Batching non-membership proofs and proving non-repetition with bilinear accumulators

Abstract

In this short paper, we provide a protocol to batch multiple non-membership proofs into a single proof of constant size with bilinear accumulators via a succinct argument of knowledge for polynomial commitments.

We use similar techniques to provide a constant-sized proof that a polynomial commitment as in [KZG10] is a commitment to a separable (square-free) polynomial. In the context of the bilinear accumulator, this can be used to prove that a committed multiset is, in fact, a set. This has applications to any setting where a Verifier needs to be convinced that no element was added more than once. This protocol easily generalizes to a succinct protocol that shows that no element was inserted more than k times.

We use the protocol for the derivative to link a committed polynomial to a commitment to its degree in zero-knowledge.

We have designed all of the protocols so that the Verifier needs to store just four elliptic curve points for any verification, despite the linear CRS. We also provide ways to speed up the verification of membership and non-membership proofs and to shift most of the computational burden from the Verifier to the Prover. Since all the challenges are public coin, the protocols can be made non-interactive with a Fiat-Shamir heuristic.

1 Introduction

A commitment scheme is a fundamental cryptographic primitive which is the digital analog of a sealed envelop. *Committing* to a message `m` is akin to putting `m` in the envelop. *Opening* the commitment is like opening the envelop and revealing the content within. Commitments are endowed with two basic properties. The *hiding* property entails that a commitment reveals no information about the underlying message. The *binding* property ensures that one cannot alter the message without altering the commitment.

A cryptographic *accumulator* is a succinct binding commitment to a set or a multiset. A Prover with access to the set/multiset can prove membership or non-membership of an element with a proof publicly verifiable against the succinct commitment held by a Verifier. Accumulators have been used for many applications including accountable certificate management [BLL00, NN98], timestamping [Bd94], group signatures and anonymous credentials [CL02], computations on authenticated data [ABC+12], anonymous e-cash [STS99b, MGGR13a], privacy-preserving data outsourcing [Sla12], updatable signatures [PS14, CJ10], and decentralized bulletin boards [FVY14, GGM14].

In this paper, we study a class of accumulators that is based on bilinear pairings of elliptic curves. First introduced by Nguyen in [Ngu05], these accumulators have the major advantage over the better known accumulator of a Merkle tree in that membership proofs are of constant size and multiple membership proofs can be batched together into a single constant-sized proof. Furthermore, it was shown in [DT08] that they also allow for non-membership proofs for elements outside the committed set. In this paper, we provide a protocol to prove non-membership of an arbitrarily large set with a constant-sized proof.

We use techniques similar to those used in our batched non-membership proof protocol to provide a protocol to succinctly demonstrate that a polynomial commitment is to a separable (square-free) polynomial. This can be used to prove - with a constant-sized proof - that no element was inserted more than once into the bilinear accumulator. It easily generalizes to a protocol to show that no element was inserted more than k times. This is not possible with a Merkle tree or (as far as we know) with the accumulator based on a hidden order group.

Since bilinear accumulators require groups far smaller than RSA groups for the same level of security, we expect them to be substantially faster than RSA accumulators when it comes to accumulation, generation of membership proofs (witnesses) and verification. This is borne out by the implementation in [Tre13] (even though it precedes the number field sieve attack). Furthermore, the hardness assumptions that underpin bilinear accumulators are the same as those in pairing-based Snarks and are *arguably* less brittle than the hardness assumptions for hidden order groups.

Furthermore, we adapt techniques from [BBF19] and [Wes18] in the bilinear accumulator setting to speed up verifications of membership/non-membership proofs and to shift most of the computational and storage burdens from the Verifier to the Prover. In particular, we provide a protocol to reduce the Verifier's task of verifying membership proofs to a constant run time independent of the number of data elements to be batched.

1.1 Structure/contributions of the paper

In section 1, we primarily provide some background and notations for bilinear accumulators and the KZG polynomial commitment scheme, including the hardness assumptions that underpin these schemes. In section 2, we describe the protocol `PoE` for verifiable computation and the succinct argument of knowledge `PoKE` along with the security proofs.

In section 3, we use the protocols from section 2 to provide a constant-sized non-membership proof for an arbitrarily large set with respect to the accumulated digest. Such a batched proof is not possible via the a Merkle tree. While accumulators based on hidden order groups famously do support batched non-membership proofs ([BBF19]), the groups are substantially larger and the proof generation times are consequently longer.

In section 4, we use the protocols from section 2 to construct a protocol that succinctly demonstrates that a KZG polynomial commitment is a commitment to a separable (square-free) polynomial. In the context of the bilinear accumulator, this can be used to prove - with a constant-sized proof - that no element was inserted more than once into the accumulator. This easily generalizes to a protocol thaat succinctly demonstrates that no element was inserted more than k times. As far as we know, this is not possible with a Merkle tree or an accumulator based on hidden order groups.

We also discuss a protocol to demonstrate a polynomial relation between two discrete logarithms in section 4. This protocol can be combined with the protocol for separable polynomial commitments to derive a protocol that succinctly demonstrates that every element element inserted into the accumulator was inserted with frequency between m and n for public integers $m \leq n$.

In the appendix, we describe a vector commitment with constant-sized openings that hinges on the universal accumulator with constant-sized membership and non-membership proofs.

1.2 Notations and terminology

As usual, $\mathbb{F}_q$ denotes the finite field with q elements for a prime power q and $\overline{\mathbb{F}}_q$ denotes its algebraic closure. $\mathbb{F}_q^*$ denotes the cyclic multiplicative group of the non-zero elements of $\mathbb{F}_q$. For

polynomials $f(X)$, $g(X) \in \mathbb{F}_q[X]$, we denote by $\mathbf{gcd}(f(X), g(X))$ the unique *monic* polynomial that generates the (principal) ideal of $\mathbb{F}_q[X]$ generated by $f(X)$ and $g(X)$.

A polynomial in $\mathbb{F}_q[X]$ is said to be *separable* or *square-free* if it is not divisible in $\mathbb{F}_q[X]$ by the square of any irreducible polynomial. Since a finite field is a perfect field, $f(X)$ being separable in $\mathbb{F}_q[X]$ is equivalent to $f(X)$ being separable in $\overline{\mathbb{F}}_q[X]$, i.e. $f(X)$ having no zeros with multiplicity ≥ 2 in $\overline{\mathbb{F}}_q[X]$. A well-known fact is that a polynomial $f(X)$ being separable is equivalent to it being relatively prime with its derivative $f'(X)$.
We now briefly introduce pairings.

Definition 1.1. For abelian groups $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, a *pairing*

$$\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$$

is a map equipped with the following properties.
1. Bilinearity: $\mathbf{e}(x_1 + x_2, y_1 + y_2) = \mathbf{e}(x_1, y_2) \cdot \mathbf{e}(x_1, y_2) \cdot \mathbf{e}(x_2, y_1) \cdot \mathbf{e}(x_2, y_2)$
$\forall\ x_1, x_2 \in \mathbb{G}_1,\ y_1, y_2 \in \mathbb{G}_2$.
2. Non-degeneracy: The image of $\mathbf{e}$ is non-trivial.
3. Efficient computability.

In pairing-based cryptography, we typically work in settings where the groups $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ are cyclic of order p for some 256-bit prime p so as to have a 128-bit security level. Such pairings $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ are classified into three types:
- Type I: $\mathbb{G}_1 = \mathbb{G}_2$.
- Type II: $\mathbb{G}_1 \neq \mathbb{G}_2$ but there is an *efficiently computable* isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$.
- Type III: There is no *efficiently computable* isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$.

1.3 Cryptographic assumptions

We state the computationally infeasible problems that the security of our constructions hinge on.

Assumption 1.1. **n-strong Diffie Hellman assumption:** *Let $\mathbb{G}$ be a cyclic group of prime order p generated by an element g, and let $s \in \mathbb{F}_p^*$. Any probabilistic polynomial-time algorithm that is given the set $\{g^{s^i} : 1 \leq i \leq n\}$ can output a pair $(a, g^{1/(s+a)}) \in \mathbb{F}_p^* \times \mathbb{G}$ with at most negligible probability.*

Assumption 1.2. Knowledge of exponent assumption (KEA): *Let $\mathbb{G}$ be a cyclic group of prime order p generated by an element g, and let $s \in \mathbb{F}_p^*$. Suppose there exists a PPT algorithm $\mathcal{A}_1$ that given pairs $(h_1, h_1^s), \cdots, (h_n, h_n^s)$ in $\mathbb{G}^2$, outputs a pair $(c_1, c_2) \in \mathbb{G}^2$ such that $c_2 = c_1^s$. Then there exists a PPT algorithm $\mathcal{A}_2$ that, with overwhelming probability, outputs a vector $(x_1, \cdots, x_n) \in \mathbb{F}_p^n$ such that*

$$c_1 = \prod_{i=1}^{n} h_i^{x_i} \quad , \quad c_2 = \prod_{i=1}^{n} (h_i^s)^{x_i}$$

A special case of the KEA assumption is that given the elements $\{g^{s^i} : 0 \leq i \leq n\}$, if a PPT algorithm $\mathcal{A}_1$ is able to output a triplet $(c_1, c_2, f(X)) \in \mathbb{G} \times \mathbb{G} \times \mathbb{F}_p[X]$ with $\deg(f(X) \geq 1$ such that $c_2 = c_1^{f(s)}$, then there is a PPT algorithm $\mathcal{A}_2$ that with overwhelming probability, outputs a polynomial $e(X)$ such that

$$c_1 = g_1^{e(s)}\ ,\ c_2 = g_1^{e(s) \cdot f(s)}.$$

The KEA assumption implies that breaking the strong Diffie-Hellman is equivalent to computing the trapdoor s. A PPT adversary $\mathcal{A}$ that can compute an element w such that $w^{s+\alpha} = g_1$ can also generate a polynomial $e(X)$ such that

$$w = g_1^{e(s)} \quad , \quad g_1 = g_1^{e(s)\cdot s}$$

and hence, s is a zero of the polynomial $e(X) \cdot X - 1$. So $\mathcal{A}$ can use a PPT algorithm such as [KS98] to factorize $e(X) \cdot X - 1$ and extract s in expected polynomial time.

Assumption 1.3. *Let $\mathbb{G}$ be a cyclic group of prime order p generated by an element g, and let $s \in \mathbb{F}_p^*$. Any probabilistic polynomial-time algorithm that is given the set $\{g^{s^i} : 1 \leq i \leq n\}$ can output a pair $(f(X), w) \in \mathbb{F}_p[X]_{\deg \geq 1} \times \mathbb{G}$ such that*

$$w^{f(s)} = g$$

with at most negligible probability.

This assumption is stronger than the n-strong Diffie Hellman assumption. However, for cryptosystems that use the KEA assumption, they are equivalent.

Lemma 1.1. *The n-strong Diffie Hellman and KEA assumptions imply Assumption 1.3.*

Proof. We show that a PPT adversary $\mathcal{A}$ that breaks Assumption 1.3 can break the n-strong Diffie Hellman with overwhelming probability. Suppose $\mathcal{A}$ outputs a non-constant polynomial $f(X)$ of degree $k \geq 1$ and an element w such that $w^{f(s)} = g$. Write $f(X) = \sum_{i=0}^{k} c_i X^i$. Then

$$(w^{\sum_{i=1}^{k} c_i s^{i-1}})^s = g \cdot w^{-c_0}$$

and the KEA assumption implies that with overwhelming probability, $\mathcal{A}$ can output a polynomial $e(X)$ such that

$$w^{\sum_{i=1}^{k} c_i s^{i-1}} = g^{e(s)} \quad , \quad g \cdot w^{-c_0} = g^{e(s)\cdot s}.$$

So

$$g = \left(g^{(1-e(s)\cdot s)\cdot c_0^{-1}}\right)^{f(s)}$$

and hence, $f(s)\cdot(e(s)\cdot s - 1) = c_0$. Now, $\mathcal{A}$ can use the [KS98] algorithm to factorize the polynomial $f(X) \cdot (e(X) \cdot X - 1) - c_0$ in expected polynomial runtime. Hence, $\mathcal{A}$ can extract the integer s - thus breaking the n-strong Diffie Hellman assumption - with overwhelming probability. □

1.4 Argument Systems

An argument system for a relation $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{W}$ is a triple of randomized polynomial time algorithms $(\texttt{PGen}, \mathcal{P}, \mathcal{V})$, where $\texttt{PGen}$ takes an (implicit) security parameter λ and outputs a common reference string (CRS) $\texttt{pp}$. If the setup algorithm uses only public randomness we say that the setup is transparent and that the CRS is unstructured. The prover $\mathcal{P}$ takes as input a statement $x \in X$, a witness $w \in W$, and the CRS $\texttt{pp}$. The verifier $\mathcal{V}$ takes as input $\texttt{pp}$ and x and after interactions with $\mathcal{P}$ outputs 0 or 1. We denote the transcript between the prover and the verifier by $\langle \mathcal{V}(\texttt{pp}, x), \mathcal{P}(\texttt{pp}, x, w) \rangle$ and write $\mathcal{V}\langle (\texttt{pp}, x), \mathcal{P}(\texttt{pp}, x, w) \rangle = 1$ to indicate that the verifier accepted the transcript. If $\mathcal{V}$ uses only public randomness we say that the protocol is *public coin*.

We now define soundness and knowledge extraction for our protocols. The adversary is modeled as two algorithms $\mathcal{A}_0$ and $\mathcal{A}_1$, where $\mathcal{A}_0$ outputs the instance $x \in \mathcal{X}$ after $\texttt{PGen}$ is run, and $\mathcal{A}_1$ runs the interactive protocol with the verifier using a state output by $\mathcal{A}_0$. In a

slight deviation from the soundness definition used in statistically sound proof systems, we do not universally quantify over the instance x (i.e. we do not require security to hold for all input instances x). This is due to the fact that in the computationally-sound setting the instance itself may encode a trapdoor of the common reference string, which can enable the adversary to fool a verifier. Requiring that an efficient adversary outputs the instance x prevents this. In our soundness definition the adversary $\mathcal{A}_1$ succeeds if he can make the verifier accept when no witness for x exists.

Definition 1.2. *We say an argument system* $(\mathtt{PGen}, \mathcal{P}, \mathcal{V})$ *for a relation* $\mathcal{R}$ *is* ***complete*** *if for all* $(x, w) \in \mathcal{R}$*,*

$$\mathtt{Pr}\big[\big\langle \mathcal{V}(\mathtt{pp}, x)\ ,\ \mathcal{P}(\mathtt{pp}, w)\big\rangle\big) = 1\ :\ \mathtt{pp} \xleftarrow{\$} \mathtt{PGen}(\lambda)\big]\ =\ 1.$$

Definition 1.3. *We say an argument system* $(\mathtt{PGen}, \mathcal{P}, \mathcal{V})$ *is* ***sound*** *if* $\mathcal{P}$ *cannot forge a fake proof except with negligible probability.*

Definition 1.4. *We say a sound argument system is an* ***argument of knowledge*** *if for any polynomial time adversary* $\mathcal{A}$*, there exists an extractor* $\mathcal{E}$ *with access to* $\mathcal{A}$*'s internal state that can, with overwhelming probability, extract a valid witness whenever* $\mathcal{A}$ *is convincing.*

Definition 1.5. *An argument system is* ***non-interactive*** *if it consists of a single round of interaction between* $\mathcal{P}$ *and* $\mathcal{V}$*.*

The Fiat-Shamir heuristic ([FS87]) can be used to transform interactive public coin argument systems into non-interactive systems. Instead of the Verifier generating the challenges, this function is performed by a public hashing algorithm agreed upon in advance.

1.5 Bilinear accumulators

We describe the setup in this section. Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be cyclic groups of order p for some prime p such that there exists a pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ which is *bilinear*, *non-degenerate* and *efficiently computable*. Fix generators g_1, g_2 of the cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ respectively. Then $\mathbf{e}(g_1, g_2)$ is a generator of $\mathbb{G}_T$. For a trapdoor $s \in \mathbb{F}_p^*$, the common reference string (CRS) is given by

$$[g_1, g_1^s, \cdots, g_1^{s^n}]\ \ ,\ \ [g_2, g_2^s, \cdots, g_2^{s^n}]$$

The generation of the CRS requires a trusted setup, which can be partially mitigated by using a secure multi-party computation. For a data multiset $\mathcal{M}$, we define the accumulated digest

$$\mathrm{Acc}(g_1, \mathcal{D}) := g_1^{\prod\limits_{m \in \mathcal{M}} (s+m)^{\mathrm{mult}(m, \mathcal{M})}} \in \mathbb{G}_1.$$

Thus, this is the [KZG10] commitment to the polynomial

$$f_{\mathcal{M}}(X) := \prod_{m \in \mathcal{M}} (X + m)^{\mathrm{mult}(m, \mathcal{M})}.$$

For a multiset $\mathcal{M}_0 \subseteq \mathcal{M}$, the membership witness for $\mathcal{M}_0$ is defined by

$$\mathrm{wit}(\mathcal{M}_0) := g_1^{f_{\mathcal{M}}(s) \big/ f_{\mathcal{M}_0}(s)}\ \in\ \mathbb{G}_1.$$

The Verifier then verifies the equation

$$\mathrm{wit}(\mathcal{M}_0)^{f_{\mathcal{M}_0}(s)} = \mathrm{Acc}(g_1,\ \mathcal{D})$$

via the pairing check

$$\mathbf{e}\Big(\mathrm{wit}(\mathcal{M}_0)\ ,\ g_2^{f_{\mathcal{M}_0}(s)}\Big) = \mathbf{e}(\mathrm{Acc}(g_1,\ \mathcal{M})\ ,\ g_2).$$

2 Verifiable computation for exponentiations

In this section, we provide the protocols PoE and PoKE for bilinear accumulators which achieve three goals.

1. They speed up the verification process by replacing some exponentiation operations by polynomial division in $\mathbb{F}_p[X]$ which is substantially cheaper.

2. They shift most of the computational burden from the Verifier to the Prover. This is useful in settings where the Prover has more computational power at his disposal.

3. They reduce the Verifier's storage burden to the set $\{g_1, g_1^s, g_2, g_2^s\}$. This is potentially useful in settings where the Verifier has a low storage capacity.

Protocol 2.1. *Proof of exponentiation with base g_1 ($\texttt{PoE}^*$):*
Parameters : A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$
Inputs: $a \in \mathbb{G}_1$; a polynomial $f(X) \in \mathbb{F}_p[X]$ of degree $\leq n$
Claim: $g_1^{f(s)} = a$

1. The Fiat-Shamir heuristic generates a challenge $\alpha \in \mathbb{F}_p^*$ (the challenge).

2. The Prover computes a polynomial $h(X) \in \mathbb{F}_p[X]$ and an element $\beta \in \mathbb{F}_p^*$ such that

$$f(X) = (X + \alpha) \cdot h(X) + \beta$$

and sends $Q := g_1^{h(s)}$ to the Verifier.

3. The Verifier computes $\beta := f(X) \pmod{(X + \alpha)}$ and accepts if and only if the equation

$$\mathbf{e}(Q\ ,\ g_2^{s+\alpha}) \cdot \mathbf{e}(g_1^{\beta}\ ,\ g_2) \stackrel{?}{=} \mathbf{e}(a\ ,\ g_2)$$

holds. □

We refer to this as $\texttt{PoE}^*[g_1,\ f(X),\ a]$. The proof consists of a single element of $\mathbb{G}_1$ and in particular, is of constant size. Note that because of the bilinearity of the pairing, we have

$$Q^{s+\alpha} \cdot g_1^{\beta} = a \Longleftrightarrow \mathbf{e}(Q\ ,\ g_2^{s+\alpha}) \cdot \mathbf{e}(g_1\ ,\ g_2^{\beta}) = \mathbf{e}(a\ ,\ g_2).$$

The *asymptotic* complexity of the Verifier remains unchanged since computing the $\mathbb{F}_p$-element $\beta := f(X) \pmod{(X + \alpha)}$ has a runtime of $\mathbf{O}(\deg(f))$ unless the polynomial $f(X)$ is sparse. But this protocol swaps exponentiation operations in the group $\mathbb{G}_1$ with polynomial division operations in $\mathbb{F}_p[X]$ which are substantially cheaper. The most obvious application is that a Prover can use the protocol $\texttt{PoE}^*$ to convince a Verifier that an element $A \in \mathbb{G}_1$ is the accumulated digest $\mathrm{Acc}(g_1,\ \mathcal{D})$ of a data set $\mathcal{D}$. The Verifier just needs the four points $\{g_1,\ g_1^s,\ g_2,\ g_2^s\}$ to check the veracity of this claim.

Clearly, the protocol can be modified for the proof of an exponentiation $g_2^{f(s)} = b$ in the group $\mathbb{G}_2$. In this case, the proof would consist of the $\mathbb{G}_2$ element $g_2^{h(s)}$. We refer to this as $\texttt{PoE}^*[g_2,\ f(X),\ b]$.

Proposition 2.2. *The protocol $\texttt{PoE}^*$ is sound in the algebraic group model.*

Proof. We consider the case where the exponentiation is in $\mathbb{G}_1$ and with base g_1. The case where the exponentiation is in $\mathbb{G}_2$ and with base g_2 is virtually identical.

Suppose a PPT adversary $\mathcal{A}$ is able to output an accepting transcript $Q \in \mathbb{G}_1$ such that

$$\mathbf{e}(Q\ ,\ g_2^{s+\alpha}) \cdot \mathbf{e}(g_1^{\beta}\ ,\ g_2) = \mathbf{e}(a\ ,\ g_2)\ \ ,\ \ \beta := f(X) \pmod{(X+\alpha)}$$

in response to a challenge α. The pairing check implies that $Q^{s+\alpha} \cdot g_1^{\beta} = a$. The KEA assumption implies that with overwhelming probability, $\mathcal{A}$ can output a polynomial $h(X)$ such that

$$Q = g_1^{h(s)}\ \ ,\ \ a = g_1^{h(s)\cdot(s+\alpha)+\beta}.$$

Setting $e(X) := h(X) \cdot (X+\alpha) + \beta$ yields $g_1^{e(s)} = a$. Now,

$$e(X) \equiv \beta \equiv f(X) \pmod{(X+\alpha)}$$

and since α was randomly and uniformly sampled from $\mathbb{F}_p$, it follows that with overwhelming probability, $e(X) = f(X)$. □

We now generalize this to bases $a \in \mathbb{G}_1$ other than g_1. We provide two versions. The second is more efficient (for the Prover) if the Prover knows a polynomial $e(X)$ of a small degree such that $g_1^{e(s)} = a$. The first is more efficient in all other cases.

Protocol 2.3. *Proof of exponent 1 for pairings* ($\texttt{PoE}-\texttt{1}$)*:*
Inputs: $a, b \in \mathbb{G}_1$; a polynomial $f(X) \in \mathbb{F}_p[X]$ of degree $\leq n$
Claim: $a^{f(s)} = b$

1. The Prover $\mathcal{P}$ sends the element $\widetilde{g}_2 := g_2^{f(s)} \in \mathbb{G}_2$ to the Verifier $\mathcal{V}$.
2. $\mathcal{P}$ sends a non-interactive proof for $\texttt{PoE}^*[g_2,\ f(X),\ \widetilde{g}_2]$ to $\mathcal{V}$.
3. $\mathcal{V}$ verifies the proof for $\texttt{PoE}^*[g_2,\ f(X),\ \widetilde{g}_2]$ and the pairing

$$\mathbf{e}(a\ ,\ g_2) \stackrel{?}{=} \mathbf{e}(b\ ,\ \widetilde{g}_2).$$

$\mathcal{V}$ accepts if and only if the pairing check holds and the $\texttt{PoE}^*$ is valid.

Proposition 2.4. *The protocol* $\texttt{PoE}-\texttt{1}$ *is sound in the algebraic group model.*

Proof. Suppose a PPT adversary $\mathcal{A}$ is able to output an element $\widetilde{g}_2 \in \mathbb{G}_2$ such that $\mathbf{e}(a\ ,\ \widetilde{g}_2) = \mathbf{e}(b\ ,\ \ g_2)$ along with a proof for $\texttt{PoE}^*[g_2,\ f(X),\ \widetilde{g}_2]$. The $\texttt{PoE}^*$ implies that with overwhelming probability, $g_2^{f(s)} = \widetilde{g}_2$. The pairing check then implies that the discrete logarithms between the pairs $(g_2,\ \widetilde{g}_2) \in \mathbb{G}_2^2$ and $(a,\ b) \in \mathbb{G}_1^2$ coincide and hence, $a^{f(s)} = b$. □

When the pairing is type III, the exponentiations in $\mathbb{G}_2$ are substantially more expensive than those in $\mathbb{G}_1$. Thus, in cases where the Prover knows a polynomial $e(X)$ of a small degree such that $a = g_1^{e(s)}$, it can be cheaper to compute the element $a^{h(s)} = g_1^{e(s)\cdot h(s)}$ instead of $g_2^{h(s)}$.

Protocol 2.5. *Proof of exponent 2 for pairings* ($\texttt{PoE}-\texttt{2}$)*:*
Parameters : A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ of groups of prime order p; generators g_1, g_2 of $\mathbb{G}_1$, $\mathbb{G}_2$ respectively;
Inputs: $a, b \in \mathbb{G}_1$; a polynomial $f(X) \in \mathbb{F}_p[X]$ of degree $\leq n$
Claim: $a^{f(s)} = b$

1. The Fiat-Shamir heuristic generates a challenge $\alpha \in \mathbb{F}_p^*$ (the challenge).
2. The Prover computes a polynomial $h(X) \in \mathbb{F}_p[X]$ and an element $\beta \in \mathbb{F}_p^*$ such that
$$f(X) = (X + \alpha) \cdot h(X) + \beta$$
and sends $Q := a^{h(s)}$ to the Verifier.
3. The Verifier computes $\beta := f(X) \pmod{(X + \alpha)}$ and accepts if and only if the equation
$$\mathbf{e}(Q\ ,\ g_2^{s+\alpha}) \cdot \mathbf{e}(a\ ,\ g_2^{\beta}) \stackrel{?}{=} \mathbf{e}(b\ ,\ g_2)$$
holds. □

We refer to these protocol as $\texttt{PoE} - \texttt{1}[a\ , f(X),\ b]$ and $\texttt{PoE} - \texttt{1}[a\ , f(X),\ b]$ respectively. We use the notation $\texttt{PoE}[a\ , f(X),\ b]$ to mean one of these two versions.

Proposition 2.6. *The protocol* $\texttt{PoE} - \texttt{2}$ *for bilinear accumulators is sound in the algebraic group model.*

Proof. Suppose, by way of contradiction, that a PPT adversary $\mathcal{A}$ produces fake witnesses Q_1, Q_2 in response to challenges α_1, α_2. Then Q_i satisfies the equation
$$Q_i^{s+\alpha} = b \cdot a^{-\beta_i}\ \ ,\ \ \beta_i := f(X) \pmod{(X + \alpha_i)} \quad (i = 1, 2).$$
The KEA assumption implies that with overwhelming probability, $\mathcal{A}$ can output polynomials $e_1(X)$, $e_2(X)$ such that
$$Q_i = g_1^{e_i(s)}\ \ ,\ \ b \cdot a^{-\beta_i} = g_1^{e_i(s)\cdot(s+\alpha_i)}\ \ ,\ \ \beta_i \equiv f(X) \pmod{(X + \alpha_i)}.$$
Writing
$$f_1(X) := (\beta_2 - \beta_1)^{-1} \cdot [e_1(X) \cdot (X + \alpha_1) - e_2(X) \cdot (X + \alpha_2)]$$
$$f_2(X) := e_1(X) \cdot (X + \alpha_1) + \beta_1 \cdot f_1(X)$$
for brevity yields $a = g_1^{f_1(s)}$, $b = g_1^{f_2(s)}$. Furthermore, the equation
$$g_1^{f_2(s)} = b = Q_1^{s+\alpha_1} \cdot a^{\beta_1} = Q_1^{s+\alpha_1} \cdot g_1^{\beta_1 \cdot f_1(s)}$$
and the strong Diffie Hellman assumption imply that with overwhelming probability,
$$f_2(X) \equiv f_1(X) \cdot \beta_1 \equiv f_1(X) \cdot f(X) \pmod{(X + \alpha_1)}.$$
Since α_1 was randomly and uniformly sampled from $\mathbb{F}_p^*$, it follows that with overwhelming probability, $f_2(X) = f_1(X) \cdot f(X)$. □

We use the protocol PoE to modify the proof of membership for a data set. The goal is to reduce the storage and computational burdens of the Verifier.

Protocol 2.7. *Protocol for set/multiset membership.*
Parameters : A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ of groups of prime order p;
Inputs: Data multisets $\mathcal{D}$, $\mathcal{D}_0$; the accumulated digest $\mathrm{Acc}(g_1,\ D)$
Claim: $\mathcal{D}_0 \subseteq \mathcal{D}$.
1. The Prover computes the polynomial $f_{\mathcal{D}_0}(X) := \prod_{d_0 \in \mathcal{D}_0} (X + d_0)$.
2. The Prover computes

$$\text{wit}(\mathcal{D}_0) := g_1^{f_{\mathcal{D}}(s)/f_{\mathcal{D}_0}(s)} \in \mathbb{G}_1$$

and sends it to the Verifier $\mathcal{V}$.

3. The Prover sends a proof for $\texttt{PoE}[\text{wit}(\mathcal{D}_0),\ f_{\mathcal{D}_0}(X),\ \text{Acc}(g_1, \mathcal{D})]$.
4. The Verifier computes $f_{\mathcal{D}_0}(X)$ and accepts if and only if the PoE is valid. □

3 Arguments of knowledge of the exponents

We next show how the protocol PoE can be adapted to provide an argument of knowledge of the logarithm. The goal is to construct a protocol with communication complexity much lower than simply sending the polynomial to the Verifier. This will be the key ingredient for batching non-memberships with a constant-sized proof.

Protocol 3.1. *Proof of knowledge of the exponent with base g_1* ($\texttt{PoKE}^*$):
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$; generators g_1, g_2 for $\mathbb{G}_1, \mathbb{G}_2$ respectively.
Inputs: Element $a \in \mathbb{G}_1$
Claim: The Prover knows a polynomial $f(X) \in \mathbb{F}_p[X]$ such that $g_1^{f(s)} = a$.

1. The Fiat-Shamir heuristic generates a challenge $\alpha \in \mathbb{F}_p^*$.
2. $\mathcal{P}$ computes the polynomial $h(X) \in \mathbb{F}_p[X]$ and the element $\beta \in \mathbb{F}_p$ such that

$$f(X) = (X + \alpha) \cdot h(X) + \beta.$$

$\mathcal{P}$ computes $Q := g_1^{h(s)}$ and sends $(Q\ ,\ \beta) \in \mathbb{G}_1 \times \mathbb{F}_p$ to $\mathcal{V}$.

3. $\mathcal{V}$ verifies the equations

$$\mathbf{e}(Q\ ,\ g_2^{s+\alpha}) \cdot \mathbf{e}(g_1^{\beta}\ ,\ g_2) \stackrel{?}{=} \mathbf{e}(a\ ,\ g_2)$$

and accepts if and only if the equation holds. □

The proof consists of an element of $\mathbb{G}_1$ and an element of $\mathbb{F}_p$. We refer to this as $\texttt{PoKE}^*[g_1,\ a]$. We note that multiple $\texttt{PoKE}^*$s can be batched together. For elements $a_1, \cdots, a_k \in \mathbb{G}_1$, a Prover can demonstrate knowledge of polynomials $f_i(X)$ such that $g_1^{f_i(s)} = a_i$ by sending a proof for $\texttt{PoKE}^*[g_1,\ \prod_{i=1}^{k} a_i^{\gamma^i}]$ in response to a randomly generated challenge $\gamma \in \mathbb{F}_p$. This proof is constant-sized and independent of the number of polynomials or their degrees.

Clearly, the protocol can be modified for the proof of the knowledge of an exponent $g_2^{f(s)} = b$ in $\mathbb{G}_2$. In this case, the proof would consist of an element of $\mathbb{G}_2$ and an element of $\mathbb{F}_p$. We refer to this as $\texttt{PoKE}^*[g_2,\ b]$.

Proposition 3.2. *The protocol* $\texttt{PoKE}^*$ *is an argument of knowledge in the algebraic group model.*

Proof. We address the case where the exponentiation is in $\mathbb{G}_1$ and with base g_1. The case where the exponentiation is in $\mathbb{G}_2$ and with base g_2 is identical. We first show that the protocol is sound and then demonstrate witness extractability.

Suppose a PPT adversary $\mathcal{A}$ is able to output an accepting transcript $(Q\ ,\ \beta) \in \mathbb{G}_1 \times \mathbb{F}_p$ such that $\mathbf{e}(Q\ ,\ g_2^{s+\alpha}) \cdot \mathbf{e}(g_1^{\beta}\ ,\ g_2) = \mathbf{e}(a\ ,\ g_2)$ in response to a challenge α. The pairing check implies that $Q^{s+\alpha} \cdot g_1^{\beta} = a$. The KEA assumption implies that with overwhelming probability, $\mathcal{A}$ can output a polynomial $h(X)$ such that

$$Q = g_1^{h(s)} \ , \ a = g_1^{h(s)\cdot(s+\alpha)+\beta}.$$

Setting $f(X) := h(X) \cdot (X + \alpha) + \beta$ yields $g_1^{f(s)} = a$, which completes the proof.

We now demonstrate witness extractability to show that this is an argument of knowledge. An extractor $\mathcal{E}$ with access to the accepting transcripts and to the CRS proceeds as follows. Given accepting transcripts (Q_i, β_i) for challenges α_i $(i = 1, \cdots, N)$, $\mathcal{E}$ uses the Chinese remainder theorem to compute a polynomial $e_N(X)$ such that

$$e_N(X) \equiv \beta \ (\text{mod } (X + \alpha_i)) \ , \ i = 1, \cdots, N.$$

If $g_1^{e_N(s)} = a$, $\mathcal{E}$ halts. Otherwise, $\mathcal{E}$ samples the next accepting transcript (Q_{N+1}, β_{N+1}) and computes the polynomial $e_{N+1}(X)$ such that

$$e_{N+1}(X) \equiv e_N(X) \ (\text{mod } \prod_{i=1}^{N}(X + \alpha_i)) \ , \ e_{N+1}(X) \equiv \beta_{N+1} \ (\text{mod } (X + \alpha_{N+1})).$$

via the Chinese remainder theorem. When the number of accepting transcripts sampled exceeds the degree of $f(X)$, the polynomial obtained by $\mathcal{E}$ is $f(X)$ with overwhelming probability. □

We now generalize this to bases $a \in \mathbb{G}_1$ other than g_1. We provide two versions. The second is more efficient (for the Prover) if the Prover knows a polynomial $e(X)$ of a small degree such that $g_1^{e(s)} = a$. The first is more efficient in all other cases. We will use $\texttt{PoKE}^*$ as a subprotocol for $\texttt{PoKE} - \texttt{1}$.

Protocol 3.3. *Proof of knowledge of the exponent* $(\texttt{PoKE} - \texttt{1})$:
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$; generators g_1, g_2 for $\mathbb{G}_1, \mathbb{G}_2$ respectively.
Inputs: Elements $a, b \in \mathbb{G}_1$
Claim: The Prover knows a polynomial $f(X) \in \mathbb{F}_p[X]$ such that $a^{f(s)} = b$.

1. The Prover $\mathcal{P}$ sends $\widetilde{g}_2 := g_2^{f(s)} \in \mathbb{G}_2$.
2. $\mathcal{P}$ sends a non-interactive proof for $\texttt{PoKE}^*[g_2, \ \widetilde{g}_2]$.
3. $\mathcal{V}$ verifies the proof for $\texttt{PoKE}^*[g_2, \ \widetilde{g}_2]$ and the equation

$$\mathbf{e}(a \ , \ \widetilde{g}_2) \stackrel{?}{=} \mathbf{e}(b \ , \ g_2).$$

$\mathcal{V}$ accepts if and only if the $\texttt{PoKE}^*$ is valid and the pairing equation holds. □

Clearly, a virtually identical proof would work if (a, b) was a pair in $\mathbb{G}_2$ instead of $\mathbb{G}_1$. Henceforth, we refer to this succinct proof as $\texttt{PoKE} - \texttt{1}[a, \ b]$ for a pair (a, b) in $\mathbb{G}_1^2$ or $\mathbb{G}_2^2$.

Proposition 3.4. *The protocol* $\texttt{PoKE} - \texttt{1}$ *is an argument of knowledge in the algebraic group model.*

Proof. We consider the case where a, b are elements of $\mathbb{G}_1$. The case where they are elements of $\mathbb{G}_2$ is virtually identical.

Suppose a PPT adversary $\mathcal{A}$ is able to output an element $\widetilde{g}_2 \in \mathbb{G}_2$ such that $\mathbf{e}(b, g_2) = \mathbf{e}(a, \widetilde{g}_2)$ along with a proof for $\texttt{PoKE}^*[g_2, \ \widetilde{g}_2]$. The $\texttt{PoKE}^*$ implies that with overwhelming probability, $\mathcal{A}$ can output a polynomial $f(X)$ such that $g_2^{f(s)} = \widetilde{g}_2$. The pairing check implies that the discrete logarithms between g_2, $\widetilde{g}_2$ and a, b coincide and hence, $a^{f(s)} = b$.

An extractor $\mathcal{E}$ can simulate the extractor for $\texttt{PoKE}^*[g_2, \ \widetilde{g}_2]$ to extract the polynomial $f(X)$ in polynomial expected time. □

When the pairing is type III, the exponentiations in $\mathbb{G}_2$ are substantially more expensive than those in $\mathbb{G}_1$. Thus, in cases where the Prover knows a polynomial $e(X)$ of a *small* degree such that $a = g_1^{e(s)}$, it can be cheaper to compute the element $a^{h(s)} = g_1^{e(s)\cdot h(s)} \in \mathbb{G}_1$ instead of $g_2^{h(s)} \in \mathbb{G}_2$.

Protocol 3.5. *Proof of knowledge of the exponent for bilinear accumulators* ($\texttt{PoKE}-2$):
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$; generators g_1, g_2 for $\mathbb{G}_1, \mathbb{G}_2$ respectively.
Inputs: Elements $a, b \in \mathbb{G}_1$
Claim: The Prover knows a polynomial $f(X) \in \mathbb{F}_p[X]$ such that $a^{f(s)} = b$.

1. The Prover $\mathcal{P}$ sends $\widetilde{g} := g_1^{f(s)} \in \mathbb{G}_1$.
2. The Fiat-Shamir heuristic generates a challenge $\alpha \in \mathbb{F}_p^*$.
3. $\mathcal{P}$ computes the polynomial $h(X) \in \mathbb{F}_p[X]$ and the element $\beta \in \mathbb{F}_p$ such that
$$f(X) = (X + \alpha) \cdot h(X) + \beta.$$
$\mathcal{P}$ computes
$$Q := a^{h(s)} \ , \ \widehat{g} := g_1^{h(s)}$$
and sends $(Q,\ \widehat{g},\ \beta) \in \mathbb{G}_1^2 \times \mathbb{F}_p^*$.
4. $\mathcal{V}$ verifies the equations
$$\mathbf{e}(Q\ ,\ g_2^{s+\alpha}) \cdot \mathbf{e}(a^{\beta}\ ,\ g_2) \stackrel{?}{=} \mathbf{e}(b\ ,\ g_2) \bigwedge \mathbf{e}(\widehat{g}\ ,\ g_2^{s+\alpha}) \cdot \mathbf{e}(g_1^{\beta}\ ,\ g_2) \stackrel{?}{=} \mathbf{e}(\widetilde{g}\ ,\ g_2)$$
and accepts if and only if both equations hold. □

If the exponentiation is in $\mathbb{G}_2$, the protocol `PoKE-1` will always be more efficient that protocol `PoKE-2`. We now describe an attack to show that the Protocol `PoKE-2` needs the Prover to send out $\widetilde{g} := g_2^{f(s)}$ *before* the challenge α is generated..

Attack: Suppose a Prover $\mathcal{P}_{\text{mal}}$ knows polynomials $h_1(X)$, $h_2(X)$ such that $g_1^{h_1(s)} = a$, $g_1^{h_2(s)} = b$ and $h_1(X)$ does not divide $h_2(X)$. With overwhelming probability, the challenge $\alpha \in \mathbb{F}_p^*$ is such that the polynomials $X + \alpha$ and $h_1(X)$ are relatively prime. On receiving the challenge α, $\mathcal{P}_{\text{mal}}$ could simply compute a polynomial $q(X) \in \mathbb{F}_p$ and an element $\beta \in \mathbb{F}_p$ such that
$$h_1(X) \cdot \beta + (X + \alpha) \cdot q(X) = h_2(X)$$
and send $Q := a^{q(s)}$, β to the Verifier. The Verifier then sees that $Q^{s+\alpha} a^{\beta} = b$ and is tricked into believing that the Prover knows a polynomial $f(X)$ such that $a^{f(s)} = b$.

Note that when $h_1(X)$ divides $h_2(X)$, this does not constitute an attack since $a^{h_2(s)/h_1(s)} = b$. But in the case where $h_1(X)$ does not divide $h_2(X)$, this attack shows that it is not sufficient for the Prover to send the pair $(Q,\ \beta) \in \mathbb{G}_1 \times \mathbb{F}_p$ to the Verifier. It is precisely to address this that we require the Prover to send the element $\widetilde{g} := g_1^{f(s)}$ before the challenge α is generated by the Fiat-Shamir heuristic.

Proposition 3.6. *The protocol* $\texttt{PoKE}-2$ *is an argument of knowledge in the algebraic group model.*

Proof. Suppose a PPT adversary $\mathcal{A}$ is able to output accepting transcripts $(\widetilde{g},\ Q_i,\ \widehat{g}_i,\ \beta_i)$ $(i = 1, 2)$ for challenges $\alpha_1,\ \alpha_2$ generated after $\widetilde{g}$ has been sent. Via the pairing checks, the Verifier verifies the equations

$$Q_i^{s+\alpha_i} = b \cdot a^{-\beta_i} \ \ , \ \ \widehat{g}_i^{s+\alpha_i} = \widetilde{g} \cdot g_1^{-\beta} \quad (i = 1, 2).$$

The KEA assumption implies that there is a PPT algorithm $\mathcal{A}$ that with overwhelming probability outputs polynomials $h_i(X)$ such that

$$g_1^{h_i(s)} = Q_i \ , \ g_1^{h_i(s)\cdot(s+\alpha)} = b \cdot a^{-\beta_i}$$

Furthermore,

$$a = g_1^{(\beta_1-\beta_2)^{-1}\cdot\left[h_1(s)-h_2(s))\right]} \ , \ b = g_1^{\beta_1\cdot[(\beta_1-\beta_2)^{-1}\left[h_1(s)-h_2(s)\right]+h_1(s)}$$

Setting

$$f_1(X) := (\beta_1 - \beta_2)^{-1}[h_1(X) - h_2(X)]$$

$$f_2(X) := \beta_1 \cdot [(\beta_1 - \beta_2)^{-1} \cdot [h_1(X) - h_2(X))] + h_1(X)$$

yields $a = g_1^{f_1(s)}$, $b = a^{f_2(s)}$. And a PPT algorithm that can output $h_1(X)$, $h_2(X)$ can also efficiently output the polynomials $f_1(X)$, $f_2(X)$.

Since the equations $\widehat{g}_i^{s+\alpha_i} = \widetilde{g}_1 \cdot g_1^{-\beta}$ $(i = 1, 2)$ hold, the KEA assumption implies that with overwhelming probability, $\mathcal{A}$ can output polynomials $e_i(X)$ $(i = 1, 2)$ such that

$$g_1^{e_i(s)} = \widehat{g}_1 \ \ , \ \ g_1^{e_i(s)\cdot(s+\alpha_i)+\beta_i} = \widetilde{g}_1.$$

Set $f(X) := e_1(X) \cdot (X + \alpha_1) + \beta_1$. Then $\widetilde{g}_1 = g_1^{f(s)}$. We argue that $f(X) \cdot f_1(X) = f_2(X)$ with overwhelming probability, which in turn will imply that $a^{f(s)} = b$ except with negligible probability.

Note that $f(X) \equiv \beta_1 \pmod{(X + \alpha_1)}$. In particular,

$$f(X) \cdot f_1(X) \equiv f_2(X) \pmod{(X + \alpha_1)}$$

and since α_1 is randomly and uniformly sampled from $\mathbb{F}_p^*$ after $\widetilde{g}$ has been sent, it follows that with overwhelming probability, $f(X) \cdot f_1(X) = f_2(X)$. Thus, $b = a^{f(s)}$

We now demonstrate witness extractability to show that this is an argument of knowledge. The extractor $\mathcal{E}$ with access to the accepting transcripts and to the CRS proceeds as follows. Given accepting transcripts (Q_i, β_i) for challenges α_i $(i = 1, \cdots, N)$, $\mathcal{E}$ uses the Chinese remainder theorem to compute the polynomial $e_N(X)$ such that

$$e_N(X) \equiv \beta \pmod{(X + \alpha_i)} \ \ , \ \ 1 = 1, \cdots, N.$$

If the equation

$$\mathbf{e}(a \ , \ g_2^{e_n(s)}) = \mathbf{e}(b \ , \ g_2)$$

holds, $\mathcal{E}$ halts. Otherwise, $\mathcal{E}$ samples the next accepting transcript (Q_{N+1}, β_{N+1}) and computes the polynomial $e_{N+1}(X)$ such that

$$e_{N+1}(X) \equiv e_N(X) \pmod{\prod_{i=1}^{N}(X + \alpha_i))} \ \ , \ \ e_{N+1}(X) \equiv \beta_{N+1} \pmod{(X + \alpha_{N+1})}.$$

via the Chinese remainder theorem. When the number of accepting transcripts sampled exceeds the degree of $f(X)$, the polynomial obtained by $\mathcal{E}$ is $f(X)$ with overwhelming probability. □

We now discuss a zero-knowledge variant of the protocol `PoKE` for bilinear accumulators. This is a honest verifier zero-knowledge argument system. It just requires the Prover to add a blinding factor to his `PoKE` proof.

Protocol 3.7. *ZK Proof of knowledge of the exponent* (ZKPoKE):
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$; generators g_1, g_2 for $\mathbb{G}_1, \mathbb{G}_2$ respectively.
Inputs: Elements $a, b \in \mathbb{G}_1$
Claim: The Prover knows a polynomial $f(X) \in \mathbb{F}_p[X]$ such that $a^{f(s)} = b$.

1. The Prover $\mathcal{P}$ chooses a random $k \in \mathbb{F}_p^*$ and sends $u := a^k \in \mathbb{G}_1$.
2. The Fiat-Shamir heuristic generates a challenge $\alpha \in \mathbb{F}_p^*$.
3. $\mathcal{P}$ sends a non-interactive proof for the $\texttt{PoKE}[a,\ b^\alpha \cdot u]$.
4. $\mathcal{V}$ computes $b^\alpha \cdot u \in \mathbb{G}_1$ and accepts if and only if the proof for $\texttt{PoKE}[a,\ b^\alpha \cdot u]$ is valid. $\square$

As was the case with the protocol PoKE, the protocol ZKPoKE can be easily modified for the setting where the exponentiation is in the group $\mathbb{G}_2$ instead of the group $\mathbb{G}_1$.

4 Protocol for the GCD and proofs of disjointness

Let $f_1(X)$, $f_2(X)$ be relatively prime polynomials in $\mathbb{F}_p[X]$ and let

$$\texttt{Com}(g_1, f_1(X)) := g_1^{f_1(s)} \ , \ \texttt{Com}(g_1, f_1(X)) := g_1^{f_2(s)} \ \in \ \mathbb{G}_1$$

be the [KZG10] commitments to these polynomials with base g_1. We provide a protocol whereby a Prover can succictly prove that the commitments are to two relatively prime polynomials.

The basic idea is that for polynomials $f_1(X)$, $f_2(X)$, $f_{1,2}(X)$, we have $\gcd(f_1(X)\ ,\ f_2(X)) = c \cdot f_{1,2}(X)$ for some constant $c \in \mathbb{F}_p$ if and only if the following hold:

1. $f_{1,2}(X)$ divides both $f_1(X)$ and $f_2(X)$
2. There exist polynomials $h_i(X)$ such that $\deg(h_1) < \deg(f_2)$ and
$$f_1(X) \cdot h_1(X) + f_2(X) \cdot h_2(X) = f_{1,2}(X).$$

The first property can be demonstrated using the protocol PoKE. The second can be demonstrated by sending $\mathbb{G}_2$-commitments to the polynomials $h_1(X)$, $h_2(X)$ along with PoKE proofs attesting that the Prover knows these committed polynomials. The equation

$$f_1(X) \cdot h_1(X) + f_2(X) \cdot h_2(X) = f_{1,2}(X)$$

can be verified via the pairing check

$$\mathbf{e}(g_1^{f_1(s)}\ ,\ g_2^{h_1(s)}) \cdot \mathbf{e}(g_1^{f_2(s)}\ ,\ g_2^{h_2(s)}) \ \stackrel{?}{=}\ \mathbf{e}(g_1^{f_{1,2}(s)}\ ,\ g_2).$$

This is an argument of knowledge for the following relation:

$$\mathcal{R}_{\texttt{GCD}}[g_1,\ (a_1,\ a_2),\ a_{1,2}] = \left\{ \begin{array}{l} \big((a_1,\ a_2,\ a_{1,2}\ \in \mathbb{G}_1) \\ f_1(X),\ f_2(X),\ f_{1,2}(X)\ \in\ \mathbb{F}_p[X]\big) : \\ g_1^{f_1(s)} = a_1\ ,\ g_1^{f_2(s)} = a_2\ ,\ g_1^{f_{1,2}(s)} = a_{1,2} \\ \mathbf{gcd}(f_1(X),\ f_2(X)) = f_{1,2}(X) \end{array} \right\}$$

Protocol 4.1. *Protocol for the greatest common divisor* (PoGCD)
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$; generators g_1, g_2 for $\mathbb{G}_1, \mathbb{G}_2$ respectively.
Inputs: Elements $a_1, a_2, a_{1,2} \in \mathbb{G}_1$
Claim: The Prover knows polynomials $f_1(X)$, $f_2(X)$, $f_{1,2}(X)$ and a constant c such that

- $a_1 = g_1^{f_1(s)}$, $a_2 = g_1^{f_2(s)}$, $a_{1,2} = g_1^{f_{1,2}(s)}$
- $\gcd(f_1(X),\ f_2(X)) = c \cdot f_{1,2}(X)$.

1. The Prover $\mathcal{P}$ sends (batchable) proofs for $\texttt{PoKE}[a_{1,2},\ a_1]$, $\texttt{PoKE}[a_{1,2},\ a_2]$.
2. $\mathcal{P}$ computes polynomials $h_1(X),\ h_2(X) \in \mathbb{F}_p[X]$ such that
$$f_1(X) \cdot h_1(X) + f_2(X) \cdot h_2(X) = f_{1,2}(X) \ , \ \deg h_1(X) < \deg f_2(X).$$
3. $\mathcal{P}$ computes the elements
$$\widehat{a}_1 := g_2^{h_1(s)} \ , \ \widehat{a}_2 := g_2^{h_2(s)} \ \in \ \mathbb{G}_2.$$
and sends $\widehat{a}_1$, $\widehat{a}_2$ to the Verifier $\mathcal{V}$ along with non-interactive proofs for $\texttt{PoKE}^*[g_1, a_1]$, $\texttt{PoKE}^*[g_1, a_2]$, $\texttt{PoKE}^*[g_2, \widehat{a}_1]$, $\texttt{PoKE}^*[g_2, \widehat{a}_2]$.
4. $\mathcal{V}$ verifies the equation
$$\mathbf{e}(a_1\ ,\ \widehat{a}_1) \cdot \mathbf{e}(a_2\ ,\ \widehat{a}_2) \stackrel{?}{=} \mathbf{e}(a_{1,2}\ ,\ g_2)$$
and the $\texttt{PoKE}^*$s. He accepts if and only if the $\texttt{PoKE}^*$s are valid and the equation holds. □

In particular, the protocol can allow a Prover to show that two committed polynomials are relatively prime. This is the special case where $f_{1,2}(X) = 1$ and $a_{1,2} = g_1$.

Proposition 4.2. *The protocol 4.1 is an argument of knowledge in the algebraic group model.*

Proof. An extractor $\mathcal{E}$ can simulate the extractors for the subprotocols $\texttt{PoKE}^*[g_1,\ a_1]$, $\texttt{PoKE}^*[g_1,\ a_2]$, $\texttt{PoKE}^*[g_2,\ \widehat{a}_1]$, $\texttt{PoKE}^*[g_2,\ \widehat{a}_2]$ to extract polynomials $f_1(X)$, $f_2(X)$, $h_1(X)$, $h_2(X)$ such that
$$a_1 = g_1^{f_1(s)}\ ,\ a_2 = g_1^{f_2(s)}\ ,\ \widehat{a}_1 = g_2^{h_1(s)}\ ,\ \widehat{a}_2 = g_2^{h_2(s)}.$$

The subprotocols $\texttt{PoKE}[a_{1,2},\ a_1]$, $\texttt{PoKE}[a_{1,2},\ a_2]$ imply that with overwhelming probability, $f_{1,2}(X)$ divides both $f_1(X)$ and $f_2(X)$ and hence, divides the $\gcd(f_1(X),\ f_2(X))$. The pairing equation
$$\mathbf{e}(a_1\ ,\ \widehat{a}_1) \cdot \mathbf{e}(a_2\ ,\ \widehat{a}_2) = \mathbf{e}(a_{1,2}\ ,\ g_2)$$
implies that
$$g_1^{f_1(s)\cdot h_1(s) + f_2(s)\cdot h_2(s)} = a_{1,2}.$$
The strong Diffie-Hellman assumption then implies that with overwhelming probability,
$$f_1(X) \cdot h_1(X) + f_2(X) \cdot h_2(X) = f_{1,2}(X),$$
whence it follows that $\gcd(f_1(X),\ f_2(X))$ divides $f_{1,2}(X)$. □

4.1 Batched non-membership proofs

In particular, the protocol for the GCD yields a protocol for batched non-membership proofs with the bilinear accumulator. We note that for data multisets $\mathcal{D}$, $\mathcal{D}_0$, the following are equivalent:

1. $\mathcal{D} \cap \mathcal{D}_0 = \emptyset$.

2. The polyomials $f_{\mathcal{D}}(X) := \prod_{d \in \mathcal{D}} (X+d)^{\text{mult}(d,\mathcal{D})}$, $f_{\mathcal{D}_0}(X) := \prod_{d_0 \in \mathcal{D}_0} (X+d_0)^{\text{mult}(d_0,\mathcal{D}_0)}$ are relatively prime.

As before, let $\mathcal{D}$ be the multiset of accumulated elements and let $\mathcal{D}_0$ be a set or a multiset of elements disjoint from $\mathcal{D}$. The accumulated digests are given by

$$\text{Acc}(g_1,\ \mathcal{D}) := g_1^{f_{\mathcal{D}}(s)} \ ,\ \ \text{Acc}(g_1,\ \mathcal{D}_0) := g_1^{f_{\mathcal{D}_0}(s)}.$$

A Prover who stores the sets $\mathcal{D}$, $\mathcal{D}_0$ can send a proof for $\texttt{PoGCD}[g_1,\ (\text{Acc}(\mathcal{D}),\ \text{Acc}(\mathcal{D}_0)),\ g_1]$ which implies that the polynomials committed in $\text{Acc}(\mathcal{D})$, $\text{Acc}(\mathcal{D}_0)$ are relatively prime and hence, the data multisets $\mathcal{D}$, $\mathcal{D}_0$ are disjoint. The proof is constant-sized, as is the verification time.

5 Non-repetition in committed sets

The non-membership proof in the preceding section boils down to succinctly proving that for elements $a_1,\ a_2 \in \mathbb{F}_p[X]$, the Prover knows relatively prime polynomials $f_1(X),\ f_2(X)$ such that $a_1 = g_1^{f_1(s)}$, $a_2 = g_1^{f_2(s)}$. For the non-membership proof of $\mathcal{D}_0$ in $\mathcal{D}$, this is achieved by setting

$$a_1 := \text{Acc}(g_1, \mathcal{D}) \ ,\ \ a_2 := \text{Acc}(g_1, \mathcal{D}_0).$$

The same technique can also be used to show that for an element $a \in \mathbb{G}_1$, the Prover knows a separable polynomial $f(X)$ such that $a = g_1^{f(s)}$. An obvious application is that the protocol can be used to demonstrate that a multiset commitment is actually a commitment to a *set* rather than to a multiset with some elements of multiplicity ≥ 2. We note that $\mathbb{F}_p$ is a perfect field and hence, a polynomial $f(X)$ being separable in $\mathbb{F}_p[X]$ is equivalent to $f(X)$ being separable in $\overline{\mathbb{F}}_p[X]$.

This does not seem possible (or at least not easy) with the other families of cryptographic accumulators: Merkle trees or the accumulators based on hidden order groups. In the former case, the proofs cannot be batched. In the later case, it boils down to proving that a committed integer is square-free, which seems difficult.

Our protocol hinges on the simple fact that a polynomial $f(X) \in \mathbb{F}_p[X]$ is square-free if and only if it is relatively prime with its derivative $f'(X)$. To this end, we first need a protocol to show that for polynomial commitments $a, b \in \mathbb{G}_1$, there is a polynomial $f(X)$ such that $a = g_1^{f(s)}$, $b = g_1^{f'(s)}$ where $f'(X)$ is the derivative of $f(X)$.

5.1 Protocol for the derivative of a polynomial

The protocol to prove that a committed polynomial is separable boils down to showing that the polynomial is relatively prime with its derivative. Thus, an important subprotocol is to succinctly show that for elements a, b in $\mathbb{G}_1$, the Prover knows a polynomial $f(X)$ such that $a = g_1^{f(s)}$, $b = g_1^{f'(s)}$.

Our protocol hinges on the following observation. For any element $\alpha \in \mathbb{F}_p$, the polynomial $f(X) - f'(\alpha)\cdot(X-\alpha)$ is $\equiv \beta \pmod{(X-\alpha)^2}$ for some $\beta \in \mathbb{F}_p$. We argue that the converse holds.

Let $f(X)$, $h(X)$ be polynomials and suppose, for a randomly generated $\alpha \in \mathbb{F}_p$, we have

$$f(X) - h(\alpha)\cdot(X-\alpha) \equiv \beta_1 \pmod{(X-\alpha)^2}$$

for some $\beta_1 \in \mathbb{F}_p$. Now,

$$f(X) - f'(\alpha) \cdot (X - \alpha) \equiv \beta_2 \pmod{(X-\alpha)^2}$$

for some $\beta_2 \in \mathbb{F}_p$. Hence,

$$(f'(X) - h(X)) \cdot (X - \alpha) \equiv \beta_2 - \beta_1 \pmod{(X-\alpha)^2},$$

which is only possible if $\beta_1 = \beta_2$ and $f'(\alpha) = h(\alpha)$. Since α was randomly generated, the Schwartz-Zippel lemma implies that with overwhelming probability, $f'(X) = h(X)$.

We use a somewhat non-standard definition of the derivative of a polynomial which will be useful to us here.

Definition 5.1. For a polynomial $f(X)$, *the derivative* of $f(X)$ is the unique polynomial $h(X)$ such that for any element $\alpha \in \mathbb{F}_p[X]$, there exists an element $\beta \in \mathbb{F}_p$ and a polynomial $q(X) \in \mathbb{F}_p[X]$ such that

$$f(X) = q(X) \cdot (X - \alpha)^2 + h(\alpha) \cdot (X - \alpha) + \beta. \qquad \square$$

Protocol 5.1. *Protocol for the derivative of a polynomial* (`PoDer`)
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$;
Inputs: Elements $a, b \in \mathbb{G}_1$
Claim: The Prover knows a polynomial $f(X) \in \mathbb{F}_p[X]$ such that $g_1^{f(s)} = a$ and $g_1^{f'(s)} = b$.

1. The Fiat-Shamir heuristic generates a challenge α.
2. The Prover $\mathcal{P}$ computes a polynomial $q(X)$ and an element $\beta \in \mathbb{F}_p$ such that
$$f(X) = q(X) \cdot (X - \alpha)^2 + f'(\alpha) \cdot (X - \alpha) + \beta$$
and sends $Q := g_1^{q(s)} \in \mathbb{G}_1$, $\beta \in \mathbb{F}_p$ to the Verifier $\mathcal{V}$ along with a non-interactive proof for $\texttt{PoKE}^*[g_1,\ Q]$.
3. $\mathcal{P}$ computes
$$\gamma := f'(\alpha)\ \ ,\ \ b_1 := g_1^{[f'(s)-\gamma]/(s-\alpha)}$$
and sends $(b_1, \gamma) \in \mathbb{G}_1 \times \mathbb{F}_p$ to $\mathcal{V}$ along with a non-interactive proof for $\texttt{PoKE}^*[g_1,\ b_1]$.
4. $\mathcal{V}$ verifies the `PoKE*`s and the equations
$$\mathbf{e}(a \cdot g_1^{-\beta}\ ,\ g_2) \stackrel{?}{=} \mathbf{e}(Q\ ,\ g_2^{(s-\alpha)^2}) \cdot \mathbf{e}(g_1^{\gamma}\ ,\ g_2^{s-\alpha}) \ \bigwedge\ \mathbf{e}(b \cdot g_1^{-\gamma}\ ,\ g_2) \stackrel{?}{=} \mathbf{e}(b_1\ ,\ g_2^{s-\alpha}).$$
$\square$

We will refer to this protocol as $\texttt{PoDer}[g_1,\ (a, b)]$. Clearly, a virtually identical protocol would work if (a, b) were a pair in $\mathbb{G}_2$ instead of $\mathbb{G}_1$.

The pairing check requires the Verifier to store $g_2^{s^2}$. This can be avoided by using a `PoE*` for the exponentiation $g_2^{(s-\alpha)^2}$ (at the cost of one more $\mathbb{G}_2$-element in the proof).

Proposition 5.2. *The protocol for the derivative of a polynomial is sound in the algebraic group model.*

Proof. Suppose a PPT adversary $\mathcal{A}$ outputs an accepting transcript in response to a challenge α generated by the Fiat-Shamir heuristic.

The pairing check $\mathbf{e}(b \cdot g_1^{-\gamma}\ ,\ g_2) \stackrel{?}{=} \mathbf{e}(b_1\ ,\ g_2^{s-\alpha})$ and the subprotocol $\texttt{PoKE}^*[g_1, b_1]$ imply that with overwhelming probability, $\mathcal{A}$ can output a polynomial $h(X)$ such that $b \cdot g_1^{-\gamma} = g_1^{h(s)\cdot(s-\alpha)}$. Setting

$$e(X) := h(X) \cdot (X - \alpha) + \gamma$$

yields $b = g_1^{e(s)}$, $e(\alpha) = \gamma$. We argue that with overwhelming probability, $e(X) = f'(X)$.

The pairing check

$$\mathbf{e}(a\ ,\ g_2) \stackrel{?}{=} \mathbf{e}(Q\ ,\ g_2^{(s-\alpha)^2}) \cdot \mathbf{e}(g_1^{\gamma}\ ,\ g_2^{s-\alpha}) \cdot \mathbf{e}(g_1^{\beta}\ ,\ g_2)$$

in conjunction with the subprotocol $\texttt{PoKE}^*[g_1,\ Q]$ implies that with overwhelming probability, $\mathcal{A}$ can output a polynomial $q(X)$ such that

$$a = g_1^{(s-\alpha)^2 \cdot q(s) + \gamma \cdot (s-\alpha) + \beta}.$$

Setting $f(X) := (X - \alpha)^2 \cdot q(X) + \gamma \cdot (X - \alpha) + \beta$ yields $a = g_1^{f(s)}$. Now,

$$f(X) \equiv \gamma \cdot (X - \alpha) + \beta \equiv e(\alpha) \cdot (X - \alpha) + \beta \pmod{(X - \alpha)^2}$$

and hence, $e(\alpha) \equiv f'(\alpha) \pmod{(X - \alpha)^2}$, which implies $e(\alpha) = f'(\alpha)$. Since α was randomly and uniformly sampled from $\mathbb{F}_p$, the Schwartz-Zippel lemma implies that with overwhelming probability, $e(X) = f'(X)$. □

5.2 Protocol for a separable polynomial commitment

We now turn to the main protocol of this section. Given a polynomial commitment in $\mathbb{G}_1$, we provide a protocol whereby a Prover can succinctly show that it is a commitment to a separable (square-free) polynomial. In the context of a bilinear accumulator, this amounts to showing that no data element was inserted more than once.

Protocol 5.3. *Protocol for separable polynomial commitment* (PoSep)
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$;
Inputs: Element $a \in \mathbb{G}_1$
Claim: The Prover knows a *separable* polynomial $f(X) \in \mathbb{F}_p[X]$ such that $g_1^{f(s)} = a$

1. The Prover computes the derivative $f'(X)$ and sends $a' := g_1^{f'(s)}$ along with a non-interactive proof for $\texttt{PoDer}[g_1,\ (a, a')]$.
2. $\mathcal{P}$ sends a proof for $\texttt{PoGCD}[g_1,\ (a,\ a'),\ g_1]$.
3. $\mathcal{V}$ accepts if and only if the PoDer and the PoGCD are both valid. □

We denote this protocol by $\texttt{PoSep}[g_1,\ a]$. Clearly, the protocol is easy to modify if the element a lies in the group $\mathbb{G}_2$ instead of $\mathbb{G}_1$.

Theorem 5.4. *The protocol for separable polynomial commitments is sound in the algebraic group model.*

Proof. It suffices to show that in case of an accepting transcript, a PPT adversary can - with overwhelming probability- output a polynomial $f(X) \in \mathbb{F}_p[X]$ such that $a = g_1^{f(s)}$ and $f(X)$ is relatively prime with its derivative $f'(X)$.

Suppose a PPT adversary $\mathcal{A}$ is able to output an accepting transcript. The subprotocol $\texttt{PoDer}[g_1,\ (a,\ a')]$ implies that with overwhelming probability, the Prover knows a polynomial $f(X)$ such that $a' = g_1^{f'(s)}$ and $a = g_1^{f(s)}$.

Furthermore, the subprotocol $\texttt{PoGCD}[g_1,\ (a,\ a'),\ g_1]$ implies that with overwhelming probability, $\mathcal{A}$ can output polynomials $h_1(X),\ h_2(X)$ such that

$$f(X) \cdot h_1(X) + f'(X) \cdot h_2(X) = 1,$$

whence it follows that with overwhelming probability, $f(X)$ and $f'(X)$ are relatively prime. $\square$

5.3 The radical of a polynomial and the underlying set of a multiset

For a polynomial $f(X) \in \mathbb{F}_p[X]$, we define its *radical* $f_{\texttt{rad}}(X)$ as the product of all distinct irreducible factors of $f(X)$. Equivalently, $f_{\texttt{rad}}(X)$ is the unique monic polynomial that generates the (principal) ideal

$$\sqrt{\big(f(X)\big)} := \big\{h(X) \in \mathbb{F}_p[X]\ :\ h(X)^N \in \big(f(X)\big) \text{ for some } N \in \mathbb{Z}\big\}\ \subseteq\ \mathbb{F}_p[X].$$

If $f(X)$ is monic, the two polynomials are linked by the equation

$$f_{\texttt{rad}}(X) = \frac{f(X)}{\gcd(f(X)\ ,\ f'(X))}$$

where $f'(X)$ denotes the derivative. Thus, if a Verifier has access to two polynomial commitments, a Prover can combine the protocols PoDer, PoGCD and PoProd to succinctly show that the second commitment represents the radical of the polynomial represented by the first. We refer to this as the protocol PoRad.

In particular, consider the case of a multiset $\mathcal{M}$ and its underlying set $\texttt{Set}(\mathcal{M})$ committed as in the [Ngu05] accumulator. The polynomials

$$f_{\mathcal{M}}(X) := \prod_{m \in \mathcal{M}} (X+m)^{\text{mult}(m,\mathcal{M})}\ \ ,\ \ f_{\texttt{Set}(\mathcal{M})}(X) := \prod_{m \in \texttt{Set}(\mathcal{M})} (X+m).$$

are such that $f_{\texttt{Set}(\mathcal{M})}$ is the radical of $f_{\mathcal{M}}(X)$.

Thus, if a Verifier has access to two [Ngu05] commitments, a Prover can combine the protocols PoDer, PoGCD and PoProd to succinctly show that the second [Ngu05] commitment represents the underlying set of the multiset represented by the first.

This is an argument of knowledge for the following relation:

$$\mathcal{R}_{\texttt{Rad}}[g_1,\ (a,\ a_{\texttt{rad}})] = \left\{ \begin{array}{l} (a,\ a_{\texttt{rad}}\ \in\ \mathbb{G}_1),\ f(X) \in \mathbb{F}_p[X],\ c \in \mathbb{F}_p) : \\ g_1^{f(s)} = a\ ,\ g_1^{c \cdot \texttt{rad}(f(s))} = a_{\texttt{rad}} \end{array} \right\}$$

Protocol 5.5. *Protocol for the radical of a polynomial* (PoRad)
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$;
Inputs: Elements $a, a_{\texttt{rad}} \in \mathbb{G}_1$
Claim: The Prover knows a polynomial $f(X) \in \mathbb{F}_p[X]$ with radical $f_{\texttt{rad}}(X)$ and a constant $c \in \mathbb{F}_p$ such that

$$g_1^{f(s)} = a\ \ ,\ \ g_1^{c \cdot f_{\texttt{rad}}(s)} = a_{\texttt{rad}}$$

1. The Prover $\mathcal{P}$ computes the derivative $f'(X)$ and sends the element $a' := g_1^{f'(s)} \in \mathbb{G}_1$

along with a proof for $\texttt{PoDer}[g_1,\ (a,\ a')]$.

2. $\mathcal{P}$ computes the polynomial $f_0(X) := \mathbf{gcd}(f(X),\ f'(X))$ and sends the element

$$a_0 := g_1^{f_0(s)} \in \mathbb{G}_1$$

along with a proof for $\texttt{PoGCD}[g_1,\ (a,\ a'),\ a_0]$.

3. $\mathcal{P}$ sends the element $b_{\texttt{rad}} := g_2^{f_{\texttt{rad}}(s)} \in \mathbb{G}_2$.

4. The Verifier $\mathcal{V}$ verifies the PoGCD, the PoDer and the (batchable) equations

$$\mathbf{e}(g_1\ ,\ b_{\texttt{rad}}) \stackrel{?}{=} \mathbf{e}(a_{\texttt{rad}}\ ,\ g_2)\ \ ,\ \ \mathbf{e}(a_0\ ,\ b_{\texttt{rad}}) \stackrel{?}{=} \mathbf{e}(a\ ,\ g_2).$$

□

5.4 A protocol for compositions of polynomials

The next protocol demonstrates that given three polynomial commitments, the third equals composition of the first two. Given polynomials $f_1(X)$, $f_2(X)$, $f_{1,2}(X)$, the following are equivalent with overwhelming probability:

1. $f_{1,2}(X) = f_1(f_2(X))$.
2. For a random challenge $\gamma \in \mathbb{F}_p$, $f_{1,2}(X) - f_1(\gamma)$ is divisible by $f_2(X) - \gamma$.

To this end, the Prover sends a commitment to $f_1(\gamma)$ and proves that it is, in fact, a commitment to $f_1(\gamma)$. This subprotocol hinges on the fact that for a *constant* $\beta \in \mathbb{F}_p$, the following are equivalent:

1. $\beta = f_2(\gamma)$
2. $f_2(X) - \beta$ is divisible by $X - \gamma$.

This is an argument of knowledge for the following relation:

$$\mathcal{R}_{\texttt{Comp}}[g_1,\ (a_1, a_2),\ a_{1,2}] = \left\{ \begin{array}{l} \big((a_1, a_2, a_{1,2} \in \mathbb{G}_1),\ f_1(X),\ f_2(X) \in \mathbb{F}_p[X]\big) : \\ g_1^{f_1(s)} = a_1\ ,\ g_1^{f_2(s)} = a_2\ ,\ g_1^{f_1(f_2(s))} = a_{1,2} \end{array} \right\}$$

The protocol PoSep can be combined with this protocol to generate a succinct proof that each element that was inserted into the accumulator was inserted precisely n times, which generalizes the protocol PoSep.

Protocol 5.6. *Proof of composition* (PoComp)
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$; generators $g_1,\ g_2$ for $\mathbb{G}_1,\ \mathbb{G}_2$ respectively.
Inputs: Elements $a_1,\ a_2,\ a_{1,2} \in \mathbb{G}_1$
Claim: The Prover knows polynomials $f_1(X),\ f_2(X)\ \in\ \mathbb{F}_p[X]$ such that

$$g_1^{f_1(s)} = a_1\ \ ,\ \ g_1^{f_2(s)} = a_2\ \ ,\ \ g_1^{f_1(f_2(s))} = a_{1,2}$$

1. The Fiat-Shamir heuristic generates a challenge γ.
2. $\mathcal{P}$ sends the $\mathbb{F}_p$-elements $\alpha_1 := f_1(\gamma)$, $\alpha_2 := f_2(\gamma)$, $\alpha_{1,2} := f_{1,2}(\gamma)$.
3. $\mathcal{P}$ sends the $\mathbb{G}_1$-elements

$$Q_1 := g_1^{[f_1(s)-\alpha_1]/[s-\gamma]} \ , \ Q_2 := g_1^{[f_2(s)-\alpha_2]/[s-\gamma]} \ , \ Q_{1,2} := g_1^{[f_{1,2}(s)-\alpha_{1,2}]/[s-\gamma]}$$

4. $\mathcal{P}$ sends the element $\widehat{Q} := g_1^{[f_{1,2}(s)-\alpha_1]/[f_2(s)-\gamma]}$.
5. $\mathcal{V}$ accepts if and only if the (batchable) equations

$$\mathbf{e}(Q_1 \ , \ g_2^{s-\gamma}) \stackrel{?}{=} \mathbf{e}(a_1 \cdot g_1^{-\alpha_1} \ , \ g_2) \ \ , \ \ \mathbf{e}(Q_2 \ , \ g_2^{s-\gamma}) \stackrel{?}{=} \mathbf{e}(a_1 \cdot g_1^{-\alpha_2} \ , \ g_2),$$

$$\mathbf{e}(Q_{1,2} \ , \ g_2^{s-\gamma}) \stackrel{?}{=} \mathbf{e}(a_{1,2} \cdot g_1^{-\alpha_{1,2}} \ , \ g_2) \ \ , \ \ \mathbf{e}(\widehat{Q} \ , \ a_2 \cdot g_2^{-\gamma}) \stackrel{?}{=} \mathbf{e}(a_{1,2} \cdot g_1^{-\alpha_1} \ , \ g_2)$$

hold. □

Proposition 5.7. *The protocol* `ZKPoComp` *is secure in the algebraic group model.*

Proof. (Sketch) Suppose a PPT algorithm $\mathcal{A}$ outputs an accepting transcript.

The pairing checks

$$\mathbf{e}(Q_1 \ , \ g_2^{s-\gamma}) \stackrel{?}{=} \mathbf{e}(a_1 \cdot g_1^{-\alpha_1} \ , \ g_2) \ \ , \ \ \mathbf{e}(Q_2 \ , \ g_2^{s-\gamma}) \stackrel{?}{=} \mathbf{e}(a_1 \cdot g_1^{-\alpha_2} \ , \ g_2),$$

imply that with overwhelming probability, $\mathcal{A}$ can output polynomials $\widehat{f}_1(X)$, $\widehat{f}_2(X)$ such that

$$g_1^{\widehat{f}_1(s)} = a_1 \ \ , \ \ g_1^{\widehat{f}_2(s)} = a_2 \ \ ,$$

$$\widehat{f}_1(X) \equiv \alpha_1 \pmod{(X-\gamma)} \ \ , \ \ \widehat{f}_2(X) \equiv \alpha_2 \pmod{(X-\gamma)}.$$

Similarly, the pairing check $\mathbf{e}(Q_{1,2} \ , \ g_2^{s-\gamma}) \stackrel{?}{=} \mathbf{e}(a_{1,2} \cdot g_1^{-\alpha_{1,2}})$ implies that with overwhelming probability, $\mathcal{A}$ can output a polynomial $\widehat{f}_{1,2}(X)$ such that

$$g_1^{\widehat{f}_{1,2}(s)} = a_{1,2} \ \ , \ \ \widehat{f}_{1,2}(X) \equiv \alpha_{1,2} \pmod{(X-\gamma)}.$$

Furthermore, the pairing check $\mathbf{e}(\widehat{Q} \ , \ a_2 \cdot g_2^{-\gamma}) \stackrel{?}{=} \mathbf{e}(a_{1,2} \cdot g_1^{-\alpha_1} \ , \ g_2)$ implies that with overwhelming probability,

$$\widehat{f}_{1,2}(X) \equiv \alpha_1 \equiv \widehat{f}_1(\gamma) \equiv \widehat{f}_1(\widehat{f}_2(X)) \pmod{(\widehat{f}_2(X) - \gamma)}$$

and since γ was randomly and uniformly generated, it follows that with overwhelming probability, $\widehat{f}_{1,2}(X) = \widehat{f}_1(\widehat{f}_2(X))$. □

5.4.1 Frequencies of elements

The protocol `PoSep` can be combined with the protocol `PoComp` to demonstrate that an element of $\mathbb{G}_1$ is a polynomial commitment to the n-th power of a separable polynomial. In the context of a bilinear accumulator, this demonstrates that any element inserted was inserted precisely n times. Furthermore, these protocols an also be used to prove that any element inserted into the accumulator was inserted no fewer than m times and no more than n times. We describe the protocol below.

The basic idea is that if a polynomial $f(X)$ is sandwiched between polynomials $f_0(X)^m$ and $f_0(X)^n$ in that it is divisible by $f_0(X)^m$ and divides $f_0(X)^n$ for a separable polynomial $f_0(X)$, then each irreducible factor of $f(X)$ divides it with some multiplicity between m and n (inclusive). The special case where $m = n$ entails succinctly proving that the multiplicity of each irreducible factor in $\mathbb{F}_p[X]$ is n.

Protocol 5.8. *Protocol for multiplicities of irreducible factors* (`PoFreq`)
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$;
Inputs: Element $a \in \mathbb{G}_1$; integers m, n with $m \leq n$.
Claim: The Prover knows a *separable* polynomial $f_0(X) \in \mathbb{F}_p[X]$ and a polynomial $f(X) \in \mathbb{F}_p[X]$ such that:
- $g_1^{f(s)} = a$
- $f_0(X)^m$ divides $f(X)$
- $f_0(X)^n$ is divisible by $f(X)$

1. The Prover $\mathcal{P}$ computes the radical $f_0(X)$ of $f(X)$ and sends

$$a_0 := g_1^{f_0(s)} \ , \ a_m := g_1^{f_0(s)^m} \ , \ a_n := g_1^{f_0(s)^n}$$

to the Verifier $\mathcal{V}$ along with proofs for $\texttt{PoRad}[g_1,\ (a_m, a_0)]$, $\texttt{PoRad}[g_1,\ (a_n, a_0)]$.

2. $\mathcal{P}$ sends a proof for $\texttt{PoSep}[g_1,\ a_0]$.

3. $\mathcal{P}$ sends the elements $g_m := g_1^{s^m}$, $g_n := g_1^{s^n}$ along with proofs for $\texttt{PoE}^*[g_1,\ X^m,\ a_m]$, $\texttt{PoE}^*[g_1,\ X^n,\ a_n]$.

4. $\mathcal{P}$ sends proofs for $\texttt{PoKE}[a_m,\ a]$, $\texttt{PoKE}[a,\ a_n]$.

5. $\mathcal{V}$ accepts if and only if all of the proofs are valid. □

6 Applications of the protocol for the derivative

In this section, we describe a protocol (and related protocols) that allows a Prover to show that a committed polynomial has degree that coincides with a committed integer. In particular, this can be used to show that two committed polynomials are of the same degree without revealing this common degree. In contrast to the previous sections, we discuss the HVZK arguments of knowledge rather than just the succinct AoK versions. This is because while it is straightforward to transform the protocols from previous sections into HVZK's, it is a bit more subtle when it comes to linking the committed polynomial to its degree.

We briefly describe the approach. We first deal with the special case where the committed polynomial $f(X)$ is a monomial, i.e. $f(X) = c_n \cdot X^n$ for some $c_n \in \mathbb{F}_p^*$, $n \in \mathbb{Z}^{\geq 1}$. Note that

$$\deg(f(X)) = n = X \cdot f'(X) \cdot f(X)^{-1}.$$

Thus, given commitments $a = g_1^{f(s)}$, $a_{\deg} = g_1^{\deg(f)}$, the Prover can verifiably send a commitment $a' = g_1^{f'(s)}$ to the derivative $f'(X)$ and then show that the the polynomials $f(X)$, $f'(X)$ are such that $f'(1) \cdot f(1)^{-1}$ is the field element committed in $a_{\deg}$.

The more general case of an arbitrary polynomial uses the special case of a monomial in conjuction with a protocol that shows that the degree of the polynomial $f_1(X) := f(X) - c_n \cdot X^n$ is less than n. Note that a polynomial $f_1(X)$ is of degree less than n is and only if the rational function $c_n X^n \cdot f_1(X^{-1})$ is a polynomial divisible by X.

The Prover first shows that $g_1^{c_n \cdot s^n}$ is a commitment to a monomial of degree n where n is the field element committed in $a_{\deg} = g_1^n$. He then sends the element $a_1 := g_1^{c_n s^n \cdot f_1(s^{-1})}$ and proves that this is a commitment to $X^n \cdot f_1(X^{-1})$. This can be done by showing that for a randomly generated challenge β, the polynomial $h(X)$ committed in a_1 is such that:

1. $h(X) - X^n \cdot f_1(\beta^{-1})$ is divisible by $X - \beta^{-1}$.
2. $h(X)$ is divisible by X.

6.1 A zero-knowledge variant of the protocol for the derivative

We will need the following zero-knowledge variant of the protocol for the derivative.

Protocol 6.1. *Zero-knowledge proof of the derivative of a polynomial* (`ZKPoDer`)
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$; generators $g_1,\ g_2$ for $\mathbb{G}_1,\ \mathbb{G}_2$ respectively.
Inputs: Elements $a, b \in \mathbb{G}_1$
Claim: The Prover knows a polynomial $f(X) \in \mathbb{F}_p[X]$ such that $g_1^{f(s)} = a$ and $g_1^{f'(s)} = b$.

1. The Fiat-Shamir heuristic generates a challenge α.
2. The Prover $\mathcal{P}$ computes the polynomial $q(X)$ and the constant $\beta \in \mathbb{F}_p$ such that
$$f(X) = q(X) \cdot (X - \alpha)^2 + f'(\alpha) \cdot (X - \alpha) + \beta.$$
3. $\mathcal{P}$ sends the elements
$$a_0 := g_1^{\beta} \ , \ C_{f'(\alpha)} := g_1^{f'(\alpha)}$$
along with (batchable) proofs for `ZKPoConst`$[g_1,\ a_0]$ and `ZKPoConst`$[g_1,\ C_{f'(\alpha)}]$.
4. $\mathcal{P}$ sends the element $a_2 := g_1^{q(s)\cdot(s-\alpha)^2}$ along with a proof for `ZKPoKE`$[g_1^{(s-\alpha)^2},\ a_2]$.
5. $\mathcal{P}$ sends a proof for `ZKPoKE`$[g_1^{s-\alpha},\ b \cdot C_{f'(\alpha)}^{-1}]$.
6. $\mathcal{V}$ verifies the `ZKPoConst`s, the `ZKPoKE`s and the equation
$$\mathbf{e}(C_{f'(\alpha)}\ ,\ g_2^{s-\alpha}) \stackrel{?}{=} \mathbf{e}(a \cdot a_2^{-1} \cdot a_0^{-1}\ ,\ g_2).$$
□

We will need the following elementary fact. We defer the proof to the appendix.

Lemma 6.2. *For a polynomial $f(X)$ and an element $\alpha \in \mathbb{F}_p$, the following are equivalent:*

1. $f(X) = c_0 \cdot (X - \alpha) \cdot f'(X)$ *for some* $c_0 \in \mathbb{F}_p^*$, $\alpha \in \mathbb{F}_p$
2. $f(X) = c \cdot (X - \alpha)^n$ *for some* $c \in \mathbb{F}_p^*$, $\alpha \in \mathbb{F}_p$, $n \in \mathbb{Z}^{\geq 0}$.

Proof. (1) $\Rightarrow$ (2): This is vacuous when $\deg(f(X)) = 1$. We now proceed by induction on the degree of $f(X)$. Let $n \geq 2$ be the degree of $f(X)$ and suppose the statement is true for polynomials of degree $\leq n - 1$.

Differentiating both sides yields
$$f'(X) = c_0 \cdot [(X - \alpha) \cdot f''(X) + f'(X)]$$
and hence,
$$f'(X) = c_0 \cdot (1 - c_0)^{-1} \cdot (X - \alpha) \cdot f''(X).$$
Since $\deg(f'(X)) < n$, it follows that $f'(X) = c_1 \cdot (X - \alpha)^{n-1}$ for some $c_1 \in \mathbb{F}_p^*$. This yields
$$f(X) = c_0 \cdot c_1 \cdot (X - \alpha)^{n-1},$$
which completes the proof.

(2) $\Rightarrow$ (1): This direction is trivial. □

The derivative $f'(X)$ has degree one less than that of $f(X)$. If $f(X)$ is divisible by $f'(X)$, the quotient $f(X) \cdot f'(X)^{-1}$ is a linear polynomial and hence, is of the form $c_0 \cdot (X - \alpha)$ for some $c \in \mathbb{F}_p^*$, $\alpha \in \mathbb{F}_p$. So $f(X) = c_0 \cdot (X - \alpha) \cdot f'(X)$ and hence, $f(X) = c \cdot (X - \alpha)^{\deg(f)}$ for some $c \in \mathbb{F}_p^*$.

For such a polynomial $f(X) = c \cdot (X - \alpha)^{\deg(f)}$, we have

$$f(0) = 0 \iff \alpha = 0 \iff f(X) = c \cdot X^{\deg(f)}.$$

Similarly,

$$f(0) = 0 \ \wedge \ f(1) = 1 \iff \alpha = 0 \ \wedge \ c = 1 \iff f(X) = X^{\deg(f)}.$$

Protocol 6.3. *Zero-knowledge proof of linear power* (`ZKPoLinPow`)
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$; generators $g_1,\ g_2$ for $\mathbb{G}_1,\ \mathbb{G}_2$ respectively.
Inputs: Elements $a \in \mathbb{G}_1$
Claim: $a = g_1^{c \cdot (s-\alpha)^n}$ for some $c, \alpha \in \mathbb{F}_p$ and $n \in \mathbb{Z}$ known to the Prover.

1. The Prover $\mathcal{P}$ computes $a' = g_1^{c \cdot n \cdot (s-\alpha)^{n-1}}$ and sends it to the Verifier $\mathcal{V}$ along with a proof of `ZKPoDer`$[g_1,\ (a, a')]$.
2. $\mathcal{P}$ generates a non-interactive proof of `ZKPoKE`$[a',\ a]$ and sends it to $\mathcal{V}$.
3. $\mathcal{V}$ accepts if and only if the `ZKPoKE` and the `ZKPoDer` are valid. □

To show that a committed polynomial $f(X)$ is $c \cdot X^n$ for *some* integer $n \geq 0$ and a constant c, a Prover needs to demonstrate the following properties:

1. $f(X)$ is divisible by the derivative $f'(X)$.
2. $f(0) = 0$.

For such a monomial $f(X) = c \cdot X^n$, the degree n is given by $n = f'(1) \cdot f(1)^{-1}$. In particular, if $f(X) = X^n$, the degree n is given by $n = f'(1)$.

This is a HVZK for the following relation:

$$\mathcal{R}_{\texttt{DegMono}}[g_1,\ (a, a_{\deg})] = \{(a, a_{\deg} \in \mathbb{G}_1),\ n \in \mathbb{Z}^{\geq 0},\ c \in \mathbb{F}_p) :\ g_1^{c \cdot s^n} = a\ ,\ g_1^n = a_{\deg}\}$$

We will later use this as a building block for a more general protocol (`ZKPoDeg`) that allows a Prover to demonstrate that the degree of an arbitrary committed polynomial coincides with a committed integer $\leq$ length(`CRS`).

Protocol 6.4. *Zero-knowledge proof of degree of monomial* (`ZKPoDegMono`)
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$; generators $g_1,\ g_2$ for $\mathbb{G}_1,\ \mathbb{G}_2$ respectively.
Inputs: Elements $a, a_{\deg} \in \mathbb{G}_1$
Claim: $a = g_1^{c \cdot s^n}$, $a_{\deg} = g_1^n$ for some $n \in \mathbb{Z}^{\geq 0}$, $c \in \mathbb{F}_p$ known to the Prover.

1. The Prover $\mathcal{P}$ sends the elements

$$a' := g_1^{c \cdot n \cdot s^{n-1}} \quad , \quad a'_1 := (a')^s = g_1^{c \cdot n \cdot s^n}$$

along with a proof for `ZKPoDer`$[g_1,\ (a,\ a')]$.

2. $\mathcal{P}$ sends a zero-knowledge proof of equality of constant discrete logarithms (Chaum-Pedersen) between $[g_1,\ a_{\deg}]$ and $[a,\ a'_1]$.

3. $\mathcal{V}$ accepts if and only if the `ZKPoDer` and the Chaum-Pedersen proofs are valid and the equation

$$\mathbf{e}(a'_1\ ,\ g_2) \stackrel{?}{=} \mathbf{e}(a'\ ,\ g_2^s)$$

holds. □

Proposition 6.5. *The protocol* `ZKPoDegMono` *is secure in the algebraic group model.*

Proof. (Sketch) Suppose a PPT algorithm $\mathcal{A}$ outputs an accepting transcript.

The pairing check $\mathbf{e}(a'_1\ ,\ g_2) \stackrel{?}{=} \mathbf{e}(a'\ ,\ g_2^s)$ implies that $a'_1 = (a')^s$. The subprotocol `ZKPoDer`$[g_1,\ (a,\ a')]$ implies that with overwhelming probability, $\mathcal{A}$ can output a polynomial $f_0(X)$ such that

$$g_1^{f_0(s)} = a = a_1^s\ \ ,\ \ g_1^{f'_0(s)} = a'.$$

The strong Diffie Hellman assumption implies that with overwhelming probability, $f(X)$ is divisible by X.

The Chaum-Pedersen proof implies that $\mathcal{A}$ can output a constant n_0 such that

$$g_1^{n_0} = a_{\deg}\ \ ,\ \ g_1^{n_0 \cdot f_0(s)} = a^{n_0} = a'_1 = (a')^s = g_1^{s \cdot f'_0(s)}.$$

Thus, with overwhelming probability, the polynomial $f_0(X)$ is such that $f'_0(X)$ divides $f(X)$ and $f(0) = 0$. The preceding lemma implies that with overwhelming probability, $f_0(X)$ is a monomial, say $f_0(X) = d_0 \cdot X^{m_0}$ for some $d_0 \in \mathbb{F}_p$, $m \in \mathbb{Z}_{\geq 0}$. Then $f'_0(X) = d_0 \cdot m_0 \cdot X^{m_0 - 1}$. So

$$n_0 = f_0(s)^{-1} \cdot f(s) = m_0,$$

whence it follows that $g_1^{\deg(f_0)} = a_{\deg}$. □

6.2 Linking a committed polynomial to its degree

The next protocol allows a Prover to succinctly demonstrate that a committed polynomial $f(X)$ is of degree less than a committed integer n. It hinges on the simple observation that the following are equivalent for any integer n and committed polynomial $f(X)$:

1. $\deg(f) < n$
2. The rational function $X^n \cdot f(X^{-1})$ is a polynomial divisible by X.

Given commitments $a = g_1^{f(s)}$, $b = g_1^n$, the Prover sends a commitment $\widetilde{g}_1 := g_1^{s^n}$ to X^n and uses the subprotocol `ZKPoDegMono` to show that this is a commitment to X^n, where n is the integer committed in b.

The Prover then sends a commitment $a^\vee := g_1^{h(s)}$ to the polynomial $h(X) := X^n \cdot f(X^{-1})$ along with a proof for `ZKPoKE`$[g_1^s,\ a^\vee]$. This demonstrates that $a^\vee$ is a commitment to a polynomial divisible by X.

For a randomly generated challenge γ, the Prover *verifiably* sends the element $g_1^{f(\gamma)}$. along with a proof that this is a commitment to the evaluation of $f(X)$ at γ. He also sends the element $a_\gamma := g_1^{s^n \cdot f(\gamma)}$ and uses the Chaum-Pedersen protocol to show that the discrete logarithm between the pair $(g_1^{s^n},\ a_\gamma)$ is a constant that coincides with the discrete logarithm between the pair $(g_1,\ g_1^{f(\gamma)})$.

The Prover then shows that the polynomial $h(X)$ committed in $a^\vee$ is such that

$$h(X) \equiv X^n \cdot f(\gamma) \pmod{(X - \gamma^{-1})}.$$

He does so by producing a $(s - \gamma^{-1})$-th root of $a^\vee \cdot a_\gamma^{-1}$. Since γ is randomly and uniformly generated, this implies that with overwhelming probability, $h(X) = X^n \cdot f(X^{-1})$.

This is a HVZK for the following relation:

$$\mathcal{R}_{\texttt{DegUp}}[g_1,\ (a_1, a_2)] = \left\{ \begin{array}{l} (a_1, a_2 \in \mathbb{G}_1),\ f(X) \in \mathbb{F}_p[X],\ n \in \mathbb{Z}) : \\ g_1^{f(s)} = a_1\ ,\ g_1^n = a_2\ ,\ \deg(f) < n \end{array} \right\}$$

Protocol 6.6. *Zero-knowledge proof of degree upper bound* (ZKPoDegUp)
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$; generators $g_1,\ g_2$ for $\mathbb{G}_1,\ \mathbb{G}_2$ respectively.
Inputs: Elements $a, b \in \mathbb{G}_1$
Claim: The Prover knows an integer n and a polynomial $f(X)$ such that

$$a = g_1^{f(s)}\ \ ,\ \ b = g_1^n\ \ ,\ \ \deg(f) < n.$$

1. The Prover $\mathcal{P}$ chooses a random $c \in \mathbb{F}_p$ and sends $\widetilde{g}_1 := g_1^{c \cdot s^n}$ along with a proof of $\texttt{ZKPoDegMono}[g_1,\ (\widetilde{g}_1,\ b)]$.
2. $\mathcal{P}$ sends the elements
$$a^\vee := g_1^{c \cdot s^n \cdot f(s^{-1})}$$
along with a proof for $\texttt{ZKPoKE}[g_1^s,\ a^\vee]$
3. The Fiat-Shamir heuristic generates a challenge γ.
4. $\mathcal{P}$ sends the elements
$$C_{f,\gamma} := g_1^{f(\gamma)}\ \ ,\ \ a_\gamma := g_1^{c \cdot s^n \cdot f(\gamma)}$$
along with a zero-knowledge proof of equality of constant discrete logarithms (Chaum-Pedersen) between $(g_1,\ C_{f,\gamma})$ and $(\widetilde{g}_1,\ a_\gamma)$.
5. $\mathcal{P}$ sends proofs for $\texttt{ZKPoKE}[g_1^{s-\gamma},\ a \cdot C_{f,\gamma}^{-1}]$ and $\texttt{ZKPoKE}[g_1^{s-\gamma^{-1}},\ a^\vee \cdot a_\gamma^{-1}]$
6. $\mathcal{V}$ verifies the ZKPoKEs, the Chaum-Pedersen proof and the ZKPoDegMono. □

Proposition 6.7. *The protocol* ZKPoDegUp *is secure in the algebraic group model.*

Proof. (Sketch) Suppose a PPT algorithm $\mathcal{A}$ outputs an accepting transcript.

The subprotocol $\texttt{ZKPoDegMono}[g_1,\ (\widetilde{g}_1,\ b)]$ implies that with overwhelming probability, $\mathcal{A}$ can output $c_0 \in \mathbb{F}_p$, $n_0 \in \mathbb{Z}$ such that

$$\widetilde{g}_1 = g_1^{c_0 \cdot s^{n_0}}\ \ ,\ \ b = g_1^{n_0}.$$

The subprotocols

$$\texttt{ZKPoKE}[g_1^{s-\gamma},\ a \cdot C_{f,\gamma}^{-1}]\ \ ,\ \ \texttt{ZKPoKE}[g_1^{s-\gamma^{-1}},\ a^\vee \cdot a_\gamma^{-1}]$$

imply that with overwhelming probability, $\mathcal{A}$ can output polynomials $f_0(X)$, $f_0^\vee(X)$, $h(X)$ such that

$$g_1^{f_0(s)} = a\ \ ,\ \ g_1^{f_0^\vee(s)} = a^\vee\ \ ,\ \ g_1^{f_0(\gamma)} = C_{f,\gamma}\ \ ,\ \ g_1^{h(s)} = a_\gamma\ \ ,\ \ h(X) \equiv f_0^\vee(X) \pmod{(X - \gamma^{-1})}$$

The Chaum-Pedersen proof implies that

$$g_1^{c_0 \cdot s^{n_0} \cdot f_0(\gamma)} = \widetilde{g}_1^{f_0(\gamma)} = a_\gamma = g_1^{h(s)}.$$

Thus,

$$c_0 \cdot X^{n_0} \cdot f_0(X^{-1}) \equiv f_0^{\vee}(X) \pmod{(X - \gamma^{-1})}$$

and since γ was randomly and uniformly generated, this implies that with overwhelming probability, $f_0^{\vee}(X) = c_0 \cdot X^{n_0} \cdot f_0(X^{-1})$.

Lastly, the subprotocol $\texttt{ZKPoKE}[g_1^s,\ a^{\vee}]$ implies that with overwhelming probability, $f_0^{\vee}(X)$ is divisible by X, whence it follows that $\deg(f_0) < n_0$. □

We note that the proofs of bounded degrees are inherently batchable. If polynomials $f_1(X), \cdots, f_j(X)$ are all of degree less than n, then with overwhelming probability, the sum

$$f_\gamma(X) := \sum_{i=1}^{j} f_i(X) \cdot \gamma^i$$

is a polynomial of degree $< n$. Given committed polynomials $f_1(X), \cdots, f_j(X)$, a Prover can demonstrate this degree upper bound by showing that for a randomly generated challenge γ, the aggregated polynomial $f_\gamma(X)$ is such that the rational function $X^n \cdot f_\gamma(X^{-1})$ is a polynomial divisible by X.

We now move to the main protcol in this section. This protocol links a polynomial commitment to a commitment to its degree. It allows a Prover to prove in zero knowledge that given commitments to a polynomial $f(X)$ and an integer n, the committed polynomial is of degree n. More formally, given elements $a, a_{\deg} \in \mathbb{G}_1$, the Prover knows a polynomial $f(X)$ such that

$$a = g_1^{f(s)} \quad , \quad a_{\deg} = g_1^{\deg(f)}.$$

In particular, this can be used to show that two committed polynomials are of the same degree without revealing this common degree. More importantly, it enables the subsequent protocols in this paper.

The Prover starts out by isolating the leading term

$$\texttt{Coef}(f(X)\ ,\ \deg(f)) \cdot X^{\deg(f)}$$

of $f(X)$, providing a commitment to this monomial and using the protocol `ZKPoDegMono` to show that this is a commitment to a monomial of a degree that coincides with an integer committed in $a_{\deg}$. The Prover then sends a commitment to the polynomial

$$f(X) - \texttt{Coef}(f(X)\ ,\ \deg(f)) \cdot X^{\deg(f)}$$

obtained by removing the leading term and uses the protocol for the degree upper bound (`ZKPoDegUp`) to show that this is a commitment to a polynomial of a degree *strictly* less than the integer committed in $a_{\deg}$.

This is a HVZK for the following relation:

$$\mathcal{R}_{\texttt{Deg}}[g_1,\ (a, a_{\deg})] = \left\{ \left((a_1, a_2 \in \mathbb{G}_1),\ f(X) \in \mathbb{F}_p[X] \right) : g_1^{f(s)} = a\ ,\ g_1^{\deg(f)} = a_{\deg} \right\}$$

Protocol 6.8. *Zero-knowledge proof of degree* (`ZKPoDeg`)
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$; generators $g_1,\ g_2$ for $\mathbb{G}_1,\ \mathbb{G}_2$ respectively.
Inputs: Elements $a, a_{\text{deg}} \in \mathbb{G}_1$
Claim: The Prover knows an integer n and a polynomial $f(X)$ of degree n such that

$$a = g_1^{f(s)} \ , \ \ a_{\text{deg}} = g_1^n.$$

1. The Prover $\mathcal{P}$ isolates the leading term $c_n \cdot X^n$ of $f(X)$ and sends the elements

$$a_1 := g_1^{c_n \cdot s^n} \ , \ \ a_2 := g_1^{f(s) - c_n \cdot s^n} \ \in \ \mathbb{G}_1.$$

2. $\mathcal{P}$ sends proofs for `ZKPoDegMono`$[g_1,\ (a_1,\ a_{\text{deg}})]$ and `ZKPoDegUp`$[g_1,\ (a_2,\ a_{\text{deg}})]$.
3. $\mathcal{V}$ verifies the `ZKPoDegUp`, the `ZKPoDegMono` and the equation $a \stackrel{?}{=} a_1 \cdot a_2$. □

6.3 Coefficients at hidden positions

The next protocol allow a Prover to prove succinctly and in zero-knowledge that given elements $a, a_1, a_2 \in \mathbb{G}_1$, he knows a polynomial $f(X)$, an integer $n \leq \deg(f)$ and a field element α such that X^n has coefficient α in $f(X)$ and a, a_1, a_2 are commitments to $f(X)$, n and α respectively. This is made possible by the protocols linking a committed polynomial to its degree.

The Prover decomposes the polynomial $f(X)$ as a sum

$$f(X) = f_-(X) + \alpha \cdot X^n + f_+(X) \cdot X^{n+1}$$

of three parts such that $\deg(f_-) < n$. In other words,

$$f_-(X) := f(X) \pmod{X^n} \ \ , \ \ f_+(X) := X^{-(n+1)} \cdot [f(X) - f_-(X) - \alpha \cdot X^n].$$

The homomorphic property implies that the commitment to $f(X)$ is a product of commitments to $f_-(X)$, $\alpha \cdot X^n$ and $f_+(X) \cdot X^{n+1}$.

The first part (i.e. the low degree part) can be isolated using the protocol (`ZKPoDegUp`). The third part (i.e. the high degree part) can be isolated by showing that it is a multiple of X^{n+1}, where n is the integer committed in $a_{\texttt{Pos}}$. This leaves us with a commitment to the monomial $\alpha \cdot X^n$.

The Prover can use the protocol `ZKPoDegMono` that show that this is a commitment $a_{\texttt{mid}}$ to a *monomial* such that this monomial evaluates to α at $X = 1$, where α is the field element committed in $a_{\texttt{Coef}}$

This is a HVZK for the following relation:

$$\mathcal{R}_{\texttt{Coef}}[g_1,\ (a,\ a_{\texttt{Pos}},\ a_{\texttt{Coef}})] = \left\{ \begin{array}{l} \big((a,\ a_{\texttt{Pos}},\ a_{\texttt{Coef}} \ \in \ \mathbb{G}_1)\ , \\ f(X)\ ,\ n \in \mathbb{Z}^{\geq 0}\ ,\ \alpha \in \mathbb{F}_p\big) : \\ g_1^{f(s)} = a\ ,\ g_1^n = a_{\texttt{Pos}}\ ,\ g_1^{\alpha} = a_{\texttt{Coef}} \\ \texttt{Coef}(f(X)\ ,\ n) = \alpha \end{array} \right\}$$

Protocol 6.9. *Zero-knowledge proof of polynomial coefficient* (`ZKPoCoef`)
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$; generators $g_1,\ g_2$ for $\mathbb{G}_1,\ \mathbb{G}_2$ respectively.
Inputs: Elements $a, a_{\texttt{Pos}}, a_{\texttt{Coef}} \in \mathbb{G}_1$

Claim: The Prover knows an integer n, a polynomial $f(X)$ of degree $\geq n$ and a field element α such that:

- $a = g_1^{f(s)}$, $a_{\texttt{Pos}} = g_1^n$, $a_{\texttt{Coef}} = g_1^\alpha$
- The coefficient of $f(X)$ at X^n is α.

1. $\mathcal{P}$ sends the elements
$$a_{\texttt{mid}} := g_1^{\alpha \cdot s^n} \quad , \quad a_{\texttt{mid},1} := a_{\texttt{mid}}^s = g_1^{\alpha \cdot s^{n+1}}$$
along with a proof for $\texttt{ZKPoDegMono}[a_{\texttt{mid}},\ (\widetilde{g_1},\ a_{\texttt{Pos}})]$.
2. $\mathcal{P}$ sends a proof for $\texttt{ZKPoKE}[g_1^{s^{-1}},\ a_{\texttt{mid}} \cdot a_{\texttt{Coef}}^{-1}]$.
3. The Prover $\mathcal{P}$ computes the polynomials
$$f_-(X) := f(X) \pmod{X^n} \quad , \quad f_+(X) := f(X) - f_-(X) - \alpha \cdot X^n.$$
4. $\mathcal{P}$ sends the element $a_- := g_1^{f_-(s)}$ along with a proof for $\texttt{ZKPoDegUp}[g_1,\ (a_-, a_{\texttt{Pos}})]$.
5. $\mathcal{P}$ sends the element $a_+ := g_1^{f_+(s)}$ along with a proof for $\texttt{ZKPoKE}[a_{\texttt{mid},1},\ a_+]$.
6. $\mathcal{V}$ verifies the ZKPoDegUp, the ZKPoDegMono, ZKPoKE, and the equations
$$a \stackrel{?}{=} a_- \cdot a_{\text{mid}} \cdot a_+ \quad , \quad \mathbf{e}(a_{\texttt{mid},1}\ ,\ g_2) \stackrel{?}{=} \mathbf{e}(a_{\texttt{mid}}\ ,\ g_2^s)$$

□

Proposition 6.10. *The protocol* ZKPoCoef *is secure in the algebraic group model.*

Proof. (Sketch) Suppose a PPT algorithm $\mathcal{A}$ outputs an accepting transcript.

The subprotocol $\texttt{ZKPoDegMono}[g_1,\ (a_{\texttt{mid}},\ a_{\texttt{Pos}})]$ implies that with overwhelming probability, $\mathcal{A}$ can output $\widehat{\alpha} \in \mathbb{F}_p$, $\widehat{n} \in \mathbb{Z}$ such that
$$a_{\texttt{mid}} = g_1^{\widehat{\alpha} \cdot s^{\widehat{n}}} \quad , \quad a_{\texttt{Pos}} = g_1^{\widehat{n}}.$$
The subprotocol $\texttt{ZKPoKE}[g_1^{s^{-1}},\ a_{\texttt{mid}} \cdot a_{\texttt{Pos}}^{-1}]$ implies that with overwhelming probability, $\mathcal{A}$ can output a polynomial $h(X)$ such that
$$g_1^{h(s)} = a_{\texttt{mid}} \quad , \quad g_1^{h(1)} = a_{\texttt{Coef}}.$$
Thus, with overwhelming probability, $h(X) = \alpha_0 \cdot X^{\widehat{n}}$ and $a_{\texttt{Coef}} = g_1^{\widehat{\alpha}}$.

The subprotocol $\texttt{ZKPoDegUp}[g_1,\ (a_-, a_{\texttt{Pos}})]$ implies that with overwhelming probability, $\mathcal{A}$ can output a polynomial $\widehat{f}_-(X)$ of degree $< \widehat{n}$ such that $g_1^{\widehat{f}_-(s)} = a_-$. Furthermore, the pairing check
$$\mathbf{e}(a_{\texttt{mid},1}\ ,\ g_2) \stackrel{?}{=} \mathbf{e}(a_{\texttt{mid}}\ ,\ g_2^s)$$
implies that $a_{\texttt{mid},1} = a_{\texttt{mid}}^s$ and hence, with overwhelming probability, $a_{\texttt{mid},1} = g_1^{\widehat{\alpha} \cdot s^{\widehat{n}+1}}$. The subprotocol $\texttt{ZKPoKE}[a_{\texttt{mid},1},\ a_+]$ implies that with overwhelming probability, $\mathcal{A}$ can output a polynomial $\widehat{f}_+(X)$ such that $\widehat{f}_+(X)$ is divisible by $X^{\widehat{n}+1}$ and $g_1^{\widehat{f}_+(s)} = a_+$.

Lastly, the equation $a \stackrel{?}{=} a_- \cdot a_{\text{mid}} \cdot a_+$ implies that setting
$$\widehat{f}(X) := \widehat{f}_-(X) + \widehat{\alpha} \cdot X^{\widehat{n}} + \widehat{f}_+(X)$$
yields $g_1^{\widehat{f}(s)} = a$ and $\texttt{Coef}(\widehat{f}(X)\ ,\ \widehat{n}) = \widehat{\alpha}$. □

References

[ABC+12] J. H. Ahn, D. Boneh, J. Camenisch, S. Hohenberger, A. Shelat and B. Waters. *Computing on authenticated data*. In Ronald Cramer, editor, TCC 2012, volume 7194 of LNCS, pages 1-20.Springer, Heidelberg, March 2012.

[BBF19] D. Boneh, B. Bunz, B. Fisch, *Batching Techniques for Accumulators with Applications to IOPs and Stateless Blockchains*, In Alexandra Boldyreva and Daniele Micciancio, editors, CRYPTO 2019, Part I, volume 11692 of LNCS, pages 561–586. Springer, Heidelberg, August 2019.

[BGG17] S. Bowe, A. Gabizon, M. Green, *A multi-party protocol for constructing the public parameters of the Pinocchio zk-SNARK*

[BGM17] S. Bowe, A. Gabizon, I. Myers, *Scalable Multi-party Computation for zk-SNARK Parameters in the Random Beacon Model* Cryptology ePrint Archive, Report 2017/1050, 2017. http://eprint.iacr.org/2017/1050.

[CF13] D. Catalano, D. Fiore, *Vector commitments and their applications*, In Kaoru Kurosawa and Goichiro Hanaoka, editors, PKC 2013, volume 7778 of LNCS, pages 55–72. Springer, Heidelberg, February / March 2013

[CL02] J. Camenisch and A. Lysyanskaya. *Dynamic accumulators and application to efficient revocation of anonymous credentials.* In Moti Yung, editor, CRYPTO 2002, volume 2442 of LNCS, pages 61-76. Springer, Heidelberg, August 2002.

[CPZ18] A. Chepurnoy, C. Papamanthou, Y. Zhang, *EDRAX : A Cryptocurrency with Stateless Transaction Validation*, Cryptology ePrint Archive, Report 2018/968, 2018. https://eprint.iacr.org/2018/968.

[DT08] I. Damgard, N. Triandopolous, *Supporting Non-membership Proofs with Bilinear-map Accumulators*, Cryptology ePrint Archive, Report 2008/538, 2008. http://eprint.iacr.org/2008/538.

[FVY14] C. Fromknecht, D. Velicanu, and S. Yakoubov. *A decentralized public key infrastructure with identity retention*. Cryptology ePrint Archive, Report 2014/803, 2014. http://eprint.iacr.org/2014/803.

[FST06] D. Freeman, M. Scott, E. Teske, *A taxonomy of pairing-friendly elliptic curves*

[FS87] A. Fiat, A. Shamir, *How to prove yourself: Practical solutions to identification and signature problems.* In Andrew M. Odlyzko, editor, CRYPTO'86, volume 263 of LNCS, pages 186–194. Springer, Heidelberg, August 1987

[GGM14] Christina Garman, Matthew Green, and Ian Miers. Decentralized anonymous credentials. In NDSS 2014.

[KS98] E. Kaltofen and V. Shoup. Subquadratic-time factoring of polynomials over finite fields. Mathematics of computation, 67(223):1179–1197, 1998

[KZG10] A. Kate, G. Zaverucha, and I. Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, ASIACRYPT 2010, volume 6477 of LNCS, pages 177–194. Springer, Heidelberg, December 2010.

[LM18] R. Lai, G. Malavolta, *Optimal succinct arguments via hidden order groups*, Cryptology ePrint Archive, Report 2018/705, 2018. https://eprint.iacr.org/2018/705

[Mil86] V. Miller, *Short Programs for functions on Curves*

[MGGR13a] 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, pages 397411. IEEE Computer Society Press, May 2013

[Ngu05] L. Nguyen, *Accumulators from bilinear pairings and applications*, CT-RSA, 3376:275–292, 2005

[PST13] C. Papamanthou, E. Shi and R. Tamassia, Signatures of correct computation, in: Theory of Cryptography 2013, Lecture Notes in Comput. Sci. 7785, Springer, Heidelberg (2013), 222–242.

[Sla12] D. Slamanig. Dynamic accumulator based discretionary access control for outsourced storage with unlinkable access - (short paper). In Angelos D. Keromytis, editor, FC 2012, volume 7397 of LNCS, pages 215222. Springer, Heidelberg, February / March 2012.

[STS99b] T. Sander and A. Ta-Shma. Flow control: A new approach for anonymity control in electronic cash systems. In Matthew Franklin, editor, FC'99, volume 1648 of LNCS, pages 46-61. Springer, Heidelberg, February 1999

[Tre13] E. Tremel, *Real world performance of cryptographic accumulators*

[Wes18] B. Wesolowski, *Efficient verifiable delay functions*, In Annual International Conference on the Theory and

Applications of Cryptographic Techniques, pages 379–407. Springer, 2019.

A Some preliminary protocols:

The following protocol allows a Prover to prove in zero-knowledge that given elements a, b in $\mathbb{G}_1$, he knows a constant α such that $b = a^\alpha$. This is basically Schnorr's protocol, which does not need pairings or the public parameters.

The key idea is that for polynomials $f(X)$, $h(X)$ and a randomly generated challenge $\gamma \in \mathbb{F}_p$, if the sum $f(X) \cdot \gamma + h(X)$ is a constant, then with overwhelming probability, the polynomials $f(X)$, $h(X)$ are constants as well.

Protocol A.1. *Zero-knowledge proof of constant discrete logarithm* (`ZKPoConst`)
(aka Schnorr's protocol)
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$;
Inputs: Elements $a, b \in \mathbb{G}_1$
Claim: The Prover knows a constant $\alpha \in \mathbb{F}_p$ such that $b = a^\alpha$

1. The Prover $\mathcal{P}$ chooses a random blinding scalar $\beta \in \mathbb{F}_p$ and sends $C_\beta := a^\beta$.
2. The Fiat-Shamir heuristic generates a challenge γ.
3. The Prover sends $m := \beta + \alpha \cdot \gamma$.
4. $\mathcal{V}$ accepts if and only if $m \in \mathbb{F}_p$ and $a^m = C_\beta \cdot b^\gamma$. □

In case the Prover needs to show that the exponent α is *non-zero*, he could combine this protocol with a proof that he knows the discrete logarithm α^{-1} between b and a. We next discuss the Chaum-Pedersen protocol in the bilinear pairing setting. As with Schnorr's protocol, it applies to the bilinear setting without any modifications.

Protocol A.2. *Zero-knowledge proof of equality of constant discrete logarithms*
(aka the Chaum-Pedersen protocol)
Parameters: A pairing $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$;
Inputs: Elements $a_1, b_1, a_2, b_2 \in \mathbb{G}_1$
Claim: The Prover knows a constant $\alpha \in \mathbb{F}_p$ such that $b_1 = a_1^\alpha$, $b_2 = a_2^\alpha$.

1. The Prover $\mathcal{P}$ chooses a random blinding scalar $\beta \in \mathbb{F}_p$ and sends
$$C_{\beta,1} := a_1^\beta \ , \ C_{\beta,2} := a_2^\beta$$
2. The Fiat-Shamir heuristic generates a challenge γ.
3. The Prover sends $m := \beta + \alpha \cdot \gamma$.
4. $\mathcal{V}$ accepts if and only if $m \in \mathbb{F}_p$ and the equations
$$a_1^m = C_{\beta,1} \cdot b_1^\gamma \ , \ a_2^m = C_{\beta,2} \cdot b_2^\gamma$$

hold. □

B List of Protocols:

The following is a list of the protocols in this paper and the relations that the protocols are arguments of knowledge for, in the algebraic group model.

1. $\texttt{PoKE}^*$ (*Proof of knowledge of the exponent* with base g_1 or g_2*)

$$\mathcal{R}_{\texttt{PoKE}^*}[g_2,\ a] = \{(a \in \mathbb{G}_1),\ f(X) \in \mathbb{F}_p[X]) : g_1^{f(s)} = a\}$$

2. PoKE (*Proof of knowledge of the exponent*)

$$\mathcal{R}_{\texttt{PoKE}}[a,\ b] = \{((a,b) \in \mathbb{G}_1^2),\ f(X) \in \mathbb{F}_p[X]) : a^{f(s)} = b\}$$

3. PoDer (*Proof of derivative*)

$$\mathcal{R}_{\texttt{Der}}[g_1,\ (a,\ b)] = \{(a, b \in \mathbb{G}_1),\ f(X) \in \mathbb{F}_p[X]) : g_1^{f(s)} = a\ ,\ g_1^{f'(s)} = b\}$$

4. PoGCD (*Proof of GCD*)

$$\mathcal{R}_{\texttt{GCD}}[g_1,\ (a_1,\ a_2),\ a_{1,2}] = \left\{ \begin{array}{l} \big((a_1,\ a_2,\ a_{1,2}\ \in \mathbb{G}_1) \\ f_1(X),\ f_2(X),\ f_{1,2}(X)\ \in\ \mathbb{F}_p[X]\big) : \\ g_1^{f_1(s)} = a_1\ ,\ g_1^{f_2(s)} = a_2,\ g_1^{f_{1,2}(s)} = a_{1,2} \\ \mathbf{gcd}(f_1(X),\ f_2(X)) = f_{1,2}(X) \end{array} \right\}$$

5. PoSep (*Proof of separable polynomial commitment*)

$$\mathcal{R}_{\texttt{Sep}}[g_1,\ a] = \{(a, b \in \mathbb{G}_1),\ f(X) \in \mathbb{F}_p[X]) : g_1^{f(s)} = a\ ,\ \mathbf{gcd}(f(X), f'(X)) = 1\}$$

6. PoRad (*Protocol for the radical*)

$$\mathcal{R}_{\texttt{Rad}}[g_1,\ (a,\ a_{\texttt{rad}})] = \left\{ \begin{array}{l} (a,\ a_{\texttt{rad}}\ \in\ \mathbb{G}_1),\ f(X) \in \mathbb{F}_p[X],\ c \in \mathbb{F}_p) : \\ g_1^{f(s)} = a\ ,\ g_1^{c \cdot \texttt{rad}(f(s))} = a_{\texttt{rad}} \end{array} \right\}$$

7. PoComp (*Proof of composition*)

$$\mathcal{R}_{\texttt{Comp}}[g_1,\ (a_1, a_2),\ a_{1,2}] = \left\{ \begin{array}{l} \big((a_1, a_2, a_{1,2} \in \mathbb{G}_1),\ f_1(X),\ f_2(X) \in \mathbb{F}_p[X]\big) : \\ g_1^{f_1(s)} = a_1\ ,\ g_1^{f_2(s)} = a_2\ ,\ g_1^{f_1(f_2(s))} = a_{1,2} \end{array} \right\}$$

8. ZKPoDegMono (*Proof of degree of monomial*)

$$\mathcal{R}_{\texttt{DegMono}}[g_1,\ (a, a_{\deg})] = \{(a, a_{\deg} \in \mathbb{G}_1),\ n \in \mathbb{Z}^{\geq 0},\ c \in \mathbb{F}_p) :\ g_1^{c \cdot s^n} = a\ ,\ g_1^n = a_{\deg}\}$$

9. ZKPoDegUp (*Proof of degree upper bound*)

$$\mathcal{R}_{\texttt{DegUp}}[g_1,\ (a_1, a_2)] = \{(a_1, a_2 \in \mathbb{G}_1),\ f(X) \in \mathbb{F}_p[X],\ n \in \mathbb{Z}) : g_1^{f(s)} = a_1\ ,\ g_1^n = a_2\ ,\ \deg(f) < n\}$$

10. ZKPoDeg (*Proof of degree*)

$$\mathcal{R}_{\texttt{Deg}}[g_1,\ (a, a_{\deg})] = \{(a, a_{\deg} \in \mathbb{G}_1),\ f(X) \in \mathbb{F}_p[X]) : g_1^{f(s)} = a\ ,\ g_1^{\deg(f)} = a_{\deg}\}$$

11. ZKPoCoef (*Proof of coefficient*)

$$\mathcal{R}_{\texttt{Coef}}[g_1,\ (a,\ a_{\texttt{Pos}},\ a_{\texttt{Coef}})] = \left\{ \begin{array}{l} \big((a,\ a_{\texttt{Pos}},\ a_{\texttt{Coef}}\ \in\ \mathbb{G}_1), \\ f(X)\ ,\ n \in \mathbb{Z}^{\geq 0}\ ,\ \alpha \in \mathbb{F}_p\big) : \\ g_1^{f(s)} = a\ ,\ g_1^n = a_{\texttt{Pos}}\ ,\ g_1^{\alpha} = a_{\texttt{Coef}} \\ \texttt{Coef}(f(X)\ ,\ n) = \alpha \end{array} \right\}$$