Public-Coin, Complexity-Preserving, Succinct Arguments for NP from Collision Resistance

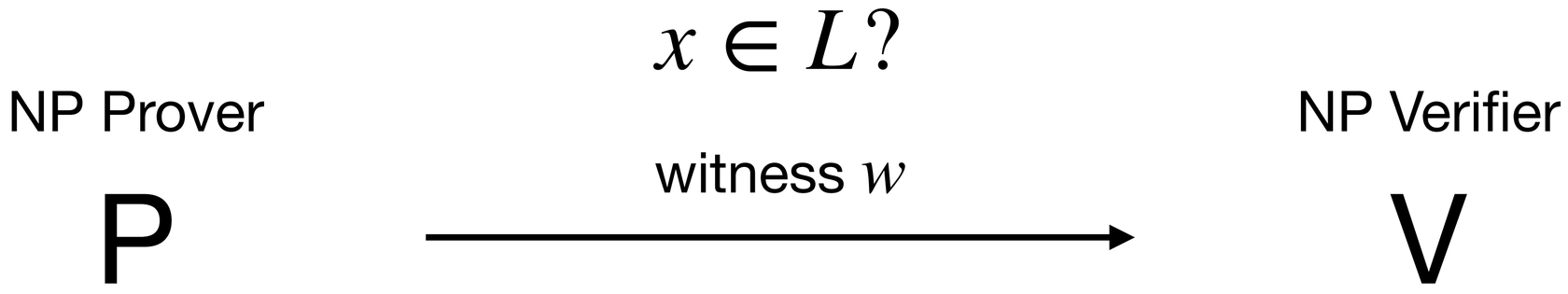
Cody Freitag[1], Omer Paneth[2], Rafael Pass[2,3]

[1]Northeastern University
[2]Tel Aviv University
[3]Cornell Tech

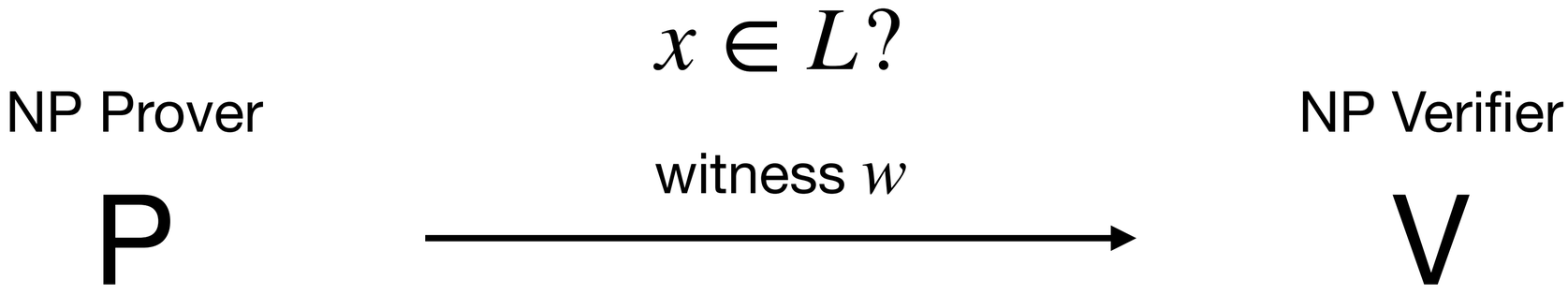
Cryptographic Proofs for NP

Cryptographic Proofs for NP

Cryptographic Proofs for NP



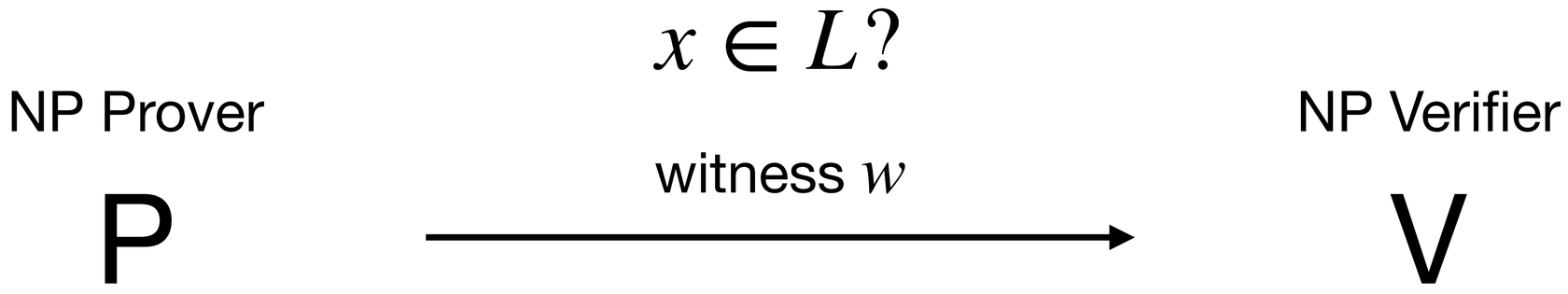
Cryptographic Proofs for NP



Completeness: If $x \in L$, $\exists w$ s.t. $V(x, w) = 1$.

Soundness: If $x \notin L$, $\forall w\ V(x, w) = 0$.

Cryptographic Proofs for NP

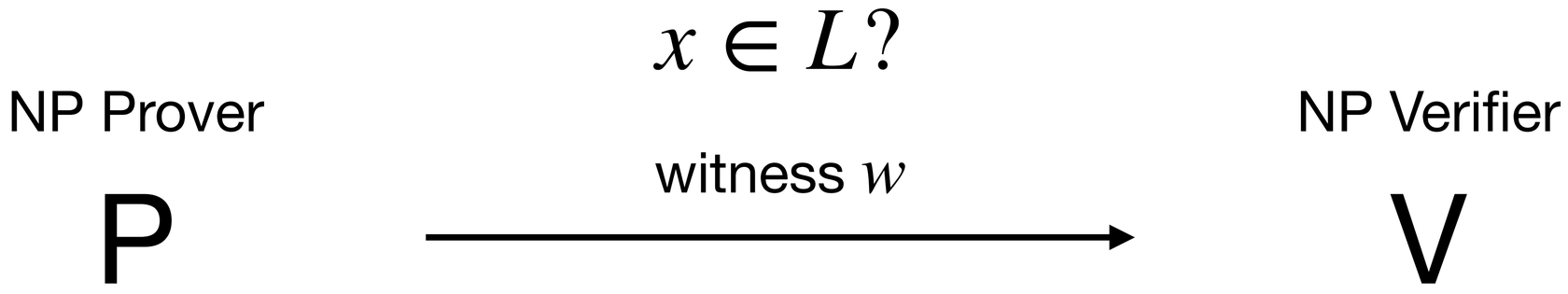


Completeness: If $x \in L$, $\exists w$ s.t. $V(x, w) = 1$.

Soundness: If $x \notin L$, $\forall w\ V(x, w) = 0$.

Equivalent to non-deterministic computation

Cryptographic Proofs for NP



Completeness: If $x \in L$, $\exists w$ s.t. $V(x, w) = 1$.

Soundness: If $x \notin L$, $\forall w\ V(x, w) = 0$.

Equivalent to non-deterministic computation

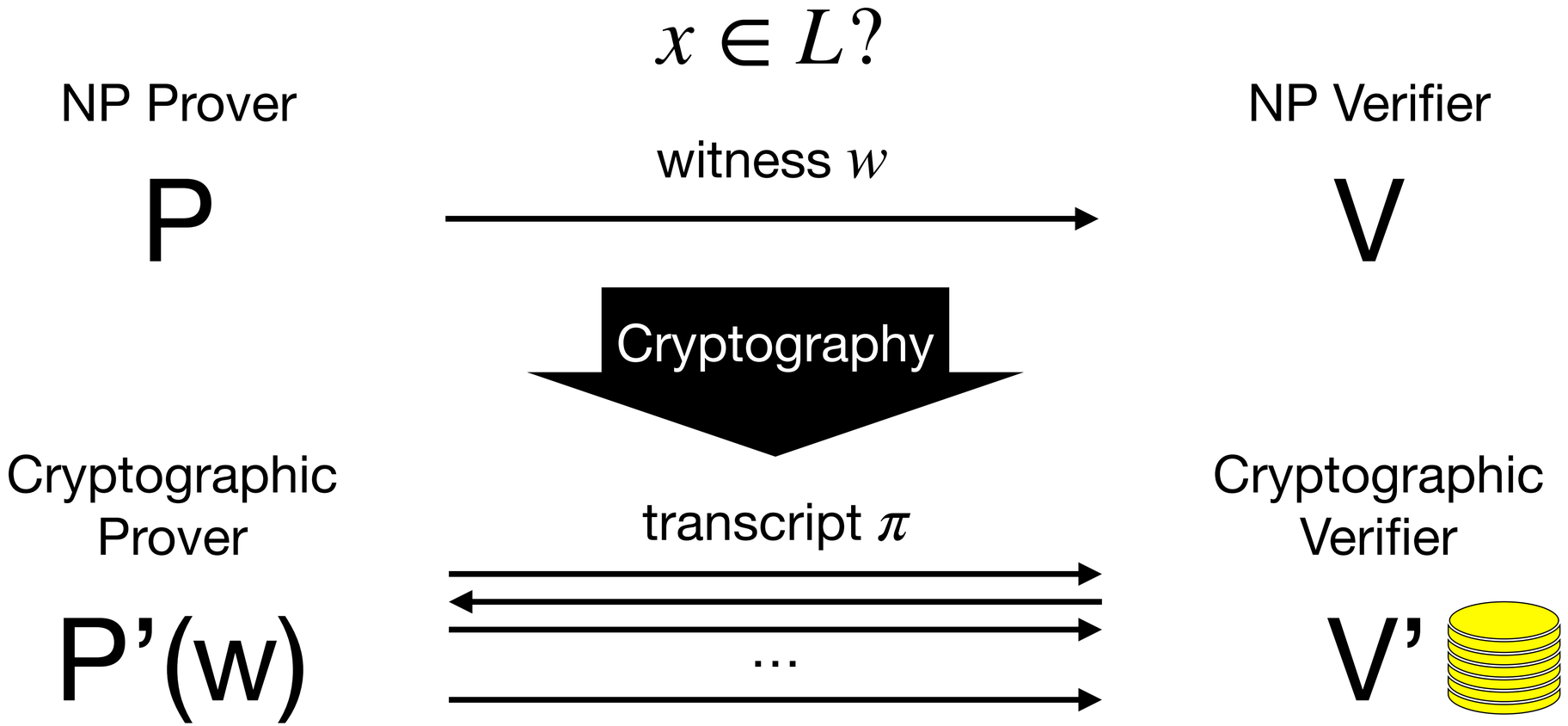
Assumptions:

- $V(x, w)$ runs in **time** T and **space** S as a **RAM program**

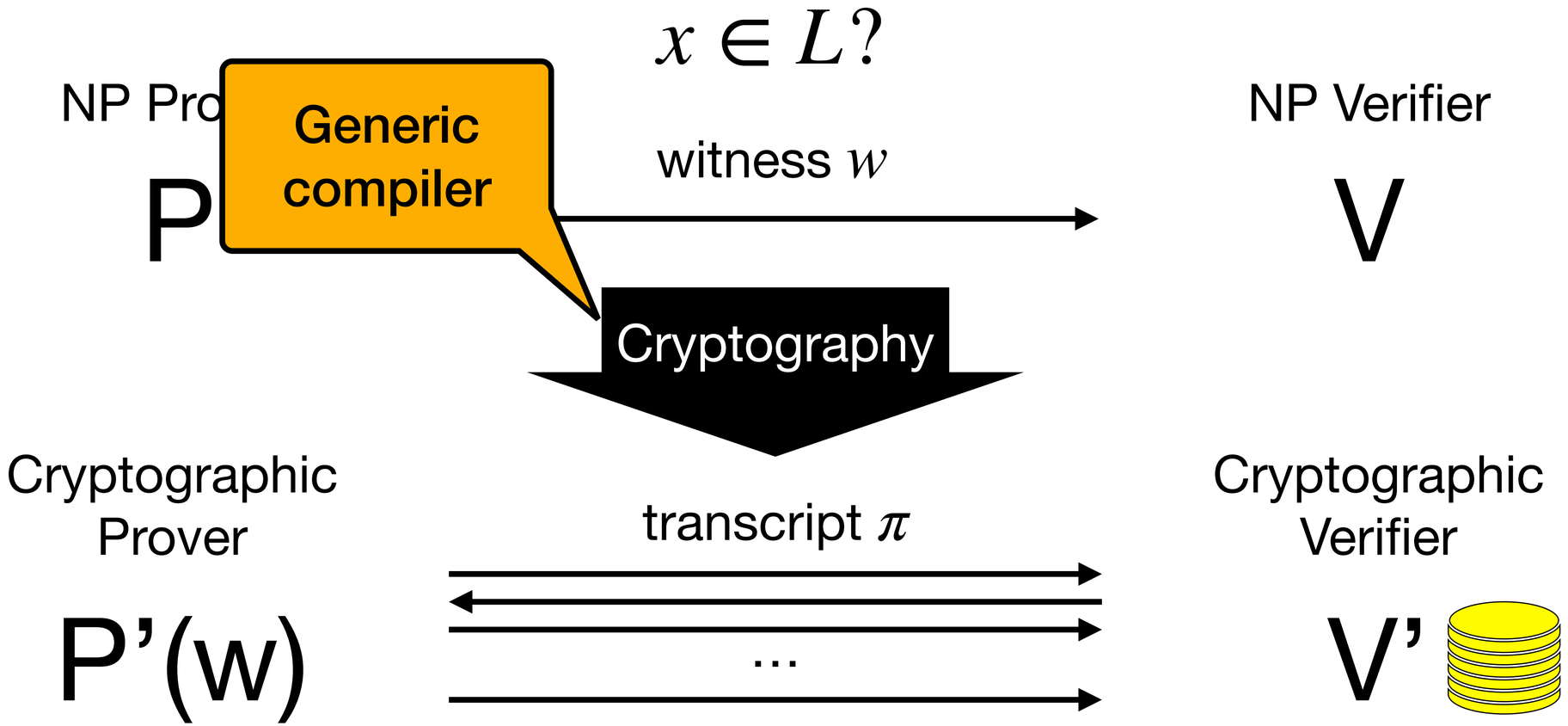
Cryptographic Proofs for NP

$x \in L?$

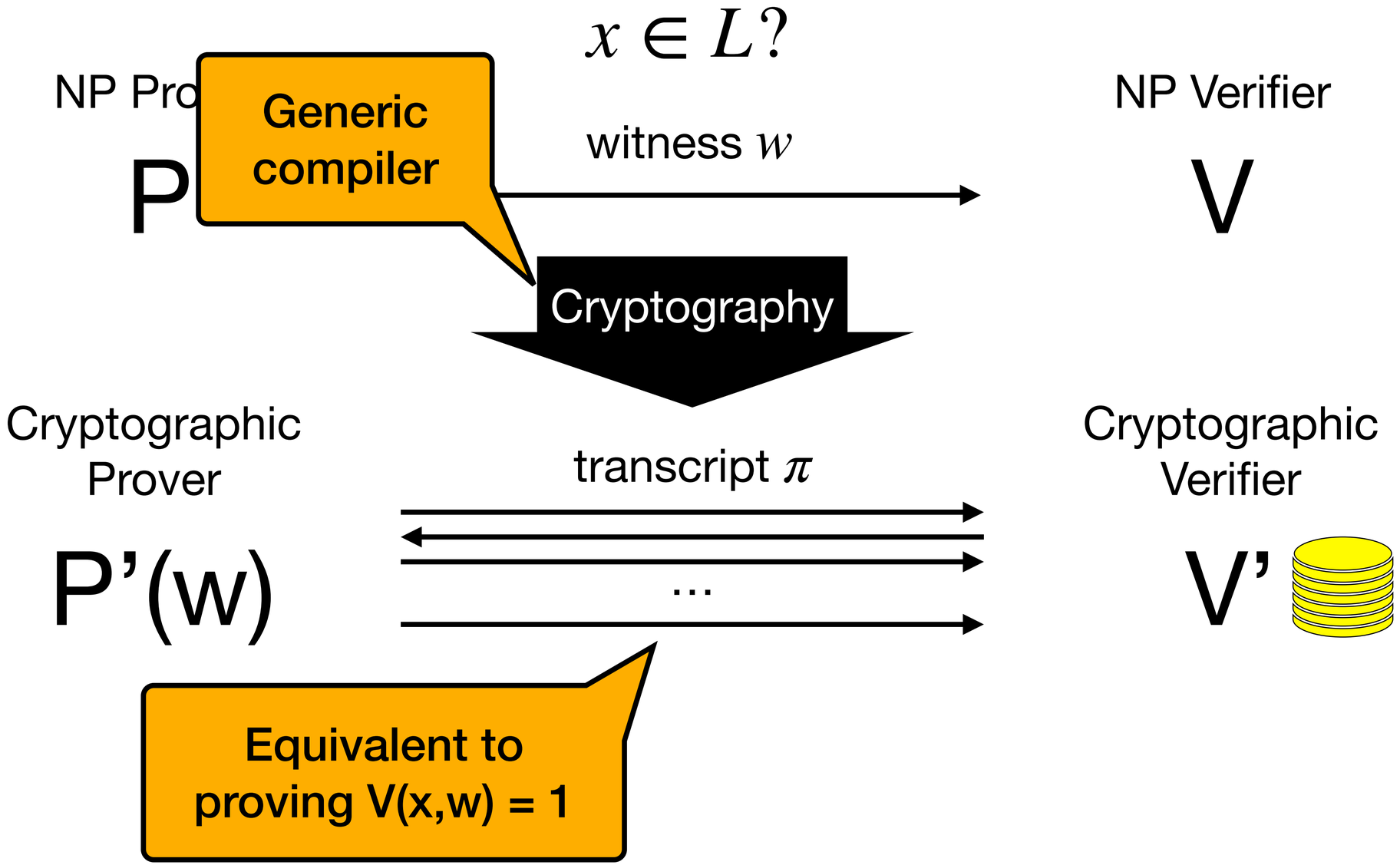
Cryptographic Proofs for NP



Cryptographic Proofs for NP



Cryptographic Proofs for NP



Cryptographic Proofs for NP

Cryptographic Prover

P’(w)

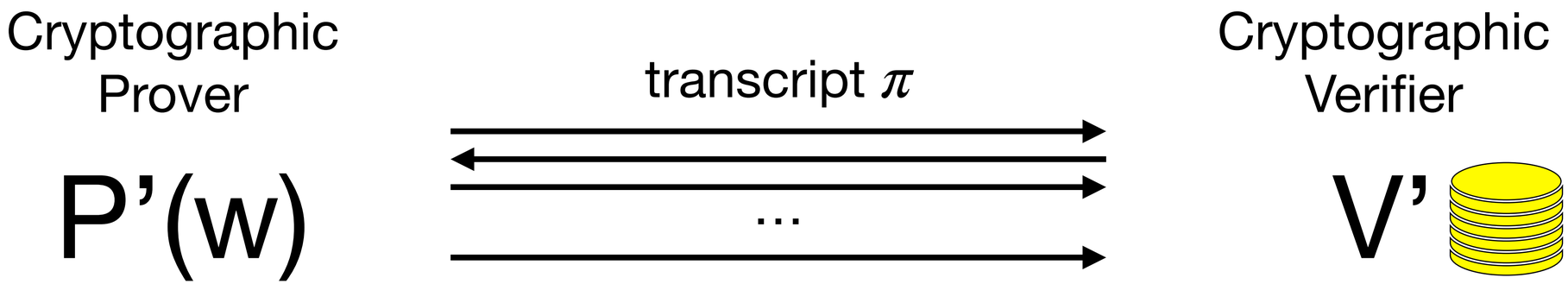
transcript π

…

Cryptographic Verifier

V’

Cryptographic Proofs for NP



Computational Soundness (Argument):

If $x \notin L$, no **poly-time** prover P* can convince V' w.h.p.

Argument of Knowledge:

If poly-time P* convinces V', it **"knows"** a witness.

Public-coin (optional):

V' sends random coins and has **no private state**.

Cryptographic Proofs for NP

Cryptographic Prover

P’(w)

transcript π

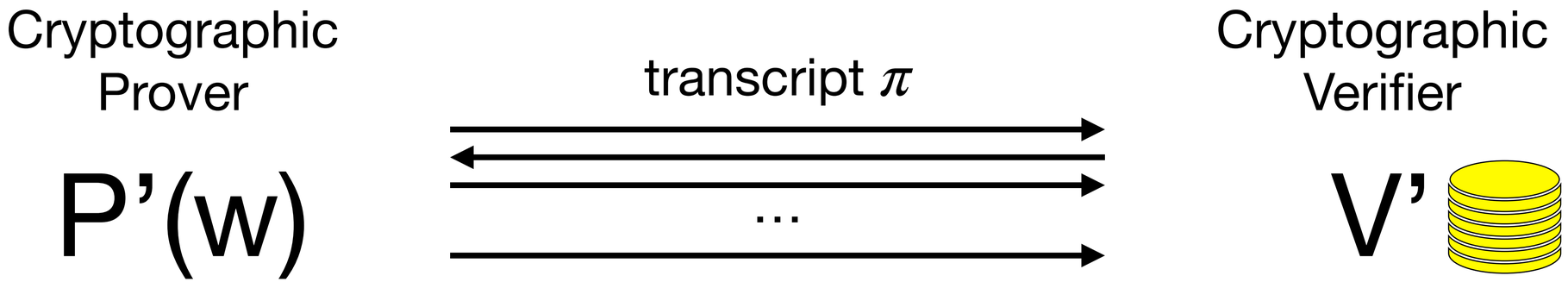
…

Cryptographic Verifier

V’

Goals: **Privacy** **Efficiency**

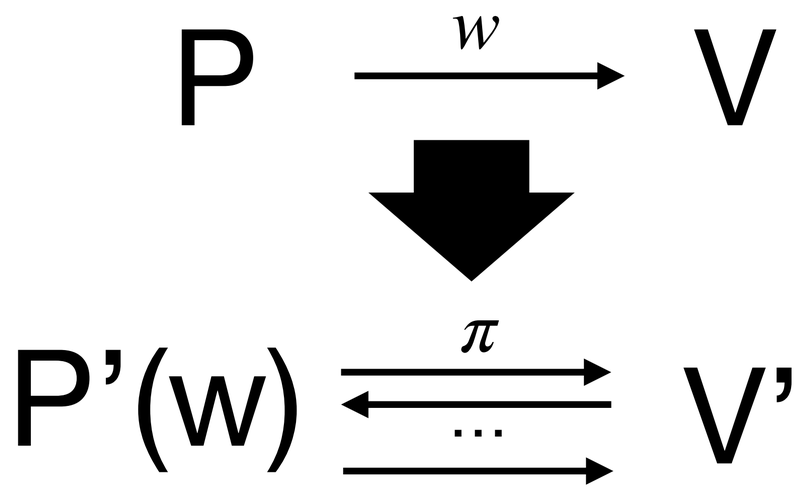
Cryptographic Proofs for NP



Goals: **Privacy** **Efficiency**

Focus of this talk

Efficiency Goals

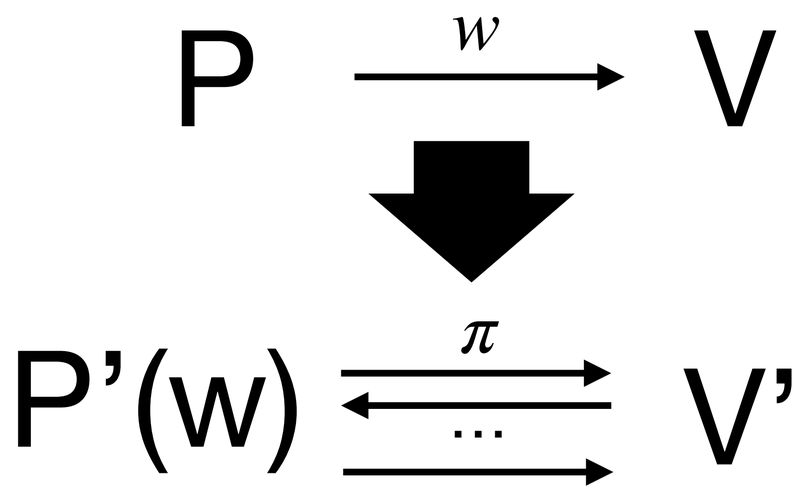


1. Low bandwidth

2. Efficient verifier

3. Scalable prover

Efficiency Goals



1. Low bandwidth

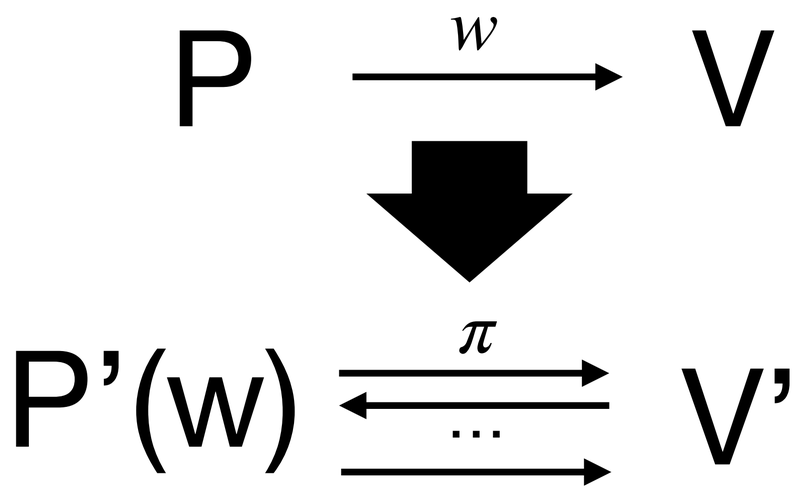
2. Efficient verifier

3. Scalable prover

Succinct Argument:

- $|\pi| = \text{polylog}(T)$
- $\text{Time}(V') = \text{poly}(|x|, \log T)$

Efficiency Goals

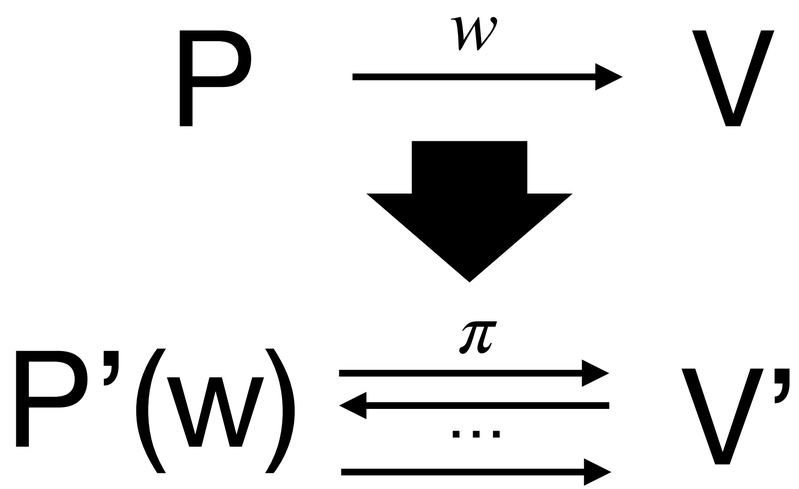


1. Low bandwidth

2. Efficient verifier

3. Scalable prover

Efficiency Goals



1. Low bandwidth

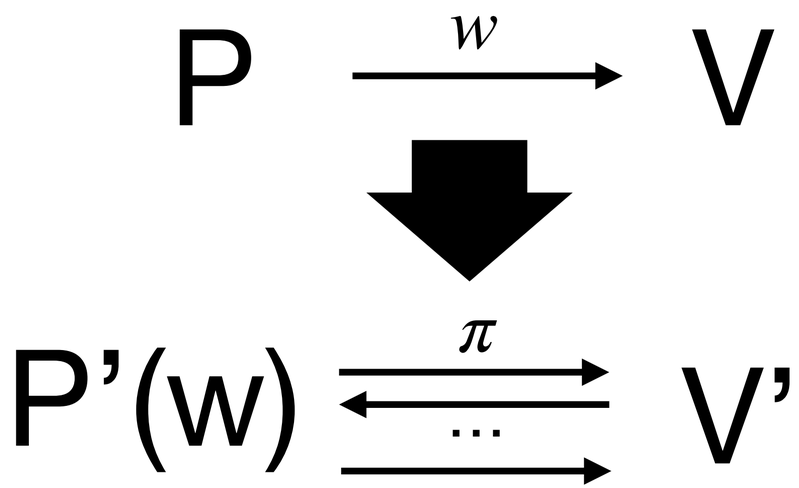
2. Efficient verifier

3. Scalable prover

Theorem [Kilian '92]:

There exists a 4-message **succinct argument** for NP from **collision-resistant hash functions**

Efficiency Goals



1. Low bandwidth

2. Efficient verifier

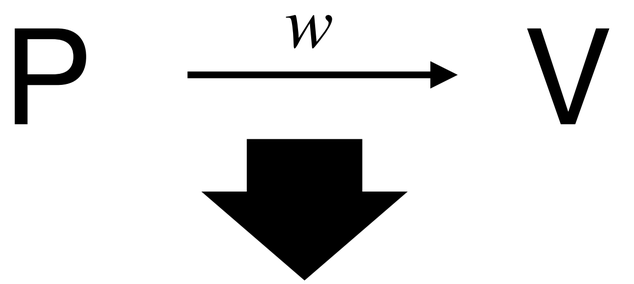
3. Scalable prover

Theorem [Kilian '92]:

There exists a 4-message **succinct argument** for NP from **collision-resistant hash functions**

Minicrypt, unstructured

Efficiency Goals



1. Low bandwidth

2. Efficient verifier

3. Scalable prover

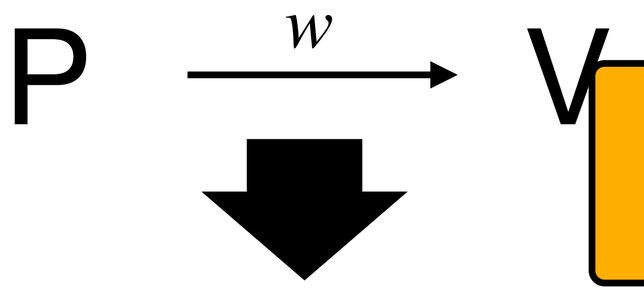
Probabilistically Checkable Proofs (PCPs) + Hash Trees

Theorem [Kilian '92]:

There exists a 4-message **succinct argument** for NP from **collision-resistant hash functions**

Minicrypt, unstructured

Efficiency Goals



V'

width

2. Efficient verifier

Poly-time in T is not scalable!

3. Scalable prover

Probabilistically Checkable Proofs (PCPs) + Hash Trees

Minicrypt, unstructured

Theorem [Kilian '92]:

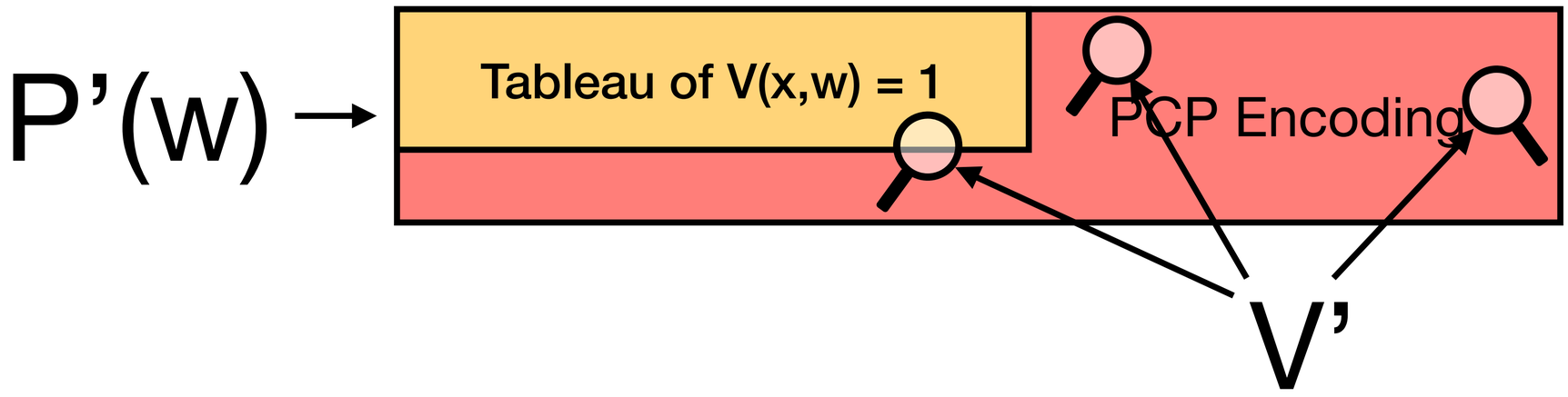
There exists a 4-message **succinct argument** for NP from **collision-resistant hash functions**

The Quest for a Scalable Prover

The Quest for a Scalable Prover

More Efficient PCPs

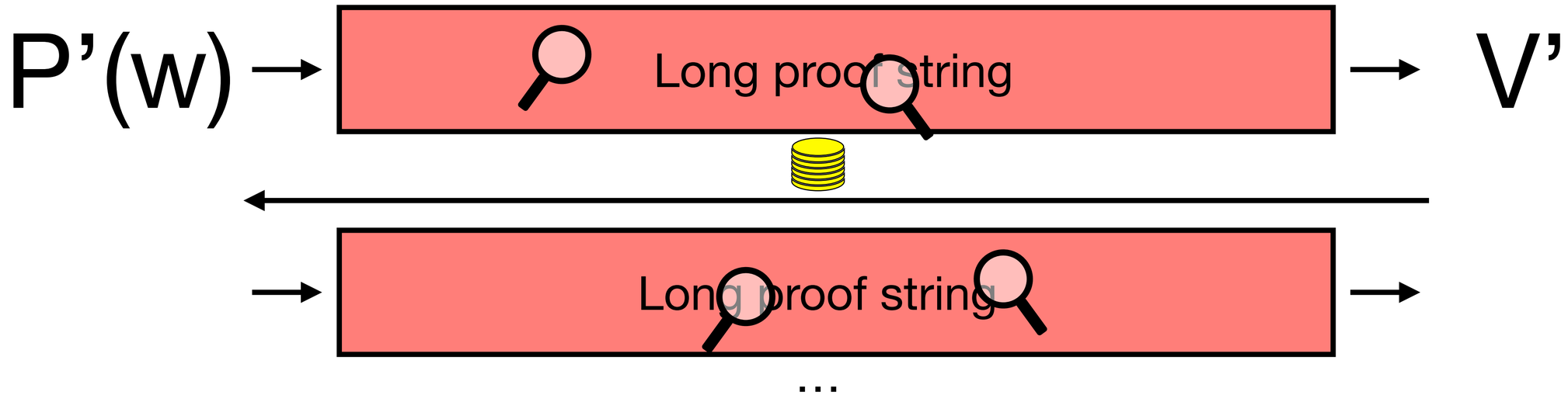
[BFLS91],[BGHSV05], [D07],[BS08],[M09], [BCGT13], [BKKMS16],…



The Quest for a Scalable Prover

More Efficient PCPs
[BFLS91],[BGHSV05], [D07],[BS08],[M09], [BCGT13], [BKKMS16],…

Interactive Oracle Proofs (IOPs)
[BCS16],[RRR16], [BCGRS17],[RR20],…

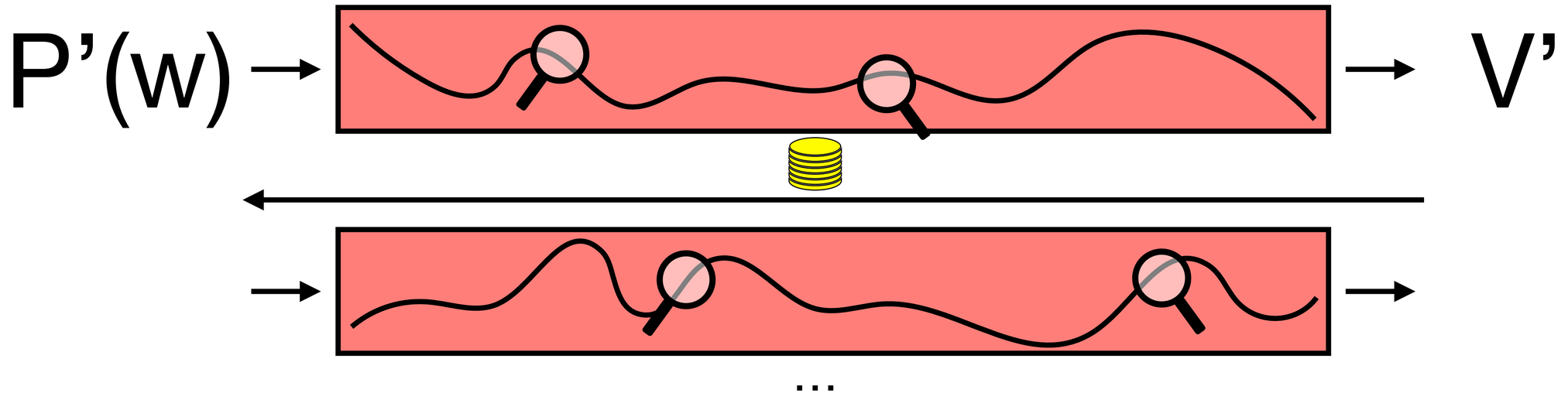


The Quest for a Scalable Prover

More Efficient PCPs
[BFLS91],[BGHSV05], [D07],[BS08],[M09], [BCGT13], [BKKMS16],…

Interactive Oracle Proofs (IOPs)
[BCS16],[RRR16], [BCGRS17],[RR20],…

Structured Oracle Proofs
[IKO07],[KZG10],[BCIOP13],[GGPR13], [BCCGP16],[BBBPWM18],[BBHR18], [BBCGI19],[BBHR19],[MBKM19], [GWC19],[S20],[CHMMVW20],[BFS20], [BHRRS20],[BHRRS21],[SZ22],…



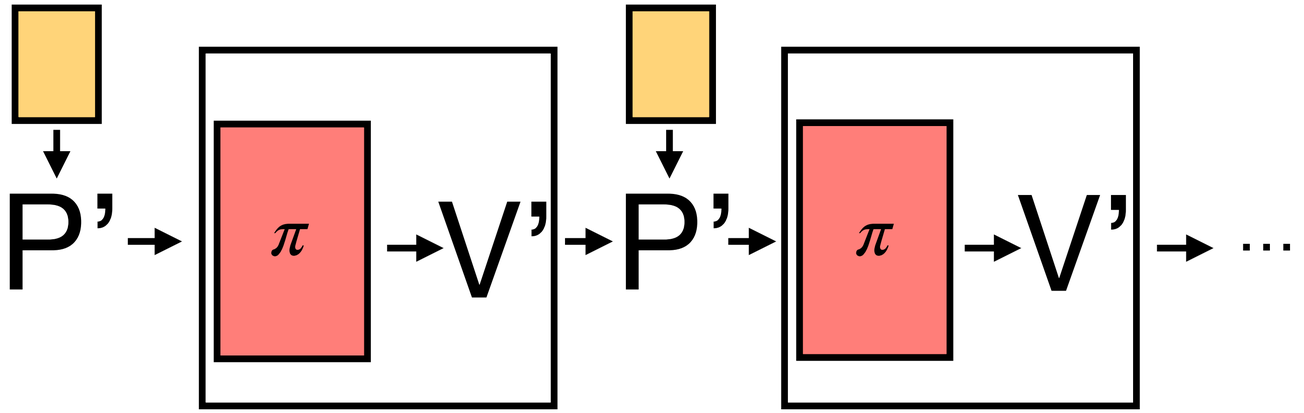
The Quest for a Scalable Prover

More Efficient PCPs
[BFLS91],[BGHSV05], [D07],[BS08],[M09], [BCGT13], [BKKMS16],…

Interactive Oracle Proofs (IOPs)
[BCS16],[RRR16], [BCGRS17],[RR20],…

Structured Oracle Proofs
[IKO07],[KZG10],[BCIOP13],[GGPR13], [BCCGP16],[BBBPWM18],[BBHR18], [BBCGI19],[BBHR19],[MBKM19], [GWC19],[S20],[CHMMVW20],[BFS20], [BHRRS20],[BHRRS21],[SZ22],…

Recursive Composition
[V08],[BCCT12],…



The Quest for a Scalable Prover

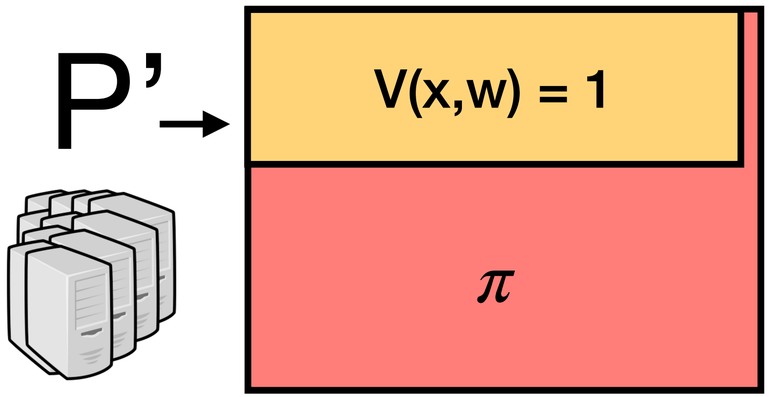
More Efficient PCPs
[BFLS91],[BGHSV05], [D07],[BS08],[M09], [BCGT13], [BKKMS16],…

Interactive Oracle Proofs (IOPs)
[BCS16],[RRR16], [BCGRS17],[RR20],…

Structured Oracle Proofs
[IKO07],[KZG10],[BCIOP13],[GGPR13], [BCCGP16],[BBBPWM18],[BBHR18], [BBCGI19],[BBHR19],[MBKM19], [GWC19],[S20],[CHMMVW20],[BFS20], [BHRRS20],[BHRRS21],[SZ22],…

Recursive Composition
[V08],[BCCT12],…

Parallelizable Provers
[EFKP20],[FPS22]



The Quest for a Scalable Prover

More Efficient PCPs
[BFLS91],[BGHSV05], [D07],[BS08],[M09], [BCGT13], [BKKMS16],…

Interactive Oracle Proofs (IOPs)
[BCS16],[RRR16], [BCGRS17],[RR20],…

Structured Oracle Proofs
[IKO07],[KZG10],[BCIOP13],[GGPR13], [BCCGP16],[BBBPWM18],[BBHR18], [BBCGI19],[BBHR19],[MBKM19], [GWC19],[S20],[CHMMVW20],[BFS20], [BHRRS20],[BHRRS21],[SZ22],…

Recursive Composition
[V08],[BCCT12],…

Parallelizable Provers
[EFKP20],[FPS22]

The Quest for a Scalable Prover

More Efficient PCPs
[BFLS91],[BGHSV05], [D07],[BS08],[M09], [BCGT13], [BKKMS16],…

Interactive Oracle Proofs (IOPs)
[BCS16],[RRR16], [BCGRS17],[RR20],…

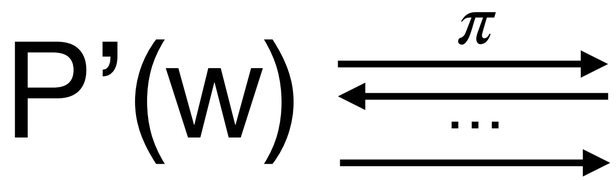
Structured Oracle Proofs
[IKO07],[KZG10],[BCIOP13],[GGPR13], [BCCGP16],[BBBPWM18],[BBHR18], [BBCGI19],[BBHR19],[MBKM19], [GWC19],[S20],[CHMMVW20],[BFS20], [BHRRS20],[BHRRS21],[SZ22],…

Recursive Composition
[V08],[BCCT12],…

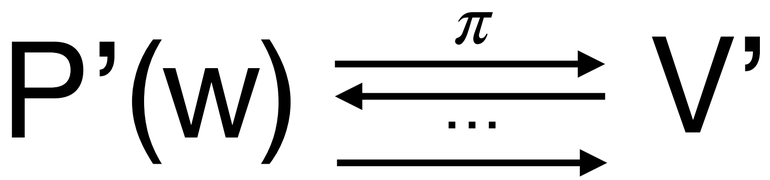
Parallelizable Provers
[EFKP20],[FPS22]

Space-Efficient/ Complexity-Preserving
[BC12],[BCCT12],[BHRRS20], [BHRRS21],[BBHV22]

Blueprint for Kilian-Based Arguments



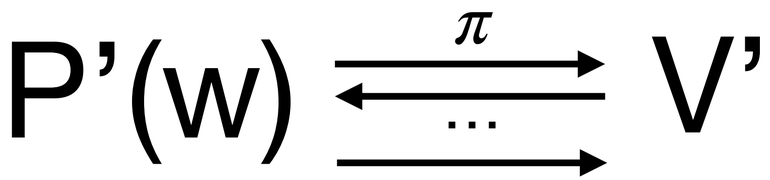
Blueprint for Kilian-Based Arguments



1. Write down computation tableau

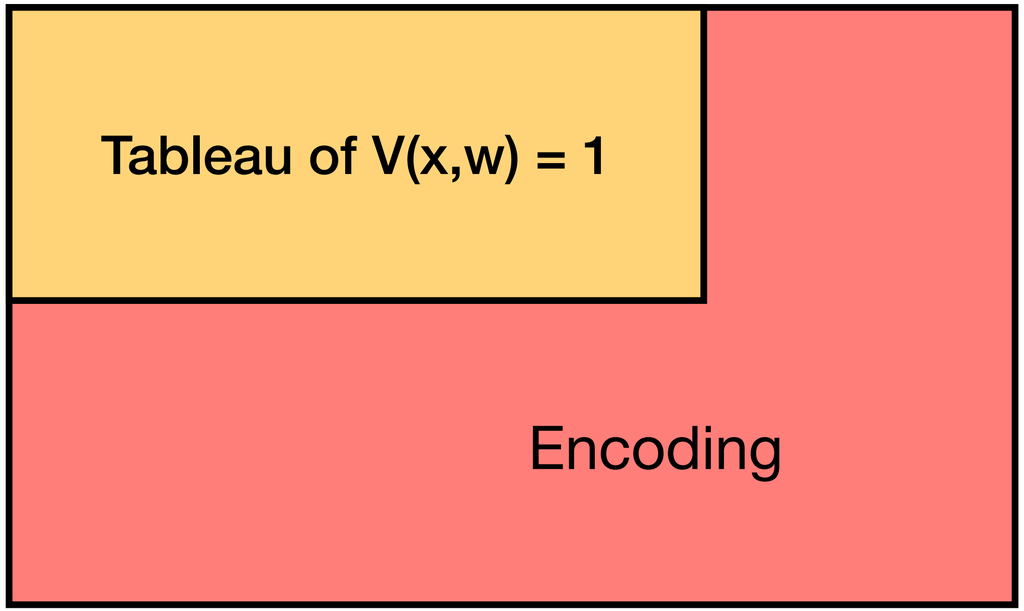
Tableau of V(x,w) = 1

Blueprint for Kilian-Based Arguments

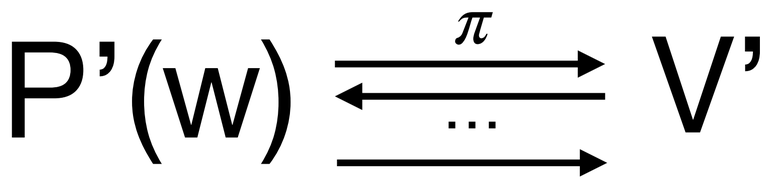


1. Write down computation tableau

2. Encode tableau



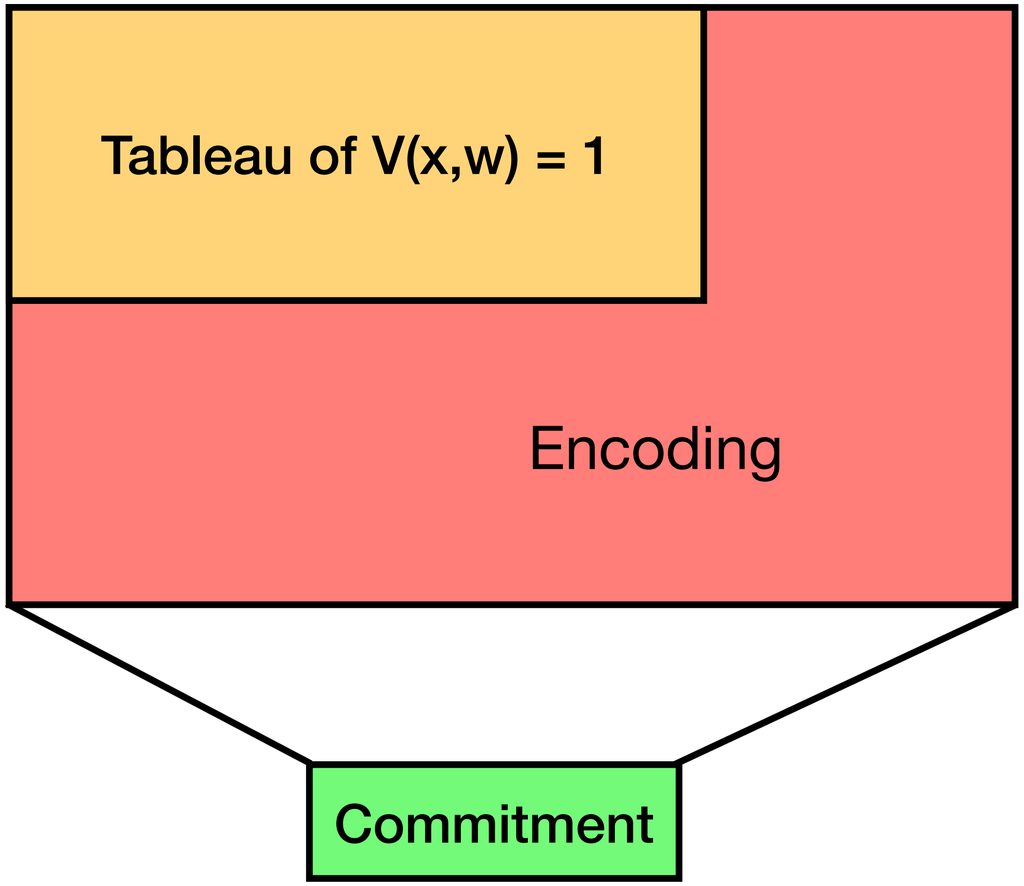
Blueprint for Kilian-Based Arguments



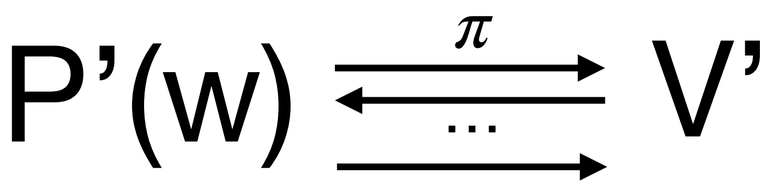
1. Write down computation tableau

2. Encode tableau

3. Cryptographically commit to encoding



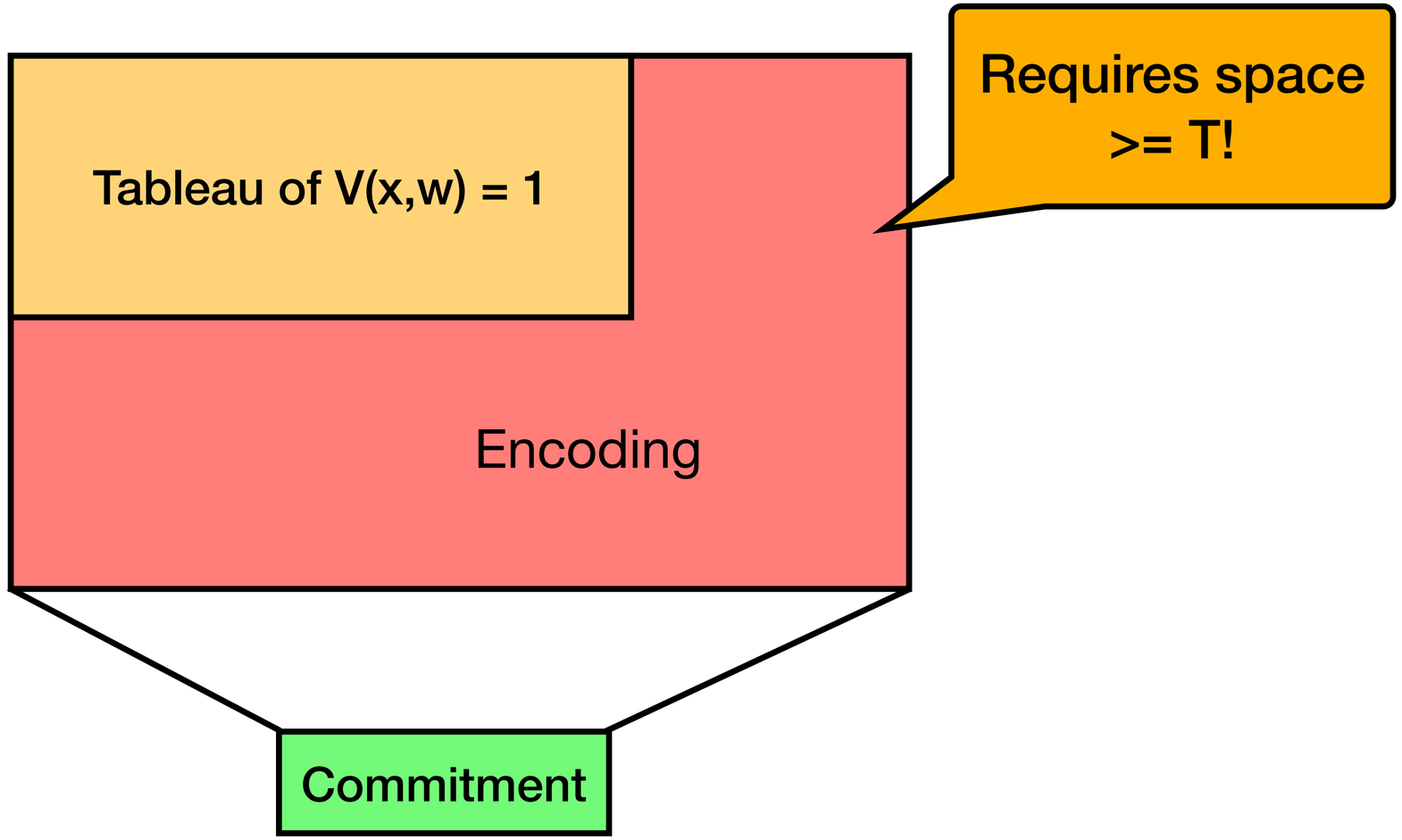
Blueprint for Kilian-Based Arguments



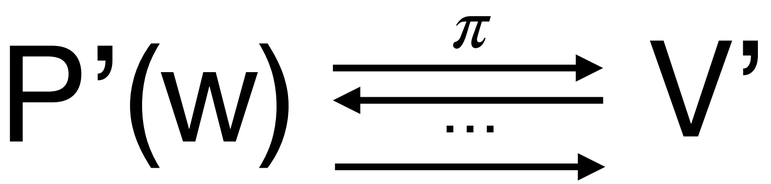
1. Write down computation tableau

2. Encode tableau

3. Cryptographically commit to encoding



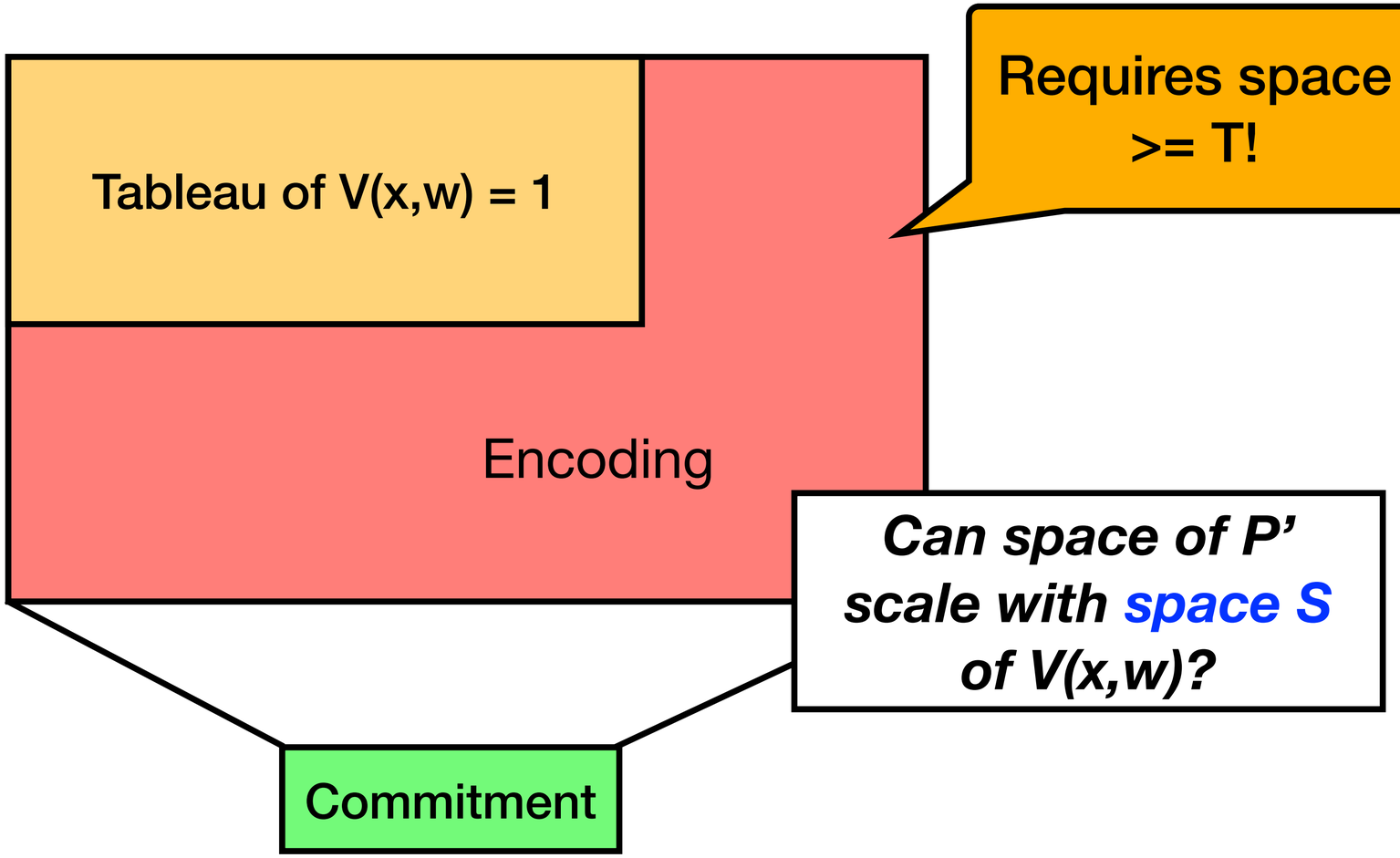
Blueprint for Kilian-Based Arguments



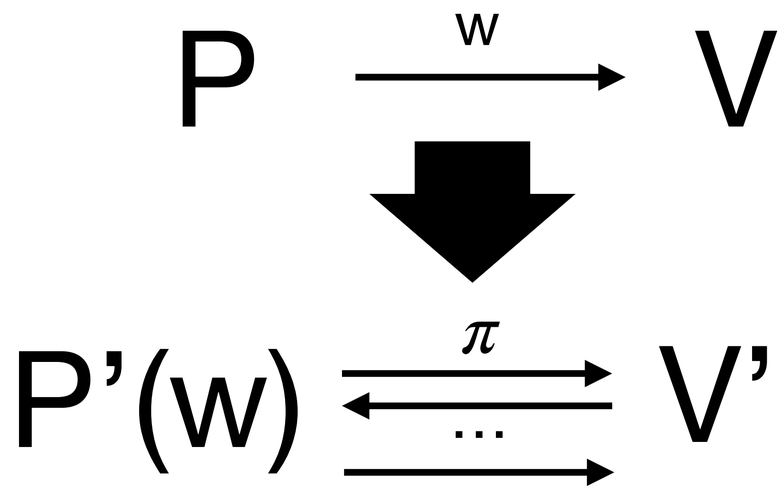
1. Write down computation tableau

2. Encode tableau

3. Cryptographically commit to encoding



Complexity-Preserving Arguments

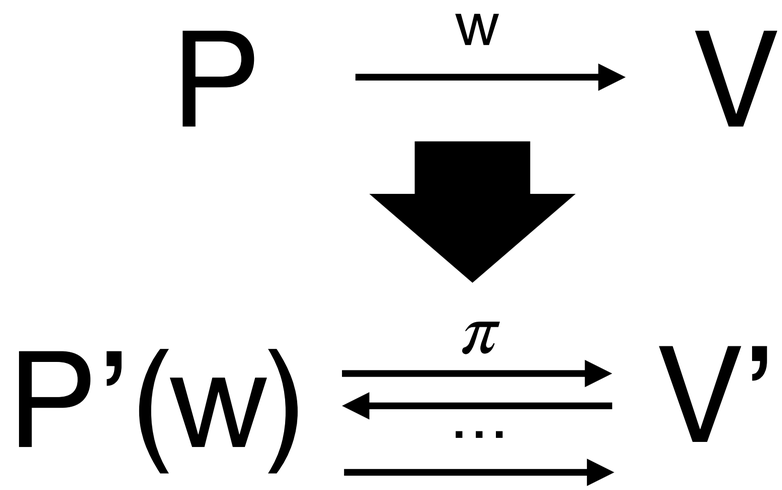


Complexity-Preserving:

- P'(w) runs in time $\tilde{O}(T)$

and space $\tilde{O}(S)$

Complexity-Preserving Ar



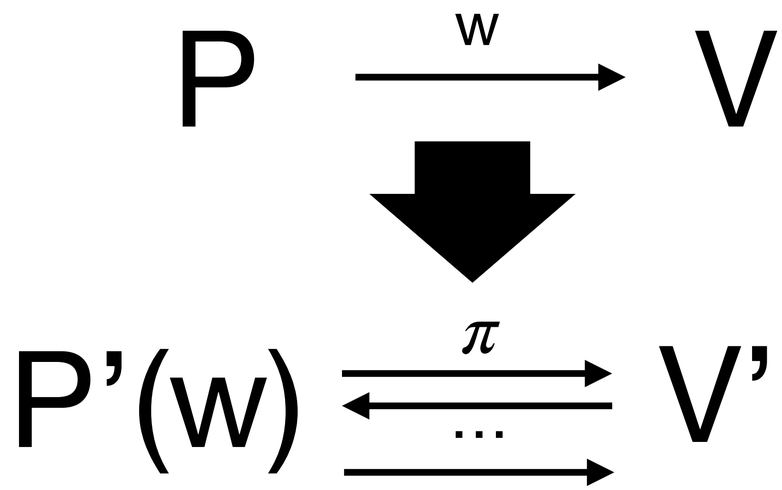
Best possible up to polylog factors!

Complexity-Preserving:

- P'(w) runs in time $\tilde{O}(T)$

and space $\tilde{O}(S)$

Complexity-Preserving Arguments

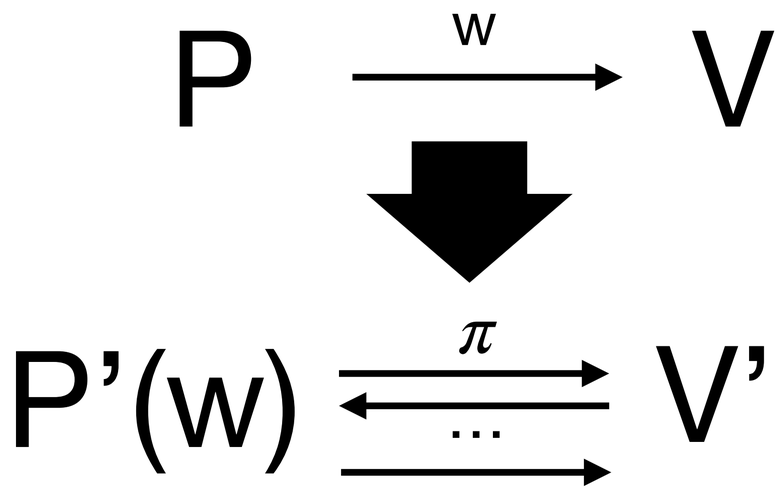


Complexity-Preserving:

- P'(w) runs in time $\tilde{O}(T)$

and space $\tilde{O}(S)$

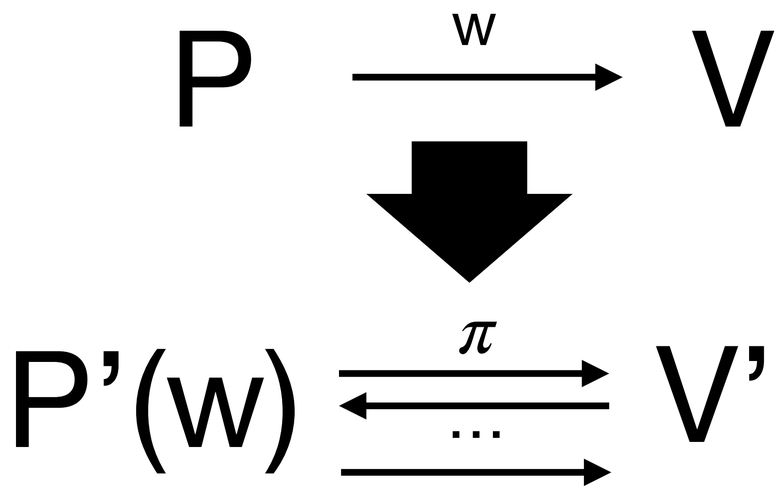
Complexity-Preserving Arguments



Complexity-Preserving:
- P'(w) runs in time $\tilde{O}(T)$
and space $\tilde{O}(S)$

Theorem [BitanskyChiesa '12]:
There exists a 4-message **complexity-preserving** succinct argument for NP from **fully homomorphic encryption**

Complexity-Preserving Arguments



Complexity-Preserving:
- P'(w) runs in time $\tilde{O}(T)$
and space $\tilde{O}(S)$

Theorem [BitanskyChiesa '12]:
There exists a 4-message **complexity-preserving** succinct argument for NP from **fully homomorphic encryption**

Open Questions:

Public verifiability?

Minicrypt?

Towards resolving these questions

Towards resolving these questions

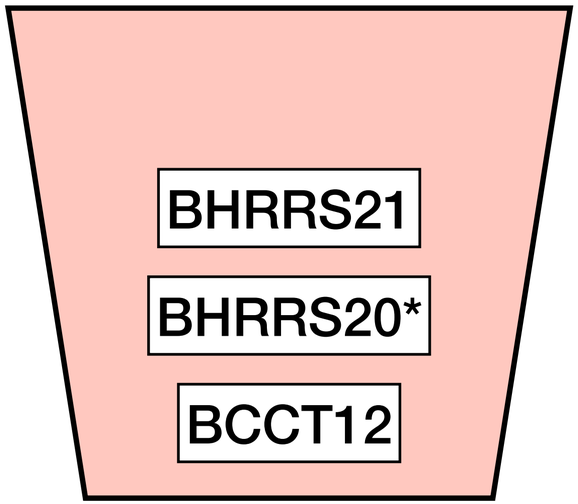
Publicly-verifiable, Complexity-Preserving, Succinct Arguments

*not fully succinct
**only privately verifiable

Towards resolving these questions

Publicly-verifiable, Complexity-Preserving, Succinct Arguments

Non-standard assumptions

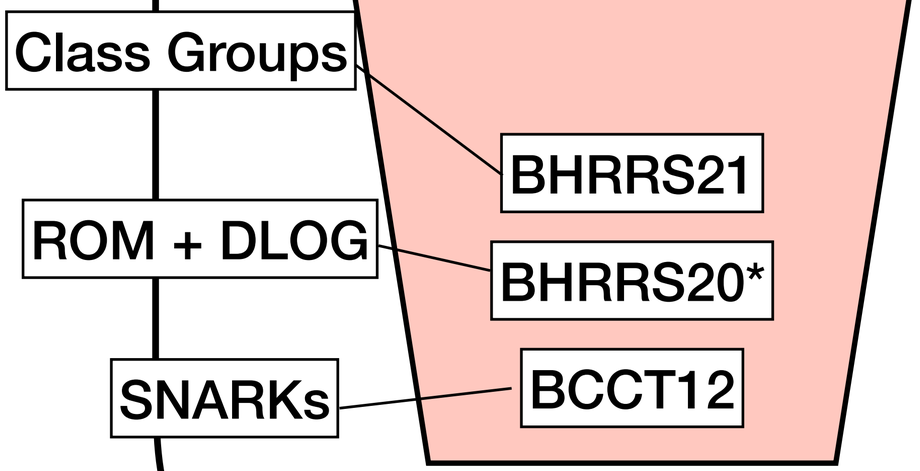


*not fully succinct
**only privately verifiable

Towards resolving these questions

Publicly-verifiable, Complexity-Preserving, Succinct Arguments

Non-standard assumptions



*not fully succinct
**only privately verifiable

Towards resolving these questions

Publicly-verifiable, Complexity-Preserving, Succinct Arguments

Non-standard assumptions

BHRRS21

BHRRS20*

BCCT12

Non-interactive for P

KLVW23

DGKV22

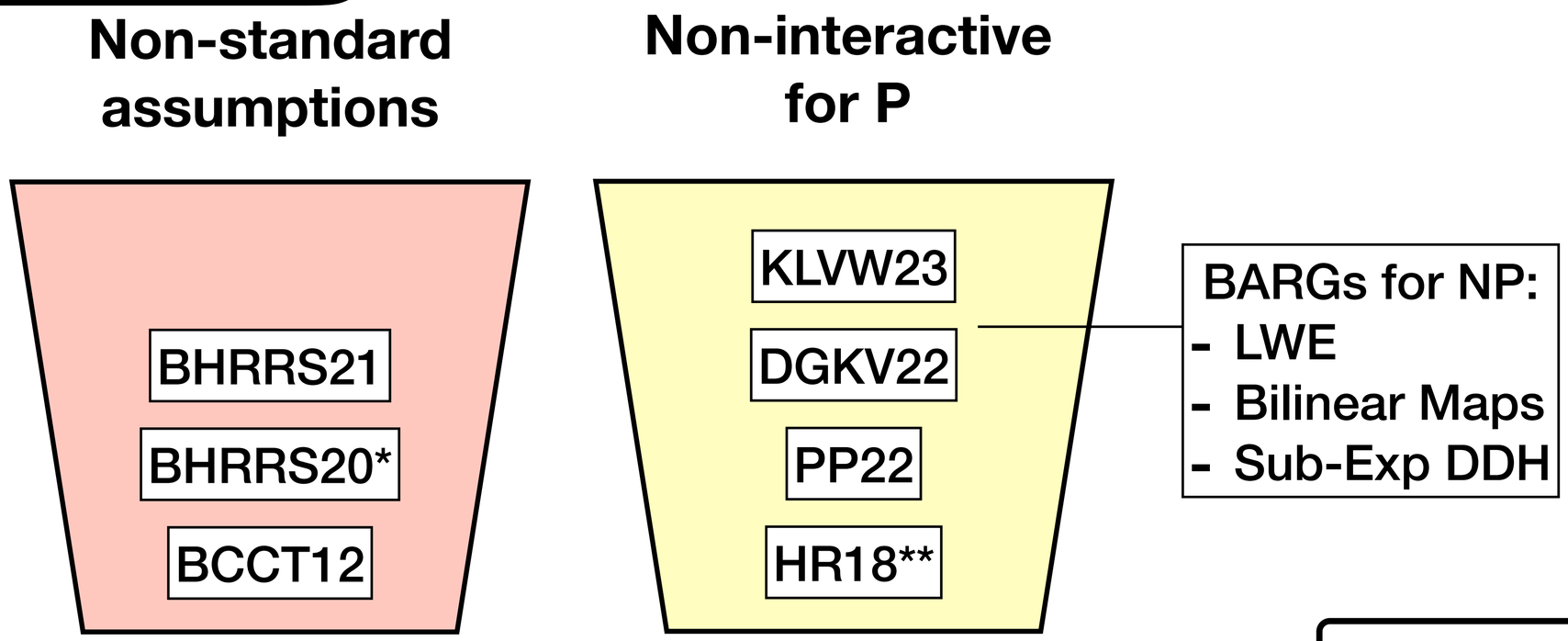
PP22

HR18**

*not fully succinct
**only privately verifiable

Towards resolving these questions

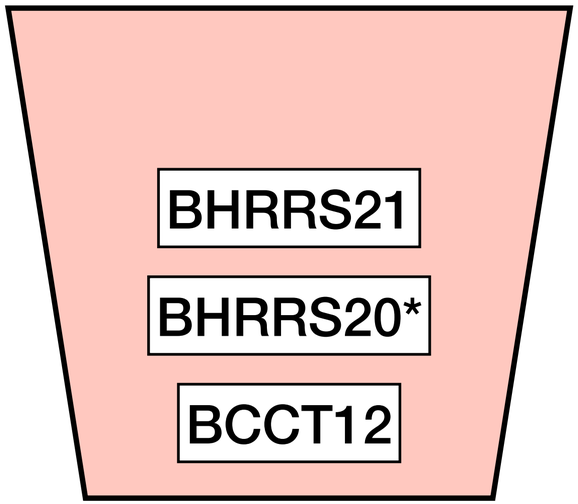
Publicly-verifiable, Complexity-Preserving, Succinct Arguments

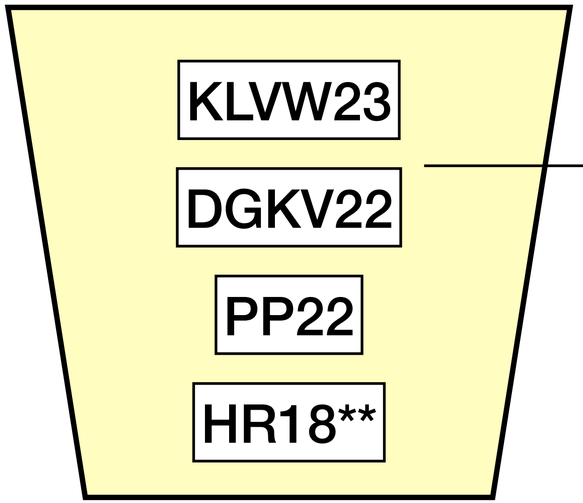


*not fully succinct
**only privately verifiable

Towards resolving these questions

Publicly-verifiable, Complexity-Preserving, Succinct Arguments





Still all Cryptomania assumptions!

BARGs for NP:
- LWE
- Bilinear Maps
- Sub-Exp DDH

*not fully succinct
**only privately verifiable

Towards resolving these questions

Publicly-verifiable, Complexity-Preserving, Succinct Arguments

Non-standard assumptions

- BHRRS21
- BHRRS20*
- BCCT12

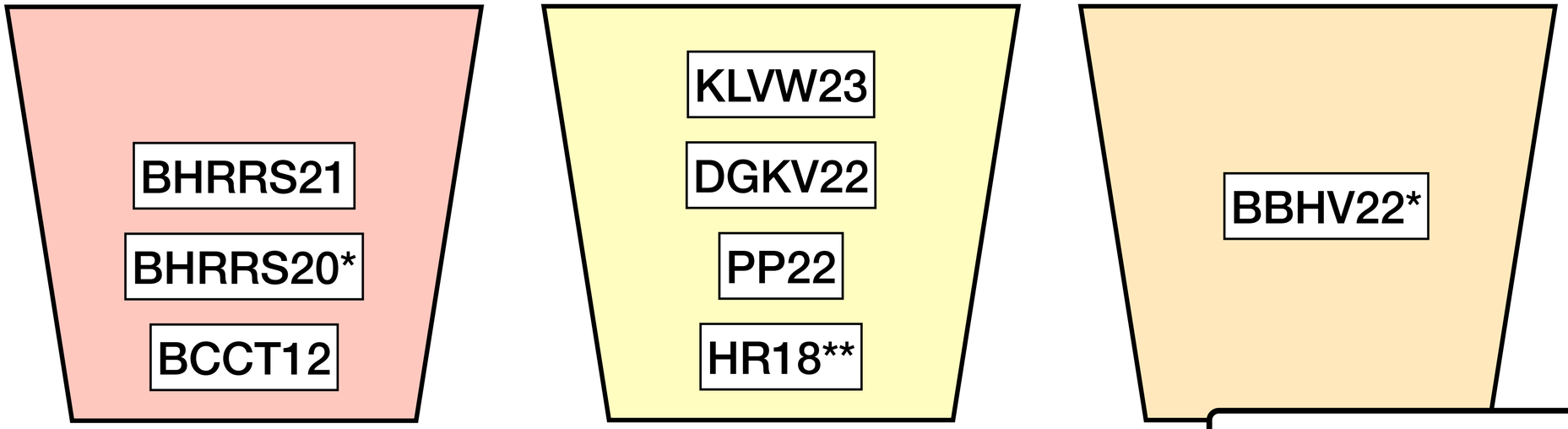
Non-interactive for P

- KLVW23
- DGKV22
- PP22
- HR18**

*not fully succinct
**only privately verifiable

Towards resolving these questions

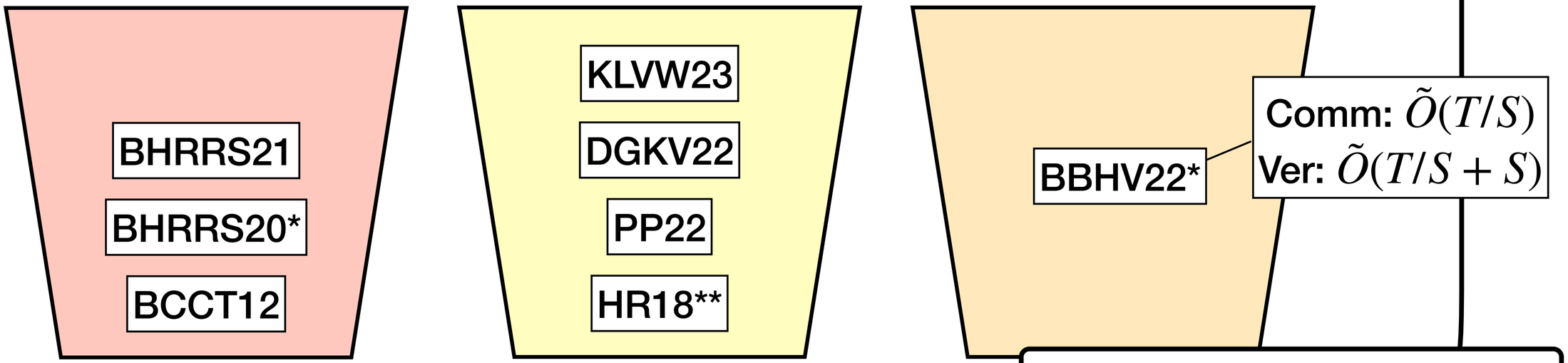
Publicly-verifiable, Complexity-Preserving, Succinct Arguments



*not fully succinct
**only privately verifiable

Towards resolving these questions

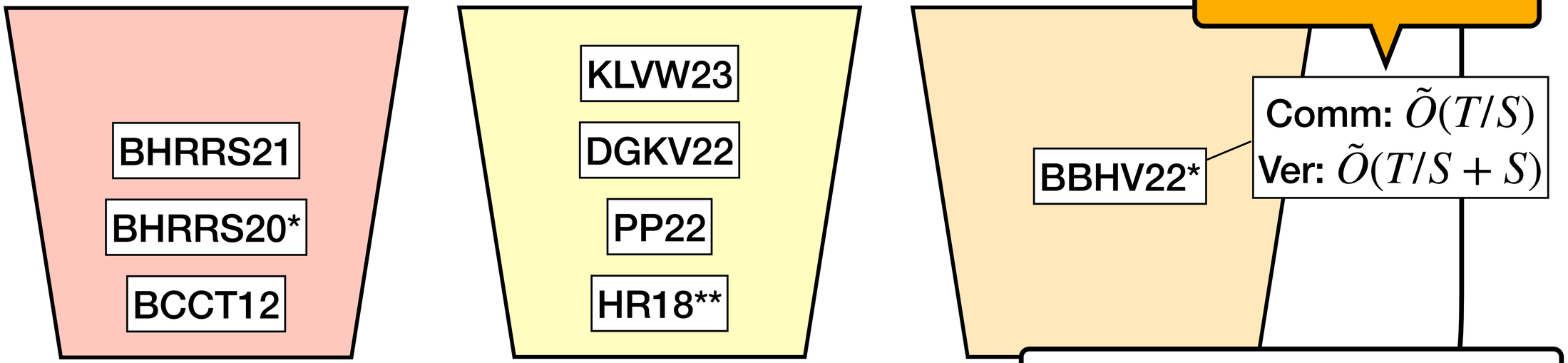
Publicly-verifiable, Complexity-Preserving, Succinct Arguments



*not fully succinct
**only privately verifiable

Towards resolving these questions

Publicly-verifiable, Complexity-Preserving, Succinct Arguments



"Tight" wrt black-box constructions

*not fully succinct
**only privately verifiable

Our Main Result

Our Main Result

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

Our Main Result

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

[BC12] Questions:

Publicly verifiable ✓

Minicrypt ✓

Our Main Result

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

[BC12] Questions:

- Publicly verifiable ✓
- Minicrypt ✓

Remaining Open Questions:

- Fixed constant rounds?
- Universal argument?
- Turing machines?

Corollaries to Our Main Result

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

Corollaries to Our Main Result

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

Zero-Knowledge

Corollaries to Our Main Result

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

Zero-Knowledge

"Commit-and-prove" via [BGGHKMR88]

Corollaries to Our Main Result

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

Zero-Knowledge

Parallelizability

"Commit-and-prove" via [BGGHKMR88]

Corollaries to Our Main Result

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

Zero-Knowledge

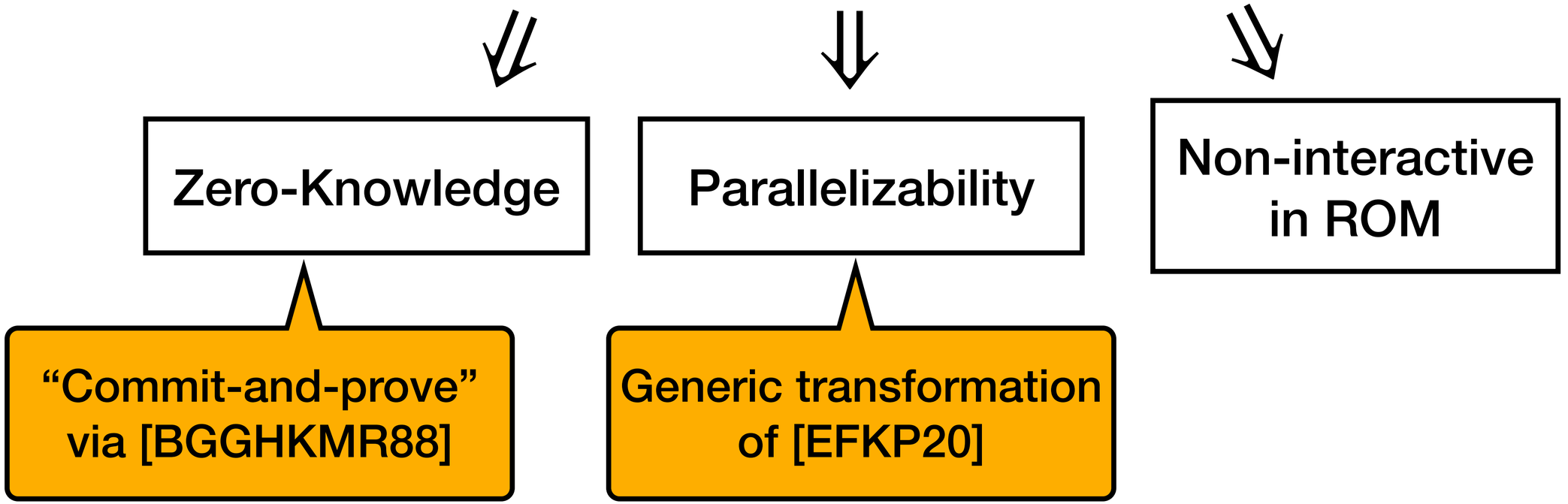
Parallelizability

"Commit-and-prove" via [BGGHKMR88]

Generic transformation of [EFKP20]

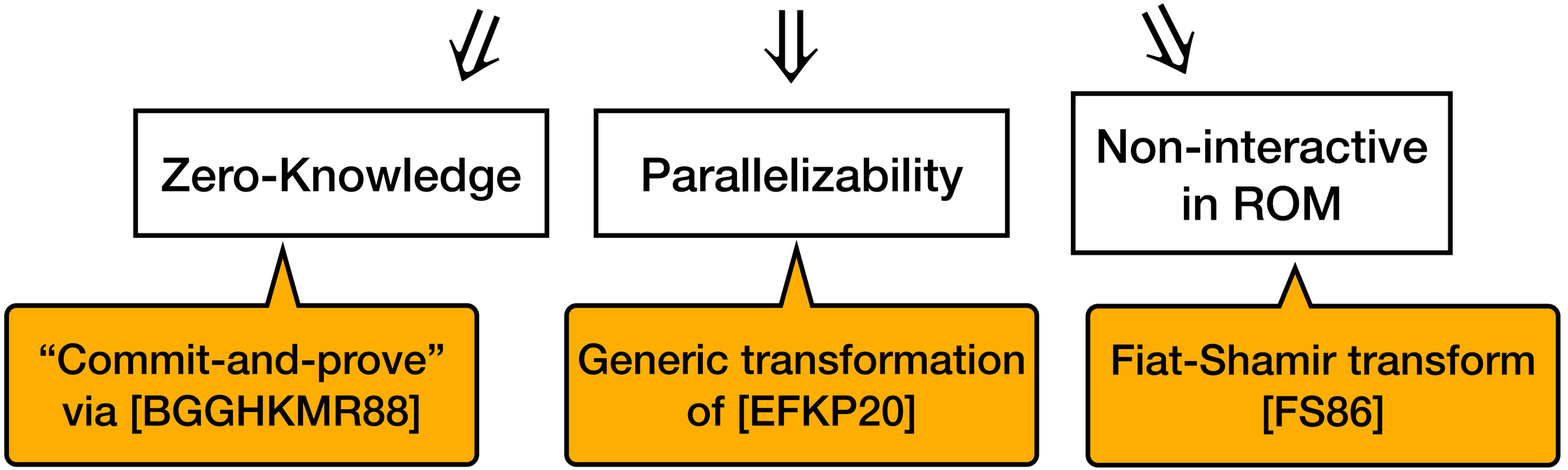
Corollaries to Our Main Result

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**



Corollaries to Our Main Result

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**



Roadmap for Our Construction

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

Roadmap for Our Construction

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

1. A $\tilde{O}(T^{1/2}S)$**-space protocol**

Roadmap for Our Construction

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

1. A $\tilde{O}(T^{1/2}S)$-space protocol

2. A solution for small space

Roadmap for Our Construction

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

1. A $\tilde{O}(T^{1/2}S)$-space protocol

2. A solution for small space

3. Extending to arbitrary space

Roadmap for Our Construction

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

Reduction via hash trees [KP16]

1. A $\tilde{O}(T^{1/2}S)$-space protocol

2. A solution for small space

3. Extending to arbitrary space

Roadmap for Our Construction

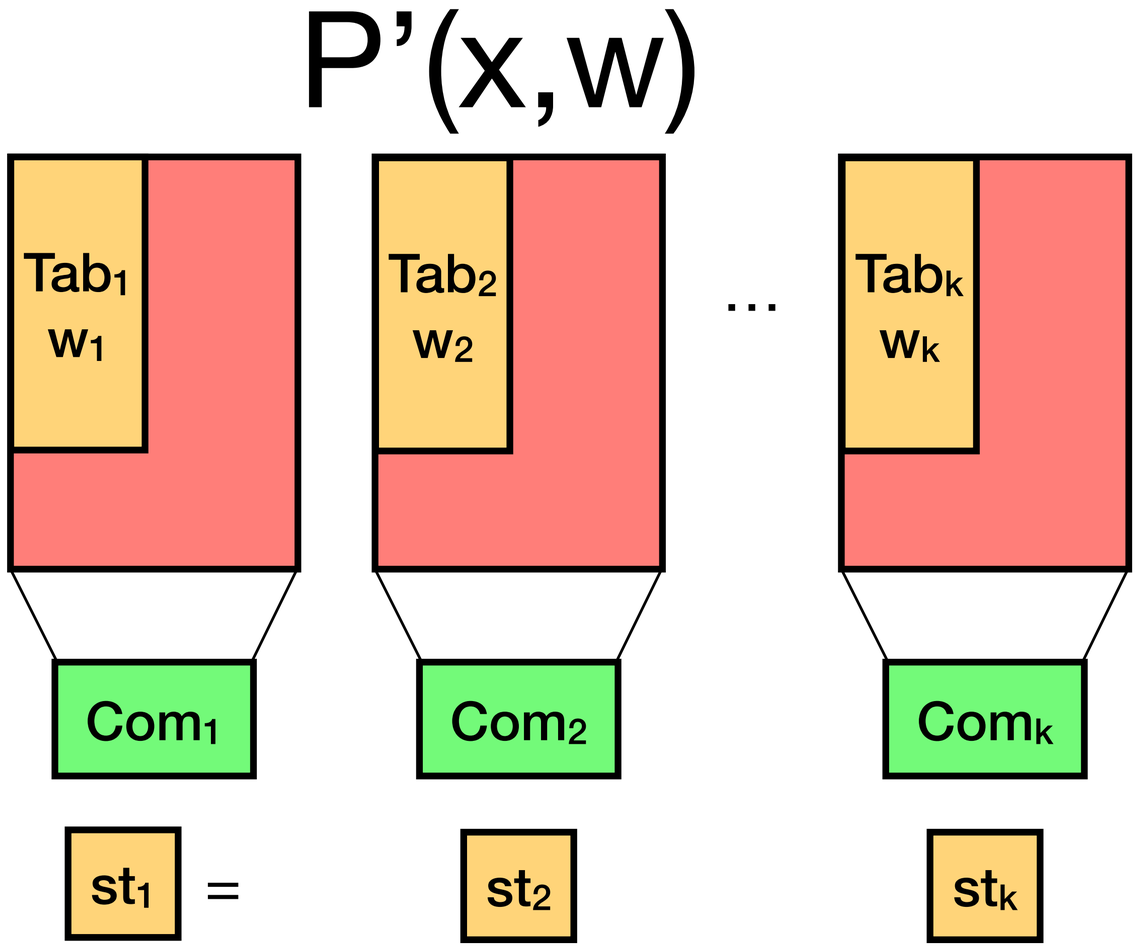
Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

1. A $\tilde{O}(T^{1/2}S)$-space protocol

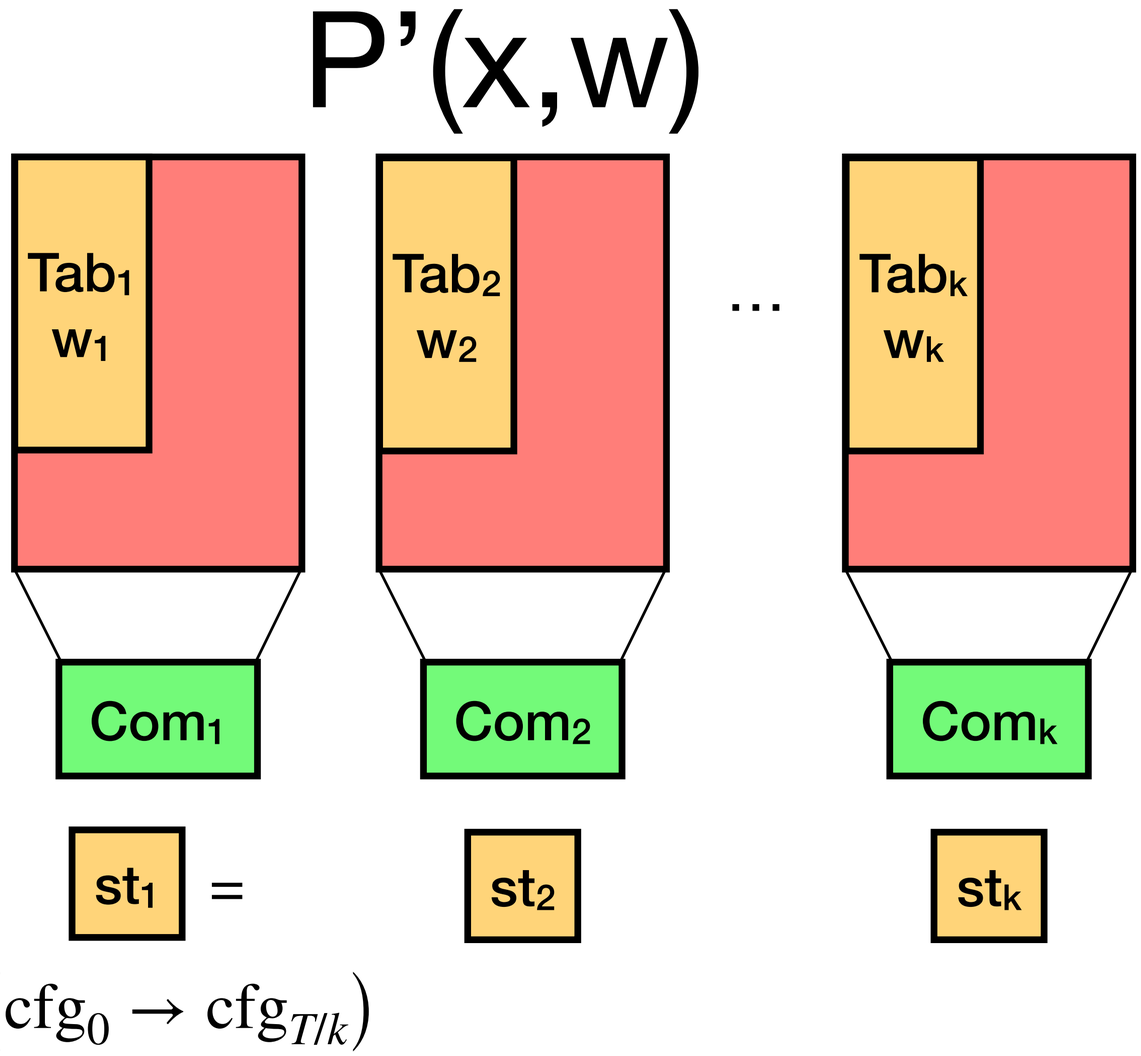
2. A solution for small space

3. Extending to arbitrary space

A $\tilde{O}(T^{1/2}S)$-space protocol: Kilian in Parallel



A $\tilde{O}(T^{1/2}S)$-space protocol: Kilian in Parallel



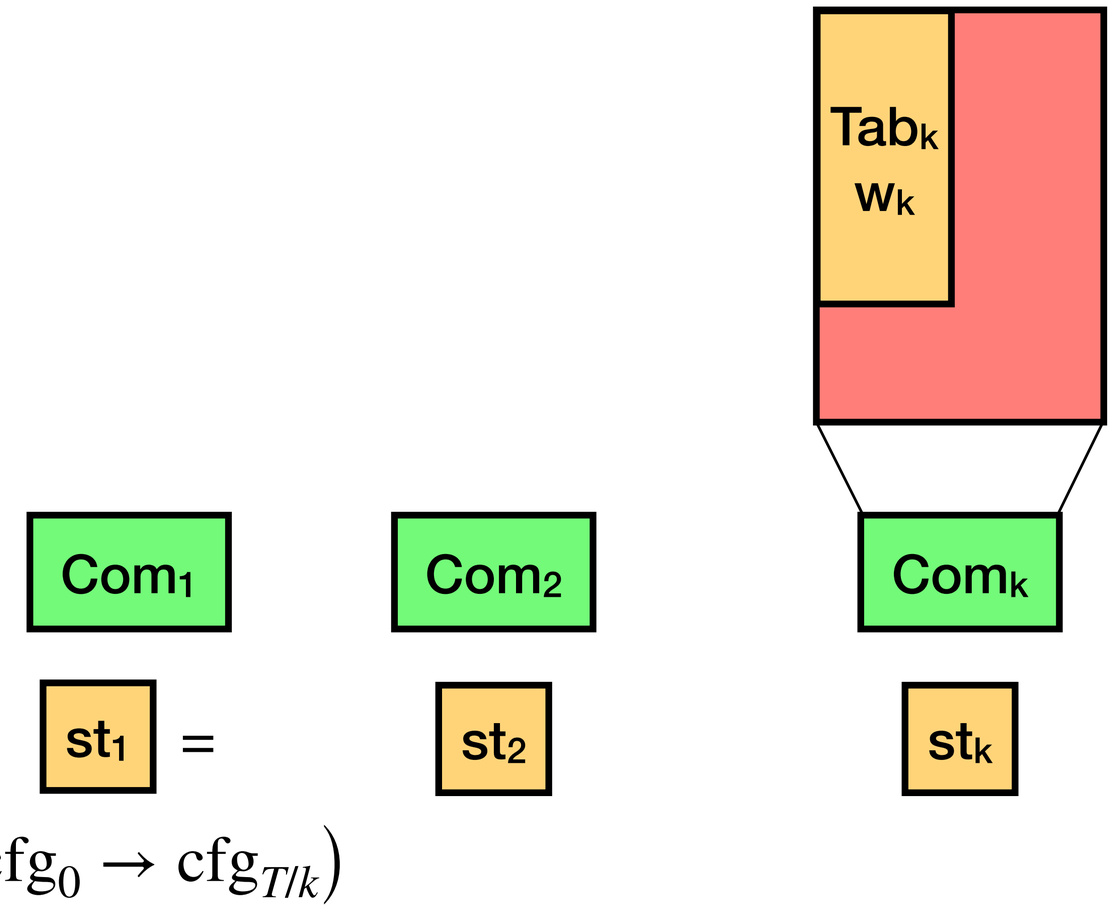
Prover efficiency:

- Time $k \cdot \tilde{O}(ST/k) = \tilde{O}(ST)$
- Space $\tilde{O}(ST/k + Sk)$

  $k = T^{1/2} \Rightarrow \tilde{O}(T^{1/2}S)$

A $\tilde{O}(T^{1/2}S)$-space protocol: Kilian in Parallel

P’(x,w)



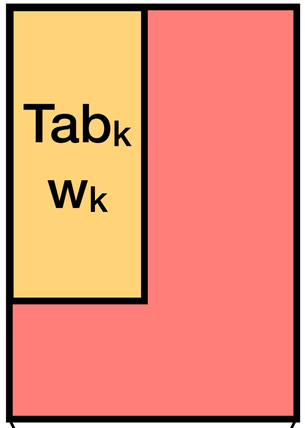
Prover efficiency:

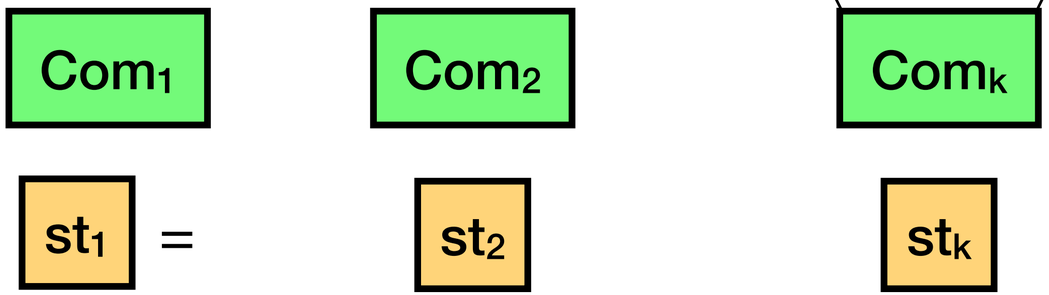
- Time $k \cdot \tilde{O}(ST/k) = \tilde{O}(ST)$
- Space $\tilde{O}(ST/k + Sk)$

  $k = T^{1/2} \Rightarrow \tilde{O}(T^{1/2}S)$

A $\tilde{O}(T^{1/2}S)$-space protocol: Kilian in Parallel

P’(x,w)





Prover efficiency:

- Time $k \cdot \tilde{O}(ST/k) = \tilde{O}(ST)$
- Space $\tilde{O}(ST/k + Sk)$

  $k = T^{1/2} \Rightarrow \tilde{O}(T^{1/2}S)$

Communication/ Verification:

- $\tilde{O}(kS) = \tilde{O}(T^{1/2}S)$

A $\tilde{O}(T^{1/2}S)$-space protocol

Intermediate Result:

There exists a public-coin 4-message argument for NP from CRH with:

- prover **time** $\tilde{O}(TS)$ and **space** $\tilde{O}(T^{1/2}S)$
- **comm/ verification** $\tilde{O}(T^{1/2}S)$.

1. A $\tilde{O}(T^{1/2}S)$-space protocol ✓

2. A solution for small space

A $\tilde{O}(T^{1/2}S)$-space protocol

Intermediate Result:

There exists a public-coin 4-message argument for NP from CRH with:

- prover **time** $\tilde{O}(TS)$ and **space** $\tilde{O}(T^{1/2}S)$
- **comm/ verification** $\tilde{O}(T^{1/2}S)$.

1. A $\tilde{O}(T^{1/2}S)$-space protocol ✓

2. A solution for small space

Need to address:

1. comm/ verification
2. prover space

A $\tilde{O}(T^{1/2}S)$-space protocol

Intermediate Result:

There exists a public-coin 4-message argument for NP from CRH with:

- prover **time** $\tilde{O}(TS)$ and **space** $\tilde{O}(T^{1/2}S)$
- **comm/ verification** $\tilde{O}(T^{1/2}S)$.

Commit-and-prove!

1. A $\tilde{O}(T^{1/2}S)$-space protocol ✓

2. A solution for small space

Need to address:

1. comm/ verification
2. prover space

A $\tilde{O}(T^{1/2}S)$-space protocol

Intermediate Result:

There exists a public-coin 4-message argument for NP from CRH with:

- prover **time** $\tilde{O}(TS)$ and **space** $\tilde{O}(T^{1/2}S)$
- **comm/ verification** $\tilde{O}(T^{1/2}S)$.

1. A $\tilde{O}(T^{1/2}S)$-space protocol ✓

2. A solution for small space

Commit-and-prove!

Need to address:

1. comm/ verification
2. prover space

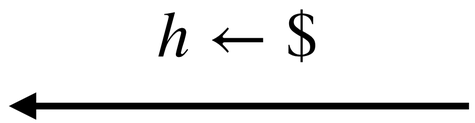
Recursion!

Recursive Protocol

$P_r(w)$ V_r

Recursive Protocol

$P_r(w)$ V_r

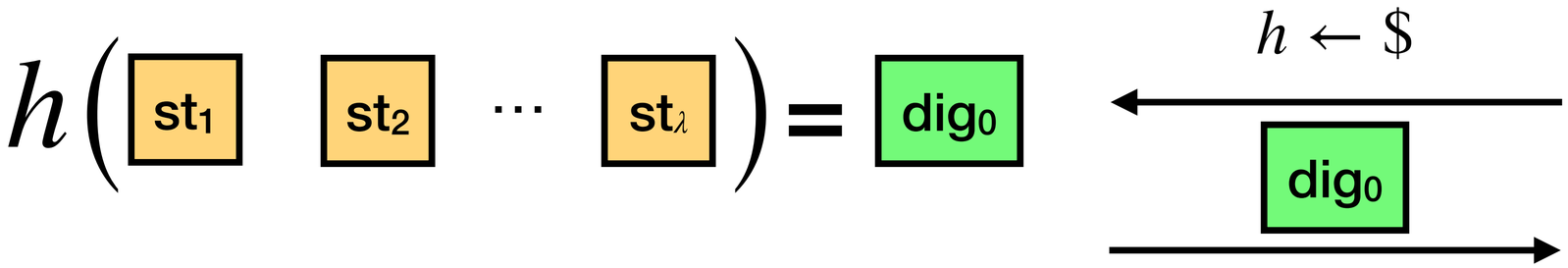


Recursive Protocol

$P_r(w)$ V_r

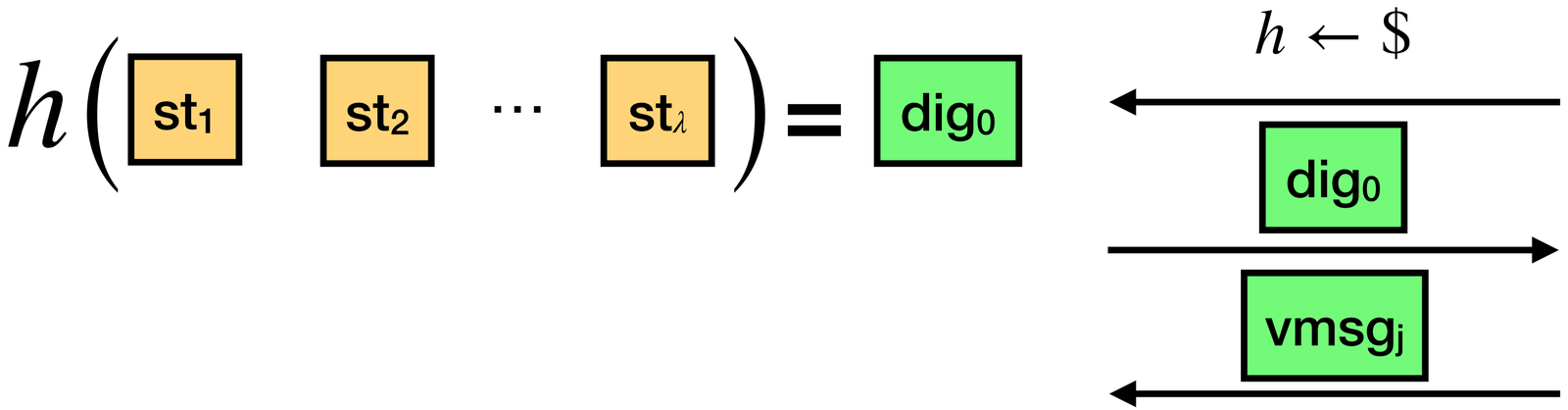
Recursive Protocol

$P_r(w)$ V_r



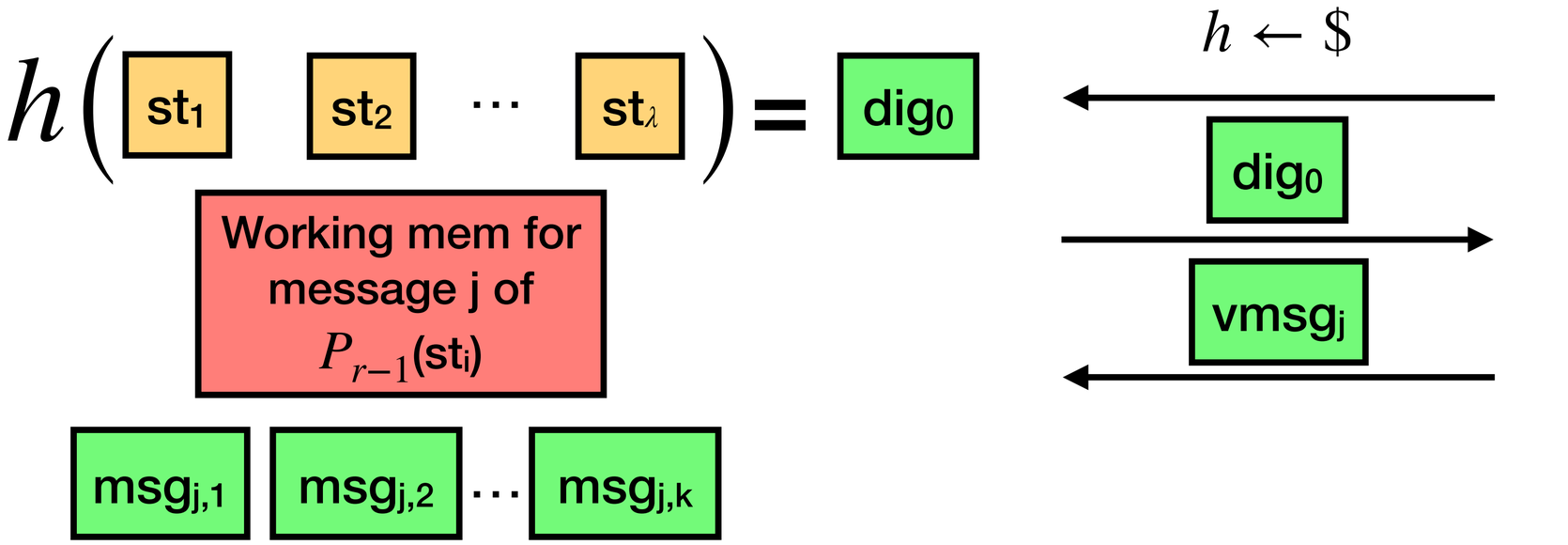
Recursive Protocol

$P_r(w)$ V_r



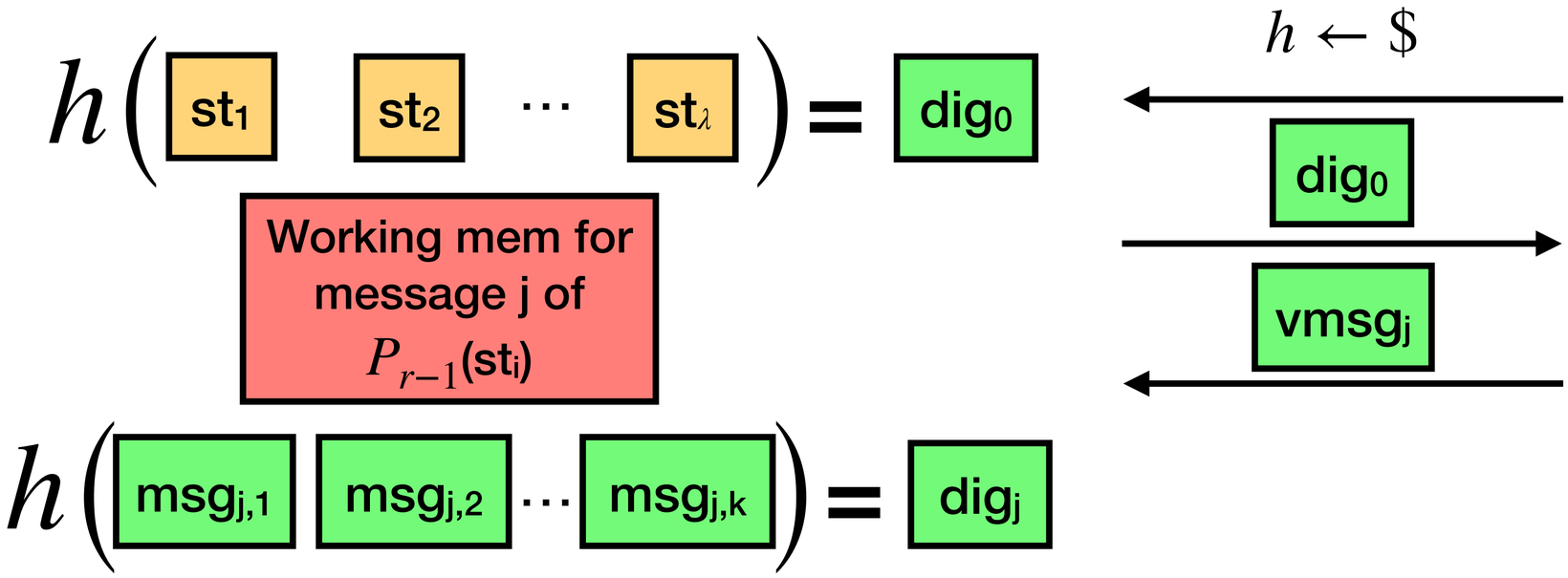
Recursive Protocol

$P_r(w)$ V_r

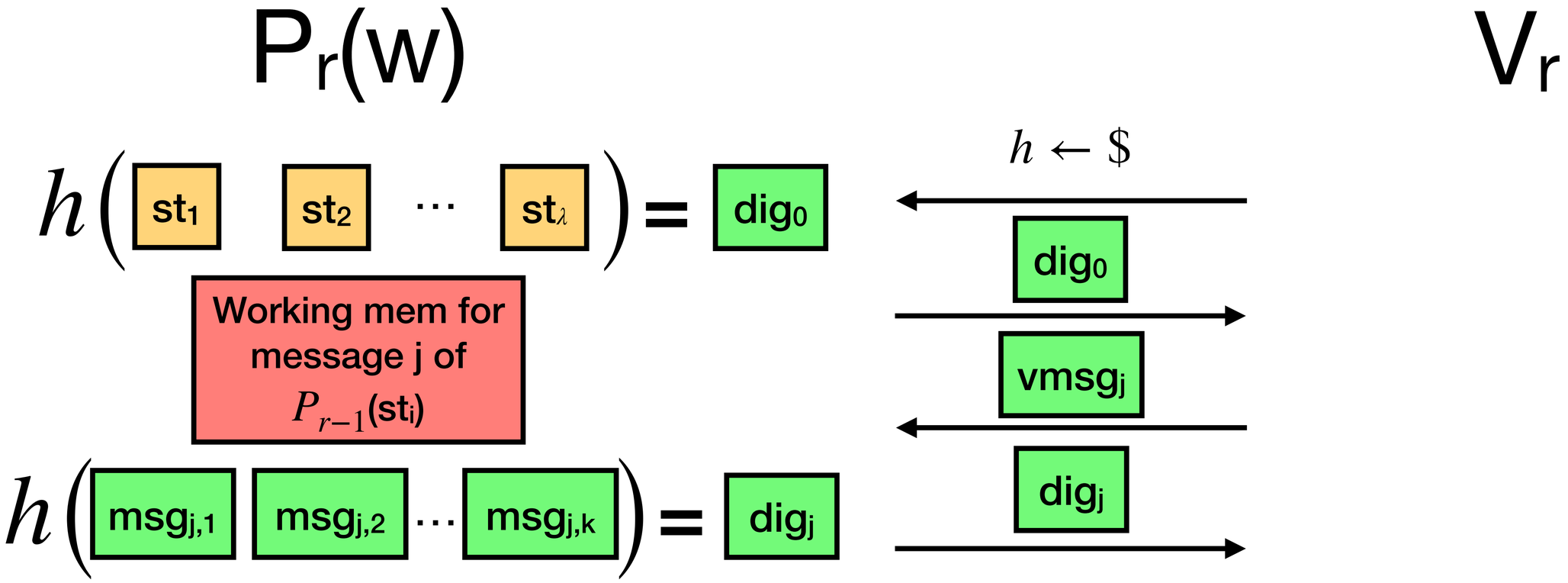


Recursive Protocol

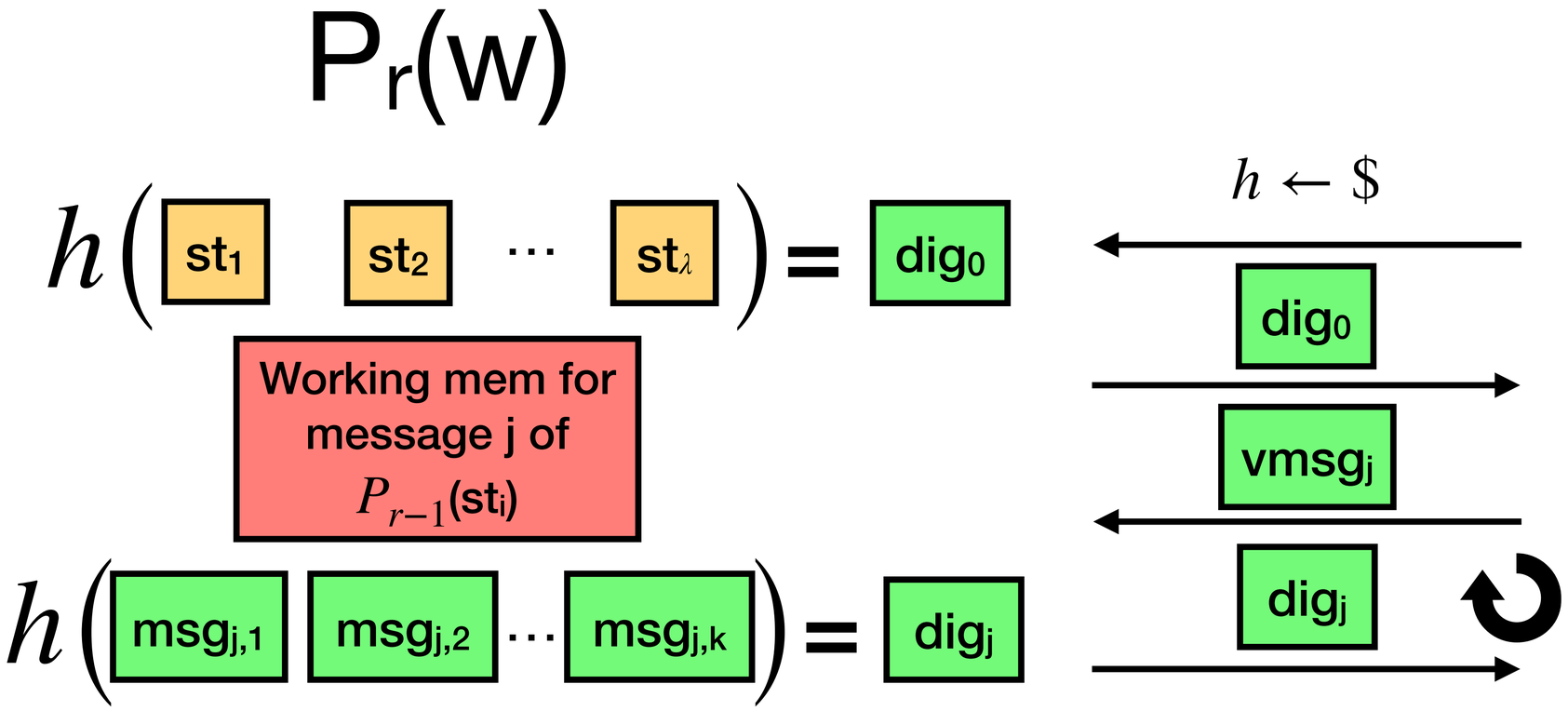
$P_r(w)$ V_r



Recursive Protocol

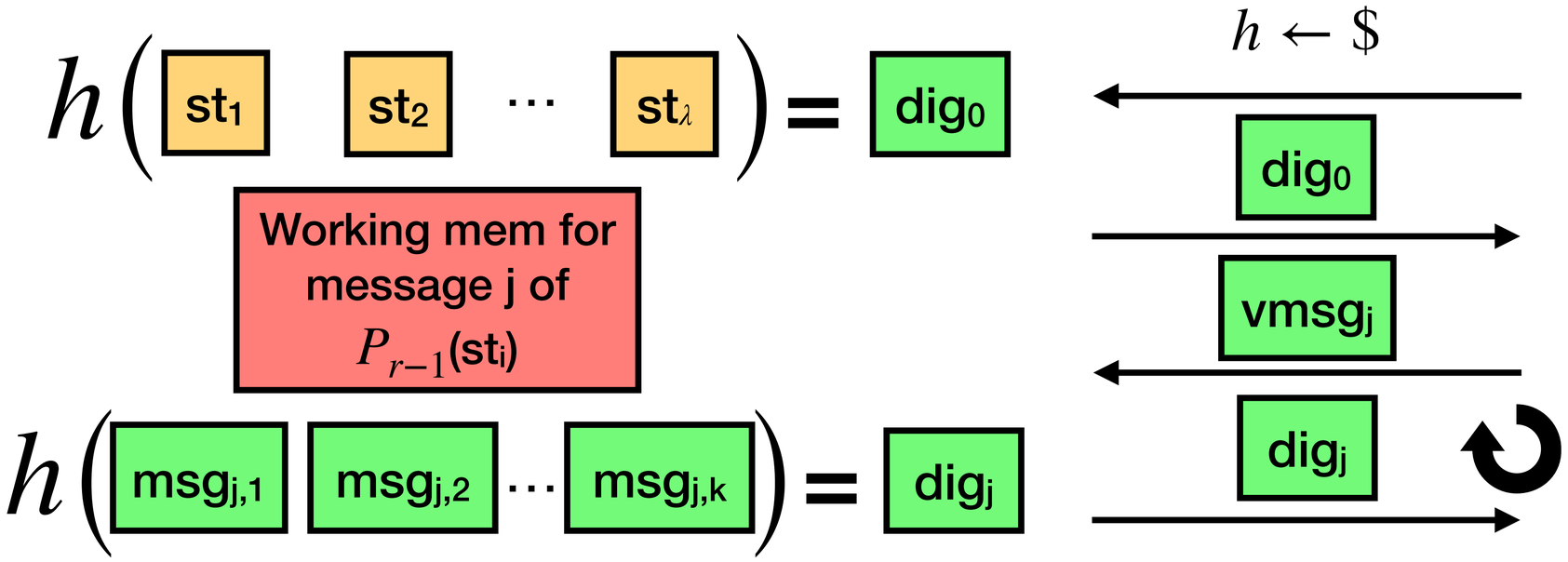


Recursive Protocol



Recursive Protocol

$P_r(w)$ V_r

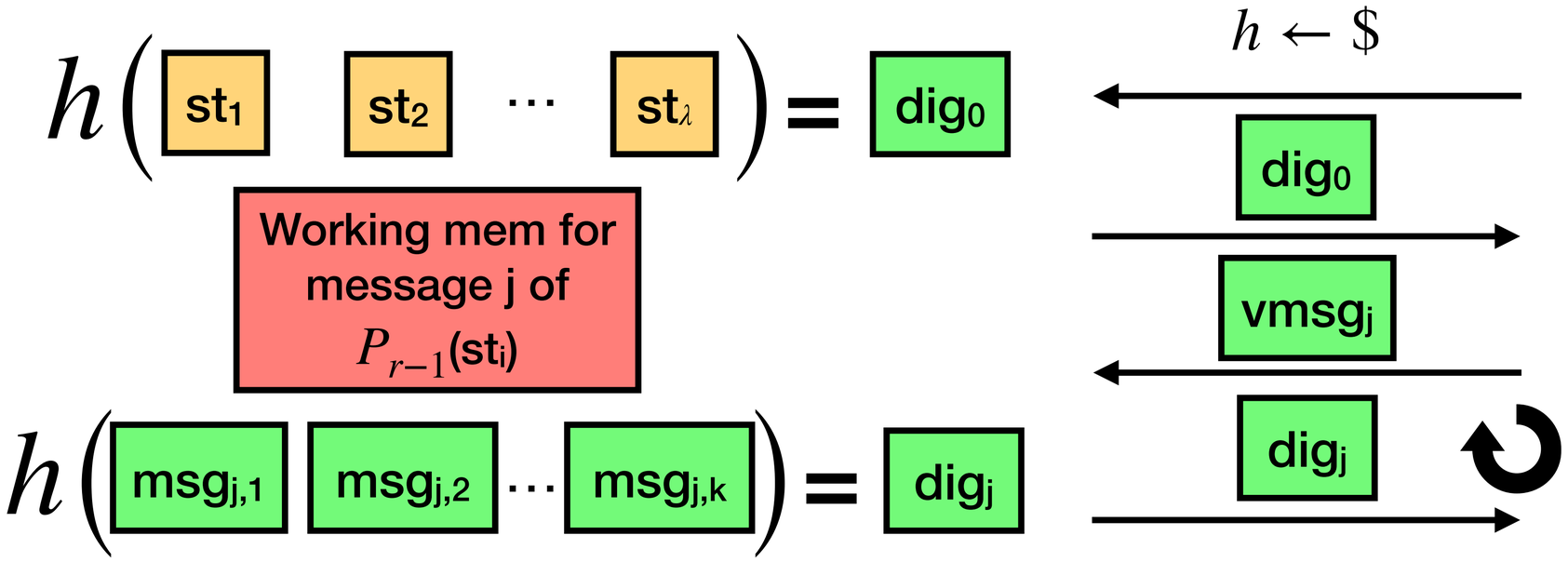


$$R_{\text{merge}} = \left\{ \begin{array}{l} ((\text{cfg}, \text{cfg}', \vec{\text{dig}}), (\vec{\text{st}}, \text{Msg})) : \\ \quad 1.\ \text{dig}_0 = h(\vec{\text{st}}) \\ \quad 2.\ \forall j \geq 1, \text{dig}_j = h(\text{Msg}_{j,*}) \\ \quad 3.\ \forall i \in [k], V_{r-1}(\text{st}_i, \text{Msg}_{*,i}) = 1 \end{array} \right\}$$

Succinct argument for R_{merge}

Recursive Protocol

$P_r(w)$ V_r

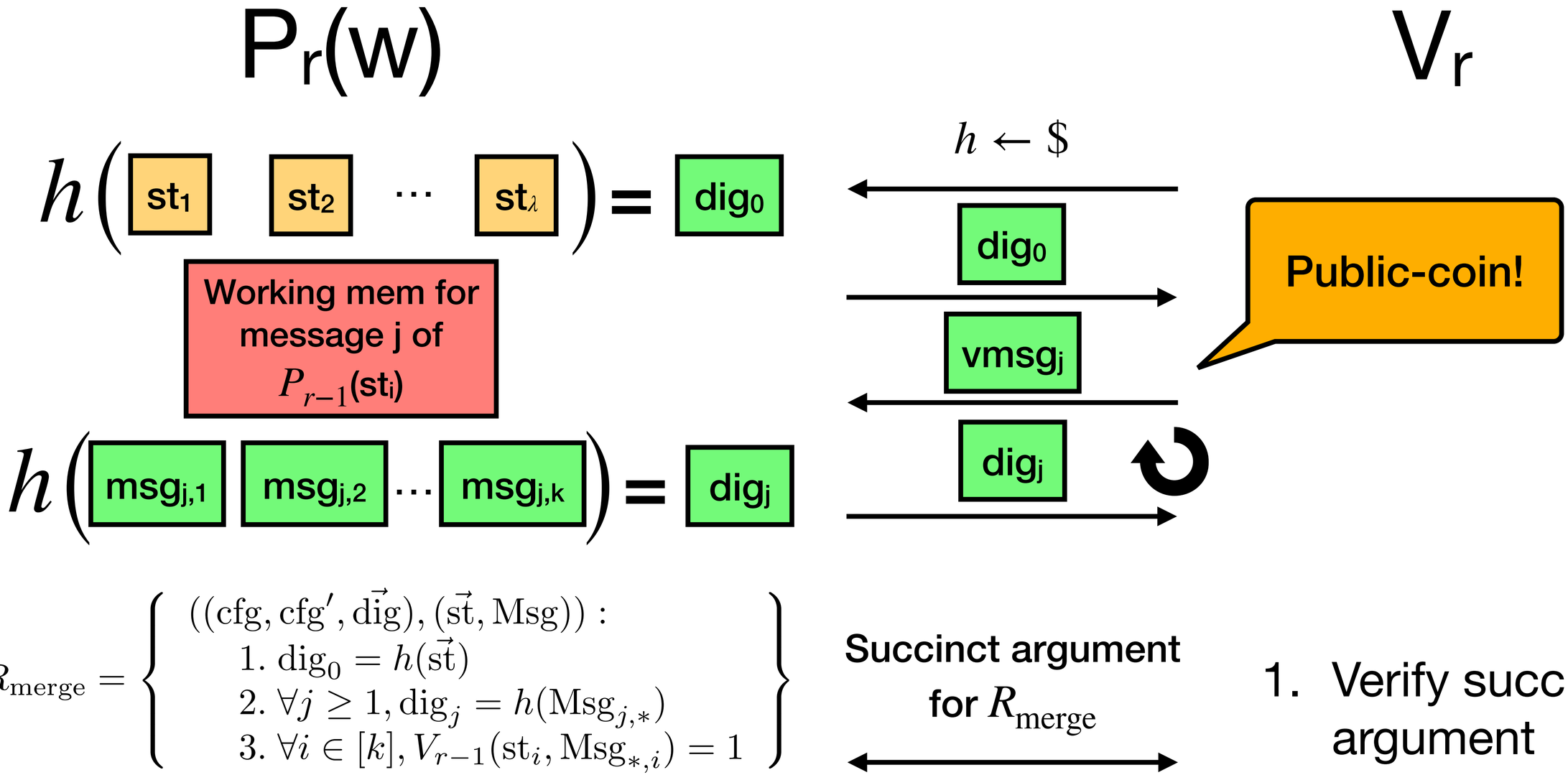


$$R_{\text{merge}} = \left\{ \begin{array}{l} ((\text{cfg}, \text{cfg}', \vec{\text{dig}}), (\vec{\text{st}}, \text{Msg})) : \\ \quad 1.\ \text{dig}_0 = h(\vec{\text{st}}) \\ \quad 2.\ \forall j \geq 1, \text{dig}_j = h(\text{Msg}_{j,*}) \\ \quad 3.\ \forall i \in [k], V_{r-1}(\text{st}_i, \text{Msg}_{*,i}) = 1 \end{array} \right\}$$

Succinct argument for R_{merge}

1. Verify succinct argument

Recursive Protocol



$$R_{\text{merge}} = \left\{ \begin{array}{l} ((\text{cfg}, \text{cfg}', \vec{\text{dig}}), (\vec{\text{st}}, \text{Msg})) : \\ \quad 1.\ \text{dig}_0 = h(\vec{\text{st}}) \\ \quad 2.\ \forall j \geq 1, \text{dig}_j = h(\text{Msg}_{j,*}) \\ \quad 3.\ \forall i \in [k], V_{r-1}(\text{st}_i, \text{Msg}_{*,i}) = 1 \end{array} \right\}$$

Succinct argument for R_{merge}

1. Verify succinct argument

Questions?

Theorem:
There exists a **public-coin** $O(\log_\lambda T)$**-message complexity-preserving** succinct argument for NP from **collision-resistant hashing**

1. A $\tilde{O}(T^{1/2}S)$ -space protocol

2. A solution for small space

3. Extending to arbitrary space

Open Questions:

Fixed constant rounds?

Universal argument?

Turing machines?