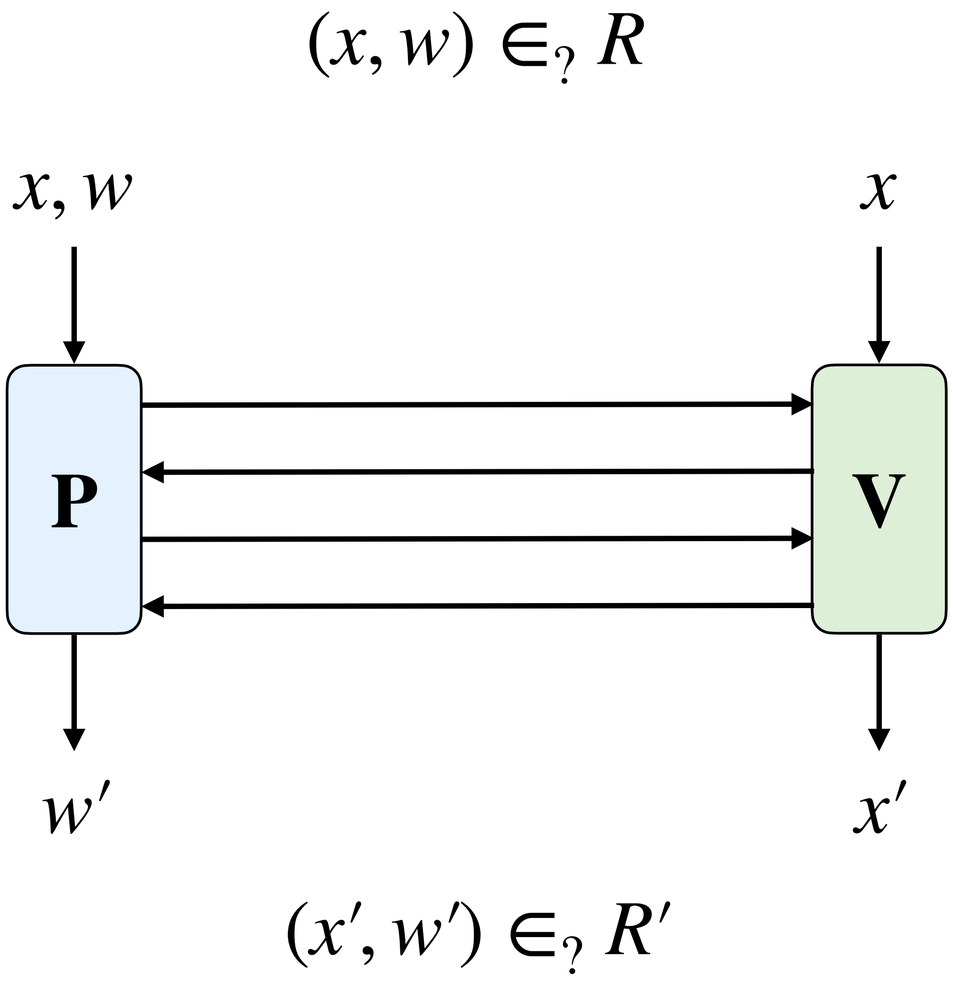
Hash-based Folding Schemes

William Wang (NYU)

with Benedikt Bünz, Alessandro Chiesa, Giacomo Fenzi, Pratyush Mishra, Wilson Nguyen

Reductions



completeness

if $(x, w) \in R$, then

$\langle \mathbf{P}, \mathbf{V} \rangle \to (x', w') \in R'$

soundness

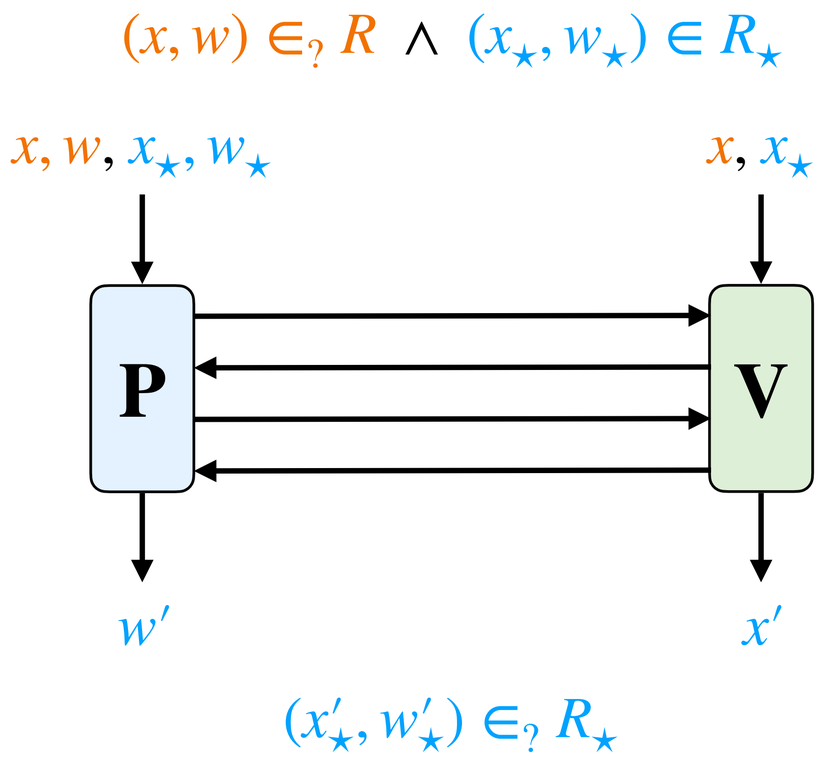
if $x \notin L(R)$, then for any $\tilde{\mathbf{P}}$

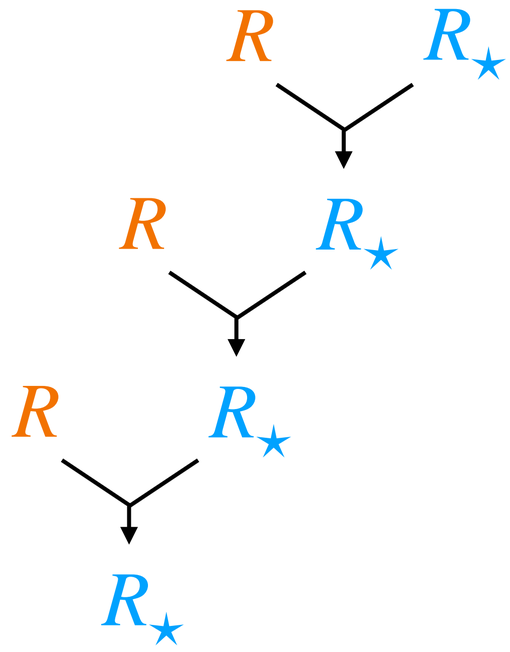
w.h.p. $\langle \tilde{\mathbf{P}}, \mathbf{V} \rangle \to x' \notin L(R')$

later: knowledge soundness

Folding/accumulation schemes [BCLMS21, KST21]

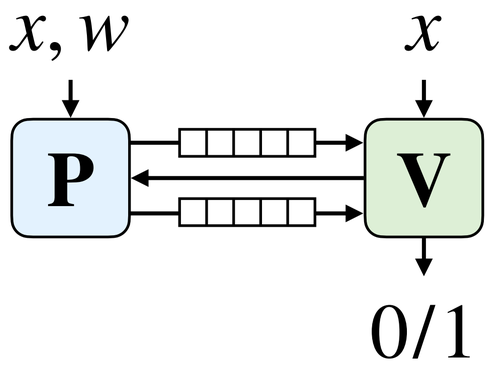
Folding scheme for R is a reduction from $R \times R_\star$ to $R_\star$.





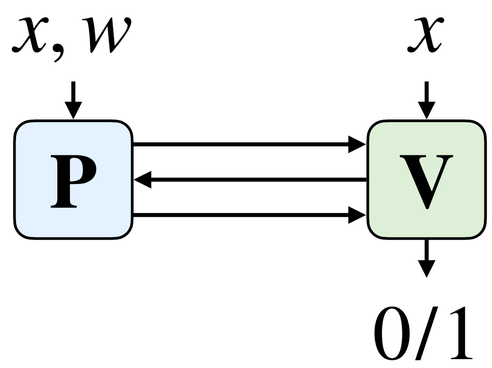
Hash-based recipe [Kil92, Mic00, BCS16, ...]

interactive oracle proof



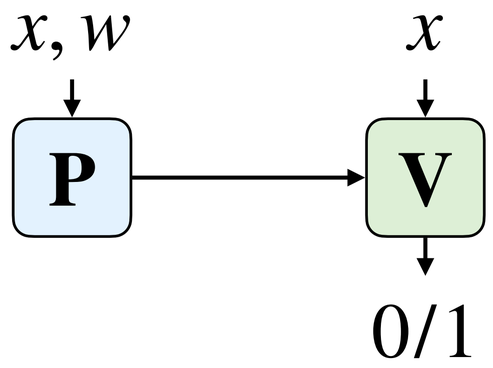
Merkle trees

succinct argument



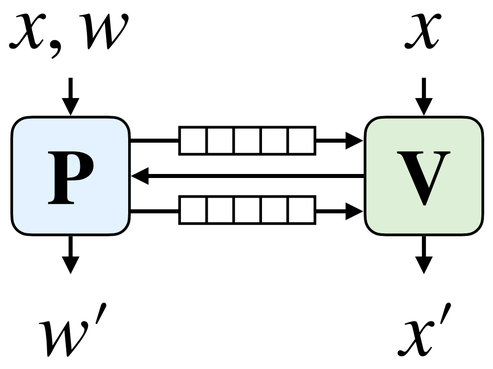
Fiat–Shamir

succinct non-interactive argument



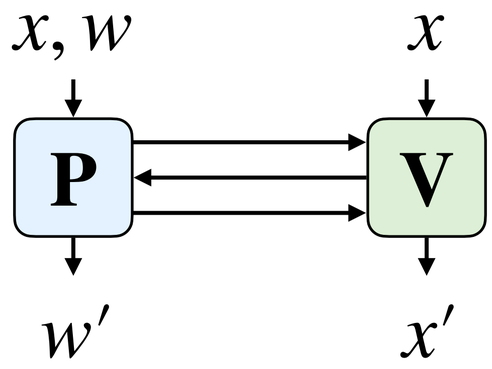
our focus

interactive oracle reduction



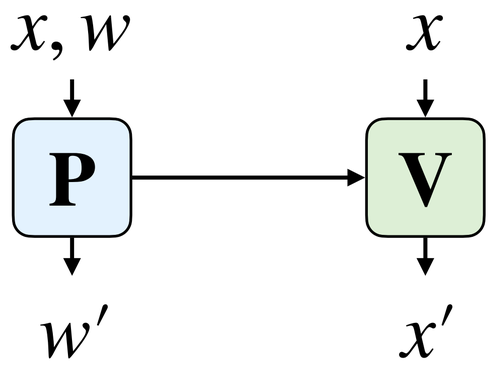
Merkle trees

succinct reduction

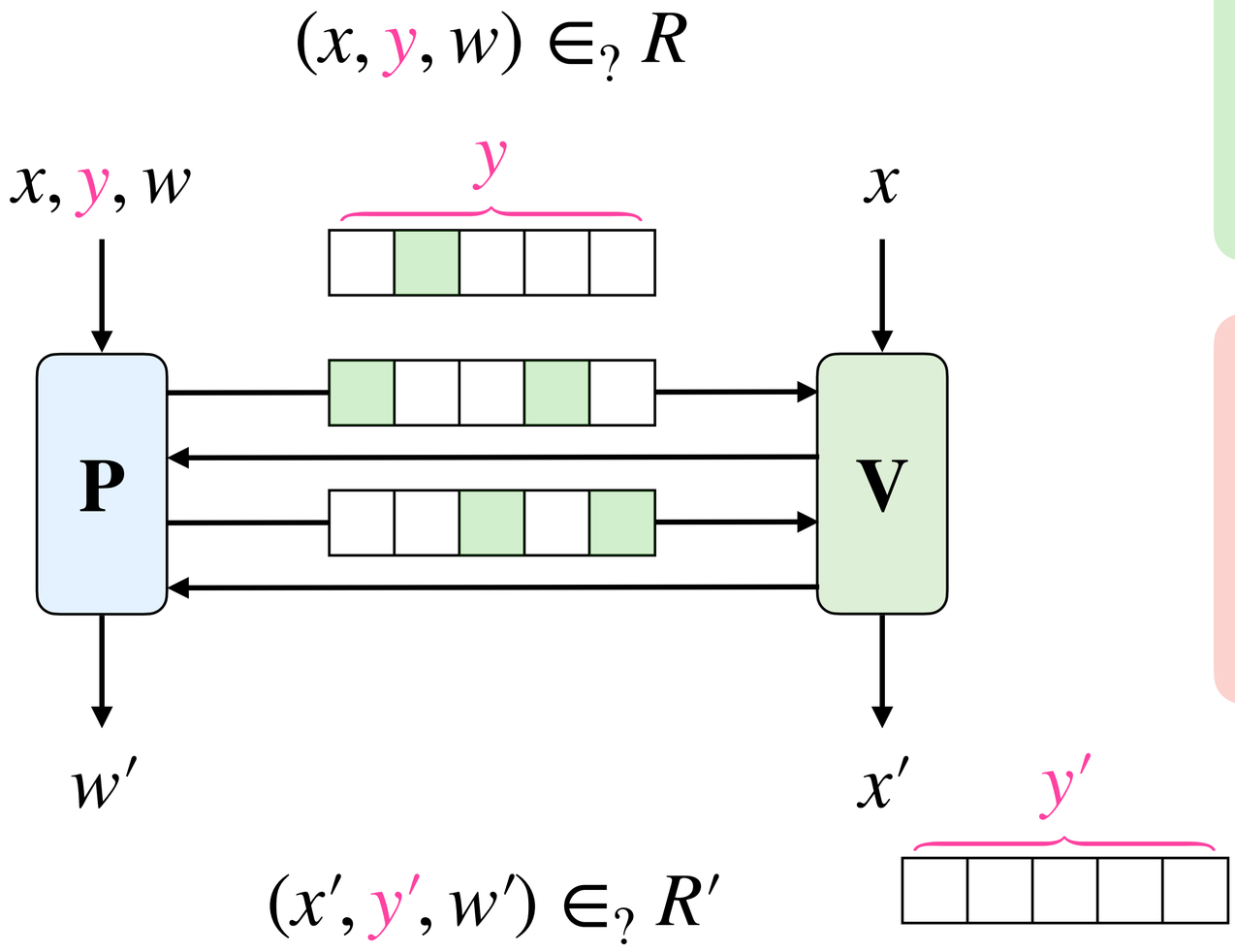


Fiat–Shamir

succinct non-interactive reduction



Interactive oracle reductions [BCGGRS19, BMNW25]



completeness

if $(x, y, w) \in R$, then

$\langle \mathbf{P}, \mathbf{V} \rangle \to (x', y', w') \in R'$

soundness

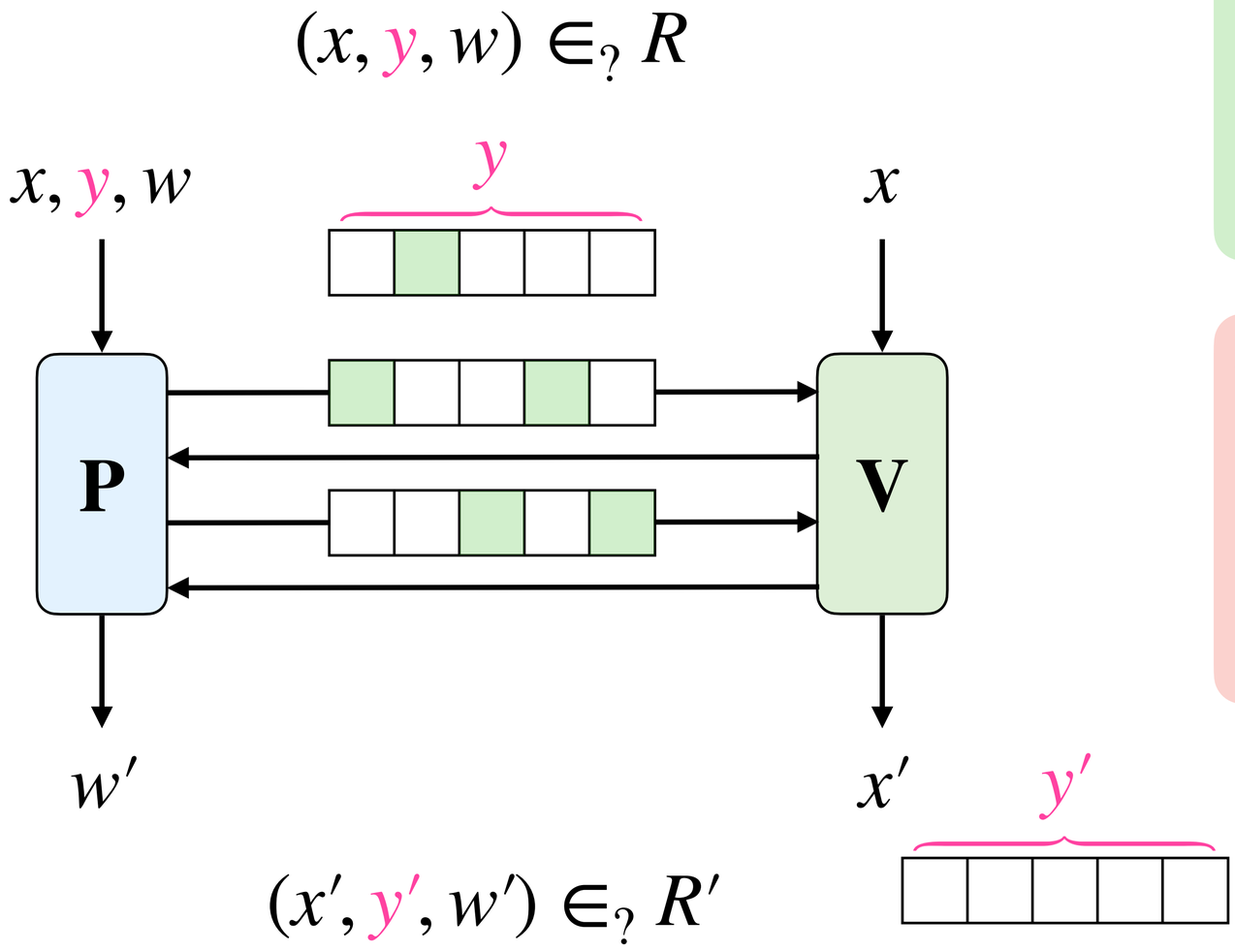
if $y \notin L(R_x)$, then

for any $\tilde{\mathbf{P}}$ w.h.p. $\langle \tilde{\mathbf{P}}, \mathbf{V} \rangle \to x', y'$

s.t. $y' \notin L(R'_{x'})$

$R_x := \{(y, w) : (x, y, w) \in R\}.$

Interactive oracle reductions [BCGGRS19, BMNW25]



completeness

if $(x, y, w) \in R$, then

$\langle \mathbf{P}, \mathbf{V} \rangle \to (x', y', w') \in R'$

soundness

if $\Delta(y, L(R_x)) > \delta$, then

for any $\tilde{\mathbf{P}}$ w.h.p. $\langle \tilde{\mathbf{P}}, \mathbf{V} \rangle \to x', y'$

s.t. $\Delta(y', L(R'_{x'})) > \delta$

$R_x := \{(y, w) : (x, y, w) \in R\}$.

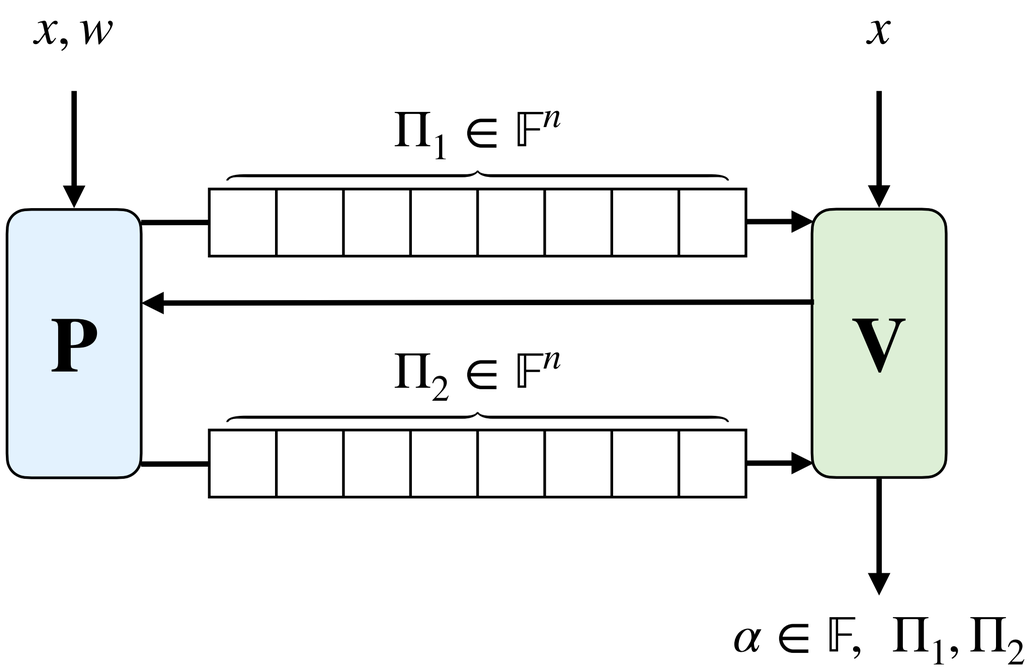
$\Delta(y, y') :=$ fraction of indices where y, y' differ.

IOR examples

Fix linear code $C \subset \mathbb{F}^n$.

reduction to proximity

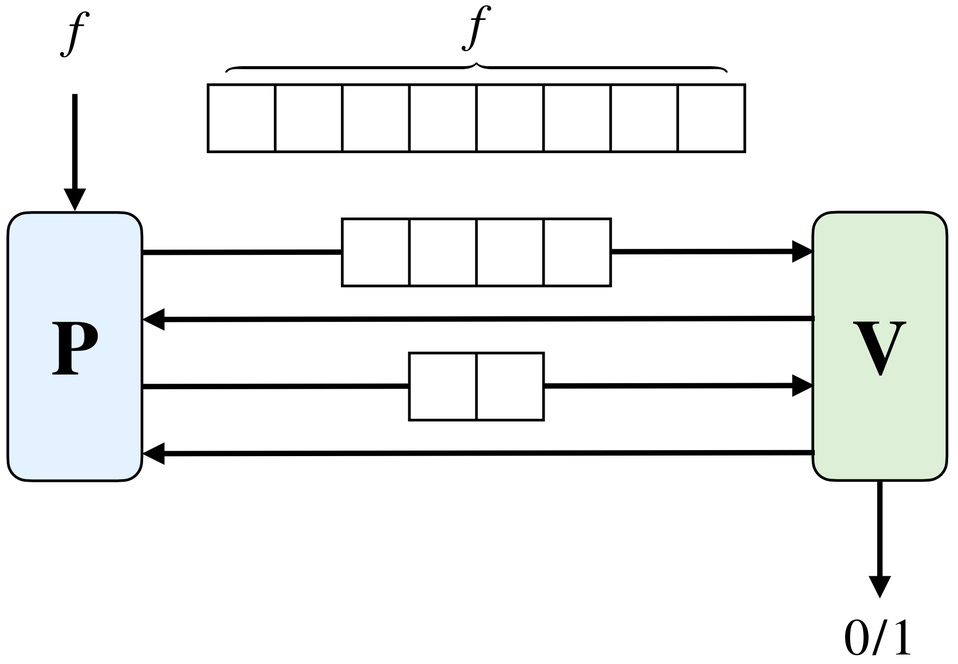
$(x, w) \in R$



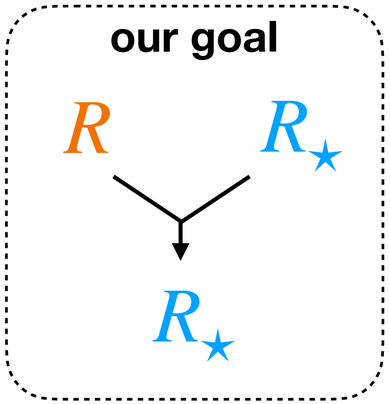
$\Pi_1 + \alpha \cdot \Pi_2 \in_? C$

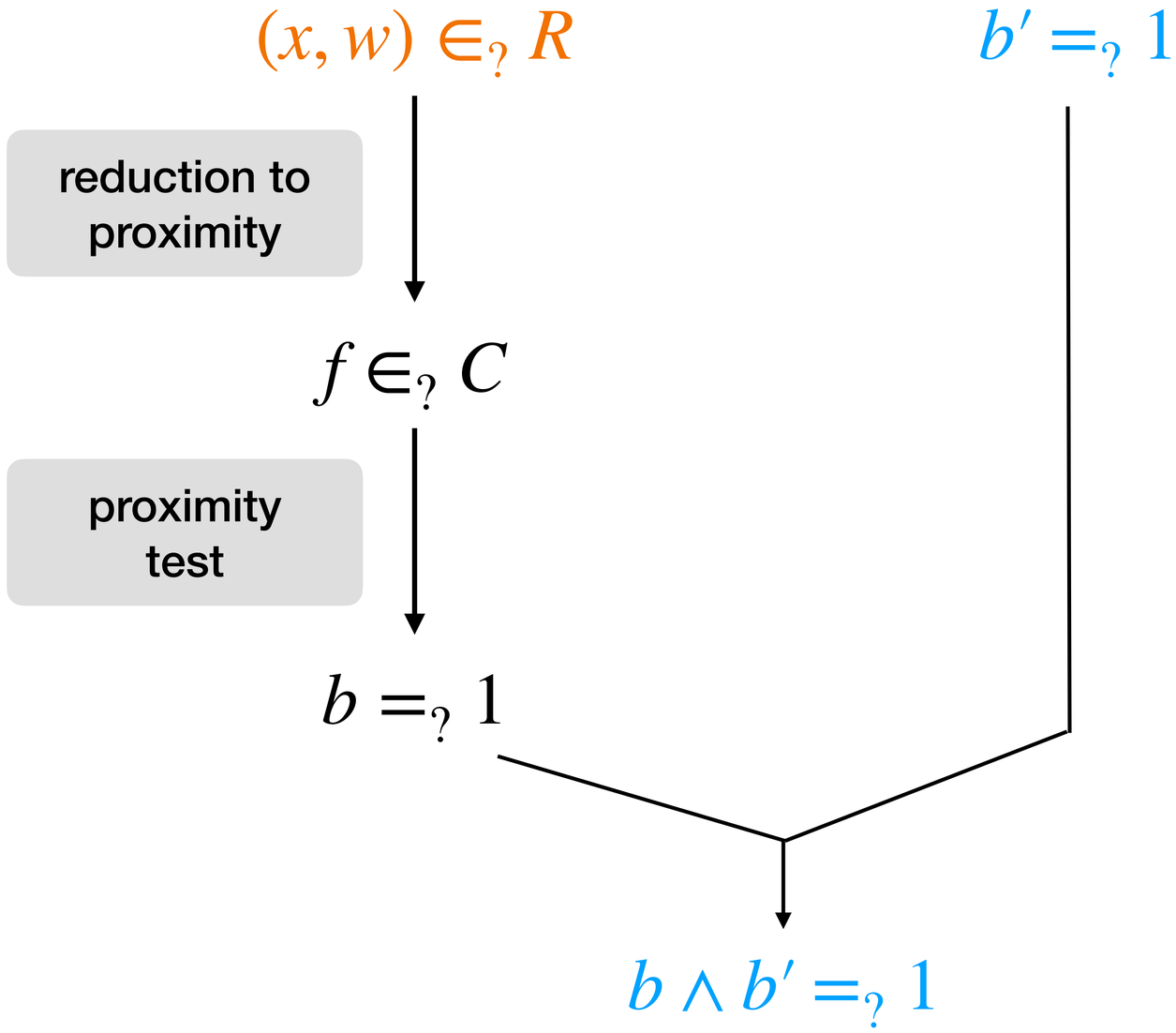
proximity test

$f \in_? C$



A "trivial" folding scheme

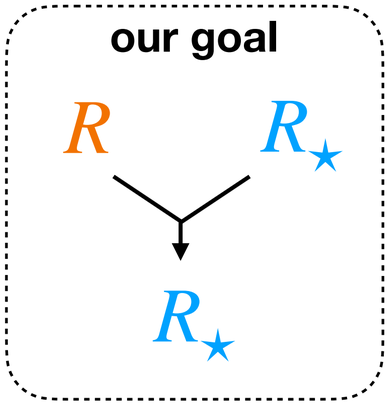


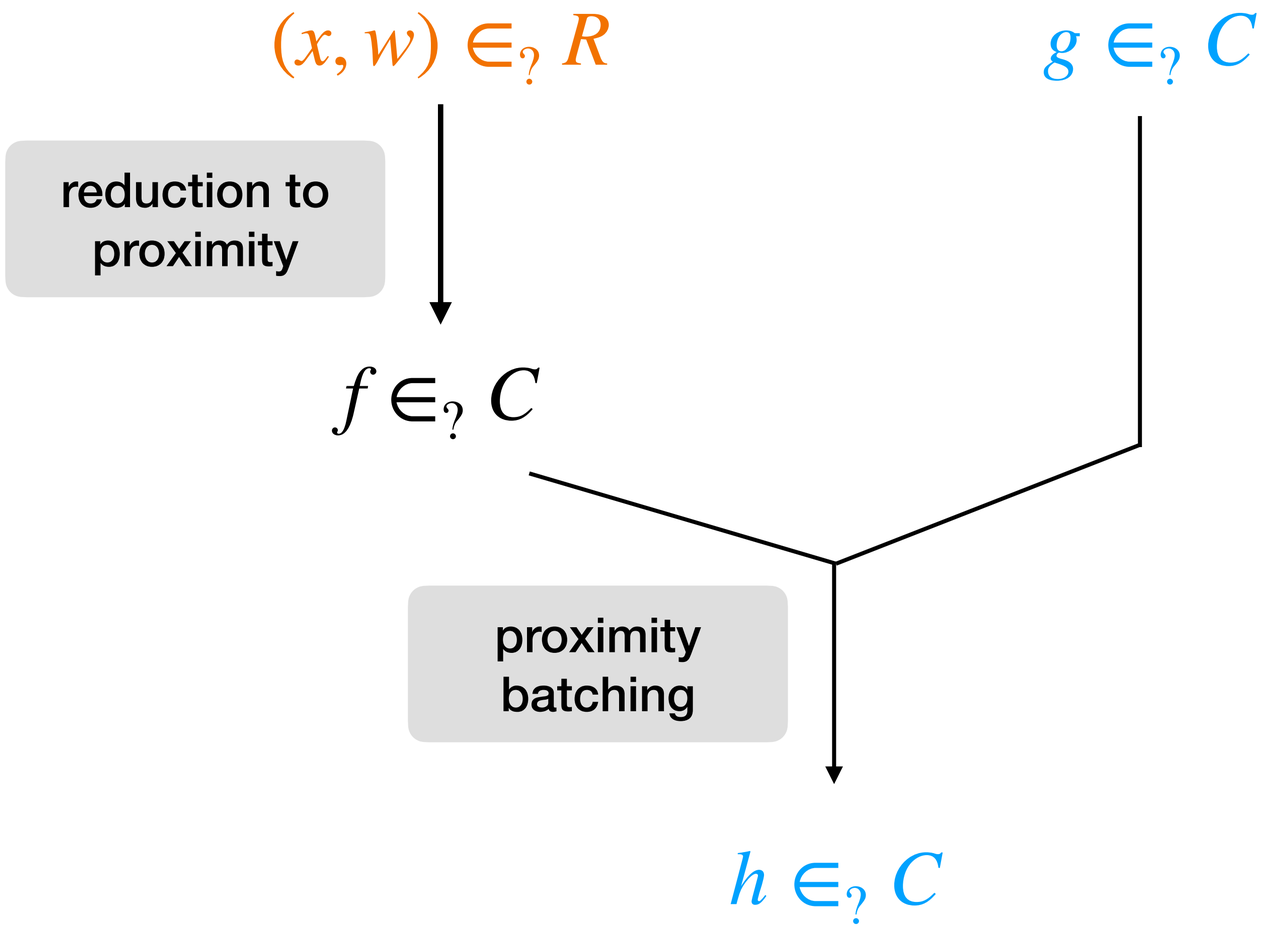


Can we do better?

- faster prover?
- smaller verifier?

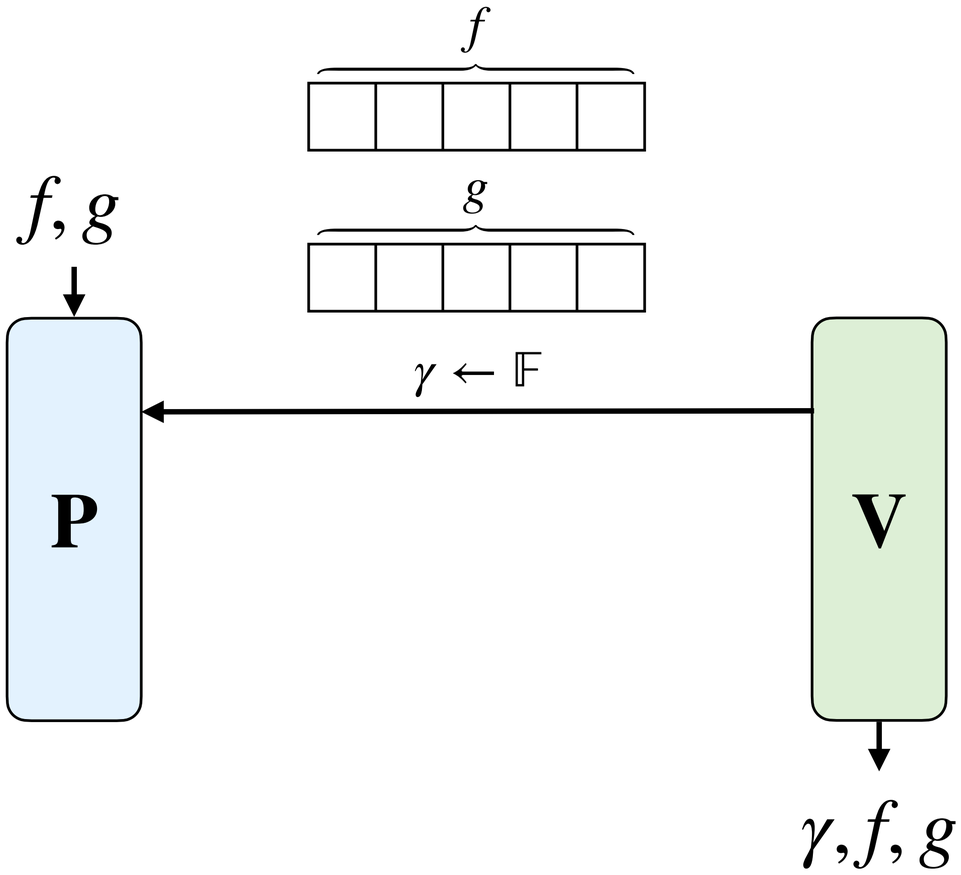
Can we do better?





Proximity batching

$f \in_? C \ \wedge \ g \in_? C$



$f + \gamma \cdot g \in_? C$

completeness

if $f, g \in C$, then

$f + \gamma \cdot g \in C$

soundness

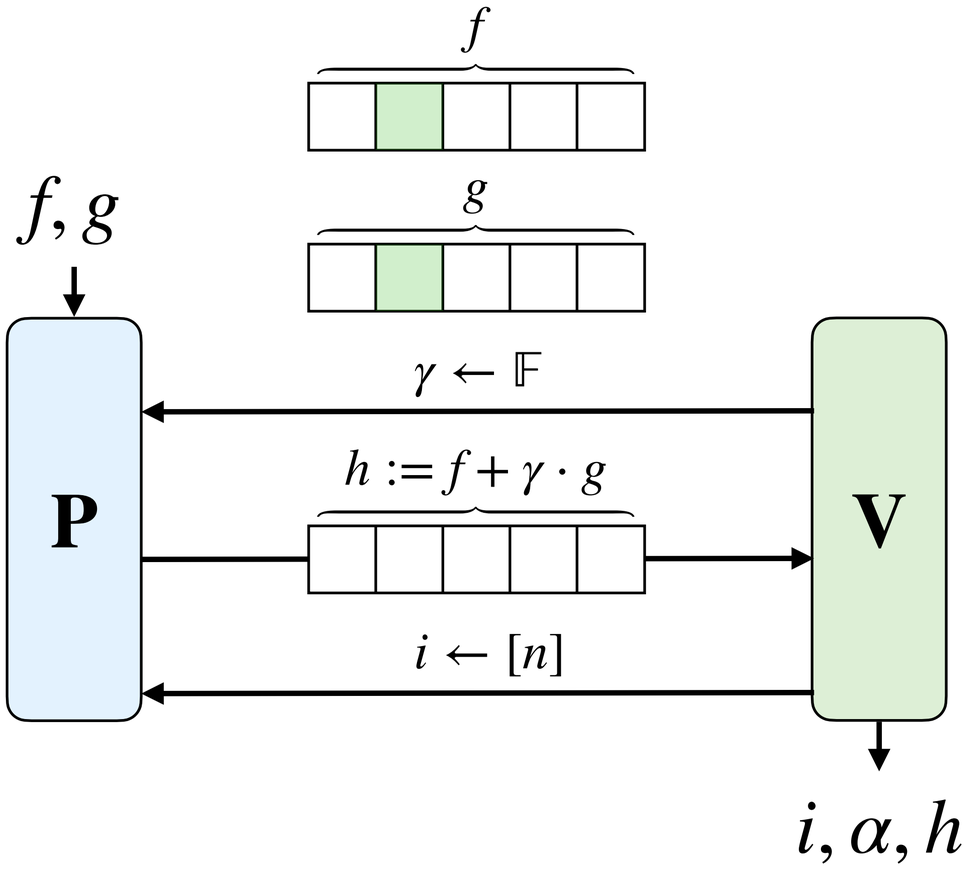
if $\Delta(f, C) > \delta$ or $\Delta(g, C) > \delta$,

then w.h.p. $\Delta(f + \gamma \cdot g, C) > \delta$

think $\delta = 1/3$, distance of C is $2/3$

Proximity batching

$$f \in_? C \ \wedge \ g \in_? C$$



$$h \in_? \{u \in C : u[i] = \alpha\}$$

completeness if $f, g \in C$, then $f + \gamma \cdot g \in C$

since $h := f + \gamma \cdot g$,

$h \in C \ \wedge \ h[i] = \alpha$

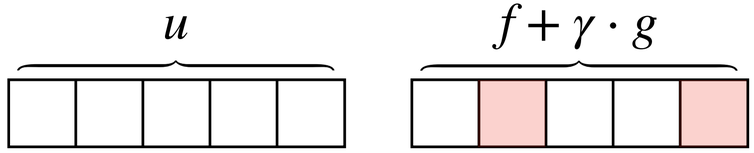
soundness if $\Delta(f, C) > \delta$ or $\Delta(g, C) > \delta$,

then w.h.p. $\Delta(f + \gamma \cdot g, C) > \delta$

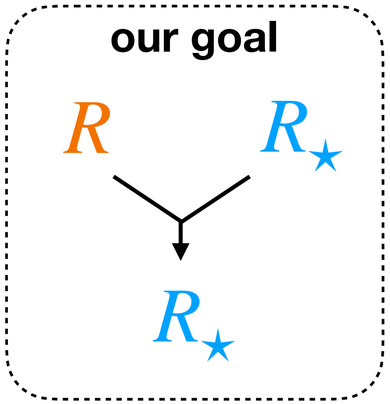
assume unique $u \in C$ with $\Delta(h, u) \leq \delta$

w.p. δ, $u[i] \neq \alpha$

amplify with $O(\lambda)$ queries



Zooming out



$(x, w) \in_? R$

$g \in_? C$

reduction to proximity

$f \in_? C$

proximity batching

$h \in_? \{u \in C : u[i] = \alpha\}$

Constrained codes

Multilinear extension: for $u \in \mathbb{F}^n$ there is a unique multilinear polynomial $\hat{u} : \mathbb{F}^{\log n} \to \mathbb{F}$ such that

$$\hat{u}(\mathbf{x}) = u[\mathsf{int}(\mathbf{x})] \text{ for } \mathbf{x} \in \{0,1\}^{\log n}$$

Constrained codes:

$$C_{\mathbf{x},\alpha} := \{u \in C : \hat{u}(\mathbf{x}) = \alpha\}$$

Fact: $\{u \in C : u[i] = \alpha\} = C_{\mathsf{int}^{-1}(i),\alpha}$

$$\begin{aligned} \mathsf{int}(0,0,0) &= 1 \\ \mathsf{int}(0,0,0) &= 2 \\ \mathsf{int}(0,1,0) &= 3 \\ &\vdots \\ \mathsf{int}(1,1,1) &= 8 \end{aligned}$$

Constrained code reductions

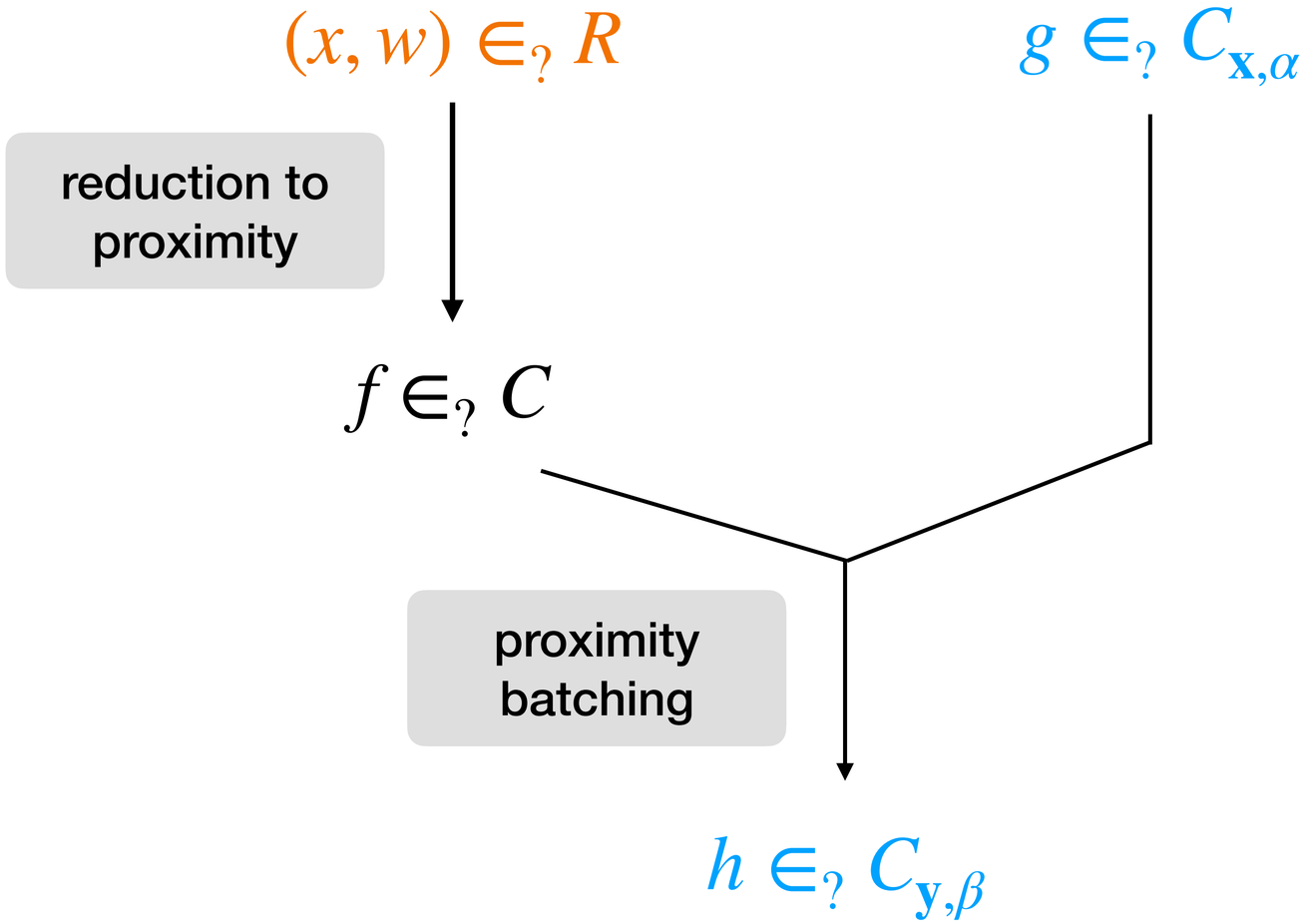
reduction toolbox

$$f_1 \in_? C_{\mathbf{x}_1,\alpha_1} \qquad f_2 \in_? C_{\mathbf{x}_2,\alpha_2}$$

$$f_1 + \gamma \cdot f_2 \in_? C_{\mathbf{y},\beta}$$

$$f \in_? C_{\mathbf{x}_1,\alpha_1} \cap \cdots \cap C_{\mathbf{x}_t,\alpha_t}$$

$$f \in_? C_{\mathbf{y},\beta}$$



WARP [BCFW25]

an essentially optimal hash-based folding scheme for R1CS (and more)

Under the hood:

- instantiable with any linear code
- proximity batching for $\{u \in C : u$ encodes an R1CS witness$\}$

Matrices $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{F}^{M \times N}$ with S non-zero entries

Instance $x \in \mathbb{F}^{N-k}$, witness $w \in \mathbb{F}^k$

$$R_{\text{R1CS}} = \left\{ (x, w) : \mathbf{A} \begin{bmatrix} x \\ w \end{bmatrix} \circ \mathbf{B} \begin{bmatrix} x \\ w \end{bmatrix} = \mathbf{C} \begin{bmatrix} x \\ w \end{bmatrix} \right\}$$

Prover cost:

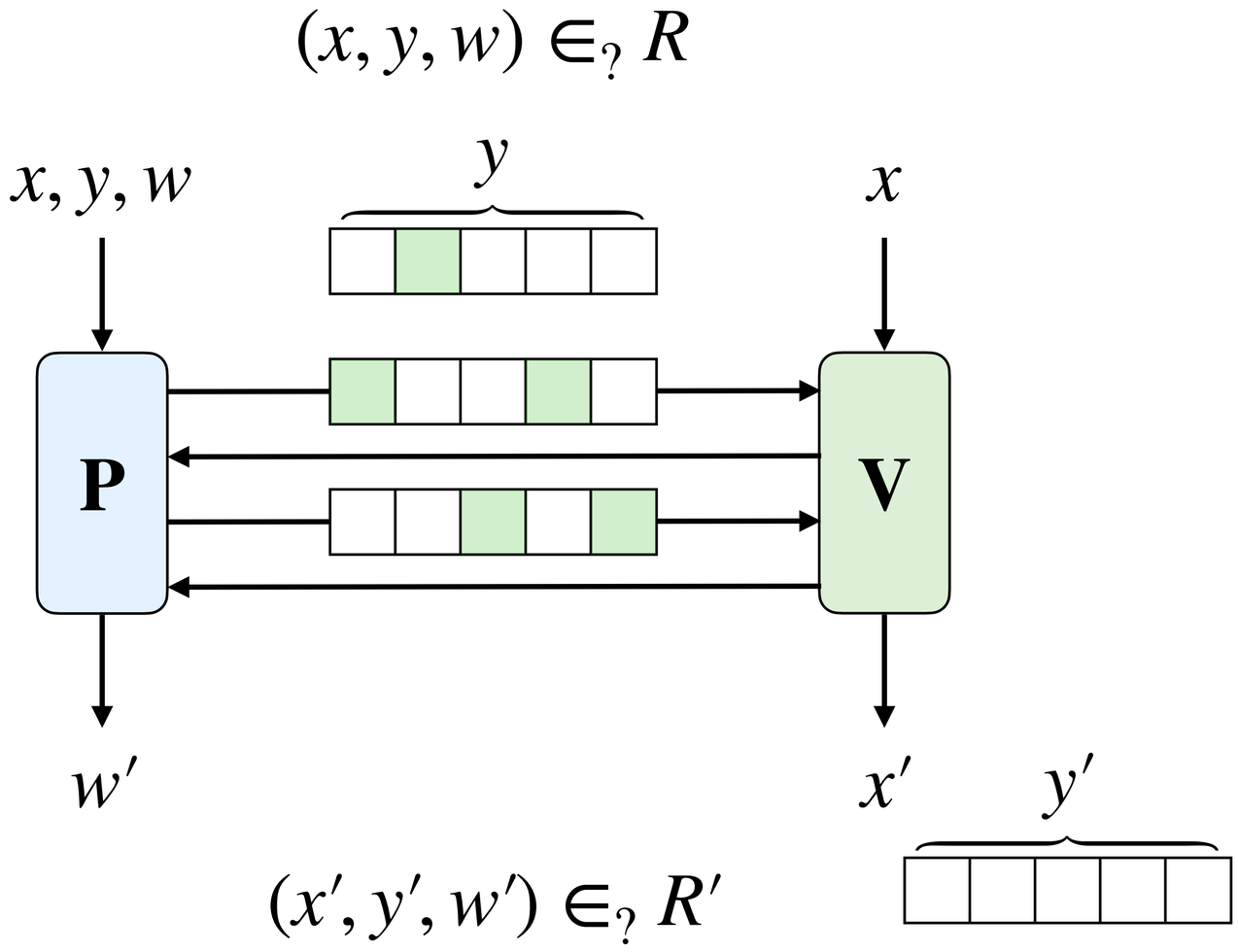
- $O(S)$ $\mathbb{F}$-ops (linear time)
- $O(k)$ random oracle queries

Verifier cost:

- $O(\log N + \log M + \lambda)$ $\mathbb{F}$-ops
- $O(\lambda \cdot \log k)$ random oracle queries

Instantiable with any $\mathbb{F}$ sufficiently large for soundness.

Knowledge soundness



knowledge soundness (no witness)

for any $\tilde{\mathbf{P}}$ w.h.p. $\langle \tilde{\mathbf{P}}, \mathbf{V} \rangle \to x', y'$:

given $\bar{y}'$ s.t.

$\Delta(y', \bar{y}') \leq \delta, \ \ \bar{y}' \in L(R'_{x'})$

$\Downarrow$

$\mathbf{E}(x, y, \tilde{\mathbf{P}}, \bar{y}') \to \bar{y}$ s.t.

$\Delta(y, \bar{y}) \leq \delta, \ \ \bar{y} \in L(R_x)$

how does extractor $\mathbf{E}$ work?

straightline: $\mathbf{E}$ runs $\mathbf{P}$ in one shot

rewinding: $\mathbf{E}$ can restore $\mathbf{P}$ to arbitrary states

- only expected polynomial time
- worse concrete efficiency
- incompatible with UC security

Extraction strategies

C has an efficient error corrector (Reed–Solomon codes)

- straightline extraction
- no linear-time encodable codes (of practical interest)

C does not have an efficient error corrector (expander codes, RAA codes)

- rewinding extraction
- linear-time encodable codes

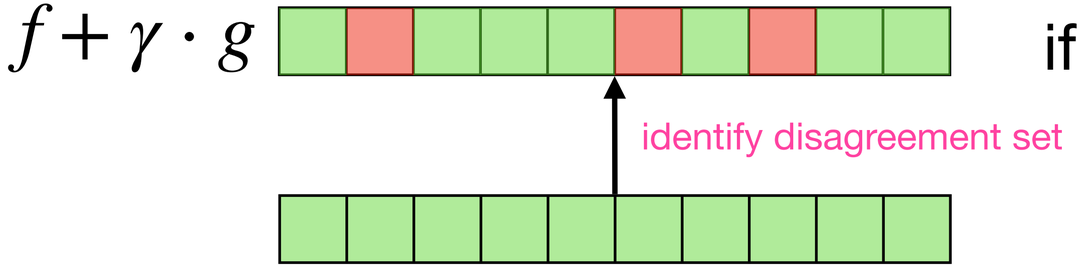
WARP: straightline extraction, even if C does not have an efficient error corrector

Straightline extraction for any linear code

then $\Delta(f, C) \leq \delta$ and $\Delta(g, C) \leq \delta$ by proximity gap

traditionally, error-correct

erase disagreement set by proximity gap with mutual correlated agreement



if $\Delta(f + \gamma \cdot g, C) \leq \delta$

given the new codeword

Straightline extraction for any linear code

Three steps

1. Leverage the new codeword
   - New definition of round-by-round knowledge soundness
2. Proximity gap with mutual correlated agreement (known for any linear code)
3. Erasure correction (known for any linear code)

Applicable to linear-time succinct arguments

- Blaze, BaseFold, Brakedown, ...

Some open questions

- Open questions for succinct arguments → folding schemes
  - Linear-time folding over small fields
  - Linear-time folding with constant round complexity
- Straightline extraction for hash-based succinct arguments

https://ia.cr/2024/474

https://ia.cr/2024/1731

https://ia.cr/2025/753