Zeromorph: Zero-Knowledge Multilinear-Evaluation Proofs from HomoMORPHic Univariate Commitments

Tohru Kohrita and Patrick Towa

Aztec Labs

Abstract. A multilinear polynomial is a multivariate polynomial of degree at most one in each variable. This paper introduces a new scheme to commit to multilinear polynomials and to later prove evaluations thereof. The scheme exponentially improves on a criterion of crucial relevance in practice but that is often overlooked in theoretical performance evaluations: the costs of generating prover randomness in zero-knowledge evaluation proofs.

The construction of the scheme is generic and relies only on the additive homomorphic property of any scheme to commit to univariate polynomials, and on a protocol to prove that committed polynomials satisfy public degree bounds. As the construction requires to check that several committed univariate polynomials do not exceed given, separate bounds, the paper also gives a method to batch executions of any degree-check protocol on homomorphic commitments.

For an n-linear polynomial, the instantiation of the scheme with a hiding version of KZG commitments (Kate, Zaverucha and Goldberg at Asiacrypt 2010) leads to a scheme with an evaluation prover that uses only $n + 2$ random field elements to compute zero-knowledge proofs. In contrast, previous constructions require 2^n random field elements, which is too costly to prove the satisfiability of arithmetic circuits used in practice. The instantiation does so without any concessions on the other performance measures compared to the state of the art.

1 Introduction

1.1 Context

The sum-check protocol [21] is at the heart of many an efficient proof system for arithmetic-circuit satisfiability [2, 6–8, 10, 16, 25, 26, 28–31]. Given a prime-order field $\mathbb{F}$, a public polynomial f in n variables with coefficients in $\mathbb{F}$ and a set $H \subseteq \mathbb{F}$, the sum-check protocol is a two-party interactive protocol [17] in which a prover shows to a verifier that the sum of the evaluations of f over H^n is a given public value. At the end of the protocol, the verifier must evaluate f at a random point determined by its own challenges. In this sense, the protocol reduces for the verifier the expensive task of computing the sum of the evaluations to the task of evaluating f at a random point.

In the context of arithmetic circuits, the "arithmetisation" of a circuit (e.g., R1CS/quadratic-span programs [15] or Plonk [13]) refers to the set of polynomial equations which are satisfied by the circuit-wire values if and only if the output of each circuit gate is correctly computed from its inputs. When the sum-check protocol is applied to prove the satisfiability of a circuit arithmetisation, the input f is then just a set of wire values in $\mathbb{F}$, and not a polynomial in a given number n of variables together with a domain H. It is then up to the protocol designer to specify H and a polynomial f such that the input values are interpreted as its evaluation over H.

Assume for simplicity that there are $N := 2^n$ such values for some positive integer n. When the input values are interpreted as the evaluations over $\{0, 1\}^n$ of a multilinear polynomial f in n variables, Thaler [28] showed that proving the satisfiability of an arithmetic circuit can be reduced via the sum-check protocol, and in $O(N)$ field operations (which is optimal), to proving the evaluations of the input polynomials at random points chosen by the verifier. His techniques are still prevalent in several proof systems for circuit satisfiability with the lowest prover computational costs.

However, evaluating an input polynomial f may still be too expensive in practice when N is large. An alternative is to have the prover initially commit to f with a so-called *polynomial-commitment scheme*, i.e., the prover sends a commitment to values representing f before the start of the sum-check protocol. At the end of the sum-check execution, instead of having the verifier compute the evaluation at a random point, the prover computes it and shows to the verifier that the evaluation is correct with respect to the commitment to f. If checking an evaluation proof is significantly less expensive than computing the evaluation, this results in significant computational savings for the verifier.

Moreover, if f is committed with a hiding scheme and if the evaluation of f does not reveal any information about the wire values (this generally requires to commit to a randomised version of f), then committing to f also enables circuit-satisfiability proofs that reveal no information beyond the fact that the circuit is satisfiable, i.e., *zero-knowledge* satisfiability proofs. Naturally, to achieve this property, the evaluation proofs must also reveal no information beyond the claimed evaluations. That is, for the circuit-satisfiability proof to be zero-knowledge, the evaluation proofs of committed polynomial must also be zero-knowledge.

Generating Prover Randomness in Practice. To achieve the zero-knowledge property, the specification proof systems typically assume that the prover has access to as many uniformly random bits as desired. However, entropy is in practice scarce and one often resorts to pseudo-random-number generators (like the Linux `/dev/random` or `/dev/urandom`) to extract long bit sequences from physical sources (such as keyboard presses or timing of interrupts) once enough entropy has been accumulated. In the case of circuit-satisfiability proofs, these pseudo-random numbers are used to generate elements in $\mathbb{F}$ with a distribution close to uniform. These pseudo-random field elements are then used in protocol implementations in lieu of the truly random field elements assumed in protocol specifications.

For instance, in the case of the BN254 pairing curve[1], the field size is a 254-bit prime, i.e., a prime in $\left\{2^{253}, \ldots, 2^{254}-1\right\}$. Given a pseudo-random-number generator, to generate a random integer in $\{0, \ldots, |\mathbb{F}|-1\}$, one may resort to rejection sampling methods or generate a 512-bit pseudo-random number and return its residue modulo $|\mathbb{F}|$. With rejection sampling, the statistical distance can be made arbitrarily small by increasing the maximum number of repetitions. With the second approach, the statistical distance between the output distribution and the uniform distribution over $\{0, \ldots, |\mathbb{F}|-1\}$ is at most 2^{-128}. This process is then repeated as many times as the prover algorithm requires a random field element to be generated.

Although costly when the number of random field elements is large, *these operations that must be run in practice are rarely accounted for in the performance evaluations of proof systems in the literature.* This paper follows a more practical approach and takes into consideration the costs of generating randomness in proof computations.

There exists a generic technique to turn hitherto known evaluation proofs into zero-knowledge ones, and it requires the prover to generate and commit to a degree-N masking polynomial with uniformly random coefficients, ahead of the execution of the evaluation protocol. For $N \approx 2^{20}$ as it arises in practice, generating that many random field elements becomes prohibitively expensive. No matter how efficient a proof system might be by any other measure, if privacy and zero-knowledge is a concern, a fast implementation is unattainable with this approach.

Requiring that many random field elements to achieve zero-knowledge also increases the likelihood of randomness failure amid proof computations, which would leak information about witnesses. Such an event can for instance occur when a faulty implementation produces low-entropy outputs, or when pseudo-random numbers are generated before enough entropy has been accumulated from physical sources, e.g., during booting or soon after a generator-state compromise. It means that from a practical-security standpoint, it is preferable to implement zero-knowledge proof systems which necessitates only a low amount of random field elements to be generated by the prover.

Before this paper, only the polynomial-commitment scheme derived from the zero-knowledge vSQL [33] polynomial-delegation scheme (i.e., a commitment scheme with committed evaluations) could achieve the zero-knowledge property with a sub-linear number ($\log N$) of prover random field elements. However, that scheme entails $\log N$ pairing computations for the verifier, which pales in comparison with the state of the art which requires only a constant number of verifier pairing computations. Besides, Section B.2.2 highlights a critical flaw in its proof of soundness and proposes an alternative proof.

[1] In 2023, it is still the only curve with pre-compiled contracts on Ethereum for elliptic-curve operations and pairing computation, which makes it the most practical choice for on-chain proof verification.

In other words, no polynomial-commitment scheme thus far achieves the zero-knowledge property with even a sub-linear amount of prover random field elements without any sacrifice on the other performance criteria.

1.2 Contributions

Multilinear Commitments. This paper proposes (in Section 4) a generic construction of a multilinear commitment scheme from any additively homomorphic scheme to commit to univariate polynomials, as well as any protocol to check degree bounds on committed polynomials. The construction is later instantiated with a hiding version (inspired by a polynomial-delegation scheme due to Zhang et al. [33]) of univariate KZG commitments [19], and with new techniques (in Section 5) to efficiently batch degree-check protocols on KZG commitments. The evaluation proofs of the resulting scheme satisfy the zero-knowledge property with only $n + 2$ random field elements to be generated by the prover, i.e., *exponentially fewer* than the traditional approach with a degree-N uniformly random masking polynomial. Moreover, it does so without any compromise on the other performance metrics compared to the state of the art (except for higher power of the reference-string trapdoor to be committed in the second group).

Namely, the evaluation proofs are constant-round (five), the prover sends $n+3$ first-group elements and performs at most $5N/2+1$ first-group scalar multiplications (in addition to $O(N)$ field operations with small constants). That is less group operations for the prover than in any scheme in the existing literature of pairing-based schemes, except for multilinear KZG commitments [24] in which the verifier performs n pairing computations (and which requires n trapdoors). Even a combination of optimised Gemini commitments [7] which use a single point to check consistency of folded polynomials (as in the FRI protocol [1]) with KZG and Shplonk [5] requires at least $3N$ first-group scalar multiplications for the prover; and the evaluation proofs are not even zero-knowledge unless one uses the generic method with a degree-N random hiding polynomial.

As for the verifier, it carries out $O(n)$ field operations, at most $2n + 2$ first-group scalar multiplications, a scalar multiplication and an addition in the second second group, and three pairing computations.

Batched Degree Checks. Section 5.4 shows that the techniques used to batch degree checks on KZG commitments in the algebraic-group model can also be used to batch executions of any degree-check protocol (e.g., FRI [1]) on univariate commitments that are additively homomorphic. If the degree-check and evaluation protocols are knowledge-sound in the standard model[2], then so is the batched degree-check protocol.

Shift Evaluations. The generic construction in Section 4 also leads to an efficient way to evaluate shifts of polynomials with only a commitment to the original polynomial. If $(a_0, \ldots, a_{N-1})$ is the vector of evaluations of a

[2] The FRI degree-check protocol is knowledge sound when instantiated with any commitment scheme with knowledge-sound evaluation proofs, as the prover must reveal evaluations of the committed polynomial during a FRI protocol execution.

multilinear polynomial f, the shift of f is defined by the vector of its evaluations $(a_1, \dots, a_{N-1}, a_0)$.

Evaluating polynomial shifts is necessary to compute some [12] of the so-called look-up arguments which show that the values of a committed field-element vector appear in a public table. For circuits in which proving input-output relations for sub-parts of a circuit is less expensive than evaluating the sub-circuit, these look-up arguments result in colossal computational savings when computing satisfiability proofs, and all the more so when such sub-circuits are repeated.

Section 7 leverages a simple identity which relates a polynomial to its shift. However, there is no multilinear known counterpart to this polynomial identity that can be used to evaluate a polynomial shifts with only commitments to the original polynomial. Section 7 gives a protocol to evaluate shifts of polynomials committed with the generic scheme. Furthermore, since the aforementioned look-up arguments often require evaluating polynomials and their shifts at the same point, Section 8 provides a protocol to batch evaluations of polynomials and polynomial shifts.

1.3 Key Technical Ideas

Let $\mathbb{F}[X_0, \dots, X_{n-1}]^{\preceq 1}$ denote the set of multilinear polynomials in n variables. Lemma 2.3.1 shows that a polynomial $f \in \mathbb{F}[X_0, \dots, X_{n-1}]^{\preceq 1}$ satisfies $f(\boldsymbol{u}) = v$, for $\boldsymbol{u} = (u_0, \dots, u_{n-1}) \in \mathbb{F}^n$ and $v \in \mathbb{F}$, if and only if there exist polynomials q_k in the first k variables $X_0, \dots, X_{k-1}$ (q_0 is constant) for all $k = 0, \dots, n-1$ such that

$$f - v = \sum_{k=0}^{n-1} (X_k - u_k) q_k.$$

However, instead of directly proving the existence of polynomials q_k such that this multilinear-polynomial identity is satisfied, the main idea of the generic construction is to use a linear isomorphism between the vector space of multilinear polynomials in n variables and the space $\mathbb{F}[X]^{<2^n}$ of univariate polynomial of degree at most $2^n - 1$, which arises from identifying an integer in $\{0, \dots, 2^n - 1\}$ with its binary representation in $\{0, 1\}^n$. The isomorphism used in the construction is

$$\mathcal{U}_n \colon \mathbb{F}[X_0, \dots, X_{n-1}]^{\preceq 1} \to \mathbb{F}[X]^{<2^n}$$
$$\prod_{j=0}^{n-1} (i_j \cdot X_j + (1 - i_j) \cdot (1 - X_j)) \mapsto \left(X^{2^0}\right)^{i_0} \cdots \left(X^{2^{n-1}}\right)^{i_{n-1}}.$$

For instance, in case $n = 2$, a polynomial

$$a_{00}(1 - X)(1 - Y) + a_{01}X(1 - Y) + a_{10}(1 - X)Y + a_{11}XY \in \mathbb{F}[X, Y]$$

is sent to the univariate polynomial

$$\begin{aligned} & a_{00} + a_{01}\left(Z^{2^0}\right)^1\left(Z^{2^1}\right)^0 + a_{10}\left(Z^{2^0}\right)^0\left(Z^{2^1}\right)^1 + a_{11}\left(Z^{2^0}\right)^1\left(Z^{2^1}\right)^1 \\ = & \, a_{00} + a_{01}Z + a_{10}Z^2 + a_{11}Z^3 \in \mathbb{F}[Z]. \end{aligned}$$

The construction then aims at proving the existence of polynomials q_k such that

$$\mathcal{U}_n(f) - v \cdot \mathcal{U}_n(1) = \sum_{k=0}^{n-1} \mathcal{U}_n(X_k q_k) - u_k \cdot \mathcal{U}_n(X_k).$$

To commit a polynomial f given as a vector of evaluations over $\{0,1\}^n$, the construction uses a univariate commitment scheme to commit to the image of f under $\mathcal{U}_n$. Various lemmas in Section 2 give expressions for $\mathcal{U}_n(1)$, $\mathcal{U}_n(X_k)$, and for $\mathcal{U}_n(X_k q_k)$ in terms of $\mathcal{U}_n(q_k)$. This leads to a univariate polynomial identity that involves $\mathcal{U}_n(f)$ and $\mathcal{U}_n(q_0), \ldots, \mathcal{U}_n(q_{n-1})$, which in turns leads to the protocol in Section 4. Lemma 2.5.2 shows that the coefficients of $\mathcal{U}_n(q_k)$ are 2^k-periodic, so the prover only commits to the sum of its first 2^k monomials, and proves that the underlying polynomial is of degree at most $2^k - 1$.

Section 5 introduces methods to efficiently batch such degree checks. The instantiation of the generic protocol with hiding KZG commitments in Section 6 then uses these batching techniques to improve the efficiency of the protocol.

The protocol (in Section 7) to evaluate the shift $f_{\leftarrow}$ of a polynomial $f := (a_0, \ldots, a_{N-1})$, given a commitment to f (which is a univariate commitment) is based on the observation that $X \cdot \mathcal{U}_n(f_{\leftarrow}) = \mathcal{U}_n(f) - a_0 + a_0 X^N$.

1.4 Related Work

The evaluation proofs of the straightforward adaption of KZG commitments to multilinear polynomials [24] also consist in proving knowledge of a form of quotients of multi-variate Euclidian division. However, that scheme requires $\log N$ trapdoors and the verifier must do $\log N + 1$ pairing computations.

The multilinear commitment scheme used in the Virgo proof system [32] is the first one to exploit univariate commitments and a one-to-one correspondence between multilinear polynomials and univariate polynomials. It then applies the Aurora [3] univariate-sum-check argument to prove evaluations. The prover has to commit to three univariate polynomials and later prove evaluations thereof, and the verifier must evaluate a circuit of size $\Omega(N \log N)$ and depth $\Omega(\log N)$. The verifier can delegate this computation to the prover via the GKR protocol [16] and only perform $O\left(\log^2 N\right)$ field operations (in addition to verifying the three univariate evaluations), but the prover then has to carry out $\Omega(N \log N)$ field operations, and the number of rounds increases to $\Omega\left(\log^2 N\right)$.

Gemini commitments [7] also use a one-to-one correspondence which, in case $n = 2$, sends a multilinear polynomial $f = a_{00} + a_{01}X + a_{10}Y + a_{11}XY \in \mathbb{F}[X, Y]$

to $a_{00} + a_{01}Z + a_{10}Z^2 + a_{11}Z^3 \in \mathbb{F}[Z]$. Since

$$\begin{aligned} v = f(u_0, u_1) &= (a_{00} + a_{10}u_1) + u_0 (a_{01} + a_{11}u_1) \\ &= [(a_{00} + a_{10}Y)(u_1) + X (a_{01} + a_{11}Y)(u_1)](u_0) \\ &= (f(u_0, Y))(u_1), \end{aligned}$$

The prover sends a commitment to $f(u_0, Z) = (a_{00} + a_{10}Z) + u_0 (a_{01} + a_{11}Z)$, the verifier checks its consistency with respect to the commitment to the image of f under the isomorphism, and given the commitment to $f(u_0, Z)$, the verifier uses a similar consistency check to ensure that $f(u_0, Z)(u_1) = f(u_0, u_1) = v$. The instantiation of the scheme with hiding[3]KZG commitments together with Shplonk evaluation-batching techniques lead to a scheme with an efficiency similar to that of the protocol in Section 6 (even though the prover in the latter scheme performs $N/2$ less first-group scalar multiplications, which matters in practical implementations with large values of N). The evaluation protocol is however not zero-knowledge unless one applies the standard technique of first committing to a masking polynomial with N random coefficients, i.e., exponentially more random nonces than the scheme in Section 6.

Bulletproofs [9] and Dory [20] have the advantage of being transparent, but verifying evaluation proofs entails $\Omega(N)$ scalar multiplication for Bulletproofs (they do not use pairings) and $\Omega(\log N)$ operations in the target group for Dory. Evaluation proofs are also not zero-knowledge unless one applies the standard technique with N random field elements.

Table 1 gives a comparison of these schemes with the instantiation from Section 6.

	Proof Size	P	V	Rounds	Rand. ($\mathbb{F}$)	Transparency
Mult. KZG [24]	$(n+1)\mathbb{G}_1$	$2N\mathbb{G}_1 + O(N)\mathbb{F}$	$n \cdot e(\cdot,\cdot)$	3	N	✘
Virgo [32] + [33]	$O(1)\mathbb{G}_1 + \Omega(nN)\mathbb{F}$	$\Omega(N)\mathbb{G}_1 + \Omega(nN)\mathbb{F}$	$O(1)e(\cdot,\cdot) + \Omega(n^2)\mathbb{F}$	$\Omega(n^2)$	$\geq 2N$	✘
Gemini [7] + [33] + [5]	$\geq (n+4)\mathbb{G}_1 + (n+1)\mathbb{F}$	$\geq 3N\mathbb{G}_1 + O(N)\mathbb{F}$	$(2n+2)\mathbb{G}_1 + 2\mathbb{G}_2 + 3e(\cdot,\cdot)$	5	N	✘
Bulletproofs [9]	$> 2n\mathbb{G}_1$	$\Omega(N)\mathbb{G}_1 + O(N)\mathbb{F}$	$\Omega(N)\mathbb{G}_1$	$\Omega(n)$	N	✓
Dory [20]	$\Omega(n)\mathbb{G}_1$	$\Omega(\sqrt{N})\, e(\cdot,\cdot) + O(N)\mathbb{F}$	$\Omega(n)\mathbb{G}_T$	$\Omega(n)$	N	✓
Section 6	$(n+3)\mathbb{G}_1$	$(5N/2+1)\mathbb{G}_1 + O(N)\mathbb{F}$	$(2n+2)\mathbb{G}_1 + 2\mathbb{G}_2 + 3e(\cdot,\cdot)$	5	$n+2$	✘

Table 1. Comparison of various multi-linear commitment schemes to the instantiation in Section 6. It is assumed that except for the scheme in Section 6 and Virgo (for which the authors propose a zero-knowledge version of their scheme), to prove in zero-knowledge that a committed n-linear ($N := 2^n$) polynomial f is such that $f(\boldsymbol{u}) = v_f$, the schemes first let the prover commit to a polynomial g with N uniformly random coefficients and send $v_g := g(\boldsymbol{u})$, the verifier sends a random challenge $x \in \mathbb{F}$, and they run the original schemes on $xf + g$ and $xv_f + v_g$. The second-to-last column shows the amount of random field elements the prover must generate in each scheme.

[3] For the commitments to leak no information about the opening which usually consists of circuit-wire values, though the claims also hold for standard KZG commitments.

Concerning shift evaluations, HyperPlonk [10] treats shifts only as part of a multivariate version of Plookup [12] whereas Section 7 gives a stand-alone protocol which can be applied in other contexts (e.g., permutation arguments [13]). However, Lemma 3.9 in the HyperPlonk paper readily leads to a stand-alone protocol for shift evaluations. It requires two multilinear evaluations of the committed polynomial, and is therefore less efficient than the scheme is Section 7. The scheme in Section 8 also shows how to batch shift evaluations (and even with standard evaluations), which the construction in HyperPlonk does not achieve. HyperPlonk also left as an open problem the task of proving evaluations of shifts of degree greater than one, and the method underpinning the construction in Section 7 readily leads to such proofs.

2 Mathematical Preliminaries

2.1 Notation

Fields and Vectors. Throughout this document, $\mathbb{F}$ denotes a field of prime order p. Vectors are denoted in bold font. For any $n \in \mathbb{N}_{\geq 1}$, unless explicitly stated otherwise, the elements of a vector of size n are always labelled from 0 to $n-1$. For any $1 \leq i \leq n$, $\boldsymbol{a}_{<i}$ denotes $(a_0, \ldots, a_{i-1})$. If $n \geq 3$, for any $0 \leq i < j < n$, $\boldsymbol{a}_{]i;j[}$ denotes $(a_{i+1}, \ldots, a_{j-1})$. Vectors $\boldsymbol{a}_{[i;j]}$, $\boldsymbol{a}_{[i;j[}$ and $\boldsymbol{a}_{]i;j]}$ are defined in a similar way.

Polynomials. For $n \in \mathbb{N}_{\geq 1}$ and $d \in \mathbb{N}$, $\mathbb{F}[X_0, \ldots, X_{n-1}]^{\preceq d}$ denotes the set of n-variate polynomials with coefficients in $\mathbb{F}$ and of individual degree at most d. In particular, $\mathbb{F}[X_0, \ldots, X_{n-1}]^{\preceq 1}$ stands for the set of *multilinear polynomials.*

Given integers $0 < k \leq n$ and a univariate polynomial $f = \sum_{i=0}^{n} a_i X^i$, $f^{<k}$ denotes the polynomial $\sum_{i=0}^{k-1} a_i X^i$.

Definition 2.1.1. For any $n \in \mathbb{N}$, let $\Phi_n(X) \coloneqq \sum_{i=0}^{2^n-1} X^i$.

Note that $(X-1)\Phi_n(X) = X^{2^n} - 1$, which implies that evaluations of Φ_n can be computed in $n+1$ multiplications and 2 additions.

2.2 Conditional Probabilities

The following lemma, proved in Appendix A.1, is used in the analysis of various protocols presented below.

Lemma 2.1. *Let n be a positive integer and $E_0, \ldots, E_{n-1}$, $H_0, \ldots, H_{n-1}$ denote probability events in a discrete probability space. Suppose that $P[E_0 \cup \cdots \cup E_{n-1}] > 0$. Then,*

$$P[H_0 \cap \cdots \cap H_{n-1} | E_0 \cup \cdots \cup E_{n-1}] \leq \sum_{i\colon P[E_i] \neq 0} P[H_i | E_i].$$

2.3 Polynomial Identities

The following lemma gives the multilinear-polynomial identity that all the evaluation protocols in this paper aim to verify.

Lemma 2.3.1. *Let n be a positive integer. Consider $\boldsymbol{u} \in \mathbb{F}^n$ and $v \in \mathbb{F}$. A polynomial $f \in \mathbb{F}[X_0, \ldots, X_{n-1}]^{\preceq 1}$ satisfies $f(\boldsymbol{u}) = v$ if and only if there exist $q_k \in \mathbb{F}[X_0, \ldots, X_{k-1}]^{\preceq 1}$ for all $0 < k < n$ and $q_0 \in \mathbb{F}$ such that*

$$f - v = \sum_{k=0}^{n-1} (X_k - u_k) q_k.$$

Moreover,

$$q_k = f\left(\boldsymbol{X}_{<k}, u_k + 1, \boldsymbol{u}_{]k;n[}\right) - f\left(\boldsymbol{X}_{<k}, \boldsymbol{u}_{[k;n[}\right).$$

Proof. If there exist polynomials $q_k \in \mathbb{F}[X_0, \ldots, X_{n-1}]$ (and a fortiori if $q_k \in \mathbb{F}[X_0, \ldots, X_{k-1}]^{\preceq 1}$) such that $f - v = \sum_{k=0}^{n-1} (X_k - u_k) q_k$, then $f(\boldsymbol{u}) - v = 0$ necessarily.

The converse can be proved by induction on n as follows. If $n = 1$, then there exists $q_0 \in \mathbb{F}$ such that $f - v = (X - u_0) q_0$ by univariate Euclidian division. If $n > 1$, assume the statement to be true for all integers less than n. The surjective map

$$\mathbb{F}[X_0, \ldots, X_{n-1}] \to \mathbb{F}[X_0, \ldots, X_{n-1}]/(X_{n-1} - u_{n-1})$$

sends $f - f(X_0, \ldots, X_{n-2}, u_{n-1})$ to 0, i.e., it is in the ideal $(X_{n-1} - u_{n-1})$, or equivalently, there exists a polynomial $q_{n-1} \in \mathbb{F}[X_0, \ldots, X_{n-1}]$ such that $f - f(X_0, \ldots, X_{n-2}, u_{n-1}) = (X_{n-1} - u_{n-1}) q_{n-1}$. Polynomial q_{n-1} is the quotient of the Euclidian division of $f - f(X_0, \ldots, X_{n-2}, u_{n-1})$ by $(X_{n-1} - u_{n-1})$ in the ring $\mathbb{F}[X_0, \ldots, X_{n-2}][X_{n-1}]$ (the division successfully terminates because $X_{n-1} - u_{n-1}$ is monic). Besides, as the polynomial $f - f(X_0, \ldots, X_{n-2}, u_{n-1})$ is of degree at most 1 in each variable, q_{n-1} must be of degree at most 1 in $X_0, \ldots, X_{n-2}$, and 0 in X_{n-1}, i.e., $q_{n-1} \in \mathbb{F}[X_0, \ldots, X_{n-2}]^{\preceq 1}$. Therefore, $f = (X_{n-1} - u_{n-1}) q_{n-1} + f(X_0, \ldots, X_{n-2}, u_{n-1})$.

By induction hypothesis, since the evaluation of $f(X_0, \ldots, X_{n-2}, u_{n-1})$ at $(u_0, \ldots, u_{n-2})$ is v, there exist $q_0 \in \mathbb{F}$ and $q_k \in \mathbb{F}[X_0, \ldots, X_{k-1}]^{\preceq 1}$ for all $k < n-1$ such that

$$f(X_0, \ldots, X_{n-2}, u_{n-1}) - v = \sum_{k=0}^{n-2} (X_k - u_k) q_k.$$

It follows that $f - v = \sum_{k=0}^{n-1} (X_k - u_k) q_k$.

It remains to show that $q_k = f\left(\boldsymbol{X}_{<k}, u_k + 1, \boldsymbol{u}_{]k;n[}\right) - f\left(\boldsymbol{X}_{<k}, \boldsymbol{u}_{[k;n[}\right)$. Note that

$$f\left(X_1, \ldots, X_{k-1}, u_k + 1, u_{k+1}, \ldots, u_{n-1}\right) = \sum_{j=1}^{k-1} (X_j - z_j) q_j + q_k$$

as $q_k \in \mathbb{F}[X_0, \dots, X_{k-1}]$, and that

$$f(X_1, \dots, X_{k-1}, u_k, u_{k+1}, \dots, u_{n-1}) = \sum_{j=1}^{k-1} (X_j - z_j) q_j.$$

These two equalities yield the result. $\square$

2.4 Polynomial Interpolation

This section recalls the definition and properties of Lagrange interpolation for univariate and multilinear polynomials.

2.4.1 Univariate Polynomials. Let A a subset of $\mathbb{F}$ of cardinality at least 2. For any $a \in A$, the *Lagrange interpolation polynomial* at a is defined as

$$L_{a,A}(X) \coloneqq \prod_{b \in A \setminus \{a\}} (X - b) \cdot (a - b)^{-1}.$$

$L_{a,A}$ is the unique polynomial of degree at most $|A| - 1$ such that for any $b \in A$, $L_{a,A}(b) = 1$ if $b = a$ and 0 otherwise.

2.4.2 Multilinear Polynomials. Let n be a positive integer. For all $\boldsymbol{i} =: (i_0, \dots, i_{n-1}) \in \{0,1\}^n$, let

$$\begin{aligned} L_{\boldsymbol{i}} = L_{\boldsymbol{i}}(X_0, \dots, X_{n-1}) &\coloneqq L_{i_0,\{0,1\}}(X_0) \cdots L_{i_{n-1},\{0,1\}}(X_{n-1}) \\ &= \prod_{j=0}^{n-1} \left(i_j \cdot X_j + (1 - i_j) \cdot (1 - X_j) \right). \end{aligned}$$

For all $\boldsymbol{i} \in \{0,1\}^n$, $L_{\boldsymbol{i}}$ is the unique multilinear polynomial that evaluates to 1 at $\boldsymbol{i}$ and 0 at any other point on $\{0,1\}^n$. The polynomial family $(L_{\boldsymbol{i}})_{\boldsymbol{i} \in \{0,1\}^n}$ constitute the Lagrange basis of multilinear polynomials over the boolean hypercube.

Lemma 2.4.1. *For any multilinear polynomial f, $\sum_{\boldsymbol{i} \in \{0,1\}^n} f(\boldsymbol{i}) \cdot L_{\boldsymbol{i}} = f$.*

Corollary 2.4.1.1. $\sum_{\boldsymbol{i} \in \{0,1\}^n} L_{\boldsymbol{i}} = 1$.

2.5 Multilinear-to-Univariate Correspondence

Motivation and Definition. As mentioned in Section 2.3, the polynomial-evaluation protocols to follow are based on the fact that a multilinear polynomial f in n variables satisfies $f(\boldsymbol{u}) = v$, for $\boldsymbol{u} \in \mathbb{F}^n$ and $v \in \mathbb{F}$, if and only if there exist $q_0 \in \mathbb{F}$ and polynomials q_k in $\mathbb{F}[X_0, \dots, X_{k-1}]^{\preceq 1}$ for all $k > 0$ such that $f - v = \sum_{k=0}^{n-1} (X_k - u_k) q_k$.

However, instead of testing this identity over multi-variate polynomials, the idea in these protocols is to test the identity over univariate polynomials of degree at most $2^n - 1$ by leveraging the isomorphism of $\mathbb{F}$-vector spaces

$$\begin{aligned}\mathcal{U}_n \colon \mathbb{F}[X_0, \ldots, X_{n-1}]^{\preceq 1} &\to \mathbb{F}[X]^{<2^n} \\ L_{\boldsymbol{i}} &\mapsto \left(X^{2^0}\right)^{i_0} \cdots \left(X^{2^{n-1}}\right)^{i_{n-1}},\end{aligned}$$

which stems from identifying an integer in $\{0, \ldots, 2^n - 1\}$ with its binary representation in $\{0,1\}^n$. The image of a multilinear polynomial under $\mathcal{U}_n$ is later referred to as its "*univariatisation*".

Since $\mathcal{U}_n$ is a linear isomorphism, $f - v = \sum_{k=0}^{n-1}(X_k - u_k)q_k$ if and only if $\mathcal{U}_n(f) - \mathcal{U}_n(v) = \sum_{k=0}^{n-1} \mathcal{U}_n(q_k X_k) - u_k \mathcal{U}_n(q_k)$. To design a test over univariate polynomials, it is then necessary to give more explicit expressions of $\mathcal{U}_n(v)$, $\mathcal{U}_n(q_k)$ and $\mathcal{U}_n(q_k X_k)$, i.e., to study the image under $\mathcal{U}_n$ of $\mathbb{F}$, and for all $0 < k < n$, $\mathbb{F}[X_0, \ldots, X_{k-1}]^{\preceq 1}$ and $X_k \cdot \mathbb{F}[X_0, \ldots, X_{k-1}]^{\preceq 1}$.

Properties. The first lemma below shows that $\mathcal{U}_n(\mathbb{F})$ is the line generated by Φ_n.

Lemma 2.5.1. *Let n be a positive integer. For any constant polynomial $a \in \mathbb{F}$, $\mathcal{U}_n(a)(X) = a \sum_{i=0}^{2^n-1} X^i = a \cdot \Phi_n(X)$.*

Proof. Corollary 2.4.1.1 implies that $a = a \cdot 1 = a \sum_{\boldsymbol{i} \in \{0,1\}^n} L_{\boldsymbol{i}}$, and the result follows by linearity of $\mathcal{U}_n$. □

The next lemma characterises the image under $\mathcal{U}_n$ of the space of multilinear polynomials in $k \leq n$ variables.

Lemma 2.5.2. *Let n be a positive integer. Consider $\hat{f} \in \mathbb{F}[X]^{\leq 2^n - 1}$ and let $f := \mathcal{U}_n^{-1}\left(\hat{f}\right)$. Then, for any $0 < k \leq n$, $f \in \mathbb{F}[X_0, \ldots, X_{k-1}]^{\preceq 1}$ if and only if $\hat{f}(X) = \Phi_{n-k}\left(X^{2^k}\right)\hat{f}^{<2^k}$. Furthermore, $\hat{f}^{<2^k} = \mathcal{U}_k(f)$.*

Proof. Suppose that $f \in \mathbb{F}[X_0, \ldots, X_{k-1}]^{\preceq 1}$. For any $\boldsymbol{i} \in \{0,1\}^k$,

$$L_{\boldsymbol{i}}(\boldsymbol{X}_{<k}) = L_{\boldsymbol{i}}(\boldsymbol{X}_{<k}) \cdot 1 \overset{2.4.1.1}{=} L_{\boldsymbol{i}}(\boldsymbol{X}_{<k}) \sum_{\boldsymbol{j} \in \{0,1\}^{n-k}} L_{\boldsymbol{j}}(\boldsymbol{X}_{[k;n[}).$$

Then,

$$f = \sum_{\boldsymbol{i} \in \{0,1\}^k} f(\boldsymbol{i}) L_{\boldsymbol{i}}(\boldsymbol{X}_{<k}) = \sum_{\boldsymbol{i},\boldsymbol{j}} f(\boldsymbol{i}) L_{\boldsymbol{i}}(\boldsymbol{X}_{<k}) L_{\boldsymbol{j}}\left(\boldsymbol{X}_{[k;n[}\right),$$

which implies that

$$\begin{aligned}
\hat{f} &= \sum_{\boldsymbol{i},\boldsymbol{j}} f(\boldsymbol{i}) \left(X^{2^0}\right)^{i_0} \cdots \left(X^{2^{k-1}}\right)^{i_{k-1}} \left(X^{2^k}\right)^{j_k} \cdots \left(X^{2^{n-1}}\right)^{j_{n-1}} \\
&= \sum_{\boldsymbol{j}} \left(X^{2^k}\right)^{j_k} \cdots \left(X^{2^{n-1}}\right)^{j_{n-1}} \cdot \underbrace{\sum_{\boldsymbol{i}} f(\boldsymbol{i}) \left(X^{2^0}\right)^{i_0} \cdots \left(X^{2^{k-1}}\right)^{i_{k-1}}}_{\hat{g}(X)} \\
&= \left(1 + X^{2^k} + \left(X^{2^k}\right)^2 + \cdots + \left(X^{2^k}\right)^{2^{n-k}-1}\right) \cdot \hat{g}(X) \\
&= \Phi_{n-k}\left(X^{2^k}\right) \cdot \hat{g}(X).
\end{aligned}$$

The definition of $\hat{g}$ shows that it is the image of f under $\mathcal{U}_k$, and the last equality shows that it is indeed $\hat{f}^{<2^k}$.

What precedes shows that

$$\mathcal{U}_n\left(\mathbb{F}[X_0, \ldots, X_{k-1}]^{\preceq 1}\right) \subseteq \Phi_{n-k}\left(X^{2^k}\right) \cdot \mathbb{F}[X]^{\leq 2^k - 1}.$$

To show that $f \in \mathbb{F}[X_0, \ldots X_{k-1}]^{\preceq 1}$ if $\hat{f} = \Phi_{n-k}\left(X^{2^k}\right) \hat{f}^{<2^k}$, it suffices to show that the above inclusion is in fact an equality. This is however immediate since the two are $\mathbb{F}$ vector spaces of the same dimension 2^k. □

Next comes a polynomial identity which characterises the image under $\mathcal{U}_n$ of $X_k \cdot \mathbb{F}[X_0, \ldots, X_{k-1}]^{\preceq 1}$ for all $k < n$.

Lemma 2.5.3. *Consider integers $0 < k < n$ as well as a polynomial $f \in \mathbb{F}[X_0, \ldots, X_{k-1}]^{\preceq 1}$. Then, $\left(X^{2^k} + 1\right) \mathcal{U}_n(X_k f) = X^{2^k} \mathcal{U}_n(f)$.*

Proof. As in the proof of Lemma 2.5.2, write

$$f = \sum_{\boldsymbol{i} \in \{0,1\}^k} f(\boldsymbol{i}) L_{\boldsymbol{i}}(\boldsymbol{X}_{<k}).$$

Then, denoting $\mathcal{U}_n(f)$ by $\hat{f}$,

$$\begin{aligned}
\mathcal{U}_n(X_k f) &= \sum_{\boldsymbol{j} \in \{0,1\}^{n-k-1}} X^{2^k} \left(X^{2^{k+1}}\right)^{j_{k+1}} \cdots \left(X^{2^{n-1}}\right)^{j_{n-1}} \hat{f}^{<2^k}(X) \\
&= \hat{f}^{<2^k}(X) \sum_{j=0}^{2^{n-k-1}-1} \left(X^{2^k}\right)^{2j+1}.
\end{aligned}$$

However,

$$\left(Y + Y^3 + Y^5 + \cdots + Y^{2^\ell - 1}\right)(Y + 1) = Y + Y^2 + \cdots + Y^{2^\ell} = Y \cdot \Phi_\ell(Y).$$

The change of variable $Y \leftarrow X^{2^k}$ and $\ell \leftarrow n-k$ implies that

$$\begin{aligned}\left(X^{2^k}+1\right)\mathcal{U}_n(X_k f) &= X^{2^k}\Phi_{n-k}\left(X^{2^k}\right)\cdot \hat{f}^{<2^k}(X)\\ &= X^{2^k}\mathcal{U}_n(f),\end{aligned}$$

with the last equality stemming from Lemma 2.5.2. □

Corollary 2.5.3.1. *Let $0<k<n$ be integers. Then,*

$$\left(X^{2^k}+1\right)(X-1)\mathcal{U}_n(X_k) = X^{2^k}\left(X^{2^n}-1\right).$$

Proof. Apply Lemma 2.5.3 to $f=1$. Lemma 2.5.1 shows that that $\mathcal{U}_n(1) = \Phi_n(X)$, and since $(X-1)\Phi_n(X) = X^{2^n}-1$, the result follows. □

Corollary 2.5.3.2. *Consider integers $0<k<n$ as well as a polynomial $f \in \mathbb{F}[X_0,\ldots,X_{k-1}]^{\preceq 1}$. Then, $\mathcal{U}_n(X_k f) = X^{2^k}\Phi_{n-k-1}\left(X^{2^{k+1}}\right)\mathcal{U}_n(f)^{<2^k}$.*

Proof. Lemma 2.5.3 shows that

$$\left(X^{2^k}+1\right)\mathcal{U}_n(X_k f) = X^{2^k}\mathcal{U}_n(f),$$

and Lemma 2.5.2 implies that

$$\left(X^{2^k}+1\right)\mathcal{U}_n(X_k f) = X^{2^k}\Phi_{n-k}\left(X^{2^k}\right)\mathcal{U}_n(f)^{<2^k}.$$

Setting $Y \leftarrow X^{2^k}$ and multiplying both sides of the equality by $(Y-1)$,

$$\left(Y^2-1\right)\mathcal{U}_n(X_k f) = Y(Y-1)\Phi_{n-k}(Y)\,\mathcal{U}_n(f)^{<2^k}.$$

However,

$$(Y-1)\Phi_{n-k}(Y) = Y^{2^{n-k}}-1 = \left(Y^2\right)^{2^{n-k-1}}-1 = \left(Y^2-1\right)\Phi_{n-k-1}\left(Y^2\right),$$

and since $\mathbb{F}[X]$ is an integral ring, the statement follows. □

3 Cryptographic Preliminaries

3.1 Notation and Convention

All algorithms are assumed to return a special error symbol $\perp$ whenever they are run on an input not in their defined input sets. An algorithm is termed "efficient" if its runtime is a polynomial function of its input size. Probabilistic algorithms which run in Polynomial Time are referred to as PPT algorithms. Given a PPT algorithm A and bit strings x and r, $\mathrm{A}(x;r)$ denotes the output of A on input x and random string r.

Given a binary relation R in the complexity class NP, L_R denotes the corresponding language, i.e., $L_R \coloneqq \{x \colon \exists w, (x,w) \in R\}$. For a pair $(x,w) \in R$, x is referred to as an instance and w as witness for the membership of x in L_R. A relation generator is a PPT algorithm that returns, on the input of a security parameter, a binary relation decidable in polynomial time.

3.2 Bilinear-Group Structures

An asymmetric bilinear group structure consists of a tuple $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$, with p a prime integer, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ groups of order p, and $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \mapsto \mathbb{G}_T$ a non-degenerate bilinear map. Given generators of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ sampled uniformly at random, these are respectively denoted $[1]_1$, $[1]_2$ and $[1]_T$, and for any $x \in \mathbb{F}$, $[x]_i$ is defined as $x \cdot [1]_i$ for all $i \in \{1, 2, T\}$.

A bilinear group structure is of type 3 if there is no efficiently computable homomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$. All pairings considered herein are of type 3.

3.2.1 Hardness Assumptions.

The constructions to come rely on the following computational assumption.

q-Discrete–Logarithm Assumption. The discrete-logarithm assumption, parametrised by a positive integer q (i.e., the q-DLOG assumption [11]), over a generator Gen of bilinear-group structures, is that for any PPT algorithm A,

$$P\left[y = x \colon \begin{array}{c} \mathbb{G} \coloneqq (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \textsc{Gen}\left(1^\lambda\right); x \leftarrow_\$ \mathbb{F}^* \\ y \leftarrow \mathrm{A}\left(\mathbb{G}, [1]_1, [x]_1, \ldots, [x^q]_1, [1]_2, [x]_2, \ldots, [x^q]_2\right) \end{array}\right]$$

is a negligible function of λ.

3.3 Algebraic-Group Model

The security analysis of the schemes herein (except for the generic constructions in Sections 4 and 5.4) is restricted to algebraic adversaries. That is, if a scheme specifies in its parameter a group and elements which belong to it, the analysis only applies to adversaries which can compute new group elements only by composing, with the group law, elements they receive as input and from the security-game challenger. This idea was formalised by Paillier and Vergnaud [23], and by Fuchsbauer, Kiltz and Loss [11] in their algebraic-group model. It essentially amounts to requiring that whenever an adversary outputs a group element, it also outputs its representation as a linear combination of the group elements it has thus far received.

As any analysis and conclusion in this model is restricted to the class of algebraic adversaries, a security reduction in the algebraic-group model may seem unconvincing since real-world adversaries could in principle perform a wider range of attacks. At the very least, it is not unlikely that a real-world adversary could have access to group elements other than those it is provided in a reduction to a computational problem. This situation could for instance occur if two parties use the same elliptic curve and independently generate group elements: the algebraic reduction for a scheme built by one party would not take into account the group elements generated by the other.

Rather than a "proof" of security under a computational assumption, a reduction in the algebraic-group model should rather be interpreted as a formalisation of the idea that, if the underlying computational problem is hard, algebraic attacks as allowed in the model are definitely not a threat to the scheme, and that only a different class of attacks could potentially put the scheme in jeopardy. On the other hand, for existing schemes, no practical attack exploiting other information that the group elements relevant to the schemes are currently known, so the algebraic-group model captures the range of attacks hitherto known in practice, and there is no reason for the situation to be different for schemes based on similar computational assumptions.

3.4 Proof Systems

A proof system for an NP-relation generator R consists of a set-up algorithm $\textsc{Setup}(R) \rightarrow (par, \tau)$ that returns public parameters and a trapdoor (which may be an empty string) on the input of a relation $R \leftarrow \mathrm{R}\left(1^{\lambda}\right)$, and of a pair

$$\langle \mathrm{P}(par, x, w), \mathrm{V}(par, x) \rangle =: \langle \mathrm{P}, \mathrm{V} \rangle (par, x; w)$$

of interactive algorithms. The transcript of a protocol execution is later denoted $\{\langle \mathrm{P}, \mathrm{V} \rangle (par, x; w)\}$. In the instantiations given in this paper, generator R calls on a generator of bilinear group structures, and the NP relation for which proofs are computed is defined over the generated bilinear group structure.

3.4.1 Properties. A proof system is expected to be complete and sound. It may additionally satisfy knowledge soundness and the honest-verifier zero-knowledge property. These properties are formally defined in Section B.1.1.

Completeness. A proof system is complete if V accepts any interaction with P on an instance x if the latter is given an input w such that $(x, w) \in R$.

Soundness. A proof system is sound if no PPT prover can make the verifier accept false statements, i.e., make the verifier accept on the input of an instance $x \notin L_R$ with non-negligible probability.

Knowledge Soundess. Stronger than the former notion of soundness, knowledge soundness (also called extractability) requires the existence of a probabilistic algorithm, called extractor, that runs in expected polynomial time and computes a witness for any instance for which the prover makes the verifier accept with a probability above a certain threshold. This threshold is called knowledge-soundness error. The knowledge-soundness error is a function of the security parameter and the size of the instance. The extractor is given black-box access to the prover algorithm and also has control over its random tape.

Honest-Verifier Zero-Knowledge. The notion of honest-verifier zero-knowledge formalises the idea that a proof should reveal no information beyond the fact that $x \in L_R$ to the verification algorithm of the proof system.

3.5 Polynomial Commitments

A polynomial-commitment scheme allows a party to commit to a polynomial and to later convince another party of evaluations of the committed polynomial.

3.5.1 Syntax. Given a field $\mathbb{F}$, a scheme to commit to univariate polynomials consists of a set of algorithms as defined below.

$\textsc{Setup}\left(1^\lambda, N_{\max} \in \mathbb{N}_{\geq 1}\right) \rightarrow par$: generates public parameters on the input of a security parameter encoded in unary and of a positive integer $N_{\max}$. The latter indicates a strict upper-bound on the maximum degree of the polynomials that are committed to, i.e., the polynomials to be committed to are of degree at most $N_{\max} - 1$. It is here tacitly assumed that the set-up algorithm also expects an auxiliary input which may for instance specify the basis (e.g., monomial or Lagrange) in which the polynomials to be committed to are represented. For simplicity, this input is omitted from the syntax. To lighten the notation, the parameters are given as an implicit input to the algorithms to follow whenever they are clear from the context.

$\textsc{Com}\left(f \in \mathbb{F}[X]^{<N_{\max}}\right) \rightarrow (C, r)$: computes a commitment to f (represented as a tuple of at most $N_{\max}$ field elements) and a piece of de-commitment information r, which typically is a random value used to compute the commitment.

$\textsc{Open}(C, f, r) \rightarrow b \in \{0, 1\}$: returns a bit indicating whether C is a valid commitment to f w.r.t. the de-commitment information r. The algorithm is said to accept if it returns 1 and to reject otherwise.

$\textsc{Eval}$: is a proof system for the language

$$\left\{(C, u, v) : \exists \left(f \in \mathbb{F}[X]^{<N_{\max}}, r\right), \textsc{Open}(C, f, r) = 1, f(u) = v\right\}.$$

The bound $N_{\max}$ on the degree of the witness f is here determined by par.

Commitments to Multilinear Polynomials. The above definition is readily adapted to the case of multilinear polynomials: polynomials are also given as a tuple of field elements to the commitment and opening algorithm. However, since the degree in each variable is known in advance to be at most 1, the integer received by the set-up algorithm, now denoted $n_{\max}$, instead represents the maximum number of variables of the polynomials to which the scheme allows to commit. The commitment and opening algorithms additionally take an integer $1 \leq n \leq n_{\max}$ specifying the number of variables of their input polynomials, now represented as a vector in $\mathbb{F}^{2^n}$.

3.5.2 Properties. A polynomial commitment scheme is expected to satisfy the following requirements.

Correctness. This property holds if the opening algorithm accepts all honestly computed commitments.

Binding. A scheme is considered binding if no two distinct polynomials can be successfully opened to the same commitment.

Hiding. A scheme is hiding if no PPT adversary can infer, with non-negligible probability, any information about the polynomial underlying a commitment.

Evaluation-Binding Proofs. The evaluation protocol is required to satisfy the following conditions.

- **IP Properties.** In addition to the completeness property, the evaluation protocol of a polynomial-commitment scheme is expected to be (knowledge) sound. The protocol may additionally be required to satisfy the (honest-verifier) zero-knowledge property.
- **Evaluation Binding.** the evaluation protocol of a polynomial-commitment scheme is also required to be *evaluation binding*, meaning that a commitment binds the prover to a function. That is, no malicious prover has a non-negligible probability to make the verifier accept in two protocol executions with the same commitment and evaluation point, but with two distinct claimed evaluations.

See Section B.2.1 for formal definitions.

Remark. *The evaluation protocol of a binding scheme is necessarily evaluation binding if it is extractable. This can be shown by contrapositive as follows. Suppose that the evaluation protocol of a scheme is extractable but not evaluation binding. It suffices to show that the scheme cannot be binding.*

If the scheme is not evaluation binding, then with non-negligible probability, a PPT adversary can compute valid proofs for two distinct evaluations at the same point w.r.t. the same commitment. Two polynomials which correspond to the commitment and which satisfy the evaluations can be extracted with non-negligible probability from the two valid proofs. However, since the evaluations are distinct at the same point, the two polynomials are necessarily distinct. This implies that two distinct openings for the same commitment can be computed with non-negligible probability and the scheme is therefore not binding.

3.5.3 Hiding KZG Commitments. Polynomial-delegation schemes are similar to polynomial-commitment schemes, except that the evaluations for which proofs are computed are also committed. Zhang, Genkin, Katz, Papdopoulos and Papamanthou [33] proposed a polynomial-delegation scheme which is similar in spirit to the pairing-based polynomial-commitment scheme due to Kate, Zaverucha and Goldberg [19]. The latter scheme stems from the observation that a polynomial $f \in \mathbb{F}[X]$ satisfies $f(u) = v$ if and only if there exists a polynomial $q \in \mathbb{F}[X]$ such that $f - v = (X - u)q$, i.e., if and only if $X - u$ divides $f - v$ in $\mathbb{F}[X]$. Evaluation proofs are non-interactive and consist of a single group element, which is a commitment to q.

The scheme which follows is a variation of standard KZG commitments and is inspired by their construction. Its main benefit is that it is hiding and only uses a single random field element to achieve this property. The standard KZG evaluation proofs are now also randomised. As a result, the prover must also send a corrective term to account for both random terms. In stark contrast, the hiding version of KZG commitments given in the original paper relies on a random hiding polynomial of the same degree as the committed polynomial.

Setup $\left(1^\lambda, N_{\max} \in \mathbb{N}_{\geq 1}\right)$:

$\mathbb{G} \coloneqq (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \textsc{Gen}\left(1^\lambda\right)$

$srs \leftarrow \left([1]_1, [\tau]_1, \ldots, \left[\tau^{N_{\max}-1}\right]_1, [\xi]_1, [1]_2, [\tau]_2, [\xi]_2\right)$ for $\tau, \xi \leftarrow_\$ \mathbb{F}^*$

Return $par \leftarrow (\mathbb{G}, srs)$.

Com $(f \coloneqq (a_0, \ldots, a_{N-1}))$:

$r \leftarrow_\$ \mathbb{F}$

$C \leftarrow a_0 \cdot [1]_1 + a_1 \cdot [\tau]_1 + \cdots + a_{N-1} \cdot \left[\tau^{N-1}\right]_1 + r \cdot [\xi]_1 = [f(\tau)]_1 + r \cdot [\xi]_1$

Return (C, r).

Open $(C, f \coloneqq (a_0, \ldots, a_{N-1}), r)$:

$$C \stackrel{?}{=} a_0 \cdot [1]_1 + a_1 \cdot [\tau]_1 + \cdots + a_{N-1} \cdot \left[\tau^{N-1}\right]_1 + r \cdot [\xi]_1.$$

Eval:

$\mathbf{P} \rightarrow \mathbf{V}$:

$\pi \leftarrow [q(\tau)]_1 + s \cdot [\xi]_1$ for q such that $f - v = (X - u)q$ and $s \leftarrow_\$ \mathbb{F}$

$\delta \leftarrow r \cdot [1]_1 - s \cdot [\tau]_1 + (s \cdot u) \cdot [1]_1 = [r - s(\tau - u)]_1$

Output (π, δ)

$\mathbf{V}$: $e(C - v \cdot [1]_1, [1]_2) \stackrel{?}{=} e(\pi, [\tau]_2 - u \cdot [1]_2) + e(\delta, [\xi]_2)$.

Properties. The scheme is correct and perfectly hiding by construction. It is also binding under the $(N_{\max} - 1)$ discrete-logarithm assumption.

The soundness of the evaluation protocol is irrelevant as the language is trivial as soon as $N_{\max} \geq 2$: by simple Lagrange interpolation, there is always a polynomial of degree 1 that takes the value $\mathrm{dlog}_{[1]_1}(C)$ at τ and satisfies the evaluation. The pair consisting of such a polynomial and zero as randomness is thus a valid witness.

Zhang et al. gave a proof [33, Proof of Theorem 1] that their polynomial-delegation satisfies, under the q-Strong-Diffie–Hellman (q-SDH) assumption [4], a binding property which amounts to the knowledge soundness property for polynomial-commitment schemes (as defined above)[4]. However, as explained in Section B.2.2, there is unfortunately a critical flaw in their proof[5] . Section B.2.2 gives a proof that the hiding KZG scheme is knowledge sound under the q-DLOG assumption, which is weaker than the q-SDH assumption. The approach in the proof is readily generalised to the case of multivariate polynomials as in the delegation scheme due to Zhang et al.

[4] With asymmetric bilinear group structures, the SDH assumption needed in their proof of the binding property of the scheme is symmetric: all powers of the trapdoor must be committed in both groups.

[5] This issue has been discussed with the authors, and the proof in Section B.2.2 stems from discussions with them.

4 Generic Construction

To prove that a multilinear polynomial f satisfies $f(\boldsymbol{u}) = v$ for public $\boldsymbol{u}$ and v, Lemma 2.3.1 shows that it is enough to prove the existence of $q_0 \in \mathbb{F}$ and of multilinear polynomials $q_1, \ldots, q_{n-1}$, with each q_k in the first k variables, such that $f - v = \sum_k (X_k - u_k) q_k$. Since the map $\mathcal{U}_n$ defined in Section 2.5 is an isomorphism, it is enough to prove the above equality over the image under $\mathcal{U}_n$ of the polynomials in the identity. The lemmas in Section 2.5 imply a univariate polynomial identity that these images must satisfy, in terms of $\mathcal{U}_n(f)$, $\mathcal{U}_n(q_k)$ and the public input.

Outline. The idea of the generic construction which follows is to commit to the image of f under $\mathcal{U}_n$. The definition of $\mathcal{U}_n$ implies that the multilinear polynomials are given in the Lagrange basis, and the univariate scheme allows to commit to polynomials given in the monomial basis. As $\mathcal{U}_n$ is an isomorphism, $f - v = \sum_k (X_k - u_k) q_k$ if and only if $\mathcal{U}_n(f) - \mathcal{U}_n(v) = \sum_k \mathcal{U}_n(q_k X_k) - u_k \mathcal{U}_n(q_k)$, and Lemmas 2.5.1 and 2.5.2, and Corollary 2.5.3.2 show that the latter identity holds if and only if

$$\mathcal{U}_n(f) - v \cdot \Phi_n(X) = \sum_k \left(X^{2^k} \Phi_{n-k-1}\left(X^{2^{k+1}} \right) - u_k \cdot \Phi_{n-k}\left(X^{2^k} \right) \right) \mathcal{U}_n(q_k)^{<2^k}.$$

In the evaluation protocol, the prover starts by sending commitments to $\mathcal{U}_n(q_k)^{<2^k}$. The verifier first checks that the committed polynomials are of degree at most $2^k - 1$ via a degree-check protocol, and then sends a uniformly random value x to the prover. The prover then shows that

$$\begin{aligned} Z_x(X) &:= \mathcal{U}_n(f)(X) - v \cdot \Phi_n(x) \\ &\quad - \sum_k \left(x^{2^k} \Phi_{n-k-1}\left(x^{2^{k+1}} \right) - u_k \cdot \Phi_{n-k}\left(x^{2^k} \right) \right) \mathcal{U}_n(q_k)^{<2^k}(X) \end{aligned}$$

evaluates to 0 at x via the evaluation protocol of the univariate scheme. Assuming the univariate scheme to be additively homomorphic, the verifier can compute a commitment to this polynomial, as it results from partially evaluating at x the terms in the above polynomial identity that the verifier can compute on its own. If the degree-check and univariate-evaluation protocols are secure in the standard model, then so is the generic construction.

Building Blocks. The scheme assumes the existence of

- Uni, a univariate-polynomial commitment scheme that is additively homomorphic
- Deg, a proof system for the language

$$\left\{ (C, d \in \mathbb{N}_{\geq 0}) : \exists \left(f \in \mathbb{F}[X]^{\leq d}, r \right), \text{Uni.Open}(C, f, r) = 1 \right\}.$$

The public inputs and witness are in the sets determined by Uni.Setup.

Formal Description.

$\textsc{Setup}\left(1^\lambda, n_{\max}\right)$: Return $par \leftarrow \textsc{Uni.Setup}\left(1^\lambda, 2^{n_{\max}}\right)$.

$\textsc{Com}\,(f, n)$: Return $(C, r) \leftarrow \textsc{Uni.Com}\,(\mathcal{U}_n(f))$.

$\textsc{Open}\,(C, f, n, r) \rightarrow b \in \{0,1\}$: Return $b \leftarrow \textsc{Uni.Open}\,(C, \mathcal{U}_n(f), r)$.

$\textsc{Eval}$:

$\mathbf{P}\,(C, \boldsymbol{u} = (u_0, \ldots, u_{n-1}), v, f, r) \rightarrow \mathbf{V}(C, \boldsymbol{u}, v)$:

For $(k = 0, \ldots, n-1)\left\{(C_k, r_k) \leftarrow \textsc{Uni.Com}\left(\mathcal{U}_n(q_k)^{<2^k}\right)\right\}$

Output $(C_0, \ldots, C_{n-1})$

$\mathbf{P} \rightleftharpoons \mathbf{V}$:

$\langle \textsc{Deg.P}, \textsc{Deg.V}\rangle\,(C, 2^n - 1; \mathcal{U}_n(f), r)$

For $(k = 0, \ldots, n-1)$

$\langle \textsc{Deg.P}, \textsc{Deg.V}\rangle\left(C_k, 2^k - 1; \mathcal{U}_n(q_k)^{<2^k}, r_k\right)$

$\mathbf{P} \leftarrow \mathbf{V}$:

$x \leftarrow_\$ \mathbb{F}$

$(C_{v,x}, 0) \leftarrow \textsc{Uni.Com}\,(v \cdot \Phi_n(x); 0)$

$C_{Z_x} \leftarrow C - C_{v,x} - \sum_k \left(x^{2^k}\Phi_{n-k-1}\left(x^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(x^{2^k}\right)\right) C_k$

Output x

$\mathbf{P}$: $r_Z \leftarrow r - \sum_k \left(x^{2^k}\Phi_{n-k-1}\left(x^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(x^{2^k}\right)\right) \cdot r_k$

$\mathbf{P} \rightleftharpoons \mathbf{V}$: $\langle \textsc{Uni.Eval.P}, \textsc{Uni.Eval.V}\rangle\,(C_{Z_x}, x, 0; Z_x, r_Z)$.

Properties. The correctness of Uni implies the correctness of the scheme. The binding property of the scheme follows from that of Uni and the fact that $\mathcal{U}_n$ is an isomorphism. The hiding property follows from that of Uni. The completeness of the evaluation protocol follows from the correctness of Uni and the completeness of Uni.Eval and Deg.

As for the knowledge soundness of the evaluation protocol, if Deg is knowledge sound, a polynomial $\hat{f}$ of degree at most 2^n-1 and polynomials $\hat{f}_k$ of degree at most $2^k - 1$ corresponding to C and C_k can be efficiently extracted from a prover that makes the verifier accept with non-negligible probability. Assuming Uni.Eval to be sound, for a uniformly random $x \in \mathbb{F}$,

$$\begin{aligned}
&\left(\hat{f} - v \cdot \Phi_n(x) - \sum_k \left(x^{2^k}\Phi_{n-k-1}\left(x^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(x^{2^k}\right)\right)\hat{f}_k\right)(x) \\
&= \hat{f}(x) - v \cdot \Phi_n(x) - \sum_k \left(x^{2^k}\Phi_{n-k-1}\left(x^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(x^{2^k}\right)\right)\hat{f}_k(x) \\
&= \left(\hat{f} - v \cdot \Phi_n - \sum_k \left(X^{2^k}\Phi_{n-k-1}\left(X^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(X^{2^k}\right)\right)\hat{f}_k\right)(x) \\
&= 0.
\end{aligned}$$

By the polynomial-identity lemma, this means that

$$\hat{f} - v \cdot \Phi_n(X) - \sum_k \left(X^{2^k} \Phi_{n-k-1}\left(X^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(X^{2^k}\right)\right) \hat{f}_k = 0$$

with probability at least $1 - 2^n/|\mathbb{F}|$.

In other words, with probability at least $1 - 2^n/|\mathbb{F}|$, for $f := \mathcal{U}_n^{-1}\left(\hat{f}\right)$ and $q_k := \mathcal{U}_n^{-1}\left(\hat{f}_k\right)$, the equality $f - v = \sum_k (X_k - u_k) q_k$ holds and $f(\boldsymbol{u}) = v$.

Lastly, if Uni is hiding and Deg and Uni.Eval are zero-knowledge, then the evaluation protocol is zero-knowledge: to simulate a protocol transcript, it suffices to commit to dummy polynomials (the commitments are indistinguishable from real ones by the hiding property of Uni) and to then run the simulators of Deg and Uni.Eval.

On Efficiency. The prover needs randomness only to compute commitments to the $\mathcal{U}_n(q_k)^{<2^k}$ polynomials, and in the degree-check and univariate-evaluation sub-protocols. The computational efficiency of the evaluation protocol is contingent on the efficiency of the degree-check protocol and of the univariate evaluation protocol. Batching degree-check protocol executions would improve the efficiency of the protocol. In the instantiation of the protocol with hiding KZG univariate commitments, the degree checks are not only batched, but also merged with the last evaluation step for increased efficiency. As a consequence, the knowledge soundness of the scheme does not directly follow from that of the generic construction and requires extra arguments.

5 Degree-Check Protocols

This section first gives a protocol to check the degree of univariate polynomials committed with the hiding KZG scheme (Section 3.5.3), as a building block for the instantiation (in Section 6) of the multilinear scheme with hiding KZG commitments. It is inspired by the polynomial identity that the variant of KZG due to Maller, Bower, Kohlweiss and Meiklejohn [22, Figure 3] enables the verifier to check.

Next comes a protocol to jointly verify an evaluation and a degree bound of a committed polynomial. The latter protocol serves as a building block for a following protocol to efficiently batch degree-check proofs. The last protocol of the section allows to batch evaluation proofs with degree check if the evaluation point and the degree bound is the same for all polynomials; it will enable an optimisation of the instantiation of the Section-4 protocol with hiding KZG commitments.

All these protocol come at the cost of requiring, in the reference string, higher (compared to the description in Section 3.5.3) powers of the first trapdoor to be committed in the second group.

5.1 Single KZG Degree Check

Given a hiding KZG commitment to a polynomial $f \in \mathbb{F}[X]$ and a non-negative integer $d < N_{\max}$, to check that $\deg f \leq d$, the idea of the protocol is to simply check that $f \cdot X^{N_{\max}-1-d} = fX^{N_{\max}-1-d} \cdot 1$ via pairing computations. If $\deg f > d$, then the prover would not be able to commit to $fX^{N_{\max}-1-d}$, as its degree $N_{\max}+\deg(f)-1-d$ would exceed $N_{\max}-1$, the maximum degree in the reference string. The proof that $\deg f \leq d$ thus precisely consists in such a commitment.

Formal Description. Let KZG denote the scheme recalled in Section 3.5.3. Given parameters *par* generated by KZG.Setup, the protocol which follows is for the language

$$\left\{(C, 0 \leq d < N_{\max}) \colon \exists \left(f \in \mathbb{F}[X]^{\leq d}, r \in \mathbb{F}\right), \mathrm{KZG.O{\scriptsize PEN}}(C, f, r) = 1\right\}.$$

$\mathbf{P} \to \mathbf{V}:$

$\pi \leftarrow \left[f(\tau)\tau^{N_{\max}-1-d}\right]_1 + s \cdot [\xi]_1$ for $s \leftarrow_\$ \mathbb{F}$

$\delta \leftarrow r \cdot \left[\tau^{N_{\max}-1-d}\right]_1 - s \cdot [1]_1$

Output (π, δ)

$\mathbf{V}: \ e\left(C, \left[\tau^{N_{\max}-1-d}\right]_2\right) \stackrel{?}{=} e\left(\pi, [1]_2\right) + e\left(\delta, [\xi]_2\right).$

Properties. Completeness follows from the fact that

$$\begin{aligned}(f(\tau) + r\xi) \cdot \tau^{N_{\max}-1-d} &= (f(\tau) + r\xi)\tau^{N_{\max}-1-d} + s\xi - s\xi \\ &= \left(f(\tau)\tau^{N_{\max}-1-d} + s\xi\right) \cdot 1 + \left(r\tau^{N_{\max}-1-d} - s\right) \cdot \xi.\end{aligned}$$

The knowledge soundness of the protocol can be proved in the algebraic-group model under the DLOG assumption with parameter $N_{\max}-1$ as follows. Suppose that there exists a PPT algebraic adversary $(\mathrm{A}, \mathrm{P}^*)$ which makes the verifier accept with non-negligible probability. A polynomial f of degree at most $N_{\max}-1$ and a field element $r \in \mathbb{F}$ such that $C = [f(\tau) + r\xi]_1$ can be extracted from the commitment C computed by A because the latter is algebraic. If $\deg f \leq d$, then (f, r) is a valid witness.

Otherwise, i.e., in the event that $\deg f > d$, let B be a reduction algorithm which interacts with the DLOG challenger and runs the adversary as sub-routine. Upon receiving a DLOG challenge tuple, algorithm B chooses $\xi \leftarrow_\$ \mathbb{F}^*$ and sets the SRS as in the scheme.

Given a proof (π, δ) computed by P^*, algorithm B can extract polynomials g and h of degree at most $N_{\max} - 1$ and field elements s and t such that $\pi = [g(\tau) + s\xi]_1$ and $\delta = [h(\tau) + t\xi]_1$. Since the verifier accepts,

$$(f(\tau) + r\xi) \cdot \tau^{N_{\max}-1-d} = g(\tau) + s\xi + (h(\tau) + t\xi) \cdot \xi.$$

It means that τ is a root of polynomial

$$k := (f + r\xi)X^{N_{\max}-1-d} - g - s\xi - (h + t\xi) \cdot \xi$$

which is of degree at most $2(N_{\max} - 1) - d$. If $\deg(f) > d$, then this polynomial is necessarily non-zero as $fX^{N_{\max}-1-d}$ is of degree at least $N_{\max}$ and $r\xi X^{N_{\max}-1-d} - g - s\xi - (h + t\xi) \cdot \xi$ is of degree at most $N_{\max} - 1$. Therefore, polynomial k can be factorised in polynomial time [14, 27], which allows B to recover τ.

In other words, the probability that a valid witness can be extracted is at least the probability that $(\mathrm{A}, \mathrm{P}^*)$ makes the verifier accept minus the supremal advantage of any PPT algorithm in solving the $(N_{\max} - 1)$-DLOG problem. The latter assumed to be negligible, the probability that a valid witness is extracted is negligibly close to the probability that the verifier accepts.

Given the trapdoor (τ, ξ), a proof on a commitment C can be simulated by choosing $\pi \leftarrow_\$ \mathbb{G}_1$ and computing $\delta \leftarrow (\tau^{N_{\max}-1-d} C - \pi)\xi^{-1}$.

5.2 KZG Evaluation Proofs with Degree Check

In preparation of the protocol to batch degree checks on KZG commitments, the following protocol allows to simultaneously prove an evaluation (at non-zero points) and the degree of a non-constant polynomial committed with KZG. Unlike the scheme in Sonic [22, Figure 3], the commitment algorithm remains the standard (hiding) KZG algorithm, which gives the flexibility to compute evaluation proofs on hiding KZG commitments with or without degree bounds, depending on the need.

More precisely, given parameters *par* generated by KZG.Setup, the following protocol is for the language

$$\left\{(C, 0 < d < N_{\max}, u \neq 0, v) \colon \exists \left(f \in \mathbb{F}[X]^{\leq d}, r \in \mathbb{F}\right), \mathrm{KZG.Open}(C, f, r) = 1, \; f(u) = v\right\}.$$

It is based on the observation that

$$f - v = q(X - u) \Leftrightarrow (f - v) \cdot X^{N_{\max}-d} = qX^{N_{\max}-d} \cdot (X - u)$$

by integrality of $\mathbb{F}[X]$. Proving that identity is enough to show that $f(u) = v$, but in addition to that, if $\deg f > d$, then the degree of $qX^{N_{\max}-d}$ would be at least $N_{\max}$ and the prover would not be able to commit to it, so a commitment to $qX^{N_{\max}-d}$ is also a proof that $\deg f \leq d$.

Formal Description.

$\mathbf{P} \rightarrow \mathbf{V}$:

$\pi \leftarrow \left[q(\tau)\tau^{N_{\max}-d}\right]_1 + s \cdot [\xi]_1$, for q s.t. $f - v = (X - u)q$ and $s \leftarrow_\$ \mathbb{F}$

$\delta \leftarrow r \cdot \left[\tau^{N_{\max}-d}\right]_1 - s \cdot [\tau]_1 + (s \cdot u) \cdot [1]_1 = \left[r\tau^{N_{\max}-d} - s(\tau - u)\right]_1$

Output (π, δ)

$\mathbf{V}$: $e\left(C - v \cdot [1]_1, \left[\tau^{N_{\max}-d}\right]_2\right) \stackrel{?}{=} e\left(\pi, [\tau]_2 - u \cdot [1]_2\right) + e\left(\delta, [\xi]_2\right).$

Properties. The completeness of the protocol stems from the fact that

$$\begin{aligned}(f(\tau)+r\xi-v)\cdot\tau^{N_{\max}-d} &= (q(\tau-u)+r\xi)\cdot\tau^{N_{\max}-d}\\ &= q\tau^{N_{\max}-d}\cdot(\tau-u)+\left(r\tau^{N_{\max}-d}\right)\cdot\xi\\ &= \left(q\tau^{N_{\max}-d}+s\xi-s\xi\right)\cdot(\tau-u)+\left(r\tau^{N_{\max}-d}\right)\cdot\xi\\ &= \left(q\tau^{N_{\max}-d}+s\xi\right)\cdot(\tau-u)\\ &+ \left(r\tau^{N_{\max}-d}-s(\tau-u)\right)\cdot\xi.\end{aligned}$$

It remains to show that the protocol is knowledge sound. It can be proved under the $(2N_{\max}-1)$-DLOG assumption in the algebraic-group model as follows. Suppose that there exists a PPT algebraic adversary $(\mathrm{A},\mathrm{P}^*)$ which makes the verifier accept with non-negligible probability.

Since the adversary is algebraic, a polynomial $f\in\mathbb{F}[X]$ of degree at most $N_{\max}-1$ and a field element r such that $C=[f(\tau)+r\xi]_1$ can be extracted from the commitment computed by A. If $\deg f\le d$ and $f(u)=v$, then (f,r) is a valid witness.

Otherwise, i.e., in the event that $\deg f>d$ or $f(u)\neq v$, let B be a reduction algorithm which interacts with the DLOG challenger and runs the adversary as sub-routine. Upon receiving a DLOG challenge tuple

$$\left([1]_1,[\tau]_1,\ldots,\left[\tau^{2N_{\max}-1}\right]_1,[1]_2,[\tau]_2,\ldots,\left[\tau^{2N_{\max}-1}\right]_2\right),$$

B chooses $\rho\leftarrow_{\$}\mathbb{F}^*$, and sets the SRS as

$$\left([1]_i,[\tau]_i,\ldots,\left[\tau^{N_{\max}-1}\right]_i,\rho\cdot\left[\tau^{2N_{\max}-1}\right]_i\right)_{i\in\{1,2\}},$$

i.e., ξ is implicitly set as $\rho\tau^{2N_{\max}-1}$. Note that because ρ and τ are independent, the distribution of this SRS is the same as that of the SRS in the scheme.

Given a proof (π,δ) computed by P^*, algorithm B can extract polynomials g and h of degree at most $N_{\max}-1$ and field elements s and t such that $\pi=[g(\tau)+s\xi]_1$ and $\delta=[h(\tau)+t\xi]_1$. Since the verifier accepts,

$$\begin{aligned}\left(f(\tau)+r\rho\tau^{2N_{\max}-1}-v\right)\cdot\tau^{N_{\max}-d} &= \left(g(\tau)+s\rho\tau^{2N_{\max}-1}\right)\cdot(\tau-u)\\ &+\left(h(\tau)+t\rho\tau^{2N_{\max}-1}\right)\cdot\rho\tau^{2N_{\max}-1},\end{aligned}$$

i.e., τ is a root of

$$\begin{aligned}k(X)\coloneqq\left(f+r\rho X^{2N_{\max}-1}-v\right)X^{N_{\max}-d}&-\left(g+s\rho X^{2N_{\max}-1}\right)(X-u)\\ &-\left(h+t\rho X^{2N_{\max}-1}\right)\rho X^{2N_{\max}-1}.\end{aligned}$$

If k is non-zero, then B can recover τ via factorisation [14, 27] and solve the DLOG problem. Consider

$$\ell(X,Y)\coloneqq(f(X)+rY-v)X^{N_{\max}-d}-(g(X)+sY)(X-u)-(h+tY)\,Y.$$

It is the pre-image of k under the linear map

$$\mathbb{F}[X,Y]^{\preceq 2(N_{\max}-1)} \to \mathbb{F}[X]^{\leq (2N_{\max}-1)^2-1}$$

which sends X^iY^j to $X^i\left(\rho X^{2N_{\max}-1}\right)^j$ for all $i,j \in \{0,\ldots,2N_{\max}-2\}$. It is an isomorphism because $\rho \neq 0$ and the $2(N_{\max}-1)$-ary decomposition of integers in $\{0,\ldots,(2N_{\max}-1)^2-1\}$ is unique, i.e., any integer in this set is uniquely written as $i+2(N_{\max}-1)\cdot j$ with $i,j \in \{0,\ldots,2N_{\max}-2\}$.

To show that k is non-zero in the event that $\deg f > d$ or $f(u) \neq v$, it suffices to show that ℓ is non-zero. If $\deg f > d$, then the term $fX^{N_{\max}-d}$ has degree at least $N_{\max}+1$, whereas all the other terms in $\ell \in \mathbb{F}[Y][X]$ are of degree at most $N_{\max}$ in X. If $\deg f \leq d$ but $f(u) \neq v$, then the constant term of ℓ as a polynomial in $\mathbb{F}[X][Y]$ is $(f-v)X^{N_{\max}-d}-(X-u)g$. If it were zero, then $X^{N_{\max}-d}$ would divide g because $X^{N_{\max}-d}$ and $X-u$ are co-prime since $u \neq 0$, i.e., there would exist $q \in \mathbb{F}[X]$ such that $f-v=(X-u)q$, which is equivalent to $f(u)=v$. The constant term can thus not be zero, and ℓ is non-zero, i.e., k is non-zero. In other words, in the event that $\deg f > d$ or $f(u) \neq v$, k is necessarily non-zero and B can recover τ by factorisation [14, 27].

Given the trapdoor (τ,ξ), a proof on a commitment C can be simulated by choosing $\pi \leftarrow_{\$} \mathbb{G}_1$ and computing $\delta \leftarrow \left(\tau^{N_{\max}-d}(C-v)-\pi(\tau-u)\right)\xi^{-1}$.

5.3 Batched Degree Checks

The following protocol allows to prove, at once, that the degree of committed polynomials do not exceed individual, public bounds, more efficiently than performing independent single degree checks. The protocol hinges on the following lemma which addresses the case where the bound is the same for all polynomials.

Lemma 5.1. *Let $m \in \mathbb{N}_{\geq 1}$ and $I=\{i_0,\ldots,i_{m-1}\}$ a set of m pairwise distinct non-negative integers. Consider $g_0,\ldots,g_{m-1} \in \mathbb{F}[X]$. For any non-negative integer d, there are at most* $\max(I)$ *values $y \in \mathbb{F}$ such that* $\deg\left(\sum_{j=0}^{m-1} y^{i_j} g_j\right) \leq d$ *if $\deg g_j > d$ for some $j \in \{0,\ldots,m-1\}$.*

Proof. Let $g_j =: \sum_{k\geq 0} a_{j,k}X^k$. Suppose that for some non-negative integer $j < m$, there exists an integer $d' > d$ such that $a_{j,d'} \neq 0$. The term of degree d' in X of $\sum_{j=0}^{m-1} Y^{i_j} g_j \in \mathbb{F}[X][Y]$ is $\sum_j Y^{i_j} a_{j,d'}$. It is a polynomial in Y of degree at most $\max(I)$, so it has at most $\max(I)$ roots, hence the claim. □

Given integers $d_0,\ldots,d_{n-1} \in \{0,\ldots,N_{\max}-1\}$, to check that polynomials $f_0,\ldots,f_{n-1} \in \mathbb{F}[X]$ satisfy $\deg f_k \leq d_k$ for all k, the most straightforward way is to leverage the polynomial identity underlying the protocol for single degree checks. That is, checking at once that

$$f_k \cdot X^{N_{\max}-1-d_k} - f_k X^{N_{\max}-1-d_k} \cdot 1 = 0$$

for all $0 \leq i < n$. To do so, it suffices to consider each polynomial on the left-hand side in the above identities as the coefficients of a polynomial in $\mathbb{F}[X][Y]$, so by the polynomial-identity lemma, it suffices to check that

$$\sum_{k=0}^{n-1} y^k f_k \cdot X^{N_{\max}-1-d_k} = \left(\sum_{k=0}^{n-1} y^k f_k X^{N_{\max}-1-d_k}\right) \cdot 1,$$

for a uniformly random y chosen by the verifier. However, performing this check via pairings would incur n pairing computations. The root cause is that this approach essentially performs a degree check for each bound d_k at each power y^k. Alternatively, at the cost of an increase in prover computation and two extra rounds of interaction, it is possible to check a polynomial identity that requires only two pairings.

Outline. The main idea is to lift the degree of all polynomials $f_0, \ldots, f_{n-1}$ to the same degree (without introduction new non-zero terms) and perform a single degree check. More concretely, for an integer $d^* \geq \max d_k$ and a random $y \in \mathbb{F}$ chosen by the verifier, the prover first commits to $f := \sum_{k=0}^{n-1} y^k X^{d^*-d_k+1} f_k$. Note that each polynomial $X^{d^*-d_k+1} f_k$ is of degree $d^* + 1$. The verifier then sends to the prover a random $x \in \mathbb{F}^*$, and they run a single degree check on polynomial

$$\begin{aligned} \zeta_x &:= f - \sum_{k=0}^{n-1} y^k x^{d^*-d_k+1} f_k \\ &= f - \sum_{d \in \{d_0, \ldots, d_{n-1}\}} x^{d^*-d+1} \sum_{k\colon d_k = d} y^k f_k \end{aligned}$$

with degree bound $d^* + 1$, and the prover also shows that $\zeta_x(x) = 0$. The verifier leverages the homomorphic property of the scheme to compute a commitment to ζ_x given the commitments to $f_0, \ldots, f_{n-1}$ and f.

If $\deg f > d^* + 1$ or $f - \sum_k y^k X^{d^*-d_k+1} f_k \neq 0$ with non-negligible probability for adversarially computed polynomials $f_0, \ldots, f_{n-1}, f$, the probability that the value x chosen by the verifier is both a root of $f - \sum_k y^k X^{d^*-d_k+1} f_k$ and in the set of values $\alpha \in \mathbb{F}$ such that $\deg\left(f - \sum_k y^k \alpha^{d^*-d_k+1} f_k\right) \leq d^* + 1$ is negligible. Indeed, if $\deg f > d^* + 1$, then the probability that x is in the set of such values α is at most $(d^* - \min d_k + 1)/|\mathbb{F}^*|$ by Lemma 5.1, and if $f - \sum_k y^k X^{d^*-d_k+1} f_k \neq 0$, then the probability that x is one of its roots is at most $(N_{\max} - 1) \cdot (d^* - \min d_k + 1)/|\mathbb{F}^*|$. Therefore, by Lemma 2.1, if an adversary could compute with non-negligible probability commitments to polynomials f and f_k such that $\deg f > d^* + 1$ or $f - \sum_k y^k X^{d^*-d_k+1} f_k \neq 0$, the probability that the verifier accepts would be negligible. In other words, if the verifier accepts, then $\deg f \leq d^* + 1$ and $f - \sum_k y^k X^{d^*-d_k+1} f_k = 0$ with overwhelming probability. A second application of Lemma 5.1 applied to $X^{d^*-d_0+1} f_0, \ldots, X^{d^*-d_{n-1}+1} f_{n-1}$ implies that with probability at least $1 - n/|\mathbb{F}|$ over the choice of y, $\deg\left(X^{d^*-d_k+1} f_k\right) \leq d^* + 1$, i.e., $\deg f_k \leq d_k$ for all $k \in \{0, \ldots, n-1\}$.

The KZG evaluation proof and the degree-check protocol from Section 5.1 allow to separately prove both statements $\zeta_x(x) = 0$ and $\deg(\zeta_x) \leq d^* + 1$. These statements can however be simultaneously proved in a single proof using the protocol from Section 5.2 (that is why challenge x must be non-zero).

Formal Description. Given parameters *par* generated by KZG.Setup, the protocol which follows is for the language[6]

$$\Big\{(C_k, 0 \leq d_k < N_{\max} - 1)_{k=0}^{n-1} : \forall k, \exists \left(f_k \in \mathbb{F}[X]^{\leq d_k}, r_k \in \mathbb{F}\right), \\ \text{KZG.Open}(C_k, f_k, r_k) = 1 \ \Big\}.$$

Let $d^* \in \{\max(d_k), \ldots, N_{\max} - 2\}$.

$\mathbf{P} \leftarrow \mathbf{V}$: $y \leftarrow_\$ \mathbb{F}$

$\mathbf{P} \rightarrow \mathbf{V}$: Output C_f for $(C_f, r) \leftarrow \text{KZG.Com}\left(f \coloneqq \sum_{k=0}^{n-1} y^k X^{d^* - d_k + 1} f_k\right)$

$\mathbf{P} \leftarrow \mathbf{V}$: $x \leftarrow_\$ \mathbb{F}^*$

$\mathbf{P} \rightarrow \mathbf{V}$:

$$\zeta_x \coloneqq f - \sum_{k=0}^{n-1} y^k x^{d^* - d_k + 1} f_k$$

$$\pi \leftarrow \left[q(\tau)\tau^{N_{\max} - d^* - 1}\right]_1 + s \cdot [\xi]_1, \text{ for } q \text{ such that } \zeta_x = (X - x)q$$

$$\delta \leftarrow \left(r - \sum_{k=0}^{n-1} y^k x^{d^* - d_k + 1} r_k\right) \cdot \left[\tau^{N_{\max} - d^* - 1}\right]_1 - s \cdot [\tau]_1 + (s \cdot x) \cdot [1]_1$$

Output (π, δ)

$\mathbf{V}$:

$$C_{\zeta_x} \leftarrow C_f - \sum_{k=0}^{n-1} y^k x^{d^* - d_k + 1} C_k$$

$$e\left(C_{\zeta_x}, \left[\tau^{N_{\max} - (d^* + 1)}\right]_2\right) \stackrel{?}{=} e\left(\pi, [\tau]_2 - x \cdot [1]_2\right) + e\left(\delta, [\xi]_2\right).$$

Properties. The completeness of the protocol is implied by the homomorphic property of KZG commitments and the completeness of the protocol from Section 5.2.

As for its knowledge soundness, the knowledge soundness of the protocol in Section 5.2 guarantees that a valid opening ζ to $C_f - \sum_{k=0}^{n-1} y^k x^{d^* - d_k + 1} C_k$, such that $\deg(\zeta_x) \leq d^* + 1$ and $\zeta_x(x) = 0$, can be extracted in the algebraic-group model given a valid proof π. For openings $f, f_0, \ldots, f_{n-1}$ to $C_f, C_0, \ldots, C_{n-1}$ extracted from an algebraic adversary, it follows that ζ_x and $f - \sum_{k=0}^{n-1} y^k x^{d^* - d_k + 1} f_k$ are both valid openings to $C_f - \sum_{k=0}^{n-1} y^k x^{d^* - d_k + 1} C_k$. If ζ_x were not equal to

[6] The protocol does not support $N_{\max} - 1$ as a bound for any polynomial since the protocol from Section 5.2 is applied with degree bound $d^* \geq \max(d_k) + 1$, and the latter cannot exceed $N_{\max} - 1$, the size of the reference reference. This is however without any loss of generality as any polynomial computed by the adversary necessarily has degree at most $N_{\max} - 1$ in the algebraic-group model.

$f - \sum_{k=0}^{n-1} y^k x^{d^*-d_k+1} f_k$, then the trapdoor τ would be a root of their difference and could be recovered by factorisation [14, 27]. On this account, under the $(N_{\max} - 1)$-DLOG assumption (and a fortiori under the $(2N_{\max} - 1)$-DLOG assumption), they are both equal with overwhelming probability. The reasoning given in the preamble then allows conclude that $\deg f_k \leq d_k$ for all $k \in \{0, \ldots, n-1\}$.

Given the trapdoor (τ, ξ), a proof on commitments $C_0, \ldots, C_{n-1}$ can be simulated by choosing $y \leftarrow_\$ \mathbb{F}$, $C_f \leftarrow_\$ \mathbb{G}_1$, $x \leftarrow_\$ \mathbb{F}^*$, C_{ζ_x} as in the scheme, $\pi \leftarrow_\$ \mathbb{G}_1$ and computing $\delta \leftarrow \left(\tau^{N_{\max}-(d^*+1)} C_{\zeta_x} - \pi(\tau - x)\right) \xi^{-1}$.

5.4 Generic Batched Degree Checks

The technique in Section 5.3 to batch degree-checks with multiple degree bounds is not limited to KZG commitments. It also applies to any knowledge sound degree-check protocol on homomorphic univariate commitments if the evaluation protocol is knowledge sound.

The main difficulty is to prove the protocol knowledge sound in the standard model. To do so, the idea is, for enough values of y that lead to a successful execution, to rewind any successful prover to the step right after it outputs C_f, and obtain accepting protocol executions for enough values of x. Openings to the input commitments $C_0, \ldots, C_{n-1}$ can then be recovered by linear combination of openings extracted from the various executions of EVAL.

Building Blocks. The scheme assumes the existence of

- (SETUP, COM, OPEN, EVAL), a univariate-polynomial commitment scheme that is additively homomorphic
- DEG, a proof system for the language

$$\left\{(C, d \in \mathbb{N}_{\geq 0}) : \exists \left(f \in \mathbb{F}[X]^{\leq d}, r\right), \textsc{Open}(C, f, r) = 1\right\}.$$

The public inputs and witness are in the sets determined by SETUP.

Formal Description. Given parameters *par* generated by SETUP, the protocol which follows is for the language

$$\Big\{(C_k, 0 \leq d_k < N_{\max} - 1)_{k=0}^{n-1} : \forall k, \exists \left(f_k \in \mathbb{F}[X]^{\leq d_k}, r_k \in \mathbb{F}\right), \textsc{Open}(C_k, f_k, r_k) = 1 \Big\}.$$

Let $d^* \in \mathbb{Z}_{\geq \max(d_k)}$.

$\mathbf{P} \leftarrow \mathbf{V}$: $y \leftarrow_\$ \mathbb{F}$

$\mathbf{P} \rightarrow \mathbf{V}$: Output C_f for $(C_f, r) \leftarrow \textsc{Com}\left(f \coloneqq \sum_{k=0}^{n-1} y^k X^{d^*-d_k+1} f_k\right)$

$\mathbf{P} \leftarrow \mathbf{V}$:

$x \leftarrow_\$ \mathbb{F}$

$C_{\zeta_x} \leftarrow C_f - \sum_{k=0}^{n-1} y^k x^{d^*-d_k+1} C_k$

Output x

P :

$\zeta_x \coloneqq f - \sum_{k=0}^{n-1} y^k x^{d^*-d_k+1} f_k$

$r_\zeta \leftarrow r - \sum_{k=0}^{n-1} y^k x^{d^*-d_k+1} r_k$

$\mathbf{P} \rightleftharpoons \mathbf{V}$: $\langle \text{Deg.P}, \text{Deg.V} \rangle (C_{\zeta_x}, d^* + 1; \zeta_x, r_\zeta)$

$\mathbf{P} \rightleftharpoons \mathbf{V}$: $\langle \text{Eval.P}, \text{Eval.V} \rangle (C_{\zeta_x}, x, 0; \zeta_x, r_\zeta)$.

Properties. The completeness of the protocol is implied by the homomorphic property of the commitment scheme and the completeness of Deg and Eval.

As regards the knowledge soundness of the protocol, consider an adversary (A, P^*) which makes the verifier accept with probability p_{P^*}. Suppose that Eval is knowledge sound with error κ_{Eval}, and that p_{P^*} is at least $16\kappa_{\text{Eval}}$. Let E_{Eval} denote an extractor for Eval that returns valid witnesses with a probability close to the success probability and up to a multiplicative polynomial factor p_{Eval}. Similarly, suppose that Deg is knowledge sound with error κ_{Deg} and that p_{P^*} is at least $16\kappa_{\text{Deg}}$, and let E_{Deg} denote an extractor for protocol Deg (with factor p_{Deg}).

Let $C_0, \ldots, C_{n-1}$ denote the commitments returned by A as part of the instance on which P^* computes a proof. Those being valid commitments[7], let $f_0, \ldots, f_{n-1}$ be any corresponding openings, i.e., polynomials for which there exist $r_0, \ldots, r_{n-1} \in \mathbb{F}$ such that $C_k = \text{Com}(f_k; r_k)$. The distribution of the tuple $(f_0, \ldots, f_{n-1})$ and y (and x) are independent, as the latter is chosen independently of the messages from the prover.

Likewise, let C_f denote the commitment output by P^* in the second round of the protocol, and let f be any opening to C_f, i.e., any polynomial for which there exist $r \in \mathbb{F}$ such that $C_f = \text{Com}(f; r)$. As above, the distributions of f and x are independent. It thus remains to define an extractor that can return, in expected polynomial time, such polynomials $f_0, \ldots, f_{n-1}$ with a probability that is close to p_{P^*}, up to a factor that is polynomial in the security parameter and the size of the instance.

It is important for the upcoming analysis to stress that even though the choice of the specific openings returned by the extractor to follow may depend on y and x, the distribution of the extracted openings to C_k are independent of any y and x, and that given a fixed y, the distribution of an extracted opening to C_f is independent of x[8].

[7] This supposes that the verification algorithm can efficiently test whether a commitment is a valid.

[8] As a simplified example, if an adversary computes a commitment C for which there are two possible openings f_0 and f_1, even if the extractor has a much higher probability, depending on the prover, to return f_0 than to return f_1 via a computation that depends on y and x, the distribution of f_0 and f_1 are independent of y and x.

Consider now an algorithm E that is given black-box access to P^*, and control over its randomness. For all $d \in \{d_0, \ldots, d_{n-1}\}$, let $n_d \coloneqq |\{k \colon d_k = d\}|$ (note that $\sum_d n_d = n$) and $m \coloneqq |\{d_0, \ldots, d_{n-1}\}|$. Algorithm E proceeds as follows.

For all $d \in \{d_0, \ldots, d_{n-1}\}$
    Repeat n_d times
        Run $\langle P^*, V \rangle$
        If V rejects, return $\perp$
        Let y denote the message V sent in the first protocol round
        Repeat $m+1$ times
            Rewind P^* to the step right after it sends C_f
            $x \leftarrow_{\$} \mathbb{F}$
            Run $\langle P^*, V \rangle$ with y and x as first messages from V
            If V rejects, return $\perp$
            Rewind P^* to the step right after V sends x
            Run E_{DEG} on P^*
            Rewind P^* to the step right after DEG ends
            Run E_{EVAL} on P^*
            If extract fails in either of the two steps, return $\perp$
            If the two extracted polynomials are not equal, return $\perp$
        End Repeat
    End Repeat
End For

Consider the event in which E does not abort, i.e., return $\perp$.

For every (y, x) generated, E_{DEG} and E_{EVAL} return an opening $\zeta_{y,x}$ (the same) to

$$C_f - \sum_{k=0}^{n-1} y^k x^{d^* - d_k + 1} C_k = C_f - \sum_{d \in \{d_0, \ldots, d_{n-1}\}} x^{d^* - d + 1} \sum_{k \colon d_k = d} y^k C_k$$

such that $\zeta_{y,x}(x) = 0$ and $\deg \zeta_{y,x} \leq d^* + 1$.

For a given y, there are $m+1$ values x which are generated. Consider the matrix consisting of rows $\left(1, \left(x^{d^*-d+1}\right)_{d \in \{d_0, \ldots, d_{n-1}\}}\right)$ for each value of x. Regarding its determinant as a polynomial in each value x generated, it is of total degree at most $(d^*+1)(m+1)$, and it is non-zero because the monomial consisting of the product of its diagonal terms appears exactly once in the polynomial. The polynomial-identity lemma then implies that the matrix is singular with probability at most $(d^*+1)(m+1)/|\mathbb{F}|$. The probability that the matrix is singular for some y is thus at most $\sum_d n_d(d^*+1)(m+1)/|\mathbb{F}| = n(d^*+1)(m+1)/|\mathbb{F}|$.

In case the matrix is invertible for all y, algorithm E can recover, for all y, an opening ζ_y to $\sum_{k:\ d_k=d} y^k C_k$ via linear combination of the openings $\zeta_{x,y}$, and an opening f_y to C_f.

Similarly, for a given $d \in \{d_0, \ldots, d_{n-1}\}$, the determinant of the matrix consisting of rows $\left(y^k\right)_{k:\ d_k=d}$ for each of the n_d values y generated has total degree at most $n_d(n-1)$. It is also non-zero because the monomial consisting of the product of its diagonal terms appears exactly once in the polynomial. By the polynomial-identity lemma, the matrix is singular with probability at most $n_d(n-1)/|\mathbb{F}|$. Consequently, the probability that the matrix is singular for some d is at most $n(n-1)/|\mathbb{F}|$. If the matrix is invertible for all d, algorithm E can recover, for all d, an opening f_k to each C_k for all k such that $d_k = d$ given openings ζ_y.

Although the choices of the extracted polynomials f_k depend on the values x and y, the distributions of the extracted polynomials f_k are independent of the values of x and y as explained above. Similarly, for a given y, the choice of the extracted polynomial f_y depends on the values x but its distribution is independent of x. Besides, for all y, $f_y = \sum_k y^k X^{d^*-d_k+1} f_k$ unless the binding property of the scheme is not satisfied, which occurs with probability at most $\varepsilon_{\mathrm{COM}}$, if the supremal advantage of any expected PPT algorithm in the binding game of the commitment scheme is at most $\varepsilon_{\mathrm{COM}}$. The arguments given in the preamble of Section 5.3 then apply.

That is, for a given y, if $\deg f_y > d^* + 1$ or $f_y - \sum_k y^k X^{d^*-d_k+1} f_k \neq 0$, then V accepts an interaction with a uniformly random x with probability at most

$$\begin{aligned}&(d^* - \min d_k + 1)/|\mathbb{F}| + (N_{\max} - 1) \cdot (d^* - \min d_k + 1)/|\mathbb{F}| \\ =\ & N_{\max}(d^* - \min d_k + 1)/|\mathbb{F}|.\end{aligned}$$

Therefore, the probability that V accepts $m+1$ interactions although $\deg f_y > d^* + 1$ or $f - \sum_k y^k X^{d^*-d_k+1} f_k \neq 0$ is at most

$$\left(N_{\max}(d^* - \min d_k + 1)/|F|\right)^{m+1}.$$

Furthermore, by Lemma 5.1 applied to $X^{d^*-d_0+1} f_0, \ldots, X^{d^*-d_{n-1}+1} f_{n-1}$, the probability over the choice of y that $\deg\left(\sum_k y^k X^{d^*-d_k+1} f_k\right) \leq d^* + 1$ although $\deg f_k \leq d_k$ for some $k \in \{0, \ldots, n-1\}$ is at most $n/|\mathbb{F}|$. Therefore, the probability that for all d, for all n_d values y generated uniformly at random, $\deg\left(\sum_k y^k X^{d^*-d_k+1} f_k\right) \leq d^* + 1$ although $\deg f_k \leq d_k$ for some $k \in \{0, \ldots, n-1\}$ is at most

$$\prod_d (n/|\mathbb{F}|)^{n_d} = (n/|\mathbb{F}|)^n.$$

It remains to analyse the probability that E never aborts. Each initial run of $\langle \mathrm{P}^*, \mathrm{V}\rangle$ succeeds with probability p_{P^*} by assumption. E aborts due to a failed initial execution with probability at most $1 - \prod_d p_{\mathrm{P}^*}^{n_d} = 1 - p_{\mathrm{P}^*}^n$.

Moreover, by an averaging argument (sometimes referred to as a heavy-row argument), conditioned on the event that y leads to a successful execution, for any real value $0 < \rho < 1$, with probability at least $(1-\rho)$ over the choice of y, the probability that P^* convinces the verifier in a protocol execution with the first message being y is at least $\rho \cdot p_{\mathrm{P}^*}$. For $\rho \leftarrow 1/2$, with probability at least $1/2$ over the choice of y, algorithm P^* succeeds with probability at least $p_{\mathrm{P}^*}/2$, conditioned on the first message being y. Consequently, the probability that P^* succeeds with y as first message given that y leads to a success is at least $(1/2)\cdot(p_{\mathrm{P}^*}/2) = p_{\mathrm{P}^*}/4$.

Similarly, with probability at least $1/2$ over the choice of x, algorithm P^* succeeds with probability at least $p_{\mathrm{P}^*}/8$, conditioned on the first and second messages from the verifier being y and x. On this account, the probability that P^* succeeds with y and x as first messages from the verifier, conditioned on the event that they lead to a success, is at least $p_{\mathrm{P}^*}/16$. It means that E aborts due to a failed execution with y and x as first messages from the verifier with probability at most $1 - \prod_d (p_{\mathrm{P}^*}/16)^{n_d(m+1)} = 1 - (p_{\mathrm{P}^*}/16)^{n(m+1)}$.

$p_{\mathrm{P}^*}/16 \geq \max(\kappa_{\mathrm{Deg}}, \kappa_{\mathrm{Eval}})$ by assumption, so for a given d, y and x, extraction fails with probability at most

$$1 - (p_{\mathrm{P}^*}/16 - \kappa_{\mathrm{Deg}})/p_{\mathrm{Deg}} + 1 - (p_{\mathrm{P}^*}/16 - \kappa_{\mathrm{Eval}})/p_{\mathrm{Eval}}.$$

If the supremal advantage of any expected PPT algorithm in the binding game of the commitment scheme is at most $\varepsilon_{\mathrm{Com}}$, the extracted polynomials are not the same with probability at most $\varepsilon_{\mathrm{Com}}$. It implies that E succeeds in the Deg extraction, the Eval extraction, and the extracted polynomials are equal with probability at least

$$\begin{aligned}
&\left(1 - \left(1 - \frac{(p_{\mathrm{P}^*}/16 - \kappa_{\mathrm{Deg}})}{p_{\mathrm{Deg}}} + 1 - \frac{(p_{\mathrm{P}^*}/16 - \kappa_{\mathrm{Eval}})}{p_{\mathrm{Eval}}} + \varepsilon_{\mathrm{Com}}\right)\right)^{n(m+1)} \\
\geq\ & \exp\left(-n(m+1)\frac{\varepsilon}{1-\varepsilon}\right) \\
=\ & \exp\left(-n(m+1)\cdot\varepsilon\left(1 + \frac{\varepsilon}{1-\varepsilon}\right)\right)
\end{aligned}$$

with

$$\varepsilon \coloneqq 1 - \frac{(p_{\mathrm{P}^*}/16 - \kappa_{\mathrm{Deg}})}{p_{\mathrm{Deg}}} + 1 - \frac{(p_{\mathrm{P}^*}/16 - \kappa_{\mathrm{Eval}})}{p_{\mathrm{Eval}}} + \varepsilon_{\mathrm{Com}}.$$

That is because for any real number $\varepsilon \neq 1$,

$$1 - \varepsilon = \frac{1-\varepsilon}{1-\varepsilon+\varepsilon} = \left(1 + \frac{\varepsilon}{1-\varepsilon}\right)^{-1} \geq \exp\left(-\frac{\varepsilon}{1-\varepsilon}\right),$$

with the last inequality implied by the convexity of function exp.

In other words, E aborts because of a failed Deg extraction, a failed Eval extraction or non-equal extracted polynomials with probability at most

$$\begin{aligned}
&1 - \exp\left(-n(m+1)\cdot\varepsilon\left(1 + \frac{\varepsilon}{1-\varepsilon}\right)\right) \\
&= 1 - \exp\left(-n(m+1)\cdot\varepsilon\,(1 + o(1))\right) \quad \text{as } \varepsilon \to 0.
\end{aligned}$$

Overall, E extracts valid witness $f_0, \dots, f_{n-1}$ with probability at least

$$\begin{aligned}1 &- n(m+1)\varepsilon_{\textsc{Com}} - n(d^*+1)(m+1)/|\mathbb{F}| - n(n-1)/|\mathbb{F}| \\ &- (N_{\max}(d^* - \min d_k + 1)/|F|)^{m+1} - (n/|\mathbb{F}|)^n \\ &- (1 - p_{\mathrm{P}^*}^n) - \left(1 - (p_{\mathrm{P}^*}/16)^{n(m+1)}\right) \\ &- \left(1 - \exp\left(-n(m+1) \cdot \varepsilon \left(1 + \frac{\varepsilon}{1-\varepsilon}\right)\right)\right).\end{aligned}$$

If necessary, algorithm E can be repeated enough times (up to a bound polynomial in λ) to make this probability as close to 1 as desired.

To simulate the transcript of a protocol execution with V, assuming Com to be hiding and Deg and Eval to be honest-verifier zero-knowledge, it suffices to generate y and x independently and uniformly at random, send a commitment to a dummy value in lieu of C_f, and to run the simulators for Deg and Eval.

5.5 Batched KZG Evaluations with Degree Check

Lemma 5.1 and Lemma 5.2 below give a straightforward way to batch the evaluation proofs with degree check from Section 5.2 if the evaluation point and the degree bound is the same for all polynomials.

Lemma 5.2. *Let m be a positive integer. Consider $f_0, \dots, f_{m-1} \in \mathbb{F}[X]$ as well as $u, v_0, \dots, v_{n-1} \in \mathbb{F}$. If $f_i(u) \neq v_i$ for some $i \in \{0, \dots, m-1\}$, then there are most $m-1$ values $y \in \mathbb{F}$ such that $\sum_i y^i(f_i - v_i) = 0 \bmod (X-u)$.*

Proof. The polynomials $\sum_i Y^i(f_i - v_i)$ and $\sum_{i:\, f_i(u) \neq v_i} Y^i(f_i - v_i)$ have the same image under the quotient map

$$\mathbb{F}[X,Y] \to (\mathbb{F}[X]/(X-u))\,[Y].$$

That image is non-zero as by assumption, $X-u$ does not divide $f_i - v_i$ for some i. Since the ideal $(X-u)\cdot\mathbb{F}[X]$ is maximal in $\mathbb{F}[X]$, the integral ring $\mathbb{F}[X]/(X-u)$ is a field, so the image of $\sum_{i:\, f_i(u) \neq v_i} Y^i(f_i - v_i)$ under the above quotient map is a polynomial of degree at most $m-1$. Therefore, it has at most $m-1$ roots in $\mathbb{F}[X]/(X-u)$, and thus also in $\mathbb{F}$ because the restriction on $\mathbb{F}$ of the quotient map $\mathbb{F}[X] \to \mathbb{F}[X]/(X-u)$ is injective. □

More precisely, for integers $m \geq 1$ and $0 \leq d < N_{\max}$, for committed polynomials $f_0, \dots, f_{m-1}$ with respective commitments $C_0, \dots, C_{m-1}$, and for elements $u \in \mathbb{F}^*, v_0, \dots, v_{m-1} \in \mathbb{F}$, proving that

$$f_0(u) = v_0, \dots, f_{m-1}(u) = v_{m-1} \text{ and } \forall i \in \{0, \dots, m-1\}, \deg(f_i) \leq d,$$

is equivalent to proving that

$$\forall i \in \{0, \dots, m-1\}, \deg(f_i) \leq d, \exists q_i \in \mathbb{F}[X], f_i - v_i - (X-u)q_i = 0.$$

The idea is then to run the protocol from Section 5.2 with $\sum_i y^i C_i$ as commitment, u as evaluation point, $\sum_i y^i v_i$ as target value, $\sum_i y^i f_i$ as witness and $\left[\sum_i y^i q_i\right]_1$ as proof, for a uniformly random $y \in \mathbb{F}$ chosen by the verifier.

If $\deg f_i \geq d$ or $f_i(u) \neq v_i$ for some $0 \leq i \leq m-1$ with non-negligible probability, given adversarial polynomials $f_0, \ldots, f_{m-1}$ extracted from $C_0, \ldots, C_{m-1}$ and a proof, then the probability that a uniformly random value y chosen by the verifier is in both the set of values $\alpha \in \mathbb{F}$ such that $\deg(\sum_i \alpha^i f_i) \leq d$ and the set of values $\alpha \in \mathbb{F}$ such that $\sum_i \alpha^i (f_i(u) - v_i) = 0$ is negligible if $m/|\mathbb{F}|$ is. Indeed, if $\deg f_i \geq d$ for some i, then the probability that y is in the set of values $\alpha \in \mathbb{F}$ such that $\deg(\sum_i \alpha^i f_i) \leq d$ is at most $(m-1)/|\mathbb{F}|$ by Lemma 5.1. If $f_i(u) \neq v_i$ for some i, then the probability that y is in the set of values $\alpha \in \mathbb{F}$ such that $\sum_i \alpha^i (f_i - v_i) = 0 \bmod (X - u)$ is at most $(m-1)/|\mathbb{F}|$ by Lemma 5.2. Therefore, by Lemma 2.1, if an adversary could compute with non-negligible probability commitments to polynomials $f_0, \ldots, f_{m-1}$, as well as values $u, v_0, \ldots, v_{n-1}$ such that $\deg f_i \geq d$ or $f_i(u) \neq v$ for some i, then the soundness of the protocol from Section 5.2 implies that the probability that the verifier accepts is negligible. That is to say, if the verifier accepts, then with overwhelming probability, $\deg f_i \leq d$ and $f_i(u) = v_i$ for all $i \in \{0, \ldots, m-1\}$.

Formal Description. The protocol below is for the following language.

$$\left\{((C_i, v_i)_{i=0}^{m-1}, 0 < d < N_{\max}, u \neq 0) \colon \forall i, \exists f_i \in \mathbb{F}[X]^{\leq d},\ \text{KZG.}\textsc{Open}(C_i, f_i) = 1,\ f_i(u) = v_i\right\}.$$

$\mathbf{P} \leftarrow \mathbf{V}:\ y \leftarrow_{\$} \mathbb{F}$

$\mathbf{P} \rightarrow \mathbf{V}:$

$s \leftarrow_{\$} \mathbb{F}$

$\pi \leftarrow \left[\sum_i y^i q_i(\tau) \tau^{N_{\max}-d}\right]_1 + s \cdot [\xi]_1$, for q_i s.t. $f_i - v_i = (X - u) q_i$

$\delta \leftarrow \left(\sum_i y^i r_i\right) \cdot \left[\tau^{N_{\max}-d}\right]_1 - s \cdot [\tau]_1 + (s \cdot u) \cdot [1]_1$

Output (π, δ)

$\mathbf{V}:$

$C \leftarrow \sum_i y^i C_i$

$e\left(C - (\sum_i y^i v_i) \cdot [1]_1, \left[\tau^{N_{\max}-d}\right]_2\right) \stackrel{?}{=} e\left(\pi, [\tau]_2 - u \cdot [1]_2\right) + e(\delta, [\xi]_2).$

Properties. The completeness of the protocol stems from the homomorphic property of KZG commitments and the completeness of the protocol from Section 5.2.

As for its knowledge soundness, the knowledge soundness of the protocol in Section 5.2 guarantees that a valid opening f to the commitment $\sum_{i=0}^{m-1} y^i C_i$ such that $\deg f \leq d$ and $f(u) = \sum_i y^i v_i$, can be extracted in the algebraic-group model given a valid proof π. For openings $f_0, \ldots, f_{m-1}$ to $C_0, \ldots, C_{m-1}$ extracted from an algebraic adversary, it follows that f and $\sum_{i=0}^{m-1} f_i$ are both valid openings to $\sum_{i=0}^{m-1} y^i f_i$. If f were not equal to $\sum_{i=0}^{m-1} y^i f_i$, then the trapdoor τ would be a root of their difference and could be recovered by factorisation [14, 27].

On this account, under the $(N_{\max}-1)$-DLOG assumption (and a fortiori under the $(2N_{\max}-1)$-DLOG assumption), they are both equal with overwhelming probability. The reasoning given in the preamble then allows to conclude that $\deg f_i \leq d$ and $f_i(u) = v_i$ for all $i \in \{0, \ldots, m-1\}$ with overwhelming probability.

Given the trapdoor (τ, ξ), a proof on commitments $C_0, \ldots, C_{m-1}$ can be simulated by choosing $y \leftarrow_\$ \mathbb{F}$, $\pi \leftarrow_\$ \mathbb{G}_1$ and computing

$$\delta \leftarrow \left(\tau^{N_{\max}-d} \sum_i y^i (C_i - v_i) - \pi(\tau - u)\right) \xi^{-1}.$$

6 Instantiation with Hiding KZG Commitments

This section instantiates the protocol from Section 4 with hiding KZG commitments. This results in a scheme in which the evaluation prover uses only $n+2$ random field elements for n-linear polynomials, an exponential improvement w.r.t. traditional methods which require 2^n random nonces. That is made possible by the idea of checking a multivariate Euclidian-division equation (equivalent to correctness of the evaluation) via its image under the univariatisation map.

To improve efficiency, the instantiation uses the protocol from Section 5.5 to batch the degree checks on the openings to commitments C_k. This protocol ends by a check that a polynomial $\zeta_{x'}$, defined by a random x' chosen by the verifier, satisfies $\zeta_{x'}(x') = 0$ and $\deg(\zeta_{x'}) \leq d^* + 1$ for any $d^* \geq 2^{n-1} - 1 = \deg\left(\mathcal{U}_n(q_{n-1})^{<2^{n-1}}\right)$.

Moreover, since replacing the last check that $Z_x(x) = 0$ in the generic protocol with a check that $Z_x(x) = 0$ and $\deg Z_x \leq 2^n - 1$ does not change the knowledge soundness of the scheme, the instantiation does so, uses $x \leftarrow x'$, and then using the protocol from Section 5.5, batches this check with the check that $\zeta_{x'}(x') = 0$ and $\deg(\zeta_{x'}) \leq d^* + 1$, for $d^* := 2^n - 2$. This means that the verifier ultimately need only do three pairing computations.

Unfortunately, setting $x \leftarrow x'$ implies that the knowledge soundness of the instantiation does not directly follows from that of the generic protocol and of the protocol in Section 5.5, and must be argued anew.

Lastly, the following lemma shows that the degree check on the input commitment C can be omitted since an opening polynomial can always be extracted from an algebraic adversary.

Lemma 6.1. *Let n be a positive integer. Consider polynomials $\hat{f}, \hat{q}_0, \ldots, \hat{q}_{n-1} \in \mathbb{F}[X]$ and a tuple $\boldsymbol{u} \in \mathbb{F}^n$. Let $f := \mathcal{U}_n^{-1}\left(\hat{f}^{<2^n}\right)$. If the equality*

$$\hat{f} - v \cdot \Phi_n(X) = \sum_k \left(X^{2^k} \Phi_{n-k-1}\left(X^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(X^{2^k}\right)\right) \hat{q}_k$$

holds and $\deg(\hat{q}_k) < 2^k$ for all $0 \leq i < n$, then $\deg\left(\hat{f}\right) < 2^n$ (hence $\hat{f} = \hat{f}^{<2^n} = \mathcal{U}_n(f)$) and $f(\boldsymbol{u}) = v$.

Proof. By definition, $\deg(\Phi_{n-k-1}) = 2^{n-k-1} - 1$, so $\deg\left(\Phi_{n-k-1}\left(X^{2^{k+1}}\right)\right) = 2^n - 2^{k+1}$, and $\deg\left(X^{2^k}\Phi_{n-k-1}\left(X^{2^{k+1}}\right)\right) = 2^n - 2^k$. Since $\deg(\hat{q}_k) \leq 2^k - 1$ by assumption, the equality implies that $\deg \hat{f} \leq 2^n - 1$.

Besides, Lemmas 2.5.1 and 2.5.2, and Corollary 2.5.3.2 show that the equality holds if and only if $f - v = \sum_k (X_k - u_k)\mathcal{U}_n^{-1}(\hat{q}_k)$, which implies that $f(\boldsymbol{u}) = v$. $\square$

Building Blocks The scheme assumes the existence of a generator Gen of bilinear group structures. In what follows, KZG denotes the univariate commitment scheme recalled in Section 3.5.3.

Formal Description.

Setup $\left(1^\lambda, n_{\max} =: \log_2 N_{\max}\right)$:

$\mathbb{G} \coloneqq (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow \textsc{Gen}\left(1^\lambda\right)$

$\tau, \xi \leftarrow_\$ \mathbb{F}^*$

$srs \leftarrow \left([1]_1, [\tau]_1, \ldots, \left[\tau^{N_{\max}-1}\right]_1, [\xi]_1, [1]_2, [\tau]_2, \ldots, \left[\tau^{N_{\max}-1}\right]_2, [\xi]_2\right)$

Return $par \leftarrow (\mathbb{G}, srs)$.

Com (f, n) : Return $(C, r) \leftarrow \mathrm{KZG}.\textsc{Com}\left(\mathcal{U}_n(f)\right)$.

Open $(C, f, n, r) \rightarrow b \in \{0, 1\}$: Return $b \leftarrow \mathrm{KZG}.\textsc{Open}\left(C, \mathcal{U}_n(f), r\right)$.

Eval :

P $(C, \boldsymbol{u} = (u_0, \ldots, u_{n-1}), v, f) \rightarrow$ **V**$(C, \boldsymbol{u}, v)$:

For $(k = 0, \ldots, n-1)\left\{(C_k, r_k) \leftarrow \mathrm{KZG}.\textsc{Com}\left(\mathcal{U}_n(q_k)^{<2^k}\right)\right\}$

Output $(C_0, \ldots, C_{n-1})$ $\qquad d_k \coloneqq 2^k - 1$

P $\leftarrow$ **V** : $y \leftarrow_\$ \mathbb{F}$

P $\rightarrow$ **V** :

$(C_{\hat{q}}, \hat{r}) \leftarrow \mathrm{KZG}.\textsc{Com}\left(\hat{q} \coloneqq \sum_{k=0}^{n-1} y^k X^{2^n - d_k - 1}\mathcal{U}_n(q_k)^{<2^k}\right)$

Output $C_{\hat{q}}$

P $\leftarrow$ **V** :

$x \leftarrow_\$ \mathbb{F}^*, z \leftarrow_\$ \mathbb{F}$

$(C_{v,x}, 0) \leftarrow \mathrm{KZG}.\textsc{Com}\left(v \cdot \Phi_n(x); 0\right)$

$C_{Z_x} \leftarrow C - C_{v,x} - \sum_k \left(x^{2^k}\Phi_{n-k-1}\left(x^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(x^{2^k}\right)\right) \cdot C_k$

Output (x, z)

P $\rightarrow$ **V** :

$r_Z \leftarrow r - \sum_k \left(x^{2^k}\Phi_{n-k-1}\left(x^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(x^{2^k}\right)\right) \cdot r_k$

$r_\zeta \leftarrow \hat{r} - \sum_{k=0}^{n-1} y^k x^{2^n - d_k - 1} r_k$

Compute q_ζ and q_Z such that $\zeta_x = (X - x)q_\zeta$ and $Z_x = (X - x)q_Z$

$\pi \leftarrow \left[(q_\zeta(\tau) + z \cdot q_Z(\tau))\tau^{N_{\max}-(2^n-1)}\right]_1 + s \cdot [\xi]_1$ for $s \leftarrow_\$ \mathbb{F}$

$\delta \leftarrow (r_\zeta + zr_Z) \cdot \left[\tau^{N_{\max}-(2^n-1)}\right]_1 - s \cdot [\tau]_1 + (s \cdot x) \cdot [1]_1$

Output (π, δ)

V :

$C_{\zeta_x} \leftarrow C_{\hat{q}} - \sum_{k=0}^{n-1} y^k x^{2^n-d_k-1} C_k$

$C_{\zeta,Z} \leftarrow C_{\zeta_x} + z \cdot C_{Z_x}$

$e\left(C_{\zeta,Z}, \left[\tau^{N_{\max}-(2^n-1)}\right]_2\right) \stackrel{?}{=} e\left(\pi, [\tau]_2 - x \cdot [1]_2\right) + e\left(\delta, [\xi]_2\right).$

Properties. The completeness of the evaluation protocol follows from the completeness of the generic protocol in Section 4, and of the protocols in Sections 5.3 and 5.5.

As for the knowledge soundness of the protocol, given a commitment C computed by an algebraic adversary, a polynomial $\hat{f} \in \mathbb{F}[X]$ of degree at most $N_{\max} - 1$ can be extracted. Likewise, a polynomial $\hat{q} \in \mathbb{F}[X]$ and polynomials $\hat{q}_k \in \mathbb{F}[X]$, each of degree at most $N_{\max} - 1$, can be extracted from $C_{\hat{q}}$ and C_k for all $k \in \{0, \ldots, n-1\}$. The knowledge soundness of the protocol in Section 5.5 implies that polynomials ζ_x and Z_x of degree at most $2^n - 1$ such that $\zeta_x(x) = Z_x(x) = 0$ can be extracted from C_{ζ_x} and C_{Z_x}. Under the $(N_{\max}-1)$-DLOG assumption (and therefore also under the $(2N_{\max}-1)$-DLOG assumption),

$$\zeta_x = \hat{q} - \sum_{k=0}^{n-1} y^k x^{2^n-d_k-1} \hat{q}_k$$

and

$$\begin{aligned} Z_x = &\hat{f} - v \cdot \Phi_n(x) \\ &- \sum_k \left(x^{2^k} \Phi_{n-k-1}\left(x^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(x^{2^k}\right)\right) \hat{q}_k. \end{aligned}$$

Similarly to the analysis of the protocol in Section 5.3, if $\deg \hat{f} > 2^n - 1$ or $\hat{q} - \sum_k y^k X^{2^n-d_k-1} \hat{q}_k \neq 0$ or

$$\hat{f} \neq v \cdot \Phi_n(X) + \sum_k \left(X^{2^k} \Phi_{n-k-1}\left(X^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(X^{2^k}\right)\right) \hat{q}_k,$$

(this last "or" condition precisely comes from the fact that the same value x was used to define both ζ_x and Z_x), then by Lemma 2.1, the probability that the verifier accepts, i.e., the probability that the value x chosen by the verifier is a root of $\hat{q} - \sum_k y^k X^{2^n-d_k-1} \hat{q}_k$, a root of

$$\hat{f} - v \cdot \Phi_n(X) - \sum_k \left(X^{2^k} \Phi_{n-k-1}\left(X^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(X^{2^k}\right)\right) \hat{q}_k$$

and in the set of $\alpha \in \mathbb{F}$ such that $\deg\left(\hat{q} - \sum_k y^k \alpha^{2^n - d_k - 1} \hat{q}_k\right) \leq 2^n - 1$ is negligible. That is, if the verifier accepts, then with overwhelming probability $\deg \hat{q}_k \leq d_k = 2^k - 1$ for all k by the analysis of knowledge soundness of the protocol in Section 5.3, and in addition to that,

$$\hat{f} - v \cdot \Phi_n(X) = \sum_k \left(X^{2^k} \Phi_{n-k-1}\left(X^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(X^{2^k}\right)\right) \hat{q}_k.$$

Lemma 6.1 shows that $f \coloneqq \mathcal{U}_n^{-1}\left(\hat{f}\right)$ is such that $f(\boldsymbol{u}) = v$. The multilinear polynomial f is thus a valid witness.

The honest-verifier zero-knowledge property of the protocol follows from that of the generic construction in Section 4 and of the protocols in Section 5.

Efficiency. The protocol consists of 5 rounds of interaction during which the prover sends $n + 3$ first-group elements and the verifier sends 3 field elements. The other costs are summarised in Table 2. The details are as follows.

Computational Costs. The prover computes a $\mathbb{G}_1$ Multi-Scalar Multiplication (MSM) of size 2^k for each commitment C_k. It also computes an MSM of size $N/2$ to commit to $\hat{q}$ as the latter has at most $N/2 - 1$ non-zero coefficients between degree $2^{n-1} - 1 = 2^n - d_{n-1} - 1$ and degree $2^n - 1$. Another MSM of size $N = 2^n$ is necessary to compute π and a last one of size 3 to compute δ. As shown in Section A.2, the prover can compute the coefficients of all $\mathcal{U}_n(q_k)^{<2^k}$ in $2^{n+1} - 3$ additions and $2^n - 2$ multiplications. Given that Φ_n can be computed in $n+1$ multiplications and two additions (see Section 2.1), and that Ruffini's rule can be used to compute q_Z and q_ζ, the prover can compute the coefficients for all these MSMs in $O(N)$ field operations with small constants.

The verifier computes $\mathbb{G}_1$ MSMs of sizes 1, $n+2$, $n+1$ and 2 to respectively compute $C_{v,x}$, C_{Z_x}, C_{ζ_x} and $C_{\zeta,Z}$. Four of the coefficients are 1, so the verifier does at most $2n+2$ first-group scalar multiplications. It also performs a scalar multiplication and an addition in $\mathbb{G}_2$ to compute $[\tau]_2 - x \cdot [1]_2$, and finally 3 pairing computations.

Randomness Complexity. The prover generates n uniformly random field elements for each commitment C_k, another one to compute $C_{\hat{q}}$ and a last one to compute π. The verifier generates three random nonces x, y and z.

Memory Costs. The memory costs on Table 2 do not take into account the input polynomial, commitment, evaluation point and claimed evaluation. They do not take into account the SRS either. Note however that the prover need not store any $\mathbb{G}_2$ element from the SRS to compute evaluation proofs. As for the verifier, it must only store $[1]_1$ and $[\xi]_1$ (to compute $C_{v,x}$), and $[1]_2, [\tau]_2, [\xi]_2$ and $\left[\tau^{N_{\max} - (2^n - 1)}\right]_2$ for all $n \in \{1, \ldots, \log_2 N_{\max}\}$.

Throughout the protocol, in addition to the field elements x, y, z received from the verifier, the prover must store the coefficient of $\mathcal{U}_n(q_k)$ (of degree at most $2^k - 1$) for all $k \in \{0, \ldots, n-1\}$, of $\hat{q}$ (of degree at most $N/2 - 1$), of q_Z (of degree at most $N-2$) and of q_ζ (of degree at most $N/2-2$). That is $3N-3$ field elements.

The verifier must store the commitments $C_0, \ldots, C_{n-1}$, $C_{\hat{q}}$ and the proof (π, δ) in addition to its own challenges x, y, z.

	$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{F}$ ops.	$e(\cdot,\cdot)$	Rand. ($\mathbb{F}$)	Mem.
P	$\sum_{k=0}^{n} 1 \cdot 2^k + 1 \cdot N/2 + 1 \cdot 3$	✘	$O(N)$	✘	$n+2$	$\geq (3N-3) \cdot \mathbb{F}$
V	$1 \cdot 1 + 1 \cdot (n+2) + 1 \cdot (n+1) + 1 \cdot 2$	$1 \cdot 2$	$O(n)$	3	3	$\geq (n+3) \cdot \mathbb{G} + 3 \cdot \mathbb{F}$

Table 2. Costs for the prover and verifier. $N =: 2^n$ is the number of coefficients of the input polynomial. The group operations are counted in terms of Multi-Scalar Multiplications (MSMs). For positive integers k, ℓ, n, m, the notation $k \cdot n + \ell \cdot m$ means that the party must perform k MSMs of size n and ℓ MSMs of size m. The randomness costs are given in terms of uniformly random field elements that the parties must generate. The memory costs do not account for the inputs to the parties, they only account for the costs of the field or group elements exchanged during the protocol and that the parties must maintain, once they are received, throughout the protocol to complete their computations.

7 Shift Evaluations

Recent SNARGs for arithmetic circuit-satisfiability leverage look-up arguments to improve efficiency. These allow provers to commit to the input and output of a function computed by a sub-part of the circuit, and to prove that they are in a pre-computed, public table of the function. This may in practice result in significant savings if computing the function is more resource-intensive than proving that the input-output pair is in the table, and even more so when the sub-circuit is repeated throughout the overall circuit.

Some of these look-up arguments, e.g., Plookup [12], require to prove evaluations of polynomials represented by a shift of witness values. That is, if $\boldsymbol{a} = (a_0, \ldots, a_{N-1})$ represents witness values, these may be interpreted as the evaluations over $\{0,1\}^{\log N}$ of a multilinear polynomial f in $\log N$ variables. The scheme in Section 4 gives a protocol to evaluate f at any point $\boldsymbol{u} \in \mathbb{F}^{\log N}$. If $\boldsymbol{a}_{\leftarrow}$ denotes the vector $(a_1, \ldots, a_{N-1}, a_0)$, then these lookup arguments require to prove evaluations of the multilinear polynomial $f_{\leftarrow}$ represent by $\boldsymbol{a}_{\leftarrow}$.

It is possible to commit to f and $f_{\leftarrow}$ and prove that the latter is the shift of the first, and then separately prove evaluations of these polynomials. A more efficient alternative is to only commit to f, and use this commitment to prove evaluations of $f_{\leftarrow}$. This section covers this second approach.

Outline. Given $n \in \mathbb{N}_{\geq 1}$ (let $N \coloneqq 2^n$) and the evaluations $\boldsymbol{a} \in \mathbb{F}^{2^n}$ over $\{0,1\}^n$ of a polynomial $f \in \mathbb{F}[X_0, \ldots, X_{n-1}]^{\preceq 1}$, with

$$a_{i_{n-1} \cdot 2^{n-1} + \cdots + i_0 \cdot 2^0} \coloneqq f(\boldsymbol{i}),$$

note that the polynomial $f_{\leftarrow}$ corresponding to $\boldsymbol{a}_{\leftarrow} \coloneqq (a_1, \ldots, a_{N-1}, a_0)$ satisfies the identity

$$X \cdot \mathcal{U}_n \left(f_{\leftarrow}\right) = \mathcal{U}_n \left(f\right) - a_0 + a_0 X^N.$$

To prove that $f_{\leftarrow}(\boldsymbol{u}) = v$ for public $\boldsymbol{u} \in \mathbb{F}^n$ and $v \in \mathbb{F}$, by Lemma 2.3.1, it suffices to prove the existence of $q_{f_{\leftarrow},k} \in \mathbb{F}[X_0, \ldots, X_{k-1}]^{\preceq 1}$ for all $0 < k < n$ and $q_{f_{\leftarrow},0} \in \mathbb{F}$ such that $f_{\leftarrow} - v = \sum_{k=0}^{n-1} (X_k - u_k) q_{f_{\leftarrow},k}$. From these two identities, and since $\mathcal{U}_n$ is an isomorphism, it is sufficient to prove that

$$\mathcal{U}_n(f) - a_0 + a_0 X^N - X \cdot \mathcal{U}_n(v) = X \cdot \left(\sum_{k=0}^{n-1} \mathcal{U}_n \left(X_k q_{f_{\leftarrow},k}\right) - u_k \mathcal{U}_n \left(q_{f_{\leftarrow},k}\right) \right).$$

Lemmas 2.5.1 and 2.5.2, and Corollary 2.5.3.2 show that it is equivalent to prove that

$$\begin{aligned} &\mathcal{U}_n(f) - a_0 + a_0 X^N - X \cdot \mathcal{U}_n(v) \\ &= X \cdot \sum_k \left(X^{2^k} \Phi_{n-k-1} \left(X^{2^{k+1}} \right) - u_k \cdot \Phi_{n-k} \left(X^{2^k} \right) \right) \mathcal{U}_n(q_{f_{\leftarrow},k})^{<2^k}. \end{aligned}$$

Given a univariate commitment to $\mathcal{U}_n(f)$, which is a commitment to f in the generic construction of Section 4, the idea of the protocol for shift evaluations is to test the above polynomial identity in the same manner as in the generic protocol. However, the verifier is not given a_0, but it can be assumed without loss of practical generality that this value is always 0. Indeed, if $\boldsymbol{a}$ represents the values of all the left inputs or all the right inputs or all the outputs across the entire circuit (as in the Plonk arithmetisation [13]), adding a leading 0 to all input and output vectors corresponds to adding a dummy addition gate with 0 as inputs and 0 as output.

Remark. *Alternatively, if it is possible with the univariate scheme to prove committed evaluations*[9]*, i.e., to prove knowledge of an opening v to a public commitment and that it is the evaluation of a committed polynomial at a public point, and if the set of polynomial commitments and the set of committed evaluations are the same, then the prover can, in the first round of the protocol, send a commitment to a_0 and prove that it knows an opening that is the evaluation of the input polynomial at 0. The prover also sends a commitment to $a_0 X^N$ and proves consistency with the commitment to a_0. The verifier can leverage the homomorphic property of the commitments to proceed as in the generic protocol.*

[9] A straightforward way to do so with hiding KZG commitments is as follows. To prove that $f - v = (X - u)q$ while keeping v private, the prover can compute a

Assuming that $a_0 = 0$, the polynomial identity to check is

$$\begin{aligned}&\mathcal{U}_n(f) - X \cdot \mathcal{U}_n(v)\\ &= X \cdot \sum_k \left(X^{2^k} \Phi_{n-k-1}\left(X^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(X^{2^k}\right)\right) \mathcal{U}_n(q_{f_{\leftarrow},k})^{<2^k}.\end{aligned}$$

A straightforward adaptation of the evaluation protocols in Sections 4 and 6 leads to a protocol to evaluate $f_{\leftarrow}$ given a commitment to any multilinear polynomial f. The only difference is that in the first rounds of those protocols, the prover commits in C_k to $\mathcal{U}_n\left(q_{f_{\leftarrow},k}\right)^{<2^k}$ instead of $\mathcal{U}_n\left(q_k\right)^{<2^k}$, and that the prover now computes $C_{v,x}$ as a commitment to $v \cdot x\Phi_n(x)$ and C_{Z_x} as

$$C - C_{v,x} - x \cdot \sum_k \left(x^{2^k} \Phi_{n-k-1}\left(x^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(x^{2^k}\right)\right) \cdot C_k.$$

The proof of knowledge soundness is close to that of the protocol in Section 6.

High-Degree Shifts. Assuming the first d coefficients to be zero, i.e., $a_0 = \cdots = a_{d-1} = 0$, the construction is readily adapted to shifts of degree d, i.e., to prove evaluations of the polynomial defined by $(a_d, \ldots, a_{N-1}, a_0, \ldots, a_{d-1})$. This assumption is once again attained in practice by adding to the execution trace as many dummy addition gates as necessary.

8 Batching Standard and Shift Evaluations

The look-up techniques mentioned in Section 7 and which require evaluations of shifted polynomials typically require evaluations of a committed polynomial and its shift *at the same point*. This section then gives a technique to batch regular evaluations with those of shifted polynomials (assuming once again that the constant terms are zero) at the same point, in case the underlying univariate commitment is KZG.

The main idea is to multiply by variable X the polynomial identity that is checked for regular evaluations, so that the right-hand side of the equality has the same form as in the identity for shifted evaluations. More concretely, the identity that is now checked for regular evaluations is

$$\begin{aligned}&X\left(\mathcal{U}_n(f) - v \cdot \Phi_n(X)\right)\\ &= X \cdot \sum_k \left(X^{2^k} \Phi_{n-k-1}\left(X^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(X^{2^k}\right)\right) \mathcal{U}_n(q_k)^{<2^k},\end{aligned}$$

KZG commitment C_v to v with randomness r_v, and in addition to a KZG proof π for the evaluation of f, the prover also shows that C_v is a commitment to a constant polynomial with the proof system from Section 5.1, i.e., the prover sends also $\pi_v \leftarrow \left[v \cdot \tau^{N_{\max}-1}\right]_1 + s_v \cdot [1]_1$ for $s_v \leftarrow_{\$} \mathbb{F}$ and $\delta_v \leftarrow r_v \cdot \left[\tau^{N_{\max}-1}\right] - s_v \cdot [1]_1$. The verifier then checks that $e\left(C - C_v, [1]_2\right) = e\left([q(\tau)]_1, [\tau]_2 - u \cdot [1]_2\right)$ and that $e\left(C_v, \left[\tau^{N_{\max}-1}\right]_2\right) = e\left(\pi_v, [1]_2\right) + e\left(\delta_v, [\xi]_2\right)$.

and recall from Section 7 that the identity for shifted evaluations is

$$\begin{aligned}&\mathcal{U}_n(f) - v \cdot X\Phi_n(X)\\ &= X \cdot \sum_k \left(X^{2^k}\Phi_{n-k-1}\left(X^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(X^{2^k}\right)\right)\mathcal{U}_n(q_{f_{\leftarrow},k})^{<2^k}.\end{aligned}$$

Outline. Given a positive integer n and non-negative integers m and ℓ, commitments $C_0, \dots, C_{m-1}$ and $D_0, \dots, D_{\ell-1}$ to multilinear polynomials $f_0, \dots, f_{m-1}$ and $g_0, \dots, g_{\ell-1}$ in $\mathbb{F}[X_0, \dots, X_{n-1}]^{\preceq 1}$, an evaluation point $u \in \mathbb{F}^n$ and claimed evaluations $v_0, \dots, v_{m-1}$ and $w_0, \dots, w_{m-1}$ in $\mathbb{F}$, to prove that $f_i(u) = v_i$ for all $0 \leq i \leq m-1$ and that $g_{i,\leftarrow}(u) = w_i$ for all $0 \leq i \leq \ell-1$, the idea is to consider each term

$$f_i - v_i - \sum_k (X_k - u_k) q_{f_i,k}$$

as the coefficients of degree $0 \leq i \leq m-1$ of a polynomial in a variable X_n, and each term

$$g_{i,\leftarrow} - w_i - \sum_k (X_k - u_k) q_{g_{i,\leftarrow},k}$$

as the coefficients of degree $m \leq i \leq m+\ell-1$ of the same polynomial. The latter should be the zero polynomial (in X_n), so the verifier tests it by checking it at a random point $\alpha \in \mathbb{F}$, i.e., the verifier sends it in the first round of the protocol. The verifier accepts if and only if the prover can show that the evaluation of $f \coloneqq \sum_{i=0}^{m-1} \alpha^i f_i + \sum_{i=0}^{\ell-1} \alpha^{m+i} g_{i,\leftarrow}$ is $v \coloneqq \sum_{i=0}^{m-1} \alpha^i v_i + \sum_{i=0}^{\ell-1} \alpha^{m+i} w_i$, i.e., if the prover can show that there exist polynomials q_k such that $f - v = \sum_k (X_k - u_k) q_k$. Note that the verifier can compute a commitment to f given the input commitments, and that by uniqueness of polynomials q_k (c.f. Lemma 2.3.1), $q_k = \sum_{i=0}^{m-1} \alpha^i q_{f_i,k} + \sum_{i=0}^{\ell-1} \alpha^{m+i} q_{g_{i,\leftarrow},k}$.

Formal Description. The protocol is for the following language (algorithm Open denotes the algorithm from Section 6).

$$\begin{aligned}\big\{(C_i, v_i)_{i=0}^{m-1}, (D_j, w_j)_{j=0}^{\ell-1}, \boldsymbol{u} \colon \forall i \exists (f_i, r_i), \forall j \exists (g_j, r_j), \textsc{Open}(C_i, f_i, r_i) = 1,\\ \textsc{Open}(D_j, g_j, r_j) = 1, f_i(\boldsymbol{u}) = v_i, g_{j,\leftarrow}(\boldsymbol{u}) = w_j \ \big\}.\end{aligned}$$

P $\leftarrow$ **V** :

$\alpha \leftarrow_{\$} \mathbb{F}$

$v \leftarrow \sum_{i=0}^{m-1} \alpha^i v_i + \sum_{i=0}^{\ell-1} \alpha^{m+i} w_i$

Output α

P $\rightarrow$ **V** :

Compute $q_0, \dots, q_{n-1}$ such that $f - v = \sum_k (X_k - u_k) q_k$

For $(k = 0, \dots, n-1) \left\{(C_{\hat{q}_k}, r_k) \leftarrow \text{KZG.Com}\left(\mathcal{U}_n(q_k)^{<2^k}\right)\right\}$

Output $\left(C_{\hat{q}_0}, \ldots, C_{\hat{q}_{n-1}}\right)$ $\quad d_k := 2^k - 1$

P $\leftarrow$ **V** : $y \leftarrow_\$ \mathbb{F}$

P $\rightarrow$ **V** :

$(C_{\hat{q}}, \hat{r}) \leftarrow \text{KZG.Com}\left(\hat{q} := \sum_{k=0}^{n-1} y^k X^{2^n - d_k - 1} \mathcal{U}_n(q_k)^{<2^k}\right)$

Output $C_{\hat{q}}$

P $\leftarrow$ **V** :

$x \leftarrow_\$ \mathbb{F}^*, z \leftarrow_\$ \mathbb{F}$

$(C_{v,x}, 0) \leftarrow \text{KZG.Com}\,(v \cdot x\Phi_n(x); 0)$

$C \leftarrow x \cdot \sum_{i=0}^{m-1} \alpha^i C_i + \sum_{i=0}^{\ell-1} \alpha^{m+i} D_i$

$C_{Z_x} \leftarrow C - C_{v,x} - x \cdot \sum_k \left(x^{2^k} \Phi_{n-k-1}\left(x^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(x^{2^k}\right)\right) \cdot C_{\hat{q}_k}$

Output (x, z)

P $\rightarrow$ **V** :

$r \leftarrow x \cdot \sum_{i=0}^{m-1} \alpha^i r_i + \sum_{i=0}^{\ell-1} \alpha^{m+i} r_i$

$r_Z \leftarrow r - x \cdot \sum_k \left(x^{2^k} \Phi_{n-k-1}\left(x^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(x^{2^k}\right)\right) \cdot r_k$

$r_\zeta \leftarrow \hat{r} - \sum_{k=0}^{n-1} y^k x^{2^n - d_k - 1} r_k$

Compute q_ζ and q_Z such that $\zeta_x = (X - x) q_\zeta$ and $Z_x = (X - x) q_Z$

$\pi \leftarrow \left[(q_\zeta(\tau) + z \cdot q_Z(\tau)) \tau^{N_{\max} - (2^n - 1)}\right]_1 + s \cdot [\xi]_1$ for $s \leftarrow_\$ \mathbb{F}$

$\delta \leftarrow (r_\zeta + z r_Z) \cdot \left[\tau^{N_{\max} - (2^n - 1)}\right]_1 - s \cdot [\tau]_1 + (s \cdot x) \cdot [1]_1$

Output (π, δ)

V :

$C_{\zeta_x} \leftarrow C_{\hat{q}} - \sum_{k=0}^{n-1} y^k x^{2^n - d_k - 1} C_{\hat{q}_k}$

$C_{\zeta,Z} \leftarrow C_{\zeta_x} + z \cdot C_{Z_x}$

$e\left(C_{\zeta,Z}, \left[\tau^{N_{\max} - (2^n - 1)}\right]_2\right) \stackrel{?}{=} e\left(\pi, [\tau]_2 - x \cdot [1]_2\right) + e\left(\delta, [\xi]_2\right).$

Properties. The completeness of the evaluation protocol follows by construction.

Similarly to the proof of knowledge soundness of the protocol in Section 6, given a valid proof from an algebraic adversary, with overwhelming probability under the $(N_{\max} - 1)$-DLOG assumption, univariate polynomials $\hat{f}_0, \ldots, \hat{f}_{m-1}$ and $\hat{h}_0, \ldots, \hat{h}_{\ell-1}$, and $\hat{q}_0, \ldots, \hat{q}_{n-1}$ can be extracted from $C_0, \ldots, C_{m-1}$ and $D_0, \ldots, D_{\ell-1}$, and $C_{\hat{q}_0}, \ldots, C_{\hat{q}_{n-1}}$, and are such that $\deg \hat{q}_k \leq 2^k - 1$ and

$$X \sum_{i=0}^{m-1} \alpha^i \hat{f}_i + \sum_{i=0}^{\ell-1} \alpha^{m+i} \hat{h}_i$$
$$= v \cdot X\Phi_n(X) + X \sum_k \left(X^{2^k} \Phi_{n-k-1}\left(X^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(X^{2^k}\right)\right) \hat{q}_k.$$

This polynomial identity implies that X divides $\sum_{i=0}^{\ell-1} \alpha^{m+i} h_i$, i.e., there exists a univariate polynomial $\hat{g}$ such that $\sum_{i=0}^{\ell-1} \alpha^{m+i} \hat{h}_i = X\hat{g}$, which implies that $\sum_{i=0}^{\ell-1} \alpha^{m+i} \hat{h}_i(0) = 0$.

If there exists $0 \leq i \leq \ell - 1$ such that $\hat{h}_i(0) \neq 0$, then there at most $m + \ell - 1$ values of $\alpha \in \mathbb{F}$ such that $\sum_{i=0}^{\ell-1} \alpha^{m+i} \hat{h}_i(0) = 0$ because the polynomial $X^m \sum_{i=0}^{\ell-1} \hat{h}_i(0) X^i \in \mathbb{F}[X]$ has at most ℓ roots.

In the event that $h_i(0) = 0$ for all i, i.e., there exists $\hat{g}_i$ such that $\hat{h}_i = X\hat{g}_i$ for all i, then denoting

$$\begin{aligned} \hat{f} &\coloneqq \sum_{i=0}^{m-1} \alpha^i \hat{f}_i + \sum_{i=0}^{\ell-1} \alpha^{m+i} \hat{g}_i \\ &= v \cdot \Phi_n(X) + \sum_k \left(X^{2^k} \Phi_{n-k-1}\left(X^{2^{k+1}}\right) - u_k \cdot \Phi_{n-k}\left(X^{2^k}\right)\right) \hat{q}_k, \end{aligned}$$

Lemma 6.1 shows that $\deg \hat{f} \leq 2^n - 1$ and $\mathcal{U}_n^{-1}\left(\hat{f}\right)(\boldsymbol{u}) = v = \sum_{i=0}^{m-1} \alpha^i v_i + \sum_{i=0}^{\ell-1} \alpha^{m+i} w_i$.

Besides, Lemma 5.1 shows that there are at most $m + \ell - 1$ values of α such that $\deg \hat{f} \leq 2^n - 1$ if $\deg \hat{f}_i \geq 2^n$ for some $0 \leq i \leq m - 1$ or $\deg \hat{g}_i \geq 2^n$ for some $0 \leq i \leq \ell - 1$.

Moreover, there are at most $m+\ell-1$ values of $\alpha \in \mathbb{F}$ such that $\mathcal{U}_n^{-1}\left(\hat{f}\right)(u) = v$ if $\mathcal{U}_n^{-1}\left(\hat{f}_i\right)(u) \neq v_i$ for some $0 \leq i \leq m - 1$ or $\mathcal{U}_n^{-1}(\hat{g}_i)(u) \neq w_i$ for some $0 \leq i \leq \ell - 1$.

Lemma A.1 then implies that with overwhelming probability, $\mathcal{U}_n^{-1}\left(\hat{f}_i\right)$ and $\mathcal{U}_n^{-1}(\hat{g}_i)$ (and their respected randomness extracted from the commitments) are valid witnesses.

Evaluations proofs can be simulated in a similar way as for the proofs in Section 6.

Acknowledgements. The authors thank Ariel Gabizon, Adrian Hamelink and Zachary J. Williamson for helpful discussions, and Sergei Iakovenko for corrections. Many thanks to Dimitrios Papadopoulos for helpful discussions about the zero-knowledge vSQL polynomial-delegation scheme [33, Proof of Theorem 1].

References

1. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Fast reed-solomon interactive oracle proofs of proximity. In: Chatzigiannakis, I., Kaklamanis, C., Marx, D., Sannella, D. (eds.) ICALP 2018. LIPIcs, vol. 107, pp. 14:1–14:17. Schloss Dagstuhl (Jul 2018). `https://doi.org/10.4230/LIPIcs.ICALP.2018.14`
2. Ben-Sasson, E., Chiesa, A., Gabizon, A., Riabzev, M., Spooner, N.: Interactive oracle proofs with constant rate and query complexity. In: Chatzigiannakis, I., Indyk, P., Kuhn, F., Muscholl, A. (eds.) ICALP 2017. LIPIcs, vol. 80, pp. 40:1–40:15. Schloss Dagstuhl (Jul 2017). `https://doi.org/10.4230/LIPIcs.ICALP.2017.40`

3. Ben-Sasson, E., Chiesa, A., Riabzev, M., Spooner, N., Virza, M., Ward, N.P.: Aurora: Transparent succinct arguments for R1CS. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part I. LNCS, vol. 11476, pp. 103–128. Springer, Heidelberg (May 2019). https://doi.org/10.1007/978-3-030-17653-2_4
4. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (May 2004). https://doi.org/10.1007/978-3-540-24676-3_4
5. Boneh, D., Drake, J., Fisch, B., Gabizon, A.: Efficient polynomial commitment schemes for multiple points and polynomials. Cryptology ePrint Archive, Report 2020/081 (2020), https://eprint.iacr.org/2020/081
6. Bootle, J., Chiesa, A., Groth, J.: Linear-time arguments with sublinear verification from tensor codes. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part II. LNCS, vol. 12551, pp. 19–46. Springer, Heidelberg (Nov 2020). https://doi.org/10.1007/978-3-030-64378-2_2
7. Bootle, J., Chiesa, A., Hu, Y., Orrù, M.: Gemini: Elastic SNARKs for diverse environments. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part II. LNCS, vol. 13276, pp. 427–457. Springer, Heidelberg (May / Jun 2022). https://doi.org/10.1007/978-3-031-07085-3_15
8. Bootle, J., Chiesa, A., Liu, S.: Zero-knowledge IOPs with linear-time prover and polylogarithmic-time verifier. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part II. LNCS, vol. 13276, pp. 275–304. Springer, Heidelberg (May / Jun 2022). https://doi.org/10.1007/978-3-031-07085-3_10
9. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE Symposium on Security and Privacy. pp. 315–334. IEEE Computer Society Press (May 2018). https://doi.org/10.1109/SP.2018.00020
10. Chen, B., Bünz, B., Boneh, D., Zhang, Z.: HyperPlonk: Plonk with linear-time prover and high-degree custom gates. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part II. LNCS, vol. 14005, pp. 499–530. Springer, Heidelberg (Apr 2023). https://doi.org/10.1007/978-3-031-30617-4_17
11. Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 33–62. Springer, Heidelberg (Aug 2018). https://doi.org/10.1007/978-3-319-96881-0_2
12. Gabizon, A., Williamson, Z.J.: plookup: A simplified polynomial protocol for lookup tables. Cryptology ePrint Archive, Report 2020/315 (2020), https://eprint.iacr.org/2020/315
13. Gabizon, A., Williamson, Z.J., Ciobotaru, O.: PLONK: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Report 2019/953 (2019), https://eprint.iacr.org/2019/953
14. von zur Gathen, J., Shoup, V.: Computing frobenius maps and factoring polynomials (extended abstract). In: 24th ACM STOC. pp. 97–105. ACM Press (May 1992). https://doi.org/10.1145/129712.129722
15. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct NIZKs without PCPs. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 626–645. Springer, Heidelberg (May 2013). https://doi.org/10.1007/978-3-642-38348-9_37
16. Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: One-time programs. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 39–56. Springer, Heidelberg (Aug 2008). https://doi.org/10.1007/978-3-540-85174-5_3

17. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal on Computing **18**(1), 186–208 (1989)
18. Groth, J.: Short pairing-based non-interactive zero-knowledge arguments. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 321–340. Springer, Heidelberg (Dec 2010). https://doi.org/10.1007/978-3-642-17373-8_19
19. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-size commitments to polynomials and their applications. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 177–194. Springer, Heidelberg (Dec 2010). https://doi.org/10.1007/978-3-642-17373-8_11
20. Lee, J.: Dory: Efficient, transparent arguments for generalised inner products and polynomial commitments. In: Nissim, K., Waters, B. (eds.) TCC 2021, Part II. LNCS, vol. 13043, pp. 1–34. Springer, Heidelberg (Nov 2021). https://doi.org/10.1007/978-3-030-90453-1_1
21. Lund, C., Fortnow, L., Karloff, H.J., Nisan, N.: Algebraic methods for interactive proof systems. In: 31st FOCS. pp. 2–10. IEEE Computer Society Press (Oct 1990). https://doi.org/10.1109/FSCS.1990.89518
22. Maller, M., Bowe, S., Kohlweiss, M., Meiklejohn, S.: Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019. pp. 2111–2128. ACM Press (Nov 2019). https://doi.org/10.1145/3319535.3339817
23. Paillier, P., Vergnaud, D.: Discrete-log-based signatures may not be equivalent to discrete log. In: Roy, B.K. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 1–20. Springer, Heidelberg (Dec 2005). https://doi.org/10.1007/11593447_1
24. Papamanthou, C., Shi, E., Tamassia, R.: Signatures of correct computation. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 222–242. Springer, Heidelberg (Mar 2013). https://doi.org/10.1007/978-3-642-36594-2_13
25. Ron-Zewi, N., Rothblum, R.D.: Local proofs approaching the witness length [extended abstract]. In: 61st FOCS. pp. 846–857. IEEE Computer Society Press (Nov 2020). https://doi.org/10.1109/FOCS46700.2020.00083
26. Setty, S.: Spartan: Efficient and general-purpose zkSNARKs without trusted setup. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 704–737. Springer, Heidelberg (Aug 2020). https://doi.org/10.1007/978-3-030-56877-1_25
27. Shoup, V.: Factoring polynomials over finite fields: Asymptotic complexity vs. reality. In: Proceedings of the IMACS Symposium (1993)
28. Thaler, J.: Time-optimal interactive proofs for circuit evaluation. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 71–89. Springer, Heidelberg (Aug 2013). https://doi.org/10.1007/978-3-642-40084-1_5
29. Wahby, R.S., Tzialla, I., shelat, a., Thaler, J., Walfish, M.: Doubly-efficient zkSNARKs without trusted setup. In: 2018 IEEE Symposium on Security and Privacy. pp. 926–943. IEEE Computer Society Press (May 2018). https://doi.org/10.1109/SP.2018.00060
30. Xie, T., Zhang, J., Zhang, Y., Papamanthou, C., Song, D.: Libra: Succinct zero-knowledge proofs with optimal prover computation. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 733–764. Springer, Heidelberg (Aug 2019). https://doi.org/10.1007/978-3-030-26954-8_24
31. Zhang, J., Liu, T., Wang, W., Zhang, Y., Song, D., Xie, X., Zhang, Y.: Doubly efficient interactive proofs for general arithmetic circuits with linear prover time. In: Vigna, G., Shi, E. (eds.) ACM CCS 2021. pp. 159–177. ACM Press (Nov 2021). https://doi.org/10.1145/3460120.3484767

32. Zhang, J., Xie, T., Zhang, Y., Song, D.: Transparent polynomial delegation and its applications to zero knowledge proof. In: 2020 IEEE Symposium on Security and Privacy. pp. 859–876. IEEE Computer Society Press (May 2020). https://doi.org/10.1109/SP40000.2020.00052
33. Zhang, Y., Genkin, D., Katz, J., Papadopoulos, D., Papamanthou, C.: A zero-knowledge version of vSQL. Cryptology ePrint Archive, Report 2017/1146 (2017), https://eprint.iacr.org/2017/1146

A Mathematical Preliminaries

A.1 Conditional Probabilities

Proof (of Lemma 2.1). As $P[E_0 \cup \cdots \cup E_{n-1}] > 0$ by hypothesis and

$$P[E_0] + \cdots + P[E_{n-1}] \geq P[E_0 \cup \cdots \cup E_{n-1}],$$

there exists $0 \leq i \leq n-1$ such that $P[E_i] > 0$.

By definition of conditional probability,

$$P[\cap_i H_i | \cup_i E_i] = P[(\cap_i H_i) \cap (\cup_i E_i)]/P[\cup_i E_i].$$

Therefore,

$$\begin{aligned} P[\cap_i H_i | \cup_i E_i] &\leq \sum_{i:\ P[E_i] \neq 0} P[(H_0 \cap \cdots \cap H_{n-1}) \cap E_i]/P[E_0 \cup \cdots \cup E_{n-1}] \\ &\leq \sum_{i:\ P[E_i] \neq 0} P[(H_0 \cap \cdots \cap H_{n-1}) \cap E_i]/P[E_i] \\ &\leq \sum_{i:\ P[E_i] \neq 0} P[H_i \cap E_i]/P[E_i] \\ &= \sum_{i:\ P[E_i] \neq 0} P[H_i | E_i]. \end{aligned}$$

□

Lemma A.1. *Let n be a positive integer and $E_0, \ldots, E_{n-1}$, $H_0, \ldots, H_{n-1}$ denote probability events in a discrete probability space. Suppose that $P[E_0 \cup \cdots \cup E_{n-1}] > 0$. Then, for any $0 \leq i_0 \leq n-1$ such that $P[E_{i_0}] > 0$,*

$$P[\cap_i H_i | \cup_i E_i] \leq P[H_{i_0} | E_{i_0}] + \sum_{i \neq i_0:\ P[E_i \overline{E_{i_0}}] \neq 0} P[H_i | E_i, \overline{E_{i_0}}].$$

Proof.

$$
\begin{aligned}
P[\cap_i H_i | \cup_i E_i] &= P[\cap_i H_i \cap E_{i_0} \cup \left(\cup_{i \neq i_0} E_i \cap \overline{E_{i_0}}\right)]/P[\cup_i E_i] \\
&\leq P[H_{i_0}|E_{i_0}] + \sum_{i \neq i_0 \colon P[E_i \cap \overline{E_{i_0}}] \neq 0} P[H_i \cap E_i \cap \overline{E_{i_0}}]/P[\cup_{j=0}^{n-1} E_j] \\
&\leq P[H_{i_0}|E_{i_0}] \\
&+ \sum_{i \neq i_0 \colon P[E_i \cap \overline{E_{i_0}}] \neq 0} P[\cap_i H_i \cap E_i \cap \overline{E_{i_0}}]/P\left[E_{i_0} \cup \left(\cup_{j \neq i_0} E_j \cap \overline{E_{i_0}}\right)\right] \\
&\leq P[H_{i_0}|E_{i_0}] + \sum_{i \neq i_0 \colon P[E_i \cap \overline{E_{i_0}}] \neq 0} P[\cap_i H_i \cap E_i \cap \overline{E_{i_0}}]/P[E_i \cap \overline{E_{i_0}}] \\
&P[H_{i_0}|E_{i_0}] + \sum_{i \neq i_0 \colon P[E_i \overline{E_{i_0}}] \neq 0} P[H_i | E_i, \overline{E_{i_0}}].
\end{aligned}
$$

□

A.2 Computation of the Quotient Polynomials.

Lemma 2.3.1 gives an expression for the q_k polynomials in terms of f and $\boldsymbol{u}$. The following lemma and corollary give an explicit method to compute evaluations of these polynomials on $\{0,1\}^n$, which is sufficient to compute their images under $\mathcal{U}_n$.

Lemma A.2.1. *Let f be an n-linear polynomial with $n > 1$. Consider $\boldsymbol{u} \in \mathbb{F}^n$. Let $q_0 \in \mathbb{F}$ and $q_k \in \mathbb{F}[X_0, \ldots, X_{k-1}]^{\preceq 1}$ for $k \in \{1, \ldots, n-1\}$ be such that $f - f(\boldsymbol{u}) = \sum_k (X_k - u_k) q_k$. Define $f_0 \coloneqq f$ and for any $k = n-2$ down to 0,*

$$f_{n-1-k} \coloneqq f - \sum_{\ell=k+1}^{n-1} (X_\ell - u_\ell) q_\ell.$$

Then, for any $k \in \{0, \ldots, n-1\}$,

$$q_k = f_{n-1-k}\left(\boldsymbol{X}_{<k}, u_k + 1, \boldsymbol{X}_{>k}\right) - f_{n-1-k}\left(\boldsymbol{X}_{<k}, u_k, \boldsymbol{X}_{>k}\right).$$

Proof. By definition,

$$f_{n-1-k} = \sum_{\ell=0}^{k} (X_\ell - u_\ell) q_\ell + f(\boldsymbol{u}).$$

Since $q_\ell \in \mathbb{F}[X_0, \ldots, X_{\ell-1}]^{\preceq 1}$, it implies that

$$f_{n-1-k}\left(\boldsymbol{X}_{<k}, u_k + 1, \boldsymbol{X}_{>k}\right) = \sum_{\ell=0}^{k-1} (X_\ell - u_\ell) q_\ell + q_k + f(\boldsymbol{u})$$

and

$$f_{n-1-k}\left(\boldsymbol{X}_{<k}, u_k, \boldsymbol{X}_{>k}\right) = \sum_{\ell=0}^{k-1}(X_\ell - u_\ell)q_\ell + f(\boldsymbol{u}).$$

The difference between the two equalities yields the result. □

Lemma A.2.2. *For all $k \in \{0, \dots, n-1\}$, for all $\boldsymbol{i} \in \{0,1\}^n$,*

$$q_k(\boldsymbol{i}) = f_{n-1-k}(\boldsymbol{i}_{<k}, 1, \boldsymbol{i}_{>k}) - f_{n-1-k}(\boldsymbol{i}_{<k}, 0, \boldsymbol{i}_{>k}).$$

Proof. Lemma A.2.1 implies that

$$q_k(\boldsymbol{i}) = f_{n-1-k}\left(\boldsymbol{i}_{<k}, u_k + 1, \boldsymbol{i}_{>k}\right) - f_{n-1-k}\left(\boldsymbol{i}_{<k}, u_k, \boldsymbol{i}_{>k}\right).$$

Writing f_{n-1-k} as $\sum_{\boldsymbol{j}\in\{0,1\}^n} f_{n-1-k}(\boldsymbol{j})L_{\boldsymbol{j}}$, for $\boldsymbol{i} \in \{0,1\}^n$,

$$f_{n-1-k}\left(\boldsymbol{i}_{<k}, u_k + 1, \boldsymbol{i}_{>k}\right) = \sum_{\boldsymbol{j}\in\{0,1\}^n} f_{n-1-k}(\boldsymbol{j})L_{\boldsymbol{j}}\left(\boldsymbol{i}_{<k}, u_k + 1, \boldsymbol{i}_{>k}\right).$$

Notice that $L_{\boldsymbol{j}}\left(\boldsymbol{i}_{<k}, u_k + 1, \boldsymbol{i}_{>k}\right)$ is zero unless $\boldsymbol{i}_{<k} = \boldsymbol{j}_{<k}$ and $\boldsymbol{i}_{>k} = \boldsymbol{j}_{>k}$. Therefore,

$$\begin{aligned}
&f_{n-1-k}\left(\boldsymbol{i}_{<k}, u_k + 1, \boldsymbol{i}_{>k}\right)\\
=&f_{n-1-k}\left(\boldsymbol{i}_{<k}, 1, \boldsymbol{i}_{>k}\right) L_{(\boldsymbol{i}_{<k},1,\boldsymbol{i}_{>k})}\left(\boldsymbol{i}_{<k}, u_k + 1, \boldsymbol{i}_{>k}\right)\\
+&f_{n-1-k}\left(\boldsymbol{i}_{<k}, 0, \boldsymbol{i}_{>k}\right) L_{(\boldsymbol{i}_{<k},0,\boldsymbol{i}_{>k})}\left(\boldsymbol{i}_{<k}, u_k + 1, \boldsymbol{i}_{>k}\right).
\end{aligned}$$

Similarly,

$$\begin{aligned}
f_{n-1-k}\left(\boldsymbol{i}_{<k}, u_k, \boldsymbol{i}_{>k}\right) =&f_{n-1-k}\left(\boldsymbol{i}_{<k}, 1, \boldsymbol{i}_{>k}\right) L_{(\boldsymbol{i}_{<k},1,\boldsymbol{i}_{>k})}\left(\boldsymbol{i}_{<k}, u_k, \boldsymbol{i}_{>k}\right)\\
&+f_{n-1-k}\left(\boldsymbol{i}_{<k}, 0, \boldsymbol{i}_{>k}\right) L_{(\boldsymbol{i}_{<k},0,\boldsymbol{i}_{>k})}\left(\boldsymbol{i}_{<k}, u_k, \boldsymbol{i}_{>k}\right).
\end{aligned}$$

The difference of the two equalities and the observation that

$$\begin{aligned}
&L_{(\boldsymbol{i}_{<k},j_k,\boldsymbol{i}_{>k})}\left(\boldsymbol{i}_{<k}, u_k + 1, \boldsymbol{i}_{>k}\right) - L_{(\boldsymbol{i}_{<k},j_k,\boldsymbol{i}_{>k})}\left(\boldsymbol{i}_{<k}, u_k, \boldsymbol{i}_{>k}\right)\\
=&\left(j_k + (-1)(1 - j_k)\right)
\end{aligned}$$

for any $j_k \in \{0,1\}$ imply that

$$q_k(\boldsymbol{i}) = f_k(\boldsymbol{i}_{<k}, 1, \boldsymbol{i}_{>k}) - f_k(\boldsymbol{i}_{<k}, 0, \boldsymbol{i}_{>k}).$$

□

Corollary A.2.2.1. *For all $k \in \{0, \dots, n-1\}$, for all $\boldsymbol{i} \in \{0,1\}^k$,*

$$q_k\left(\boldsymbol{i}, \boldsymbol{0}^{n-k}\right) = f_{n-1-k}\left(\boldsymbol{i}, 1, \boldsymbol{0}\right) - f_{n-1-k}\left(\boldsymbol{i}, 0, \boldsymbol{0}\right).$$

Moreover, if $k < n-1$, for any $i_k \in \{0,1\}$,

$$f_{n-1-k}(\boldsymbol{i}, i_k, \boldsymbol{0}) = f_{n-1-(k+1)}(\boldsymbol{i}, i_k, 0) + u_{k+1}q_{k+1}(\boldsymbol{i}, i_k, \boldsymbol{0}).$$

Proof. Lemma A.2.2 implies that

$$q_k\left(\boldsymbol{i},\mathbf{0}^{n-k}\right) = f_{n-1-k}\left(\boldsymbol{i},1,\mathbf{0}\right) - f_{n-1-k}\left(\boldsymbol{i},0,\mathbf{0}\right).$$

In addition to that, for $k < n-1$, by definition of f_{n-1-k} and $f_{n-1-(k+1)}$,

$$f_{n-1-(k+1)} - f_{n-1-k} = (X_{k+1} - u_{k+1})q_{k+1}.$$

Therefore, for any $\boldsymbol{j} \in \{0,1\}^n$ such that $j_{k+1} = 0$,

$$f_{n-1-k}(\boldsymbol{j}) = f_{n-1-(k+1)}(\boldsymbol{j}) + u_{k+1}q_{k+1}(\boldsymbol{j}).$$

Applying this equality to $(\boldsymbol{i},0,\mathbf{0})$ and $(\boldsymbol{i},1,\mathbf{0})$ gives the claim. □

Computation. The coefficient of the $\mathcal{U}_n(q_k)^{<2^k}$ polynomials for all $0 \le k \le n-1$ can be computed as follows, in light of the expression in Lemma A.2.2 for the evaluations of q_{n-1} in terms of f and the relation in Corollary A.2.2.1 between the evaluations of q_k and q_{k+1}.

By definition, the coefficients of $\mathcal{U}_n(q_k)^{<2^k}$ are the evaluations $q_k\left(\boldsymbol{i},\mathbf{0}^{n-k}\right)$ for all $\boldsymbol{i} \in \{0,1\}^k$. Given the $f(\boldsymbol{i})$ for all $\boldsymbol{i} \in \{0,1\}^n$ as input, the coefficients of $\mathcal{U}_n(q_k)^{<2^k}$ can be computed as follows.

For $k = n-1$ down to 0
    For $\boldsymbol{i} \in \{0,1\}^k$
        If $k = n-1$
            $q_{n-1}(\boldsymbol{i},0) \leftarrow f_0(\boldsymbol{i},1) - f_0(\boldsymbol{i},0)$
        Else
            $q_k\left(\boldsymbol{i},\mathbf{0}^{n-k}\right) \leftarrow f_{n-1-k}\left(\boldsymbol{i},1,\mathbf{0}\right) - f_{n-1-k}\left(\boldsymbol{i},0,\mathbf{0}\right)$
        End If
        If $k > 0$
            $f_{n-1-(k-1)}(\boldsymbol{i},\mathbf{0}) \leftarrow f_{n-1-k}(\boldsymbol{i},\mathbf{0}) + u_{k+1}q_{k+1}(\boldsymbol{i},\mathbf{0})$
        End If
    End For
End For

Costs. In the above algorithm, for all $\boldsymbol{i} \in \{0,1\}^{n-1}$, computing $q_{n-1}(\boldsymbol{i},0)$ requires 1 addition, and computing $f_1(\boldsymbol{i},0)$ requires 1 addition and 1 multiplication. For $1 \le k \le n-2$, for all $\boldsymbol{i} \in \{0,1\}^k$, computing $q_k\left(\boldsymbol{i},\mathbf{0}^{n-k}\right)$ entails 1 addition, and computing $f_{n-1-(k-1)}(\boldsymbol{i},\mathbf{0})$ requires 1 addition and 1 multiplication. For $k = 0$, computing q_0 requires 1 addition. In total, that is $2^{n-1} * 2 + \sum_{k=1}^{n-2} 2^k * 2 + 1 = 2^n + 2 * \left(2^{n-1} - 2\right) + 1 = 2^{n+1} - 3$ additions and $2^{n-1} + \sum_{k=1}^{n-2} 2^k + 1 = 2^{n-1} + \left(2^{n-1} - 2\right) + 1 = 2^n - 2$ multiplications.

B Cryptographic Preliminaries

B.1 Proof Systems

B.1.1 Properties. The properties of proof systems mentioned in Section 3.4 are defined as follows.

Soundness. Formally, a proof system is sound if for all pairs $(\mathrm{A}, \mathrm{P}^*)$ of PPT algorithms,

$$P\left[\mathrm{V}_{\langle \mathrm{P}^*(st), \cdot \rangle}(x) = 1, x \notin L_R \colon \begin{array}{c} R \leftarrow \mathrm{R}\left(1^\lambda\right) \\ (par, \tau) \leftarrow \textsc{Setup}\left(R\right) \\ (st, x) \leftarrow \mathrm{A}(par) \end{array}\right]$$

is a negligible function of λ.

Knowledge Soundess. A proof system is knowledge sound with error κ if for any PPT algorithm P^*, there exists an extractor E that runs in expected polynomial time and has control over the randomness of P^*, and a real polynomial p such that for any PPT algorithm A, for all $R \leftarrow \mathrm{R}\left(1^\lambda\right), (par, \tau) \leftarrow \textsc{Setup}(R)$, for any state $st \in \{0, 1\}^*$ and any instance $x \in \{0, 1\}^*$ such that $P\left[(st, x) = \mathrm{A}(par)\right]$ is non-negligible and

$$P\left[\mathrm{V}_{\langle \mathrm{P}^*(st), \cdot \rangle}(par, x) = 1\right] > \kappa(\lambda, |x|),$$

$$\begin{aligned} &P\left[(x, w) \in R \colon w \leftarrow \mathrm{E}^{\langle \mathrm{P}^*(st), \cdot \rangle}(par, x)\right] \\ &\geq \left(P\left[\mathrm{V}_{\langle \mathrm{P}^*(st), \cdot \rangle}(par, x) = 1\right] - \kappa(\lambda, |x|)\right) / p(\lambda, |x|). \end{aligned}$$

Honest-Verifier Zero-Knowledge. A proof system is honest-verifier zero-knowledge if there exists a PPT simulator S such that for any PPT algorithm A, for all $R \leftarrow \mathrm{R}\left(1^\lambda\right), (par, \tau) \leftarrow \textsc{Setup}(R)$, for any state $st \in \{0, 1\}^*$ and any instance-witness pair $(x, w) \in R$ such that $P\left[(st, x, w) = \mathrm{A}(par)\right]$ is non-negligible,

$$\left|P\left[\mathrm{A}\left(st, \{\langle P, V \rangle(par, x; w)\}\right) = 1\right] - P\left[\mathrm{A}\left(st, \mathrm{S}(par, \tau, x)\right) = 1\right]\right|$$

is a negligible function of λ.

B.2 Polynomial Commitments

B.2.1 Security Properties. A polynomial-commitment scheme is expected to satisfy the properties which follow.

Biding. A polynomial-commitment scheme is computationally binding if for all $N \in \mathbb{N}_{\geq 1}$, for any PPT algorithm A,

$$P\left[\textsc{Open}(par, C, f_i, r_i) = 1, f_0 \neq f_1 : \begin{array}{c} par \leftarrow \textsc{Setup}\left(1^\lambda, N\right) \\ (C, (f_i, r_i)_{i=0,1}) \leftarrow \mathrm{A}(par) \end{array}\right]$$

is a negligible function of λ.

A scheme is perfectly binding if for any valid commitment, there is exactly one polynomial to which it can be opened.

Hiding. A polynomial-commitment scheme is computationally (resp. statistically) hiding if for all $N \in \mathbb{N}_{\geq 1}$, for any PPT (resp. computationally unbounded) algorithm A,

$$\left| P\left[b = b' : \begin{array}{l} par \leftarrow \textsc{Setup}\left(1^\lambda, N\right) \\ (C, f_0, f_1, st) \leftarrow \mathrm{A}(par) \\ b \leftarrow_\$ \{0,1\} \\ (C, r) \leftarrow \textsc{Com}(par, f_b) \\ b' \leftarrow \mathrm{A}(st, C) \\ \text{for } \beta \in \{0,1\} \\ \quad\quad \text{if } f_\beta \in \mathbb{F}[X]^{<N} \text{ and } f_{1-\beta} \notin \mathbb{F}[X]^{<N} \\ \quad\quad\quad\quad b' \leftarrow_\$ \{0,1\} \end{array} \right] - 1/2 \right|$$

is a negligible function of λ. A scheme is perfectly hiding if it is statistically hiding and the above difference is exactly nil.

Evaluation Binding. A polynomial-commitment scheme is (computationally) *evaluation biding* if for all $N \in \mathbb{N}_{\geq 1}$, for any pair $(\mathrm{A}, \mathrm{P}^*)$ of PPT algorithms,

$$P\left[\begin{array}{c} \mathrm{V}_{\langle \mathrm{P}^*(st), \cdot \rangle}(par, C, u, v) = 1, \\ \mathrm{V}_{\langle \mathrm{P}^*(st), \cdot \rangle}(par, C, u, v') = 1, \\ v \neq v' \end{array} : \begin{array}{c} par \leftarrow \textsc{Setup}\left(1^\lambda, N\right) \\ (C, u, v, v', st) \leftarrow \mathrm{A}(par) \end{array}\right]$$

is a negligible function of λ.

B.2.2 Hiding KZG Commitments. This section first explains the flaw in the proof [33, Proof of Theorem 1] of Zhang et al. for their polynomial-delegation scheme [10]. Next comes a proof that the evaluation protocol of the hiding KZG commitment scheme (see Section 3.5.3) is knowledge sound under the q-DLOG assumption in the algebraic-group model.

[10] Note that Zhang et al. gave a proof in the standard model, but it requires an additional knowledge-of-exponent assumption [18], as mechanism to enable the extraction of polynomials from adversaries, and its effect is to double the size of the reference string and of the evaluation proofs (compared to the scheme in Section 3.5.3).

On the Proof of Zhang et al. The explanation which follows the notation and the page references of the e-print version [33] of the paper received on 2017-11-27. It is therefore not stand-alone as that would otherwise require to completely copy their proof until the critical step.

In the proof, to compute a solution to the SDH problem, the reduction algorithm computes the Euclidian division of a univariate polynomial K', and it is crucial that it is non-constant. K' is defined in terms of multivariate polynomials extracted from the adversary. To show that it is not constant, on page 9, a polynomial $f^{''}$ is defined in terms of polynomials $q'_1, ..., q'_{\ell+1}$ extracted from the adversary, and in the second equality on page 10, variable $x_{\ell+1}$ is substituted with 0.

However, polynomial K', which is assumed for the sake of contradiction to be constant, is defined for specific values of $\rho_2, ..., \rho_{\ell+1}$ chosen by the reduction algorithm and fixed in the parameters before the adversary chooses $q'_1, ..., q'_{\ell+1}$. These latter polynomials thus depend on $\rho_2, ..., \rho_{\ell+1}$. Substituting $x_{\ell+1}$ with 0 would amount to changing $\rho_{\ell+1}$ to 0, and therefore redefining these polynomials, i.e., redefining a new polynomial K', and the proof can therefore not hold.

Knowledge Soundness. The knowledge soundness can be proved under the DLOG assumption with parameter $N_{\max}$. Suppose that there exists a PPT adversary $(\mathrm{A}, \mathscr{P}^*)$ which computes with non-negligible probability an instance (C, u, v) and a valid proof (π, δ).

Since the adversary is algebraic, one can extract polynomials f, g and h of degree at most $N_{\max} - 1$ as well as field elements r, s and t such that

$$C = [f(\tau) + r\xi]_1, \quad \pi = [g(\tau) + s\xi]_1$$
$$\text{and} \;\; \delta = [h(\tau) + t\xi]_1 .$$

If $f(u) = v$, then (f, r) is a valid witness.

Otherwise, i.e., in the event that $f(u) \neq v$, consider an algorithm B that runs the adversary as sub-routine and interacts with a DLOG challenger with parameters $N_{\max}$. Upon receiving a DLOG tuple

$$\left([1]_1, [\tau]_1, \ldots, \left[\tau^{N_{\max}}\right]_1, [1]_2, [\tau]_2, \ldots, \left[\tau^{N_{\max}}\right]_2\right),$$

algorithm B chooses ρ uniformly at random in $\mathbb{F}^*$, computes $[\xi]_1 \leftarrow \rho \cdot \left[x^{N_{\max}+1}\right]_1$ and $[\xi]_2 \leftarrow \rho \cdot \left[x^{N_{\max}+1}\right]_2$ and sets the reference string as in the specification of the scheme. Note that the distribution of the SRS is the same as the one produced by the set-up algorithm.

The pair (π, δ) being a valid proof w.r.t. (C, u, v), the verification pairing equation is satisfied, which means that

$$f(\tau) + r\rho\tau^{N_{\max}} - v = \left(g(\tau) + s\rho\tau^{N_{\max}}\right)(\tau - u) + \left(h(\tau) + t\rho\tau^{N_{\max}}\right)\rho\tau^{N_{\max}}.$$

Therefore, x is a root of the polynomial

$$k(X) \coloneqq f + r\rho X^{N_{\max}} - v - \left(g + s\rho X^{N_{\max}}\right)(X - u) - \left(h + t\rho X^{N_{\max}}\right)\rho X^{N_{\max}}.$$

The degree of k is at most $2N_{\max}$, so if k is a non-zero polynomial, B can recover τ by factorisation in polynomial time [14, 27] and solve the DLOG problem. Therefore, it is enough to show that $k \neq 0$. Consider the linear map $\mathbb{F}[X,Y]^{\leq N_{\max}-1} \to \mathbb{F}_{{N_{\max}}^2-1}[X]$ that sends X^iY^j to $X^i \cdot \left(\rho X^{N_{\max}}\right)^j$ for for $0 \leq i,j \leq N_{\max}-1$. It is an isomorphism because $\rho \neq 0$ and the $N_{\max}$-ary decomposition of integers in $\left\{0,\ldots,{N_{\max}}^2-1\right\}$ is unique, i.e., any integer in this set is uniquely written as $i + N_{\max} \cdot j$ with $i,j \in \{0,\ldots,N_{\max}-1\}$. Note that

$$\ell(X,Y) \coloneqq f(X) + rY - v - (g(X) + sY)(X-u) - (h(X) + tY)\,Y$$

is the pre-image of k under this isomorphism. On this account, it is enough to show that $\ell(X,Y) \neq 0$.

The constant term of $\ell(X,Y)$ as a polynomial in $\mathbb{F}[X][Y]$ is $f - v - g(X-u)$. It cannot be zero as it would otherwise imply that $f(u) = v$. Therefore, $\ell(X,Y) \neq 0$ and B can solve the $N_{\max}$-DLOG problem.

In other words, the probability that a valid witness can be extracted is at least the probability that $(\mathrm{A}, \mathrm{P}^*)$ makes the verifier accept minus the supremal advantage of any PPT algorithm in solving the $N_{\max}$-DLOG problem. The latter assumed to be negligible, the probability that a valid witness is extracted is negligibly close to the probability that the verifier accepts.