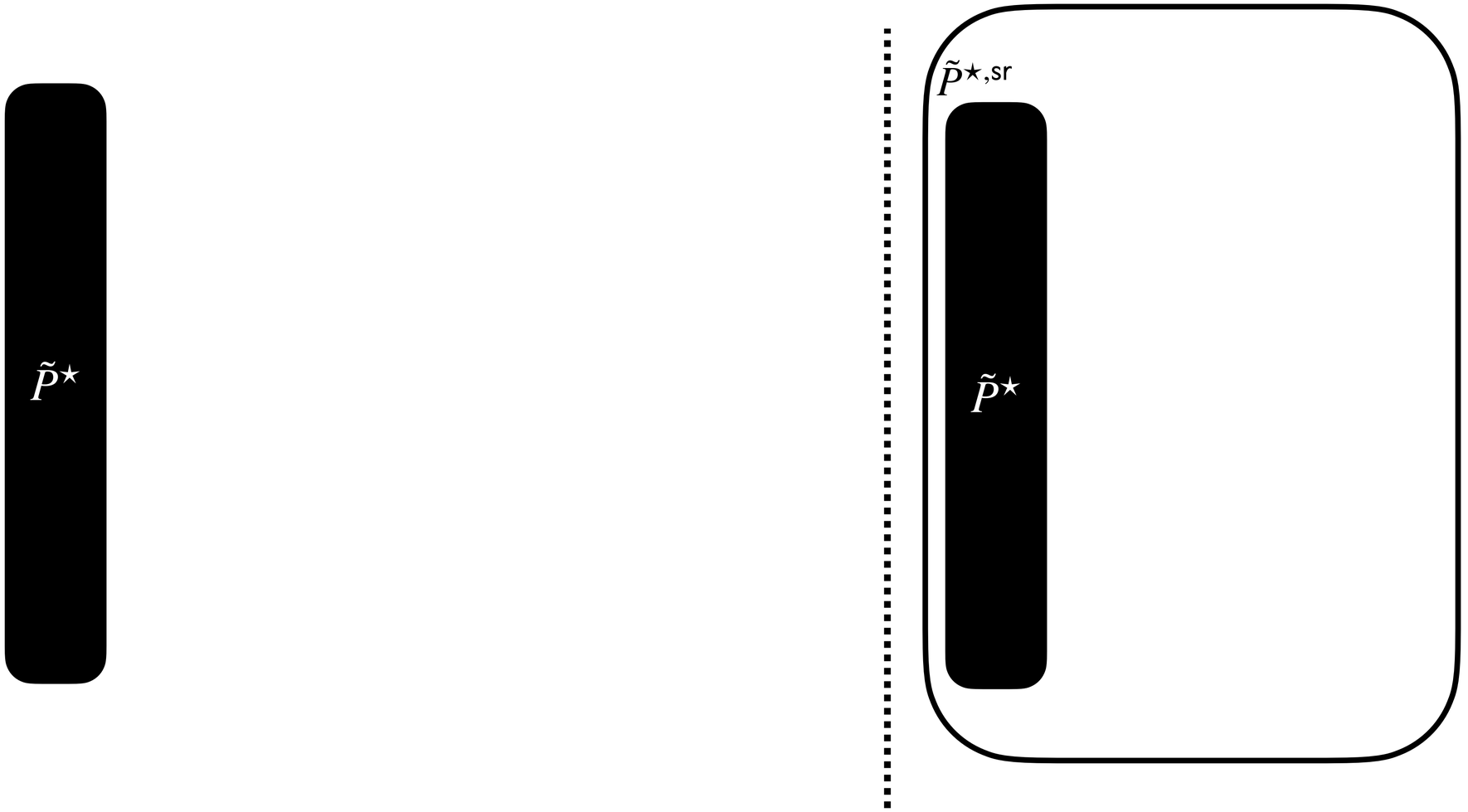
Quantum case

Our construction in summary: $\tilde{P}^{\star,\mathsf{sr}}$ simulates $\tilde{P}^{\star}$.

Quantum case

Our construction in summary: $\tilde{P}^{\star,\mathsf{sr}}$ simulates $\tilde{P}^{\star}$.

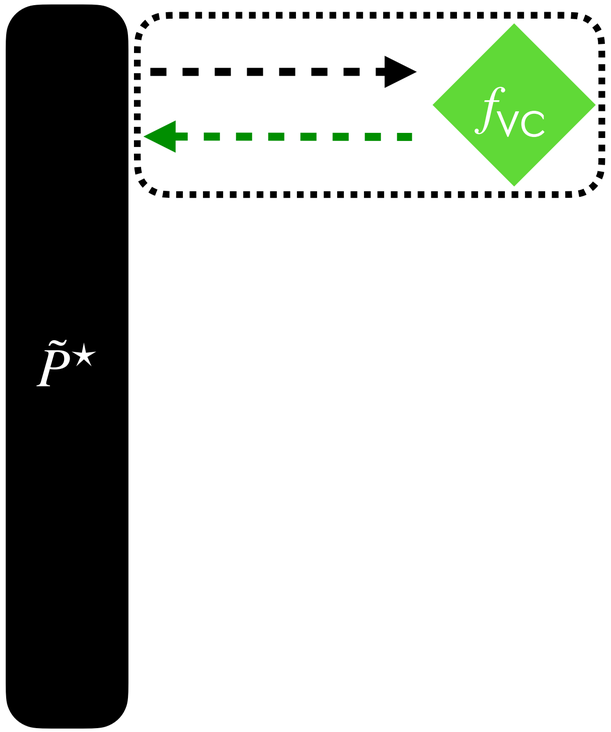
Quantum case

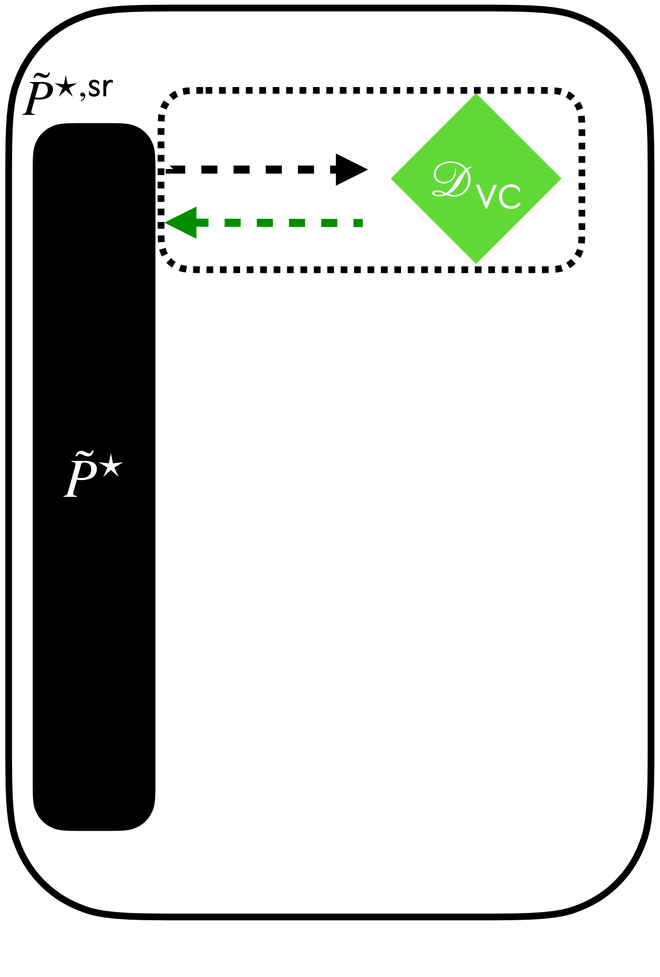


Our construction in summary: $\tilde{P}^{\star,\mathsf{sr}}$ simulates $\tilde{P}^{\star}$.

Quantum case

How to answer quantum f_{VC} queries?

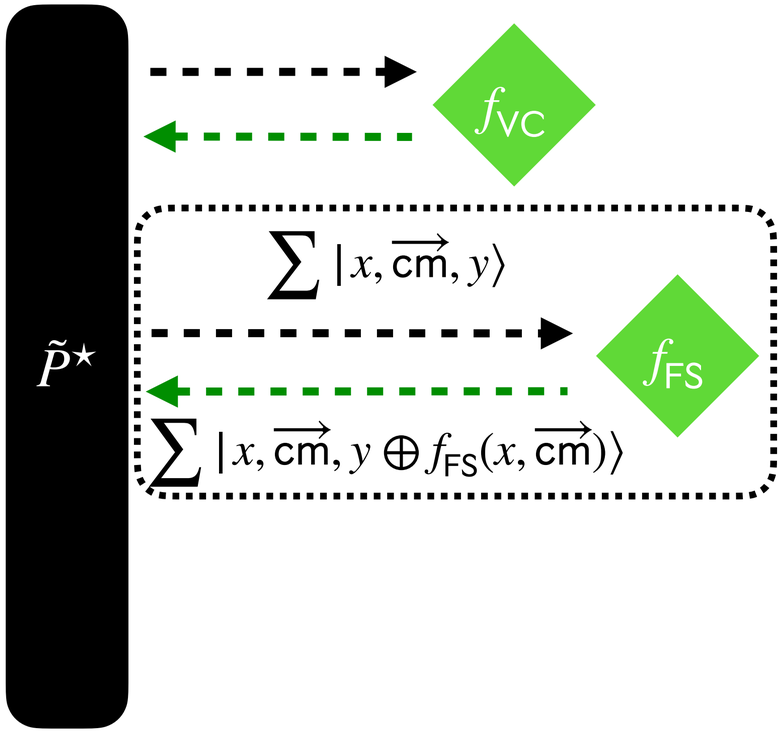


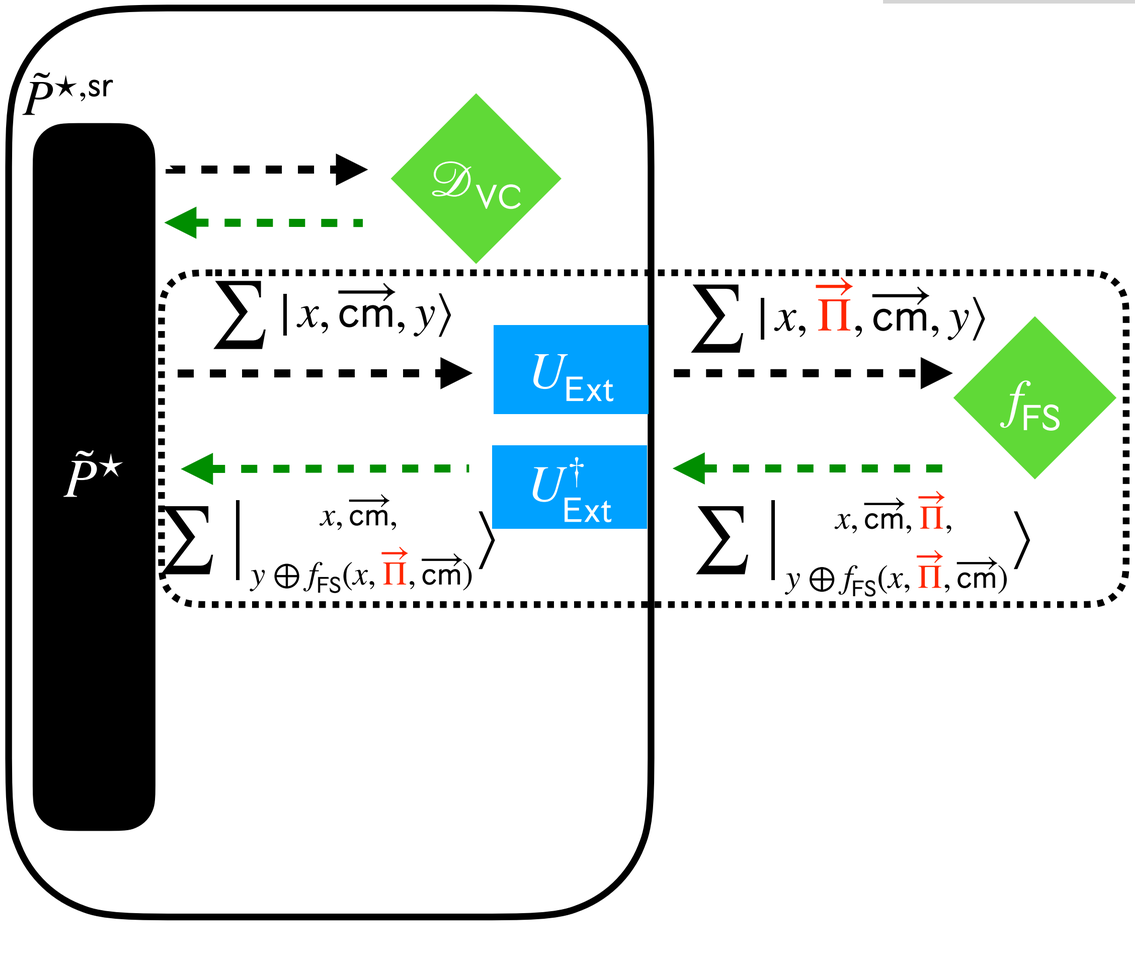


Our construction in summary: $\tilde{P}^{\star,\mathsf{sr}}$ simulates $\tilde{P}^{\star}$.

Quantum case

How to answer quantum f_{FS} queries?

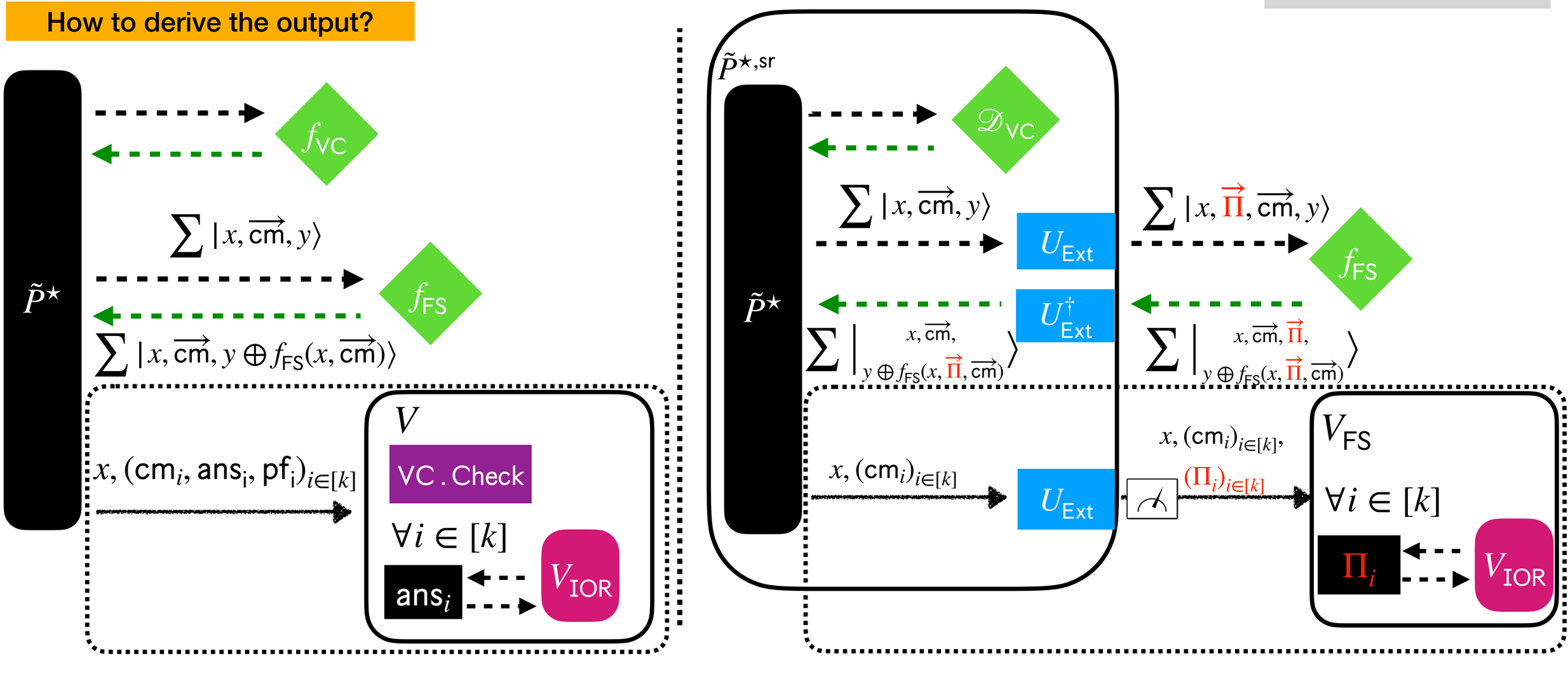




Our construction in summary: $\tilde{P}^{\star,\mathsf{sr}}$ simulates $\tilde{P}^{\star}$.

Quantum case

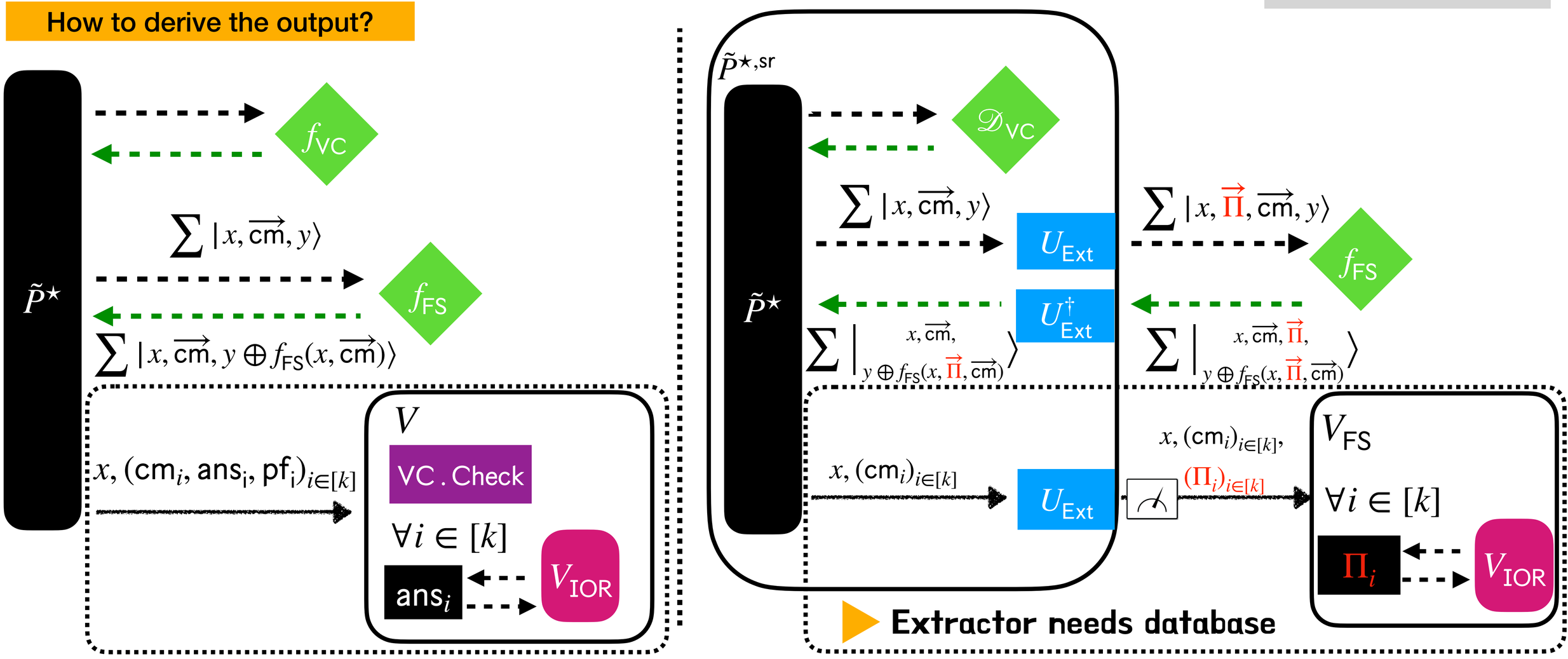
How to derive the output?



Our construction in summary: $\tilde{P}^{\star,\mathsf{sr}}$ simulates $\tilde{P}^{\star}$.

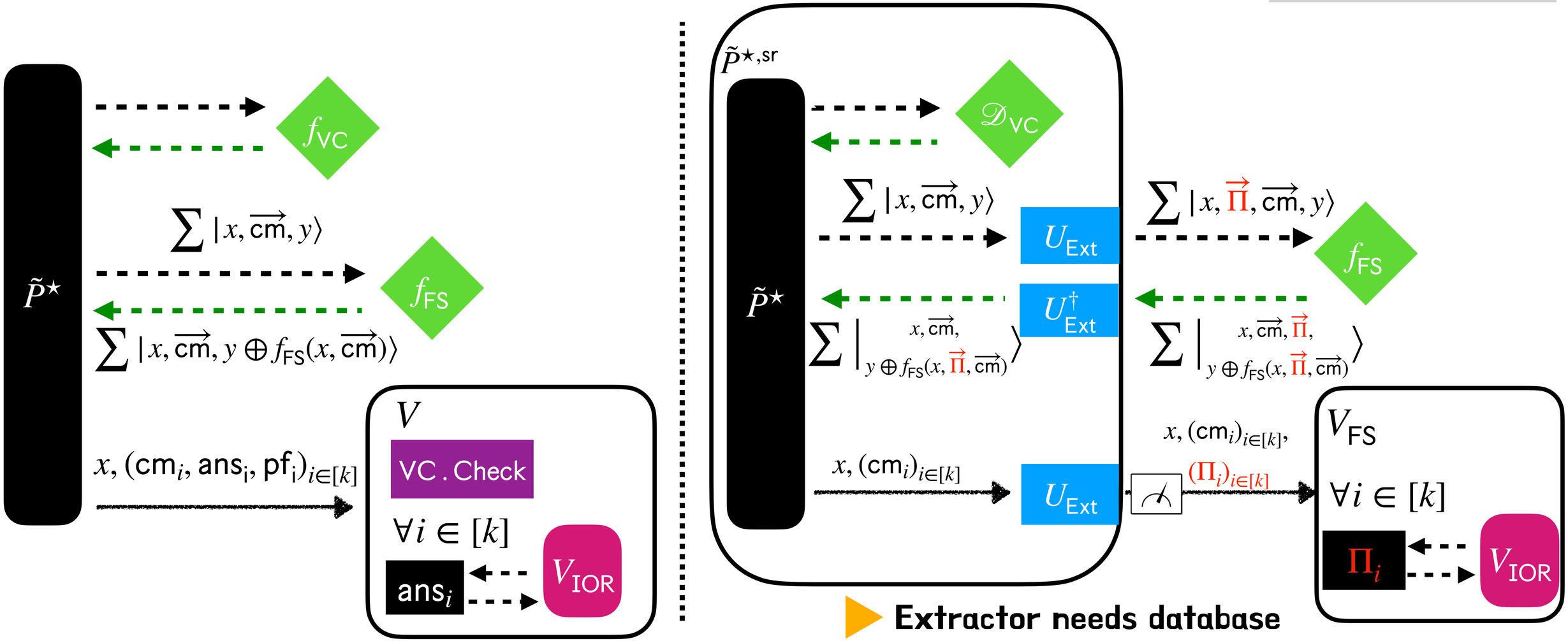
Quantum case

How to derive the output?



Our construction in summary: $\tilde{P}^{\star,\mathsf{sr}}$ simulates $\tilde{P}^{\star}$.

Quantum case



Goal: we want to show $\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon^{\star}_{\mathsf{VC}}$

Quantum case

Goal: we want to show

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon_{\mathsf{VC}}^{\star}$$

Quantum case

Goal: we want to show

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon_{\mathsf{VC}}^{\star}$$

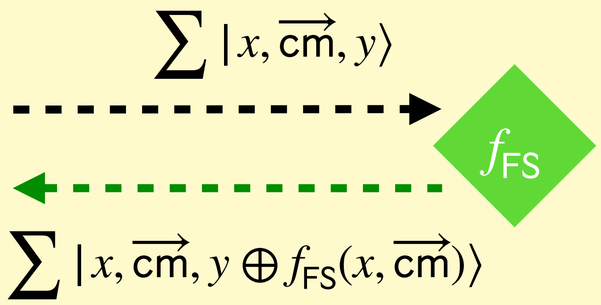
Difference 1

Quantum case

Goal: we want to show

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon_{\mathsf{VC}}^{\star}$$

Difference 1

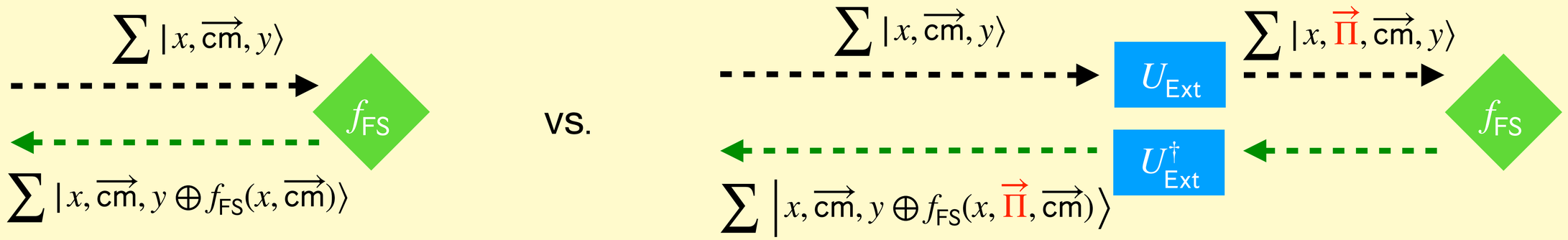


Quantum case

Goal: we want to show

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon_{\mathsf{VC}}^{\star}$$

Difference 1

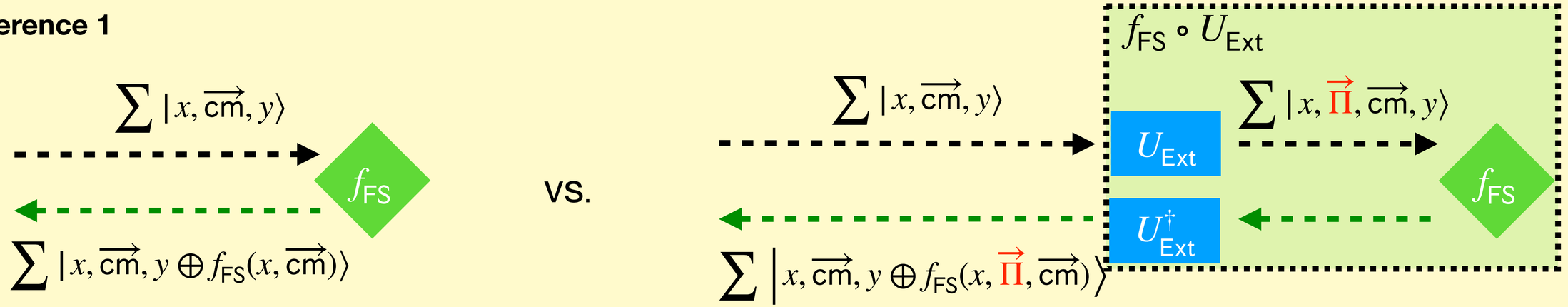


Quantum case

Goal: we want to show

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon_{\mathsf{VC}}^{\star}$$

Difference 1

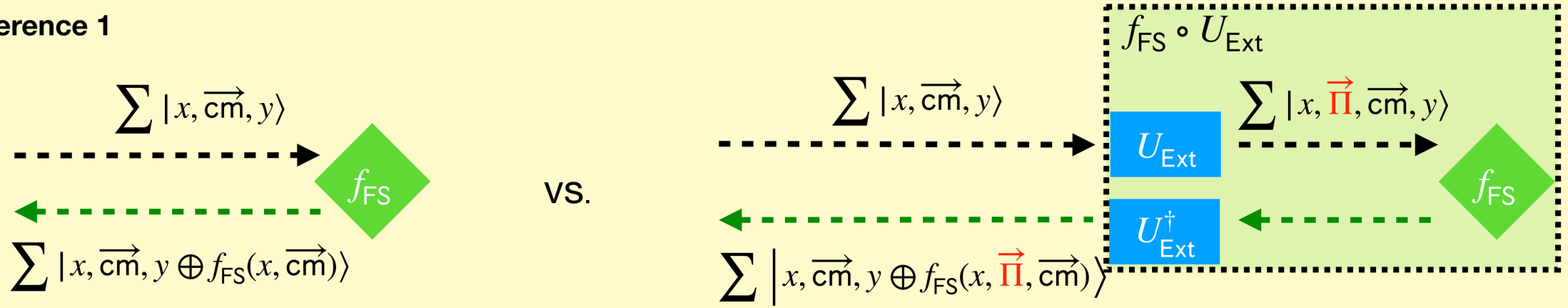


Quantum case

Goal: we want to show

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon_{\mathsf{VC}}^{\star}$$

Difference 1



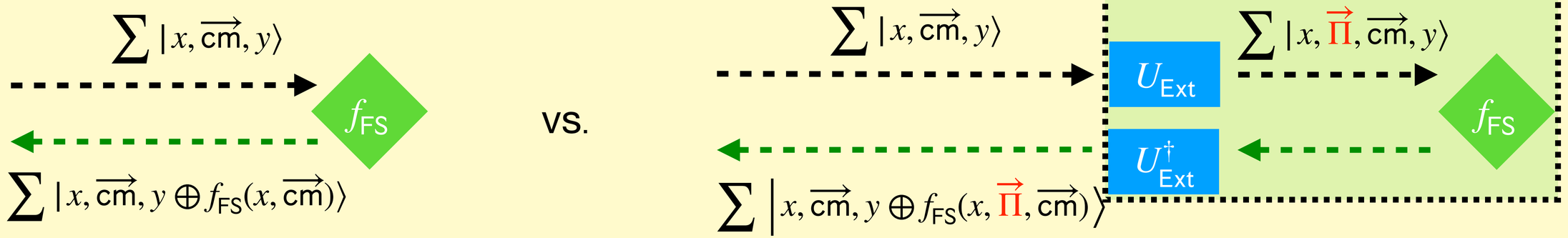
Our PQ VC Property 1: Online consistency

Quantum case

Goal: we want to show

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon_{\mathsf{VC}}^{\star}$$

Difference 1



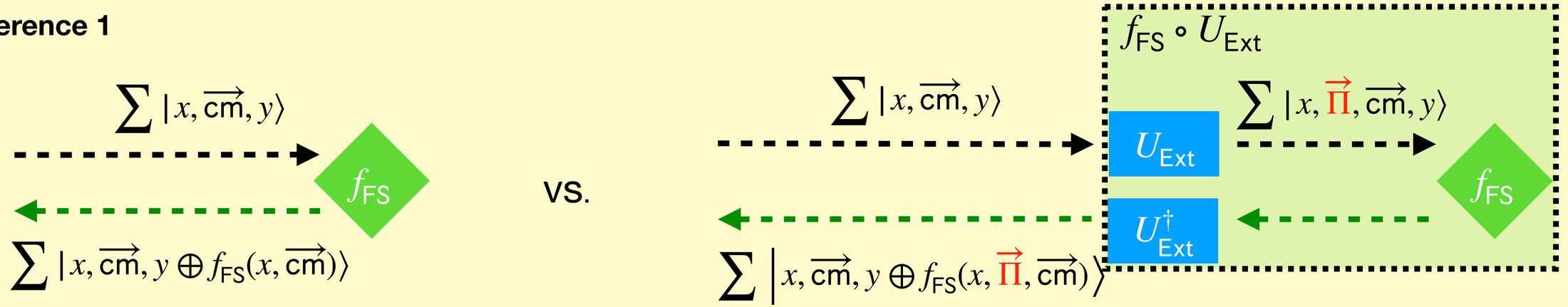
Our PQ VC Property 1: Online consistency

Quantum case

Goal: we want to show

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon_{\mathsf{VC}}^{\star}$$

Difference 1



Our PQ VC Property 1: Online consistency

Quantum case

Goal: we want to show

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon_{\mathsf{VC}}^{\star}$$

Quantum case

Goal: we want to show

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon_{\mathsf{VC}}^{\star}$$

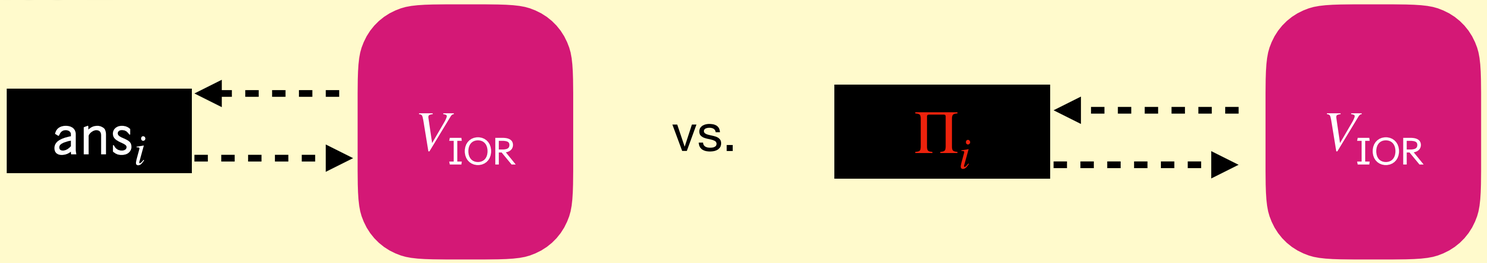
Difference 2

Quantum case

Goal: we want to show

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon_{\mathsf{VC}}^{\star}$$

Difference 2

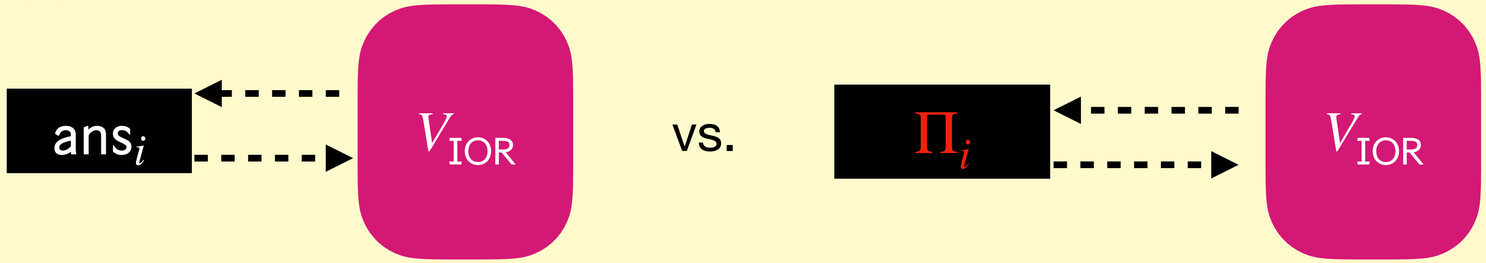


Quantum case

Goal: we want to show

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon_{\mathsf{VC}}^{\star}$$

Difference 2



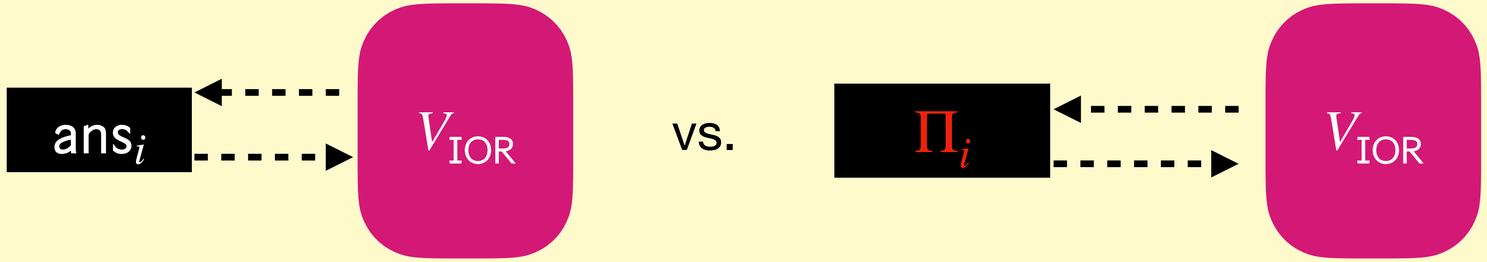
Our PQ VC Property 2: Offline extractability

Quantum case

Goal: we want to show

$$\Pr[\tilde{P}^{\star,\mathsf{sr}} \text{ wins PQSR game}] \geq \Pr[\tilde{P}^{\star} \text{ fools } V] - \epsilon^{\star}_{\mathsf{VC}}$$

Difference 2



Our PQ VC Property 2: Offline extractability

if VC . Check $= 1$

Is this the right VC PQ extractability definition?

Is this the right VC PQ extractability definition?

Similar to the classical VC extractability definition

🤔

Is this the right VC PQ extractability definition?

✅ Similar to the classical VC extractability definition

✅ Strong enough to prove BCS[IOR, VC] is post-quantum secure*

Is this the right VC PQ extractability definition?

Similar to the classical VC extractability definition

Strong enough to prove BCS[IOR, VC] is post-quantum secure*

More Challenges!

🤔 **Is this the right VC PQ extractability definition?**

✅ Similar to the classical VC extractability definition

✅ Strong enough to prove BCS[IOR, VC] is post-quantum secure*

😯 **More Challenges!**

*For instances that include oracles: require extra VC properties

🤔 **Is this the right VC PQ extractability definition?**

✅ Similar to the classical VC extractability definition

✅ Strong enough to prove BCS[IOR, VC] is post-quantum secure*

😲 **More Challenges!**

*For instances that include oracles: require extra VC properties

*For knowledge soundness: more caveats (later)

🤔 **Is this the right VC PQ extractability definition?**

✅ Similar to the classical VC extractability definition

✅ Strong enough to prove BCS[IOR, VC] is post-quantum secure*

? Does MT satisfy this?

😯 **More Challenges!**

*For instances that include oracles: require extra VC properties

*For knowledge soundness: more caveats (later)

Is this the right VC PQ extractability definition?

Similar to the classical VC extractability definition

Strong enough to prove BCS[IOR, VC] is post-quantum secure*

Does MT satisfy this?

Next part

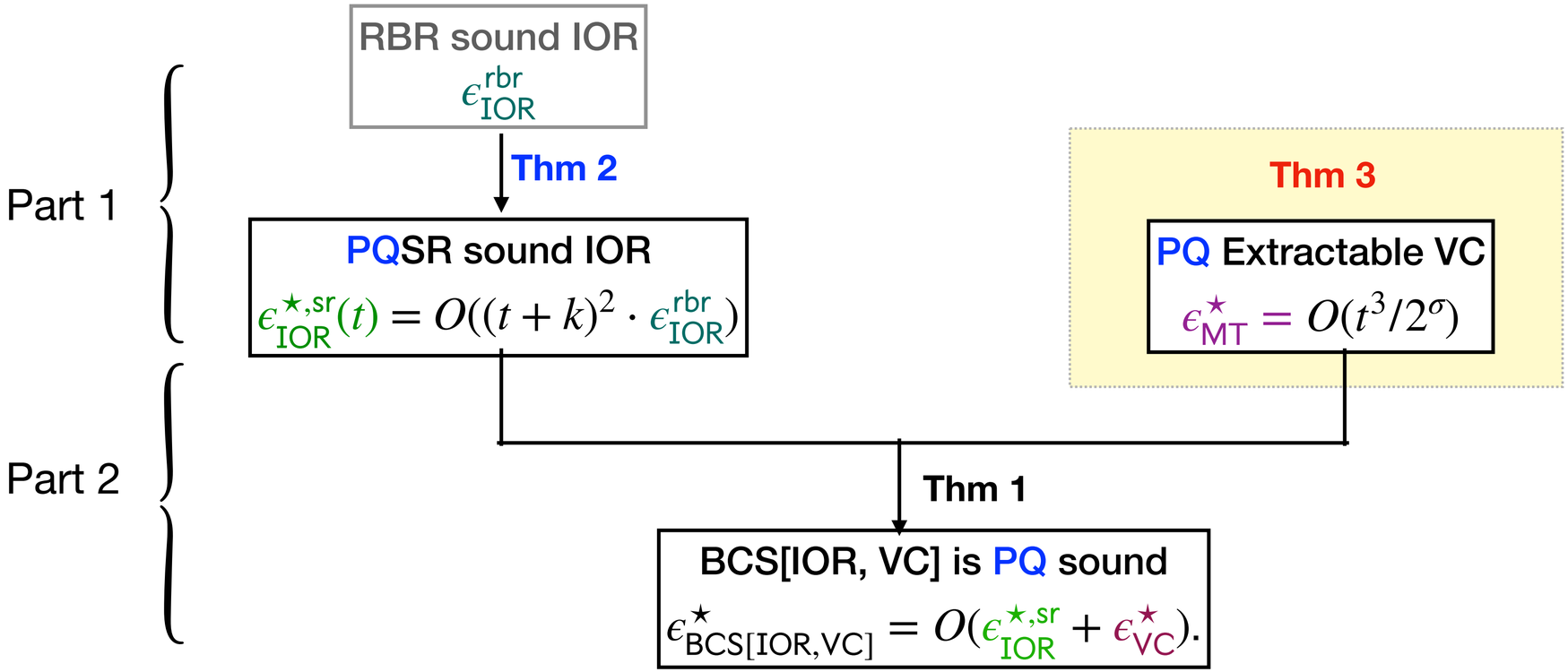
Takeaways

More Challenges!

*For instances that include oracles: require extra VC properties

*For knowledge soundness: more caveats (later)

MT has PQ extractability error $O(t^3/2^\sigma)$



Recall our PQ VC properties

Recall our PQ VC properties

Our PQ VC Property 1: Online consistency

Recall our PQ VC properties

Our PQ VC Property 1: Online consistency

Recall our PQ VC properties

Our PQ VC Property 1: Online consistency

Our PQ VC Property 2: Offline extractability

Recall our PQ VC properties

Our PQ VC Property 1: Online consistency

Our PQ VC Property 2: Offline extractability

if $\mathsf{VC.Check} = 1$

Recall our PQ VC properties

Our PQ VC Property 1: Online consistency

Proof uses the instability lemma from [CMS19].

Our PQ VC Property 2: Offline extractability

if $\mathsf{VC.Check} = 1$

Recall our PQ VC properties

Need new techniques

Our PQ VC Property 1: Online consistency

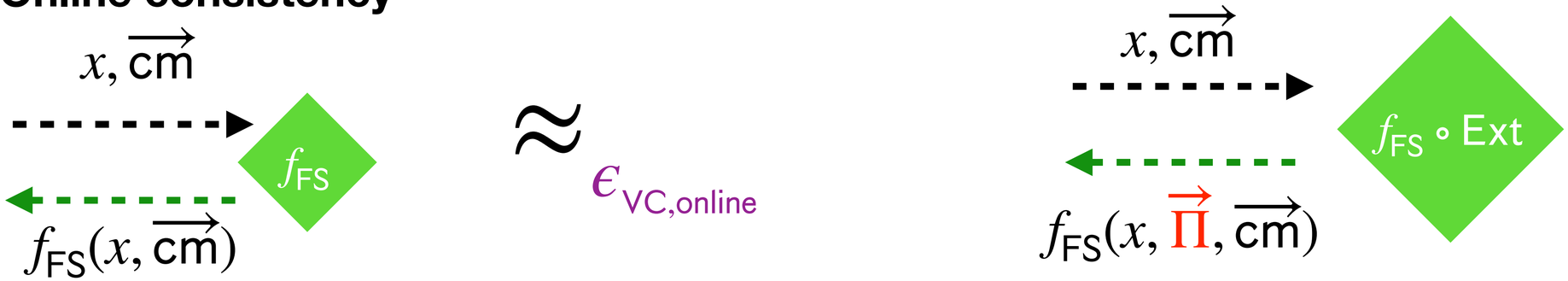
Proof uses the instability lemma from [CMS19].

Our PQ VC Property 2: Offline extractability

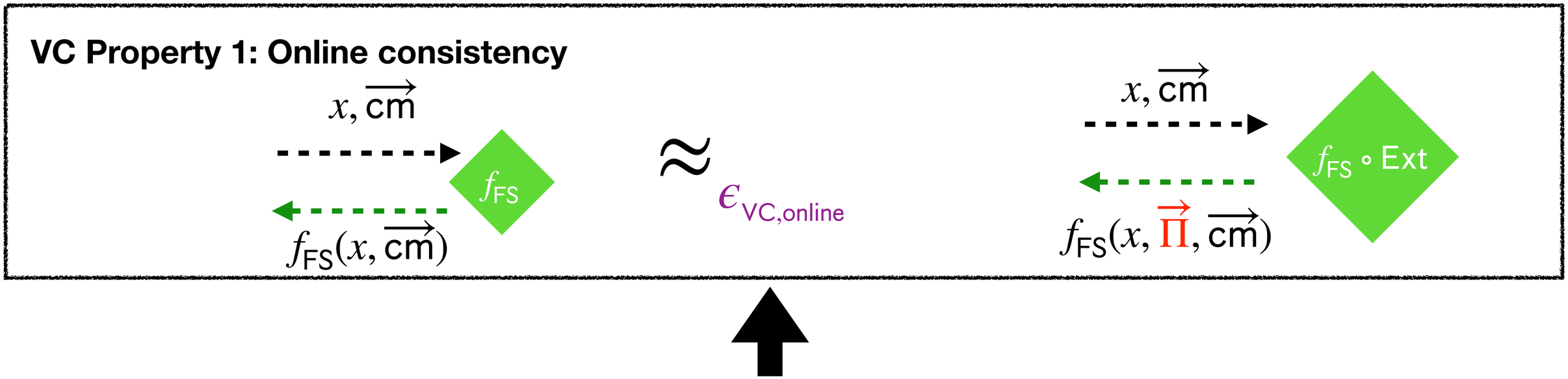
if $\mathsf{VC.Check} = 1$

Online consistency

VC Property 1: Online consistency



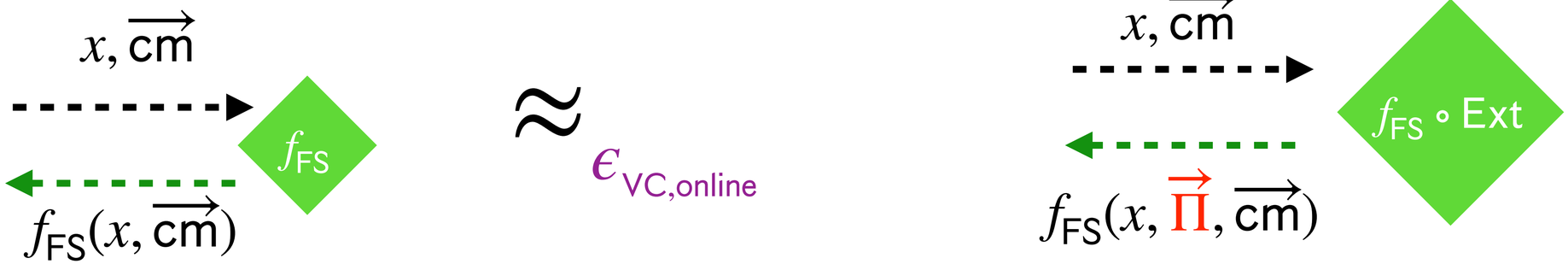
Online consistency



Extract later is the same as extract earlier

Online consistency

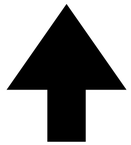
VC Property 1: Online consistency



Extract later is the same as extract earlier

i.e. every cm queried by f_{FS} is mapped to the same Π even after more VC queries

PQ Online consistency



Extract later is the same as extract earlier

i.e. every cm queried by f_{FS} is mapped to the same Π even after more VC queries

PQ Online consistency

Our PQ VC Property 1: Online consistency

Extract later is the same as extract earlier

i.e. every cm queried by f_{FS} is mapped to the same Π even after more VC queries

PQ Online consistency

Our PQ VC Property 1: Online consistency

But now cm is in superposition

Extract later is the same as extract earlier

i.e. every cm queried by f_{FS} is mapped to the same Π even after more VC queries

PQ Online consistency

Our PQ VC Property 1: Online consistency

But now cm is in superposition

Extract later is the same as extract earlier

i.e. every cm queried by f_{FS} is mapped to the same Π even after more VC queries

Our solution: Consider the extraction results for **every** cm in the **database of** f_{FS}

PQ Online consistency

Our PQ VC Property 1: Online consistency

But now cm is in superposition

Extract later is the same as extract earlier

i.e. every cm queried by f_{FS} is mapped to the same Π even after more VC queries

Our solution: Consider the extraction results for **every** cm in the **database of** f_{FS}

and show the results does not change after more quantum f_{VC} queries

PQ Online consistency

Our PQ VC Property 1: Online consistency

But now cm is in superposition

Extract later is the same as extract earlier

i.e. every cm queried by f_{FS} is mapped to the same Π even after more VC queries

Our solution: Consider the extraction results for **every** cm in the **database of** f_{FS}

and show the results does not change after more quantum f_{VC} queries

We want some unitary that reads $\mathscr{D}_{\mathsf{FS}}$ and do extraction coherently on those cm to almost commute with a VC quantum query !

PQ Online consistency

Our PQ VC Property 1: Online consistency

But now cm is in superposition

Extract later is the same as extract earlier

i.e. every cm queried by f_{FS} is mapped to the same Π even after more VC queries

Our solution: Consider the extraction r

and show the results doe

For this talk, let's consider U_{Ext}
that does extraction on **only one** cm coherently.

We want some unitary that reads $\mathscr{D}_{\mathsf{FS}}$ and do extraction coherently on those cm to almost commute with a VC quantum query !

Prior commutator bounds

Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

Implies that for MT, U_{Ext} almost commutes with the quantum query.

Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

Implies that for MT, U_{Ext} almost commutes with the quantum query.

But not tight enough.

Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

Implies that for MT, U_{Ext} almost commutes with the quantum query.

But not tight enough.

Even worse when there are a lot of cm ….

Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

Implies that for MT, U_{Ext} almost commutes with the quantum query.

But not tight enough.

Even worse when there are a lot of cm ….

[CMS19]: The commutator between a database property P and a quantum query can be bounded by a **classical** quantity.

Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

Implies that for MT, U_{Ext} almost commutes with the quantum query.

But not tight enough.

Even worse when there are a lot of cm ….

[CMS19]: The commutator between a database property P and a quantum query can be bounded by a **classical** quantity.

For database property P, consider the **binary partition** $= \{P, \bar{P}\}$,

Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

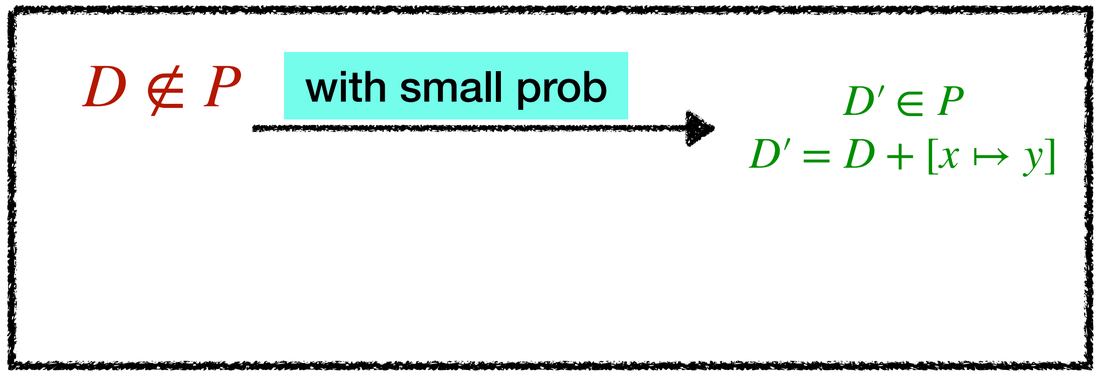
Implies that for MT, U_{Ext} almost commutes with the quantum query.

But not tight enough.

Even worse when there are a lot of cm ….

[CMS19]: The commutator between a database property P and a quantum query can be bounded by a **classical** quantity.

For database property P, consider the **binary partition** $= \{P, \bar{P}\}$,



Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

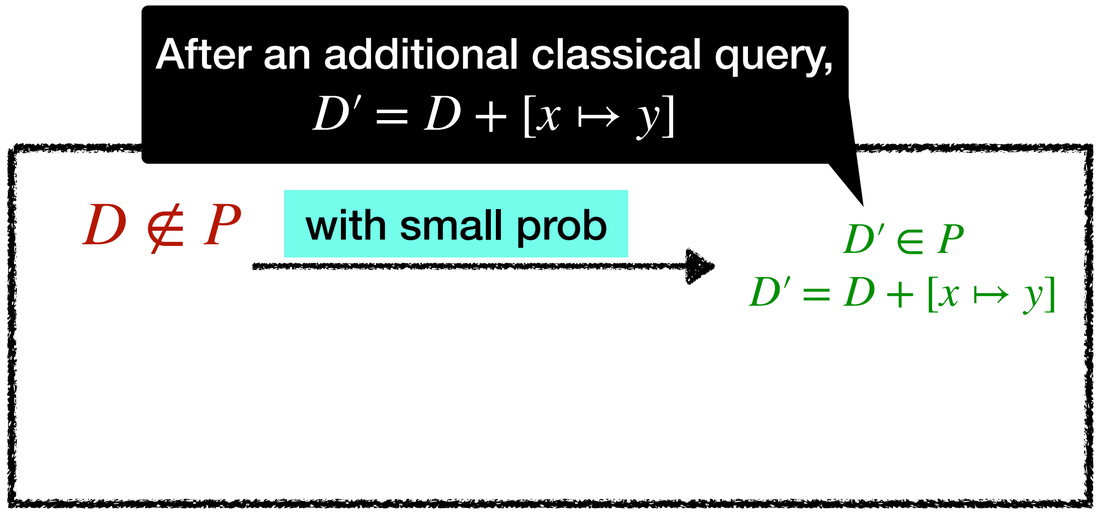
Implies that for MT, U_{Ext} almost commutes with the quantum query.

But not tight enough.

Even worse when there are a lot of cm ….

[CMS19]: The commutator between a database property P and a quantum query can be bounded by a **classical** quantity.

For database property P, consider the **binary partition** $= \{P, \bar{P}\}$,



Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

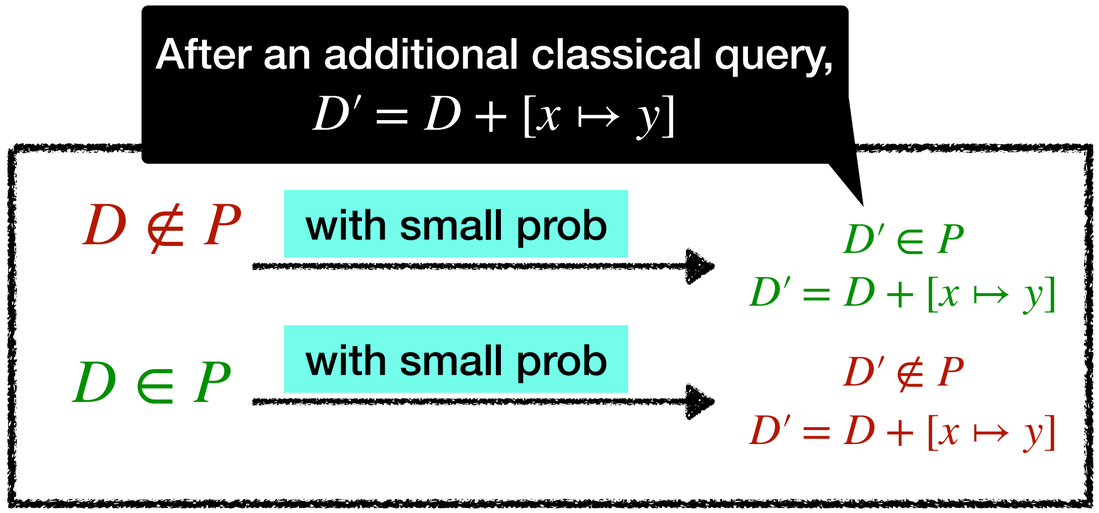
Implies that for MT, U_{Ext} almost commutes with the quantum query.

But not tight enough.

Even worse when there are a lot of cm ….

[CMS19]: The commutator between a database property P and a quantum query can be bounded by a **classical** quantity.

For database property P, consider the **binary partition** $= \{P, \bar{P}\}$,



Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

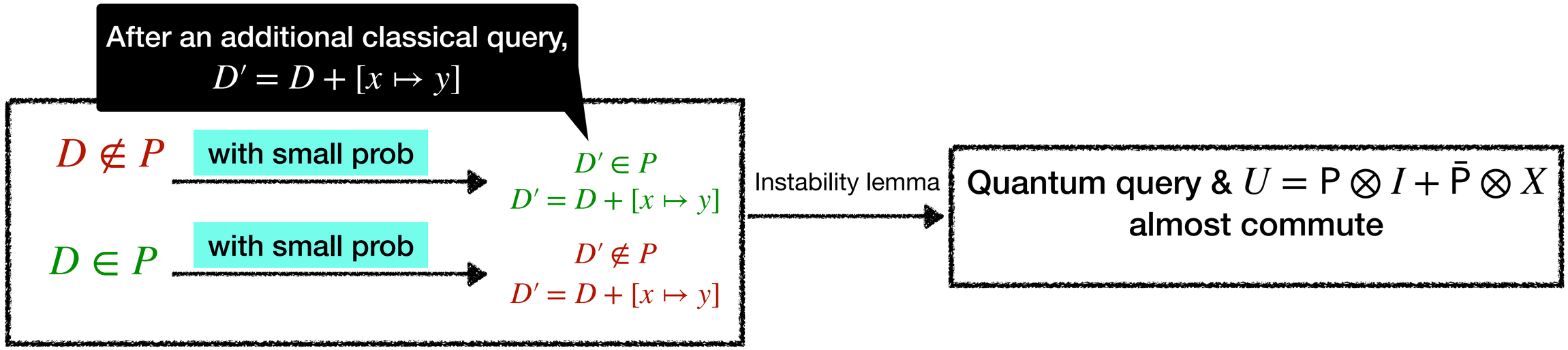
Implies that for MT, U_{Ext} almost commutes with the quantum query.

But not tight enough.

Even worse when there are a lot of cm ….

[CMS19]: The commutator between a database property P and a quantum query can be bounded by a **classical** quantity.

For database property P, consider the **binary partition** $= \{P, \bar{P}\}$,



Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

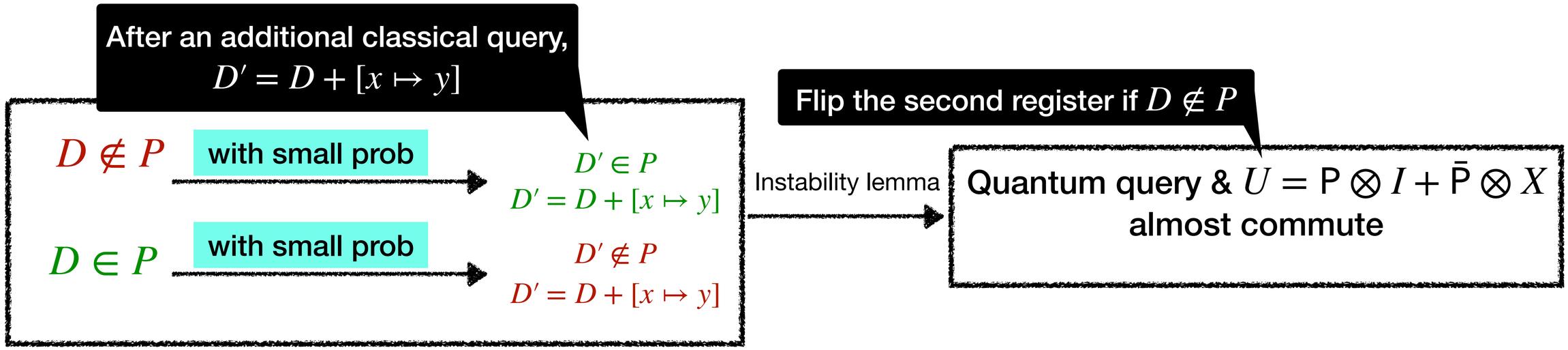
Implies that for MT, U_{Ext} almost commutes with the quantum query.

But not tight enough.

Even worse when there are a lot of cm ….

[CMS19]: The commutator between a database property P and a quantum query can be bounded by a **classical** quantity.

For database property P, consider the **binary partition** $= \{P, \bar{P}\}$,



Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

Implies that for MT, U_{Ext} almost commutes with the quantum query.

But not tight enough.

Even worse when there are a lot of cm ….

[CMS19]: The commutator between a database property P and a quantum query can be bounded by a **classical** quantity.

For database property P, consider the **binary partition** $= \{P, \bar{P}\}$,

After an additional classical query, $D' = D + [x \mapsto y]$

$D \notin P$ with small prob → $D' \in P$, $D' = D + [x \mapsto y]$

$D \in P$ with small prob → $D' \notin P$, $D' = D + [x \mapsto y]$

Instability lemma →

Projector on the databases in P

Flip the second register if $D \notin P$

Quantum query & $U = \mathsf{P} \otimes I + \bar{\mathsf{P}} \otimes X$ almost commute

Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

Implies that for MT, U_{Ext} almost commutes with the quantum query.

But not tight enough.

Even worse when there are a lot of cm ….

[CMS19]: The commutator between a database property P and a quantum query can be bounded by a **classical** quantity.

For database property P, consider the **binary partition** $= \{P, \bar{P}\}$,

After an additional classical query, $D' = D + [x \mapsto y]$

$D \notin P$ s.t. $|D| \le t$ — with small prob → $D' \in P$, $D' = D + [x \mapsto y]$

$D \in P$ s.t. $|D| \le t$ — with small prob → $D' \notin P$, $D' = D + [x \mapsto y]$

Instability lemma

Flip the second register if $D \notin P$

Projector on the databases in P

Quantum query & $U = \mathsf{P} \otimes I + \bar{\mathsf{P}} \otimes X$ almost commute

Conditioned on $|D| \le t$

Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

Implies that for MT, U_{Ext} almost commutes with the quantum query.

But not tight enough.

Even worse when there are a lot of cm ….

[CMS19]: The commutator between a database property P and a quantum query can be bounded by a **classical** quantity.

For database property P, consider the **binary partition** $= \{P, \bar{P}\}$,

After an additional classical query, $D' = D + [x \mapsto y]$

$D \notin P$ s.t. $|D| \le t$ — with small prob → $D' \in P$, $D' = D + [x \mapsto y]$

$D \in P$ s.t. $|D| \le t$ — with small prob → $D' \notin P$, $D' = D + [x \mapsto y]$

Instability lemma

Projector on the databases in P

Flip the second register if $D \notin P$

Quantum query & $U = \mathsf{P} \otimes I + \bar{\mathsf{P}} \otimes X$ almost commute

Conditioned on $|D| \le t$

A classical quantity that is usually easy to analyze

Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

Implies that for MT, U_{Ext} almost commutes with the quantum query.

But not tight enough.

Even worse when there are a lot of cm ….

[CMS19]: The commutator between a database property P and a quantum query can be bounded by a **classical** quantity.

For database property P, consider the **binary partition** $= \{P, \bar{P}\}$,

After an additional classical query, $D' = D + [x \mapsto y]$

$D \notin P$ s.t. $|D| \le t$ — with small prob → $D' \in P$, $D' = D + [x \mapsto y]$

$D \in P$ s.t. $|D| \le t$ — with small prob → $D' \notin P$, $D' = D + [x \mapsto y]$

Instability lemma

Projector on the databases in P

Flip the second register if $D \notin P$

Quantum query & $U = \mathsf{P} \otimes I + \bar{\mathsf{P}} \otimes X$ almost commute

Conditioned on $|D| \le t$

A classical quantity that is usually easy to analyze

It does not work for U_{Ext}.
U_{Ext} does not form a binary partition.

Prior commutator bounds

[DFMS22]: For basic commitments, U_{Ext} almost commutes with the quantum query.

Implies that for MT, U_{Ext} almost commutes with the quantum query.

But not tight enough.

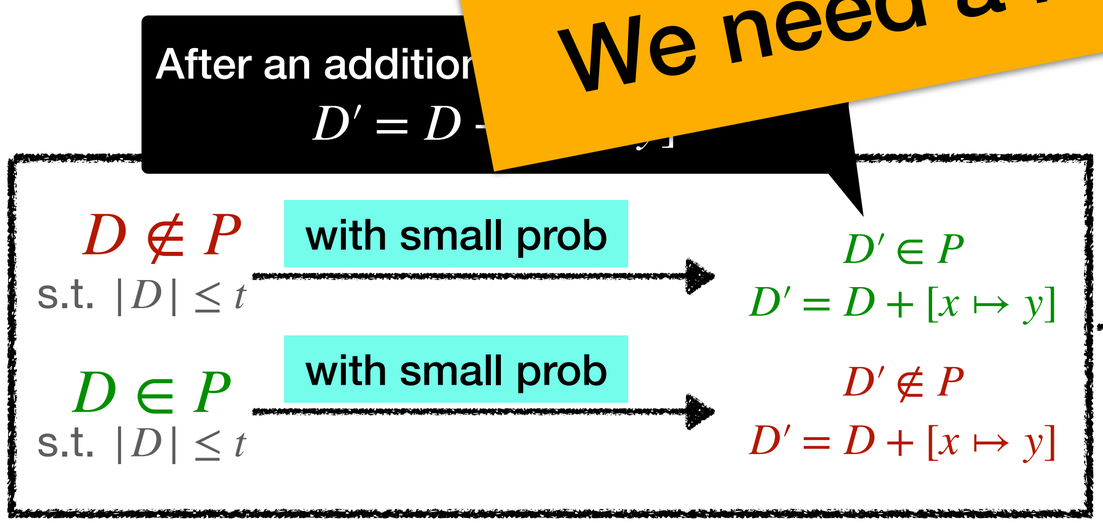
Even worse when there are a lot of cm

[CMS19]: The commutator between a database propert… …ounded by a **classical** quantity.

For database property P, consi…

We need a new technique!

After an additio… $D' = D + …$



Instability lemma

Projector on the databases in P

Flip the second register if $D \notin P$

Quantum query & $U = \mathsf{P} \otimes I + \bar{\mathsf{P}} \otimes X$ almost commute

Conditioned on $|D| \le t$

A classical quantity that is usually easy to analyze

It does not work for U_{Ext}.
U_{Ext} does not form a binary partition.

Our generalized instability lemma

Our generalized instability lemma

For **any partition** $\{P_i\}_i$,

Our generalized instability lemma

For **any partition** $\{P_i\}_i$,

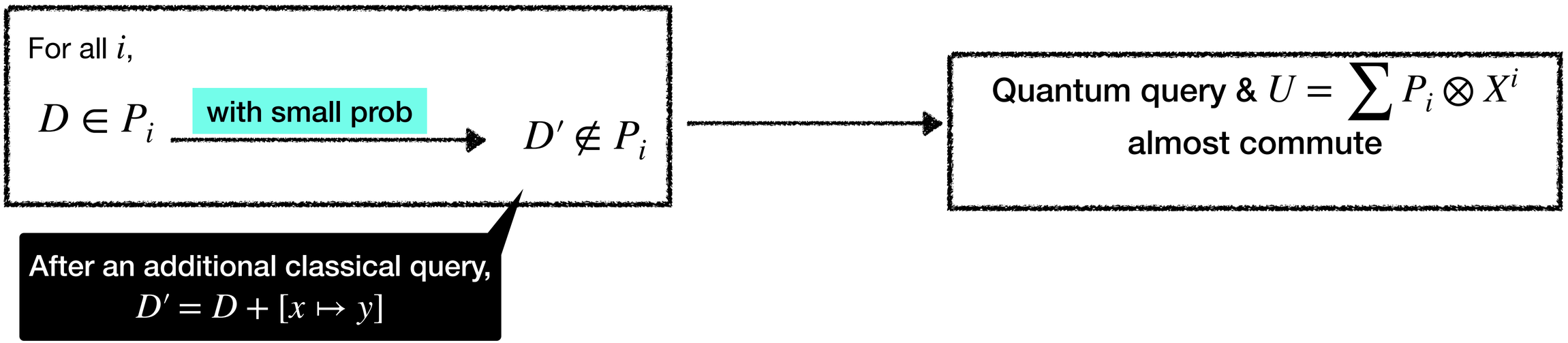
For all i,

$D \in P_i$ **with small prob** $\longrightarrow$ $D' \notin P_i$

After an additional classical query,
$D' = D + [x \mapsto y]$

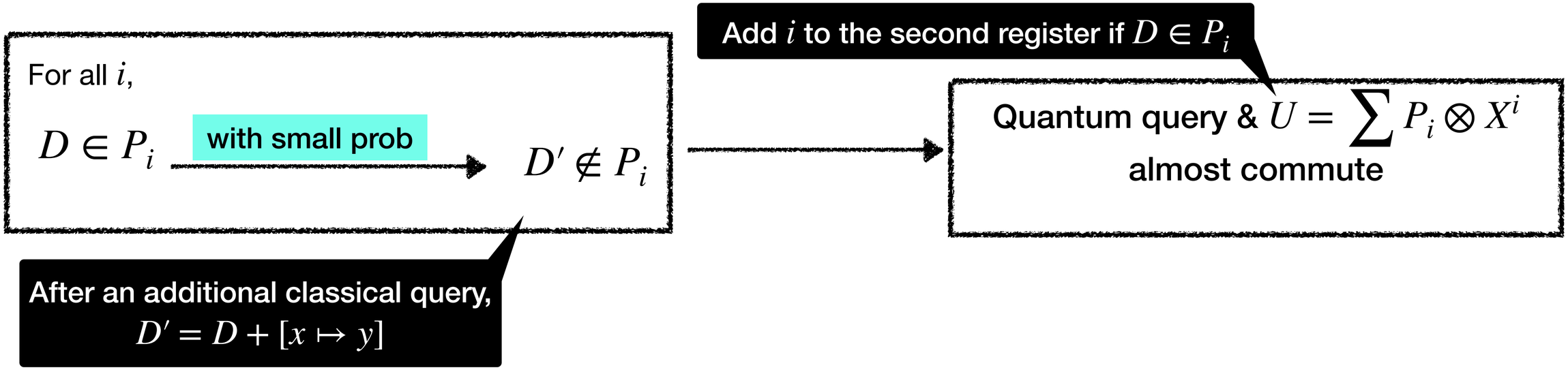
Our generalized instability lemma

For **any partition** $\{P_i\}_i$,



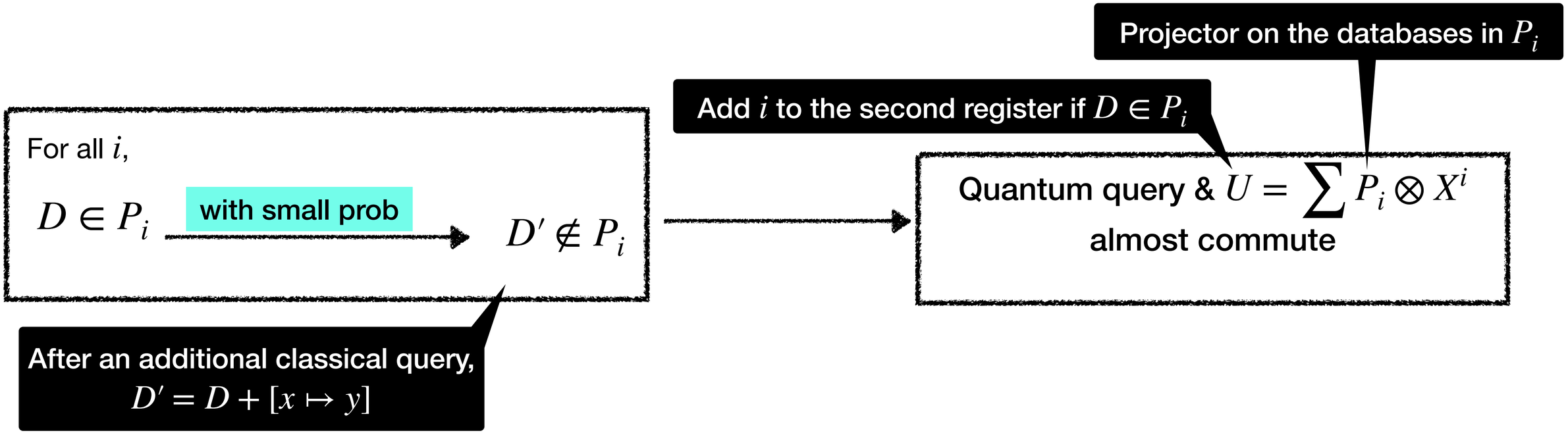
Our generalized instability lemma

For **any partition** $\{P_i\}_i$,



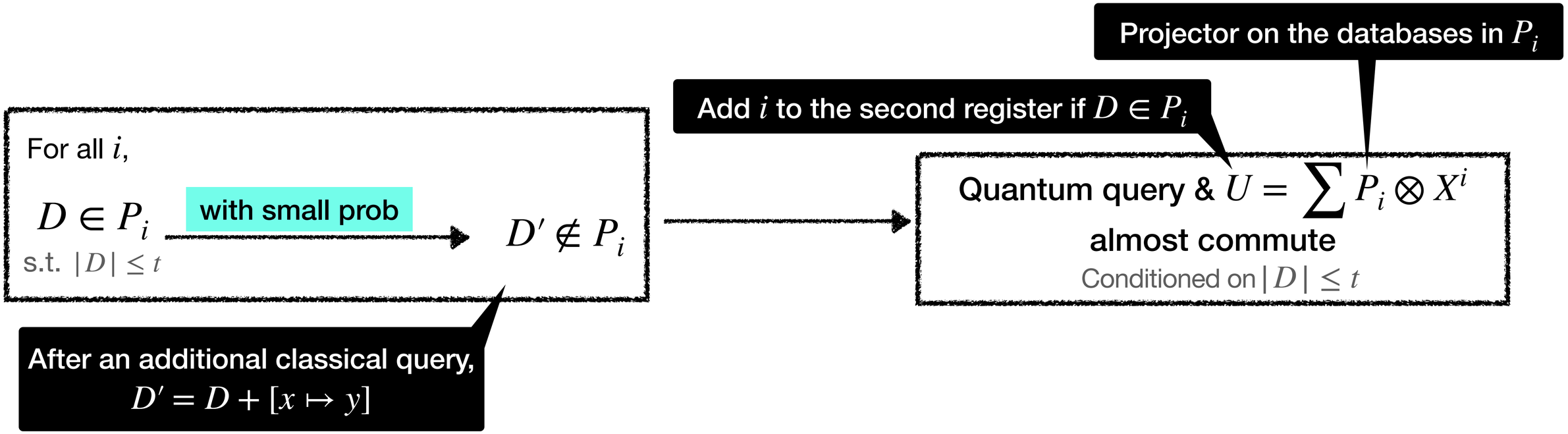
Our generalized instability lemma

For **any partition** $\{P_i\}_i$,



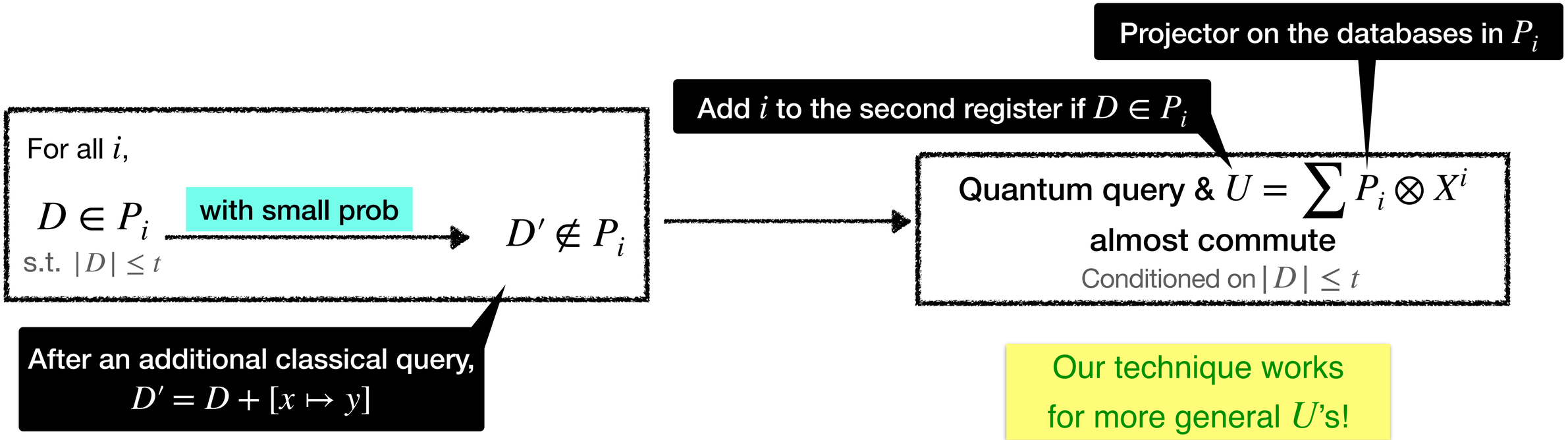
Our generalized instability lemma

For **any partition** $\{P_i\}_i$,



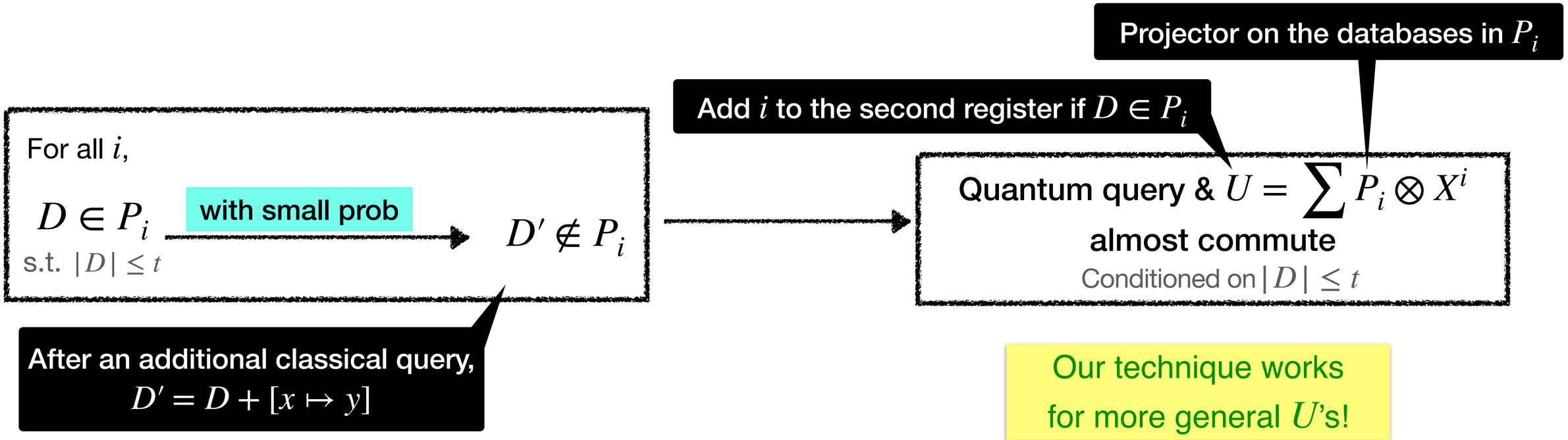
Our generalized instability lemma

For **any partition** $\{P_i\}_i$,



Our generalized instability lemma

For **any partition** $\{P_i\}_i$,



Our technique works
for more general U's!

This in particular includes U_{Ext} for MT!

Our generalized instability lemma

For **any partition** $\{P_i\}_i$,

Projector on the databases in P_i

Add i to the second register if $D \in P_i$

For all i,

$D \in P_i$ s.t. $|D| \leq t$ —with small prob→ $D' \notin P_i$

After an additional classical query, $D' = D + [x \mapsto y]$

→ Quantum query & $U = \sum P_i \otimes X^i$ almost commute

Conditioned on $|D| \leq t$

Our technique works for more general U's!

This in particular includes U_{Ext} for MT!

And this also includes the unitary that reads $\mathscr{D}_{\mathsf{FS}}$ and does the extraction!

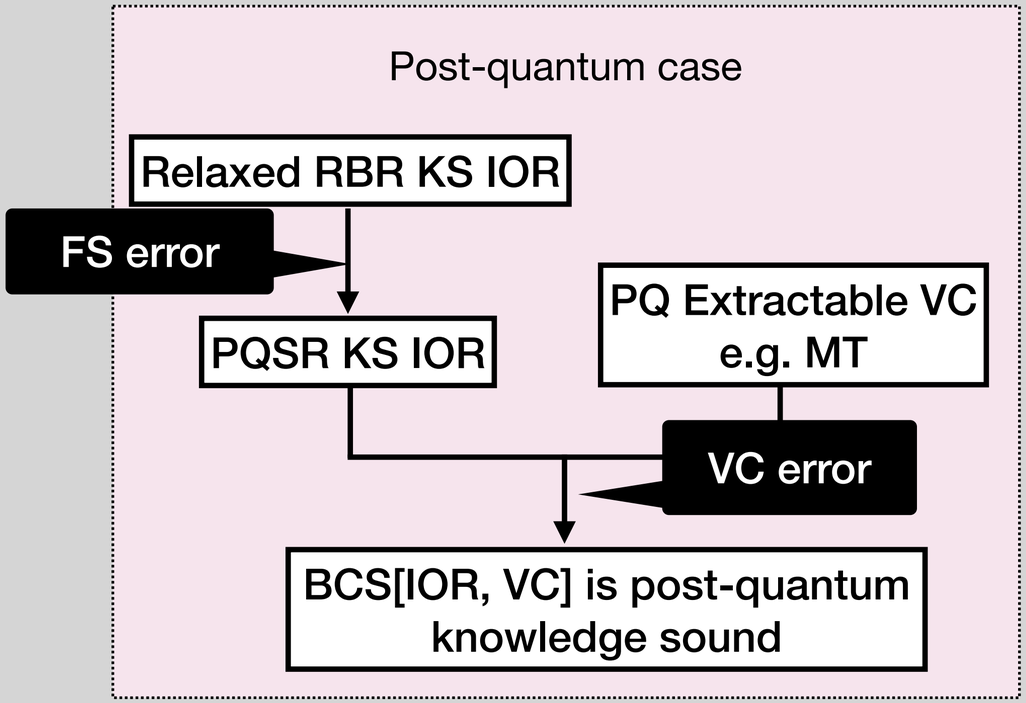
Takeaways

Takeaways

- BCS[IOR, MT] is a post-quantum straight-line knowledge sound SNRDX in the QROM
  if the underlying IOR satisfies (even a weaker variant of) round-by-round knowledge soundness.

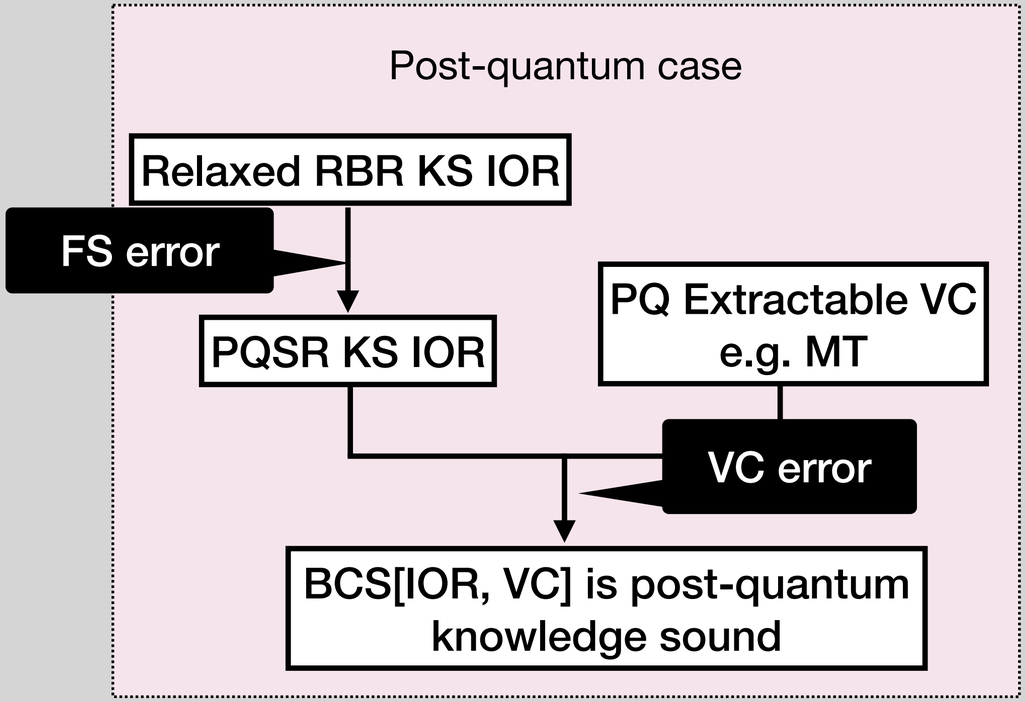
Takeaways

- BCS[IOR, MT] is a post-quantum straight-line knowledge sound SNRDX in the QROM if the underlying IOR satisfies (even a weaker variant of) round-by-round knowledge soundness.
- Our proof analyzes the error from FS and MT separately through an intermediate FS-style security notion (PQSR), mirroring the classical proof.



Takeaways

- BCS[IOR, MT] is a post-quantum straight-line knowledge sound SNRDX in the QROM if the underlying IOR satisfies (even a weaker variant of) round-by-round knowledge soundness.
- Our proof analyzes the error from FS and MT separately through an intermediate FS-style security notion (PQSR), mirroring the classical proof.



Thank you!

ia.cr/2025/2166

More technical details

More technical details

Can we allow adversaries to query different oracles simultaneously?

More technical details

Can we allow adversaries to query different oracles simultaneously?

YES!

More technical details

Can we allow adversaries to query different oracles simultaneously?

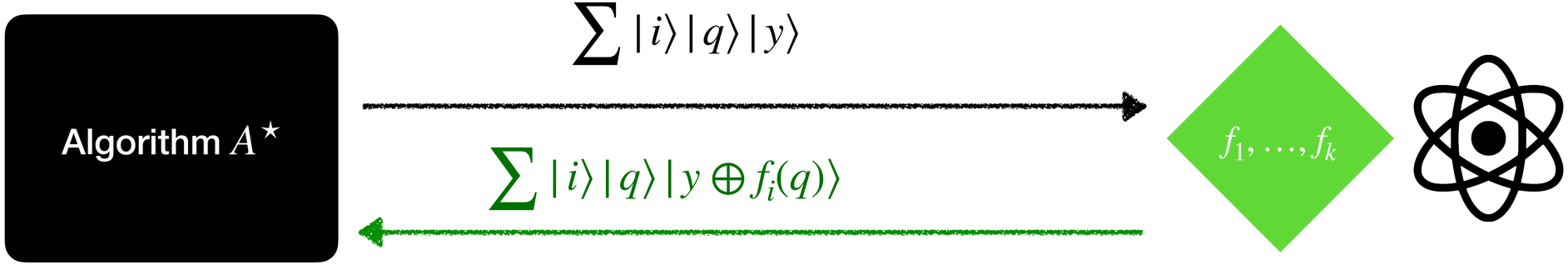
YES! Superposition query model.

More technical details

Can we allow adversaries to query different oracles simultaneously?

YES!

Superposition query model.



More technical details

More technical details

Classical knowledge soundness: There exists an extractor E such that for every efficient adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^f \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E(x, \tilde{\pi}, x', w', D) \end{array}\right] \leq \kappa$$

More technical details

Classical knowledge soundness: There exists an extractor E such that for every efficient adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^f \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E(x, \tilde{\pi}, x', w', D) \end{array}\right] \leq \kappa$$

Can we have a reasonable post-quantum knowledge soundness definition?

More technical details

Classical knowledge soundness: There exists an extractor E such that for every efficient adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^f \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E(x, \tilde{\pi}, x', w', D) \end{array}\right] \leq \kappa$$

Can we have a reasonable post-quantum knowledge soundness definition?

A naïve proposal: $D \to \mathscr{D}$

More technical details

Classical knowledge soundness: There exists an extractor E such that for every efficient adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^f \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E(x, \tilde{\pi}, x', w', D) \end{array}\right] \leq \kappa$$

Can we have a reasonable post-quantum knowledge soundness definition?

A naïve proposal: $D \to \mathscr{D}$

PQ knowledge soundness (first attempt): There exists an extractor E such that for every efficient quantum adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^f \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E(x, \tilde{\pi}, x', w', \mathscr{D}) \end{array}\right] \leq \kappa$$

More technical details

Classical knowledge soundness: There exists an extractor E such that for every efficient adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^f \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E(x, \tilde{\pi}, x', w', D) \end{array}\right] \leq \kappa$$

Can we have a reasonable post-quantum knowledge soundness definition?

A naïve proposal: $D \to \mathscr{D}$

PQ knowledge soundness (first attempt): There exists an extractor E such that for every efficient quantum adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^f \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E(x, \tilde{\pi}, x', w', \mathscr{D}) \end{array}\right] \leq \kappa$$

Problem: no sequential composition. $\tilde{P}$ cannot run E, and E might destroy $\mathscr{D}$ arbitrarily.

More technical details

Classical knowledge soundness: There exists an extractor E such that for every efficient adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^f \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E(x, \tilde{\pi}, x', w', D) \end{array}\right] \leq \kappa$$

Can we have a reasonable post-quantum knowledge soundness definition?

More technical details

Classical knowledge soundness: There exists an extractor E such that for every efficient adversary $\tilde{P}$,

$$\Pr\left[(x',w') \in R' \wedge (x,w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x,\tilde{\pi},w') \leftarrow \tilde{P}^f \\ x' \leftarrow V^f(x,\tilde{\pi}) \\ w \leftarrow E(x,\tilde{\pi},x',w',D) \end{array}\right] \leq \kappa$$

Can we have a reasonable post-quantum knowledge soundness definition?

YES!

More technical details

Classical knowledge soundness: There exists an extractor E such that for every efficient adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^f \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E(x, \tilde{\pi}, x', w', D) \end{array}\right] \leq \kappa$$

Can we have a reasonable post-quantum knowledge soundness definition?

YES!

PQ knowledge soundness: There exists an extractor E such that for every efficient quantum adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^{f, U_{\mathsf{Extract}}} \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E^{f, U_{\mathsf{Extract}}}(x, \tilde{\pi}, x', w') \end{array}\right] \leq \kappa$$

More technical details

Classical knowledge soundness: There exists an extractor E such that for every efficient adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^f \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E(x, \tilde{\pi}, x', w', D) \end{array}\right] \leq \kappa$$

Can we have a reasonable post-quantum knowledge soundness definition?

YES!

PQ knowledge soundness: There exists an extractor E such that for every efficient quantum adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^{f, U_{\mathsf{Extract}}} \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E^{f, U_{\mathsf{Extract}}}(x, \tilde{\pi}, x', w') \end{array}\right] \leq \kappa$$

sequential composition ✓

More technical details

Classical knowledge soundness: There exists an extractor E such that for every efficient adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^f \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E(x, \tilde{\pi}, x', w', D) \end{array}\right] \leq \kappa$$

Can we have a reasonable post-quantum knowledge soundness definition?

YES!

PQ knowledge soundness: There exists an extractor E such that for every efficient quantum adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^{f, U_{\mathsf{Extract}}} \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E^{f, U_{\mathsf{Extract}}}(x, \tilde{\pi}, x', w') \end{array}\right] \leq \kappa$$

So VC adversary should be strengthened as well…

sequential composition ✓

More technical details

Classical knowledge soundness: There exists an extractor E such that for every efficient adversary $\tilde{P}$,

$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^f \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E(x, \tilde{\pi}, x', w', D) \end{array}\right] \leq \kappa$$

Can we have a reasonable post-quantum knowledge soundness definition?

YES!

PQ knowledge soundness: There exists an extractor E such that for every efficient quantum adversary $\tilde{P}$,

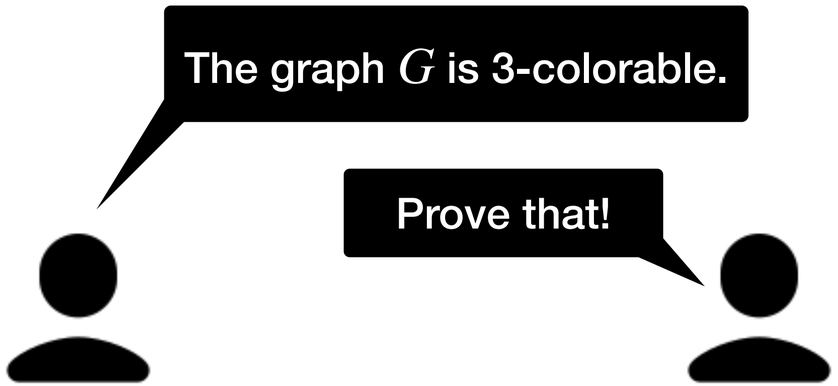
$$\Pr\left[(x', w') \in R' \wedge (x, w) \notin R \;\middle|\; \begin{array}{c} f \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \tilde{\pi}, w') \leftarrow \tilde{P}^{f, U_{\mathsf{Extract}}} \\ x' \leftarrow V^f(x, \tilde{\pi}) \\ w \leftarrow E^{f, U_{\mathsf{Extract}}}(x, \tilde{\pi}, x', w') \end{array}\right] \leq \kappa$$

So VC adversary should be strengthened as well…

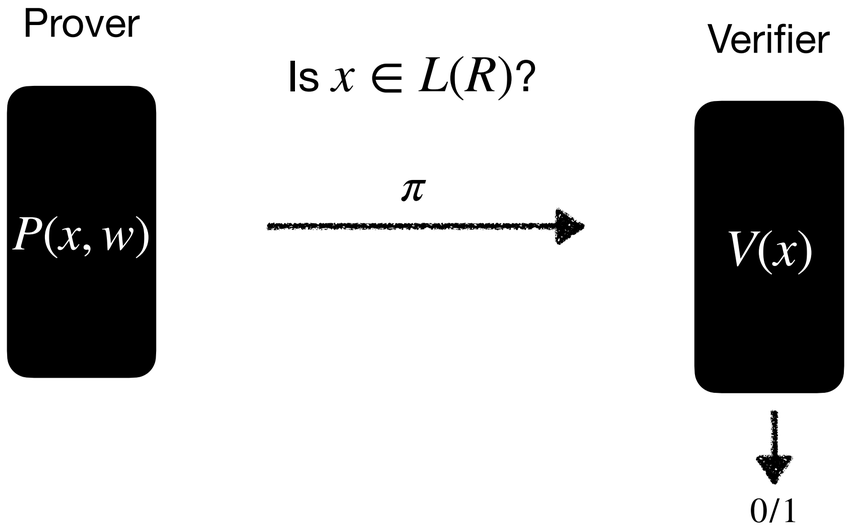
sequential composition ✓

And more…

Succinct non-interactive arguments (SNARGs)



But a coloring of G is too long…



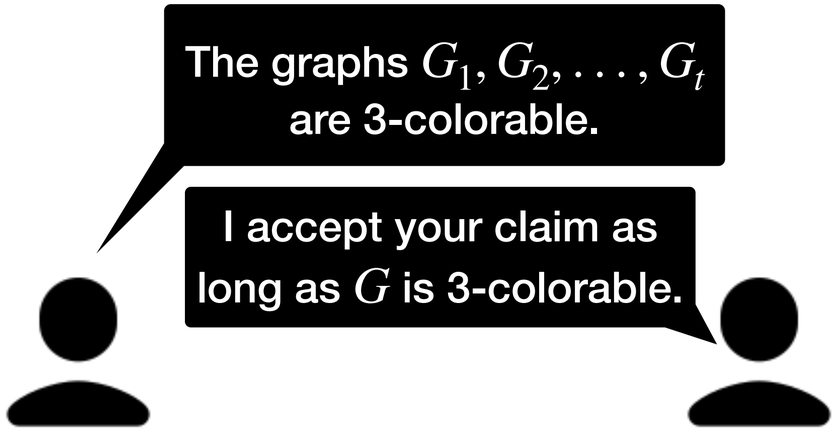
Completeness: $\forall\,(x,w) \in R,\ \Pr\left[1 \leftarrow V(x,\pi) \;\middle|\; \pi \leftarrow P(x,w)\right] = 1.$

Soundness: For every efficient adversary $\tilde{P}$, $\Pr\left[x \notin L(R) \wedge 1 \leftarrow V(x,\tilde{\pi}) \;\middle|\; (x,\tilde{\pi}) \leftarrow \tilde{P}\right] \leq \epsilon.$

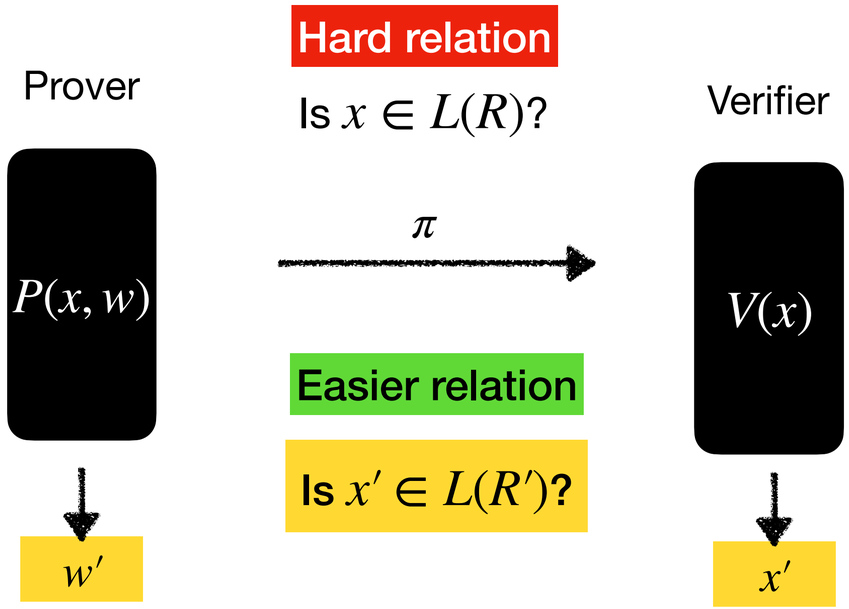
Succinctness: $|\pi| \ll |w|$.

Knowledge soundness: $\exists\,\mathscr{E}$, $\forall$ efficient adversary $\tilde{P}$, $\Pr\left[(x,w) \notin R \wedge 1 \leftarrow V(x,\tilde{\pi}) \;\middle|\; (x,\tilde{\pi}) \leftarrow \tilde{P}, w \leftarrow \mathscr{E}(x,\tilde{\pi})\right] \leq \epsilon.$

Succinct non-interactive reductions (SNRDXs)



Then G is checked via other protocols



Completeness: $\forall\,(x,w) \in R, \Pr\left[(x',w') \in R' \;\middle|\; (\pi,w') \leftarrow P(x,w), x' \leftarrow V(x,\pi)\right] = 1.$

Soundness: For every efficient adversary $\tilde{P}$, $\Pr\left[(x',w') \in R' \wedge x \notin L(R) \;\middle|\; (x,\tilde{\pi},w') \leftarrow \tilde{P}, x' \leftarrow V(x,\tilde{\pi})\right] \leq \epsilon.$

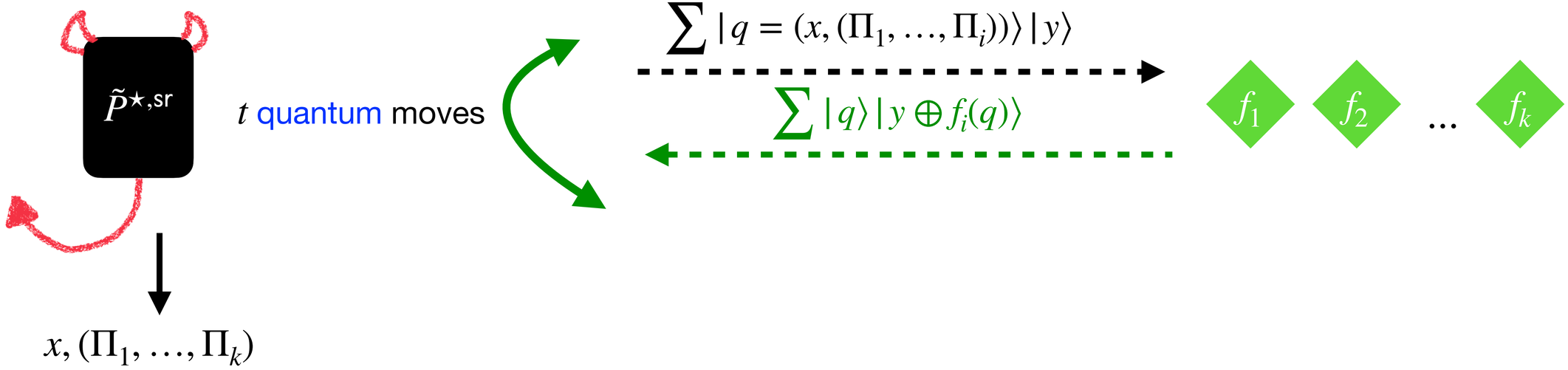
Succinctness: $|\pi| \ll |w|$.

Knowledge soundness: $\exists\,\mathscr{E}$, $\forall$ efficient adversary $\tilde{P}$, $\Pr\left[(x',w') \in R' \wedge (x,w) \notin R \;\middle|\; \begin{array}{c}(x,\tilde{\pi},w') \leftarrow \tilde{P},\\ x' \leftarrow V(x,\tilde{\pi}),\\ w \leftarrow \mathscr{E}(x,\tilde{\pi},w',x')\end{array}\right] \leq \epsilon.$

Our PQ state-restoration captures the PQ FS error

$\epsilon_{\text{IOR}}^{\star,\text{sr}}$ = the PQ soundness error of FS[IOR]

Quantum adversary



$x, (\Pi_1, \ldots, \Pi_k)$

Soundness:

$\forall$ t-move quantum adversary $\tilde{P}^{\star,\text{sr}}$,

$$\Pr\left[x \notin L \wedge x' \in L' \;\middle|\; \begin{array}{c} \forall i, f_i \leftarrow (\{0,1\}^* \to \{0,1\}^\sigma) \\ (x, \Pi_1, \ldots, \Pi_k, \rho_1, \ldots, \rho_k) \leftarrow \langle \tilde{P}^{\star,\text{sr}}, \text{Game}^{(f_i)_{i\in[k]}} \rangle \\ x' \leftarrow V_{\text{IOR}}^{(\Pi_i)_{i\in[k]}}(x; \rho_1, \ldots, \rho_k) \end{array} \right] \leq \epsilon_{\text{IOR}}^{\star,\text{sr}}(t).$$