Caulk+: Table-independent lookup arguments

Jim Posen[1] and Assimakis A. Kattis[2]

[1] Ulvetanna jimpo AT ulvetanna.io
[2] New York University kattis AT cs.nyu.edu

Abstract. The recent work of Caulk [ZBK+22] introduces the security notion of *position hiding linkability* for vector commitment schemes, providing a zero-knowledge argument that a committed vector's elements comprise a subset of some other committed vector. The protocol has very low cost to the prover in the case where the size m of the subset vector is much smaller than the size n of the one containing it. The asymptotic prover complexity is $O(m^2 + m \log n)$, where the $\log n$ dependence comes from a subprotocol showing that the roots of a blinded polynomial are all nth roots of unity. In this work, we show how to simplify this argument, replacing the subprotocol with a polynomial divisibility check and thereby reducing the asymptotic prover complexity to $O(m^2)$, removing any dependence on n.

Keywords: polynomial committments · vector commitments · zero-knowledge

1 Introduction

The work in [ZBK+22], named Caulk, introduces the notion of *position-hiding linkability* for vector commitment schemes. Two efficient schemes for linking to a committed n-length vector are provided: one for arguing membership of a single committed element and one for arguing multimembership of an m-length subvector. The two protocols use the polynomial commitment scheme of [KZG10], or KZG commitment, over appropriately sized subgroups as the vector commitment scheme of choice. In particular, an n-length vector commitment can be constructed as a commitment to the polynomial interpolation of the vector elements over an order-n multiplicative subgroup of the field.

The asymptotic prover efficiency for the single-element and m-element subvector membership arguments are $O(\log n)$ and $O(m \log n + m^2)$ respectively. Caulk achieves sublinear proving times by precomputing vector commitment opening witnesses that take $O(n)$ time to compute naively. Since the evaluation set over which the vector elements lie is a multiplicative subgroup, there is an efficient method to aggregate the precomputed witnesses into a batched witness [TAB+20]. These elements are blinded with randomly sampled elements during the proving phase to provide the position-hiding property.

1.1 Our Contribution

We present an improvement to the position-hiding linkability arguments of Caulk which reduces the prover complexity to $O(m^2)$ for an m-element membership proof, removing any dependence on the value of n. The $\log n$ asymptotic factor in Caulk comes from a subprotocol for the claim that certain blinded evaluation points of the committed polynomial are n-th roots of unity. Our optimization stems from replacing this with a pairing check constraining the evaluation points to be roots of a polynomial dividing $X^n - 1$. While this modification requires the prover to precompute and store one extra witness element per vector index in addition to the one already required in the original scheme, it enjoys improved concrete efficiency and a simpler implementation.

There are two challenges to overcome in constructing this protocol. The prover generates a randomized polynomial of the form $Z(X) = r\prod_{i\in I}(X-\omega^i)$ with a multiplicative blinding factor r, where ω is a primitive n-th root of unity and I is a non-empty subset of $[n]$. A divisibility check that $Z(X)H(X) = X^n - 1$ for some quotient polynomial H guarantees the claim, except for the condition that Z has at least one root. Since Z has only a single degree of blinding, we must be careful when showing that Z has a root so as not to leak any information about that root. The other challenge is that computing the quotient polynomial H can take $O(n)$ time. To circumvent this, we leverage even further the ability to precompute and store witness elements for pairing checks, which is already a core part of the original proof system.

Scheme	Proof size	Prover work	Verifier work
Caulk, lookup table	$14\mathbb{G}_1, 1\mathbb{G}_2, 4\mathbb{F}$	$O_\lambda(m^2 + m\log n)$	$O_\lambda(\log m), 4\text{P}$
Caulk, Pedersen link	$6\mathbb{G}_1, 2\mathbb{G}_2, 4\mathbb{F}$	$O_\lambda(\log n)$	$O_\lambda(1), 4\text{P}$
This work, lookup table	$7\mathbb{G}_1, 1\mathbb{G}_2, 2\mathbb{F}$	$O_\lambda(m^2)$	$O_\lambda(\log m), 3\text{P}$
This work, Pedersen link	$10\mathbb{G}_1, 1\mathbb{G}_2, 5\mathbb{F}$	$O_\lambda(1)$	$O_\lambda(1), 3\text{P}$

Figure 1: Comparison of this work with prior work

2 Preliminaries

2.1 Notation

Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be cyclic groups of prime order p, written with additive notation. The finite field $\mathbb{F}_p$ with p elements will sometimes be abbreviated as $\mathbb{F}$. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a pairing: an efficiently computable, non-degenerate bilinear map. Let there be generators $[1]_1, [1]_2, [1]_T$ of $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$, respectively, with $[1]_T = e([1]_1, [1]_2)$. For all elements $\alpha \in \mathbb{F}$ and $\gamma \in \{1, 2, T\}$, the notation $[\alpha]_\gamma$ represents the element $\alpha[1]_\gamma \in \mathbb{G}_\gamma$. The set of polynomials over $\mathbb{F}$ of degree at most d is written as $\mathbb{F}_{\leq d}[X]$. For any set $S \subset \mathbb{F}, Z_S(X) := \prod_{v\in S}(X - v) \in \mathbb{F}[X]$ is the monic polynomial of degree $|S|$ which vanishes on S. The set of powers of a value can be written $x^S := \{x^i\}_{i\in S}$ for $S \subset \mathbb{Z}$.

2.2 Algebraic Group Model

We analyze security of our protocols in the Algebraic Group Model (AGM) [FKL18]. In the AGM, whenever an adversary outputs a group element $\boldsymbol{a} \in \mathbb{G}_\gamma$ with $\gamma \in \{1, 2\}$, they also output an *algebraic representation* as a linear combination of the $\mathbb{G}_\gamma$ elements that the adversary has access to from the public parameters and structured reference string (SRS).

2.3 Real and Ideal Pairing Checks

We borrow the terminology of real and ideal pairing checks from [GWC19]. An SRS has degree q if its elements equal $\text{SRS}_i = [f(x)]_i$ for uniformly sampled $x \in_R \mathbb{F}$ and some $f \in \mathbb{F}_{<q}[X]$, where $i \in [q]$. Let $f_{i,j}$ denote the corresponding polynomial for the j-th element of SRS_i and a, b be the vectors in $\mathbb{F}^q$ whose encodings in $\mathbb{G}_1, \mathbb{G}_2$ are returned by algebraic adversary $\mathcal{A}$. A *real pairing check* is defined as:

$$(a \cdot T_1) \cdot (T_2 \cdot b) = 0,$$

for some matrices T_1, T_2 over $\mathbb{F}$. Real pairing checks can be efficiently computed from the encoded elements and pairing function $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

If we operate in the AGM, for each output $[a_j]_i$, $\mathcal{A}$ also outputs a vector v for which $a_j = \sum v_\ell f_{i,\ell}(x) = R_{i,j}(x)$ for $R_{i,j}(X) := \sum v_\ell f_{i,\ell}(X)$. For $i \in \{1, 2\}$, let $R_i = (R_{i,j})_j$ be a vector of polynomials over $\mathbb{F}$. The corresponding ideal check then is given by:

$$(R_1 \cdot T_1) \cdot (T_2 \cdot R_2) \equiv 0.$$

2.4 Cryptographic Assumptions

We use the formulation of the q-DLOG assumption from [GWC19], Definition 2.1.

Definition 1 (q-DLOG assumption, [GWC19] Definition 2.1, verbatim)**.** Fix integer q. The q-DLOG assumption states that given

$$[1]_1, [x]_1, ..., [x^q]_1, [1]_2, [x]_2, ..., [x^q]_2$$

for uniformly chosen $x \in \mathbb{F}$, the probability of an efficient $\mathcal{A}$ outputting x is negligible.

We also use Lemma 2.2 from [GWC19], which follows from the q-DLOG assumption in the AGM.

Lemma 1 ([GWC19] Lemma 2.2, verbatim)**.** *Assume the q-DLOG for* $(\mathbb{G}_1, \mathbb{G}_2)$*. Given an algebraic adversary* $\mathcal{A}$ *participating in a protocol with a degree q SRS, the probability of any real pairing check passing is larger by at most an additive negligible factor than the probability the corresponding ideal check holds.*

2.5 Vector Commitments

We recall vector commitment schemes with a trusted setup. Here PP denotes the public parameters of the protocol.

Setup$(\text{PP}) \to (\text{SRS}, x)$. Given the public parameters, perform the trusted setup, producing structured reference string SRS and trapdoor x.

Commit$(\text{PP}, \text{SRS}, \vec{c}, r) \to C$. Given an input vector $\vec{c}$ and randomness r, produce a commitment C.

Prove$_{\text{Open}}(\text{PP}, \text{SRS}, C, i, c_i, \vec{c}, r) \to \pi_i$. Given a commitment C and claimed opening (i, c_i) along with the committed vector $\vec{c}$ and randomness r, produce a proof π_i.

Verify$_{\text{Open}}(\text{PP}, \text{SRS}, C, i, c_i, \pi_i) \to \{0, 1\}$. Given a commitment C and claimed opening (i, c_i), verify the opening proof π_i.

Prove$_{\text{Open}}$ and **Verify**$_{\text{Open}}$ can be generalized to public-coin interactive protocols allowing interaction between prover and verifier. We write **Verify**$^{\mathcal{P}}_{\text{Open}}$ for the verification algorithm interacting with a prover $\mathcal{P}$. The following security property is associated with vector commitments:

Definition 2 (Position Binding)**.** A vector commitment is *position binding* if for all efficient adversaries $\mathcal{A}$ the following probability is negligible:

$$\Pr\left[\begin{array}{l} c_i \neq c'_i \\ \text{Verify}^{\mathcal{A}}_{\text{Open}}(\text{PP}, \text{SRS}, C, i, c_i, \pi) = 1 \\ \text{Verify}^{\mathcal{A}}_{\text{Open}}(\text{PP}, \text{SRS}, C, i, c'_i, \pi') = 1 \end{array} \middle| \begin{array}{rl} \text{SRS} & \leftarrow \text{Setup}(\text{PP}) \\ (C, i, c_i, \pi, c'_i, \pi') & \leftarrow \mathcal{A}(\text{PP}, \text{SRS}) \end{array} \right].$$

2.6 Position-Hiding Linkability

We restate the definition of position-hiding linkable vector commitments as stated in [ZBK+22], which extends the definition of a vector commitment scheme. A vector commitment scheme has position-hiding linkability if there is a zero-knowledge argument of knowledge for the following witness relation:

$$\mathcal{R}_{\text{Link}} := \left\{ \left(\begin{array}{l} \text{PP}, \text{SRS} \,; \\ C, A, n, m \,; \\ \vec{c}, r_c, \vec{a}, r_a \end{array} \right) \middle| \begin{array}{l} \text{Commit}(\vec{c}, r_c) = C \\ \text{Commit}(\vec{a}, r_a) = A \\ \forall i \in [m], \exists j \in [n], a_i = c_j \end{array} \right\}.$$

2.7 Zero-Knowledge with Precomputation

We use the standard definition of honest-verifier zero-knowledge for public-coin interactive protocols: informally, that there exists an efficient algorithm **Simulate** which can produce an accepting transcript that is computationally indistinguishable from a real one between a prover and an honest verifier. In our setting, the prover is allowed to precompute advice inputs from the public parameters and SRS to reduce its online execution time when given an instance. We consider a model where the distinguisher cannot discriminate based on timing information between an execution where the prover has precomputed advice and one where they have not, assuming the precomputation is polynomial-time. In practice, if timing information is available the prover will precompute and store advice for all instances it may generate proofs for.

3 Lookup Argument Construction

Our protocol is based on the one in section 7 of [ZBK+22]. The commitments $\boldsymbol{c}$ and $\boldsymbol{a}$ are to vectors $\vec{c} \in \mathbb{F}^n$ and $\vec{a} \in \mathbb{F}^m$ respectively. Assume both n and m are powers of two[1]. Let $\mathbb{H}$ and $\mathbb{V}$ be multiplicative subgroups of $\mathbb{F}$ of size n and m. The vector commitment scheme used is a KZG commitment over a polynomial which evaluates to the committed vector over some multiplicative subgroup of $\mathbb{F}$. In particular, $\boldsymbol{c}$ commits to a polynomial $C(X)$ which evaluates to $\vec{c}$ over $\mathbb{H}$ and $\boldsymbol{a}$ commits to a polynomial $A(X)$ which evaluates to $\vec{a}$ over $\mathbb{V}$. Let ω be a generator of $\mathbb{H}$ and ν be a generator of $\mathbb{V}$. The protocol is then a zero-knowledge argument for the following relation $\mathcal{R}^{\text{KZG}}_{\text{Link}}$, which instantiates $\mathcal{R}_{\text{Link}}$ with KZG commitments:

$$\mathcal{R}^{\text{KZG}}_{\text{Link}} := \left\{ \left(\begin{array}{l} \{[x^{k-1}]_1, [x^{k-1}]_2\}_{k \in [d]} \,; \\ \boldsymbol{c}, \boldsymbol{a}, \mathbb{H}, \mathbb{V}, \omega, \mu \,; \\ C(X), A(X), I \subset [n] \end{array} \right) \middle| \begin{array}{l} \boldsymbol{c} = [C(x)]_1 \\ \boldsymbol{a} = [A(x)]_1 \\ \forall i \in [m], \exists j \in I, A(\mu^i) = C(\omega^j) \end{array} \right\}. \quad (1)$$

Let $I \subset [n]$ be the set of indices in $\vec{c}$ that $\vec{a}$ takes values from and $u : [m] \to I$ be a mapping such that $a_i = c_{u(i)}$ for all $i \in [m]$. The protocol begins with the prover computing polynomials $Z_I, C_I, U \in \mathbb{F}_{\leq m}[X]$ so that the following polynomial identities hold over Z_I:

$$C(X) - C_I(X) = 0 \mod Z_I, \quad (2)$$

$$Z_{\mathbb{H}} = 0 \mod Z_I, \quad (3)$$

and the following identities hold over $Z_{\mathbb{V}}$:

[1] The vectors can always be padded with duplicate elements up to the nearest power of two length.

$$C_I(U(X)) - A(X) = 0 \mod Z_{\mathbb{V}}, \tag{4}$$
$$Z_I(U(X)) = 0 \mod Z_{\mathbb{V}}. \tag{5}$$

Intuitively, Z_I is a low-degree polynomial which vanishes on ω^I, C_I is a low-degree polynomial which agrees with C on ω^I, and U maps $\mathbb{V}$ to ω^I. Concretely, the prover computes the Lagrange polynomials $\{\tau_i\}_{i\in I} \subset \mathbb{F}_{<|I|}[X]$ over ω^I and the Lagrange polynomials $\{\mu_j\}_{j\in[m]} \subset \mathbb{F}_{<m}[X]$ over $\mathbb{V}$. They then define:

$$Z_I(X) = \prod_{i\in I} X - \omega^i,$$
$$C_I(X) = \sum_{i\in I} c_i \tau_i(X),$$
$$U(X) = \sum_{j\in[m]} u(j)\mu_j(X).$$

Now, we *could* proceed with the standard compilation of polynomial IOPs to regular IOPs. However, equations 2 and 3 involve polynomials with degree up to n, so computing a KZG commitment opening would take $O(n)$ time. We notice that neither equation involves polynomial composition and so we can enforce the constraints with real pairing checks at the point x from the structured reference string SRS. This approach has the benefit that the quotient elements for the pairing check can be computed in $O(m^2)$ time from precomputed values.

Define $W_1, W_2 \in \mathbb{F}_{<n}[X]$ to be such that $C - C_I = Z_I W_1$ and $Z_{\mathbb{H}} = Z_I W_2$. The prover will look up precomputed values $\{[W_1^{(i)}(x)]_2, [W_2^{(i)}(x)]_2\}_{i\in I}$, where $W_1^{(i)}(X) = (C(X) - c_i)/(X - \omega^i), W_2^{(i)}(X) = Z_{\mathbb{H}}/(X - \omega^i)$, and then compute:

$$[W_1(x)]_2 = \sum_{i\in I} \frac{[W_1^{(i)}(x)]_2}{\prod_{j\in I, i\neq j} \omega^i - \omega^j},$$
$$[W_2(x)]_2 = \sum_{i\in I} \frac{[W_2^{(i)}(x)]_2}{\prod_{j\in I, i\neq j} \omega^i - \omega^j}.$$

After sending these quotient elements for the pairing checks corresponding to equations 2 and 3, the verifier will query equations 4 and 5 at a challenge point α. The prover will provide polynomial commitment opening proofs which can be computed in $O(m \log m)$ time due to the lower degree bound on the polynomials involved.

The final ingredient is to blind Z_I, C_I, U appropriately to preserve zero-knowledge. While C_I and U can be blinded by respectively adding multiples of Z_I and $Z_{\mathbb{V}}$, Z_I can only be blinded by a single multiplicative factor. At first glace, this presents a problem because the prover must present the evaluation of Z_I at a challenge point during the last step of the protocol and there may not be sufficient degrees of randomness to blind both the evaluation and commitment to Z_I itself. Fortunately however, the KZG openings can be batched together using verifier-supplied randomness in such a way that additional blinding of C_I prevents information leakage.

Figure 2: Interactive Protocol for $\mathcal{R}_{\text{Link}}^{\text{KZG}}$

Public inputs:

- Prime order cyclic groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ with bilinear map e and generators $[1]_1, [1]_2$
- Scalar field $\mathbb{F}$
- Structured reference string $[x]_1, ..., [x^{d-1}]_1, [x]_2, ..., [x^{d-1}]_2$

Common inputs:

- Multiplicative subgroup $\mathbb{H} < \mathbb{F}^*$ with order n and generator ω
- Multiplicative subgroup $\mathbb{V} < \mathbb{F}^*$ with order m and generator ν
- KZG commitment $\boldsymbol{c}$ to $C(X)$ with evaluation points in $\mathbb{H}$
- KZG commitment $\boldsymbol{a}$ to $A(X)$ with evaluation points in $\mathbb{V}$

Witness inputs:

- Set of indices $I \subset [n]$
- Values $\{c_i\}_{i \in I}$
- Polynomials $C(X), A(X)$
- Mapping $u : [m] \to I$

Precomputed inputs:

- $[W_1^{(i)}(x)]_2$ for all $i \in I$ where $W_1^{(i)}(X) = (C(X) - c_i)/(X - \omega^i)$
- $[W_2^{(i)}(x)]_2$ for all $i \in I$ where $W_2^{(i)}(X) = Z_{\mathbb{H}}(X)/(X - \omega^i)$

Round 1 Prover:

- Randomly sample blinding factors $r_1, ..., r_6$
- Compute the Lagrange basis polynomials $\{\tau_i(X)\}_{i \in [m]}$ over ${\omega^j}_{j \in I}$
- Define $Z_I'(X) = r_1 \prod_{i \in I}(X - \omega^i)$
- Define $C_I(X) = \sum_{i \in I} c_i \tau_i(X)$
- Define blinded $C_I'(X) = C_I(X) + (r_2 + r_3 X + r_4 X^2) Z_I'(X)$
- Define $U(X)$ to be the degree $m-1$ interpolation over $\mathbb{V}$ with $U(\nu_i) = \omega^{u(i)}, \forall i \in [m]$
- Define blinded $U'(X) = U(X) + (r_5 + r_6 X) Z_{\mathbb{V}}(X)$
- Publish $\boldsymbol{z_I} = [Z_I'(x)]_1, \boldsymbol{c_I} = [C_I'(x)]_1, \boldsymbol{u} = [U'(x)]_1$

Round 2 Verifier: Send random challenge χ_1, χ_2
Round 2 Prover:

- Compute $[W_1(x) + \chi_2 W_2(x)]_2 = \sum_{i \in I} \frac{[W_1^{(i)}(x)]_2 + \chi_2 [W_2^{(i)}(x)]_2}{\prod_{j \in I, i \neq j} \omega^i - \omega^j}$
- Compute $H(X) = (Z_I'(U'(X)) + \chi_1 (C_I'(U'(X)) - A(X)))/Z_{\mathbb{V}}(X)$
- Publish $\boldsymbol{w} = r_1^{-1}[W_1(x) + \chi_2 W_2(x)]_2 - [r_2 + r_3 x + r_4 x^2]_2$, $\boldsymbol{h} = [H(x)]_1$

Round 3 Verifier: Send random challenge α
Round 3 Prover: Output $v_1, v_2, \pi_1, \pi_2, \pi_3$ where

$$\begin{aligned} P_1(X) &\leftarrow Z'_I(X) + \chi_1 C'_I(X) \\ P_2(X) &\leftarrow Z'_I(U'(\alpha)) + \chi_1(C'_I(U'(\alpha)) - A(X)) - Z_{\mathbb{V}}(\alpha)H(X) \\ (v_1, \pi_1) &\leftarrow \text{KZG.Open}(U'(X), \alpha) \\ (v_2, \pi_2) &\leftarrow \text{KZG.Open}(P_1(X), v_1) \\ (0, \pi_3) &\leftarrow \text{KZG.Open}(P_2(X), \alpha) \end{aligned}$$

Verifer: Compute $\boldsymbol{p_1} = \boldsymbol{z_I} + \chi_1 \boldsymbol{c_I}$ and $\boldsymbol{p_2} = [v_2]_1 - \chi_1 \boldsymbol{a} - Z_{\mathbb{V}}(\alpha)\boldsymbol{h}$ and verify

$$\begin{aligned} 1 &\leftarrow \text{KZG.Verify}(\boldsymbol{u}, \alpha, v_1, \pi_1) \\ 1 &\leftarrow \text{KZG.Verify}(\boldsymbol{p_1}, v_1, v_2, \pi_2) \\ 1 &\leftarrow \text{KZG.Verify}(\boldsymbol{p_2}, \alpha, 0, \pi_3) \\ e((\boldsymbol{C} - \boldsymbol{c_I}) + \chi_2[x^n - 1]_1, [1]_2) &= e(\boldsymbol{z_I}, \boldsymbol{w}) \end{aligned}$$

Prover complexity is $O(m^2)$, with the limiting steps being polynomial interpolations of Z_I and C_I in round 1 and the aggregation of the precomputed KZG witnesses to produce $\boldsymbol{w}$ in round 2. The verifier verifies three KZG commitment openings and one additional pairing check. Notice that we can drop one degree of blinding from $U'(X)$ as compared to Caulk because $U'(X)$ is not opened as a KZG commitment in the subprotocol to show well-formedness. Furthermore, one fewer pairing is required for verification because the degree bound check used in the subprotocol is eliminated. We can use the same standard batching techniques described in section 8 of [ZBK+22] to reduce the number of pairing checks from 4 to 3 and the number of $\mathbb{G}_1$ elements in the proof from 8 to 7.

Theorem 1. *The protocol in Figure 2 is a zero-knowledge argument of knowledge for the relation in equation 1 with verifier complexity $O_\lambda(1)$ and prover complexity $O_\lambda(m^2)$, granted the prover has precomputed KZG witnesses for C and $X^n - 1$ at all indices in I.*

Proof. The proof of knowledge soundness for Theorem 1 is given in Appendix A and the proof of zero-knowledge is given in Appendix B.

Theorem 2. *There exists a vector commitment scheme with position-hiding linkability, verifier complexity $O_\lambda(1)$, and prover complexity $O_\lambda(m^2)$, for m the size of the subset and n the size of the table. The commitment scheme requires a trusted setup and requires the prover to precompute and store a constant number of elements per linked index that take $O(n)$ time to compute each or $O(n \log n)$ time to compute as a batch.*

Proof. The KZG polynomial commitment scheme over evaluation domains which are multiplicative subgroups is a vector commitment scheme per Section 4.6, "KZG as Vector Commitment Scheme", of [ZBK+22]. The protocol in Figure 2 provides position-hiding linkability with the required asymptotic complexity per Theorem 1.

4 Linking to Pedersen Commitments

Section 6 of [ZBK+22] presents a specific argument for linking a Pedersen commitment to an element in a committed vector. In this setting, the SRS contains one additional random element $\boldsymbol{h} \in \mathbb{G}_1$ for which the discrete log relations to all other SRS elements are unknown. Then we can construct a zero-knowledge argument for the witness relation:

$$\mathcal{R}_{\text{PC-Link}} := \left\{ \left(\begin{array}{l} \{[x^{k-1}]_1, [x^{k-1}]_2\}_{k\in[d]}, \boldsymbol{h} \,; \\ \boldsymbol{c}, \boldsymbol{v}, \mathbb{H}, \omega \,; \\ C(X), j, v, r \end{array} \right) \middle| \begin{array}{l} \boldsymbol{c} = [C(x)]_1 \\ \boldsymbol{v} = [v]_1 + r\boldsymbol{h} \\ v = C(\omega^j) \end{array} \right\}.$$

Care must be taken when modifying the argument from section 6 of [ZBK+22] to replace the unity subprotocol with a divisibility check. The divisibility check does not guarantee that Z_I has a root, and if it is a constant polynomial then the main pairing check does not correspond to a blinded KZG opening. In the generalized argument this is not an issue because equation 5 ensures that Z_I has a root.

Instead, we will compose the generalized lookup argument with a generalized Schnorr proof to produce an efficient argument for the single element case. It is well known that the classic Schnorr argument of knowledge of a discrete logarithm can be generalized to more complex group homomorphisms from a scalar field to a prime order group [Sch91]. In this setting we generalize Schnorr's protocol to an argument of knowledge of the shared opening to two Pedersen commitments with different bases in $\mathbb{G}_1$.

The prover samples $k \leftarrow \mathbb{F}$ and computes a polynomial $A(X) = v + k(X-1)$. The commitment to $A(X)$ is $\boldsymbol{a} = v[1]_1 + k[x-1]_1$. The prover and verifier run the lookup argument as a subprotocol with $\boldsymbol{a}$ and $\mathbb{V} = \{1\}$. Finally they engage in a proof of knowledge of v, r, k such that:

$$\begin{aligned} \boldsymbol{v} &= v[1]_1 + r\boldsymbol{h}, \\ \boldsymbol{a} &= v[1]_1 + k[x-1]_1. \end{aligned}$$

Figure 3: Interactive Protocol for $\mathcal{R}_{\text{PC-Link}}$

Public inputs:

- Prime order cyclic groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ with bilinear map e and generators $[1]_1, [1]_2$
- Scalar field $\mathbb{F}$
- Structured reference string $[x]_1, ..., [x^{d-1}]_1, [x]_2, ..., [x^{d-1}]_2$
- Independent $\mathbb{G}_1$ generator $\boldsymbol{h}$

Common inputs:

- Multiplicative subgroup $\mathbb{H} < \mathbb{F}^*$ with order n and generator ω
- KZG commitment $\boldsymbol{C}$ to $C(X)$ with evaluation points in $\mathbb{H}$
- Pedersen commitment $\boldsymbol{v}$

Witness inputs:

- Value v, Pedersen commitment randomness r, index i
- Polynomial $C(X)$

Precomputed inputs:

- $[W_1^{(i)}(x)]_2$ where $W_1^{(i)}(X) = (C(X) - c_i)/(X - \omega^i)$
- $[W_2^{(i)}(x)]_2$ where $W_2^{(i)}(X) = Z_{\mathbb{H}}(X)/(X - \omega^i)$

Round 1 Prover:

- Randomly sample blinding factors $k, \hat{v}, \hat{r}, \hat{k} \leftarrow \mathbb{F}$
- Prover outputs $\boldsymbol{a} = [v]_1 + k[x-1]_1$
- Prover and verifier engage in Link protocol with $\boldsymbol{c}, \boldsymbol{a}, A(X) = v + k(X-1), \mathbb{V} = \{1\}, I = \{i\}$ (Figure 2)

Round 2 Prover: Output $\tilde{\boldsymbol{v}} = [\hat{v}]_1 + \hat{r}\boldsymbol{h}, \tilde{\boldsymbol{a}} = [\hat{v}]_1 + \hat{k}[x-1]_1$

Round 3 Verifier: Sample random χ
Round 3 Prover: Output $s_v = \hat{v} + \chi v, s_r = \hat{r} + \chi r, s_k = \hat{k} + \chi k$

Verifier: Verify that

$$[s_v]_1 + s_r\boldsymbol{h} = \tilde{\boldsymbol{v}} + \chi\boldsymbol{v}$$
$$[s_v]_1 + s_k[x-1]_1 = \tilde{\boldsymbol{a}} + \chi\boldsymbol{a}.$$

5 Acknowledgements

We thank Arantxa Zapico for discussions and clarifications on the original Caulk protocol. We thank Oana Ciobotaru for identifying several mistakes in the presentation, including the statement of the verifier complexity. We thank Michal Zajic, Janno Siim, Helger Lipmaa, and Roberto Parisella for identifying mistakes in the protocol specification which violated correctness and zero-knowledge guarantees.

References

[FKL18] Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 33–62, Cham, 2018. Springer International Publishing.

[GWC19] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. Plonk: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Paper 2019/953, 2019. https://eprint.iacr.org/2019/953.

[KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010*, pages 177–194, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[Sch91] C. P. Schnorr. Efficient signature generation by smart cards. *J. Cryptol.*, 4(3):161–174, jan 1991.

[TAB+20] Alin Tomescu, Ittai Abraham, Vitalik Buterin, Justin Drake, Dankrad Feist, and Dmitry Khovratovich. Aggregatable subvector commitments for stateless cryptocurrencies. In Clemente Galdi and Vladimir Kolesnikov, editors, *Security and Cryptography for Networks*, pages 45–64, Cham, 2020. Springer International Publishing.

[ZBK+22] Arantxa Zapico, Vitalik Buterin, Dmitry Khovratovich, Mary Maller, Anca Nitulescu, and Mark Simkin. Caulk: Lookup arguments in sublinear time. Cryptology ePrint Archive, Paper 2022/621, 2022. https://eprint.iacr.org/2022/621.

A Proof of Theorem 1: Knowledge Soundness

The protocol for position-hiding linking has knowledge soundness in the Algebraic Group Model of [FKL18]. Knowledge soundness is defined by a game Game_{KS} involving an algebraic adversary $\mathcal{A}$ and an efficient extractor $\mathcal{E}$. Given an SRS, the adversary outputs an instance $\boldsymbol{c}, \boldsymbol{a}, \omega, \mu$ and produces an interactive argument for the verifier. The extractor then outputs polynomials C, A. The adversary wins if 1) the verifier accepts and 2) it is not the case that $\boldsymbol{c} = [C(x)]_1 \wedge \boldsymbol{a} = [A(x)]_1 \wedge \forall i \in [m], \exists j \in [n], A(\mu^i) = C(\omega^j)$. The protocol has knowledge soundness if there exists an $\mathcal{E}$ so that no adversary wins the game with greater than negligible probability over the verifier's randomness.

Because the structured reference string consists of the powers of x up to x^{d-1} lifted to $\mathbb{G}_1$ and $\mathbb{G}_2$, an algebraic representation of a group element in either group can be interpreted as the coefficients of a polynomial in $\mathbb{F}_d[X]$. The adversary outputs $\boldsymbol{z_I}, \boldsymbol{c_I}, \boldsymbol{u}, \boldsymbol{h} \in \mathbb{G}_1, \boldsymbol{w} \in \mathbb{G}_2$ along with their representations, and so the extractor learns the corresponding polynomials Z'_I, C'_I, U', W, H. The real pairing check

$$e((\boldsymbol{C} - \boldsymbol{c_I}) + \chi_2[x^n - 1]_1, [1]_2) = e(\boldsymbol{z_I}, \boldsymbol{w}),$$

corresponds to the ideal pairing check

$$C - C'_I + \chi_2(X^n - 1) = Z'_I W.$$

Consequently by Lemma 1, the above polynomial identity holds except with negligible probability. Therefore, $Z'_I \mid C - C'_I + \chi_2(X^n - 1)$. Since χ_2 is sampled after the prover commits to C'_I, Z'_I, except with probability $\frac{1}{|\mathbb{F}|}$ it must be that $Z'_I \mid C - C'_I$ and $Z'_I \mid X^n - 1$. Let I be the set of roots of Z'_I. Since $Z'_I \mid X^n - 1$, it follows that $I \subset \mathbb{H}$, and since $Z'_I \mid C - C'_I$, it follows that $C(y) = C'_I(y)$ for all $y \in I$.

By the knowledge soundness of the KZG polynomial commitment scheme and the Schwartz-Zippel lemma, the following polynomial identity holds except with negligible probability because the evaluation holds at a random point α:

$$Z'_I(U'(X)) + \chi_1(C'_I(U'(X)) - A(X)) = Z_{\mathbb{V}}(X)H(X).$$

Therefore, $Z_{\mathbb{V}}(X) \mid Z'_I(U'(X)) + \chi_1(C'_I(U'(X)) - A(X))$. Since χ_1 is sampled after the prover commits to Z'_I, C'_I, U', except with probability $\frac{1}{|\mathbb{F}|}$ it must be that $Z_{\mathbb{V}} \mid Z'_I(U'(X))$ and $Z_{\mathbb{V}} \mid C'_I(U'(X)) - A(X)$. Then, $Z'_I(U'(y)) = 0$ and $C'_I(U'(y)) = A(y)$ for all $y \in \mathbb{V}$. Furthermore, $U'(y) \in I$ for all $y \in \mathbb{V}$, since I is the set of roots of Z'_I by definition. Now, for any $i \in [m]$, $A(\mu^i) = C'_I(U'(\mu^i))$. There exists a $y \in I$ with $U'(\mu^i) = y$ since U' maps $\mathbb{V}$ to I. For all $y \in I$, $A(\mu^i) = C'_I(y) = C(y)$. Let j be such that $y = \omega^j$, which we know to exist because $I \subset \mathbb{H}$ and ω generates $\mathbb{H}$. Then when the extractor outputs C, A, it holds that for all $\forall i \in [m], \exists j \in [n], A(\mu^i) = C(\omega^j)$, meaning the adversary loses the game.

B Proof of Theorem 1: Zero Knowledge

We describe the $\textbf{Simulate}_{\text{Link}}$ algorithm that, given an instance $\boldsymbol{c}, \boldsymbol{a}$ and the trapdoor value x convinces an interactive verifier to accept. This is similar to the argument in Appendix F of [ZBK+22]. The simulator samples $s_1, ..., s_8 \leftarrow \mathbb{F}$ at random and outputs $\boldsymbol{z_I} = [s_1]_1, \boldsymbol{c_I} = \boldsymbol{c} - [s_2]_1, \boldsymbol{u} = [s_3]_1$. The simulator then receives χ_1, χ_2 and outputs

$\boldsymbol{w} = s_1^{-1}[s_2 + \chi_2 Z_{\mathbb{H}}(x)]_2, \boldsymbol{h} = [s_4]_2$. As in Appendix F, the simulator receives α, outputs $v_1 = s_5, v_2 = s_6$, and computes KZG evaluation proofs:

$$\begin{aligned}
\pi_1 &= (x - \alpha)^{-1}(\boldsymbol{u} - [v_1]_1), \\
\pi_2 &= (x - v_1)^{-1}(\boldsymbol{z_I} + \chi_1 \boldsymbol{c_I} - [v_2]_1), \\
\pi_3 &= (x - \alpha)^{-1}([v_2]_1 - \chi_1 \boldsymbol{a} - Z_{\mathbb{V}}(\alpha)\boldsymbol{h}).
\end{aligned}$$

It can be seen that the simulator's outputs satisfy the pairing check.

$$\begin{aligned}
& e((\boldsymbol{C} - \boldsymbol{c_I}) + \chi_2[x^n - 1]_1, [1]_2) \\
=& e([s_2]_1 + \chi_2[Z_{\mathbb{H}}(x)]_1, [1]_2) \\
=& e([1]_1, [s_2]_2 + \chi_2[Z_{\mathbb{H}}(x)]_2) \\
=& e([s_1]_1, s_1^{-1}([s_2]_2 + \chi_2[Z_{\mathbb{H}}(x)]_2)) \\
=& e(\boldsymbol{z_I}, \boldsymbol{w})
\end{aligned}$$

We note the distribution of output elements matches a valid distribution because:

- $\boldsymbol{z_I}$ is blinded by r_1 for the prover and s_1 for the simulator,
- $\boldsymbol{c_I}$ is blinded by r_2 for the prover and s_2 for the simulator,
- $\boldsymbol{u}$ is blinded by r_5 for the prover and s_3 for the simulator,
- $\boldsymbol{w}$ uniquely satisfies the pairing check,
- $\boldsymbol{h}$ is blinded by r_3 for the prover and s_4 for the simulator,
- v_1 is blinded by r_6 for the prover and s_5 for the simulator,
- v_2 is blinded by r_4 for the prover and s_6 for the simulator,
- π_1, π_2, π_3 uniquely satisfy the KZG openings.