Efficient polynomial commitment schemes for multiple points and polynomials

Dan Boneh
Stanford University

Justin Drake
Ethereum Foundation

Ben Fisch
Stanford University

Ariel Gabizon
AZTEC Protocol

May 27, 2021

Abstract

We present an enhanced version of the Kate, Zaverucha and Goldberg polynomial commitment scheme [KZG10] where a single group element can be an opening proof for multiple polynomials each evaluated at a different arbitrary subset of points.

As a sample application we "plug in" this scheme into the PLONK proving system[GWC19] to obtain improved proof size and prover run time at the expense of additional verifier $\mathbb{G}_2$ operations and pairings, and additional $\mathbb{G}_2$ SRS elements.

We also present a second scheme where the proof consists of two group elements and the verifier complexity is better than previously known batched verification methods for [KZG10].

1 Introduction

Polynomial commitment schemes (PCS)[KZG10] have become a central ingredient in recent constructions of succinct arguments [MBKM19, Gab19, CHM+19, GWC19, BFS19] when one desires a "universal and updatable" setup procedure [GKM+]. They "force" a prover to answer verifier queries according to a fixed polynomial of bounded degree.

The use of a PCS typically starts by an initial prover message $\mathsf{com}(f)$ corresponding to the commitment to a polynomial f. Then, in the more straightforward use of this primitive, whenever the prover sends a value $s \in \mathbb{F}$ which is allegedly the value $f(z)$ for z known to the verifier, the prover will send a corresponding "opening proof" π that this is indeed the case. When a PCS is used in this way during a protocol execution for several polynomials and several evaluation points, prover run time and communication will increase with each of these opening proofs.

Thus, it is of interest to construct PCS where the prover overhead doesn't grow, or at least grows more slowly with the number of openings.

1.1 Previous work and our results

The notion of a PCS was introduced in the influential work of Kate, Zaverucha and Goldberg [KZG10]. They presented a pairing-based scheme where an opening proof π consists of a single $\mathbb{G}_1$ group element.

[MBKM19], who introduced the use of [KZG10] for universal and updatable SNARKs, modified the PCS of [KZG10] in the random oracle model, so that a single $\mathbb{G}_1$ element can be an opening proof for several polynomials *at the same point* $z \in \mathbb{F}$. [Gab19, CHM+19, GWC19] followed and used similar single-point multi-polynomial batching protocols.

[KZG10] give in their paper a less known version of their scheme allowing for a one $\mathbb{G}_1$ element opening proof for *one* polynomial at several evaluation points.

For the case of multiple polynomials and evaluation points, [CHM+19, GWC19] use randomized techniques for batching pairing equations to improve verification efficiency; however opening proof size and prover computation still grow linearly with the number of distinct points.

In this paper, we give two PCS for multiple evaluation points and polynomials.

- In our first scheme the opening proof is only a single $\mathbb{G}_1$ element, but verifier operations are considerably heavier than previous variants of [KZG10] when the number of distinct evaluation points is large (cf. Lemma 3.3).
- In our second scheme the opening proof is two $\mathbb{G}_1$ elements, and the verifier complexity is somewhat better than previous multipoint variants of [KZG10] (cf. Lemma 4.1).

We compare the performance of our PCS to a more straightforward batched version of the [KZG10] scheme as in [GWC19]. For simplicity, we look at the restricted case where we want to open t polynomials all with the same degree bound n, each at one *distinct* point. See Lemma 3.3 and 4.1 for the more detailed efficiency properties in the general case (where each polynomial is opened at a subset of points, and the subsets may repeat).

Table 1: Comparison of opening complexity for t polynomials on t distinct points. In prover/verifier work columns $\mathbb{G}_i$ means scalar multiplication in $\mathbb{G}_i$, $\mathbb{F}$ means addition or multiplication in $\mathbb{F}$, and **P** means pairing.

	SRS size	**prover work**	**proof length**	**verifier work**
KZG as in [GWC19]	n $\mathbb{G}_1$, 2 $\mathbb{G}_2$	$t \cdot n$ $\mathbb{G}_1$, $O(t \cdot n \log n)$ $\mathbb{F}$	t $\mathbb{G}_1$	$3t-2$ $\mathbb{G}_1$, 2 **P**
This work, ver. 1	n $\mathbb{G}_1$, $t+1$ $\mathbb{G}_2$	n $\mathbb{G}_1$, $O(t \cdot n + n \log n)$ $\mathbb{F}$	1 $\mathbb{G}_1$	$t-1$ $\mathbb{G}_1$, t^2 $\mathbb{G}_2$, $t+1$ **P**
This work, ver. 2	n $\mathbb{G}_1$, 2 $\mathbb{G}_2$	$2n$ $\mathbb{G}_1$, $O(t \cdot n + n \log n)$ $\mathbb{F}$	2 $\mathbb{G}_1$	$t+3$ $\mathbb{G}_1$, 2 **P**

Application to PLONK: The PLONK proving system [GWC19] allows generating proofs of knowledge for assignments to fan-in two arithmetic circuits with a universal and updat-

able SRS (see the paragraph on this topic in Section 2.1). Most of the prover computation involves committing to several polynomials and opening them at two distinct evaluation points. Plugging in our first PCS to PLONK allows saving in proof length and prover work related to the opening proof of the second evaluation point (we do not give full details, but all that is needed is repreating the transformation of Lemma 4.7 in [GWC19] using the PCS of Lemma 3.3 instead of the PCS used there to obtain the new result).

We compare the PLONK scheme when using the [KZG10]-based PCS in [GWC19] and the first PCS of this paper in Table 2. As in [GWC19] we present two versions of PLONK where one optimizes fast proving, and the other small proof length.

Table 2: Comparison of PLONK efficiency for fan-in two circuit with n gates.

	SRS size	**prover group exponentations**	**proof length**	verifier work
[GWC19] (fast)	n $\mathbb{G}_1$, 2 $\mathbb{G}_2$	$9n$ $\mathbb{G}_1$ exp	9 $\mathbb{G}_1$, 7 $\mathbb{F}$	18 $\mathbb{G}_1$, 2 **P**
This work (fast)	n $\mathbb{G}_1$, 3 $\mathbb{G}_2$	$8n$ $\mathbb{G}_1$ exp	8 $\mathbb{G}_1$ 7 $\mathbb{F}$	18 $\mathbb{G}_1$, 4 $\mathbb{G}_2$, 3 **P**
[GWC19] (small)	$3n$ $\mathbb{G}_1$, 2 $\mathbb{G}_2$	$11n$ $\mathbb{G}_1$ exp	7 $\mathbb{G}_1$, 7 $\mathbb{F}$	16 $\mathbb{G}_1$, 2 **P**
This work (small)	$3n$ $\mathbb{G}_1$, 3 $\mathbb{G}_2$	$10n$ $\mathbb{G}_1$ exp	6 $\mathbb{G}_1$ 7 $\mathbb{F}$	16 $\mathbb{G}_1$, 4 $\mathbb{G}_2$, 3 **P**

SHPLONK? Our second PCS does not give interesting tradeoffs for PLONK as two evaluation points are not enough for its advantages to "kick in". However, in a scenario where constraints between *more than two* evaluation points are used, e.g. [Dra], the advantages of both of our new schemes will become more prominent. Thus, the PCS of this paper encourage designing constraint systems using multiple SHifts and Permutations over Largange bases for Oecumenical Noninteractive arguments of Knowledge.

2 Preliminaries

2.1 Terminology and conventions

We assume our field $\mathbb{F}$ is of prime order. We denote by $\mathbb{F}_{<d}[X]$ the set of univariate polynomials over $\mathbb{F}$ of degree smaller than d. In expressions involving both polynomials and constants, we will write $f(X)$ instead of f for to help distinguish the two; but in contexts where it is clear f is a polynomial, we will simply write f for brevity.

We assume all algorithms described receive as an implicit parameter the security parameter λ.

Whenever we use the term "efficient", we mean an algorithm running in time $\mathsf{poly}(\lambda)$. Furthermore, we assume an "object generator" $\mathcal{O}$ that is run with input λ before all protocols, and returns all fields and groups used. Specifically, in our protocol $\mathcal{O}(\lambda) = (\mathbb{F}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e, g_1, g_2, g_t)$ where

- $\mathbb{F}$ is a prime field of super-polynomial size $r = \lambda^{\omega(1)}$.

- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t$ are all groups of size r, and e is an efficiently computable non-degenerate pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t$.
- g_1, g_2 are uniformly chosen generators such that $e(g_1, g_2) = g_t$.

We usually let the λ parameter be implicit, i.e. write $\mathbb{F}$ instead of $\mathbb{F}(\lambda)$. We write $\mathbb{G}_1$ and $\mathbb{G}_2$ additively. We use the notations $[x]_1 := x \cdot g_1$ and $[x]_2 := x \cdot g_2$.

We often denote by $[n]$ the integers $\{1, \ldots, n\}$. We use the acronym e.w.p for "except with probability"; i.e. e.w.p γ means with probability *at least* $1 - \gamma$.

Universal SRS-based public-coin protocols We describe public-coin (meaning the verifier messages are uniformly chosen) interactive protocols between a prover and verifier; when deriving results for non-interactive protocols, we implicitly assume we can get a proof length equal to the total communication of the prover, using the Fiat-Shamir transform/a random oracle. Using this reduction between interactive and non-interactive protocols, we can refer to the "proof length" of an interactive protocol.

We allow our protocols to have access to a structured reference string (SRS) that can be derived in deterministic $\mathsf{poly}(\lambda)$-time from an "SRS of monomials" of the form $\left\{[x^i]_1\right\}_{a \le i \le b}$, $\left\{[x^i]_2\right\}_{c \le i \le d}$, for uniform $x \in \mathbb{F}$, and some integers a, b, c, d with absolute value bounded by $\mathsf{poly}(\lambda)$. It then follows from Bowe et al. [BGM17] that the required SRS can be derived in a universal and updatable setup[GKM$^+$] requiring only one honest participant; in the sense that an adversary controlling all but one of the participants in the setup does not gain more than a $\mathsf{negl}(\lambda)$ advantage in its probability of producing a proof of any statement.

For notational simplicity, we sometimes use the SRS srs as an implicit parameter in protocols, and do not explicitly write it.

2.2 Analysis in the AGM model

For security analysis we will use the Algebraic Group Model of Fuchsbauer, Kiltz and Loss [FKL18]. In our protocols, by an *algebraic adversary* $\mathcal{A}$ in an SRS-based protocol we mean a $\mathsf{poly}(\lambda)$-time algorithm which satisfies the following.

- For $i \in \{1, 2\}$, whenever $\mathcal{A}$ outputs an element $A \in \mathbb{G}_i$, it also outputs a vector v over $\mathbb{F}$ such that $A =< v, \mathsf{srs}_i >$.

Idealized verifier checks for algebraic adversaries We introduce some terminology to capture the advantage of analysis in the AGM.

First we say our srs *has degree* Q if all elements of srs_i are of the form $[f(x)]_i$ for $f \in \mathbb{F}_{<Q}[X]$ and uniform $x \in \mathbb{F}$. In the following discussion let us assume we are executing a protocol with a degree Q SRS, and denote by $f_{i,j}$ the corresponding polynomial for the j'th element of srs_i.

Denote by a, b the vectors of $\mathbb{F}$-elements whose encodings in $\mathbb{G}_1, \mathbb{G}_2$ an algebraic adversary $\mathcal{A}$ outputs during a protocol execution; e.g., the j'th $\mathbb{G}_1$ element output by $\mathcal{A}$ is $[a_j]_1$.

By a "real pairing check" we mean a check of the form

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

for some matrices T_1, T_2 over $\mathbb{F}$. Note that such a check can indeed be done efficiently given the encoded elements and the pairing function $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t$.

Given such a "real pairing check", and the adversary $\mathcal{A}$ and protocol execution during which the elements were output, define the corresponding "ideal check" as follows. Since $\mathcal{A}$ is algebraic when he outputs $[a_j]_i$ he also outputs a vector v such that, from linearity, $a_j = \sum v_\ell f_{i,\ell}(x) = R_{i,j}(x)$ for $R_{i,j}(X) := \sum v_\ell f_{i,\ell}(X)$. Denote, for $i \in \{1, 2\}$ the vector of polynomials $R_i = (R_{i,j})_j$. The corresponding ideal check, checks as a polynomial identity whether

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

The following lemma is inspired by [FKL18]'s analysis of [Gro16], and tells us that for soundness analysis against algebraic adversaries it suffices to look at ideal checks. Before stating the lemma we define the Q-DLOG assumption similarly to [FKL18].

Definition 2.1. *Fix an integer Q. The* Q-DLOG assumption for $(\mathbb{G}_1, \mathbb{G}_2)$ *states that given*

$$[1]_1, [x]_1, \ldots, \left[x^Q\right]_1, [1]_2, [x]_2, \ldots, \left[x^Q\right]_2$$

for uniformly chosen $x \in \mathbb{F}$, the probability of an efficient $\mathcal{A}$ outputting x is $\mathsf{negl}(\lambda)$.

The following lemma is proved in [GWC19]-based on the arguments of [FKL18].

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

Knowlege soundness in the Algebraic Group Model We say a protocol $\mathscr{P}$ between a prover $\mathbf{P}$ and verifier $\mathbf{V}$ for a relation $\mathcal{R}$ has *Knowledge Soundness in the Algebraic Group Model* if there exists an efficient E such that the probability of any algebraic adversary $\mathcal{A}$ winning the following game is $\mathsf{negl}(\lambda)$.

1. $\mathcal{A}$ chooses input x and plays the role of $\mathbf{P}$ in $\mathscr{P}$ with input x.
2. E given access to all of $\mathcal{A}$'s messages during the protocol (including the coefficients of the linear combinations) outputs ω.
3. $\mathcal{A}$ wins if
   (a) $\mathbf{V}$ outputs acc at the end of the protocol, and
   (b) $(\mathsf{x}, \omega) \notin \mathcal{R}$.

2.3 Polynomial commitment schemes

We define polynomial commitment schemes similarly to [GWC19]. Specifically, we define the open procedure in a batched setting having multiple polynomials and evaluation points. In the context of multiple points, it will be more convenient to assume the alleged evaluations of a polynomial f on a set $S \subset \mathbb{F}$ are given as a *polynomial* $r \in \mathbb{F}_{<|S|}[X]$ with $r(z) = f(z)$ for each $z \in S$. Under this convention, the condition that the evaluations are correct; i.e. $r(z) = f(z)$ for each $z \in S$, is equivalent to $f(X) - r(X)$ being divisble by $Z_S(X)$; where $Z_S(X) := \prod_{z \in S}(X - z)$.

Definition 2.3. *A polynomial commitment scheme is a triplet* $\mathscr{S} = (\mathsf{gen}, \mathsf{com}, \mathsf{open})$ *such that*

- $\mathsf{gen}(d)$ *- is a randomized algorithm that given positive integer* d *outputs a structured reference string (SRS)* srs*.*
- $\mathsf{com}(f, \mathsf{srs})$ *- is an algorithm that given a polynomial* $f \in \mathbb{F}_{<d}[X]$ *and an output* srs *of* $\mathsf{gen}(d)$ *returns a commitment* cm *to* f*.*
- open *is a public coin protocol between parties* P_{PC} *and* V_{PC}*.* P_{PC} *is given* $f_1, \ldots, f_k \in \mathbb{F}_{<d}[X]$*.* P_{PC} *and* V_{PC} *are both given*

  1. *positive integers* $d, t = \mathsf{poly}(\lambda)$*,*
  2. $\mathsf{srs} = \mathsf{gen}(d)$*,*
  3. *a subset* $T = \{z_1, \ldots, z_t\} \subset \mathbb{F}$*,*
  4. *subsets* $S_1, \ldots, S_k \subset T$*,*
  5. $\mathsf{cm}_1, \ldots, \mathsf{cm}_k$ *- the alleged commitments to* $f_1, \ldots, f_k$*,*
  6. $\left\{r_i \in \mathbb{F}_{<|S_i|}[X]\right\}_{i \in [k]}$ *- the polynomials describing the alleged correct openings, i.e. having* $r_i(z) = f_i(z)$ *for each* $i \in [k], z \in S_i$*.*

  At the end of the protocol V_{PC} *outputs* acc *or* rej*; such that*

  - ***Completeness:*** *Fix any* $k, t = \mathsf{poly}(\lambda)$*,* $T = \{z_1, \ldots, z_t\} \subset \mathbb{F}$*,* $S_1, \ldots, S_k \subset T$*,* $f_1, \ldots, f_k \in \mathbb{F}_{<d}[X], \left\{r_i \in \mathbb{F}_{<|S_i|}[X]\right\}_{i \in [k]}$*. Suppose that for each* $i \in [k]$*,* $\mathsf{cm}_i = \mathsf{com}(f_i, \mathsf{srs})$*, and for each* $i \in [k]$ *we have* $Z_{S_i} | (f_i - r_i)$*. Then if* P_{PC} *follows* open *correctly with these values,* V_{PC} *outputs* acc *with probability one.*
  - ***Knowledge soundness in the algebraic group model:*** *There exists an efficient* E *such that for any algebraic adversary* $\mathcal{A}$ *and any choice of* $d = \mathsf{poly}(\lambda)$ *the probability of* $\mathcal{A}$ *winning the following game is* $\mathsf{negl}(\lambda)$ *over the randomness of* $\mathcal{A}$*,* V_{PC} *and* gen*.*

    1. *Given* d *and* $\mathsf{srs} = \mathsf{gen}(d)$*,* $\mathcal{A}$ *outputs* $\mathsf{cm}_1, \ldots, \mathsf{cm}_k \in \mathbb{G}_1$*.*
    2. E*, given access to the messages of* $\mathcal{A}$ *during the previous step, outputs* $f_1, \ldots, f_k \in \mathbb{F}_{<d}[X]$*.*

3. *$\mathcal{A}$ outputs* $T = \{z_1, \ldots, z_t\} \subset \mathbb{F}$, $S_1, \ldots, S_k \subset T, \left\{r_i \in \mathbb{F}_{<|S_i|}[X]\right\}_{i\in[k]}$.
4. *$\mathcal{A}$ takes the part of* P_{PC} *in the protocol* open *with the inputs* $\mathsf{cm}_1, \ldots, \mathsf{cm}_k$, $T, S_1, \ldots, S_k$, $\{r_i\}$.
5. *$\mathcal{A}$ wins if*
   - V_{PC} *outputs* acc *at the end of the protocol.*
   - *For some* $i \in [k]$, $Z_{S_i} \nmid (f_i - r_i)$.

3 Our first scheme

We first state the following straightforward claim that will allow us to efficiently "uniformize" checks on different evaluation points.

Claim 3.1. *Fix subsets* $S \subset T \subset \mathbb{F}$, *and a polynomial* $g \in \mathbb{F}_{<d}[X]$. *Then* $Z_S(X)$ *divides* $g(X)$ *if and only if* $Z_T(X)$ *divides* $Z_{T\setminus S}(X) \cdot g(X)$.

We also use the following claim, which is part of Claim 4.6 in [GWC19] where a proof of it can be found.

Claim 3.2. *Fix* $F_1, \ldots, F_k \in \mathbb{F}_{<n}[X]$. *Fix* $Z \in \mathbb{F}_{<n}[X]$ *that decomposes to distinct linear factors over* $\mathbb{F}$. *Suppose that for some* $i \in [k]$, $Z \nmid F_i$. *Then, e.w.p* $k/|\mathbb{F}|$ *over uniform* $\gamma \in \mathbb{F}$, Z *does not divide*

$$G := \sum_{j=1}^{k} \gamma^{j-1} \cdot F_j.$$

We present our first PCS.

1. $\mathsf{gen}(d)$ - choose uniform $x \in \mathbb{F}$. Output $\mathsf{srs} = \left([1]_1, [x]_1, \ldots, \left[x^{d-1}\right]_1, [1]_2, [x]_2, \ldots, \left[x^t\right]_2\right)$.
2. $\mathsf{com}(f, \mathsf{srs}) := [f(x)]_1$.
3. $\mathsf{open}\left(d, t, \{\mathsf{cm}_i\}_{i\in[k]}, T = \{z_1, \ldots, z_t\} \subset \mathbb{F}, \{S_i \subset T\}_{i\in[k]}, \{r_i\}_{i\in[k]}\right)$:
   (a) V_{PC} sends a random $\gamma \in \mathbb{F}$.
   (b) P_{PC} computes the polynomial

$$h(X) := \sum_{i\in[k]} \gamma^{i-1} \cdot \frac{f_i(X) - r_i(X)}{Z_{S_i(X)}}$$

   and using srs computes and sends $W := [h(x)]_1$.
   (c) V_{PC} computes for each $i \in [k]$, $Z_i := \left[Z_{T\setminus S_i}(x)\right]_2$.
   (d) V_{PC} computes

$$F := \prod_{i\in[k]} e\left(\gamma^{i-1} \cdot (\mathsf{cm}_i - [r_i(x)]_1), Z_i\right).$$

(e) V_{PC} outputs acc if and only if

$$F = e(W, [Z_T(x)]_2).$$

We argue knowledge soundness for the above protocol. More precisely, we argue the existence of an efficient E such that an algebraic adversary $\mathcal{A}$ can only win the KS game described in Section 2.3 w.p. $\mathsf{negl}(\lambda)$.

Let $\mathcal{A}$ be such an algebraic adversary.

$\mathcal{A}$ begins by outputting $\mathsf{cm}_1, \ldots, \mathsf{cm}_k \in \mathbb{G}_1$. Each cm_i is a linear combination $\sum_{j=0}^{d-1} a_{i,j} \left[x^j\right]_1$. E, who is given the coefficients $\{a_{i,j}\}$, simply outputs the polynomials

$$f_i(X) := \sum_{j=0}^{d-1} a_{i,j} \cdot X^j.$$

$\mathcal{A}$ now outputs $T = \{z_1, \ldots, z_t\} \subset \mathbb{F}, \{S_i \subset T\}_{i\in[k]}, \{r_i\}_{i\in[k]}$. Assume that for some $i^* \in [k],$, we have $Z_{S_{i^*}} \nmid (f_{i^*} - r_{i^*})$. We show that for any strategy of $\mathcal{A}$ from this point, $\mathrm{V}_{\mathsf{poly}}$ outputs acc w.p. $\mathsf{negl}(\lambda)$.

In the first step of open, $\mathrm{V}_{\mathsf{poly}}$ chooses a random $\gamma \in \mathbb{F}$. Let

$$f(X) := \sum_{i\in[t]} \gamma^{i-1} \cdot Z_{T\setminus S_i}(X) \cdot (f_i(X) - r_i(X)).$$

We know from Claim 3.1 that $F_{i^*} := Z_{T\setminus S_{i^*}} \cdot (f_{i^*} - r_{i^*})$ is not divisible by Z_T. Thus, using Claim 3.2, we know that e.w.p $k/|\mathbb{F}|$ over γ, f is not divisble by Z_T. Now $\mathcal{A}$ outputs $W = [H(x)]_1$ for some $H \in \mathbb{F}_{<d}[X]$. According to Lemma 2.2, it suffices to upper bound the probability that the ideal check corresponding to the real pairing check in the protocol passes. It has the form

$$f(X) \equiv H(X) Z_T(X).$$

The check passing implies that $f(X)$ is divisible by Z_T. Thus the ideal check can only pass w.p. $k/|\mathbb{F}| = \mathsf{negl}(\lambda)$ over the randomness of $\mathrm{V}_{\mathsf{poly}}$, which implies the same thing for the real check according to Lemma 2.2.

We summarize the efficiency properties of the scheme.

Lemma 3.3. *There is a PCS $\mathscr{S} = (\mathsf{gen}, \mathsf{com}, \mathsf{open})$ such that*

1. *For positive integer d, $\mathsf{srs} = \mathsf{gen}(d)$ consists of d $\mathbb{G}_1$ elements and $t+1$ $\mathbb{G}_2$ elements.*
2. *For integer $n \leq d$ and $f \in \mathbb{F}_{<n}[X]$, computing $\mathsf{com}(f, \mathsf{srs})$ requires n $\mathbb{G}_1$-exponentiations.*
3. *Given $T := (z_1, \ldots, z_t) \in \mathbb{F}^t, f_1, \ldots, f_k \in \mathbb{F}_{<d}[X], \{S_i\}_{i\in[k]}$, denote by k^* the number of distinct subsets $\{S_1^*, \ldots, S_{k^*}^*\}$ in $\{S_i\}$; and let $K := t + \sum_{i\in[k^*]} (t - |S_i^*|)$. and denote $n := \max\{\deg(f_i)\}_{i\in[k]}$. Let $\mathsf{cm}_i = \mathsf{com}(f_i)$. Then $\mathsf{open}(\{\mathsf{cm}_i\}, \{f_i\}, T, \{S_i \subset T\}, \{r_i\}, \mathsf{srs})$ requires*
   (a) *A single $\mathbb{G}_1$ element to be passed from $\mathrm{P}_{\mathsf{poly}}$ to $\mathrm{V}_{\mathsf{poly}}$.*
   (b) *At most n $\mathbb{G}_1$-exponentiations of $\mathrm{P}_{\mathsf{poly}}$.*
   (c) *$k-1$ $\mathbb{G}_1$-exponentiations, K $\mathbb{G}_2$-exponentiations and k^*+1 pairings of $\mathrm{V}_{\mathsf{poly}}$.*

4 Reducing verifier operations at the expense of proof length

We describe a variant of the scheme of Section 3 where we eliminate the verifier's $\mathbb{G}_2$ operations and reduce the number of pairings to two. This comes at the cost of an extra $\mathbb{G}_1$ element sent by the prover. Roughly speaking, while in Section 3 V_{PC} used $\mathbb{G}_2$ and pairing operations to compute the evaluation of a certain polynomial f encoded in the target group $\mathbb{G}_t$, in this protocol P_{PC} gives V_{PC} this evaluation encoded in $\mathbb{G}_1$, accompanied by a proof that it is correct. We first describe the PCS, and end the section by stating the obtained final result.

1. $\mathsf{gen}(d)$ outputs $\mathsf{srs} = \left([1]_1, [x]_1, \ldots, \left[x^{d-1}\right]_1, [1]_2, [x]_2\right)$ for a random $x \in \mathbb{F}$.
2. $\mathsf{com}(f_i) = [f_i(x)]_1$.
3. We describe the open procedure twice below. First, in a way that will be convenient for the security analysis, and later in an equivalent more concise way that also optimizes verifier operations, .e.g. moves operations from $\mathbb{G}_2$ into $\mathbb{G}_1$ when possible.

$\underline{\mathsf{open}(\{\mathsf{cm}_i\}, T, \{S_i\}, \{r_i\})}$:

1. V_{PC} sends random $\gamma \in \mathbb{F}$.
2. P_{PC} computes the polynomial

$$f(X) := \sum_{i \in [k]} \gamma^{i-1} \cdot Z_{T \setminus S_i}(X) \cdot (f_i(X) - r_i(X)).$$

   Recall that f is divisible by Z_T according to Claim 3.2, and define $h(X) := f(X)/Z_T(X)$. Using srs, P_{PC} computes and sends $W := [h(x)]_1$.
3. V_{PC} sends random $z \in \mathbb{F}$.
4. P_{PC} computes the polynomial

$$L(X) := f_z(X) - Z_T(z) \cdot h(X),$$

   where

$$f_z(X) := \sum_{i \in [k]} \gamma^{i-1} \cdot Z_{T \setminus S_i}(z) \cdot (f_i(X) - r_i(z))$$

   Note that $L(z) = f(z) - Z_T(z) \cdot h(z) = 0$, and thus $(X - z)$ divides L. P_{PC} sends $W' := \left[\frac{L(x)}{x-z}\right]_1$.
5. V_{PC} computes:

$$F := \sum_{i \in [k]} \gamma^{i-1} \cdot Z_{T \setminus S_i}(z) \cdot (\mathsf{cm}_i - [r_i(z)]_1) - Z_T(z) \cdot W$$

6. V_{PC} outputs acc if and only if

$$e(F, [1]_2) = e(W', [x - z]_2).$$

We argue knowledge soundness for the above protocol. More precisely, we argue the existence of an efficient E such that an algebraic adversary $\mathcal{A}$ can only win the KS game w.p. $\mathsf{negl}(\lambda)$. The proof begins identically to the previous one.

Let $\mathcal{A}$ be such an algebraic adversary.

$\mathcal{A}$ begins by outputting $\mathsf{cm}_1, \ldots, \mathsf{cm}_k \in \mathbb{G}_1$. Each cm_i is a linear combination $\sum_{j=0}^{d-1} a_{i,j} \left[x^j\right]_1$. E, who is given the coefficients $\{a_{i,j}\}$, simply outputs the polynomials

$$f_i(X) := \sum_{j=0}^{d-1} a_{i,j} \cdot X^j.$$

$\mathcal{A}$ now outputs $T = \{z_1, \ldots, z_t\} \subset \mathbb{F}, \{S_i \subset T\}_{i\in[k]}, \{r_i\}_{i\in[k]}$. Assume that for some $i^* \in [k]$, we have $Z_{S_{i^*}} \nmid (f_{i^*} - r_{i^*})$. We show that for any strategy of $\mathcal{A}$ from this point, $\mathrm{V}_{\mathsf{poly}}$ outputs acc w.p. $\mathsf{negl}(\lambda)$.

In the first step of open, $\mathrm{V}_{\mathsf{poly}}$ chooses a random $\gamma \in \mathbb{F}$. Let

$$f(X) := \sum_{i\in[k]} \gamma^{i-1} \cdot Z_{T\setminus S_i} \cdot (f_i(X) - r_i(X)).$$

We know from Claim 3.1 that $F_{i^*} := Z_{T\setminus S_{i^*}}(f_{i^*} - r_{i^*})$ is not divisible by Z_T. Thus, using Claim 3.2, we know that e.w.p $k/|\mathbb{F}|$ over γ, f is not divisble by Z_T. Assume we are in this case. Now $\mathcal{A}$ outputs $W = [H(x)]_1$ for some $H \in \mathbb{F}_{<d}[X]$, followed by V_{PC} sending uniform $z \in \mathbb{F}$. Since we are in the case that f is not divisble by Z_T, we know there are at most $2d$ values $z \in \mathbb{F}$ such that $f(z) = H(z) \cdot Z_T(z)$; and thus z chosen by V_{PC} is of this form only w.p. $\mathsf{negl}(\lambda)$. Assume we are in the case that z sent by V_{PC} is not of this form. P_{PC} now outputs $W' = [H'(x)]_1$ for some $H' \in \mathbb{F}_{<d}[X]$. According to Lemma 2.2, it suffices to upper bound the probability that the ideal check corresponding to the real pairing check in step 6 passes. Denoting

$$L'(X) := \sum_{i\in[k]} \gamma^{i-1} Z_{T\setminus S_i}(z) \cdot (f_i(X) - r_i(z)) - Z_T(z) \cdot H(X),$$

the ideal check has the form

$$L'(X) \equiv H'(X) \cdot (X - z),$$

and thus can pass for some $H' \in \mathbb{F}_{<d}[X]$ only if L' is divisible by $(X - z)$, which means $L'(z) = 0$. However

$$L'(z) = \sum_{i\in[k]} \gamma^{i-1} Z_{T\setminus S_i}(z) \cdot (f_i(z) - r_i(z)) - Z_T(z) \cdot H(z) = f(z) - Z_T(z) \cdot H(z),$$

and we are in the case where $f(z) \neq Z_T(z) \cdot H(z)$. In summary, the ideal check can only pass w.p. $\mathsf{negl}(\lambda)$ over the randomness of V_{PC}, which implies the same thing for the real check according to Lemma 2.2.

4.1 The open procedure, "cleaned up" and optimized

$\underline{\mathsf{open}(\{\mathsf{com}(f_i)\}, \{S_i\}, \{r_i\})}$:

1. V_{PC} sends a random challenge $\gamma \in \mathbb{F}$.
2. P_{PC} sends $W := [(f/Z_T)(x)]_1$ where
$$f := \sum_{i \in [k]} \gamma^{i-1} \cdot Z_{T \setminus S_i}(f_i - r_i).$$
3. V_{PC} sends a random evaluation point $z \in \mathbb{F}$
4. P_{PC} sends $W' := [(L(x)/(x-z)]_1$ where
$$L := \sum_{i \in [k]} \gamma^{i-1} Z_{T \setminus S_i}(z) \cdot (f_i - r_i(z)) - Z_T(z) \cdot (f/Z_T).$$
5. V_{PC} outputs acc iff $e(F + zW', [1]_2) = e(W', [x]_2)$, where
$$F := \sum_{i \in [k]} \gamma^{i-1} Z_{T \setminus S_i}(z) \cdot \mathsf{cm}_i - \left[\sum_{i \in [k]} \gamma^{i-1} Z_{T \setminus S_i}(z) r_i(z) \right]_1 - Z_T(z) W.$$

From the description and analysis we obtain

Lemma 4.1. *There is a PCS $\mathscr{S} = (\mathsf{gen}, \mathsf{com}, \mathsf{open})$ such that*

1. *For positive integer d, $\mathsf{srs} = \mathsf{gen}(d)$ consists of d $\mathbb{G}_1$ elements and 2 $\mathbb{G}_2$ elements.*
2. *For integer $n \leq d$ and $f \in \mathbb{F}_{<n}[X]$, computing $\mathsf{com}(f, \mathsf{srs})$ requires n $\mathbb{G}_1$-exponentiations.*
3. *Given $T := (z_1, \ldots, z_t) \in \mathbb{F}^t, f_1, \ldots, f_k \in \mathbb{F}_{<d}[X], \{S_i\}_{i \in [k]}$ and denote $n := \max\{\deg(f_i)\}_{i \in [k]}$. Let $\mathsf{cm}_i = \mathsf{com}(f_i)$. Then $\mathsf{open}(\{\mathsf{cm}_i\}, \{f_i\}, T, \{S_i \subset T\}, \{r_i\}, \mathsf{srs})$ requires*
   (a) *2 $\mathbb{G}_1$ elements sent from P_{PC} to V_{PC}.*
   (b) *at most $2n + 1$ $\mathbb{G}_1$-exponentiations of P_{PC}.*
   (c) *$k + 3$ $\mathbb{G}_1$-exponentiations and 2 pairings of V_{PC}.*

Acknowledgements

Part of this research was conducted while the second author was supported by Protocol Labs. We thank Zachary J. Williamson for helpful conversations. We thank Suyash Bagad for a correction.

References

[BFS19] B. Bünz, B. Fisch, and A. Szepieniec. Transparent snarks from DARK compilers. *IACR Cryptology ePrint Archive*, 2019:1229, 2019.

[BGM17] S. Bowe, A. Gabizon, and I. Miers. Scalable multi-party computation for zk-snark parameters in the random beacon model. Cryptology ePrint Archive, Report 2017/1050, 2017. `https://eprint.iacr.org/2017/1050`.

[CHM+19] A. Chiesa, Y. Hu, M. Maller, P. Mishra, N. Vesely, and N. P. Ward. Marlin: Preprocessing zksnarks with universal and updatable SRS. *IACR Cryptology ePrint Archive*, 2019:1047, 2019.

[Dra] J. Drake. *https://ethresear.ch/t/slonk-a-simple-universal-snark/6420.*

[FKL18] G. Fuchsbauer, E. Kiltz, and J. Loss. The algebraic group model and its applications. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, pages 33–62, 2018.

[Gab19] A. Gabizon. Auroralight:improved prover efficiency and SRS size in a sonic-like system. *IACR Cryptology ePrint Archive*, 2019:601, 2019.

[GKM+] J. Groth, M. Kohlweiss, M. Maller, S. Meiklejohn, and I. Miers. Updatable and universal common reference strings with applications to zk-snarks. *IACR Cryptology ePrint Archive*, 2018.

[Gro16] J. Groth. On the size of pairing-based non-interactive arguments. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 305–326, 2016.

[GWC19] A. Gabizon, Z. J. Williamson, and O. Ciobotaru. PLONK: permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. *IACR Cryptology ePrint Archive*, 2019:953, 2019.

[KZG10] A. Kate, G. M. Zaverucha, and I. Goldberg. Constant-size commitments to polynomials and their applications. *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, pages 177–194, 2010.

[MBKM19] M. Maller, S. Bowe, M. Kohlweiss, and S. Meiklejohn. Sonic: Zero-knowledge snarks from linear-size universal and updateable structured reference strings. *IACR Cryptology ePrint Archive*, 2019:99, 2019.