Scalable Transparent ARguments-of-Knowledge

Michael Riabzev

Department of Computer Science, Technion

DIMACS Workshop on Outsourcing Computation Securely

Joint work with Eli Ben-Sasson, Iddo Bentov, and Yinon Horesh

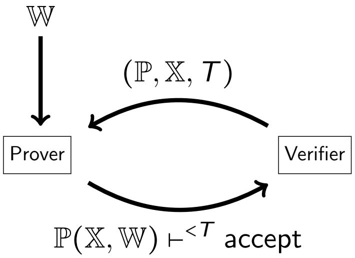
Talk outline

- Our result
- Novel theory review (Low degree testing)
- Concrete implementation performance review

Our result

Today I will tell you about STARK:

- "Scalable Transparent ARgument of Knowledge"
- New construction (theory+implementation[1]) featuring:
    - Perfect witness-indistinguishability
    - Publicly verifiable
    - No trusted-setup
    - Universal
    - Succinct verification
- And additionally:
    - Post-quantum secure
    - Scalable prover (quasi-linear)



[1] Proof-of-concept in C++

Computational model

Interactive Oracle Proofs (IOP)[BCS16, RRR16][2]:

- A generalization of IP[GMR89] and PCP[BFL91, AS98]
- Verifier interacts with the Prover
- Prover's messages too big for the verifier to read entirely
  - Also known as *oracles*

[2] also known as PCIP in [RRR16]

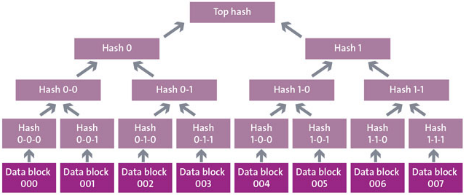
Computational model

Interactive Oracle Proofs (IOP)[BCS16, RRR16][2]:

- A generalization of IP[GMR89] and PCP[BFL91, AS98]
- Verifier interacts with the Prover
- Prover's messages too big for the verifier to read entirely
  - Also known as *oracles*

Realistic argument-system:

- Using Merkle trees [Kil92, Kil95, Mic00, BCS16]
- Noninteractive system : Fiat-Shamir heuristic



[2] also known as PCIP in [RRR16]

Cryptographic assumption

- Inner protocol (IOP):
  - Provably sound[3]
  - Provably perfect zero-knowledge
- Compilation to (noninteractive) argument system:
  - Using the random oracle model
- Implementation:
  - Simulating a random-oracle using a hash-function

[3] Implementation uses security conjectures to improve concrete efficiency.

Our result

Features

A peak under the hood

- Improvements
- Novel low-degree test
- Measurements

Summary

Our result

Features

A peak under the hood

- Improvements
- Novel low-degree test
- Measurements

Summary

STARK (this work) introduces improvements over SCI [BBCGGHPRSTV17] in several aspects: (Ben-Sasson, Bentov, Chiesa, Gabizon, Genkin, Hamilis, Pergament, R, Silberstein, Tromer, Virza)

- Privacy — witness indistinguishability based on [BCGV16]

STARK (this work) introduces improvements over SCI [BBCGGHPRSTV17] in several aspects: (Ben-Sasson, Bentov, Chiesa, Gabizon, Genkin, Hamilis, Pergament, R, Silberstein, Tromer, Virza)

- Privacy — witness indistinguishability based on [BCGV16]
- Arithmetization — optimized for interactive systems
  - Disclaimer: RAM usage introduces $\sim 8T \log T$ additive overhead to witness size
  - in addition to $O(T)$ witness size when no RAM is used
  - Derived from SCI

STARK (this work) introduces improvements over SCI [BBCGGHPRSTV17] in several aspects: (Ben-Sasson, Bentov, Chiesa, Gabizon, Genkin, Hamilis, Pergament, R, Silberstein, Tromer, Virza)

- Privacy — witness indistinguishability based on [BCGV16]
- Arithmetization — optimized for interactive systems
  - Disclaimer: RAM usage introduces $\sim 8T \log T$ additive overhead to witness size
  - in addition to $O(T)$ witness size when no RAM is used
  - Derived from SCI
- Low degree test — optimized for interactive systems

STARK (this work) introduces improvements over SCI [BBCGGHPRSTV17] in several aspects: (Ben-Sasson, Bentov, Chiesa, Gabizon, Genkin, Hamilis, Pergament, R, Silberstein, Tromer, Virza)

- Privacy — witness indistinguishability based on [BCGV16]
- Arithmetization — optimized for interactive systems
  - Disclaimer: RAM usage introduces $\sim 8T \log T$ additive overhead to witness size
  - in addition to $O(T)$ witness size when no RAM is used
  - Derived from SCI
- Low degree test — optimized for interactive systems
- Hash-tree commitment — optimization based on queries patter
  - Reducing communication complexity

STARK (this work) introduces improvements over SCI [BBCGGHPRSTV17] in several aspects: (Ben-Sasson, Bentov, Chiesa, Gabizon, Genkin, Hamilis, Pergament, R, Silberstein, Tromer, Virza)

- Privacy — witness indistinguishability based on [BCGV16]
- Arithmetization — optimized for interactive systems
  - Disclaimer: RAM usage introduces $\sim 8T \log T$ additive overhead to witness size
  - in addition to $O(T)$ witness size when no RAM is used
  - Derived from SCI
- Low degree test — optimized for interactive systems
- Hash-tree commitment — optimization based on queries patter
  - Reducing communication complexity
- System — code optimizations

STARK (this work) introduces improvements over SCI [BBCGGHPRSTV17] in several aspects: (Ben-Sasson, Bentov, Chiesa, Gabizon, Genkin, Hamilis, Pergament, R, Silberstein, Tromer, Virza)

- Privacy — witness indistinguishability based on [BCGV16]
- Arithmetization — optimized for interactive systems
  - Disclaimer: RAM usage introduces $\sim 8T \log T$ additive overhead to witness size
  - in addition to $O(T)$ witness size when no RAM is used
  - Derived from SCI
- Low degree test — optimized for interactive systems
- Hash-tree commitment — optimization based on queries patter
  - Reducing communication complexity
- System — code optimizations

STARK (this work) introduces improvements over SCI [BBCGGHPRSTV17] in several aspects: (Ben-Sasson, Bentov, Chiesa, Gabizon, Genkin, Hamilis, Pergament, R, Silberstein, Tromer, Virza)

- Privacy — witness indistinguishability based on [BCGV16]
- Arithmetization — optimized for interactive systems
  - Disclaimer: RAM usage introduces $\sim 8T \log T$ additive overhead to witness size
  - in addition to $O(T)$ witness size when no RAM is used
  - Derived from SCI
- Low degree test — optimized for interactive systems
- Hash-tree commitment — optimization based on queries patter
  - Reducing communication complexity
- System — code optimizations

In this talk we focus on the novel low-degree test

IOPP novel low-degree test

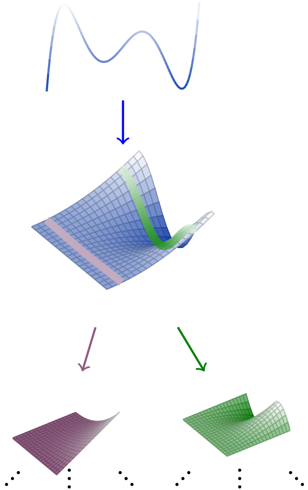
Theorem ([BBHR17])

Given oracle access to an evaluation $f : S \to \mathbb{F}_{2^n}$ over $\mathbb{F}_2$ linear **subspace** $S \subset \mathbb{F}_{2^n}$*, there is an IOPP protocol to verify f is close to degree $d < \frac{|S|}{3}$, with the following properties:*

- *Total proof size* $< \frac{|S|}{2}$.
- *Round complexity* $\frac{\log|S|}{2}$.
- *Prover complexity* $< 4|S|$ *arithmetic operations over* $\mathbb{F}_{2^n}$.
    - *Highly parallelizable.*
- *Query complexity is* $2\log|S|$.
- *Soundness:*
  $\Pr[Reject | dist(f, \mathcal{C}) = \delta] \geq \min\left(\delta, \frac{1}{4} - \frac{3d}{4|S|}\right) - 3\frac{|S|}{|\mathbb{F}_{2^n}|}$.
    - *Close to δ in the unique-decoding-radius.*
    - *Shown to be tight there.*

Low-degree testing in the Interactive-Oracle-Proof model

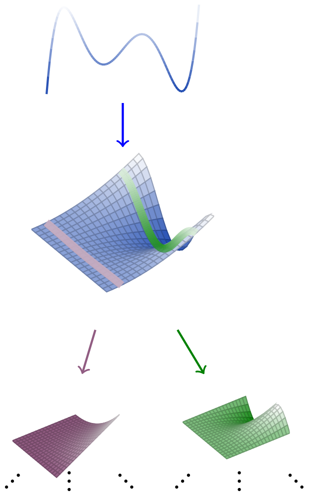
- **Redundancy addition:** Prover transforms univariate polynomial $p(x)$ into a bivariate polynomial $Q(x, y)$
- **Invariant:** $\deg_y(Q) = \deg(p)/4$
- **Verification:** Verifier chooses random x_0 and verifies $q(y) = Q(x_0, y)$ is low-degree
    - By repeating the test recursively
    - Until $\deg(q)$ is small enough



Low-degree testing — more details

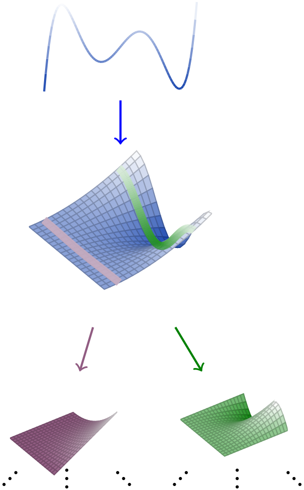
The transformation $T : \mathbb{F}[x] \to \mathbb{F}[x,y]$ is basically a biased version of [?]:

- $p(x) \in \mathbb{F}[x]$ is evaluated over $V = \mathrm{Span}\{b_1, b_2, \dots, b_n\}$
- Define:
  - $V_0 := \mathrm{Span}\{b_1, b_2\}$
  - $V_1 := \mathrm{Span}\{b_3, \dots, b_n\}$
  - $Z_{V_0}(x) := \prod_{\alpha \in V_0} (x - \alpha)$
- $T(p) = Q(x,y)$ where $Q(x,y) := p(x) \mod (y - Z_{V_0}(x))$
- Features:
  - $\forall x : Q(x, Z_{V_0}(x)) = p(x)$
  - $\deg_x(Q) < 4$
  - $\deg_y(Q) = \deg(p)/4$



Low-degree testing — advantages of interactivity

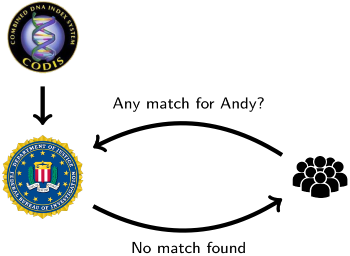
- Deeper recursion is possible due to provers adaptivity
- 'Lightweight' prover algorithm
- Better soundness:
  - Rows are low degree by definition
  - Any column can be queried



Benchmark : Forensics DNA blacklist

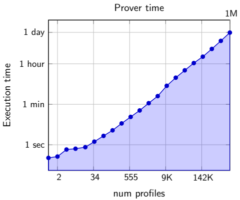
- FBI has the forensics DB
- knows hash digest of the DB
    - Davies-Meyer-AES160
- FBI provide Andy's DNA profiling[4] result with an integrity proof
- The program verified:

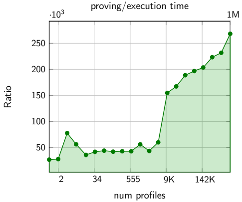
```
def prog(database):
    currHash = 0

    for currEntry in database:
        if currEntry matches AndysDNA:
            REJECT
        currHash = Hash(currEntry, currVal)

    if currHash == expectedHash : ACCEPT
    else : REJECT
```

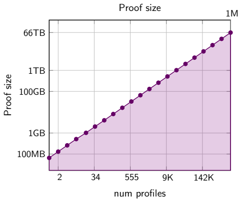


[4] Based on https://www.fbi.gov/services/laboratory/biometric-analysis/codis

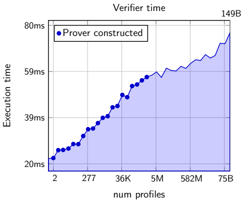
Machine specifications:
Prover: *CPU*: 4 X AMD Opteron(tm) Processor 6328 (32 cores total, 3.2GHz), *RAM*: 512GB
Verifier: *CPU*: Intel(R) Core(TM) i7-4600 2.1GHz, *RAM*: 12GB, *Circuit*: runtime simulated for long inputs
Security: *Security level*: 80 bits (Probability of cheating $< 2^{-80}$)

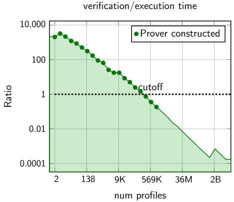


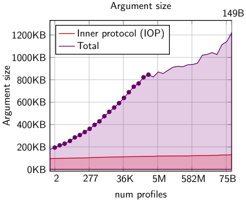




Conclusions: Prover asymptotic behaviour as predicted; Proving is $\sim \times 50K$ slower than program execution



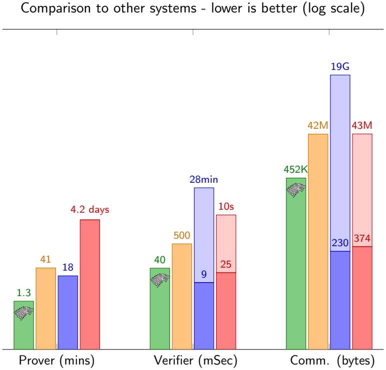




Conclusions: Verifier asymptotic behaviour as predicted; Speedup achieved only for a few generated arguments

Comparison to other approaches

Machine specifications:
CPU: 4 X AMD Opteron(tm) Processor 6328 (32 cores total, 3.2GHz), *RAM*: 512GB
Benchmark:
Executing subset-sum solver for 64K TinyRAM steps (9 elements — exhaustive algorithm).

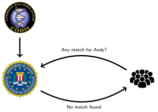


- **STARK**
- **SCI**[BBCGGHPRSTV17] — based on IOP.
- **KOE**[BCGTV13] — zkSNARK based on Knowledge Of Exponent hardness. **Non-succinct setup required.**
- **IVC**[BCTV14] — Incrementally Verifiable Computation based on KOE. **Setup required (succinct).**

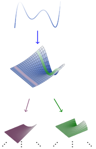
Fastest prover;
Verification ~ fastest so far;
CC lowest; Argument ~ $\times 1K$ longer "best"

Summary

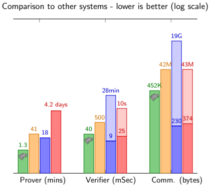
STARK Introduction:



New low-degree test:

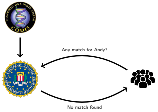


Concrete measurements:

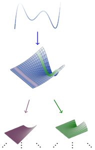


Summary

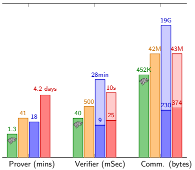
STARK Introduction:



New low-degree test:



Concrete measurements:



THANK YOU

Sanjeev Arora and Shmuel Safra.
Probabilistic checking of proofs: a new characterization of NP.

Journal of the ACM, 45(1):70–122, 1998.
Preliminary version in FOCS '92.

Eli Ben-Sasson, Alessandro Chiesa, Ariel Gabizon, and Madars Virza.
Quasilinear-size zero knowledge from linear-algebraic PCPs.
In *Proceedings of the 13th Theory of Cryptography Conference*, TCC '16-A, pages 33–64, 2016.

Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner.
Interactive oracle proofs.
In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, pages 31–60, 2016.

László Babai, Lance Fortnow, and Carsten Lund.

Non-deterministic exponential time has two-prover interactive protocols.
Computational Complexity, 1:3–40, 1991.
Preliminary version appeared in FOCS '90.

Shafi Goldwasser, Silvio Micali, and Charles Rackoff.
The knowledge complexity of interactive proof systems.
SIAM Journal on Computing, 18(1):186–208, 1989.
Preliminary version appeared in STOC '85.

Joe Kilian.
A note on efficient zero-knowledge proofs and arguments.
In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, STOC '92, pages 723–732, 1992.

Joe Kilian.
Improved efficient arguments.
In *Proceedings of the 15th Annual International Cryptology Conference*, CRYPTO '95, pages 311–324, 1995.

Silvio Micali.
Computationally sound proofs.
SIAM Journal on Computing, 30(4):1253–1298, 2000.
Preliminary version appeared in FOCS '94.

Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum.
Constant-round interactive proofs for delegating computation.
In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 49–62, 2016.