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

ARKA RAI CHOU DHURI
PAVEL HUBÁČEK
CHETHAN KAMATH
KRZYSZTOF PIETRZAK
ALON ROSEN
GUY ROTHBLUM

JOHNS HOPKINS UNIVERSITY
CHARLES UNIVERSITY IN PRAGUE
IST AUSTRIA
IST AUSTRIA
IDC HERZLIYA
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 $\text{CLS} \subseteq \text{PPAD}$
NASH and PPAD

[P94, DGP05, CDT09]

- **Total Functional NP**
- Totality via "parity argument in directed graphs"

Diagram:
- **TFNP**
- **PPAD**
- **NASH**
- **FP**

- Crypto
  - FACTORING
  - DISCRETE-LOG
  - LWE
Average-case hardness in TFNP

- Hard-on-Average \( L \in \text{NP} \) [HNY17]
- Factoring [BO06, J16]
- **Today**: Soundness of FS for sumcheck protocol
- Obfuscation [HY17]
- One-way Permutations [P94]
- Hash Functions
- Obfuscation [AKV04, BPR15, GPS16, KS17]
Theorem: Indistinguishability obfuscation (iO) implies hardness in PPAD/CLS
1. iO $\rightarrow$ SINK-OF-VERIFIABLE-LINE (SVL)
2. SVL $\rightarrow$ NASH (END-OF-LINE) [AKV04]
2* SVL $\rightarrow$ END-OF-METERED-LINE ($\in$ CLS) [HY17]

Bottom-line: Focus on hard SVL instances
Exponential-sized graph with vertices in $\{0,1\}^n$

- Path defined by circuit $S: \{0,1\}^n \rightarrow \{0,1\}^n$
- Verifier circuit $V: [2^n] \times \{0,1\}^n \rightarrow \text{ACCEPT/REJECT}$
- Promise: $V(i, \sigma_i) = \text{ACCEPT} \iff \sigma_i = S^i(\sigma_1)$
- Solution: $\sigma_L = S^L(\sigma_1)$
SVL IS NO EASIER THAN BREAKING FIAT-SHAMIR
SVL as Verifiable Counter for #SAT

Reduce #SAT to SVL

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

Challenge: How to verify $\sigma_i$?

Solution: Append a succinct proof $\pi_i$
SVL as Verifiable Counter for \#SAT

\[ V(i, \sigma_i, \pi_i) = \text{ACCEPT} \]

\[ \sigma_i \text{ is the number of satisfying assignments between } 0^n \text{ and } i \]
SVL as Verifiable Counter for #SAT

Challenge: getting $\pi_i$ to be of size $\text{poly}(n)$
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

Sumcheck proof for $\sigma_i$

SVL counter $\mathbf{S}_0, \mathbf{V}_0, L_0$ for $\varphi(z_3, \ldots, z_0)$

$\Downarrow$

SVL counter $\mathbf{S}_9, \mathbf{V}_9, L_9$ for $\varphi(z_3, \ldots, z_0, z_0)$

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 $\varphi(z_1, ..., z_i, z_{i+1})$
- Left path: Run counter on $\varphi(z_1, ..., z_i, 0)$
- Right path: Run counter on $\varphi(z_1, ..., z_i, 1)$
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 $\varphi(z_1, \ldots, z_i, 1) + \text{update}$

Number of steps: $L_{i+1} = 2L_i$
Proof size: $P_{i+1} = 2P_i \Rightarrow P_n = 2^n$
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 $\varphi(z_1, \ldots, z_i, 1) + \text{updates}$
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 \( \pi^0_i \circ \pi^1_i \) into \( \pi_{i+1} \)

Number of steps: \( L_{i+1} = 3L_i \Rightarrow L = L_n = 2^{n \cdot \log(3)} \)
Proof size: \( P_{i+1} = P_i + \text{poly}(n) \Rightarrow P_n = \text{poly}(n) \)
Fiat-Shamir for Sumcheck

Challenge: Off-path vertices due to
1. Soundness errors: accepting proof $\pi$ 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≠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?