

# SLAP



## Succinct Lattice-Based Polynomial Commitment Schemes from Standard Assumptions

Giacomo Fenzi @ **EPFL**

Joint work with:  
**Martin Albrecht**  
**Ngoc Khanh Nguyen**

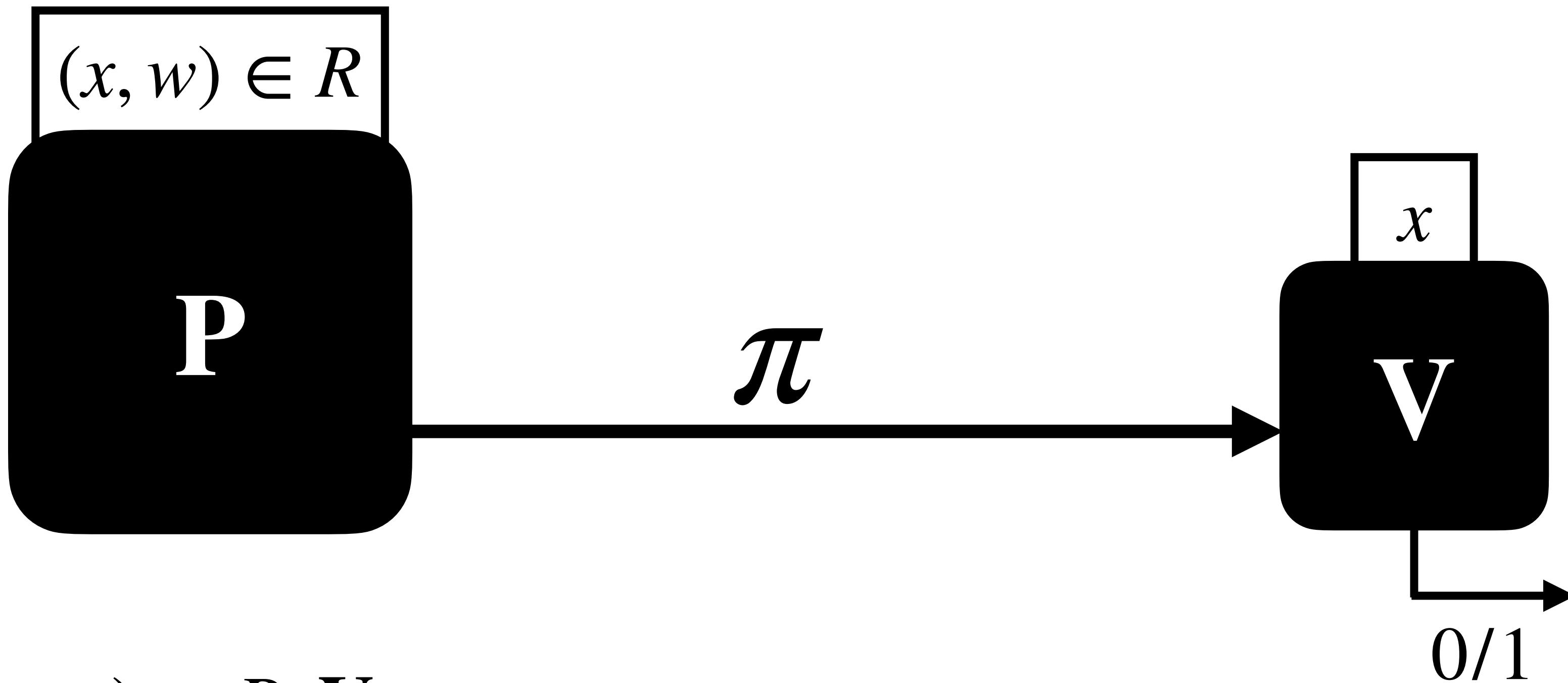
Oleksandra Lapiha



# Motivation

# SNARKs

(Succinct Non-Interactive ARguments of Knowledge)



**Complete:** if  $(x, w) \in R$ , V accepts.

**Non-interactive:** P sends a single message.

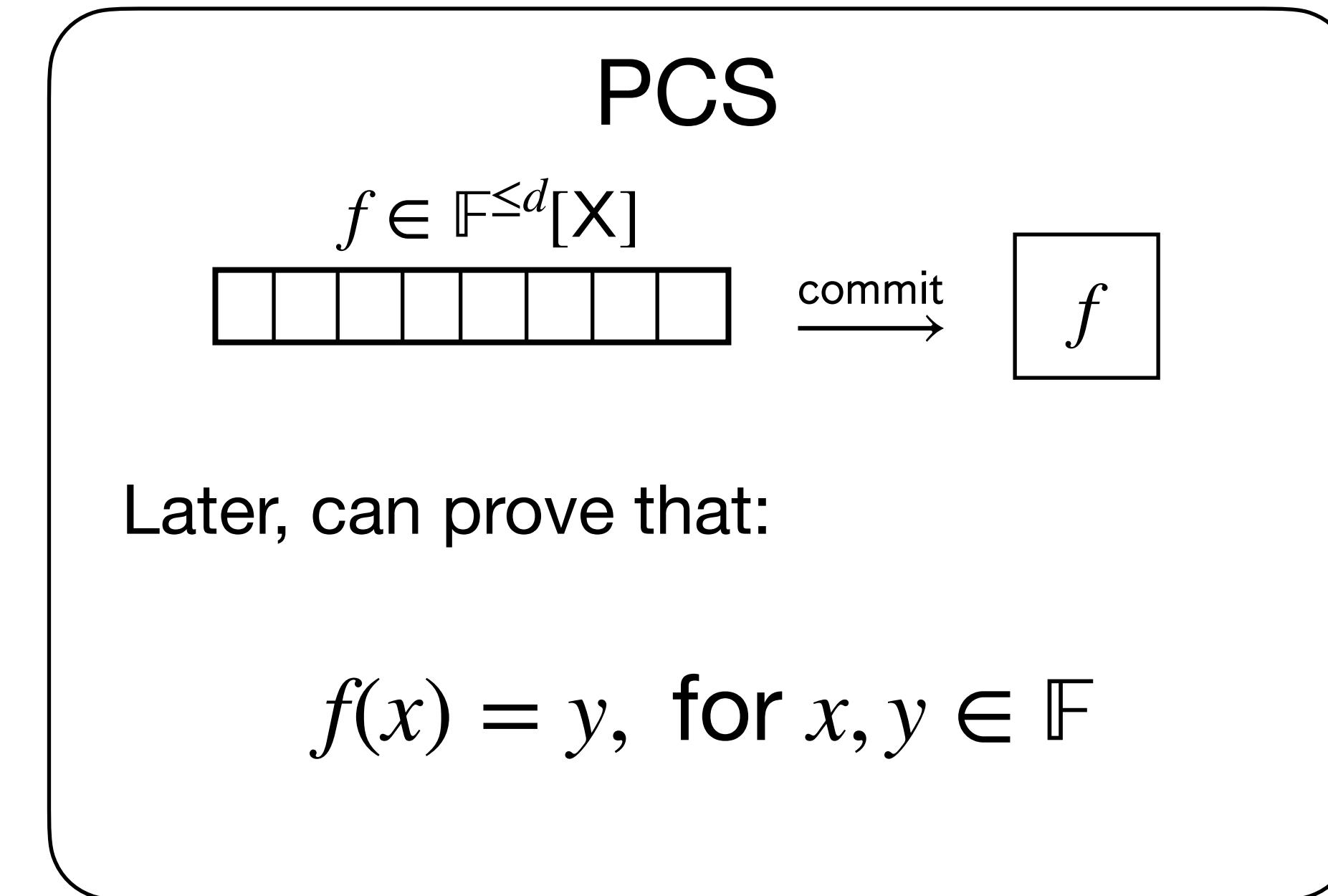
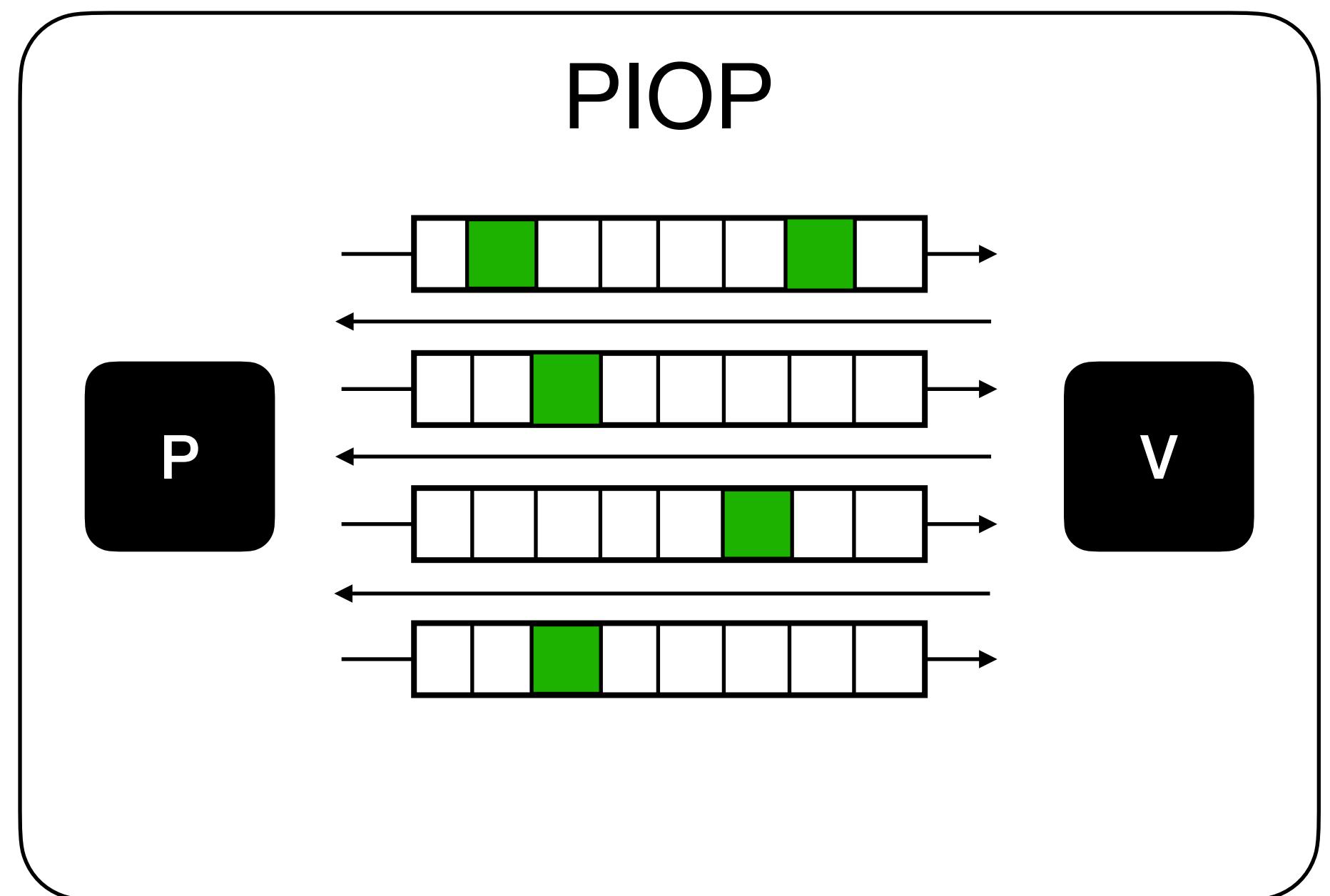
**Succinct:**  $\pi \ll w$  and verifier is fast.

**Knowledge Sound:** if  $V(x, \pi) = 1$ , can extract  $w$  such that  $(x, w) \in R$

# Constructing SNARKs

# The modular way™

# **We focus on this!**



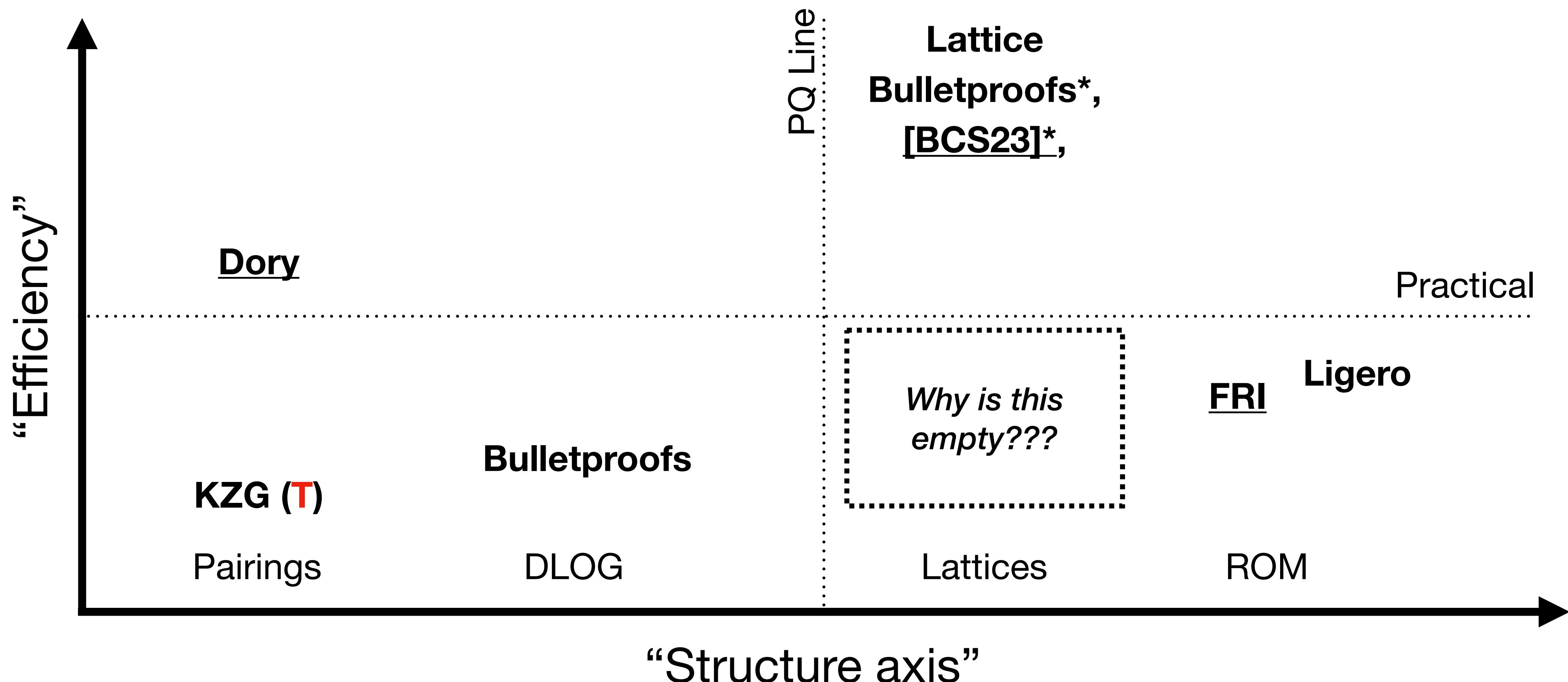
- Oracles are polynomials
  - Security is information-theoretical
  - Proof length is  $\Omega(n)$  (not succinct)
  - Verifiers are very efficient

- Cryptography goes here!
  - Computational security
  - We can achieve succinctness

# Zoo of Polynomial Commitments

A very incomplete list...

Underlined: succinct verification  
\*: interactive (no FS)  
(T): trusted setup



# Our Results

# SLAP: Succinct Lattice-Based Polynomial Commitments from Standard Assumptions

Martin R. Albrecht

[martin.albrecht@kcl.ac.uk](mailto:martin.albrecht@kcl.ac.uk), [sandboxaq.com](mailto:sandboxaq.com)

King's College London and SandboxAQ

Giacomo Fenzi

[giacomo.fenzi@epfl.ch](mailto:giacomo.fenzi@epfl.ch)

EPFL

Oleksandra Lapiha

[sasha.lapiha.2021@live.rhul.ac.uk](mailto:sasha.lapiha.2021@live.rhul.ac.uk)

Royal Holloway, University of London

Ngoc Khanh Nguyen

[khanh.nguyen@epfl.ch](mailto:khanh.nguyen@epfl.ch)

EPFL

We construct a non-interactive lattice-based polynomial commitment with:

1. Succinct proofs
2. Succinct verification time
3. Binding under (M)SIS



# Techniques

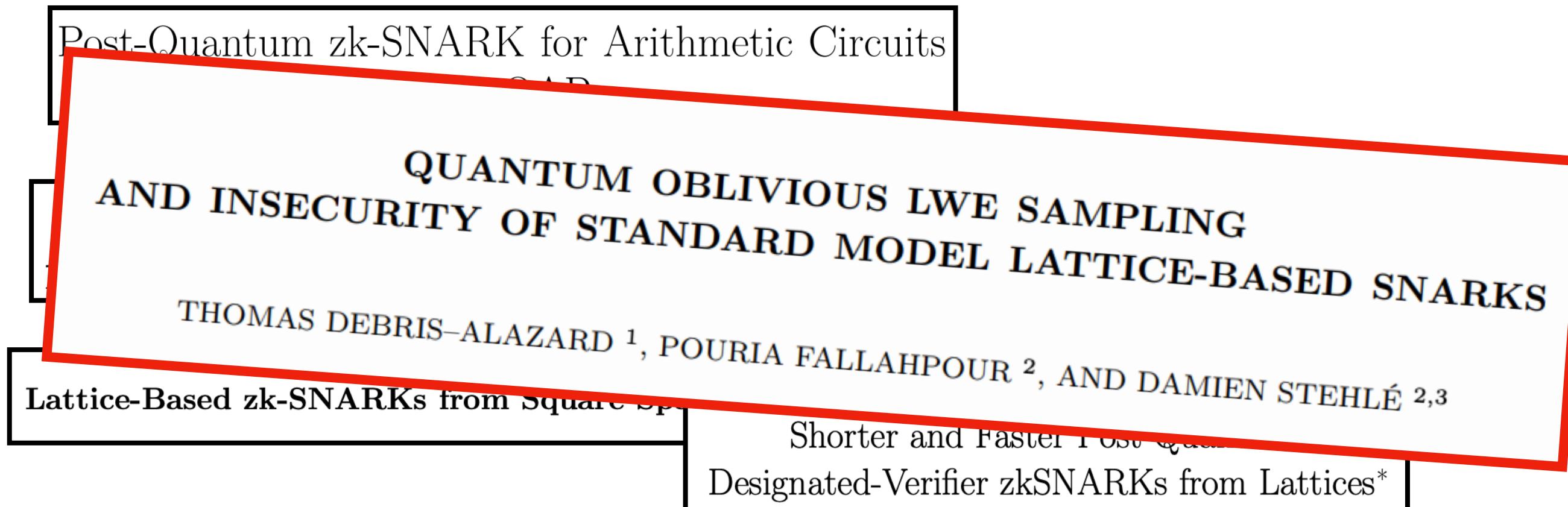
# Lattice-Based SNARKs

## How to get around [GW11]?

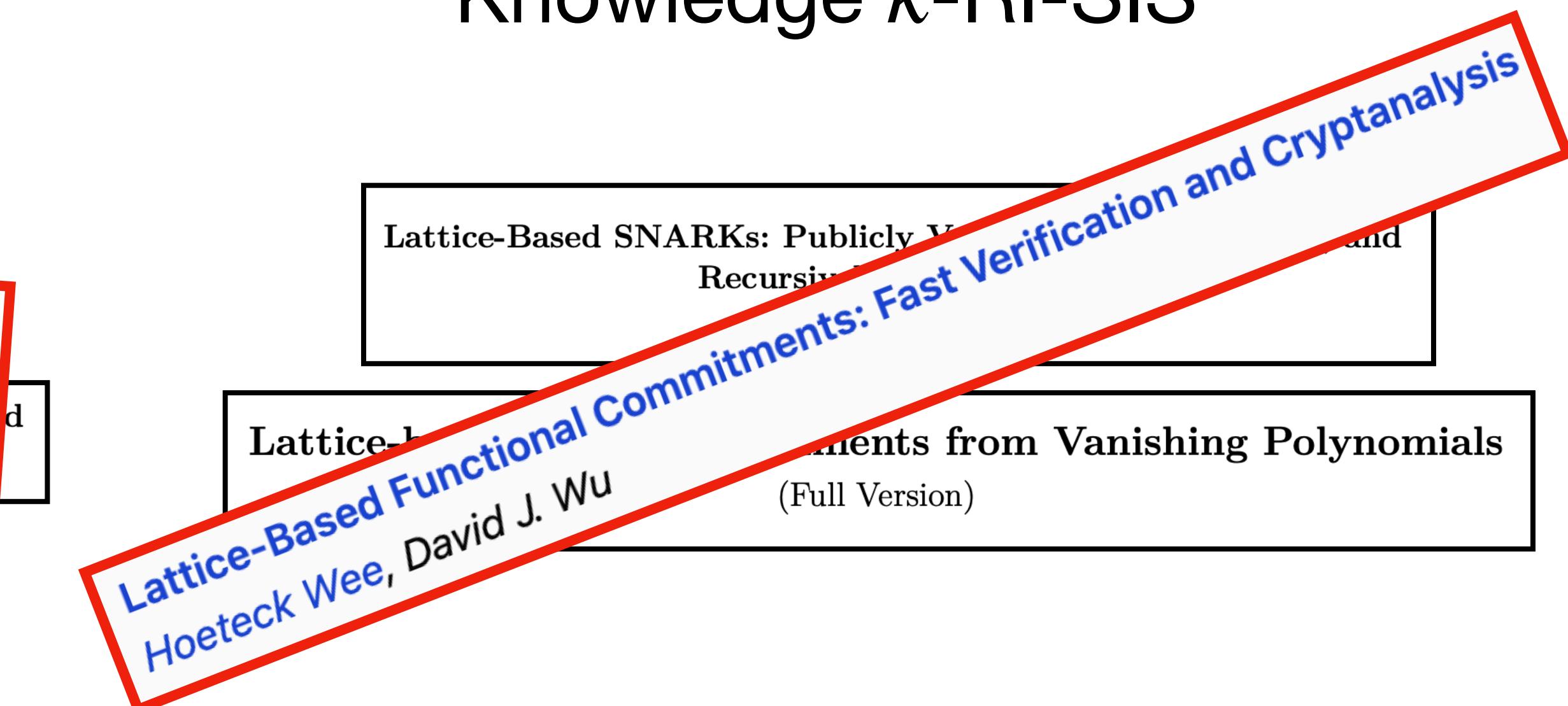
[GW11] - You cannot get **SNARG** from falsifiable assumptions.

## Knowledge Assumptions

### Oblivious LWE Sampling



### Knowledge $k$ -RI-SIS



# Lattice Assumptions ❤️ ROM

- Knowledge assumptions in “lattice-land”: hard to define and easy-ish to break
- ROM takes care of extraction and non-interactivity.



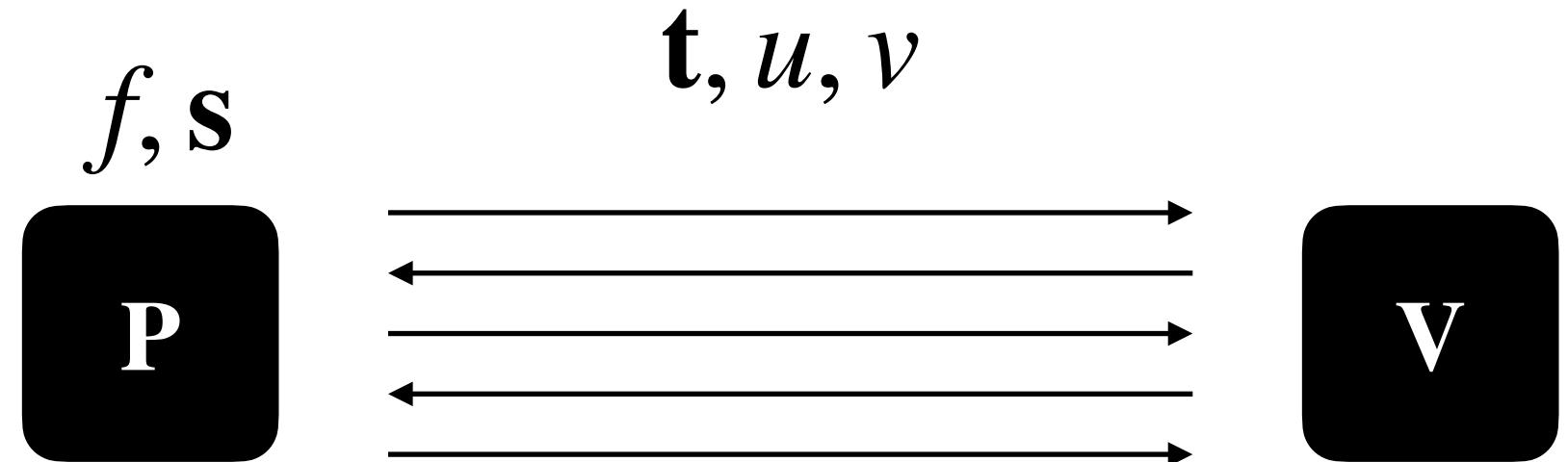
- Use lattices to get succinctness in the interactive protocol.
- **Open Question:** ROM alone is sufficient for efficient PCS (e.g. FRI), what do we gain by using lattices?

# Building succinct PCS

## Commitment Scheme

- Commit to a vector  $\mathbf{f} \in \mathcal{R}_q^d$
- Commitment  $\mathbf{t}$ , opening  $\mathbf{s}$
- Binding under lattice assumption

## Evaluation Protocol



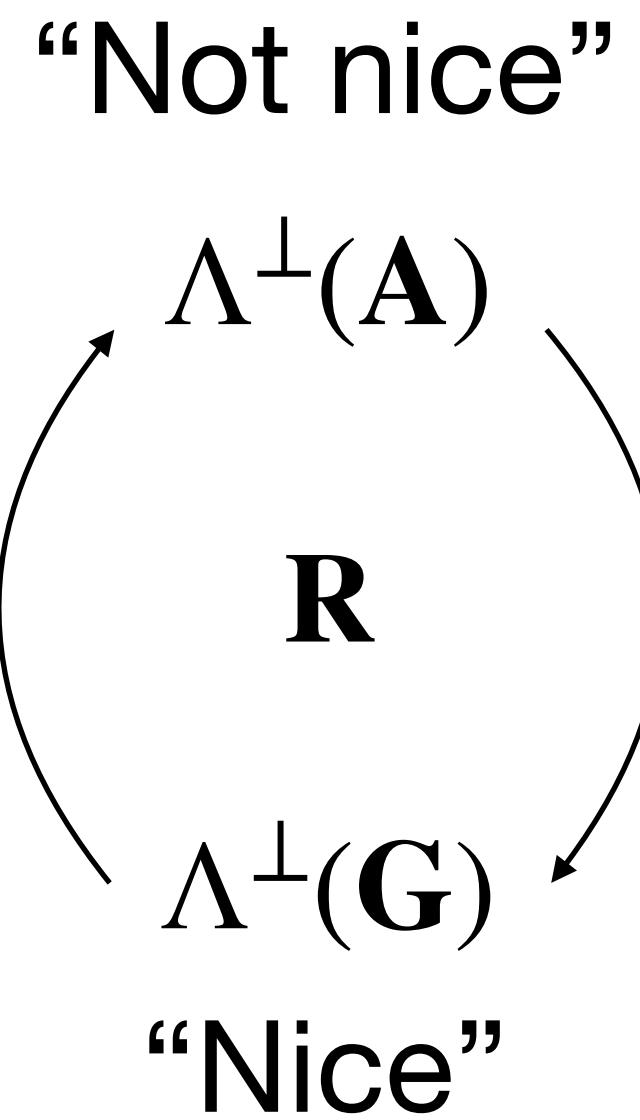
“I know  $f$  such that  $f(u) = v$  and an opening  $s$  for  $\mathbf{f} := \text{coeff}(f)$  to  $\mathbf{t}$ ”

- Need  $t \ll d$ , binding for  $\mathbf{f}$  of arbitrary norm
- Need communication complexity  $\ll d$

- Need V's running time to be  $\ll d$

# Trapdoors [MP12]

- Let  $\mathbf{G}$  be a “gadget matrix”
- Can sample  $(\mathbf{A}, \mathbf{R})$  such that  $\mathbf{A}\mathbf{R} = \mathbf{G}$ , with  $\mathbf{R}$  short.
- Given  $\mathbf{A}, \mathbf{R}, \mathbf{v}$ , can sample short  $\mathbf{s}$  such that  $\mathbf{A}\mathbf{s} = \mathbf{v}$ .

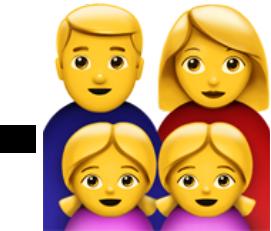


# Trapdoor Resampling [WW23]

- Given  $(\mathbf{A}, \mathbf{R})$ , can sample new trapdoor  $\mathbf{T}$  for some matrix  $\mathbf{B}$  “related” to  $\mathbf{A}$
- BASIS style assumption say:

“Given  $\mathbf{A}, \mathbf{B}, \mathbf{T}$ , hard to find short  $\mathbf{x}$  for  $\mathbf{A}\mathbf{x} = \mathbf{0}$ ”

# BASIS-[WW23]



```
SampSIS(A★)
return ⊥
```

## BASIS Game

$$A^* \leftarrow \mathcal{R}_q^{m \times n}$$

$$\text{aux} \leftarrow \text{Samp}(A^*)$$

return (A<sup>★</sup>, aux) to  $\mathcal{A}$

$\mathcal{A}$  wins if it finds  $x$ :

- $A^*x = 0$
- $0 < x \leq \beta$

Samp<sub>BASIS,ℓ</sub>(A<sup>★</sup>)

Sample  $\mathbf{a}, A_2, \dots, A_\ell$

$$A_1 := \begin{bmatrix} \mathbf{a}^\top \\ A^* \end{bmatrix}, B := \begin{bmatrix} A_1 & \cdots & -\mathbf{G} \\ \ddots & \ddots & \\ \cdots & A_d & -\mathbf{G} \end{bmatrix}$$

return ( $\mathbf{a}, (A_i)_i, B^{-1}(\mathbf{G})$ )

Samp<sub>PRISIS,ℓ</sub>(A<sup>★</sup>)

Sample  $\mathbf{a}, w$

$$A := \begin{bmatrix} \mathbf{a}^\top \\ A^* \end{bmatrix}, B := \begin{bmatrix} w^0 A & \cdots & -\mathbf{G} \\ \ddots & \ddots & \\ \cdots & w^{\ell-1} A & -\mathbf{G} \end{bmatrix}$$

return ( $\mathbf{a}, w, B^{-1}(\mathbf{G})$ )

# PRISIS Commitments I

A starting point [FMN23]

Given  $\mathbf{B} := \begin{bmatrix} w^0 \mathbf{A} & \dots & -\mathbf{G} \\ \ddots & \ddots & \vdots \\ \dots & w^{\ell-1} \mathbf{A} & -\mathbf{G} \end{bmatrix}$  and trapdoor  $\mathbf{T}$  for  $\mathbf{B}$

Use  $\mathbf{T}$  to sample short  $\mathbf{s}_0, \dots, \mathbf{s}_{\ell-1}, \hat{\mathbf{t}}$  such that:

$$\mathbf{B} \begin{bmatrix} \mathbf{s}_0 \\ \vdots \\ \mathbf{s}_{\ell-1} \\ \hat{\mathbf{t}} \end{bmatrix} = \begin{bmatrix} -f_0 w^0 \mathbf{e}_1 \\ \vdots \\ -f_{\ell-1} w^{\ell-1} \mathbf{e}_1 \end{bmatrix}$$

The commitment is  $\mathbf{t} := \mathbf{G}\hat{\mathbf{t}}$  and the openings are  $(\mathbf{s}_i)_i$ .

To open check that

$$\mathbf{A}\mathbf{s}_i + f_i \mathbf{e}_1 = w^{-i} \mathbf{t} \text{ and } \mathbf{s}_i \text{ short}$$

# PRISIS Commitments II

Pros  and Cons 

- Commitment is **succinct**.
- Supports committing to messages of **arbitrary** size.
- Algebraic structure enables **efficient evaluation protocol**.
- Binding under **non-standard** PRISIS assumption.
- Time to commit is **quadratic**.
- Common reference string is **quadratic**.
- **Trusted** setup

Can we do better?

# Small-Dimension PRISIS

[FMN23]:  $\ell = 2$  reduces to MSIS

**Lemma 3.6** (PRISIS  $\implies$  MSIS). *Let  $n > 0, m \geq n$  and denote  $t = (n+1)\tilde{q}$ . Let  $q = \omega(N)$ . Take  $\epsilon \in (0, 1/3)$  and  $\mathfrak{s} \geq \max(\sqrt{N \ln(8Nq)} \cdot q^{1/2+\epsilon}, \omega(N^{3/2} \ln^{3/2} N))$  such that  $2^{10N}q^{-\lfloor \epsilon N \rfloor}$  is negligible. Let*

$$\sigma \geq \delta \sqrt{tN \cdot (N^2 \mathfrak{s}^2 m + 2t)} \cdot \omega(\sqrt{N \log nN}).$$

*Then, PRISIS <sub>$n, m, N, q, 2, \sigma, \beta$</sub>  is hard under the MSIS <sub>$n, m, N, q, \beta$</sub>  assumption.*

# Multi-Instance BASIS

## $h$ -instance BASIS Game

$$\mathbf{A}_1^\star, \dots, \mathbf{A}_h^\star \leftarrow \mathcal{R}_q^{m \times n}$$

$$\text{aux}_i \leftarrow \text{Samp}(\mathbf{A}_i^\star) \text{ for } i \in [h]$$

return  $((\mathbf{A}_i^\star, \text{aux}_i)_i)$  to  $\mathcal{A}$

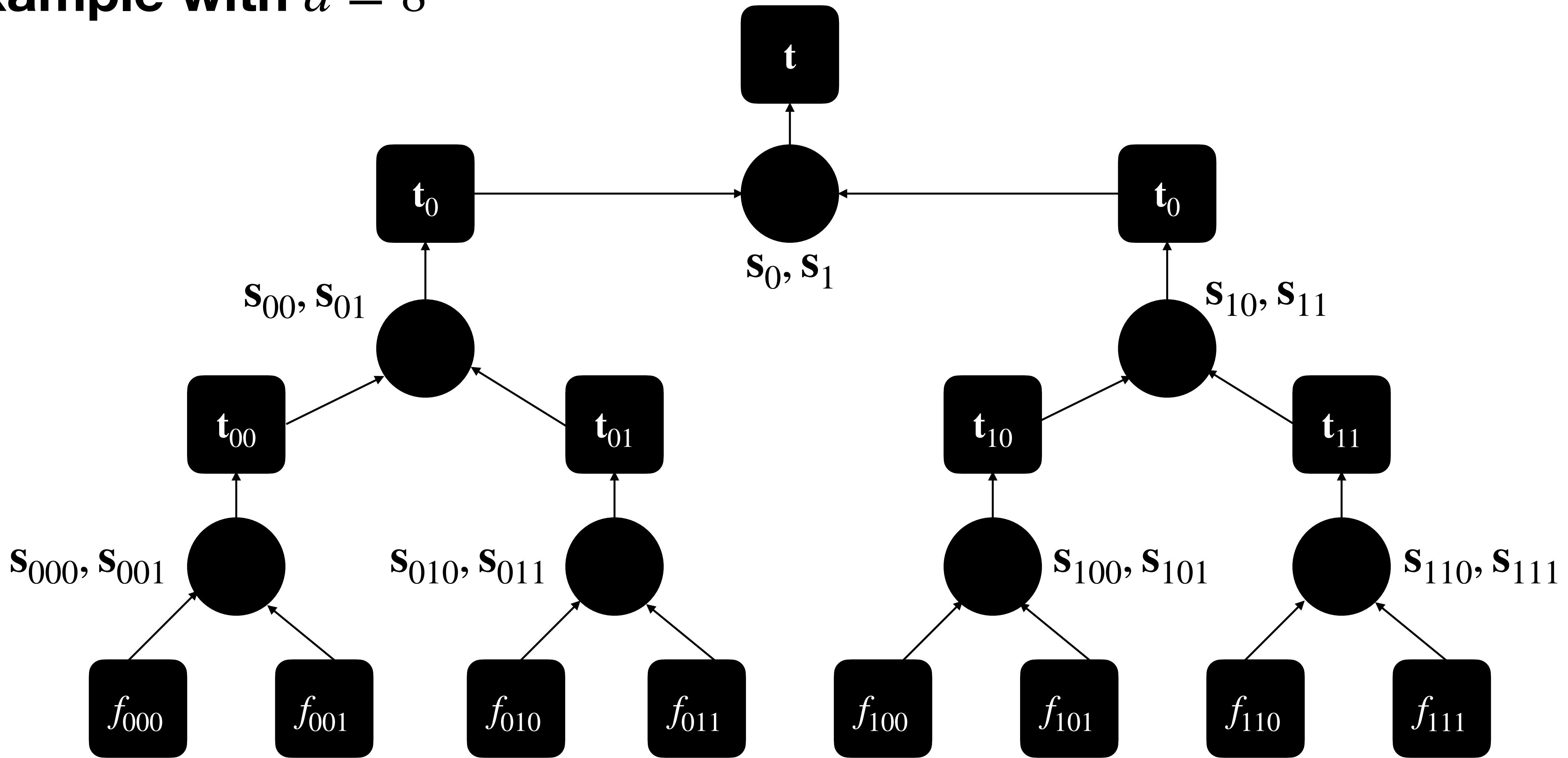
$\mathcal{A}$  wins if it finds  $\mathbf{x}$ :

- $[\mathbf{A}_1^\star, \dots, \mathbf{A}_h^\star] \cdot \mathbf{x} = 0$
- $0 < \mathbf{x} \leq \beta$

For  $\ell = O(1)$ , if PRISIS <sub>$\ell$</sub>  is hard so is  $h$ -PRISIS <sub>$\ell$</sub> !

# Merkle-PRISIS I

Example with  $d = 8$



# Merkle-PRISIS II

## How to check an opening

- Each layer has its own  $\text{crs}_j := (\mathbf{A}_j, w_j, \mathbf{T}_j)$  for  $j \in [h := \log d]$
- Check that all local openings are correct. I.e. check that, for  $\mathbf{b} \in \{0,1\}^h$ :

$$\sum_{j \in [h]} w_j^{b_j} \mathbf{A}_j \mathbf{s}_{\mathbf{b}:j} + f_{\mathbf{b}} \cdot \mathbf{e} = \mathbf{t}$$

- And, of course, that all the openings  $\mathbf{s}_{\mathbf{b}}$  are short for  $\mathbf{b} \in \{0,1\}^{\leq h}$
- **Binding:** subtract two verification equation:

reduces to  $h$ -PRISIS $_{\ell}$  i.e. **MSIS!**

# Merkle-PRISIS III

Pros  and Cons 

- Commitment is **succinct**.
- Supports committing to messages of **arbitrary** size.
- Time to commit is **quasi-linear**.
- Common reference string is **logarithmic**.
- Binding under **standard** SIS assumption.
- **Trusted** setup

Can we do an efficient evaluation protocol?

# Evaluation Protocol I

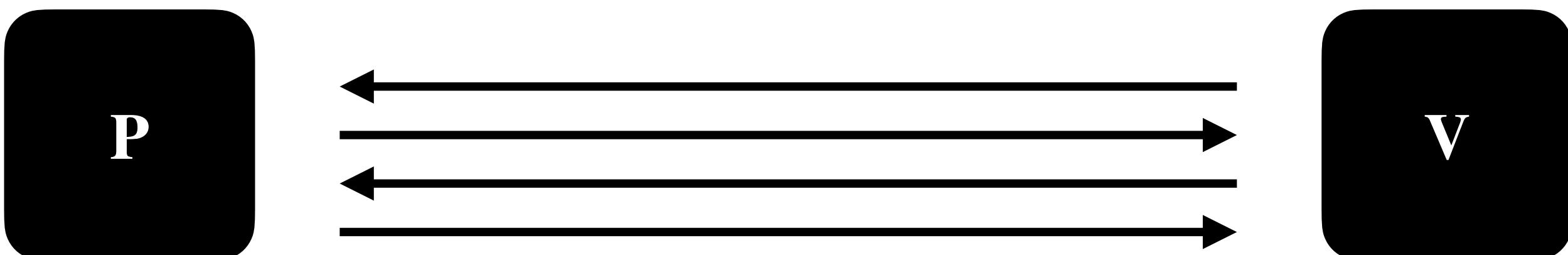
## Strategy

**Prover** knows:

- Polynomial  $f \in \mathcal{R}_q^{<d}[X]$  and openings  $(s_b)_b$

**Verifier** knows:

- Common reference string crs
- Commitment  $t$
- Claim:  $f(u) = v$  and  $\text{Open}(\text{crs}, t, f, (s_b)_b) = 1$



**Prover** now knows:

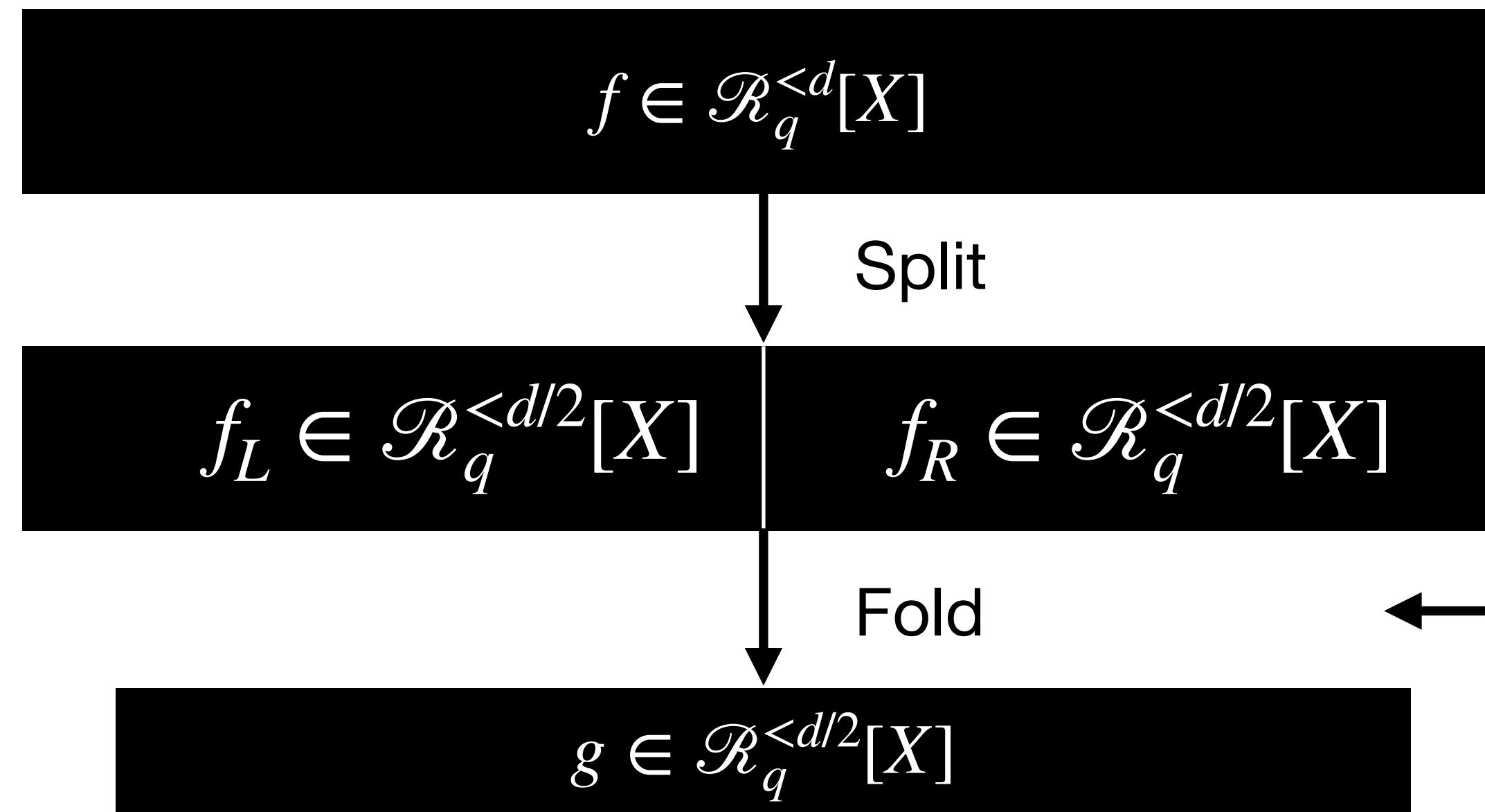
- Polynomial  $g \in \mathcal{R}_q^{<d/2}[X]$  and openings  $(z_b)_b$

**Verifier** now knows:

- Common reference string  $\text{crs}'$
- Commitment  $t'$
- New claim:  $g(u') = v'$  and  $\text{Open}(\text{crs}', t', g, (z_b)_b) = 1$

# Evaluation Protocol II

## Split and fold (Evaluations)



*Fast Reed-Solomon Interactive Oracle Proofs of Proximity*  
Eli Ben-Sasson\* Iddo Bentov† Ynon Horesh\* Michael Riabzev\*

$$f(X) = f_L(X^2) + X \cdot f_R(X^2)$$

$$\alpha_0, \alpha_1 \rightarrow V$$

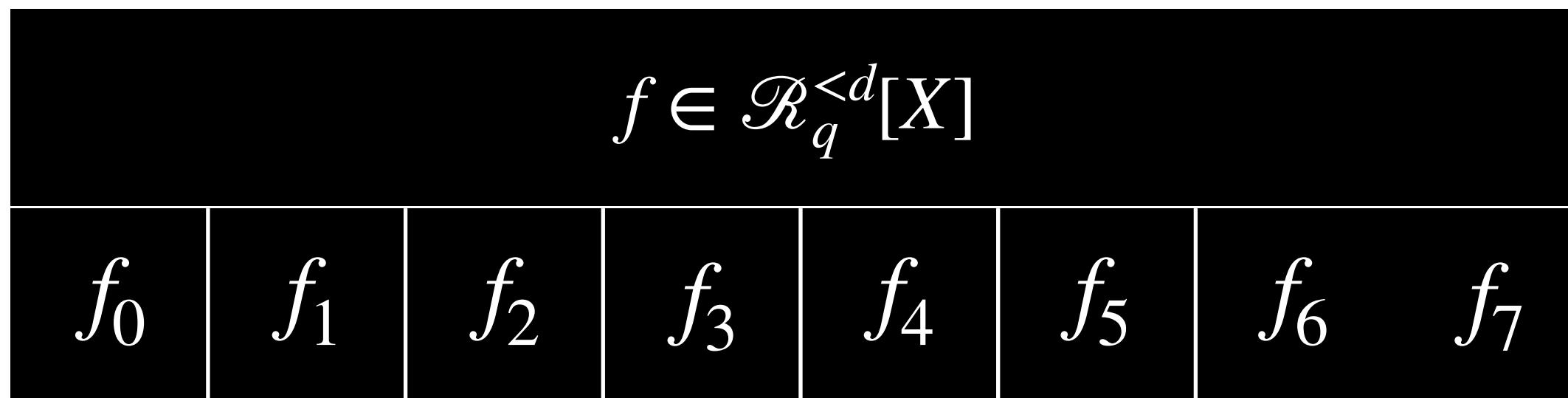
$$g(X) = \alpha_0 f_L(X) + \alpha_1 f_R(X)$$

Ask prover to send  $z_0 = f_L(u^2), z_1 = f_R(u^2)$ . Check  $z_0 + uz_1 = z$

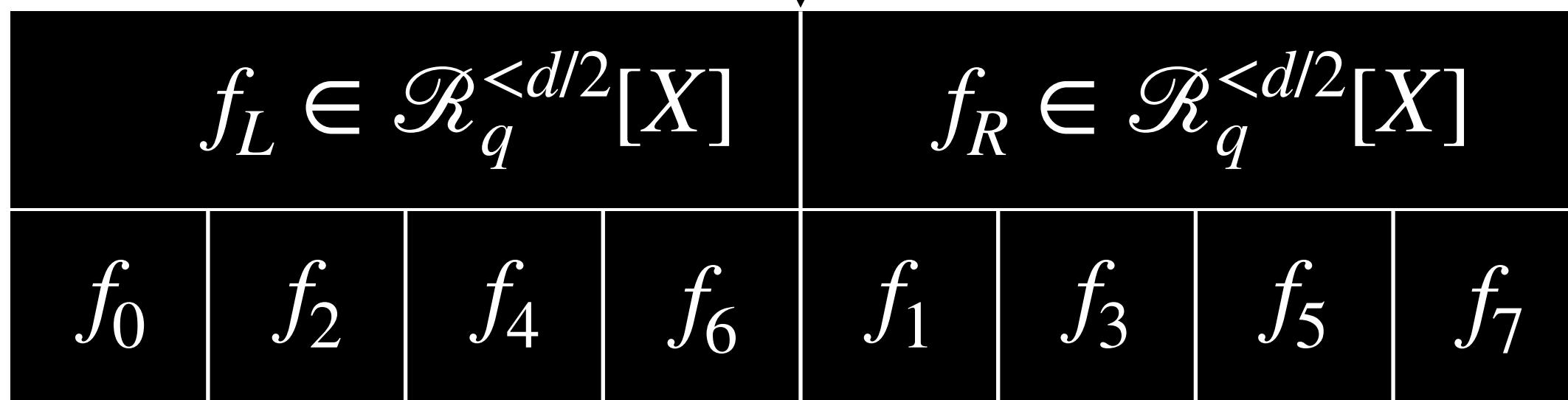
If  $f(u) = v$ , then  $g(u^2) = \alpha_0 z_0 + \alpha_1 z_1$ .

# Evaluation Protocol III

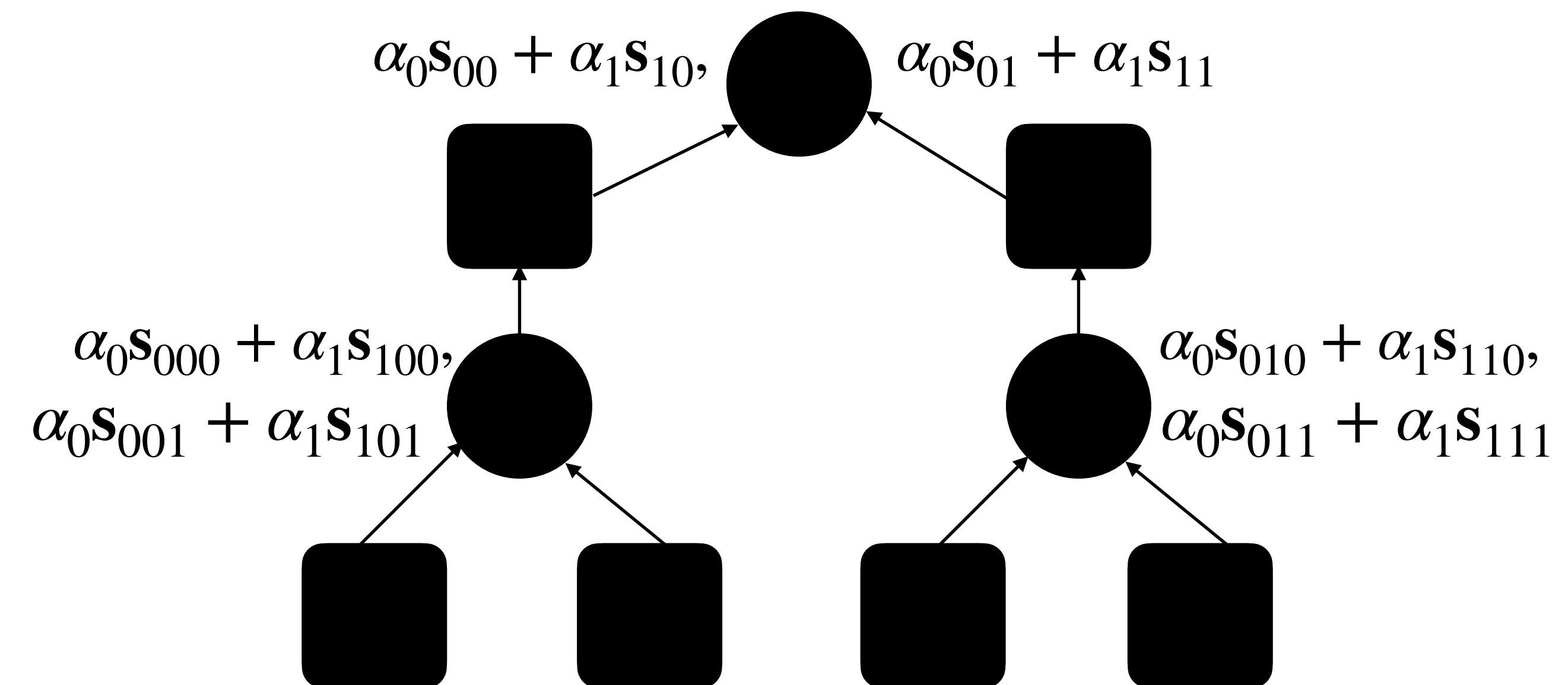
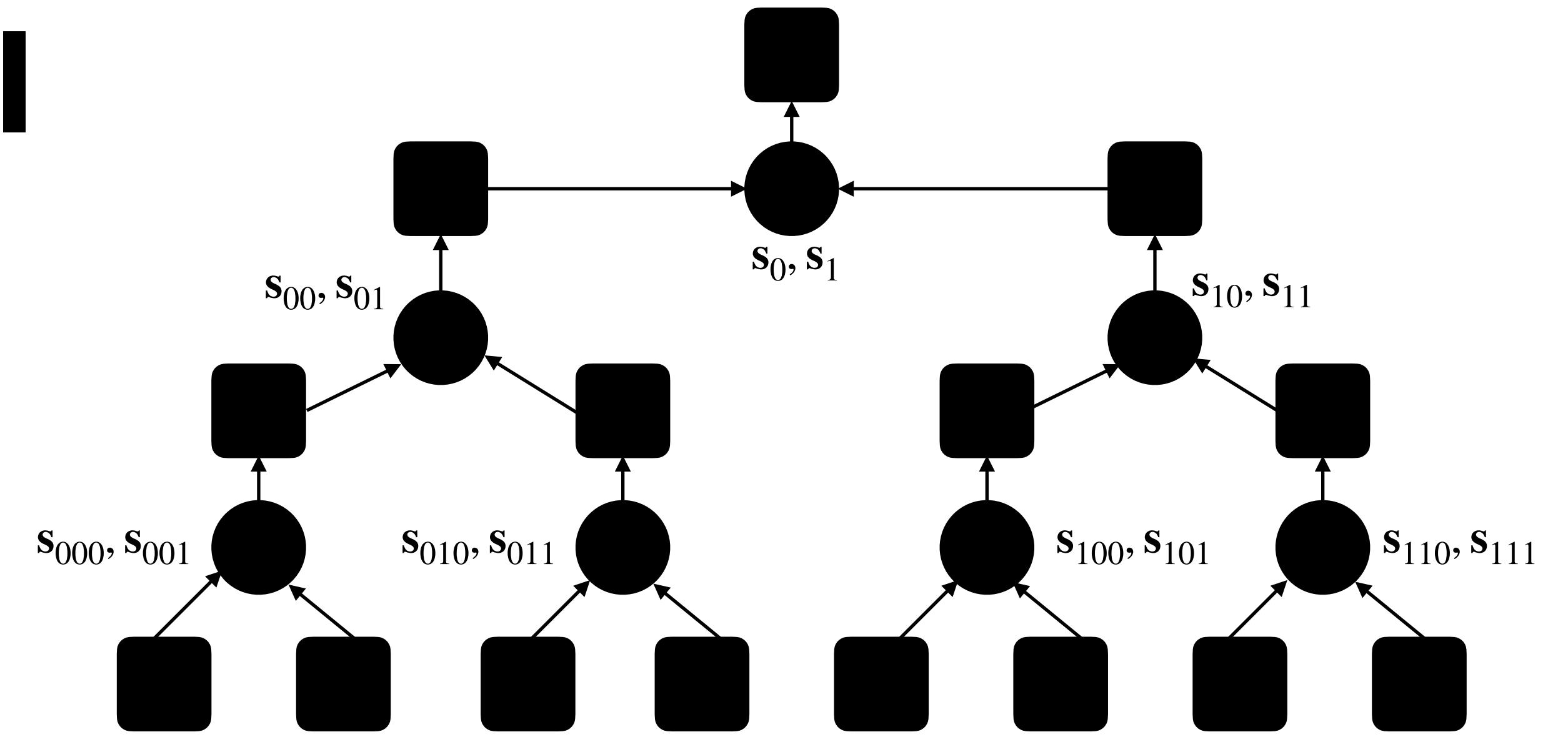
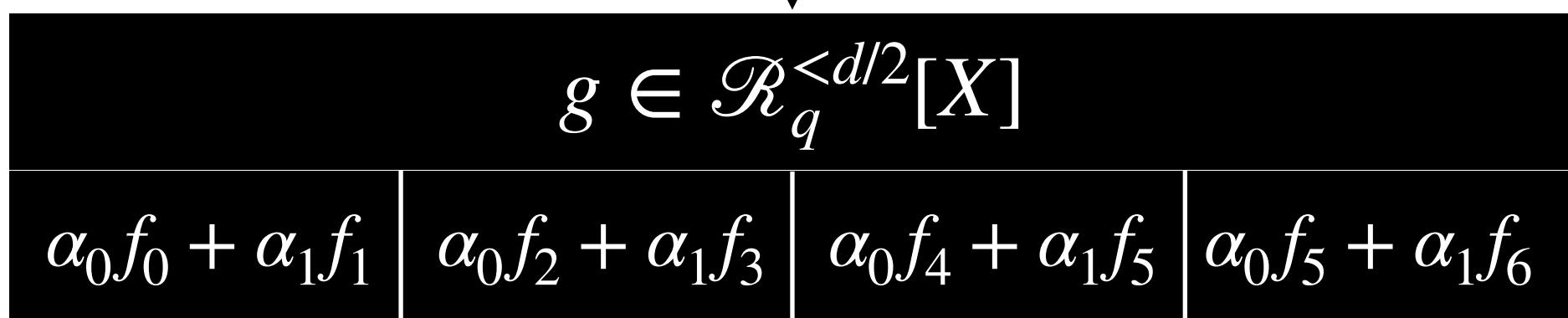
## Split and fold (Openings)



Split



Fold



# Evaluation Protocol IV

## Split and fold (Commitment)

- We have shown how to compute new evaluations and openings
- If  $\alpha_i$  are short, the new openings also are.
- How does the verifier compute new commitment? With some magic:

$$\sum_{j \in [h-1]} w_{1+j}^{b_{1+j}} \mathbf{A}_{1+j} \mathbf{s}_{\mathbf{b}:1+j} + g_{\mathbf{b}} \mathbf{e} = \alpha_0 \cdot (\mathbf{t} - w_1^0 \mathbf{A}_1 \mathbf{s}_0) + \alpha_1 \cdot (\mathbf{t} - w_1^1 \mathbf{A}_1 \mathbf{s}_1)$$

- Prover reveals  $\mathbf{s}_0, \mathbf{s}_1$ . Verifier sets RHS as new updated commitment.

# Evaluation Protocol V

## Putting it all together

### Basic $\Sigma$ -Protocol

#### Prover

$$f(\mathbf{X}) = f_0(\mathbf{X}^2) + \mathbf{X}f_1(\mathbf{X}^2)$$

$$z_i := f_i(u^2) \text{ for } i \in \mathbb{Z}_2$$

$$g(\mathbf{X}) := \alpha_0 f_0(\mathbf{X}) + \alpha_1 f_1(\mathbf{X})$$

$$\mathbf{z}_\mathbf{b} := \alpha_0 \mathbf{s}_{\mathbf{b},0} + \alpha_1 \mathbf{s}_{\mathbf{b},1} \text{ for } \mathbf{b} \in \mathbb{Z}_2^{\leq h-1}$$

#### Verifier

$\xrightarrow{z_0, z_1, \mathbf{s}_0, \mathbf{s}_1}$  Check:  $z_0 + uz_1 =? z$ ; Check:  $\mathbf{s}_0, \mathbf{s}_1$  short

$$\alpha_0, \alpha_1 \leftarrow \{ X^i : i \in \mathbb{Z} \}$$

$$\text{crs}' := (\mathbf{A}_{1+t}, w_{1+t}, \mathbf{T}_{1+t})_{t \in [h-1]}$$

$$\mathbf{t}' := \alpha_0 \cdot (\mathbf{t} - w_1^0 \mathbf{A}_1 \mathbf{s}_0) + \alpha_1 \cdot (\mathbf{t} - w_1^1 \mathbf{A}_1 \mathbf{s}_1)$$

$$u' := u^2; z' := \alpha_0 \cdot z_0 + \alpha_1 \cdot z_1$$

$$\text{Check: } g(u') = z'$$

$$\text{Check: } \text{Open}(\text{crs}', \mathbf{t}', g, (\mathbf{z}_\mathbf{b})_\mathbf{b}) = 1$$

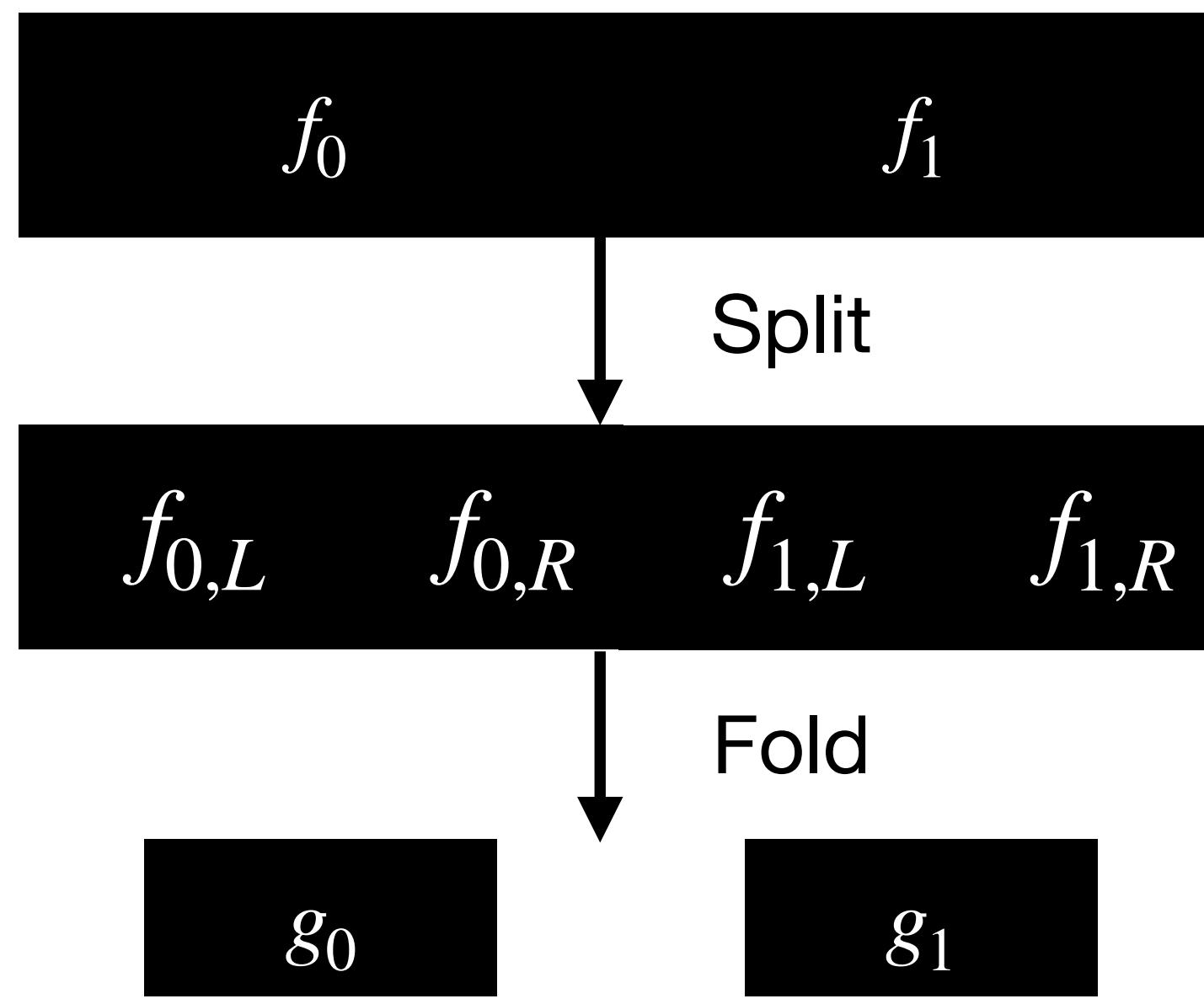
# Are we done?

- Apply protocol recursively  $\log d$  times and send final opening  $O(1)$ .
- Knowledge soundness follows from **coordinate-wise special soundness**.
- Commitment is **succinct**, verifier also **succinct**.
- **Problem** 🤔: Knowledge soundness error is  $1/\text{poly}(\lambda)$ .
- Can be made negligible by parallel repetition, but then **no Fiat-Shamir!**
- Change the challenge space?
  - Non-subtractive challenge space => Blowup in extraction, cannot do more than  $\log \log d$  recursions => only **quasi-polylogarithmic** sizes.
  - Subtractive challenge space => Challenge space of size at most  $\text{poly}(\lambda)$  [AL21]

# Claim bundling I

Let's prove something harder!

- Instead of proving  $f(u) = v$ , show that, for  $\iota \in [r]$ ,  $f_\iota(u) = v_\iota$
- As in [FMN23], our protocol can be easily extended to deal with this.



Randomness is now:

$$\begin{bmatrix} \alpha_{0,L,0}, \alpha_{0,R,0}, \alpha_{1,L,0}, \alpha_{1,R,0} \\ \alpha_{0,L,1}, \alpha_{0,R,1}, \alpha_{1,L,1}, \alpha_{1,R,1} \end{bmatrix} \in (\mathcal{C}^r)^{2r} \quad \begin{matrix} \alpha_{\iota,i,\kappa} \text{ folds } f_{\iota,i} \\ \text{into } g_\kappa \end{matrix}$$

Folded polynomial:

$$g_0 := \alpha_{0,L,0}f_{0,L} + \alpha_{0,R,0}f_{0,R} + \alpha_{1,L,0}f_{1,L} + \alpha_{1,R,0}f_{1,R}$$

$$g_1 := \alpha_{0,L,1}f_{0,L} + \alpha_{0,R,1}f_{0,R} + \alpha_{1,L,1}f_{1,L} + \alpha_{1,R,1}f_{1,R}$$

# Claim bundling II

## What did we gain?

- Now, protocol is  $2r$  coordinate-wise special sound with challenge space of size roughly  $\text{poly}(\lambda)^r$
- Setting  $r$  to be  $\text{polylog}(\lambda)$ , we achieve **negligible knowledge error!**
- Our protocol can now be made **non-interactive** using FS.
- To prove a single claim  $f(u) = v$ , simply set  $f_1, \dots, f_r = f$  and  $v_1, \dots, v_r = v$ .

# Recap:

## What we talked about

- PRISIS and Merkle-PRISIS commitments
- Multi-instance PRISIS assumptions
- $h\text{-PRISIS}_2$  reduces to MSIS
- Succinct evaluation protocol for Merkle-PRISIS
- Boosting soundness via claim bundling

# There is more!

## What we did not talk about

- Folding more at each step
- Coordinate-wise special soundness
- Honest-verifier zero knowledge for our PCS
- Transforming PCS for  $\mathcal{R}_q$  in those for  $\mathbb{Z}_q$  (efficient packing)
- Twin- $k$ - $M$ -ISIS is no easier than  $2k$ - $M$ -ISIS
- Setting concrete parameters
- Reductions... all the reductions

# Conclusion



# - SLAP

A non-interactive lattice-based  
polynomial commitment with succinct  
proofs and verification time, from  
standard lattice assumptions.

# Open Questions



- Can we get succinct lattice-based polynomial commitments under **100KB**?
- Can we get  $\text{negl}(\lambda)$  knowledge error in one-shot (no claim bundling)?
- Is  $\text{PRISIS}_\ell$  with  $\ell > 2$  still secure?



# Thank you!