Caulk: Lookup Arguments in Sublinear Time

Arantxa Zapico*[1], Vitalik Buterin[2], Dmitry Khovratovich[2], Mary Maller[2], Anca Nitulescu[3], and Mark Simkin[2]

[1] Universitat Pompeu Fabra†
[2] Ethereum Foundation‡
[3] Protocol Labs§

Abstract

We present *position-hiding linkability* for vector commitment schemes: one can prove in zero knowledge that one or m values that comprise commitment cm all belong to the vector of size N committed to in C. Our construction Caulk can be used for membership proofs and lookup arguments and outperforms all existing alternatives in prover time by orders of magnitude.

For both single- and multi-membership proofs Caulk beats SNARKed Merkle proofs by the factor of 100 even if the latter instantiated with Poseidon hash. Asymptotically our prover needs $O(m^2 + m \log N)$ time to prove a batch of m openings, whereas proof size is $O(1)$ and verifier time is $O(\log(\log N))$.

As a lookup argument, Caulk is the first scheme with prover time sublinear in the table size, assuming $O(N \log N)$ preprocessing time and $O(N)$ storage. It can be used as a subprimitive in verifiable computation schemes in order to drastically decrease the lookup overhead.

Our scheme comes with a reference implementation and benchmarks.

1 Introduction

A vector commitment is a basic cryptographic scheme, which lies at the foundation of numerous constructions and protocols. In a nutshell, a vector commitment is a compact data structure that "contains" a (potentially very big) number of elements and allows *proving* that a specific element has been committed to it. A natural requirement is that a proof is *succinct* and *unforgeable*. A Merkle tree is a well known example of vector commitment.

The rise of privacy-preserving applications makes it vital to make proofs zero-knowledge i.e. hiding the element that is asserted to be in the commitment while still establishing a certain relationship, or *link*, to that element. Thus a vector commitment to $\boldsymbol{v} = (v_1, \ldots, v_N)$ is linkable if it permits proving that you know a secret s_i mathematically linked to v_i. The simplest example is the proof of authorization where a party proves the knowledge of a secret key beyond one of public key in the set. A more elaborate example is a *proof of coin ownership* in private cryptocurrencies: with coins stored as hashes of secret k and value v in a list or a tree, prove that you can spend v by showing its k in zero knowledge. A third example is a lookup argument in verifiable computation: prove that intermediate values $a_1, a_2, \ldots, a_k$ are all contained in a certain table (e.g., a table of all 16-bit numbers for the purpose of overflow checks in financial or mathematical computations). Applications also include membership proofs, ring signatures, anonymous credentials and other schemes.

So far all these examples have been handled by working but not so efficient mechanisms, which limit scalability and adoption. The first version of the Zcash cryptocurrency [30] used a SHA-2-based Merkle tree to store the coins and the Groth16 [20] SNARK to prove the coin ownership. Relatively heavy machinery of Groth16 and the unfit of SHA-2 to prime-field circuits made the resulting prover time of 40

*This work was done while Arantxa Zapico was an intern at the Ethereum Foundation.

†arantxa.zapico@upf.edu
‡{v_buterin, mary.maller, mark.simkin}@ethereum.org, khovratovich@gmail.com
§anca@protocol.ai

seconds barely usable. Even the most recent developments of algebraic hashes [1, 19] reduce *prover time* by the order of magnitude only. Another application of concern, lookup tables, so far has required the generic construction of Plookup [17], and, the last but not least, makes the prover be *at least as big* as the table itself no matter how many values they look up.

1.1 Our Contributions

In this paper we present a novel construction to add position-hiding linkability for vector commitments, named Caulk, which solves both problems with unprecedented efficiency. We construct a proof of membership with concrete efficiency of the factor of 100x over Poseidon Merkle trees, with the same asymptotic of $O(\log N)$ for N-sized commitments. Our construction naturally extends to proof of subset memberships, thus leading the way to more efficient lookup arguments. We have removed the bottleneck of big tables by achieving the yet impossible $O(m \log N + m^2)$ cost for m-subvector lookups. The Verifier is succinct as it requires only $O(\log(\log N))$ scalar operations as well as constant number of pairings to verify a constant-size proof. We envision the widespread deployment of our construction both in generic lookup-equipped proof systems [17, 27] and specific applications with membership proofs.

The prover benefits are even more extreme when compared with Merkle-SNARKs that use SHA-2. Caulk also has constant proof sizes and $\mathcal{O}(\log(\log N))$ verifier time. It achieves statistical zero-knowledge and soundness in the algebraic group model, and requires a universal setup and $O(N)$ storage.

1.2 Paper Structure

We start with a technical overview of Caulk in Section 2. We present two instances: one for single-element proof of commitment (the $m = 1$ case), and the other for proving an m-subset: all values committed to cm are elements of the vector committed in C.

Related work is discussed in Section 3. Section 4 provides a self-complete description of the tools we use, particularly the KZG commitment and precomputation techniques, and can be skipped by a knowledgeable reader.

In Section 5 we identify our constructions as special cases of a more general family of protocols that add what we define as *position-hiding linkability* to vector commitment schemes. This primitive asserts that all (hidden) entries committed to cm are also (publicly) committed in C. Position-hiding refers to the fact that no information about which elements were taken to construct cm should be leaked. We formalize its definition as well as the security notions to be held.

In Section 6 we formally describe Caulk for the case $m = 1$ and prove it is statistically zero-knowledge and sound in the algebraic group model. As an important building block we also introduce a construction that demonstrates that a pedersen commitment contains a root of unity.

In Section 7 we extend Caulk even further to m-subset proofs, with some values possibly repeating. In this scenario Caulk can be seen as a lookup table, and is thus a prover efficient alternative to schemes such as Plookup [17]. We discuss various optimizations in Section 8.

Caulk comes with an open source reference implementation in Rust using arkworks library [1]. In Section 9 we compare its efficiency with some rival schemes.

2 Caulk in a nutshell

We present two constructions, one for the case $m = 1$, and another for $m > 1$. The starting point for both of them is a public vector $\vec{c}$ encoded as a polynomial $C(X) = \sum_{i=1}^{N} c_i \lambda_i(X)$, where $\{\lambda_i(X)\}_{i=1}^{N}$ are the Lagrange interpolation polynomials corresponding to some set of roots of unity $\mathbb{H} = \{1, \omega, \ldots, \omega^{N-1}\}$, with $\omega^N = 1$. Both prover and verifier have access to a KZG commitment to $C(X)$, i.e., to a $\mathbb{G}_1$-element $\mathsf{C} = \sum_{i=1}^{N} c_i [\lambda_i(x)]_1$ where x is secret.

[1] https://github.com/caulk-crypto/caulk

2.1 Construction for $m = 1$

Our idea is for the prover to demonstrate, in zero-knowledge, that a pedersen commitment cm opens to c_i and that they know a verifying KZG proof $[Q_i]_1$ such that

$$e(\mathsf{C} - [c_i]_1, [1]_2) = e([Q_i]_1, [a(x - \omega_i)]_2), \text{for some blinding factor } a.$$

Our prover time is unaffected by the computation of the non-hiding KZG proof $[Q_i]_1$ because we can compute all $[Q_i]_1$s in advance using $N \log N$ group operations as in [28, 14]. As a result our prover does require linear storage.

First the prover demonstrates knowledge of c_i and r such that $\mathsf{cm} = [c_i + \mathsf{h}r]_1$ for unknown h given to them as $[\mathsf{h}]_1$ in the setup. We use standard arguments of knowledge for pedersen commitments in order to prove well formation of cm. The challenge is then to prove well formation of a commitment to $[a(x - \omega_i)]_2$ and prove that ω_i is a root of unity i.e. that $\omega_i^N = 1$.

In order to avoid working with unnecessarily big polynomials, we introduce a new subgroup of roots of unity $\mathbb{V}_n = \{1, \ldots, \sigma^{n-1}\}$ with $n = \log(N) + 6$ and $\sigma^n = 1$. We create a polynomial $f(X)$ of degree n such that, using its first 4 coefficients, the prover can convince the verifier that for $z(X)$ of degree 1, $z(X) = aX + b = a(X + \frac{b}{a})$, $f(\sigma^4) = \frac{a}{b}$. The other coefficients of $f(X)$ are constructed so $f(\sigma^{4+i}) = \left(\frac{a}{b}\right)^{2^i}$, and the last one used to show that $\left(\frac{a}{b}\right)^N = \left(\frac{b}{a}\right)^N = 1$. By iteratively demonstrating that $f(\sigma^{4+i+1}) = f(\sigma^{4+i}) \times f(\sigma^{4+i})$ we can compute the powers of $\frac{a}{b}$ up to $2^{\log N} = N$ while performing only $O(\log(N))$ computations.

2.2 Construction for $m > 1$

We extend our results to $m > 1$ so that we prove that cm is a commitment to $\vec{a} = (c_{i_1}, c_{i_2}, \ldots, c_{i_m})$, where c_{i_j} is an element in $\vec{c}$ for all $j = 1, \ldots, m$. We commit to $\vec{a}$ as a polynomial $\phi(X) = \sum_j c_{i_j} \mu_j(X)$ where $\{\mu_j(X)\}_{j=1}^m$ are Lagrange interpolation polynomials over $\mathbb{V}_m = \{\nu, \ldots, \nu^{m-1}, \nu^m = 1\}$. Let $I = \{i_1, \ldots, i_m\} \subset [N]$, where each $i \in I$ is included only once; that is, the set of all index i such that c_i is an element of $\vec{a}$, without repetitions.

Tomescu et al. observed that multiple KZG evaluation proofs can be naturally aggregated into a single proof [28]. Denoting $\tau_i(X)$ as the lagrange polynomials that evaluates to 1 at ω^i and 0 at all others ω^k they show that

$$C(X) - \sum_{i \in I} c_i \tau_i(X) = \prod_{i \in I} (X - \omega^i) H(X) \text{ if and only if } C(\omega^i) = c_i. \tag{1}$$

and that $[H(x)]_1$ can be efficiently computed assuming that the non-hiding KZG proof $[Q_i]_1$ have been precomputed (as in the case $m = 1$). Given access to individual proofs $Q_i(X)$, the prover can perform linear combinations with them and compute $H(X)$ in $O(|I|) = O(m)$ time. We discuss this aggregation in more detail in Section 4.6.1.

Now the prover generates a hiding commitments to $\vec{c}_I = (c_i)_{i \in I}$ and $z_I(X) = \prod_{i \in I}(X - \omega^i)$, and outputs them together with a commitment to the polynomial $H(X)$ such that (1) holds. Concretely we have polynomial equation

$$C(X) - C_I(X) = z_I(X) H(X) \tag{2}$$

asserted against commitments $\mathsf{C}, [C_I], [z_I], [H]$. It remains to prove that (i) $z_I(X)$ has the right form, (ii) $[C_I]$ is a commitment to the same values as $\mathsf{cm} = \sum_j^m c_{i_j} \mu_j(X)$ but in a different basis: τ_i vs μ_j. For the first statement we introduce an auxiliary polynomial $u(X) = \sum_{j=1}^m \omega^{i_j} \mu_j(X)$. We prove that $u(X)$'s coefficients are Nth roots of unity by providing a proof that $u^j(X) = u^{j-1}(X) u^{j-1}(X)$ for $j = 1, \ldots, m$, when evaluated at elements in $\mathbb{V}_m$, and showing that $u_0(X) = u(X)$ and $u_n(X) = 1$. Then it remains to show that $z_I(X)$ vanishes at coefficients of $u(X)$ i.e. $z_I(u(X))$ vanishes at $\mathbb{V}_m$. This is done by providing $H_2(X)$ such that $z_I(u(X)) = z_{V_m}(X) H_2(X)$. Note that the argument holds also when $u(X)$ has repeated coefficients.

The second statement is proven by asserting the polynomial equation

$$C_I(u(X)) - \phi(X) = \prod_{j=1}^m (X - \nu^j) H(X),$$

Scheme	Trusted Params	|srs|	Proof size	Prover work	Verifier work
Merkle trees + zkSNARKs	Updatable	$m\log(N)$	$13\mathbb{G}_1, 8\mathbb{F}$	$\tilde{O}(m\log(N))$	2P
RSA accumulators	Yes	$O(1)$	$2\ \mathcal{G}$	$O(\log(m))$	m exp
Caulk single opening (Sec. 6)	Updatable	$O(N)$	$6\mathbb{G}_1$, $2\mathbb{G}_2$, $4\mathbb{F}$	$\tilde{O}(\log(N))$	4P
Caulk lookup (Sec. 7)	Updatable	$O(N)$	$14\mathbb{G}_1, 1\mathbb{G}_2, 4\mathbb{F}$	$\tilde{O}(m^2 + m\log(N))$	4P

Table 1: Cost comparison of our scheme with alternative proofs for membership and lookups. N is the size of the table and m the size of the set to be opened. We consider that Merkle trees + zk-SNARKs are implemented using Marlin [13] and note that these numbers are different with other SNARKs. Note that the asymptotic prover work for the Merkle trees + zkSNARKs hides the large constants involved in arithmetising hash functions. The RSA accumulator asymptotics hides large constants: for example $\mathcal{G}$ denotes a hidden order group that has larger size than $\mathbb{G}_1$, $\mathbb{G}_2$.

thus linking an input $\phi(X)$ in the known basis $\{\mu_j(X)\}_{j=1}^m$ to $C_I(X)$ in the unknown basis $\{\tau_i(X)\}_{i\in I}$. In our protocol, the procedure above is performed by also including blinders to hide the positions and the values taken to construct $z_I(X), \vec{c}_I$ and $\vec{a}$.

3 Related Work

Merkle-SNARK Zcash protocol [30] proposed a SNARK over a circuit described a Merkle tree opening for the anonymous proof of coin ownership. It remains a very popular approach for various set membership proof protocols [29, 31]. The prover costs are logarithmic in the number of tree leafs, but the concrete efficiency varies depending on the hash function that comprises the tree [1, 19]. Regular hash functions such as SHA-2 are known to be very slow, whereas algebraic alternatives are rather novel and some applications are reluctant to use them.

Pairing Based Camenisch et al.[10] describe a vector commitment that only requires constant prover and verifier costs. However the commitments themselves are computed by a trusted third party and have linear size because the prover requires access to $g^{\frac{1}{x-c_i}}$ for all c_i in the vector and x secret.

Discrete-Log Based In the discrete-logarithm setting a series of works have looked into achieving logarithmic sized zero-knowledge membership proof [3, 21, 8, 9]. These have the advantage that there is no trusted setup or pairings. The prover and verifier costs are asymptotically dominated by a linear number of field operations. For modest sized vectors this can be practical because the number of more computationally intensive group operations is logarithmic.

RSA Accumulators Camenisch and Lysyanskaya [11] design a proof of knowledge protocol for linking a commitment over a prime ordered from to an RSA accumulator. There are no a-priori bounds on the size of the vector. This approach is used by Zerocoin [25] which is a privacy preserving payments system (the predecessor to Zerocash [5]). Benarroch et al. [6] improve on this result by allowing the use of prime ordered groups of of "standard" size, e.g., 256 bits, whereas [11] needs a much larger group. RSA based schemes have constant sized public parameters. Benarroch et al. [6] and later Campanelli et al. [12] enjoy a prover time that is over twice as fast as zk-SNARKs over Merkle trees with Pedersen hashes and their implementation has not yet been heavily optimised. The verifier time is also asymptotically and concretely efficient. However they require either a trusted RSA modulus or class groups and the proof size is relatively large (about 2 to 5 KBytes).

4 Preliminaries

A bilinear group gk is a tuple $gk = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathcal{P}_1, \mathcal{P}_2)$ where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of prime order q, the elements $\mathcal{P}_1, \mathcal{P}_2$ are generators of $\mathbb{G}_1, \mathbb{G}_2$ respectively. We also consider $\hat{\mathcal{P}}_1$ another generator of $\mathbb{G}_1$ and will denote it as $[\mathsf{h}]_1$, where h is unknown and $\mathsf{h}\mathcal{P}_1 = \hat{\mathcal{P}}_1$. $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable, non-degenerate bilinear map, and there is an efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. Elements in $\mathbb{G}_\gamma$, are denoted implicitly as $[a]_\gamma = a\mathcal{P}_\gamma$, where $\gamma \in \{1, 2, T\}$ and $\mathcal{P}_T = e(\mathcal{P}_1, \mathcal{P}_2)$. With this notation, $e([a]_1, [b]_2) = [ab]_T$. When informally introducing techniques, we may use $[a]$ to refer to element a given in either $\mathbb{G}_1$ or $\mathbb{G}_2$.

4.1 Lagrange Polynomials and Roots of Unity

We use ω to denote a root of unity such that $\omega^N = 1$, and define $\mathbb{H} = \{1, \omega, \ldots, \omega^{N-1}\}$. Also, we let $\lambda_i(X)$ denote the i^{th} lagrange polynomial, i.e., $\lambda_i(X) = \prod_{s \neq i-1} \frac{X-\omega^s}{\omega^{i-1}-\omega^s}$ and $z_H(X) = \prod_{i=0}^{N-1}(X-\omega^i) = X^N - 1$ the vanishing polynomial of $\mathbb{H}$. We will additionally consider smaller groups of roots of unity in Sections 6, 7 and 7.1, that will be introduced accordingly.

4.2 Security parameters and algorithms

Let $\lambda \in \mathbb{N}$ denote the security parameter and 1^λ its unary representation. A function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$ is called *negligible* if for all $c > 0$, there exists k_0 such that $\mathsf{negl}(k) < \frac{1}{k^c}$ for all $k > k_0$. For a non-empty set S, let $x \xleftarrow{\$} S$ denote sampling an element of S uniformly at random and assigning it to x.

Let PPT denote probabilistic polynomial-time. Algorithms are randomized unless explicitly noted otherwise. Let $y \leftarrow A(x; r)$ denote running algorithm A on input x and randomness r and assigning its output to y. Let $y \xleftarrow{\$} A(x)$ denote $y \leftarrow A(x; r)$ for a uniformly random r.

4.3 Games

Code-based games are used in security definitions [4]. A game $\mathsf{Game}^{\mathsf{sec}}_{\mathcal{A}}(\lambda)$, played with respect to a security notion sec and adversary $\mathcal{A}$, has a main procedure whose output is the output of the game.

4.4 Cryptographic Assumptions

The security of our protocols holds in the Algebraic Group Model (AGM) of Fuchsbauer et al.([15]), using the dlog, qDHE and qSDH assumptions [18, 7]. In the AGM adversaries are restricted to be *algebraic*, namely, when an adversary $\mathcal{A}$ gets some group elements as input and outputs another group element, it can provide some algebraic representation of the latter in terms of the former.

Definition 4.1 (Algebraic Adversary)**.** *Let $\mathbb{G}$ be a cyclic group of order p. We say that a PPT adversary $\mathcal{A}$ is algebraic if there exists an efficient extractor $\mathcal{E}_{\mathcal{A}}$ that, given the inputs $([x_1], \ldots, [x_m])$ of $\mathcal{A}$, outputs a representation $\mathbf{z} = (z_1, \ldots, z_m)^\top \in \mathbb{F}^m$, where $\mathbb{F}$ is the finite field of p elements, for every group element $[y]$ in the output of $\mathcal{A}$ such that:*

$$\mathsf{Adv}^{alg}_{\mathbb{G},\mathcal{A}}(\lambda) = \left[\begin{array}{c} [y] \leftarrow \mathcal{A}([x_1], \ldots, [x_m]), \mathbf{z} \leftarrow \mathcal{E}_{\mathcal{A}}([y], [x_1], \ldots, [x_m]), \\ \text{and } [y] \neq \sum_{j=1}^{m} z_j [x_j] \end{array} \right] = \mathsf{negl}(\lambda).$$

4.5 The KZG Polynomial Commitment Scheme

Our constructions heavily rely on the KZG polynomial commitment scheme (Def. A.3) that we describe below, as well as its adaptation for vector commitment that we explain in the next section. For efficiency, we slightly modify the polynomial commitment in order to add degree checks to the original protocol, without incurring in extra proof elements or pairings. The polynomial commitment introduced by Kate, Zaverucha and Goldberg in [22] is a tuple of algorithms $(\mathsf{KZG.Setup}, \mathsf{KZG.Commit}, \mathsf{KZG.Open}, \mathsf{KZG.Verify})$ such that:

- $\mathsf{srs}_{\mathsf{KZG}} \leftarrow \mathsf{KZG.Setup}(\mathsf{par}_{\mathsf{KZG}}, d)$: On input the system parameters and a degree bound d, it outputs a structured reference string $\mathsf{srs}_{\mathsf{KZG}} = \left(\{[x^i]_{1,2}\}_{i=1}^{d}\right)$.
- $\mathsf{C} \leftarrow \mathsf{KZG.Commit}(\mathsf{srs}_{\mathsf{KZG}}, p(X))$: It outputs $\mathsf{C} = [p(x)]_1$.
- $(s, \pi_{\mathsf{KZG}}) \leftarrow \mathsf{KZG.Open}(\mathsf{srs}_{\mathsf{KZG}}, p(X), \alpha)$: Prover computes

  $$q(X) = \frac{p(X) - p(\alpha)}{X - \alpha},$$

  sets $s = p(\alpha), [Q]_1 = [q(x)x^{d-\deg+2}]_1$, and outputs $(s, \pi_{\mathsf{KZG}} = [Q]_1)$.
- $1/0 \leftarrow \mathsf{KZG.Verify}(\mathsf{srs}_{\mathsf{KZG}}, \mathsf{C}, \deg, \alpha, s, \pi_{\mathsf{KZG}})$: Verifier accepts if and only if

  $$e(\mathsf{C} - s, [x^{d-\deg+2}]_2) = e([Q]_1, [x - \alpha]_2).$$

Security. It has been proven in [22, 13, 16] that the original KZG protocol, i.e., where $[Q]_1 = [q(x)]_1$ and the pairing equation is $e(\mathsf{C} - s, [1]_2) = e([Q]_1, [x - \alpha]_2)$, is a polynomial commitment scheme that satisfies completeness, evaluation blinding and extractability as in Def. A.3 in the AGM, under the dlog assumption. What is more, Marlin presents an alternative version of KZG with degree checks that does not require additional powers in $\mathbb{G}_2$. For our construction, we claim that adding $x^{d-\deg+2}$ to the pairing and element $[Q]_1$ does not affect completeness or extractability. We also argue that under the AGM, no PPT adversary $\mathcal{A}$ can break soundness by providing a commitment to a polynomial $p(X)$ such that $\deg(p) > \deg$. Indeed, if that is the case, $\deg(Q) = d + 1$ for $Q(X)$ the algebraic representation of $[Q]_1$, which will imply an attack to the d-DHE assumption, as the srs only contains powers $[x^i]_1$ up to d.

4.6 KZG as Vector Commitment Scheme

There is a natural isomorphism between vectors of size m and polynomials of degree $m - 1$; where we can represent $\vec{v} = (v_1, \ldots, v_m) \in \mathbb{F}^m$ as $V(X) = \sum_{j=1}^{m} v_j B_j(X)$, where $\mathcal{B} = \{B_j(X)\}_{j=1}^{m}$ is a basis of the space of polynomials of degree up to $m - 1$, and vice versa. This fact implies as well a natural relation between polynomial and vector commitments (Def. A.2), where in particular, the former implies the latter. What is more, when the basis $\mathcal{B}$ chosen to encode the vector consists of Lagrange polynomials we have vector commitments with *easy* individual position openings: evaluating $V(X)$ in the $i - 1$th interpolation point returns v_i.

In this work we will use the protocol by Kate et al. for both cases, polynomial and vector commitments. For the latter, we will not only consider individual openings but also subset openings. In particular, let $\mathbb{H} = \{1, \omega, \ldots, \omega^{N-1}\}$ be a set of roots of unity and $\{\lambda_i(X)\}_{i=1}^{N}$ its corresponding Lagrange interpolation set, with vanishing polynomial $z_H(X)$. That is, $\lambda_i(\omega^{i-1}) = 1$ and $\lambda_i(\omega^j) = 0$ for all $j \neq i - 1$. We have that for some polynomial $H(X)$,

$$V(X) - s = (X - \omega^i)H(X) \text{ if and only if } V(\omega^i) = v_{i+1} = s.$$

For a polynomial $V_I(X) = \sum_{i \in I} s_i \tau_i(X)$ where s_{i+1} are claimed values for v_{i+1} and $\{\tau_i(X)\}_{i \in I}$ the Lagrange interpolation polynomials of the set $\{\omega^i\}_{i \in I}$,

$$V(X) - V_I(X) = \prod_{i \in I}(X - \omega^i)H(X) \text{ iff } V(\omega^i) = v_{i+1} = s_{i+1} \text{ for all } i \in I.$$

4.6.1 Multiple Openings

A KZG proof of opening can naturally be extended to open one polynomial in many points. Indeed, let $p(X)$ be a polynomial, $\vec{\alpha} \in \mathbb{F}^m$ a vector of opening points and $\vec{s}$ such that $s_i = p(\alpha_i)$ for all $i = 1, \ldots, m$. Define $C_{\vec{\alpha}}(X)$ as the unique polynomial of degree $m - 1$ such that $C_{\vec{\alpha}}(\alpha_i) = s_i$ for all $i \in [m]$. We have that $p(\alpha_i) = s_i$ for all $i = 1, \ldots, m$ if and only if there exists $q(X)$ such that

$$p(X) - C_{\vec{\alpha}}(X) = \prod_{i=1}^{m}(X - \alpha_i)q(X)$$

We can thus redefine the KZG prover and verifier the following way:

- $(s, \pi_{\mathsf{KZG}}) \leftarrow \mathsf{KZG.Open}\big(\mathsf{srs}_{\mathsf{KZG}}, p(X), \deg, \vec{\alpha}\big)$: Prover computes $\{\tau_i(X)\}_{i=1}^{m}$ the interpolation Lagrange polynomials for the set $\{\alpha_i\}_{i=1}^{m}$, $z_\alpha(X) = \prod_{i=1}^{m}(X - \alpha_i)$ and define $C_{\vec{\alpha}}(X) = \sum_{i=1}^{m} p(\alpha_i)\tau_i(X)$. Then, it computes
$$q(X) = \frac{p(X) - C_{\vec{\alpha}}(X)}{z_\alpha(X)} \ ,$$
sets $s_i = p(\alpha_i), [Q]_1 = [q(x)]_1$, and outputs $(\vec{s}, \pi_{\mathsf{KZG}} = [Q]_1)$.
- $1/0 \leftarrow \mathsf{KZG.Verify}\big(\mathsf{srs}_{\mathsf{KZG}}, \mathsf{C}, \deg, \vec{\alpha}, \vec{s}, \pi_{\mathsf{KZG}}\big)$: The verifier computes $\{\tau_i(X)\}_{i=1}^{m}$, $\mathsf{C}_{\vec{\alpha}} = [C_{\vec{\alpha}}(x)]_1$, $[z_\alpha(x)]_2$ and verifies
$$p(X) - C_{\vec{\alpha}}(X) = q(X)z_\alpha(X)$$
by making the pairing check
$$e(\mathsf{C} - \mathsf{C}_{\vec{\alpha}}, [1]_2) = e([Q]_1, [z_\alpha(x)]_2),$$
and outputs 1 if and only if the equation is satisfied and $\deg(p) \leq d$.

4.6.2 KZG for Bivariate Polynomials

For the protocol in Section 7.1 we will use bivariate polynomials, or polynomials of higher degree. What this mean is that, if we have a bivariate polynomial $P(X, Y)$ with degree up to $d_1 - 1$ in X and $d_2 - 1$ in Y then we require a universal setup with $d_1 d_2$ powers. We work with a version of KZG that uses a univariate setup because these are already available for multiple different curves (i.e. we do *not* need a specialist setup just for our protocol and can work with prior KZG setups).

We observe that, by using the KZG open algorithm, we can commit to $P(X, Y)$ as $[P(x^{d_2}, x)]_1$. We must open $P(X, Y)$ in two steps. First we *partially* open $P(X, Y)$ at some point $X = \alpha$ to a commitment $[P(\alpha, x)]_1$. The partial proof is given by a commitment $[w_\alpha(x^{d_2}, x)]$ to a partial witness

$$w_\alpha(X, Y) = \frac{P(X, Y) - P(\alpha, Y)}{X - \alpha}$$

We then fully evaluate $P(\alpha, X)$ at $Y = \beta$ via a standard KZG proof with a degree bound of $d_2 - 1$ on $[P(\alpha, x)]_1$.

4.6.3 Subset openings

By applying the ideas above to vector commitments, we have that the KZG scheme can be applied to prove openings to subsets rather than positions. Consider the subset of $\mathbb{H}$ $\{\omega^i\}_{i \in I} = \{\omega^{i_j}\}_{j=1}^m$ with $I \subset [N]$ such that $|I| = m$ and its vanishing polynomial $z_I(X) = \prod_{i \in I}(X - \omega^i)$. We can prove that a commitment $C = [C(x)]_1$ to vector $\vec{v}$ is such that $C(\omega^{i_j}) = s_j$, i.e., $v_{i_j+1} = s_j$ for $j = 1, \ldots, m$. The proof is given by $Q = [q(x)]_1$ for

$$q(X) = \frac{C(X) - \sum_{j=1}^m s_j \tau_j(X)}{z_I(X)} \ .$$

where

$$\tau_j(X) = \prod_{k=1, k \neq j}^{m} \frac{X - \omega^{i_k}}{\omega^{i_j} - \omega^{i_k}}$$

is the langrange polynomial evaluating to 0 at ω^{i_k} for $k \neq j$ and 1 at ω^{i_j}. The verifier checks the equation

$$C(X) - \sum_{j=1}^{m} s_j \tau_j(X) = q(X) z_I(X)$$

By making the pairing check

$$e(\mathsf{C} - [\sum_{j=1}^{m} s_j \tau_j(x)]_1, [1]_2) = e(Q, [z_I(x)]_2)$$

We will additionally use a result by Tomescu et al. [28] that allows the prover to compute Q in time $\mathcal{O}(m \log^2(m))$ given it already has stored proofs $Q_1, \ldots, Q_m$ that $c(\omega^{i_j}) = s_j$. Indeed the prover sets

$$q(X) = \sum_{j=1}^{m} \frac{q_j(X)}{\prod_{k=1, k \neq j}^{m}(\omega^{i_j} - \omega^{i_k})} \text{ and } Q = \sum_{j=1}^{m} \frac{Q_j}{\prod_{k=1, k \neq j}^{m}(\omega^{i_j} - \omega^{i_k})}$$

This is a correct computation of Q because

$$\begin{aligned} C(X) - \sum_{j=1}^{m} s_j \tau_j(X) &= C(X)(1 - \sum_{j=1}^{m} \tau_j(X)) + \sum_{j=1}^{m} (c(X) - s_j)\tau_j(X) \\ &= \sum_{j=1}^{m} (c(X) - s_j)\tau_j(X) \end{aligned}$$

where we are using that $(1 - \sum_{j=1}^{m} \tau_j(X)) = 0$ for any set of lagrange polynomials. Further

$$\sum_{j=1}^{m}(C(X) - s_j)\tau_j(X) = \sum_{j=1}^{m}(C(X) - s_j) \prod_{k=1,k\neq j}^{m} \frac{(X - \omega^{i_k})}{(\omega^{i_j} - \omega^{i_k})}$$
$$= \left(\sum_{j=1}^{m} \frac{(C(X) - s_j)}{(X - \omega^{i_j}) \prod_{k=1,k\neq j}^{m}(\omega^{i_j} - \omega^{i_k})}\right) \prod_{k=1}^{m}(X - \omega^{i_k})$$
$$= \left(\sum_{j=1}^{m} \frac{q_j(X)}{\prod_{k=1,k\neq j}^{m}(\omega^{i_j} - \omega^{i_k})}\right) \prod_{k=1}^{m}(X - \omega^{i_k})$$

The provers asymptotic time is thus dominated by the computation of $\prod_{k=1,k\neq j}^{m} \frac{1}{(\omega^{i_j} - \omega^{i_k})}$.

Remark 1. *We remark that precomputing all the proofs $Q_1, \ldots, Q_n$ that $c(\omega^{i_j}) = s_j$ can be achieved in time $\mathcal{O}(n \log n)$ using techniques by Feist and Khovratovich [14]. The overview of this technique by Tomescu et al. ([28], Section 3.4.4,* "Computing All u_i's Fast"*) is explained well.*

4.7 Proof of Opening of a Pedersen Commitment

Pedersen commitment schemes are a particular case of vector commitments. We will consider them for committing to single values in a zero knowledge way. Thus, the srs will additionally output $[h]_1$ for some secret h and the commitment to some element s is computed as $v[1]_1 + r[\mathsf{h}]_1 = [v + hr]$, for some randomly sampled $\mathsf{h} \in \mathbb{F}$. We suggest a standard Fiat-Shamired Sigma protocol [24] to demonstrate knowledge of v, r such that $\mathsf{cm} = [v + \mathsf{h}r]_1$ for some v, r:

$$R_{\mathsf{ped}} = \{(\mathsf{cm};\ (v, r)) : \quad \mathsf{cm} = [v + \mathsf{h}r]_1\}$$

The proof consists of $R = [s_1 + \mathsf{h}s_2]_1$, $t_1 = s_1 + vc$ and $t_2 = s_2 + rc$, where $c = \mathsf{H}(\mathsf{cm}, R)$ and s_1, s_2 are elements chosen by the verifier. At the end, the verifier checks that $R + c \cdot \mathsf{cm} = [t_1 + \mathsf{h}t_2]_1$.

5 Position-Hiding Linkable Vector Commitments

We introduce the concept of position-hiding linkable vector commitment schemes. Informally, two vector commitment schemes VC_1 and VC_2 are position-hiding linkable if a prover is able to convince a verifier that for a given commitments C corresponding to VC_1 and cm corresponding to VC_2, it is true that all the elements in the vector committed in cm are also elements of the vector committed in C.

Basicallly, position-hiding linkability allows the prover to extract or isolate in zero-knowledge elements from some public set or table, and later prove further attributes on them. This new primitive should satisfy three security notions: completeness, as usual; *linkability*, that captures the fact that if the proof verifies then there is no element committed in cm that is not also committed in C; and *position-hiding*, which holds only if no information about the set of elements in C that have been used to construct cm is leaked.

Definition 5.1 (Position-Hiding Linkability for Vector Commitments)**.** *Two vector commitment schemes VC_1 and VC_2 are position-hiding linkable if there exist algorithms $(\mathsf{Setup}_{\mathsf{link}}, \mathsf{Prove}_{\mathsf{link}}, \mathsf{Verify}_{\mathsf{link}}, \mathsf{Simulate}_{\mathsf{link}})$ that behave as follows,*

- $\mathsf{Setup}_{\mathsf{link}}(1^\lambda, d_1, d_2)$: *takes as input the security parameter, bounds on the length of vectors in VC_1 and VC_2, and outputs common parameters srs that include $\mathsf{srs}_1 = \mathsf{VC}_1.\mathsf{srs}$ and $\mathsf{srs}_2 = \mathsf{VC}_2.\mathsf{srs}$ as well as trapdoor x, including the corresponding trapdoors x_1 and x_2.*
- $\mathsf{Prove}_{\mathsf{link}}(\mathsf{srs}, r, r', \vec{v}, \vec{v}_I)$: *on input the srs, commitment randomness r to vector $\vec{v} \in \mathbb{F}^N$ and commitment randomness r' to $\vec{a} \in \mathbb{F}^m$, outputs a proof π that there exists some $I \subset [N]$ such that for all $j = 1, \ldots, m$, $a_j = v_i$ for some $i \in I$.*
- $\mathsf{Verify}_{\mathsf{link}}(\mathsf{srs}, \mathsf{C}, \mathsf{C}', \pi)$: *On input the srs, commitments C and C', and proof π, accepts or rejects.*

- $\mathsf{Simulate}_{\mathsf{link}}(x_1, x_2, \mathsf{C}, \mathsf{C}')$: *On input the trapdoors* x_1, x_2 *and commitments* C *and* C'*, outputs a simulated proof* π_{sim}*,*

and satisfy the following properties:

Completeness: *For all* N, t *with* $t \leq N$*, all* $\vec{v} \in \mathbb{F}^N$*, and all* $\vec{a} \in \mathbb{F}^m$ *it holds that:*

$$\Pr\left[\mathsf{Verify}_{\mathsf{link}}(\mathsf{srs}, \mathsf{C}, \mathsf{C}', \pi) = 1 \;\middle|\; \begin{array}{c}(\mathsf{srs}, x) \leftarrow \mathsf{Setup}_{\mathsf{link}}(1^\lambda, N, t);\\ \mathsf{C} \leftarrow \mathsf{Commit}(\mathsf{srs}_1, \vec{v}, r);\\ \mathsf{C}' \leftarrow \mathsf{Commit}(\mathsf{srs}_2, \vec{a}, r');\\ \pi \leftarrow \mathsf{Prove}_{\mathsf{link}}(\mathsf{srs}, r, r', \vec{v}, \vec{a})\end{array}\right] = 1.$$

Linkability *For all* N, m *with* $m \leq N$*, and all PPT adversaries, there exists an extractor* $\mathcal{X}_{\mathcal{A}}$ *such that:*

$$\Pr\left[\begin{array}{c}\mathsf{Verify}_{\mathsf{link}}(\mathsf{srs}, \mathsf{C}, \mathsf{C}', \pi) = 1 \;\wedge\\ |\vec{v}| = N \;\wedge\\ \big(\exists\ j \in [m]\ s.t.\ a_j \neq c_i\ \forall i \in [N]\big) \vee\\ \mathsf{VC}_2.\mathsf{Commit}(\mathsf{srs}_2, \vec{a}, r') \neq \mathsf{C}'\big)\end{array} \;\middle|\; \begin{array}{c}(\mathsf{srs}, x) \leftarrow \mathsf{Setup}_{\mathsf{link}}(1^\lambda, N, m);\\ \vec{v} \leftarrow \mathcal{A}(\mathsf{srs});\\ \mathsf{C} \leftarrow \mathsf{VC}_1.\mathsf{Commit}(\mathsf{srs}_1, \vec{v});\\ (\pi, \mathsf{C}') \leftarrow \mathcal{A}(\mathsf{srs}_1, \mathsf{srs}_2, \mathsf{C});\\ (\vec{a}, r') \leftarrow \mathcal{X}_{\mathcal{A}}(\mathsf{C}', \pi)\end{array}\right] = \mathsf{negl}(\lambda).$$

Position-Hiding *For all* N, m *with* $m \leq N$*, for all* $\vec{v}$ *and* $\vec{a}$*, all PPT adversaries* $\mathcal{A}$*, there exists a PPT algorithm* $\mathsf{Simulate}_{\mathsf{link}}$ *such that:*

$$\left[\mathcal{A}(\mathsf{srs}, \mathsf{C}, \mathsf{C}', \pi) = 1 \;\middle|\; \begin{array}{c}(\mathsf{srs}, x) \leftarrow \mathsf{Setup}_{\mathsf{link}}(1^\lambda, N, m)\\ \mathsf{C} \leftarrow \mathsf{Commit}(\mathsf{srs}_1, \vec{v}, r)\\ \mathsf{C}' \leftarrow \mathsf{Commit}(\mathsf{srs}_2, \vec{a}, r')\\ \pi \leftarrow \mathsf{Prove}_{\mathsf{link}}(\mathsf{srs}, r, r', \vec{v}, \vec{a})\end{array}\right] \approx_c \left[\mathcal{A}(\mathsf{srs}, \mathsf{C}, \mathsf{C}', \pi_{\mathsf{sim}}) = 1 \;\middle|\; \begin{array}{c}(\mathsf{srs}, x) \leftarrow \mathsf{Setup}_{\mathsf{link}}(1^\lambda, N, m)\\ \mathsf{C} \leftarrow \mathsf{Commit}(\mathsf{srs}_1, \vec{v}, r)\\ \mathsf{C}' \leftarrow \mathsf{Commit}(\mathsf{srs}_2, \vec{a}, r')\\ \pi_{\mathsf{sim}} \leftarrow \mathsf{Simulate}_{\mathsf{link}}(x, \mathsf{C}, \mathsf{C}')\end{array}\right]$$

In the next sections, we introduce position-hiding linkability for KZG commitments of arbitrary size and pedersen commitments for single elements (Section 6), as well as for two KZG commitments (Section 7).

6 Linking Vectors with Elements

In this section we present a method to link a commitment C to a vector $\vec{c} \in \mathbb{F}^N$ computed as $\mathsf{C} = [C(x)]_1$ with $C(X) = \sum_{i=1}^{N} c_i \lambda_i(X)$, to a pedersen commitment cm. By this we mean a method for a prover to convince a verifier that there exists an i such that C opens to v at some Nth rooth of unity ω^i and $\mathsf{cm} = [v + \mathsf{h}r]_1$.

We will consider two groups of roots of unity:

- $\mathbb{H} = \{1, \omega, \ldots, \omega^{N-1}\}$ of size N with $\omega^N = 1$, Lagrange interpolation polynomials $\{\lambda_i(X)\}_{i=1}^N$ where $\lambda_i(\omega^{i-1}) = 1$ and $\lambda_i(\omega^j) = 0$ if $j \neq i - 1$, and vanishing polynomial $z_H(X)$.
- $\mathbb{V}_n = \{1, \nu, \ldots, \nu^{n-1}\}$ of size $n = \log(N) + 6$ with $\nu^n = 1$, Lagrange interpolation polynomials $\{\rho_s(X)\}_{s=1}^n$ and vanishing polynomial $z_{V_n}(X)$.

Our construction can be divided into three main components. The first one is a proof of knowledge for the element v committed in cm, that is a proof for relation R_{ped} as defined in Section 4.7. The second is a modified protocol for computing blinded versions of KZG openings for statements $C(\omega^i) = v$ that does not reveal the coordinate $i + 1$ or the evaluation v, which we describe below. The high-level idea here is to re-randomize a regular KZG opening with an additional blinding factor. Our third component then proves that the re-randomized vanishing polynomial used for the KZG opening is well-formed, i.e., a NIZK argument (as in Def. A.1) for the relation

$$R_{\mathsf{unity}} = \left\{(\mathsf{srs}, [z]_2;\ (a, i)) : \quad [z]_2 = [a(x - \omega^i)]_2 \ \wedge\ (\omega^i)^N = 1\right\}.$$

6.1 Our Blinded Evaluation Construction

Our prover takes $(r' = \perp, \vec{c})$ and (r, v) as input, where the first tuple represents the vector inside the (deterministic) KZG commitment and the second tuple represents the randomness and value for the pedersen commitment. Let $C(X) = \sum_{i=1}^{N} c_i \lambda_i(X)$ be the polynomial encoding vector $\vec{c}$. In a regular KZG opening for position $i+1$, the prover would compute $q(X) = \frac{C(X)-v}{x-\omega^i}$ and reveal $Q = [q(x)]_1$. Instead, our prover computes a special kind of obfuscated commitment to ω^i by selecting a random a and committing to $z(X) = aX - b = a(X - \omega^i)$ where $\omega^i = \frac{b}{a}$, i.e. the commitment contains $[z]_2 = [z(x)]_2$. Note that this is simply the KZG cofactor $X - \omega^i$ multiplied by a blinding factor a. The blinding factor is necessary, because the set $\{\omega^i\}_{i=0}^{m-1}$ is polynomial sized, so revealing $[x - \omega^i]_1$ would allow the verifier to do a brute force search to find the index. The prover then computes $[T]_1 = [T(x)]_1$ and $[S]_2 = [S(x)]_2$, where

$$T(X) = \frac{q(X)}{a} + \mathsf{h}s \quad \text{and} \quad S(X) = -r - sz(X)$$

and s is a uniformly random value chosen by the prover. $T(X)$ is the KZG quotient polynomial $q(X)$ divided by a (the blinding factor above) to compensate for $z(X)$ having that blinding factor. The additional term $[\mathsf{h}s]_1$ mixed in to fully blind the evaluation $[\frac{q(X)}{a}]_1$ and preserve zero-knowledge. $[S]_2$ is a term that compensates for the h terms in both $[T]_1$ and cm. In the pairing equation that checks these points, $[S]_2$ will be paired with h to ensure that it can only cancel out terms containing h and cannot make incorrect quotient polynomials appear correct.

We also provide two proofs of knowledge π_{ped} and π_{unity} as described in Section 4.7 and Section 6.2 respectively. The proof π_{ped} is for v, r such that $\mathsf{cm} = [v + \mathsf{h}r]_1$. The proof π_{unity} is for a, b such that $[z]_2 = [ax - b]_2$ and $a^N = b^N$. The verifier checks the pairing equation

$$e(\mathsf{C} \cdot \mathsf{cm}^{-1}, [1]_2) = e([T]_1, [z]_2) + e([h]_1, [S]_2).$$

This equation checks that, for the polynomials $C(X), T(X), z(X), S(X)$ encoded in $\mathsf{C}, [T]_1, [z]_2$, and $[S]_2$ respectively, it holds that

$$C(X) - v - \mathsf{h}r = T(X)z(X) + \mathsf{h}S(X).$$

Now with $T(X) = \frac{q(X)}{a} + s\mathsf{h}$, $z(X) = a(X - \omega^i)$, and $S(X) = -r - sz(X)$, this is

$$C(X) - v - \mathsf{h}r = \left(\frac{q(X)}{a} + s\mathsf{h}\right) z(X) - \mathsf{h}r - \mathsf{h}sz(X) \iff C(X) - v = \left(\frac{q(X)}{a}\right) z(X).$$

The full description of our protocol is given in Figure 1.

Theorem 1. *Let R_{ped} and R_{unity} be relations for which zero-knowledge argument of knowledge systems are given. The construction in Figure 1 implies position-hiding linkability for the commitment schemes corresponding to C and cm in the algebraic group model under the q-SDH and dlog assumptions.*

The proof is in Appendix B.

6.2 Correct computation of $z(X)$

The purpose of this section is provide a zero-knowledge proof of knowledge for relation R_{unity}, i.e. that the prover knows a, b such that $[z]_2 = [ax - b]_2$ and $a^N = b^N$. This proof is used as a subprotocol in Fig. 1's construction of a linkable vector commitment in Section 6.

In order to prove that $\frac{a}{b}$ is inside the evaluation domain (i.e. is a root of unity) in zero-knowledge we add another polynomial $f(X)$ of degree $n = \log(N) + 6$. The polynomial $f(X)$ essentially recovers $\frac{a}{b}$ from $[z]_2$ and then includes its powers 2^i until $i = \log(N)$. It will be enough then to prove that (i) $f(X)$ is correctly formed with respect to $[z]_2$, (ii) it does indeed contain all 2-powers of $\frac{a}{b}$, and (iii) the coefficient corresponding to $(\frac{a}{b})^{2^{\log(N)}} = (\frac{a}{b})^N$ equals 1.

The core of our construction is the following lemma, that we prove in Appendix C:

Lemma 1. *Let $z(X)$ be a polynomial of degree 1, $n = \log(N) + 6$ and σ such that $\sigma^n = 1$. If there exists a polynomial $f(X) \in \mathbb{F}[X]$ such that*

Prover: Sample blinders $a, s \xleftarrow{\$} \mathbb{F}$

Compute $C(X) = \sum_{i=1}^{N} c_i \lambda_i(X)$, encoding of $\vec{c}$ and $\mathsf{cm} = v[1]_1 + r[\mathsf{h}]_1$

Define

$$z(X) = a(X - \omega^i), \quad T(X) = \frac{C(X) - v}{z(X)} + s\mathsf{h}, \quad S(X) = -r - sz(X)$$

$\pi_{\mathsf{ped}} \leftarrow \mathsf{Prove}(R_{\mathsf{ped}}, \mathsf{cm}, (v, r))$

$\pi_{\mathsf{unity}} \leftarrow \mathsf{Prove}(R_{\mathsf{unity}}, (\mathsf{srs}, [z]_2), (a, a\omega^i))$

Set $[z]_2 = [z(x)]_2, [T]_1 = [T(x)]_1, [S]_2 = [S(x)]_2$ and return $([z]_2, [T]_1, [S]_2, \pi_{\mathsf{ped}}, \pi_{\mathsf{unity}})$

Verifier: Accept if and only if the following conditions hold

$$e(\mathsf{C} - \mathsf{cm}, [1]_2) = e([T]_1, [z]_2) + e([\mathsf{h}]_1, [S]_2)$$
$$1 \leftarrow \mathsf{Verify}_{\mathsf{ped}}(\mathsf{srs}, \mathsf{cm}, \pi_{\mathsf{ped}})$$
$$1 \leftarrow \mathsf{Verify}_{\mathsf{unity}}(\mathsf{srs}, [z]_2, \pi_{\mathsf{unity}})$$

Figure 1: Zero-knowledge proof of membership. Shows that (v, r) is an opening of cm and that C opens to v at ω^i.

1. $f(X) = z(X)$ *for* $1, \sigma$.
2. $f(\sigma^2)(1 - \sigma) = f(1) - f(\sigma)$
3. $f(\sigma^3) = \sigma f(\sigma^2) - f(\sigma)$
4. $f(\sigma^4) f(\sigma^3) = f(\sigma^2)$
5. $f(\sigma^{4+\log(N)}) = f(\sigma^{5+\log(N)}\sigma^{-1}) = 1$
6. $f(\sigma^{4+i+1}) = f(\sigma^{4+i})^2$, *for all* $i = 0, \ldots, \log(N) - 1$

Then, $z(X) = aX - b$, *where* $\frac{b}{a}$ *is an* N*-th root of unity.*

In our protocol the prover will construct the polynomial $f(X)$ as

$$f(X) = (a - b)\rho_1(X) + (a\sigma - b)\rho_2(X) + a\rho_3(X) + b\rho_4(X) + \sum_{i=0}^{\log(N)} \left(\frac{a}{b}\right)^{2^i} \rho_{5+i}(X). \tag{3}$$

and commit to it in zero-knowledge. Then, it will show it is correct by comparing $f(\sigma^i)$ with the corresponding values from the constraints in Lemma 1. Namely, for some α chosen by the verifier, it sets $\alpha_1 = \sigma^{-1}\alpha$, $\alpha_2 = \sigma^{-2}\alpha$ and sends $v_1 = f(\alpha_1)$ and $v_2 = f(\alpha_2)$ along with the corresponding proofs of opening. Given v_1, v_2 it then shows that the following polynomial, which proves the constraints in Lemma 1, evaluates to 0 in α:

$$\begin{aligned} p_\alpha(X) = &- h(X) z_{V_n}(\alpha) + \big(f(X) - z(X)\big)\big(\rho_1(\alpha) + \rho_2(\alpha)\big) + \big((1 - \sigma)f(X) - f(\alpha_2) + f(\alpha_1)\big)\rho_3(\alpha) \\ &+ \big(f(X) + f(\alpha_2) - \sigma f(\alpha_1)\big)\rho_4(\alpha) + \big(f(X) f(\alpha_1) - f(\alpha_2)\big)\rho_5(\alpha) \\ &+ \big(f(X) - f(\alpha_1) f(\alpha_1)\big) \prod_{i \notin [5, \ldots, 4+\log(N)]} (\alpha - \sigma^i) + \big(f(\alpha_1) - 1\big)\rho_n(\alpha). \end{aligned}$$

Note that the polynomials that are already evaluated in α in $p_\alpha(X)$ are thus that either the verifier can compute its own, or are opened by the prover.

Using v_1, v_2, the commitments to $h(X), f(X)$ and after computing $\rho_i(\alpha)$ for $i = 1, 2, 3, 4, n-1, n$ and $\prod_{i \notin [5, \ldots, 4+\log(N)]}(\alpha - \sigma^i)$, the verifier computes a commitment $[P]_1$ to $p_\alpha(X)$ and checks that (i) v_1, v_2

are correct openings of $f(X)$ at $\alpha_1 = \sigma^{-1}\alpha$ and $\alpha_2 = \sigma^{-2}\alpha$, (*ii*) 0 is a correct opening of $p_\alpha(X)$ at α, and (*iii*) $[z]_2$ has degree 1.

For this last check, we ask the prover to include a term $X^{d-1}z(X)$ in $h(X)$ and then the verifier computes $[P]_1$ without the terms including $z(X)$, i.e, without $-X^d z(X) z_{V_n}(\alpha) - z(X)(\rho_1(\alpha) + \rho_2(\alpha))$. It will instead add them in the group via the pairing later, to assure that it cannot be the case that $\deg(z) > 1$, unless $\deg(p_\alpha) > d$, which is not possible under the AGM.

We describe the protocol in Fig. 2.

Theorem 2. *The protocol in Fig. 2 is a zero-knowledge and knowledge-sound argument (as defined in Def.A.1) for relation* $\mathsf{R}_{\mathsf{unity}}$ *if KZG is a sound polynomial commitment scheme, under the the Algebraic Group and Random Oracle models.*

The proof is in Appendix D.

7 Lookup tables for hiding values

In this section we present position-hiding linkability for KZG vector commitment schemes. The aim is to prove that a commitment cm contains a *subset* of some larger vector committed in C. We refer to a subset as opposite to subvector since our scheme proves that all the elements committed in cm are also committed in C, but with no specific order and possible repetitions. This is essentially a lookup table if we consider that C contains the honestly generated table.

Concrete efficiency. Our lookup proof has preprocessing time for C of $N \log N$ $\mathbb{G}_2$, for N the size of the table. Prover time is $m \log(N)$ for m the size of the subset, proof size is constant and verifier time $\log \log N$ scalar multiplications and constant number of pairing checks; additionally, update of proofs can be done in $O(N)$ $\mathbb{G}_2$ operations;

Preliminaries We will consider three evaluation domains

1. $\mathbb{H} = \{1, \omega, \ldots, \omega N - 1\}$ is a group of roots of unity with Lagrange and vanishing polynomials $\{\lambda_i(X)\}_{i=1}^N, z_H(X)$.
2. For subset $\mathbb{H}_I = \{\omega^i\}_{i \in I}$ of $\mathbb{H}$ defined by $I \subset [N]$, we define $\{\tau_i(X)\}_{i \in I}$ as its interpolation Lagrange polynomials with degree $|I| - 1$. Note that typically $\mathbb{H}_I$ is not a subgroup.
3. For some constant m that bounds the size of the vector committed in cm, we consider another group of roots of unity $\mathbb{V}_m = \{1, \nu, \ldots, \nu^{m-1}\}$, where $\nu^m = 1$, as well as its Lagrange and vanishing polynomials, $\{\mu_j(X)\}_{j=1}^m$ and $z_{V_m}(X)$.

Our scheme uses a subprotocol a NIZK argument of knowledge for relation $\mathsf{R}_{\mathsf{unity}}$,

$$R_{\mathsf{unity}} = \left\{ (\mathsf{srs}, [z_I]_2, N;\ (I, r)) :\ I \subset [N]\ \wedge\ [z_I]_2 = r \prod_{i \in I} (X - \omega^i),\ \text{with } (\omega^i)^N = 1,\ \forall\ i \in I \right\}$$

In our protocol, the prover takes as input a commitment $C(X) = \sum_{i=1}^N c_i \lambda_i(X)$ to the lookup table $\vec{c}$, a structured reference string srs, and a commitment

$$\mathsf{cm} = [\phi(x)]_1 = \left[\sum_{j=1}^m a_j \mu_j(x) + a_{m+1} z_{V_m}(x) \right]_1$$

to some vector $\vec{a}$ and the opening witness $\vec{a} = (a_1, \ldots, a_{m+1})$. Here a_m is a random field element that blinds cm. The prover must show that it knows an opening $\phi(X) = \sum_{j=1}^m a_j \mu_j(X) + a_{m+1} z_{V_m}(X)$ to cm such that $a_j \in \{c_i\}_{i=1}^N$ for all $1 \le j \le m$. The full argument is given in Fig. 3 and can be divided into three steps.

First, the prover considers the subset $I \subset [N]$ such that for all $j = 1, \ldots, m$, $a_j = c_{i+1}$ for some $i \in I$, and constructs the subvector $\vec{c}_I = (c_{i+1})_{i \in I}$ of $\vec{c}$. It commits to it in the Lagrange basis corresponding

Common input: $[z]_2$

Prover: Sample $r_0, r_1, r_2, r_3 \xleftarrow{\$} \mathbb{F}$ and let $r(X) \leftarrow r_1 + r_2 X + r_3 X^2$

$$f(X) = (a-b)\rho_1(X) + (a\sigma - b)\rho_2(X) + a\rho_3(X) + b\rho_4(X) + \sum_{i=0}^{\log(N)} \Big(\frac{a}{b}\Big)^{2^i} \rho_{5+i}(X) + r_0\rho_{5+\log(N)}(X) + r(X)z_{V_n}(X),$$

$$\begin{aligned} p(X) = {} & \big(f(X) - (aX-b)\big)\big(\rho_1(X) + \rho_2(X)\big) + \big((1-\sigma)f(X) - f(\sigma^{-2}X) + f(\sigma^{-1}X)\big)\rho_3(X) \\ & + \big(f(X) + f(\sigma^{-2}X) - \sigma f(\sigma^{-1}X)\big)\rho_4(X) + \big(f(X)f(\sigma^{-1}X) - f(\sigma^{-2}X)\big)\rho_5(X) \\ & + \big(f(X) - f(\sigma^{-1}X)f(\sigma^{-1}X)\big) \prod_{i \notin [5;\ 4+\log(N)]} (X - \sigma^i) + \big(f(\sigma^{-1}X) - 1\big)\rho_n(X), \end{aligned}$$

Set $\hat{h}(X) = \frac{p(X)}{z_{V_n}(X)}$, $h(X) = \hat{h}(X) + X^{d-1}z(X)$ and output $([F]_1 = [f(x)]_1, [H]_1 = [h(x)]_1)$.

Verifier : Send challenge $\alpha \in \mathbb{F}$

Prover : $\alpha_1 = \sigma^{-1}\alpha$, $\alpha_2 = \sigma^{-2}\alpha$;

$$\begin{aligned} p_\alpha(X) = {} & -z_{V_n}(\alpha)h(X) + \big(f(X) - z(X)\big)\big(\rho_1(\alpha) + \rho_2(\alpha)\big) + \big((1-\sigma)f(X) - f(\alpha_2) + f(\alpha_1)\big)\rho_3(\alpha) \\ & + \big(f(X) + f(\alpha_2) - \sigma f(\alpha_1)\big)\rho_4(\alpha) + \big(f(X)f(\alpha_1) - f(\alpha_2)\big)\rho_5(\alpha) \\ & + \big(f(X) - f(\alpha_1)f(\alpha_1)\big) \prod_{i \notin [5;\ 4+\log(N)]} (\alpha - \sigma^i) + \big(f(\alpha_1) - 1\big)\rho_n(\alpha), \end{aligned}$$

Compute

$$\begin{aligned} ((v_1, v_2), \pi_1) &\leftarrow \mathsf{KZG.Open}(\mathsf{srs}_{\mathsf{KZG}}, f(X), \deg = \bot, (\alpha_1, \alpha_2)) \\ (0, \pi_2) &\leftarrow \mathsf{KZG.Open}(\mathsf{srs}_{\mathsf{KZG}}, p_\alpha(X), \deg = \bot, \alpha), \end{aligned}$$

and output $\big(v_1, v_2, \pi_1, \pi_2\big)$.

Verifier : Set $\alpha_1 = \sigma^{-1}\alpha$; $\alpha_2 = \sigma^{-2}\alpha$,

$$\begin{aligned} [P]_1 = {} & -z_{V_n}(\alpha)[H]_1 + \big(\rho_1(\alpha) + \rho_2(\alpha)\big)[F]_1 + \rho_3(\alpha)\big((1-\sigma)[F]_1 + v_1 - v_2\big) + \rho_4(\alpha)\big([F]_1 + v_2 - \sigma v_1\big) \\ & + \rho_5(\alpha)\big(v_1[F]_1 - v_2\big) + \rho_n(\alpha)\big(v_1 - 1\big) + \prod_{i \notin [5,\ldots,4+\log(N)]} (\alpha - \sigma^i)\big([F]_1 - v_1^2\big), \end{aligned}$$

Parse $\pi_2 = [q]_1$ and accept if and only if

$$\begin{aligned} 1 &\leftarrow \mathsf{KZG.Verify}\big(\mathsf{srs}_{\mathsf{KZG}}, [F]_1, \deg = \bot, (\alpha_1, \alpha_2), (v_1, v_2), \pi_1\big), \\ e\big([P]_1, [1]_2\big) + e\big(-(\rho_1(\alpha) + \rho_2(\alpha)) - z_{V_n}(\alpha)[x^{d-1}]_1, [z]_2\big) &= e\big([q]_1, [x-\alpha]_2\big) \end{aligned}$$

Figure 2: NIZK argument of knowledge for $\mathsf{R}_{\mathsf{unity}}$ and $\deg(z) \leq 1$.

to $\{\omega^i\}_{i\in I}$; namely, $C_I(X) = \sum_{i\in I} c_{i+1}\tau_i(X)$. Basically, the prover isolates the elements of $\vec{c}$ that will compare with $\vec{a}$ so they can work with polynomials of smaller degree.

To convince the verifier that all the elements in $C_I(X)$ are elements of $C(X)$, it provides commitments to $z_I(X), H_1(X)$ such that

$$C(X) - C_I(X) = z_I(X)H_1(X). \tag{4}$$

Here is the place where the precomputation is used: $C(X)$ has degree N and so does $H_1(X)$. In order to compute a commitment to $H_1(X)$, we use the $O(N \log N)$ method described in Section 4.6.3. This is at the same time the most expensive step in updating a proof whenever $C(X)$ is changed. However, if c_i values are updated in known order, and we precompute an opening for τ_i, then whenever new c_i is available all openings can be updated in $O(N)$ time, hence the claimed update cost.

Our challenge now is hiding $C_I(X)$ and $z_I(X)$ from the verifier without breaking soundness. In our solution the prover first demonstrates that $z_I(X)$ is of the right form, meaning it is the vanishing polynomial of some subset I of $\mathbb{H}$; specifically, we need not only a hiding commitment but also a zero-knowledge proof of well formation of $z_I(X)$.

We divide the proof of well formation of $z_I(X)$ in two steps. First, the prover creates the polynomial $u(X) = \sum_{j=1}^{m} \omega^{i_j}\mu_j(X)$ of degree $m-1$ whose coefficients are the roots of unity $\{\omega^i\}_{i\in I}$ and prove, in zero knowledge, its well formation. For that, it demonstrates that for all $\nu^j \in \mathbb{V}$ it is the case that $(u(\nu^j))^N = 1$, via a call to a subprotocol Π_{unity} that we describe in Section 7.1. This guarantees that $u(X)$ is a commitment to elements in $\mathbb{H}$.

On input a commitment to $u(X)$ as above and given that $u(X)$ passes the verification of Π_{unity}, we prove well formation of $z_I(X)$. To achieve this we use the fact that all the coefficients of $u(X)$ in the basis $\{\mu_j(X)\}_{j=1}^m$ are roots of $z_I(X)$. For that, prover convinces verifier that

$$z_I(u(X)) = z_{V_m}(X)H_2(X), \text{ for some polynomial } H_2(X). \tag{5}$$

Finally, note that $C_I(X)$ has been committed to in an unknown-to-the-verifier Lagrange basis, which is $\{\tau_i(X)\}$. So the last step of our argument consists on linking the commitment to $C_I(X)$ with $[\phi(x)]_1$, which is an input to the argument and a commitment to the same element in a known basis. The prover does so by providing $H_3(X)$ such that

$$C_I(u(X)) - \phi(X) = z_{V_m}(X)H_3(X). \tag{6}$$

In order to achieve zero-knowledge, upon receiving an aggregation challenge χ from the verifier, prover actually provides one commitment $[H_2]_1 + \chi[H_3]_1$ to prove equations 5 and 6 together.

Note that for equation 4 to be satisfied, $C_I(X)$ cannot take more than once each of the coefficients of $C(X)$. On the other hand, when linking $C_I(X)$ and $\phi(X)$ through equation 6, we can only prove that all the coefficients of $\phi(X)$ in the basis $\{\mu_j(X)\}_{j=1}^m$ are also coefficients of $C_I(X)$ in the basis $\{\tau_i(X)\}_{i\in I}$, but we cannot say in which order or how many times each of them appears. At the end, what we get, is a lookup table argument that assures that some element $[\phi(x)]_1$ is a commitment in the Lagrange basis $\{\mu_j(X)\}_{j=1}^m$ to some vector $\vec{a} = (a_1, \ldots, a_m)$ such that for all $j = 1, \ldots, m$ there exists some $i_j \in I$ such that $a_j = c_{i_j+1}$, i.e., a lookup table for potentially repeated indexes.

Theorem 3. *Suppose that the argument of Fig. 3 is instantiated with a knowledge-sound scheme for relation $\mathcal{R}_{\mathsf{unity}}$. Then in the AGM with non-programmable ROs, either the argument of Fig. 3 implies linkability for the vector commitment schemes of C and cm, or there exists an adversary that breaks the q-SDH assumption.*

The proof is in Appendix. E

Subtables There is another nice feature that can be derived by the protocol in Fig. 3 and is the creation of sub-lookup tables. Namely, for some $I \subset [N]$, prover generates $t(X) = \prod_{i\in I}(X - c_i)$. To prove well formation of it, after having some $C_I(X)$ that has been proven correct, it shows that there exists some $H_3(X)$ such that

$$t(\tilde{C}_I(X)) = z_{V_m}(X)H_3(X).$$

Then, for any polynomial $a(X)$ of degree up to $m-1$, if there exists $H_4(X)$ such that

$$t(a(X)) = z_{V_m}(X)H_4(X),$$

then the coefficients of $a(X)$ in the basis $\{\mu_j(X)\}_{j=1}^m$ are coefficients of $C_I(X)$ in basis $\{\tau_i(X)\}_{i\in I}$, with no specific order and potential repetitions.

Common input: $C = [C(x)]_1$, for $C(X) = \sum_{i=1}^{N} c_i \lambda_i(X)$ and $\mathsf{cm} = [\phi(x)]_1$.

Prover: Take as input srs and $\phi(X)$ and proof $[Q(x)]_2$ attesting that $\{c_{i+1}\}_{i\in I}$ are openings of C. I.e., a commitment to $Q(X) = \frac{C(X) - \sum_{i\in I} c_{i+1}\tau_i(X)}{\prod_{i\in I}(X-\omega^i)}$.

- Choose blinders $r_1, r_2, r_3, r_4, r_5, r_6, r_7 \xleftarrow{\$} \mathbb{F}$ uniformly at random.
- For $\mathbb{H}_I = \{w^i\}_{i\in I}$, compute the interpolation polynomials $\{\tau_i(X)\}_{i\in I}$.
- Define $z_I(X) = r_1 \prod_{i\in I}(X - \omega^i)$ and $C_I(X) = \sum_{i\in I} c_{i+1}\tau_i(X) + (r_2 + r_3 X + r_4 X^2) z_I(X)$.
- Find $[H_1(x)]_2 = [r_1^{-1} Q(x) - (r_2 + r_3 x + r_4 x^2)]_2$ such that $C(X) - C_I(X) = z_I(X) H_1(X)$.
- Define ω_{i_j} as the jth element in $\{\omega^i\}_{i\in I}$ and compute

$$u(X) = \sum_{j=1}^{m} \omega^{i_j} \mu_j(X) + (r_5 + r_6 X + r_7 X^2) z_{V_m}(X).$$

- Compute a proof π_{unity} that $[u]_1$ has been correctly computed as in Fig. 4
- Output $[C_I]_1 = [C_I(x)]_1$, $[z_I]_1 = [z_I(x)]_1$, $[u]_1 = [u(x)]_1$, $[H_1]_2 = [H_1(x)]_2$, π_{unity}.

Verifier: Send challenge $\chi \in \mathbb{F}$

Prover:
- Find $H_2(X)$ such that $z_I(u(X)) + \chi(C_I(u(X)) - \phi(X)) = z_{V_m}(X) H_2(X)$
- Output $[H_2]_1 = [H_2(x)]_1$,.

Verifier : Send challenge $\alpha \in \mathbb{F}$

Prover : Compute

$$\begin{aligned}
p_1(X) &\leftarrow z_I(X) + \chi C_I(X) \\
p_2(X) &\leftarrow z_I(u(\alpha)) + \chi(C_I(u(\alpha)) - \phi(X)) - z_{V_m}(\alpha) H_2(X) \\
(v_1, \pi_1) &\leftarrow \mathsf{KZG.Open}(\mathsf{srs}_{\mathsf{KZG}}, u(X), \deg = \perp, \alpha) \\
(v_2, \pi_2) &\leftarrow \mathsf{KZG.Open}(\mathsf{srs}_{\mathsf{KZG}}, p_1(X), \deg = \perp, v_1) \\
(0, \pi_3) &\leftarrow \mathsf{KZG.Open}(\mathsf{srs}_{\mathsf{KZG}}, p_2(X), \deg = \perp, \alpha)
\end{aligned}$$

Output $\big(v_1, v_2, \pi_1, \pi_2, \pi_3\big)$.

Verifier : Compute $[P_1]_1 \leftarrow [z_I]_1 + \chi[C_I]_1$ and $[P_2]_1 \leftarrow v_2 - \chi\mathsf{cm} - z_{V_m}(\alpha)[H_2]_1$.

Accept if and only if (i) $V_{\pi_{\mathsf{unity}}}$ accepts, (ii)

$$\begin{aligned}
1 &\leftarrow \mathsf{KZG.Verify}\big(\mathsf{srs}_{\mathsf{KZG}}, [u]_1, \deg = \perp, \alpha, v_1, \pi_1\big) \\
1 &\leftarrow \mathsf{KZG.Verify}\big(\mathsf{srs}_{\mathsf{KZG}}, [P_1]_1, \deg = \perp, v_1, v_2, \pi_2\big) \\
1 &\leftarrow \mathsf{KZG.Verify}\big(\mathsf{srs}_{\mathsf{KZG}}, [P_2]_1, \deg = \perp, \alpha, 0, \pi_3\big), \text{ and } (iii)
\end{aligned}$$

$$e\big([C]_1 - [C_I]_1, [1]_2\big) = e\big([z_I]_1, [H_1]_2\big) \tag{7}$$

Figure 3: Lookup table for non-repeated indexes that uses a proof for $\mathsf{R}_{\mathsf{unity}}$ as blackbox.

7.1 Multi-Unity Proof or Proving well formation of $u(X)$

Let $u(X) = \sum_{j=1}^{m} \omega^{i_j} \mu_j(X) + r(X) z_{V_m}(X)$, where ω^{i_j} is the j-th element in I. The aim of this section is to prove in zero-knowledge that $u(X)$ is well formed. Namely, that $u(X) = \sum_{j=1}^{m} u_j \mu_j(X) + r(X) z_{V_m}(X)$ is such that all its coefficients are elements in $\mathbb{H}$ and thus, they are all Nth roots of unity, or what is the same, that $u_j^N = 1$ for all $j = 1, \ldots, m$.

For this argument, we will consider another group of roots of unity $\mathbb{V}_n = \{\sigma, \ldots, \sigma^{n-1}, \sigma^n = 1\}$ of size $n = \log(N)$, Lagrange interpolation polynomials $\{\rho_s(X)\}_{s=1}^{n}$ and vanishing polynomial $z_{V_n}(X)$.

Techniques. The prover first defines $\vec{u}_0 = (u_1, \ldots, u_m) \in \mathbb{F}^m$ to be the vector whose elements are the coefficients of $u(X)$. They then iteratively define $\vec{u}_j = \vec{u}_{j-1} \circ \vec{u}_{j-1}$. In other words they set

- $\vec{u}_1 = \vec{u}_0 \circ \vec{u}_0 = (u_1^2, \ldots, u_m^2)$;
- $\vec{u}_2 = \vec{u}_1 \circ \vec{u}_1 = (u_1^{2^2}, \ldots, u_m^{2^2})$;
- $\vec{u}_j = \vec{u}_{j-1} \circ \vec{u}_{j-1} = (u_1^{2^j}, \ldots, u_m^{2^j})$

They then must prove three conditions to the verifier: (i) $\vec{u}_0$ consists on the coefficients of $u(X)$, (ii) equation $\vec{u}_j = \vec{u}_{j-1} \circ \vec{u}_{j-1}$ holds for all $j = 1, \ldots, n-1$ and (iii) $\vec{u}_{n-1} \circ \vec{u}_{n-1} = \vec{1}$. Together this gives that all the coefficients u_j are Nth roots of unity.

As we are working with encodings as polynomials rather than vectors, the prover sets $u_0(X) = u(X)$, $u_n(X) = \mathsf{id}(X)$ (for $\mathsf{id}(X)$ the polynomial that evaluates to 1 over $\mathbb{V}_m$), and shows to the verifier that each of the following equations hold:

$$\begin{aligned} u(X)u(X) - u_1(X) &\equiv z_{V_m}(X) H_1(X), \\ &\vdots \\ u_{n-1}(X)u_{n-1}(X) - \mathsf{id}(X) &\equiv z_{V_m}(X) H_n(X), \end{aligned}$$

To aggregate all of these checks into one verification equation we consider $\{\rho_s(Y)\}$ the linear independent Lagrange interpolation polynomials over $\mathbb{V}_n$ and demonstrate that

$$\left(u^2(X)\rho_1(Y) + \sum_{s=2}^{n} u_{s-1}^2(X)\rho_s(Y) \right) - \left(\sum_{s=1}^{n-1} u_s(X)\rho_s(Y) + \mathsf{id}(X)\rho_n(Y) \right) = z_{V_m}(X) h_2(X, Y), \tag{8}$$

for some polynomial $h_2(X, Y)$.

In the remainder of this section the prover aims to demonstrate that (8) holds at a challenge point (α, β).

Proving (8): Strategy We prove (8) by showing that for some polynomial $h_1(Y)$, the polynomial

$$p(Y) = \underbrace{\big(u^2(\alpha)\rho_1(\beta) + \sum_{s=2}^{n} u_{s-1}^2(\alpha)\rho_s(\beta) + z_{V_n}(\beta)(-h_1(\beta) + h_1(Y)) \big)}_{\text{Denote } \xi_1} - \underbrace{\big(\sum_{s=1}^{n-1} u_s(\alpha)\rho_s(\beta) + \mathsf{id}(\alpha)\rho_n(\beta) \big)}_{\text{Denote } \xi_2} - \underbrace{z_{V_m}(\alpha) h_2(\alpha, Y)}_{\text{Denote } \xi_4}$$

evaluates to 0 at $Y = \beta$. For this the prover sends several values needed to reconstruct the commitment $[P]_1$ to $p(Y)$, and then provides a proof that $[P]_1$ opens to 0 at β.

Proving (8): Extra Notation First note that since the polynomials $\rho_s(Y)$ take 1 and 0 values only, we obtain that for all $Y \in \mathbb{V}_n$

$$u^2(X)\rho_1(Y) + \sum_{s=2}^{n} u_{s-1}^2(X)\rho_s(Y) = \big(u(X)\rho_1(Y) + \sum_{s=2}^{n} u_{s-1}(X)\rho_s(Y) \big)^2$$

We denote $\bar{U}(X,Y) = \sum_{s=2}^{n} u_{s-1}(X)\rho_s(Y)$ and $U(X,Y) = u(X)\rho_1(Y) + \bar{U}(X,Y)$ The prover begins by sending one commitment $[\bar{U}]_1$ to $\bar{U}(X,Y)$ and a second commitment $[h_2]$ to $h_2(X,Y)$. These are *bivariate* commitments. While there exist bivariate polynomial commitment schemes [26], these are incompatible with universal power-of-tau setups that are publicly available [23]. We thus instead view $\bar{U}(X,Y)$ and $h_2(X,Y)$ as the univariate polynomials $U(X^n,X)$ and $h_2(X^n,X)$. See Section 4.6.2 for more details.

The verifier responds with a random challenge $X = \alpha$.

Proving (8)**: Definition of and commitment to** h_1 The prover now wishes to find h_1 such that $p(Y)$ can be fully defined and its commitment can be computed by the verifier. They first provide a partial opening $[\bar{U}_\alpha]_1$ to $\bar{U}(\alpha,Y)$ and proves this is consistent with $[\bar{U}]_1$. They also open $[u(x)]_1$ at α to get $v_1 = u(\alpha)$. This allows the verifier to compute a commitment to the polynomial $U(\alpha,Y)$ as $U = [u(\alpha)]_1\rho_1(x) + [\bar{U}_\alpha]_1$.

The prover sends a commitment $[h_1]_1 = [h_1(x)]_1$ to $h_1(Y)$ such that

$$\sum_{s=1}^{n} u_{s-1}^2(\alpha)\rho_s(Y) = \left(U(\alpha,Y)\right)^2 + h_1(Y)z_{V_n}(Y). \tag{9}$$

The verifier responds with a second random challenge $Y = \beta$ and then (9) appears as

$$\sum_{s=1}^{n} u_{s-1}^2(\alpha)\rho_s(\beta) = \left(U(\alpha,\beta)\right)^2 + h_1(\beta)z_{V_n}(\beta) \tag{10}$$

Proving (8)**: Degree bound** The prover must show that $\bar{U}_1(X,1) = 0$ i.e. that there is no $\rho_1(Y)$ term. This convinces the verifier that the first term of $U(\alpha,Y)$ is indeed $u(\alpha)\rho_1(Y)$. When opening $[\bar{U}_\alpha]_1$ we enforce a degree bound of $n-1$. This is necessary because we are capturing bivariate polynomials with a univariate polynomial commitment scheme and we need to enforce that there are no X^n terms lingering in $\bar{U}(\alpha,X)$.

Proving (8)**: Sending** ξ_1 The prover communicates ξ_1 by opening $[\bar{U}_\alpha]_1$ to v_2 at $Y = \beta$ and verifier gets

$$\xi_1 = \{(10)\} = U(\alpha,\beta)^2 = (u(\alpha)\rho_1(\beta) + \bar{U}(\alpha,\beta))^2 = (v_1\rho_1(\beta) + v_2)^2$$

Proving (8)**: Sending** ξ_2 The prover communicates $\xi_2 = \sum_{s=1}^{n-1} u_s(\alpha)\rho_s(\beta)$. To do this we open $[\bar{U}(\alpha,Y)] = [\bar{U}_\alpha]_1$ to v_3 at $Y = \sigma\beta$ for σ the generator of $\mathbb{V}_n$. Indeed

$$\bar{U}(\alpha,\sigma\beta) = \sum_{s=2}^{n} u_{s-1}(\alpha)\rho_s(\sigma\beta) = \sum_{s=1}^{n-1} u_s(\alpha)\rho_s(\beta) \tag{11}$$

Proving (8)**: Finale** Finally the verifier can compute a commitment to $p(Y)$ as $[p(Y)]_1 = [(v_2 + v_1\rho_1(\beta))^2]_1 + z_{V_n}(\beta)[h_1]_1 - [v_3 + \mathsf{id}(\alpha)\rho_n(\beta)]_1 - z_{V_m}(\alpha)[h_2]_1$. Thus the prover finishes by demonstrating that $p(\beta) = 0$.

The protocol is shown in Figure 4.

Efficiency. In the protocol of Fig. 3, the work of the prover is dominated by the computation of $H(X)$ and $p_2(X)$ which have degree m^2, because $[H_1]$ is formed in time m by using the pre-computed individual proofs, and all the other proof elements are commitments to polynomials of degree m. In the protocol of Fig. 4, prover work is dominated by the computation of $[\bar{U}]_1$ and $[h_2]_1$ that are commitments to polynomials of degree $m\log(N)$.

Theorem 4. *The protocol in Figure 4 is a knowledge-sound argument for relation* $\mathsf{R}_{\mathsf{unity}}$ *under the algebraic group model and random oracle model if the qSDH, qDHE, and qSFrac assumptions hold.*

Proof. We proceed through a series of games to show that the protocol defined in Fig. 3 satisfies knowledge soundness. We set Game_0 to be the knowledge soundness game as defined in Definition A.1 and consider

Common input: $[u]_1$ where $[u]_1 = [u_0(X)]_1$

Prover: Take as input srs and $u(X)$

Samples blinders $t_1, \ldots t_n \leftarrow \mathbb{F}$.

For $s = 1, \ldots, n$, define $u_s(X) = \sum_{j=1}^{m} \left(\omega^{ij}\right)^{2^s} \mu_j(X) + t_s z_{V_m}(X)$,

Define $U(X,Y) = \sum_{s=1}^{n} u_{s-1}(X)\rho_s(Y)$.

Define $\bar{U}(X,Y) = U(X,Y) - u(X)\rho_1(Y)$

Define $h_2(X) = \sum_{s=1}^{n} \rho_s(Y) H_s(X)$ for $H_s(X) = (u_{s-1}^2(X) - u_s(X))/z_{V_m}(X)$

Output $\left([\bar{U}]_1 = [\bar{U}(x^n, x)]_1, [h_2]_1 = [h_2(x^n, x)]_1\right)$

Verifier: Send challenge $\alpha \in \mathbb{F}$

Prover: Define $h_1(Y) \leftarrow \left(U^2(\alpha, Y) - \sum_{s=1}^{n} u_{s-1}^2(\alpha)\rho_s(Y)\right)/z_{V_n}(Y)$

Output $[h_1]_1 = [h_1(x)]_1$

Verifier: Send challenge $\beta \in \mathbb{F}$

Prover:

$$
\begin{aligned}
p(Y) &\leftarrow (U^2(\alpha,\beta) - h_1(Y) z_{V_n}(\beta)) - \bar{U}(\alpha, \beta\sigma) + \mathsf{id}(\alpha)\rho_n(\beta)) - z_{V_m}(\alpha) h_2(\alpha, Y) \\
(v_1, \pi_1) &\leftarrow \mathsf{KZG.Open}\big(\mathsf{srs}, u(X), \deg = \perp, X = \alpha\big) \\
([\bar{U}(\alpha, x)]_1, \pi_2) &\leftarrow \mathsf{KZG.Open}\big(\mathsf{srs}, \bar{U}(X,Y), \deg = \perp, X = \alpha\big) \\
([h_2(\alpha, x)]_1, \pi_3) &\leftarrow \mathsf{KZG.Open}\big(\mathsf{srs}, h_2(X,Y), \deg = \perp, X = \alpha\big) \\
((0, v_2, v_3), \pi_4) &\leftarrow \mathsf{KZG.Open}\big(\mathsf{srs}, \bar{U}(\alpha, Y), \deg = n-1, Y = (1, \beta, \beta\sigma)\big) \\
(0, \pi_5) &\leftarrow \mathsf{KZG.Open}\big(\mathsf{srs}, p(Y), \deg = n-1, Y = \beta\big)
\end{aligned}
$$

Set $\left([\bar{U}_\alpha]_1 = [\bar{U}(\alpha, x)]_1, [h_{2,\alpha}]_1 = [h_2(\alpha, x)]_1\right.$ and output $\left.\left([\bar{U}_\alpha]_1, [h_{2,\alpha}]_1, v_1, v_2, v_3, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5\right)\right.$

Verifier: Compute $U \leftarrow v_1\rho_1(\beta) + v_2$, $\quad [P]_1 \leftarrow U^2 - [h_1]_1 z_{V_n}(\beta) - (v_3 + \mathsf{id}(\alpha)\rho_n(\beta)) - z_{V_m}(\alpha)[h_{2,\alpha}]_1$

Accept if and only if

$$
\begin{aligned}
1 &= \mathsf{KZG.Verify}\big(\mathsf{srs}_{\mathsf{KZG}}, [u]_1, \deg = \perp, X = \alpha, v_1, \pi_1\big) \\
1 &= \mathsf{KZG.Verify}\big(\mathsf{srs}_{\mathsf{KZG}}, [\bar{U}]_1, \deg = \perp, X = \alpha, [\bar{U}_\alpha]_1, \pi_2\big) \\
1 &= \mathsf{KZG.Verify}\big(\mathsf{srs}_{\mathsf{KZG}}, [h_2]_1, \deg = \perp, X = \alpha, [h_{2,\alpha}]_2, \pi_3\big) \\
1 &= \mathsf{KZG.Verify}\big(\mathsf{srs}_{\mathsf{KZG}}, [\bar{U}_\alpha]_1, \deg = n-1, Y = (1, \beta, \beta\sigma), (0, v_2, v_3), \pi_4\big) \\
1 &= \mathsf{KZG.Verify}\big(\mathsf{srs}_{\mathsf{KZG}}, [P]_1, \deg = n-1, Y = \beta, 0, \pi_5\big)
\end{aligned}
$$

Figure 4: Argument for proving that some polynomial $u(X)$ has Nth roots of unity as coefficients in the basis $\{\mu_j(X)\}_{j=1}^{m}$.

an algebraic adversary $\mathcal{A}$ against it which has advantage $\mathsf{Adv}^{\mathsf{knowledge\text{-}sound}}_{\mathcal{A}}(\lambda)$. We define Game_1 and Game_2 and specify reductions $\mathcal{B}_1$ and $\mathcal{B}_2$ such that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{k\text{-}sound}}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}}(\lambda) &\leq \mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{qSDH}}_{\mathcal{B}_1}(\lambda)\\ &\leq \mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{qSDH}}_{\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{qDHE}}_{\mathcal{B}_1}(\lambda)\\ &\leq \mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{qSDH}}_{\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{qDHE}}_{\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{qSDH}}_{\mathcal{B}_3}(\lambda)\\ &\leq \mathsf{Adv}^{\mathsf{qSDH}}_{\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{qDHE}}_{\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{qSFrac}}_{\mathcal{B}_3}(\lambda) + \mathsf{negl}(\lambda)\end{aligned}$$

In Game_0 the adversary will return $[u]_1 = [u(x)]$ along with a proof. We define Game_1 identically to Game_0, but after the adversary returns $[u]_1$ and a proof, Game_1 additionally checks whether for $u(X), \bar{U}_\alpha(X), p(X)$ the algebraic representations of $[u]_1, [\bar{U}_\alpha]_1, [P]_1$, it is true that $u(\alpha) = v_1$, $\bar{U}_\alpha(1) = 0$, $\bar{U}_\alpha(\beta) = v_2$, $\bar{U}_\alpha(\beta\sigma) = v_3$, and $p(\beta) = 0$; and it aborts if one of the conditions does not hold.

The redution $\mathcal{B}_1$ takes as input the challenge $[y_1]_1, \ldots, [y_q]_1$. It runs the following reduction $\mathcal{B}_{\mathsf{KZG}}$ as a subroutine. The $\mathcal{B}_{\mathsf{KZG}}$ runs the adversary $\mathcal{A}$ against Game_0 over an srs in which $[x]_1 = [y_1]_1$. Whenever $\mathcal{A}$ returns an output which wins the Game_0 game, if $(f(X), \mathbf{v}, \mathbf{z})$ for some $(f(X), \mathbf{v}, \mathbf{z}) \in \{(u(X), v_1, \alpha), (\bar{U}_\alpha(X), (1, \beta, \sigma\beta), (0, v_2, v_3)), (p(X), \beta, 0)\}$ is such that $f(v_i) \neq z_i$, then $\mathcal{B}_{KZG}$ computes $f(z) = v'$ and a valid proof π'. It outputs $([f(x)]_1, \mathbf{z}, \mathbf{v}, \pi)$ and $([f(x)]_1, \mathbf{z}, \mathbf{v}', \pi')$ and wins evaluation binding as they are both proofs that verify and open to different elements. Then $\mathcal{B}_{\mathsf{qSDH}}$ can extract a qSDH solution from these openings following the proof in Theorem 3 of [22]. Thus

$$\mathsf{Adv}^{\mathsf{k\text{-}sound}}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{qSDH}}_{\mathcal{B}_1}(\lambda)$$

Now Game_2 behaves identically as Game_1 but it additionally checks that $\deg(\bar{U}_\alpha) \leq n - 1$ and $\deg(h_2) \leq n - 1$. If it is not the case, it aborts. Suppose $\mathcal{A}$ returns either $\deg(\bar{U}_\alpha) = n - 1 + d$ or $\deg(h_2) = n - 1 + d$ for some $d > 0$. We argue the advantage of $\mathcal{A}$ in Game_1 and Game_2 is the same unless we can build an adversary $\mathcal{B}_2$ that succeeds against qDHE. The $\mathcal{B}_2$ takes as input the challenge $[y_1]_1, \ldots, [y_{q+d-1}]_1$ and runs the adversary $\mathcal{A}$ against Game_1 over an srs in which $[x]_1 = [y_1]_1$. Whenever $\mathcal{A}$ returns an output which wins the Game_1 game, if $(f(X), \mathbf{v}, \mathbf{z})$ for

$$(f(X), \mathbf{v}, \mathbf{z}) \in \{(\bar{U}_\alpha(X), (1, \beta, \sigma\beta), (0, v_2, v_3)), (p(X), \beta, 0)\}$$

is such that $f(X)$ has degree greater than $n - 1$, then the corresponding proof $\pi = [q(x)]_1$ has a representation $q(X)$ has degree $q + 1$. Thus $\mathcal{B}_2$ succeeds in returning $[\pi - \sum_{i=0}^{q-1} x^i]_1$ and

$$\mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{qDHE}}_{\mathcal{B}_1}(\lambda)$$

We define Game_3 identically to Game_2, but after the adversary returns $[u]_1$ and a proof, Game_3 additionally checks whether for $\bar{U}(X), h_2(X), \bar{U}_\alpha(X), h_{2,\alpha}(X)$ the algebraic representations of $[\bar{U}]_1, [\bar{U}_\alpha]_1, [h_2]_1, [h_{2,\alpha}]_1$, it is true that

$$\bar{U}_\alpha(X) = \sum_{i,j} \alpha^i \bar{U}_{ni} X^j \text{ and } h_{2,\alpha}(X) = \sum_{i,j} \alpha^i h_{2,ni} X^j$$

and it aborts if one of the conditions does not hold.

The redution $\mathcal{B}_3$ against qSFrac [18] takes as input the challenge $[y_1]_1, \ldots, [y_q]_1$ and runs the adversary $\mathcal{A}$ against Game_2 over an srs in which $[x]_1 = [y_1]_1$. Whenever $\mathcal{A}$ returns an output which wins the Game_2 game, if $(f(X), V, z)$ for

$$(f(X), \phi(X), z) \in \{(\bar{U}(X), \bar{U}_\alpha(X), \alpha), (h_2(X), h_{2,\alpha}(X), \alpha)\}$$

is such that $\phi(X) \neq \phi'(X) = \sum_{i,j} \alpha^i f_{ni} X^j$, then set π be the proof for $(f(X), \phi(X), z)$. Then $\mathcal{B}_3$ returns

$$\phi(X) - \phi'(X), (X^n - z), \pi - \left[\frac{f(x) - \phi'(x)}{x^n - z}\right]_1$$

We have that $\deg(\phi(X) - \phi'(X)) < \deg(X^n - z)$ because $\phi(X)$ has degree bounded by $n - 1$. Hence this is as a valid solution and

$$\mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}(\lambda) + \mathsf{Adv}^{\mathsf{qSFrac}}_{\mathcal{B}_3}(\lambda)$$

Lets see that the advantage of $\mathcal{A}$ in Game_3 is negligible.

Consider $h_1(X), h_2(X,Y)$ the algebraic representations of $[h_1]_1$, $[h_2]_1$. We can use the equations verified by Game_1 and replace the corresponding values in $p(X)$, obtaining

$$\begin{aligned} p(X) &= (v_1\rho_1(\beta) + v_2)^2 - h_1(X)z_{V_n}(\beta) - (v_3 + \mathsf{id}(\alpha)\rho_n(\beta)) - z_{V_m}(\alpha)h_{2,\alpha}(X) \\ &= \big(u(\alpha)\rho_1(\beta) + \bar{U}_\alpha(\beta)\big)^2 - h_1(X)z_{V_n}(\beta) - (\bar{U}_\alpha(\beta\sigma) + \mathsf{id}(\alpha)\rho_n(\beta)) - z_{V_m}(\alpha)h_{2,\alpha}(X) \\ &= \big(u(\alpha)\rho_1(\beta) + \bar{U}(\alpha,\beta)\big)^2 - h_1(X)z_{V_n}(\beta) - (\bar{U}(\alpha,\beta\sigma) + \mathsf{id}(\alpha)\rho_n(\beta)) - z_{V_m}(\alpha)h_2(\alpha,X) \end{aligned}$$

From the fact that $p(\beta) = 0$ we get that

$$0 = \big(u(\alpha)\rho_1(\beta) + \bar{U}(\alpha,\beta)\big)^2 - (\bar{U}(\alpha,\beta\sigma) + \mathsf{id}(\alpha)\rho_n(\beta)) - z_{V_m}(\alpha)h_2(\alpha,\beta) - h_1(\beta)z_{V_n}(\beta)$$

Since $[u]_1, [\bar{U}]_1$, $[h_1]_1$, $[h_2]_1$ have been sent by the prover before it sees challenges β, we have that except in the case where $(Y = \beta)$ is a root of the polynomial below, which happens with negligible probability, for all Y,

$$0 = \big(u(\alpha)\rho_1(Y) + \bar{U}(\alpha,Y)\big)^2 - (\bar{U}(\alpha,Y\sigma) + \mathsf{id}(\alpha)\rho_n(Y)) - z_{V_m}(\alpha)h_2(\alpha,Y) - h_1(Y)z_{V_n}(Y) \tag{12}$$

Thus we have that

$$\begin{aligned} i = 0 &\Rightarrow 0 = u^2(\alpha) - \bar{U}(\alpha,\sigma^1) - z_{V_m}(\alpha)h_2(\alpha,\sigma^1) \\ 1 \le i \le n-1 &\Rightarrow 0 = \bar{U}^2(\alpha,\sigma^i) - \bar{U}(\alpha,\sigma^{i+1}) - z_{V_m}(\alpha)h_2(\alpha,\sigma^i) \\ i = n &\Rightarrow 0 = \bar{U}^2(\alpha,\sigma^{n-1}) - \mathsf{id}(\alpha) - z_{V_m}(\alpha)h_2(\alpha,\sigma^1) \end{aligned}$$

Since $[u]_1, [\bar{U}]_1$, $[h_2]_1$ have been sent by the prover before it sees challenges α, we have that except in the case where $(X = \alpha)$ is a root of the polynomial below, which happens with negligible probability, for all X,

$$\begin{aligned} i = 0 &\Rightarrow 0 = u^2(X) - \bar{U}(X,\sigma^1) - z_{V_m}(X)h_2(X,\sigma^1) \\ 1 \le i \le n-1 &\Rightarrow 0 = \bar{U}^2(X,\sigma^i) - \bar{U}(X,\sigma^{i+1}) - z_{V_m}(X)h_2(X,\sigma^i) \\ i = n &\Rightarrow 0 = \bar{U}^2(X,\sigma^{n-1}) - \mathsf{id}(X) - z_{V_m}(X)h_2(X,\sigma^1) \end{aligned}$$

Over $\nu \in V_m$ we thus have that

$$\begin{aligned} i = 0 &\Rightarrow 0 = u^2(\nu) - \bar{U}(\nu,\sigma^1) \\ 1 \le i \le n-1 &\Rightarrow 0 = \bar{U}^2(\nu,\sigma^i) - \bar{U}(\nu,\sigma^{i+1}) \\ i = n &\Rightarrow 0 = \bar{U}^2(\nu,\sigma^{n-1}) - 1 \end{aligned}$$

Together these gives us the desired requirement that $u^N(\nu) = 1$ for all $\nu \in V_m$ except with negligible probability. □

Theorem 5. *The protocol in Fig. 3 and 4 implies position-hiding linkability between the vector commitment schemes of* C *and* cm*, provided that the zk proof for* $\mathsf{R}_{\mathsf{unity}}$ *is instantiated with a the protocol in Fig. 4 and provided that* $\log(N) > 6$.

The proof is in Appendix F

8 Optimizations

In this section we describe some optimizations we apply to the protocols in Fig. 3 and 4 in order to achieve the efficiency claimed in Table 1.

Opening t polynomials in one point. As noted in [16],[13], whenever we have t openings of different polynomials at the same point i.e. for $t = 2$ this would be of the form

$$\begin{aligned} \pi_1 &\leftarrow \mathsf{KZG.Open}(\mathsf{srs}_{\mathsf{KZG}}, f_1(X), \deg = d, \alpha) \\ \pi_2 &\leftarrow \mathsf{KZG.Open}(\mathsf{srs}_{\mathsf{KZG}}, f_2(X), \deg = d, \alpha) \end{aligned}$$

then we can send a single opening proof π as opposed to t opening proofs $\pi_1, \ldots, \pi_t$.

Batching Pairings. We also apply standard techniques to batch pairings that share the same elements in one of the two groups. Namely, we can aggregate the equations

$$e([a]_1, [b_1]_2) = e([c_1]_1, [d]_2) \text{ and } e([a]_1, [b_2]_2) = e([c_2]_1, [d]_2),$$
$$\text{as } e([a]_1, [b_1 + \gamma b_2]_2) = e([c_1 + \gamma c_2]_1, [d]_2)$$

for γ some random field element sampled by the verifier.

Note that we can adapt KZG openings equations so they can be batched further, namely if we parse the verification pairing as $e\big([F_1]_1 - s_1 + [Q_1]_1\alpha, [1]_2\big) = e\big([Q_1]_1, [x]_2\big)$, then two openings of different polynomials at different points can be verified by two pairings.

Fig. 1 and 2: In Fig. 1 proofs have the form $([z]_2, [T]_1, [S]_2, \pi_{\mathsf{ped}}, \pi_{\mathsf{unity}})$. See that π_{ped} consists of 1 $\mathbb{G}_1$ and 2$\mathbb{F}$. In Fig. 2 proofs have the form $([F]_1, [H]_1, v_1, v_2, \pi_1, \pi_2)$ which amounts to 4$\mathbb{G}_1$ and 2$\mathbb{F}$. Thus we have a total of 6$\mathbb{G}_1$, 2$\mathbb{G}_2$ and 4$\mathbb{F}$.

For the verifier, their first pairing check in Fig. 1 uses pairings of the form $e(*, [1]_2)$, $e(*, [z]_2)$, and $e([h]_1, *)$ amounting to 3 pairings. The Pedersen verifier uses no pairings. In Fig. 2 we have a KZG verifier which uses pairings of the form $e(*, [1]_2)$, $e(*, [x]_2)$, and a pairing check that uses pairings of the form $e(*, [1]_2)$, $e(*, [z]_2)$, and $e(*, [x]_2)$. Thus we can batch the pairing checks to get a total of 4 unique pairings over the two constructions.

Fig. 3 and 4: In Fig. 3 proofs have the form $([C_I]_1, [z_I]_1, [u]_1, [H_1]_2, [H_2]_1, v_1, v_2, \pi_1, \pi_2, \pi_3, \pi_{\mathsf{unity}})$. Here the π_1, π_3 are both openings at the same α and can be batched into one proof. Thus there are 7$\mathbb{G}_1$, 1$\mathbb{G}_2$ and 2$\mathbb{F}$ in addition to the π_{unity}. In 2 proofs have that form $\big([\bar{U}]_1, [h_2]_1, [h_1]_1, [\bar{U}_\alpha]_1, [h_{2,\alpha}]_1, v_1', v_2', v_3', \pi_1', \pi_2', \pi_3', \pi_4', \pi_5'\big)$. Here we can send the same verifier challenge α in both Fig. 3 and Fig. 4 (assuming we run the protocols in parallel) which allows us to avoid sending v_1', π_1' in Fig. 4. Further, this allows us to batch the proofs (π_2', π_3') with the proof for (π_1, π_3) because these all use the same α. Thus π_{unity} contributes 7$\mathbb{G}_1$, and 2$\mathbb{F}$ Thus we have a total of 14$\mathbb{G}_1$, 1$\mathbb{G}_2$ and 4$\mathbb{F}$.

For the verifier, their pairing check in Fig. 3 uses pairings of the form $e(*, [1]_2)$ and $e([z_I]_1)$. We also have 3 KZG verifiers which use pairings of the form $e(*, [1]_2)$, $e(*, [x]_2)$. This amounts to 2 batched pairings. In Fig. 2 we have a 5 KZG verifiers. Two use a degree check and thus use pairings of the form $e(*, [1]_2)$, $e(*, [x]_2)$, and $e(*, [x^{d-n+1}]_2)$. The others have the usual pairings as these do not have degree checks. Thus we can batch the pairing checks to get a total of 4 unique pairings over the two constructions.

9 Implementation

We have implemented our scheme in Rust using the arkworks library [2], and have released the implementation in open source[2]. The code contains a subroutine that computes all KZG openings, which we need for fast proof preprocessing and which can be used in other projects.

In Fig. 5 and Fig. 6 we compare Caulk with its alternatives in the same scenarios. The single opening ($m = 1$) scenario works with a vector of size N and considers the following schemes:

- MT-Pos: SNARKed Merkle Poseidon tree with N elements. We used a legosnark implementation[3].
- Caulk: the $m = 1$ version;
- MT-SHA: SNARKed Merkle SHA-2 tree with N elements.
- RSA acc: RSA-2048 accumulator of N elements. We used a legosnark implementation.

The prover time is given in Fig. 5. We see that Caulk is almost 100 times as fast as the Poseidon tree, and 10 times as fast as the RSA accumulator.

The m-opening scenario works with a lookup of size m in a vector of size N and considers the following schemes:

[2] https://github.com/caulk-crypto/caulk
[3] https://github.com/matteocam/libsnark-lego/

- MT-Pos-20: SNARKed Merkle Poseidon tree with $N = 2^{20}$ elements.
- MT-Pos-8: SNARKed Merkle Poseidon tree with $N = 2^8$ elements.
- Caulk-8: Caulk with $N = 2^8$.
- Caulk-20: Caulk with $N = 2^{20}$.
- RSA acc: RSA-2048 accumulator of $N = 2^{16}$ elements (the timings are the same for other N).

We see that Caulk is faster than the RSA accumulator for smaller N but approaches it for big N. Both are significantly faster than Merkle-SNARK.

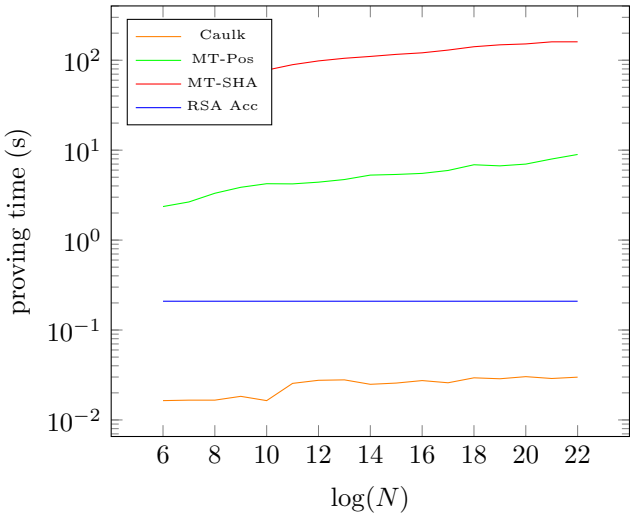

Figure 5: Comparison for single openings

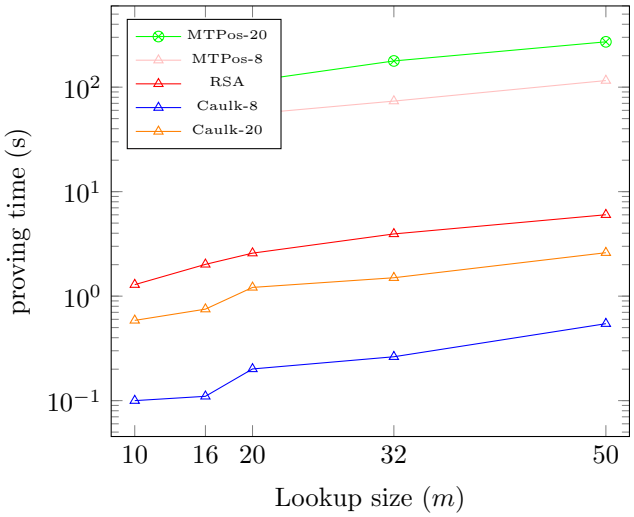

Figure 6: Comparison for lookup tables

References

[1] M. R. Albrecht, L. Grassi, C. Rechberger, A. Roy, and T. Tiessen. MiMC: Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity. In *ASIACRYPT 2016*, volume 10031 of *LNCS*, pages 191–219, 2016.

[2] arkworks contributors. `arkworks` zksnark ecosystem, 2022.

[3] S. Bayer and J. Groth. Zero-knowledge argument for polynomial evaluation with application to blacklists. In T. Johansson and P. Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 646–663. Springer, 2013.

[4] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006.

[5] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy, SP 2014, Berkeley, CA, USA, May 18-21, 2014*, pages 459–474. IEEE Computer Society, 2014.

[6] D. Benarroch, M. Campanelli, D. Fiore, K. Gurkan, and D. Kolonelos. Zero-knowledge proofs for set membership: Efficient, succinct, modular. In N. Borisov and C. Díaz, editors, *Financial Cryptography and Data Security - 25th International Conference, FC 2021, Virtual Event, March 1-5, 2021, Revised Selected Papers, Part I*, volume 12674 of *Lecture Notes in Computer Science*, pages 393–414. Springer, 2021.

[7] D. Boneh and X. Boyen. Short signatures without random oracles. In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer, 2004.

[8] J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, J. Groth, and C. Petit. Short accountable ring signatures based on DDH. In G. Pernul, P. Y. A. Ryan, and E. R. Weippl, editors, *Computer Security - ESORICS 2015 - 20th European Symposium on Research in Computer Security, Vienna, Austria, September 21-25, 2015, Proceedings, Part I*, volume 9326 of *Lecture Notes in Computer Science*, pages 243–265. Springer, 2015.

[9] J. Bootle and J. Groth. Efficient batch zero-knowledge arguments for low degree polynomials. In M. Abdalla and R. Dahab, editors, *Public-Key Cryptography - PKC 2018 - 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part II*, volume 10770 of *Lecture Notes in Computer Science*, pages 561–588. Springer, 2018.

[10] J. Camenisch, R. Chaabouni, and A. Shelat. Efficient protocols for set membership and range proofs. In J. Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings*, volume 5350 of *Lecture Notes in Computer Science*, pages 234–252. Springer, 2008.

[11] J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In M. Yung, editor, *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76. Springer, 2002.

[12] M. Campanelli, D. Fiore, S. Han, J. Kim, D. Kolonelos, and H. Oh. Succinct zero-knowledge batch proofs for set accumulators. *IACR Cryptol. ePrint Arch.*, page 1672, 2021.

[13] A. Chiesa, Y. Hu, M. Maller, P. Mishra, P. Vesely, and N. Ward. Marlin: Preprocessing zksnarks with universal and updatable srs. In A. Canteaut and Y. Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of*

Cryptographic Techniques, Virtual Conference, May 1-15, 2020, Proceedings, Part I, volume 12105 of *Lecture Notes in Computer Science*, pages 738–768. Springer, 2020.

[14] D. Feist and D. Khovratovich. Fast amortized kate proofs.

[15] G. Fuchsbauer, E. Kiltz, and J. Loss. The algebraic group model and its applications. In H. Shacham and A. Boldyreva, editors, *CRYPTO 2018, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, 2018.

[16] A. Gabizon and Z. J. Williamson. Plonk: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. *IACR Cryptol. ePrint Arch.*, page 953, 2019.

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

[18] E. Ghadafi and J. Groth. Towards a classification of non-interactive computational assumptions in cyclic groups. *IACR Cryptol. ePrint Arch.*, page 343, 2017.

[19] L. Grassi, D. Khovratovich, A. Roy, C. Rechberger, and M. Schofnegger. Poseidon: A new hash function for zero-knowledge proof systems. *Usenix Security 2021*, 2021.

[20] J. Groth. On the size of pairing-based non-interactive arguments. In *EUROCRYPT (2)*, volume 9666 of *Lecture Notes in Computer Science*, pages 305–326. Springer, 2016.

[21] J. Groth and M. Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 253–280. Springer, 2015.

[22] A. Kate, G. M. Zaverucha, and I. Goldberg. Constant-size commitments to polynomials and their applications. In *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 177–194. Springer, 2010.

[23] M. Kohlweiss, M. Maller, J. Siim, and M. Volkhov. Snarky ceremonies. In *ASIACRYPT (3)*, volume 13092 of *Lecture Notes in Computer Science*, pages 98–127. Springer, 2021.

[24] U. M. Maurer. Unifying zero-knowledge proofs of knowledge. In *AFRICACRYPT*, volume 5580 of *Lecture Notes in Computer Science*, pages 272–286. Springer, 2009.

[25] I. Miers, C. Garman, M. Green, and A. D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013*, pages 397–411. IEEE Computer Society, 2013.

[26] C. Papamanthou, E. Shi, and R. Tamassia. Signatures of correct computation. In *TCC*, volume 7785 of *Lecture Notes in Computer Science*, pages 222–242. Springer, 2013.

[27] L. Pearson, J. Fitzgerald, H. Masip, M. Bellés-Muñoz, and J. L. Muñoz-Tapia. Plonkup: Reconciling plonk with plookup. *IACR Cryptol. ePrint Arch.*, page 86, 2022.

[28] A. Tomescu, I. Abraham, V. Buterin, J. Drake, D. Feist, and D. Khovratovich. Aggregatable subvector commitments for stateless cryptocurrencies. In C. Galdi and V. Kolesnikov, editors, *Security and Cryptography for Networks - 12th International Conference, SCN 2020, Amalfi, Italy, September 14-16, 2020, Proceedings*, volume 12238 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2020.

[29] Tornado cash privacy solution version 1.4, 2021. `https://tornado.cash/Tornado.cash_whitepaper_v1.4.pdf`.

[30] ZCash protocol specification, 2022, 1st February. `https://github.com/zcash/zips/blob/master/protocol/protocol.pdf`.

[31] Zksync rollup protocol, 2021. `https://github.com/matter-labs/zksync/blob/master/docs/protocol.md`.

A Definitions

A.1 Non-Interactive Zero-Knowledge Argument of Knowledge

Let $\mathcal{R}$ be a family of universal relations. Given a relation $\mathsf{R} \in \mathcal{R}$ and an instance x we call w a *witness* for x if $(\mathsf{x}, w) \in \mathsf{R}$, $\mathcal{L}(\mathsf{R}) = \{\mathsf{x} |\ \exists w : (\mathsf{x}, w) \in \mathsf{R}\}$ is the language of all the x that have a witness w in the relation R, while $\mathcal{L}(\mathsf{R})$ is the language of all the pairs (x, R) such that $\mathsf{x} \in \mathcal{L}(\mathsf{R})$. We will assume R it is implicit as prover and verifier input.

Definition A.1. *A Non-Interactive Zero-Knowledge Argument of Knowledge is a tuple of PPT algorithms* $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{Simulate})$ *such that:*

- $(\mathsf{srs}, x) \leftarrow \mathsf{Setup}(\mathcal{R})$*: On input a family of relations* $\mathcal{R}$*,* Setup *outputs a structured reference string* srs *and a trapdoor* x*;*
- $\pi \leftarrow \mathsf{Prove}(\mathsf{srs}, (\mathsf{x}, w))$*: On input a pair* $(\mathsf{x}, w) \in \mathsf{R}$*, it outputs a proof* π *of the fact that* $\mathsf{x} \in \mathcal{L}(\mathsf{R})$*;*
- $1/0 \leftarrow \mathsf{Verify}(\mathsf{srs}, \mathsf{x}, \pi)$*: On input the srs, the instance* x *and the proof, it produces a bit expressing acceptance* (1)*, or rejection* (0)*;*
- $\pi_{\text{sim}} \leftarrow \mathsf{Simulate}(\mathsf{srs}, x, \mathsf{x})$*: The simulator has the srs, the trapdoor* x *and the instance* x *as inputs and it generates a simulated proof* π_{sim}*,*

and that satisfies completeness, knowledge soundness and zero-knowledge as defined below.

Completeness: holds if an honest prover will always convince an honest verifier. Formally, $\forall\ \mathsf{R} \in \mathcal{R}, (\mathsf{x}, w) \in \mathsf{R}$,

$$\Pr\left[\ \mathsf{Verify}(\mathsf{srs}, \mathsf{x}, \pi) = 1 \ \middle|\ \begin{array}{l}(\mathsf{srs}, x) \leftarrow \mathsf{Setup}(\mathcal{R}) \\ \pi \leftarrow \mathsf{Prove}(\mathsf{srs}, (\mathsf{x}, w))\end{array}\ \right] = 1.$$

Knowledge-Soundness: captures the fact that a cheating prover cannot, except with negligible probability, create a proof π accepted by the verification algorithm unless it has a witness w such that $(\mathsf{x}, w) \in \mathsf{R}$. Formally, for all PPT adversaries $\mathcal{A}$, there exists a PPT extractor $\mathcal{E}$ such that the following probability is negligible in λ

$$\Pr\left[\ (\mathsf{x}, w) \notin \mathsf{R} \wedge \mathsf{Verify}(\mathsf{srs}, \mathsf{x}, \pi) = 1 \ \middle|\ \begin{array}{l}(\mathsf{srs}, x) \leftarrow \mathsf{Setup}(\mathcal{R}) \\ (\mathsf{x}, \pi) \leftarrow \mathcal{A}(\mathsf{srs}) \\ w \leftarrow \mathcal{E}(\mathsf{srs}, \mathsf{x}, \pi)\end{array}\ \right]$$

Zero-Knowledge: $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{Simulate})$ is zero-knowledge if for all $\mathsf{R} \in \mathcal{R}$, instances x and PPT adversaries $\mathcal{A}$,

$$\Pr\left[\ \mathcal{A}(\mathsf{srs}, \pi) = 1 \ \middle|\ \begin{array}{l}(\mathsf{srs}, x) \leftarrow \mathsf{Setup}(\mathcal{R}) \\ \mathsf{x} \leftarrow \mathcal{A}(\mathsf{srs}) \\ \pi \leftarrow \mathsf{Prove}(\mathsf{srs}, (\mathsf{x}, w))\end{array}\ \right] \approx \Pr\left[\ \mathcal{A}(\mathsf{srs}, \pi_{\text{sim}}) = 1 \ \middle|\ \begin{array}{l}(\mathsf{srs}, x) \leftarrow \mathsf{Setup}(\mathcal{R}) \\ \mathsf{x} \leftarrow \mathcal{A}(\mathsf{srs}) \\ \pi_{\text{sim}} \leftarrow \mathsf{Simulate}(\mathsf{srs}, x, \mathsf{x})\end{array}\ \right].$$

Definition A.2 (Vector Commitment Scheme)**.** *A* Vector Commitment Scheme *is a tuple of algorithms* $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Verify})$ *such that:*

- $(x, \mathsf{srs}) \leftarrow \mathsf{Setup}(\mathsf{par}, d)$*: On input the system parameters and a bound* d *on the size of the vectors, it outputs a structured reference string and trapdoor* x*.*
- $\mathsf{C} \leftarrow \mathsf{Commit}(\mathsf{srs}, \vec{v}, r)$*: On input the srs, a vector* $\vec{v}$*, and randomness* r *it outputs a commitment* C*.*
- $(v_i, \pi) \leftarrow \mathsf{Open}(\mathsf{srs}, \vec{v}, r, i)$*: On input the srs, the vector, its size , the commitment randomness, and a position* $i \in [m]$ *it outputs* $v_i \in \mathbb{F}$ *and proof* π *that* v_i *is the* i*th element of vector* $\vec{v}$*.*
- $1/0 \leftarrow \mathsf{Verify}(\mathsf{srs}, \mathsf{C}, i, v_i, \pi)$*: On input the srs, the commitment, position, claimed value* v_i*, and the proof, it outputs a bit indicating acceptance or rejection.*

A vector commitment scheme should satisfy the following properties:

Correctness: *It captures the fact that an honest prover will always convince an honest verifier. Namely, for all vectors* $\vec{v} \in \mathbb{F}^N$ *and* $i \in [N]$

$$\mathsf{Pr}\left[\begin{array}{c|r} \mathsf{Verify}\big(\mathsf{srs}, \mathsf{C}, i, v_i, \pi\big) = 1 & \mathsf{srs} \leftarrow \mathsf{Setup}\big(\mathsf{par}, N\big) \\ & \mathsf{C} \leftarrow \mathsf{Commit}\big(\mathsf{srs}, \vec{v}, r\big) \\ & (v_i, \pi) \leftarrow \mathsf{Open}\big(\mathsf{srs}, \vec{v}, r, i\big) \end{array}\right] = 1$$

(Weak) Position Binding: *Captures the fact that no PPT adversary* $\mathcal{A}$ *should be able to present for one commitment two valid openings for the same position. Formally:*

$$\mathsf{Pr}\left[\begin{array}{c|r} \begin{array}{c}\mathsf{Verify}\big(\mathsf{srs}, \mathsf{C}, i, y, \pi\big) = 1, \\ \mathsf{Verify}\big(\mathsf{srs}, \mathsf{C}, i, y', \pi'\big) = 1 \\ \text{and } y \neq y'\end{array} & \begin{array}{r}\mathsf{srs} \leftarrow \mathsf{Setup}\big(\mathsf{par}, N\big) \\ (\vec{v}, r, i, y, y', \pi, \pi') \leftarrow \mathcal{A}\big(\mathsf{srs}\big) \\ \mathsf{C} \leftarrow \mathsf{Commit}\big(\mathsf{srs}, \vec{v}, r\big)\end{array} \end{array}\right] \approx 0$$

(Strong) Position Binding: *Captures the fact that no PPT adversary* $\mathcal{A}$ *should be able to present for one commitment two valid openings for the same position. Formally:*

$$\mathsf{Pr}\left[\begin{array}{c|r} \begin{array}{c}\mathsf{Verify}\big(\mathsf{srs}, \mathsf{C}, i, y, \pi\big) = 1, \\ \mathsf{Verify}\big(\mathsf{srs}, \mathsf{C}, i, y', \pi'\big) = 1 \\ \text{and } y \neq y'\end{array} & \begin{array}{r}\mathsf{srs} \leftarrow \mathsf{Setup}\big(\mathsf{par}, N\big) \\ (\mathsf{C}, i, y, y', \pi, \pi') \leftarrow \mathcal{A}\big(\mathsf{srs}\big)\end{array} \end{array}\right] \approx 0$$

Knowledge Soundness: *Captures the fact that whenever the prover provides a valid opening, it knows a valid pair* $(p(X), p(\alpha)) \in \mathbb{F}[X] \times \mathbb{F}$*, where* $deg(p) \leq \deg$*. Formally, for all PPT adversaries* $\mathcal{A}$ *there exists an efficient extractor* $\mathcal{E}$ *such that:*

$$\mathsf{Pr}\left[\begin{array}{c|r} \begin{array}{c}\mathsf{Verify}\big(\mathsf{srs}, \mathsf{C}, i, y, \pi\big) = 1 \\ \wedge\ v_i \neq y\end{array} & \begin{array}{r}\mathsf{srs} \leftarrow \mathsf{Setup}\big(\mathsf{par}, N\big) \\ \mathsf{C} \leftarrow \mathcal{A}\big(\mathsf{srs}\big) \\ \vec{v} \leftarrow \mathcal{E}\big(\mathsf{srs}, \mathsf{C}, N\big) \\ (i, y, \pi) \leftarrow \mathcal{A}\big(\mathsf{srs}, \vec{v}, N, i\big)\end{array} \end{array}\right] \approx 0$$

Definition A.3 (Polynomial Commitment Scheme)**.** *A* Polynomial Commitment Scheme *is a tuple of algorithms* $\big(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Verify}\big)$ *such that:*

- $(x, \mathsf{srs}) \leftarrow \mathsf{Setup}\big(\mathsf{par}, d\big)$*: On input the system parameters and a degree bound* d*, it outputs a structured reference string and trapdoor* x*.*
- $\mathsf{C} \leftarrow \mathsf{Commit}\big(\mathsf{srs}, p(X), r\big)$*: On input the srs and a polynomial* $p(X)$*, and randomness* r *it outputs a commitment* C *to* $p(X)$*.*
- $(s, \pi) \leftarrow \mathsf{Open}\big(\mathsf{srs}, p(X), r, \alpha\big)$*: On input the srs, the polynomial, commitment randomness* r*, a query point* $\alpha \in \mathbb{F}$*, it outputs* $s \in \mathbb{F}$ *and an evaluation proof* π *that* $s = p(\alpha)$*.*
- $1/0 \leftarrow \mathsf{Verify}\big(\mathsf{srs}, \mathsf{C}, \deg, \alpha, s, \pi\big)$*: On input the srs, the commitment, degree bound, query and evaluation points* α, s*, and the proof of correct evaluation, it outputs a bit indicating acceptance or rejection.*

A polynomial commitment scheme should satisfy the following properties:

Completeness: *It captures the fact that an honest prover will always convince an honest verifier. Formally, for any polynomial* $p(X)$ *such that* $deg(p) \leq d$ *and query point* $\alpha \in \mathbb{F}$ *the following probability is 1:*

$$\mathsf{Pr}\left[\begin{array}{c|r} \mathsf{Verify}\big(\mathsf{srs}, \mathsf{C}, \deg, \alpha, s, \pi\big) = 1 & \begin{array}{r}\mathsf{srs} \leftarrow \mathsf{Setup}\big(\mathsf{par}, d\big) \\ \mathsf{C} \leftarrow \mathsf{Commit}\big(\mathsf{srs}, p(X), r\big) \\ s = p(\alpha), \deg(p) = \deg \\ (s, \pi) \leftarrow \mathsf{Open}\big(\mathsf{srs}, p(X), r, \alpha\big)\end{array} \end{array}\right]$$

Soundness: *Captures the fact that a cheating prover should not be able to convince the verifier of a false opening. Formally, for all stateful PPT adversaries* $\mathcal{A}$*:*

$$\mathsf{Pr}\left[\begin{array}{c|r} \begin{array}{c}\big(p(\alpha) \neq s\ \vee\ deg(p) > \deg\big) \\ \wedge \\ \mathsf{Verify}\big(\mathsf{srs}, \mathsf{C}, \deg, \alpha, s, \pi\big) = 1\end{array} & \begin{array}{r}\mathsf{srs} \leftarrow \mathsf{Setup}\big(\mathsf{par}, d\big) \\ (p(X), \mathsf{C}) \leftarrow \mathcal{A}\big(\mathsf{srs}\big) \\ \alpha \leftarrow \mathbb{F} \\ (s, \pi) \leftarrow \mathcal{A}\big(\alpha\big)\end{array} \end{array}\right] \approx 0$$

Evaluation Binding: *Captures the fact that no PPT adversary $\mathcal{A}$ should be able to present two valid openings for different values but same evaluation point. Formally:*

$$\mathsf{Pr}\left[\begin{array}{c|c} \mathsf{Verify}(\mathsf{srs},\mathsf{C},\deg,\alpha,s,\pi)=1, & \mathsf{srs}\leftarrow\mathsf{Setup}(\mathsf{par},N) \\ \mathsf{Verify}(\mathsf{srs},\mathsf{C},\deg,\alpha,s',\pi')=1 & (\mathsf{C},\alpha,s,s',\pi,\pi')\leftarrow\mathcal{A}(\mathsf{srs}) \\ \text{and } s\neq s' & \end{array}\right]\approx 0$$

Extractability: *Captures the fact that whenever the prover provides a valid opening, it knows a valid pair $(p(X),p(\alpha))\in\mathbb{F}[X]\times\mathbb{F}$, where $deg(p)\leq\deg$. Formally, for all PPT adversaries $\mathcal{A}$ there exists an efficient extractor $\mathcal{E}$ such that:*

$$\mathsf{Pr}\left[\begin{array}{c|r} \mathsf{Verify}(\mathsf{srs},\mathsf{C},\deg,\alpha,s,\pi)=1 & \mathsf{srs}\leftarrow\mathsf{Setup}(\mathsf{par},\deg) \\ \wedge & \mathsf{C}\leftarrow\mathcal{A}(\mathsf{srs}) \\ \big(p(\alpha)\neq s\ \vee\ deg(p)>\deg\big) & p(X)\leftarrow\mathcal{E}(\mathsf{srs},\mathsf{C},\deg) \\ & \alpha\leftarrow\mathcal{A}(\mathsf{srs},\mathsf{C},\deg) \\ & (s,\pi)\leftarrow\mathcal{A}(\mathsf{srs},p(X),\deg,\alpha) \end{array}\right]\approx 0$$

B Proof of Thm 1

Proof. We will proceed through a series of games to show that the protocol defined in Fig. 1 satisfies the linkability property. Let $\mathcal{A}$ be an arbitrary algebraic PPT adversary in the linkability game and let $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{linkability}}(\lambda)$ be their advantage. Let Game_0 be defined as in Definition 5.1, which is where we want to bound the adversary's success probability. We define $\mathsf{Game}_1,\mathsf{Game}_2$ and denote $\mathsf{Adv}_{\mathcal{A}}^{G_i}$ as the advantage of the adversary $\mathcal{A}$ in game i. We also specify reductions $\mathcal{B}_{\mathsf{unity}},\mathcal{B}_{\mathsf{ped}},\mathcal{B}_{\mathsf{dlog}},\mathcal{B}_{\mathsf{qSDH}}$ such that

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathsf{linkability}} &= \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_0}\leq\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_1}(\lambda)+\mathsf{Adv}_{\mathcal{B}_{\mathsf{unity}}}(\lambda)\\ &\leq\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2}(\lambda)+\mathsf{Adv}_{\mathcal{B}_{\mathsf{ped}}}(\lambda)+\mathsf{Adv}_{\mathcal{B}_{\mathsf{unity}}}(\lambda)\\ &\leq\mathsf{Adv}_{\mathcal{B}_{\mathsf{unity}}}(\lambda)+\mathsf{Adv}_{\mathcal{B}_{\mathsf{ped}}}(\lambda)+\mathsf{Adv}_{\mathcal{B}_{\mathsf{dlog}}}(\lambda)+\mathsf{Adv}_{\mathcal{B}_{\mathsf{qSDH}}}(\lambda)\end{aligned}$$

In Game_0 the adversary will return cm along with a proof $([z]_2=[z(x)]_2,[T]_1=[T(x)]_1,[S]_2=[S(x)]_2,\pi_{\mathsf{ped}},\pi_{\mathsf{unity}})$. We define Game_1 identically to Game_0, but after the adversary returns cm along with the proof, Game_1 additionally checks whether there exists a,b such that $z(X)=a(X-b)$ with $a^N=b^N$ and abort if this is not the case. Note that Game_1 can extract $z(X)$, the algebraic representation of $[z]_2$, because the adversary $\mathcal{A}$ is algebraic .

We observe that the adversary's advantage in Game_0 and Game_1 is identical, unless it manages to break the knowledge soundness of R_{unity}. Given such an $\mathcal{A}$, we can thus directly get a reduction $\mathcal{B}_{\mathsf{unity}}$ against the knowledge soundness of R_{unity} and let the advantage of this adversary be $\mathsf{Adv}_{\mathcal{B}_{\mathsf{unity}}}$. The reduction $\mathcal{B}_{\mathsf{unity}}$ simply runs $\mathcal{A}$ and returns π_{unity} that is returned by $\mathcal{A}$. It thus holds that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{linkability}}(\lambda)=\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_0}(\lambda)\leq\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_1}(\lambda)+\mathsf{Adv}_{\mathcal{B}_{\mathsf{unity}}}(\lambda).$$

Now define Game_2, which is identical to Game_1, but after the (algebraic) adversary $\mathcal{A}$ outputs cm the game Game_2 extracts v and r such that $\mathsf{cm}=[v+\mathsf{h}r]_1$. If this extraction fails, meaning that cm is not correctly formed, then Game_2 aborts. We note that the $\mathcal{A}$'s advantage in Game_1 is identical to its advantage in Game_2, unless it manages to break the knowledge soundness of R_{ped}. Given $\mathcal{A}$, we can construct a reduction $\mathcal{B}_{\mathsf{ped}}$ against the knowledge soundness of R_{ped} analogously to the reduction above and let the advantage of this adversary be $\mathsf{Adv}_{\mathcal{B}_{\mathsf{ped}}}$. We observe that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_1}(\lambda)\leq\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2}(\lambda)+\mathsf{Adv}_{\mathcal{B}_{\mathsf{ped}}}(\lambda).$$

Recall that any adversary who successful wins Game_2 must output a proof that satisfies the following equation from the verification procedure

$$\begin{aligned}C(x)-v-\mathsf{h}r=T(x)z(x)+\mathsf{h}S(x)\Leftrightarrow\\ C(x)-v=T(x)a(x-\omega^i)+\mathsf{h}(r+S(x)),\end{aligned}$$

while at the same time it must hold that

$$C(X) - v \neq (X - \omega^i)aT(X)$$

for any polynomial $aT(X)$, since v is not in the committed vector $\vec{c}$. Intuitively, the adversary cannot satisfy this equation, since h is unknown to the prover and thus $(r + S(X))$ is chosen independently of h. More formally, we consider two cases here. If

$$C(x) - v \neq T(x)a(x - \omega^i)$$

then we can construct a reduction $\mathcal{B}_{\mathsf{dlog}}$ breaking the discrete logarithm problem. Else if

$$C(x) - v = T(x)a(x - \omega^i)$$

then we can construct a reduction $\mathcal{B}_{\mathsf{qSDH}}$ breaking the qSDH problem.

The reduction $\mathcal{B}_{\mathsf{dlog}}$ takes as input a challenge $[y]_1$. It runs the adversary $\mathcal{A}$ against Game_2 over an srs in which $[\mathsf{h}]_1 = [y]_1$ and $\mathcal{B}_{\mathsf{dlog}}$'s choice of x (where x is the trapdoor information of the KZG commitment). Whenever the adversary returns an output $([z]_2 = [z(x)]_2, [T]_1 = [T(x)]_1, [S]_2 = [S(x)]_2, \pi_{\mathsf{ped}}, \pi_{\mathsf{unity}})$ which wins the Game_2 game, then $\mathcal{B}_{\mathsf{dlog}}$ returns

$$\mathsf{h} = \frac{C(x) - v - T(x)z(x)}{r + S(x)},$$

where $T(X), r$ and $S(X)$ are extracted from the outputs of $\mathcal{A}$. The reduction's success probability is exactly the success probability of the adversary conditioned on $(r + S(x)) \neq 0$.

The reduction $\mathcal{B}_{\mathsf{qSDH}}$ takes as input the challenge $[y_1]_1, \ldots, [y_q]_1$. It runs the following reduction $\mathcal{B}_{\mathsf{KZG}}$ as a subroutine. The $\mathcal{B}_{\mathsf{KZG}}$ runs the adversary $\mathcal{A}$ against Game_2 over an srs in which $[x]_1 = [y_1]_1$ and $\mathcal{B}_{\mathsf{KZG}}$'s choice of h. Whenever the adversary returns an output $([z]_2 = [z(x)]_2, [T]_1 = [T(x)]_1, [S]_2 = [S(x)]_2, \pi_{\mathsf{ped}}, \pi_{\mathsf{unity}})$ which wins the Game_2 game, then $\mathcal{B}_{\mathsf{KZG}}$ returns the KZG openings

$$(v, [a^{-1}T]_1) \text{ and } (C(\omega^i), [\frac{C(x) - C(\omega^i)}{x - \omega^i}]_1)$$

for $v \neq C(x)$. Then $\mathcal{B}_{\mathsf{qSDH}}$ can extract a qSDH solution from these openings following the proof in Theorem 1 of [22].

We can thus conclude that

$$\mathsf{Adv}^{\mathsf{linkability}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}_{\mathsf{unity}}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_{\mathsf{ped}}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_{\mathsf{dlog}}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_{\mathsf{qSDH}}}(\lambda).$$

Lastly, we prove the position hiding property of our construction. We define a simulator $\mathsf{Simulate}$ that has access to the trapdoor x of srs that is indistinguishable from an honest prover. First, $\mathsf{Simulate}$ calls the simulators of R_{ped} and R_{unity} on input the trapdoor x, and gets simulated proofs π'_{ped} and π'_{unity}. Then, it samples $a, r, s \leftarrow \mathbb{F}$ and sets $[z']_2 = [a]_2$, $[S']_2 = [s]_2$, $[T']_1 = (\mathsf{C} \cdot \mathsf{cm}^{-1} - [\mathsf{h}s]_1)/a$, and outputs $([z']_2, [T']_1, [S']_2, \pi'_{\mathsf{ped}}, \pi'_{\mathsf{unity}})$. Note that honestly generated $[z]_2, [S]_2$ are randomized by a and s, respectively, and thus indistinguishable from $[z']_2, [S']_2$. Finally, $[T']_1$ is the only element satisfying the verifying equation for given $[z']_2, [S']_2$ and thus indistinguishable from honest $[T]_1$ as well, which concludes the proof. □

C Proof of Lemma1

Proof. Because $z(X)$ has degree 1, there exist $a, b \in \mathbb{F}$ such that $z(X) = aX - b$.

From the first condition, we have $f(1) = a(1) = a - b$, and $f(\sigma) = a(\sigma) = a\sigma - b$. From items 2 and 3,

$$f(\sigma^2) = \frac{f(1) - f(\sigma)}{1 - \sigma} = \frac{a - a\sigma}{1 - \sigma} = a,$$
$$f(\sigma^3) = \sigma f(\sigma^2) - f(\sigma) = \sigma a - a\sigma + b = b$$

By substituting $f(\sigma^2) = a$ and $f(\sigma^3) = b$ into condition 4 we see that $f(\sigma^4) = \frac{a}{b}$. Therefore, from item 5 we have that for every $i = 0, \ldots, \log(N) - 1$, $f(\sigma^{4+i+1}) = f(\sigma^{4+i})^2 = \left(\frac{a}{b}\right)^{2^{i+1}}$. In particular, $f(\sigma^{4+(\log(N)-1)+1}) = \left(\frac{a}{b}\right)^{2^{\log(N)}} = \left(\frac{a}{b}\right)^N$, that equals 1 by the 5th condition, proves that $\frac{a}{b}$ is a Nth root of unity as required.

□

D Proof of Thm. 2

Proof. We proceed through a series of games to show that the protocol defined in Fig. 2 satisfies knowledge soundness. We set Game_0 to be the soundness game as in Def. A.1 and consider an algebraic adversary $\mathcal{A}$ against it which has advantage $\mathsf{Adv}^{\mathsf{k\text{-}sound}}_{\mathcal{A}}$. We define Game_1, Game_2 and specify reductions $\mathcal{B}_{\mathsf{qSDH}}$ and $\mathcal{B}_{\mathsf{qDHE}}$ such that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{k\text{-}sound}}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}}(\lambda) &\leq \mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_{\mathsf{qSDH}}}(\lambda)\\ &\leq \mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_{\mathsf{qDHE}}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_{\mathsf{qSDH}}}(\lambda)\\ &\leq \mathsf{Adv}_{\mathcal{B}_{\mathsf{qSDH}}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_{\mathsf{qDHE}}}(\lambda) + \mathsf{negl}(\lambda).\end{aligned}$$

In Game_0 the adversary will return $[z]_2$ along with a proof $([F]_1 = [f(x)]_1, [H]_2 = [h(x)], v_1, v_2, \pi_1, \pi_2)$. We also consider $\hat{p}(X)$, the algebraic representation of $[P]_1$ as constructed by the verifier. Note that π_2 is KZG opening proof for $p(X) = \hat{p}(X) - z(X)(\rho_1(\alpha) + \rho_2(\alpha) - z_{V_n}(\alpha)X^{d-1})$ opening to 0 at α. We define Game_1 identically to knowledge soundness, but after the adversary returns $[z]_2$ along with the proof, Game_2 additionally checks whether $f(\alpha_1) = v_1$, $f(\alpha_2) = v_2$, $p(\alpha) = 0$ and aborts otherwise. Note that Game_1 can extract $f(X)$, $h(x)$ because the adversary $\mathcal{A}$ is algebraic, and $p(X)$ is constructed from them.

We show the probability that $f(\alpha_1) = v_1$, $f(\alpha_2) = v_2$, $p(\alpha) = 0$ is bounded by qSDH. We construct a reduction $\mathcal{B}_{\mathsf{qSDH}}$ that takes as input a challenge $[y]_1, \ldots, [y_q]_1$. It runs the following reduction $\mathcal{B}_{\mathsf{KZG}}$ as a subroutine. The $\mathcal{B}_{\mathsf{KZG}}$ runs the adversary $\mathcal{A}$ against Game_0 over an srs in which $[x]_1 = [y]_1$. Whever the adversary returns an output $([F]_1 = [f(x)]_1, [H]_2 = [h(x)], v_1, v_2, \pi_1, \pi_2, \pi_3)$ that wins the Game_0 but not the Game_1 game, then $\mathcal{B}_{\mathsf{KZG}}$ returns the KZG openings

$$\Big((v_1, [F]_1) \text{ and } (f(\alpha_1), [\frac{f(x) - f(\alpha_1)}{x - \alpha_1}])\Big), \Big((v_2, [F]_1) \text{ and } (f(\alpha_2), [\frac{f(x) - f(\alpha_2)}{x - \alpha_2}])\Big) \text{ or}$$
$$\Big((0, [P]_1) \text{ and } (p(\alpha), [\frac{p(x) - p(\alpha)}{x - \alpha}])\Big)$$

for either $v_1 \neq f(\alpha_1)$, $v_2 \neq f(\alpha_2)$ or $p(\alpha) \neq 0$. Then $\mathcal{B}_{\mathsf{qSDH}}$ can extract a q-SDH solution from these openings following the proof of Theorem 3 in [22]. Thus

$$\mathsf{Adv}^{\mathsf{k\text{-}sound}}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_{\mathsf{qSDH}}}(\lambda).$$

We define Game_2 as Game_1 except that Game_2 additionally checks whether $\deg(z) \leq 1$ for $z(X)$ being the algebraic representation of $[z]_2$, and aborts otherwise. We show that $\mathcal{A}$'s advantage in both games is the same unless it breaks qDHE. Indeed, assume $\deg(z) = 2$, we construct an adversary $\mathcal{B}_{\mathsf{qDHE}}$ against qDHE. The $\mathcal{B}_{\mathsf{qDHE}}$ takes as input the challenge $[y_1]_1, \ldots, [y_q]_1$ and runs $\mathcal{A}$ against Game_1 over an srs in which $[x]_1 = [y]_1$. When $\mathcal{A}$ returns an output $([F]_1 = [f(x)]_1, [H]_2 = [h(x)], v_1, v_2, \pi_1, \pi_2)$ that wins the Game_1 but not the Game_2 game, then $\mathcal{B}_{\mathsf{qDHE}}$ extracts $\hat{p}(X) = \sum_{s=0}^{d+1} \hat{p}_s X^s$ as the algebraic representation of $[P]_1$ computed by the verifier. Note that, since $\big(-\rho_1(\alpha) - \rho_2(\alpha) - z_{V_n}(\alpha)X^{d-1}\big)z(X)$ does not vanish at $X = \alpha$, we have that $\hat{p}_{d+1} \neq 0$. Then, $\mathcal{B}_2$ sets $\hat{P}(X) = P(X) - \hat{p}_{d+1}X^{d+1}$ and outputs $\big([P]_1 - [\hat{P}(x)]_1\big)\frac{1}{\hat{p}_{d+1}} = [x^{d+1}]_1$, wining d-DHE. Thus

$$\mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_{\mathsf{qDHE}}}(\lambda).$$

Finally, let us show that

$$\mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda).$$

Consider $f(X)$, $h(X)$ the algebraic representations of $[F]_1$, $[H]_1$. The algebraic representation of the element $[P]_1$ that the verifier constructs is

$$\begin{aligned}p(X) = &-z_{V_n}(\alpha)h(X) + \big(\rho_1(\alpha) + \rho_2(\alpha)\big)f(X) + \rho_3(\alpha)\big((1-\sigma)f(X) + v_1 - v_2\big) + \rho_4(\alpha)\big(f(X) + v_2 - \sigma v_1\big)\\ &+ \rho_5(\alpha)\big(v_1 f(X) - v_2\big) + \rho_n(\alpha)\big(v_1 - 1\big) + \prod_{i \notin [5, \ldots, 4+\log(N)]} (\alpha - \sigma^i)\big(f(X) - v_1^2\big)\end{aligned}$$

Since Game_2 checks that $v_1 = f(\sigma^{-1}\alpha), v_2 = f(\sigma^{-2}\alpha)$, we can replace these values and see that

$$\begin{aligned}p(X) = &-z_{V_n}(\alpha)h(X) + \big(\rho_1(\alpha) + \rho_2(\alpha)\big)f(X) + \rho_3(\alpha)\big((1-\sigma)f(X) + f(\sigma^{-1}\alpha) - f(\sigma^{-2}\alpha)\big)\\ &+ \rho_4(\alpha)\big(f(X) + f(\sigma^{-2}\alpha) - \sigma f(\sigma^{-1}\alpha)\big) + \rho_5(\alpha)\big(f(\sigma^{-1}\alpha)f(X) - f(\sigma^{-2}\alpha)\big)\\ &+ \rho_n(\alpha)\big(f(\sigma^{-1}\alpha) - 1\big) + \prod_{i\notin[5,\ldots,4+\log(N)]}(\alpha - \sigma^i)\big(f(X) - f(\sigma^{-1}\alpha)^2\big)\end{aligned}$$

Now, because $p(\alpha) = 0$ and α has been chosen by the verifier after the prover has sent $[H]_1, [F]_1$, except in the negligible case that α is a root of $p(X)$, we have that $p(X) \equiv 0$, i.e,

$$\begin{aligned}z_{V_n}(X)h(X) = &-\big(\rho_1(X) + \rho_2(X)\big)f(X) + \rho_3(X)\big((1-\sigma)f(X) + f(\sigma^{-1}X) - f(\sigma^{-2}X)\big)\\ &+ \rho_4(X)\big(f(X) + f(\sigma^{-2}X) - \sigma f(\sigma^{-1}X)\big) + \rho_5(X)\big(f(\sigma^{-1}X)f(X) - f(\sigma^{-2}X)\big)\\ &+ \rho_n(X)\big(f(\sigma^{-1}X) - 1\big) + \prod_{i\notin[5,\ldots,4+\log(N)]}(X - \sigma^i)\big(f(X) - f(\sigma^{-1}X)^2\big)\end{aligned}$$

$z_{V_n}(X)$ divides the right side of the equation and thus, the latter vanishes for all the powers $\{\sigma^i\}_{i=0}^{n-1}$. This implies that

- $f(1) = a(1)$, $f(\sigma) = a(\sigma)$
- $f(\sigma^2) = \frac{v_2 - v_1}{1-\sigma} = \frac{f(\sigma^2\sigma^{-2}) - f(\sigma^2\sigma^{-1})}{1-\sigma} = \frac{f(1) - f(\sigma)}{1-\sigma}$
- $f(\sigma^3) = rf(\sigma^3\sigma^{-1}) - f(\sigma^3\sigma^{-2}) = rf(\sigma^2) - f(\sigma)$
- $f(\sigma^4)f(\sigma^4\sigma^{-1}) = f(\sigma^4\sigma^{-2})$, i.e, $f(\sigma^4)f(\sigma^3) = f(\sigma^2)$
- $1 = f(\sigma^{5+\log(N)}\sigma^{-1}) = f(\sigma^{4+\log(N)})$
- $\big(f(\sigma^{4+i+1}) - f(\sigma^{4+i+1}\sigma^{-1})f(\sigma^{4+i+1}\sigma^{-1})\big)\prod_{j\notin[5,\ldots,4+\log(N)]}(\sigma^i - \sigma^j) = 0$ for all $i = 0, \ldots, \log(N) - 1$.

  Note that $\prod_{j\notin[5,\ldots,4+\log(N)]}(\sigma^i - \sigma^j) \neq 0$ implies that $0 = f(\sigma^{4+i+1}) - f(\sigma^{4+i+1}\sigma^{-1})f(\sigma^{4+i+1}\sigma^{-1}) = f(\sigma^{4+i+1}) - f(\sigma^{4+i})^2$.

By Lemma 1 we have that $z(X) = aX - b$ where $\frac{a}{b}$ is an N-th root of unity.

For zero-knowledge, we define a simulator $\mathsf{Simulate}$ that has access to the trapdoor of srs and is indistinguishable from an honest prover. The simulator first chooses s_1, s_2, v_1, v_2 uniformly at random and sets $[F]_1 = [s_1]_1$ and $[H]_1 = [s_2]_1$. It computes $\alpha_1 = \sigma^{-1}\alpha$, $\alpha_2 = \sigma^{-2}\alpha$. It then computes $[w_1]_1 = \big([F]_1 - v_1\tau_1(x) - v_2\tau_2(x)\big)\frac{1}{(x-\alpha_1)(x-\alpha_2)}$, for $\tau_1(x) = \frac{x-\alpha_2}{\alpha_1-\alpha_2}$, $\tau_2(x) = \frac{x-\alpha_1}{\alpha_2-\alpha_1}$.

It sets $[P]_1$ the same as the verifier i.e.

$$\begin{aligned}[P]_1 = &-[H]_1 z_{V_n}(\alpha) + [F]_1\big(\rho_1(\alpha) + \rho_2(\alpha)\big) + \big([F]_1(1-\sigma) - v_2 + v_1\big)\rho_3(\alpha) + \big([F]_1 + v_2 - \sigma v_1\big)\rho_4(\alpha)\\ &+ \big([F]_1 v_1 - v_2\big)\rho_5(\alpha) + \big(v_1 - 1\big)\rho_n(\alpha) + \big([F]_1 - v_1^2\big)\prod_{i\notin[5,\ldots,4+\log(N)]}(\alpha - \sigma^i)\end{aligned}$$

and then computes $[w_2]_1 = [P]_1\frac{1}{x-\alpha}$. It returns $([F]_1, [H]_1, v_1, v_2, \pi_1 = [w_1]_1, \pi_2 = [w_2]_2)$.

We must argue that the simulators output is distributed identically to the honest provers. Then the provers components are randomised by

$$\begin{aligned}F: &\quad r_0\rho_{5+\log(N)}(x) & \qquad H: &\quad r(x)\\ v_1: &\quad r(\sigma^{-1}\alpha)z_{V_n}(\alpha) & \qquad v_2: &\quad r(\sigma^{-2}\alpha)z_{V_n}(\alpha)\end{aligned}$$

and the elements $[w]_1, [w]_2, [w]_3$ are the unique elements satisfying the verifies equations given $[F]_1, [H]_1, v_1, v_2$. The probability that $r_1\rho_{6+\log(N)}(x)$, $r(x)$, $r(\sigma^{-1}\alpha)z_{V_n}(\alpha)$, $r(\sigma^{-2}\alpha)z_{V_n}(\alpha)$ are dependent at random α is negligible because $r(X)$ is a random degree 2 polynomial and the probability that $\sigma^{-1}\alpha = x$ or $\sigma^{-2}\alpha = x$ is $\frac{2}{|\mathbb{F}|}$. Where the simulators terms $[F]_1, [H]_1, v_1, v_2$ are chosen uniformly at random and $[w_1]_1, [w_2]_1$ are the unique terms that satisfy the verifies equations, we have that these distributions are identical except with negligible probability. □

E Proof of Thm. 3

Proof. We will proceed through a series of games to show that the protocol defined in Fig. 3 satisfies linkability as defined in Def. 5.1. Let $\mathcal{A}$ be an arbitrary PPT adversary in the linkability game with advantage $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{linkability}}(\lambda)$. We define Game_1, Game_2 and specify reductions $\mathcal{B}_1$ and $\mathcal{B}_2$ such that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{linkability}}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{qSDH}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{k\text{-}sound}}(\lambda) + \mathsf{negl}(\lambda)$$

Let us transition from the linkability game for the protocol of Fig. 3 to a game Game_1. Game_1 behaves as linkability except that when $\mathcal{A}$ returns v_1, v_2, Game_1 checks whether $u(\alpha) = v_1$, $p_1(v_1) = v_2$, and $p_2(\alpha) = 0$, for $u(X), p_1(X), p_2(X)$, the algebraic representations of $[u]_1, [P_1]_1 = [z_I]_1 + \chi[C_I]_1$, and $[P_2]_1 = v_2 - \chi\mathsf{cm} - z_{V_m}(\alpha)[H_2]_1$. If not then Game_1 aborts. We design $\mathcal{B}_1$ such that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{linkability}}(\lambda) \leq \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_1}(\lambda) + \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{qSDH}}(\lambda)$$

Indeed, assume that $\mathcal{A}$ succeeds against linkability but not Game_1. Then this corresponds to the case where $\mathcal{A}$ returns verifying v_1, v_2, π_1, π_2, π_3 but the equality does not hold for some $p(X) \in \{u(X), p_1(X), p_2(X)\}$. Thus $\mathcal{B}_1$ takes as input a challenge $[y_1]_1, \ldots, [y_q]_1$ and runs the following reduction $\mathcal{B}_{KZG}$ as a subroutine. The $\mathcal{B}_{KZG}$ runs the adversary $\mathcal{A}$ against Game_0 over an srs in which $[x]_1 = [y_1]_1$. Whenever the adversary wins the Game_0 but not the Game_1 game, then $\mathcal{B}_{KZG}$ returns the KZG opening

$$(v, \pi) \text{ and } (f(\alpha), [(f(x) - f(\alpha))/(x - \alpha)]_1)$$

for $(v, f(X))$ corresponding to either $(v_1, u(X))$, $(v_2, p_1(X))$, $(v_3, p_2(X))$ and π the corresponding proof. Then $\mathcal{B}_{\mathsf{qSDH}}$ can extract a solution from these openings following the proof in Theorem 1 in [22].

Now let us transition to a new game. Game_2 behaves identically except that when $\mathcal{A}$ returns $[u]_1$, then Game_2 checks whether its algebraic representation $u(X)$ is such that $u(\nu^j)^N = 1$ for all j. If not then Game_2 aborts. We design $\mathcal{B}_2$ such that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_1}(\lambda) \leq \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2}(\lambda) + \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{k\text{-}sound}}(\lambda)$$

Assume that $\mathcal{A}$ succeeds against Game_1 but not Game_2. Then $\mathcal{B}_2$ chooses $[u]_1 = [u(x)]_1$ in its own game and uses it as input to run $\mathcal{A}$. When $\mathcal{A}$ returns π_{unity}, $\mathcal{B}_2$ forwards it and wins knowledge-soundness of Π_{unity} whenever $\mathcal{A}$ succeeds.

Next we transition to a game Game_3 that behaves as Game_2 except that when $\mathcal{A}$ returns its proof, Game_3 checks whether $C(X) - C_I(X) = z_I(X)H_1(X)$, for $C(X), C_I(X), z_I(X), H_1(X)$ the algebraic representations of $[C]_1, [C_I]_1, [H_1]_2, [z_I]_1$. If not then Game_3 aborts. We design $\mathcal{B}_3$ such that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2}(\lambda) \leq \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_3}(\lambda) + \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{qSDH}}(\lambda)$$

The $\mathcal{B}_3$ takes as input a challenge $[y_1]_1, \ldots, [y_q]_1$ and runs the adversary $\mathcal{A}$ against Game_2 over an srs in which $[x]_1 = [y_1]_1$. Whenever the adversary wins the Game_2 but not the Game_3 game, then $\mathcal{B}_3$ learns

$$d(X) = C(X) - C_I(X) - z_I(X)H_1(X)$$

such that $d(x) = 0$ and $d(X) \neq 0$. Thus $\mathcal{B}_3$ returns $(1, [1/(x-1)]_1)$ as a valid q-SDH solution.

Finally we show that the probability that Game_3 returns 1 but that for some $j \in [m]$, and for $\vec{c}$ such that $C(X) = \sum_{i=1}^{N} c_i \lambda_i(X)$,

$$\phi(\nu^j) \notin \vec{c}$$

is negligible.

Recall that $p_2(\alpha) = v_2 - \chi\mathsf{cm} - z_{V_m}(\alpha)H_2(\alpha) = z_I(v_1) + \chi C_I(v_1) - \chi\mathsf{cm} - z_{V_m}(\alpha)H_2(\alpha) = z_I(u(\alpha)) + \chi C_I(u(\alpha)) - \chi\mathsf{cm} - z_{V_m}(\alpha)H_2(\alpha) = 0$. First, because α has been sent by the verifier after the prover commits to $\phi(X), z_I(X), u(X), H_2(X)$ and $C_I(X)$, we have that

$$z_I(u(X)) + \chi C_I(u(X)) - \chi\phi(X) - z_{V_m}(X)H_2(X) = 0$$

for all X except with negligible probability. Further, because χ has been sent by the verifier after the prover commits, we have that there exists $H_{2,1}(X)$ and $H_{2,2}(X)$ such that

$$0 = z_I(u(X)) - z_{V_m}(X)H_{2,1}(X)$$
$$0 = C_I(u(X)) - \phi(X) - z_{V_m}(X)H_{2,2}(X)$$

except with negligible probability.

Thus,

$$z_I(u(\nu^j)) = z_I(\omega^{i_j}) = 0 \text{ for all } j = 1, \ldots, m.$$

and $z_I(X) = \prod_{j=1}^{m}(X - \omega^{i_j})\hat{z}(X) = \prod_{i \in I}(X - \omega^i)\hat{z}(X)$, for some polynomial $\hat{z}(X)$. From the second equation we also we have that

$$C_I(u(\nu^j)) = \phi(\nu^j) \; \forall \; j \in [m], \text{i.e., } C_I(\omega^{i_j}) = \phi(\nu^j).$$

Using

$$C(u(X)) - C_I(u(X)) = z_I(u(X))H_1(u(X))$$

we hence gets that

$$0 = C(u(\nu^j)) - C_I(u(\nu^j)) = C(\omega^k) - \phi(\nu^j)$$

which concludes the proof.

□

F Proof of Thm. 5

Proof. We first define a simulator $\mathsf{Simulate}$ and then argue that their transcript is indistinguishable from an honest provers transcript. The $\mathsf{Simulate}$ subverts the setup algorithm such that it knows the secret x contained in $[x]_1, [x^2]_1, [x^3]_1, \ldots$. It takes as input some instance (C, cm) and aims to generate a verifying transcript.

It samples $s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8 \leftarrow \mathbb{F}$ at random and outputs $[C_I]_1 = [s_1]_1, [z_I]_1 = [s_2]_1$, $[u]_1 = [s_3]_1$, $[H_1]_2 = [(C - s_1)/s_2]_2$ and a simulated proof π_{unity} that we describe in the next paragraph. After receiving χ it outputs $[H_2]_1 = [s_4]_1$. After receiving α it outputs $v_1 = s_5$, $v_2 = s_6$. and

$$\begin{aligned}
\pi_1 &= [(u - v_1)/(x - \alpha)]_1 \\
\pi_2 &= [(z_I + \chi C_I)/(x - v_1)]_1 \\
\pi_3 &= [(v_2 - \chi \mathsf{cm} - z_{V_m}(\alpha)H_2)/(x - \alpha)]
\end{aligned}$$

To simulate π_{unity} the simulate $\mathsf{Simulate}$ outputs $[\bar{U}]_1 = [s_7]_1$, $[h_2]_1 = [s_8]_1$. After receiving α it outputs $[h_1]_1 = [s_9]_1$. After receiving β it outputs $[\bar{U}_\alpha]_1 = [s_{10}]_1$, $[h_{2,\alpha}] = [s_{11}]_1$ and $v_1 = s_{12}$, $v_2 = s_{13}$, $v_3 = s_{14}$ and

$$\begin{aligned}
\pi_1 &= [(u - v_1)/(x - \alpha)]_1 \\
\pi_2 &= [(\bar{U} + \bar{U}_\alpha)/(x - \alpha)]_1 \\
\pi_3 &= [(h_2 - h_{2,\alpha})/(x - \alpha)]_1 \\
\pi_4 &= [x^{\mathsf{max_deg}-n}(\bar{U}_\alpha + \ell(x))/(x - 1)(x - \beta)(x - \beta\sigma)]_1 \\
\pi_5 &= [x^{\mathsf{max_deg}-n}((v_1\rho_1(\beta) + v_2)^2 - h_1 z_{V_n}(\beta) - (v_3 + \mathsf{id}(\alpha)\rho_n(\beta)) - z_{V_m}(\alpha)h_{2,\alpha})/(x - \beta)]_1
\end{aligned}$$

where $\ell(x)$ is the polynomial that interpolates to $(0, v_2, v_3)$ at $(1, \beta\beta\sigma)$.

We now argue $\mathsf{Simulate}$'s output is indistinguishable from an honest prover's output.

We consider each of the elements in Fig. 3 separately and argue they are identically distributed with overwhelming probability.

- $[C_I]_1$ is blinded by r_2 for the prover and s_1 for the simulator.
- $[z_I]_1$ is blinded by r_1 for the prover and s_2 for the simulator.
- $[u]_1$ is blinded by r_5 for the prover and s_3 for the simulator.
- $[H_1]_2$ is the unique element satisfied by the pairing check for both the prover and simulator given $[C_I]_1$ and $[z_I]_1$.
- $[H_2]_1$ is blinded by r_3 for the prover and s_4 for the simulator. Note that $r_3 \frac{\chi u(x) z_I(u(x))}{z_{V_m(x)}}$ is non-zero with overwhelming probability.

- v_1 is blinded by r_6 for the prover and s_5 for the simulator. Note that $r_6 \alpha z_{V_m}(\alpha)$ is non-zero with overwhelming probability.
- v_2 is blinded by r_4 for the prover and s_6 for the simulator. Note that $r_4 u^2 \alpha z_I(u(\alpha))$ is non-zero with overwhelming probability.
- π_1, π_2, π_3 are the unique element satisfied by the KZG opening checks for both the prover and the simulator.

Finally we consider each of the elements in Fig. 4 separately and argue they are identically distributed with overwhelming probability.

- $[\bar{U}]_1$ is blinded by t_1 for the prover and s_7 for the simulator.
- $[h_2]_1$ is blinded by t_2 for the prover and s_8 for the simulator. Note that there exists a $\rho_2(x)t_2$ term in the provers $[h_2]_1$ which is linearly independent from all other terms and thus not cancelled with overwhelming probability.
- $[h_1]_1$ is blinded by t_3 for the prover and s_9 for the simulator. Note that there is a $t_3^2 z_{V_m}^2(\alpha) \frac{\rho_4^2(x) - \rho_4(x)}{z_{V_n}(x)}$ term in the provers $[h_1]_1$ which is linearly independent from all other terms.
- $[\bar{U}_\alpha]_1$ is blinded by t_4 for the prover and s_{10} for the simulator. Note that there is a $t_4 z_{V_m}(\alpha)\rho_5(x)$ term in the provers $[\bar{U}_\alpha]_1$ which is linearly independent from all other terms.
- $[h_{2,\alpha}]_1$ is blinded by t_5 for the prover and s_{11} for the simulator. Note that there is a $\rho_2(x)t_2$ term in the provers $[h_{2,\alpha}]_1$ which is linearly independent from all other terms.
- v_1 is blinded by r_7 for the prover and s_{12} for the simulator.
- v_2 is blinded by t_5 for the prover and s_{13} for the simulator. Note that there is a $t_5 z_{V_m}(\alpha)\rho_6(\beta)$ term in the provers v_2 which is linearly independent from all other terms.
- v_3 is blinded by t_6 for the prover and s_{14} for the simulator. Note that there is a $t_6 z_{V_m}(\alpha)\rho_7(\beta)$ term in the provers v_3 which is linearly independent from all other terms.
- $\pi_1, \pi_2, \pi_3, \pi_4, \pi_5$ are the unique elements satisfied by the KZG opening checks for both the prover and the simulator.

□