cq:* Cached quotients for fast lookups

Liam Eagen   Dario Fiore   Ariel Gabizon
IMDEA   Zeta Function Technologies

December 18, 2022

Abstract

We present a protocol for checking the values of a committed polynomial $\phi(X) \in \mathbb{F}_{<N}[X]$ over a multiplicative subgroup $\mathbb{H} \subset \mathbb{F}$ of size n are contained in a table $T \in \mathbb{F}^N$. After an $O(N \log N)$ preprocessing step, the prover algorithm runs in time $O(n \log n)$. Thus, we continue to improve upon the recent breakthrough sequence of results starting from Caulk [ZBK+22], which was the first to achieve sublinear complexity in the full table size N, Caulk+ [PK22, ?, ?], that has so far reached prover time $O(n \log^2 n)$.

1 Introduction

The *lookup problem* is fundamental to the efficiency of modern zk-SNARKs. Somewhat informally, it asks for a protocol to prove the values of a committed polynomial $\phi(X) \in \mathbb{F}_{<n}[X]$ are contained in a table T of size N of predefined legal values. When the table T corresponds to an operation without an efficient low-degree arithmetization in $\mathbb{F}$, such a protocol produces significant savings in proof construction time for programs containing the operation. Building on previous work of [BCG+18], plookup [GW20] was the first to explicitly describe a solution to this problem in the polynomial-IOP context. plookup described a protocol with prover complexity quasilinear in both n and N. This left the intriguing question of whether the dependence on N could be made *sublinear* after performing a preprocessing step for the table T. Caulk [ZBK+22] answered this question in the affirmative by leveraging bi-linear pairings, achieving a run time of $O(n^2 + n \log N)$. Caulk+ [PK22] improved this to $O(n^2)$ getting rid of the dependence on table size completely.

However, the quadratic dependence on n of these works makes them impractical for a circuit with many lookup gates. We resolve this issue by giving a protocol called cq that is quasi-linear in n and has no dependence on N after the preprocessing step.

*Pronounced as "seek you".

1.1 Comparison of results

Table with relative proof size, prover ops, verifier ops
caulk caulk+ flookup baloo this work

1.2 Overview

-logarithmic derivative method
- For large table problem is computing A that agrees with $M/(t+\beta)$ on $\mathbb{V}$
- Need way to compute A

2 Preliminaries

2.1 Notation:

$\mathbb{H}$- small space $\mathbb{V}$- big space Lagrange bases for big and small space
AGM - real and ideal pairing checks, agm - real and ideal pairing KZG

2.2 log derivative method

Lemma from mvlookup

Lemma 2.1. *Given $f \in \mathbb{F}^n$, and $t \in \mathbb{F}^N$, we have $f \subset t$ as sets if and only if for some $m \in \mathbb{F}^N$ the following identity of rational functions holds*

$$\sum_{i\in[n]} \frac{1}{X+f_i} = \sum_{i\in[N]} \frac{m_i}{X+t_i}.$$

3 Cached quotients

Theorem 3.1. *Fix $T \in \mathbb{F}_{<N}[X]$, and a subgroup $\mathbb{V} \subset \mathbb{F}$ of size N. There is an algorithm that after a preprocessing step of $O(N \cdot \log N)$ operations. Given input $f \in \mathbb{F}_{<n}[X]$ computes in $O(n \cdot \log n)$ $\mathbb{G}_2$ operations $\mathsf{cm} = [Q(x)]_2$ where $Q \in \mathbb{F}_{<N}[X]$ is such that*

$$f(X) \cdot T(X) = Q(X) \cdot Z_{\mathbb{V}}(X) + R(X),$$

for $R(X) \in \mathbb{F}_{<N}[X]$

Lemma 3.2. *Fix $T \in \mathbb{F}_{<N}[X]$, and a subgroup $\mathbb{V} \subset \mathbb{F}$ of size N. There is an algorithm that given the $\mathbb{G}_1$ elements $\left\{\left[x^i\right]_1\right\}_{i\in\{0,\ldots,N\}}$ computes for $i \in [N]$, the elements $q_i := [Q_i(x)]_1$ where $Q_i(X) \in \mathbb{F}[X]$ is such that*

$$L_i(X) \cdot T(X) = t_i \cdot L_i(X) + Z_{\mathbb{V}}(X) \cdot Q_i(X)$$

in $O(N \cdot \log N)$ $\mathbb{G}_1$ operations.

Lemma 3.3. *Fix $T \in \mathbb{F}_{<N}[X]$, and a subgroup $\mathbb{V} \subset \mathbb{F}$ of size N. There is an algorithm that given the $\mathbb{G}_1$ elements $\left\{\left[x^i\right]_1\right\}_{i\in\{0,\ldots,N\}}$ computes for $i \in [N]$, the elements $q_i := \left[^{d-N} \cdot Q_i(x)\right]_1$ where $Q_i(X) \in \mathbb{F}[X]$ is such that*

$$L_i(X) \cdot T(X) = t_i \cdot L_i(X) + Z_{\mathbb{V}}(X) \cdot Q_i(X)$$

in $O(N \cdot \log N)$ $\mathbb{G}_1$ operations.

4 Main protocol

Definition 4.1. *$\mathcal{R}$ is all pairs (cm, f) such that cm is a commitment to f and $f|_{\mathbb{H}} \subset T$.*
..bla problem is relation is defined only after srs is chosen

Ad-hoc dfn of ks protocol for table lookup
Relations dependent on srs.

1. Fix table T
2. Choose srs
3. relation is (cm, f) such that $[f(x)]_1 = \mathsf{cm}$ and $f|_{\mathbb{H}} \subset T$.
4. **P** sends cm
5. E outputs f
6. **P** and **V** Pr that **V** outputs acc but $f|_{\mathbb{H}} \not\subset T$ is $\mathsf{negl}(\lambda)$

Main protocol: Input (cm, f).

1. **P** computes poly $m \in \mathbb{F}_{<N}[X]$ such that $m_i =$ number of times t_i appears in $f|_{\mathbb{H}}$
2. **P** sends $[m(x)]_1$.
3. **V** chooses and sends random $\beta \in \mathbb{F}$.
4. **P** computes $A \in \mathbb{F}_{<N}[X]$ such that for $i \in [N]$, $A_i = m_i/(t_i + \beta)$.
5. **P** sends $[A(x)]_1$.
6. **P** computes $q_A := [Q_A(x)]_2$ where $Q_A \in \mathbb{F}_{<N}[X]$ is such that

$$A(X)(T(X) + \beta) - m(X) = Q_A(X) \cdot Z_{\mathbb{V}}(X)$$

7. **P** computes $B \in \mathbb{F}_{<n}[X]$ such that for $i \in [n]$, $B_i = 1/(f_i + \beta)$.
8. **P** sends $[B(x)]_1$

9. **P** computes $Q_B(X)$ such that

$$B(X)(f(x)+\beta)-1=Q_B(X)\cdot Z_{\mathbb{H}}(X)$$

10. **P** computes and sends the values $a=A(0), b=B(0)$.

11. **V** sends random $\alpha\in\mathbb{F}$.

12. **V** checks that $N\cdot a=witsize\cdot b$.

13. **P** computes and sends $p=[P(x)]_1$ where

$$P(X):=A(X)\cdot X^{d-N}+\alpha\cdot B(X)X^{d-n}$$

14. **V** sends random $\gamma\in\mathbb{F}$.

15. **P** sends $A(\gamma), B(\gamma), Q_B(\gamma), f(\gamma)$.

16. **V** checks P is correct

17. **P** sends KZG proofs to all these poly openings.

Lemma 4.2. *The element q_A in Step 6 can be computed in XX $\mathbb{G}_2$ operations and $O(n\log n)$ $\mathbb{F}$ operations*

References

[BCG+18] J. Bootle, A. Cerulli, J. Groth, S. K. Jakobsen, and M. Maller. Arya: Nearly linear-time zero-knowledge proofs for correct program execution. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*, volume 11272 of *Lecture Notes in Computer Science*, pages 595–626. Springer, 2018.

[GW20] A. Gabizon and Z. J. Williamson. plookup: A simplified polynomial protocol for lookup tables. *IACR Cryptol. ePrint Arch.*, page 315, 2020.

[PK22] J. Posen and A. A. Kattis. Caulk+: Table-independent lookup arguments. 2022.

[ZBK+22] A. Zapico, V. Buterin, D. Khovratovich, M. Maller, A. Nitulescu, and M. Simkin. Caulk: Lookup arguments in sublinear time. *IACR Cryptol. ePrint Arch.*, page 621, 2022.