FINDING A NASH EQUILIBRIUM IS NO EASIER THAN BREAKING FIAT-SHAMIR

ARKA RAI CHOUDHURI	JOHNS HOPKINS UNIVERSITY
PAVEL HUBÁČEK	CHARLES UNIVESITY IN PRAGUE
CHETHAN KAMATH	IST AUSTRIA
KRZYSZTOF PIETRZAK	IST AUSTRIA
ALON ROSEN	IDC HERZLIYA
GUY ROTHBLUM	WEIZMANN INSTITUTE OF SCIENCE

STOC'2019, PHOENIX, AZ

Today

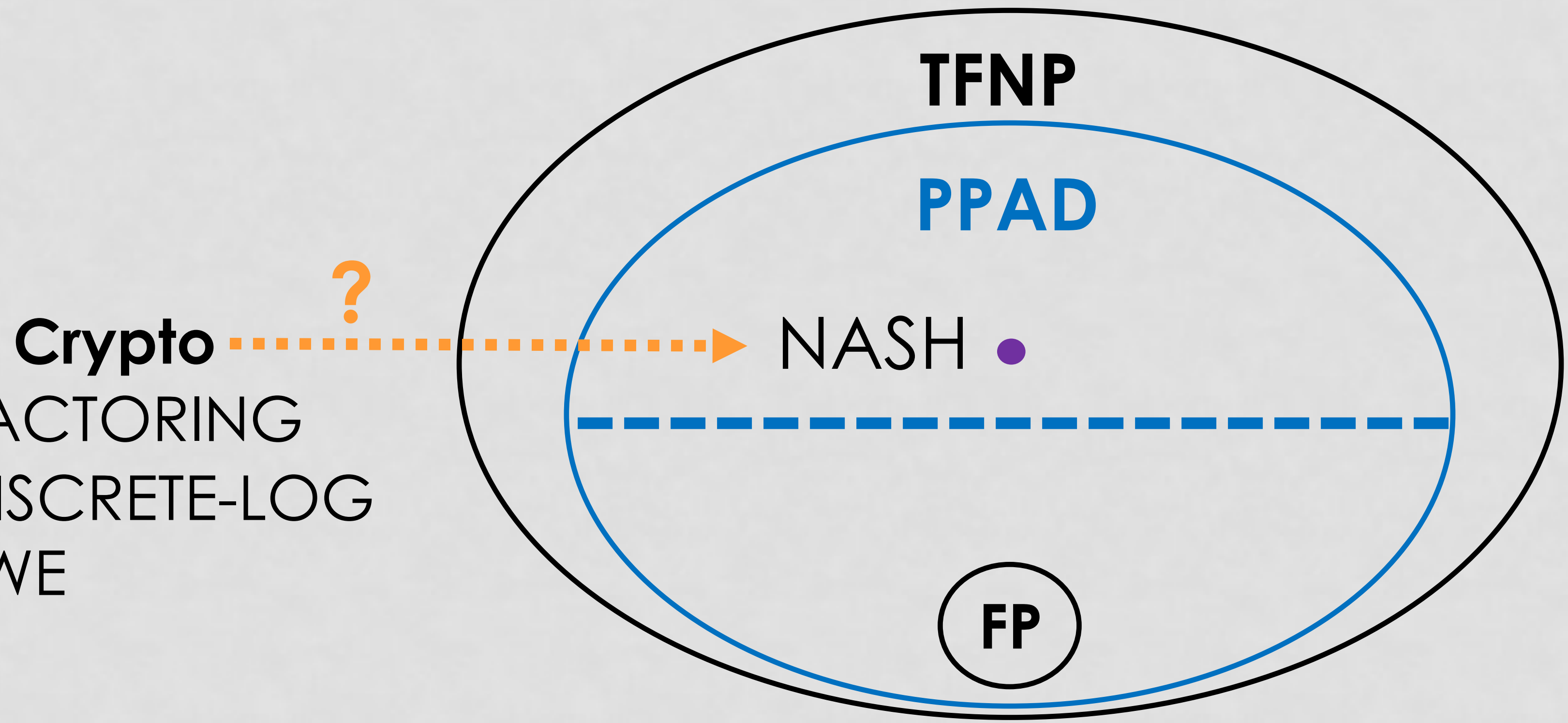
- Average-case hardness in PPAD

- Theorem: PPAD is as hard as breaking soundness of Fiat-Shamir when applied to the sumcheck protocol
- Corollary: Average-case hardness in PPAD relative to a random oracle

- Result extends to CLS $\subseteq$ PPAD

NASH and PPAD

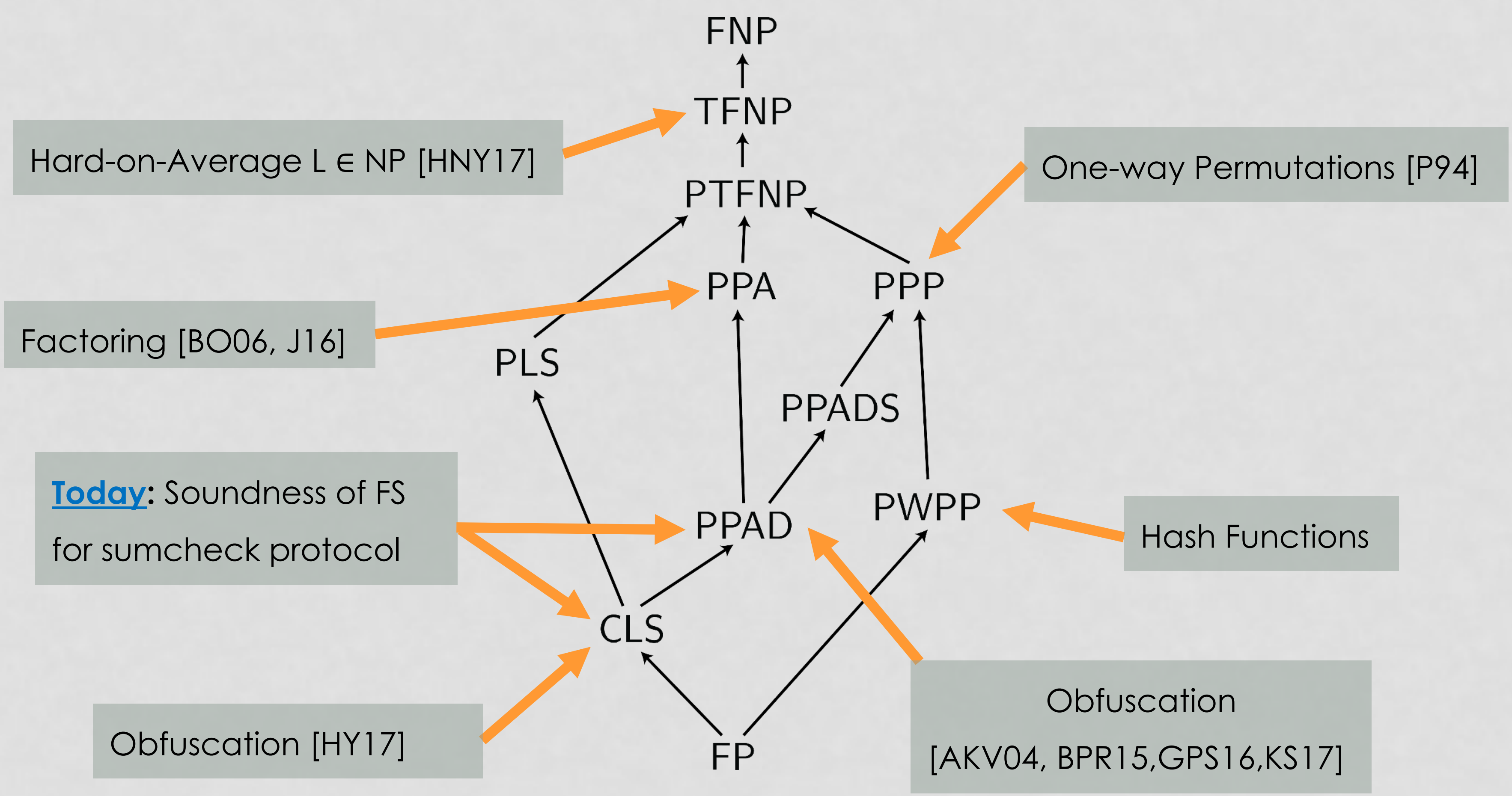
[P94,DGP05,CDT09]



- FACTORING
- DISCRETE-LOG
- LWE

- **T**otal **F**unctional **NP**
- Totality via "parity argument in directed graphs"

Average-case hardness in TFNP



PPAD-Hardness from Obfuscation

[BPR15]

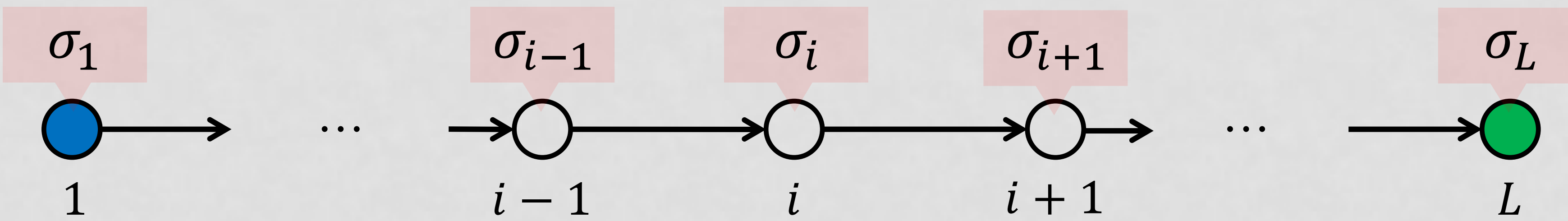
<u>Theorem</u>: Indistinguishability obfuscation (iO) implies hardness in PPAD/CLS

1. iO ⟶ SINK-OF-VERIFIABLE-LINE (SVL)
2. SVL ⟶ NASH (END−OF−LINE) [AKV04]

2.* SVL ⟶ END-OF-METERED-LINE ($\in$ CLS) [HY17]

<u>Bottom-line</u>: Focus on hard SVL instances

SINK-OF-VERIFIABLE-LINE (SVL)
[AKV04, BPR15]

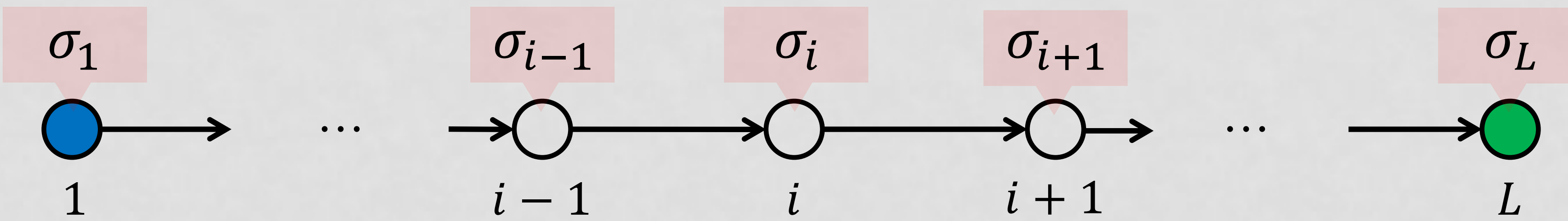


Exponential-sized graph with vertices in $\{0,1\}^n$

- Path defined by circuit $\boldsymbol{S}: \{0,1\}^n \rightarrow \{0,1\}^n$
- Verifier circuit $\boldsymbol{V}: [2^n] \times \{0,1\}^n \rightarrow \texttt{ACCEPT/REJECT}$
- Promise: $\boldsymbol{V}(i, \sigma_i) = \texttt{ACCEPT} \Leftrightarrow \sigma_i = \boldsymbol{S}^i(\sigma_1)$
- Solution: $\sigma_L = \boldsymbol{S}^L(\sigma_1)$

SVL IS NO EASIER THAN BREAKING FIAT-SHAMIR

SVL as Verifiable Counter for #SAT



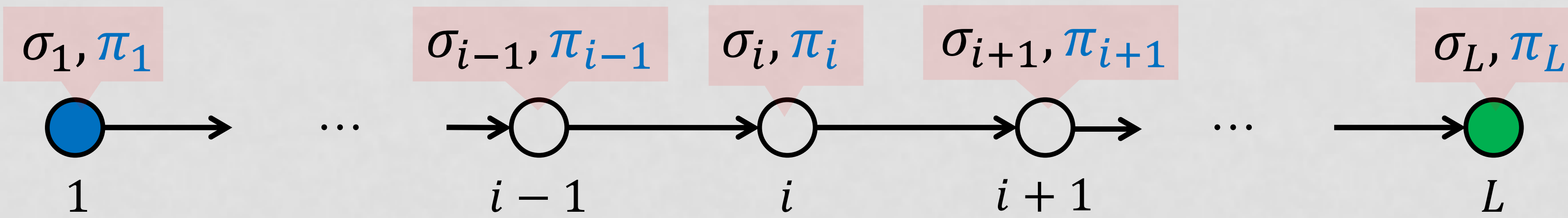
Reduce #SAT to SVL

- $\varphi(z_1, \ldots, z_n) \mapsto (\boldsymbol{S}, \boldsymbol{V}, L \leftarrow 2^n)$
- $\sigma_i \leftarrow$ # of satisfying assignments between 0^n and i

<u>Challenge</u>: How to verify σ_i?

<u>Solution</u>: Append a succinct proof π_i

SVL as Verifiable Counter for #SAT

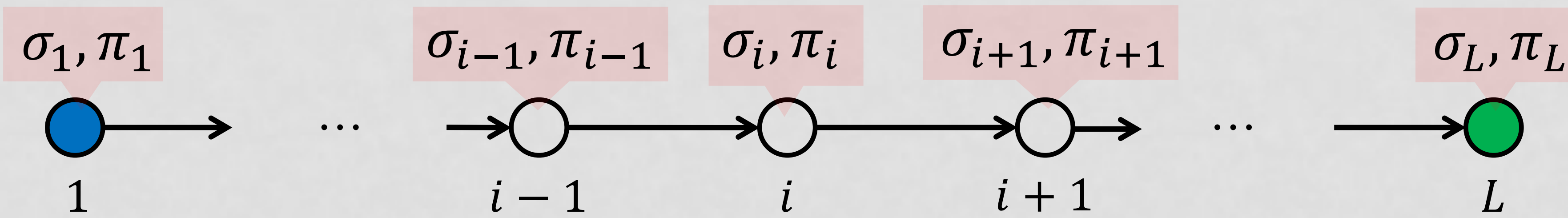


$$\boldsymbol{V}(i, \sigma_i, \pi_i) = \texttt{ACCEPT}$$

$$\Updownarrow$$

σ_i is the number of satisfying assignments between 0^n and i

SVL as Verifiable Counter for #SAT



Challenge: getting π_i to be of size $poly(n)$

Solution: use sumcheck protocol [LFKN92]

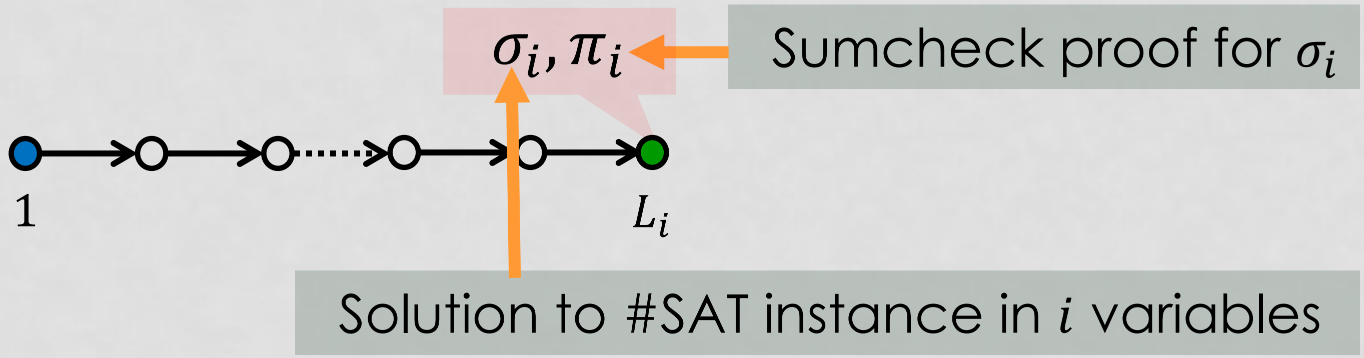
Challenge: protocol is interactive

Solution: Fiat-Shamir transform [FS86]

Challenge: computing $S(\sigma_i, \pi_i) = (\sigma_{i+1}, \pi_{i+1})$

Solution: incremental proof update

Recursive Approach



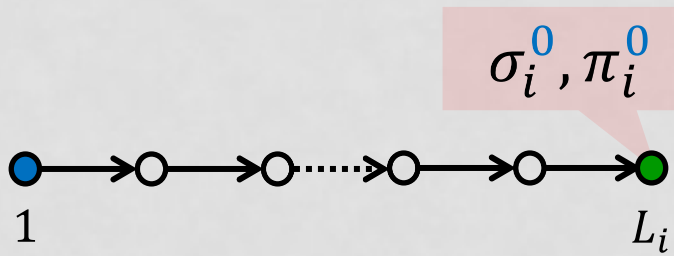
SVL counter $(\boldsymbol{S}_i, \boldsymbol{V}_i, L_i)$ for $\varphi(z_1, \ldots, z_i)$

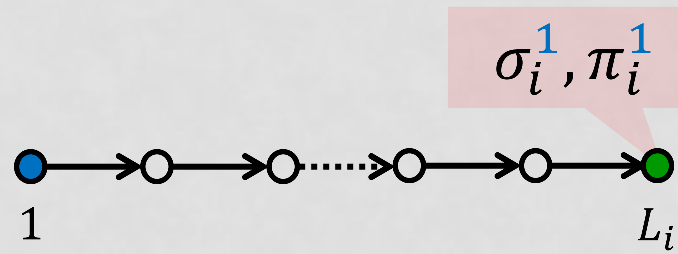
$$\Downarrow$$

SVL counter $(\boldsymbol{S}_{i+1}, \boldsymbol{V}_{i+1}, L_{i+1})$ for $\varphi(z_1, \ldots, z_i, z_{i+1})$

Base case: Length one, with empty proof

Naïve Construction

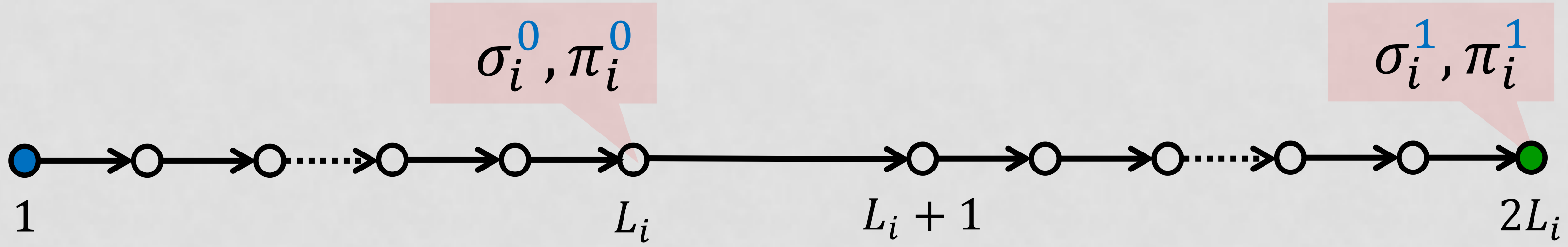




Counter for $\varphi(z_1, \ldots, z_i, z_{i+1})$

- Left path: Run counter on $\varphi(z_1, \ldots, z_i, 0)$
- Right path: Run counter on $\varphi(z_1, \ldots, z_i, 1)$

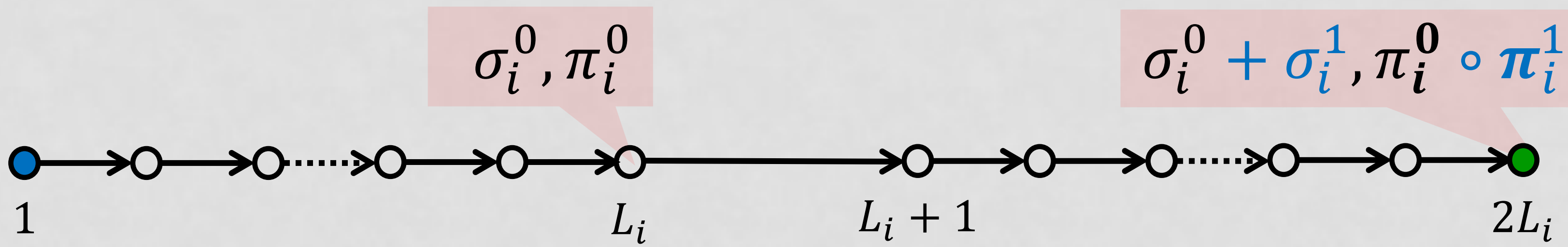
Naïve Construction



Counter for $\varphi(z_1, \ldots, z_i, z_{i+1})$

- Left path: Run counter on $\varphi(z_1, \ldots, z_i, 0)$
- Right path: Run counter on $\varphi(z_1, \ldots, z_i, 1)$

Naïve Construction



Counter for $\varphi(z_1, \dots, z_i, z_{i+1})$

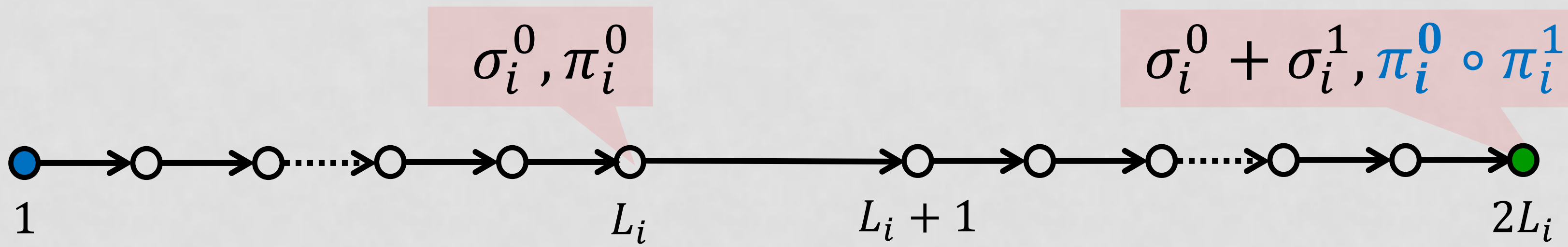
- Left path: Run counter on $\varphi(z_1, \dots, z_i, 0)$
- Right path: Run counter on $\varphi(z_1, \dots, z_i, 1)$ + update

Number of steps: $L_{i+1} = 2L_i$

Proof size: $P_{i+1} = 2P_i \implies P_n = 2^n$

Issue: exponential blow-up in proof/label size

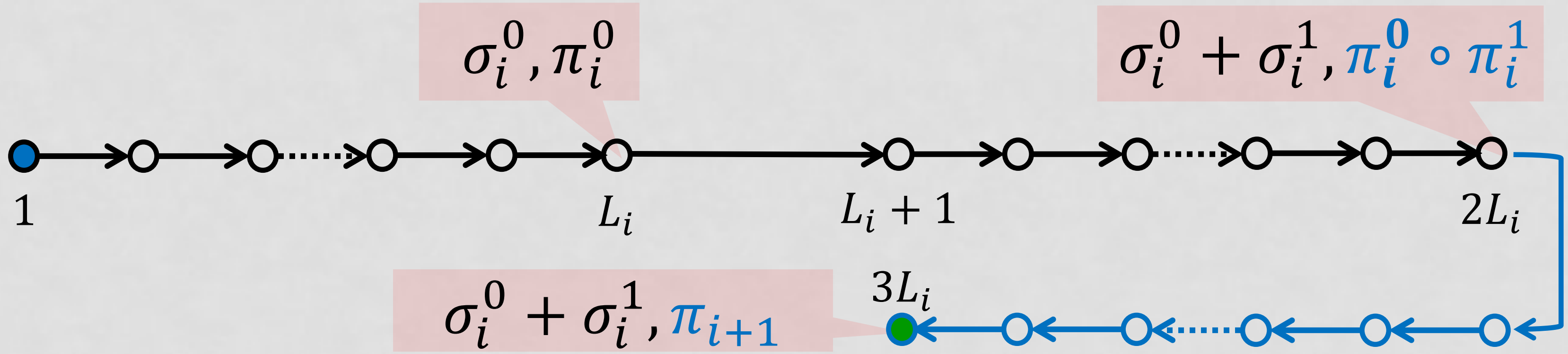
New Idea: Incremental Merge



Counter for $\varphi(z_1, \ldots, z_i, z_{i+1})$:

- Left path: Run counter on $\varphi(z_1, \ldots, z_i, 0)$
- Right path: Run counter on $\varphi(z_1, \ldots, z_i, 1)$ + updates

New Idea: Incremental Merge



Counter for $\varphi(z_1, \ldots, z_i, z_{i+1})$:

- Left path: Run counter on $\varphi(z_1, \ldots, z_i, 0)$
- Right path: Run counter on $\varphi(z_1, \ldots, z_i, 1)$ + updates
- Merge path: Run counter for merging $\pi_i^0 \circ \pi_i^1$ into π_{i+1}

Number of steps: $L_{i+1} = 3L_i \Longrightarrow L = L_n = 2^{n \cdot \log(3)}$

Proof size: $P_{i+1} = P_i + poly(n) \Longrightarrow P_n = poly(n)$

Fiat-Shamir for Sumcheck

Challenge: Off-path vertices due to

1. Soundness errors: accepting proof π for false statements y
2. Ambiguous proofs: accepting proof $\pi' \neq \pi$ for true statement y

Solution: Use "relaxed" SVL

Main assumption: resulting non-interactive argument is unambiguously sound for poly-time provers

Sanity check: True relative to a random oracle (and hence PPAD$\not\subseteq$FP relative to a random oracle)

Future Directions

- Instantiating Fiat-Shamir for sumcheck
  - Optimal hardness of circular-secure FHE: full version
  - From plain LWE?
- Factoring in PPAD?
  - PPAD-hardness from number-theoretic assumptions: eprint 2019/619, 2019/667
- Sampling small(ish) hard instances of NASH

THANK YOU. QUESTIONS?