Lookup Arguments

12 July 2025

Outline

Backgrounds

SNARKs

A SNARK for NP relation $(x; w) \in \mathcal{R}$ is a non-interactive argument produced by a prover and verified by a verifier.

- Completeness.
- Knowledge Soundness.
- Succinctness: proof size is sublinear in $|x|$.
- Sublinear verification time: verifier time is sublinear in $|x|$.

Introduction

Proving Subsequence

A string **a** is a subsequence of **b** if one can delete some characters in **b** to achive **a**.

Example:

- $\mathbf{a} = abb$ is a subsequence of $\mathbf{b} = acbadb$.
- $\mathbf{a} = abb$ is not a subsequence of $\mathbf{b} = acadb$.

Approach by Thakur [Tha23]: Let $n = |\mathbf{a}|$ and $m = |\mathbf{b}|$. If **a** is a subsequence of **b**, then

- One can determine $p_1, \ldots, p_n$ such that each $(p_i, a_i) \in \{(j, b_j)\}_{j \in [m]}$ for $i \in [n]$.
- $p_1 < \cdots < p_n$.

$$p_1 < \cdots < p_n \iff p_i - p_{i-1} \in \{1, \ldots, m-1\} \text{ for } i \in 2 \ldots n.$$

Lookup in Proving Machine's Execution

Consider a machine with instruction set $\{F_1, \ldots, F_m\}$ and a program counter (pc) compute as follows:

- Step 0: $\mathsf{pc}_0 = 1$.
- For each $i \in 1 \ldots n$, $(\mathsf{pc}_i, \mathsf{st}_i) := F'_i(w_i, \mathsf{st}_{i-1}) = F_{\mathsf{pc}_{i-1}}(w_i, \mathsf{st}_{i-1})$ where w_i is some witness for step i.

To prove this machine's execution, a necessary part is showing that $F'_i = F_{\mathsf{pc}_{i-1}}$ for $i \in [n]$. This is equivalently show that

$$(\mathsf{pc}_{i-1}, F'_i) \in \{(j, F_j)\}_{j \in [m]} \ \forall i \in [n].$$

Related Works: MuxProofs [DXNT25], SublonK [CGG+24].

Lookup Arguments

Problem Statement

$$\mathbf{s} = (s_1, \ldots, s_n) \text{ for some } n \in \mathbb{N},$$
$$\mathbf{t} = (t_1, \ldots, t_n) \text{ for some } m \in \mathbb{N}.$$

Show that

$$\mathbf{s} \preceq \mathbf{t}.$$

Equivalently, $s_i \in \{t_i\}_{i \in m}$ for all $i \in [n]$.

Trivial Approaches

For each i, simply show that $s_i \in \{t_i\}_{i \in m}$, namely, set membership.

Issues:

- Large proof size, linear in $n = |\mathbf{s}|$.
- Verifier time is not sublinear.
- Potentially employing complex primitives, e.g., Merkle Trees.

Plookup's Approach

Plookup [GW20] overcomes inefficiencies by the method.

Sorting. Sort $(\mathbf{s}\|\mathbf{t}) = (s_1, \ldots, s_n, t_1, \ldots, t_m)$ by $\mathbf{t}$ to achieve $\mathbf{v}$.

Example. Let $\mathbf{s} = (1, 2, 2)$ and $\mathbf{t} = (2, 3, 1)$. Then,

- $(\mathbf{s}\|\mathbf{t}) = (1, 2, 2, 2, 3, 1)$.
- $\mathbf{v}$, set to be "$(\mathbf{s}\|\mathbf{t})$ after sorted by $\mathbf{t}$", becomes

$$\mathbf{v} = (2, 2, 2, 3, 1, 1).$$

Observation. $((v_1, v_2), \ldots, (v_{n+m-1}, v_{n+m}))$ is a permuation of $((s_1, s_1), \ldots, (s_n, s_n), (t_1, t_2), \ldots, (t_{m-1}, t_m))$ if and only if

$$\mathbf{s} \preceq \mathbf{t} \text{ and } \mathbf{v} \text{ is a sorting of } (\mathbf{s}\|\mathbf{t}) \text{ by } \mathbf{t}.$$

Haböck's Approach

$\mathbf{s} \preceq \mathbf{t}$ if and only if one can determine $\mu_1, \ldots, \mu_m$ such that

$$\sum_{i=1}^{n} \frac{1}{X + s_i} = \sum_{j=1}^{m} \frac{\mu_j}{X + t_j}.$$

Idea for Construction.

- Compute and commit to $a_i = \frac{1}{\chi + s_i}$ and $b_j = \frac{\mu_j}{\chi + t_j}$.
- Using ZeroCheck [CBBZ23, Set20] to verify

$$a_i \cdot (\chi + s_i) - 1 = 0 \ \forall i \in [n] \text{ and } b_j \cdot (\chi + t_j) - \mu_j = 0 \ \forall j \in [m].$$

- Using sumcheck [LFKN90] to verify $\sum_{i=1}^{n} a_i = \sum_{j=1}^{m} b_j$.

Lookup Arguments with Preprocessing - Brief Introduction

Lookup Arguments with Preprocessing

Recalling Problem Statement: Show that $\mathbf{s} \preceq \mathbf{t}$ where $\mathbf{s} = (s_1, \ldots, s_n)$ and $\mathbf{t} = (t_1, \ldots, t_m)$.

Previous Mentioned Lookup Arguments: Prover time is at leat $\mathcal{O}(m+n)$.

Lookup Arguments with Preprocessing:

- Preprocess $\mathbf{t}$ in advance.
- Prove $\mathbf{s} \preceq \mathbf{t}$ whose running time loosely depends on m.
- Applicable to problems where $m \gg n$.
- Works: Caulk [ZBK+22], Caulk+ [PK22], cq [EFG22], cq+, cq++ [CFF+24], Locq [ZSG24], ...

Thank You!

[CBBZ23] Binyi Chen, Benedikt Bünz, Dan Boneh, and Zhenfei Zhang.
HyperPlonk: Plonk with Linear-Time Prover and High-Degree Custom Gates.
In *Advances in Cryptology – EUROCRYPT 2023*, volume 14005 of *Lecture Notes in Computer Science*, pages 499–530. Springer Nature Switzerland, 2023.

[CFF+24] Matteo Campanelli, Antonio Faonio, Dario Fiore, Tianyu Li, and Helger Lipmaa.
Lookup Arguments: Improvements, Extensions and Applications to Zero-Knowledge Decision Trees.

In Qiang Tang and Vanessa Teague, editors, *Public-Key Cryptography – PKC 2024*, volume 14602 of *Lecture Notes in Computer Science*, pages 337–369. Springer Nature Switzerland, 2024.

[CGG+24] Arka Rai Choudhuri, Sanjam Garg, Aarushi Goel, Sruthi Sekar, and Rohit Sinha.
SublonK: Sublinear Prover PlonK.
In *Privacy Enhancing Technologies Symposium – PETS 2024*, pages 314–335, 2024.

[DXNT25] Zijing Di, Lucas Xia, Wilson Nguyen, and Nirvan Tyagi.
MuxProofs: Succinct Arguments for Machine Computation from Vector Lookups.
In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology – ASIACRYPT 2024*, volume 15488 of *Lecture Notes in Computer Science*, pages 236–265. Springer Nature Singapore, 2025.

[EFG22] Liam Eagen, Dario Fiore, and Ariel Gabizon.
cq: Cached quotients for fast lookups.
Cryptology ePrint Archive, Paper 2022/1763, 2022. https://eprint.iacr.org/2022/1763.

[GW20] Ariel Gabizon and Zachary J. Williamson.
plookup: A simplified polynomial protocol for lookup tables.
Cryptology ePrint Archive, Report 2020/315, 2020.

[LFKN90] C. Lund, L. Fortnow, H. Karloff, and N. Nisan.
Algebraic methods for interactive proof systems.
In *Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science – FOCS 1990*, volume 1, pages 2–10. IEEE, 1990.

[PK22] Jim Posen and Assimakis A. Kattis.
Caulk+: Table-independent lookup arguments.
Cryptology ePrint Archive, Paper 2022/957, 2022.

References v

[Set20] Srinath Setty.
Spartan: Efficient and General-Purpose zkSNARKs Without Trusted Setup.
In *Advances in Cryptology – CRYPTO 2020*, volume 12172 of *Lecture Notes in Computer Science*, pages 704–737. Springer International Publishing, 2020.

[Tha23] Steve Thakur.
A flexible Snark via the monomial basis.
Cryptology ePrint Archive, Paper 2023/1255, 2023.

[ZBK+22] Arantxa Zapico, Vitalik Buterin, Dmitry Khovratovich, Mary Maller, Anca Nitulescu, and Mark Simkin. **Caulk: Lookup Arguments in Sublinear Time.** In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security – CCS 2022*, pages 3121–3134. Association for Computing Machinery, 2022.

[ZSG24] Yuncong Zhang, Shi-Feng Sun, and Dawu Gu. **Efficient KZG-Based Univariate Sum-Check and Lookup Argument.** In Qiang Tang and Vanessa Teague, editors, *Public-Key Cryptography – PKC 2024*, volume 14602 of *Lecture*

Notes in Computer Science, pages 400–425. Springer Nature Switzerland, 2024.