Boomy: Batch Opening Of Multivariate polYnomial commitment

Thomas Lavaur[1,2] and Jérôme Lacan[1]

[1] ISAE-SUPAERO, University of Toulouse,France name.surname@isae-supaero.fr
[2] University Toulouse III Paul Sabatier, France

Abstract. We present Boomy, a multivariate polynomial commitment scheme enabling the proof of the evaluation of multiple points: batch openings. Boomy is the natural extension of two popular protocols: the univariate polynomial commitment scheme of Kate, Zaverucha and Goldberg [18] and its multivariate counterpart from Papamanthou, Shi and Tamassia [23]. In the special case of univariate, i.e., for only one evaluation point, Boomy matches these two previous schemes. Our construction is proven secure under the selective security model. In this paper, we present Boomy's complexity and the applications on which it can have a significant impact. In fact, Boomy is perfectly suited to tackling blockchain data availability problems, shrinking existing challenges. We also present special lower-complexity cases that occur frequently in practical situations.

1 Introduction

Polynomial commitment schemes, whether univariate or multivariate, are a key component in modern cryptography. They allow a prover to convince (with overwhelming probability) a verifier that they know a polynomial of bounded degree that evaluates at a given value, and this without revealing the polynomial itself, rendering the protocol zero-knowledge. To do that, the prover engages their polynomial through a commitment. They can later generate a proof of one or several evaluations of the same polynomial. Given the proof and the commitment, the verifier is then able to enforce the correct evaluation of the committed polynomial.

Univariate polynomial commitment protocols are used rather regularly for their simplicity and performance. They are relatively easy to set up and often offer better computation and communication complexities than their multivariate counterparts. However, univariate polynomial commitments are flexibility-limited: they cannot efficiently represent certain complex data types or mathematical relationships. Multivariate polynomial commitments are the answer to this shortcoming of univariate protocols.

Polynomial commitment schemes are beneficial to many applications such as building zero-knowledge protocols [12], secret sharing [18] or confidential cryptographic transactions [17].

1.1 Related Works and Contribution

The first polynomial commitment scheme proposed by [18] in 2010 allowed a prover to convince a verifier of the evaluation of a committed univariate polynomial at one or more points. Later, [23] built on this work, introducing multivariate polynomials, but only at a single point of evaluation, opening up new paths for new verifiable computation schemes. Boomy is the natural continuation of these schemes, generalizing multivariate polynomial commitments to a batch of evaluation points as presented below in Figure 1.

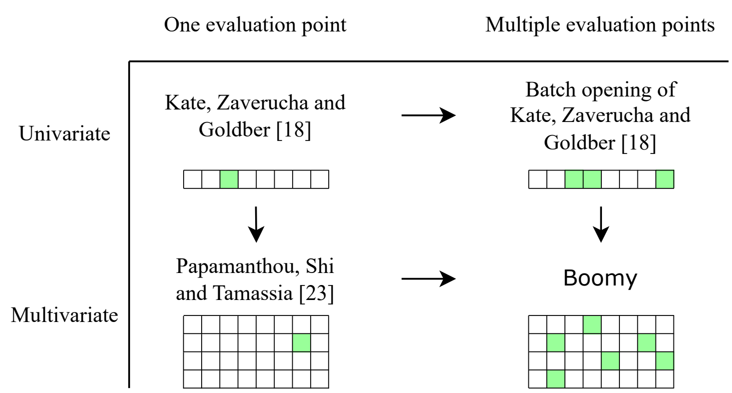

Fig. 1. Boomy's positioning relative to other works.

Since 2013, other works have proposed multivariate polynomial commitments in different security models and with different cryptographic assumptions. We followed the same approach as in [18] and based Boomy on pairing-friendly groups but other works have proposed schemes based on the discrete logarithm problem [6] or the random oracle problem [21]. This enables protocols of multivariate polynomial commitments that either require a transparent setup like [21] and [6] or does not require any setup [28]. This last article is potentially post-quantum since it is only based on hash functions. All of these improvements are made to the detriment of the proof size and/or the opening and verification complexities.

Multivariate polynomial commitments have proven to be of genuine interest to the field of verifiable computation. Using the results of [23], the authors of [10] introduced one of the first universal Succint Non-interactive Arguments of Knowledge (universal SNARK or SNORK) based on multivariate polynomial commitments in parallel with [14] which uses univariate polynomials. Recently, [9] proposed a new version of [28] based on pairing-friendly fields that can only commit multi-linear polynomials (each variable is of degree at most one) but accelerates the work presented in [14].

In this paper we propose a multivariate polynomial commitment supporting batch opening, which is defined in [18] as the opening of several evaluation points at once. We based our technique on [23] by extending and generalizing their scheme using the Gröbner bases theory. This new approach also allows the description and interpretation of their scheme from new perspectives. We study the complexity of this scheme, which we have called Boomy, both in the general case and in the special case where points are all distinct in one dimension. Both complexities are summarized in Table 1 below:

	general case	special case
pk size	$d\mathbb{G}_1$	$d\mathbb{G}_1$
vk size	$(k^n)\mathbb{G}_1,\ (k^n)\mathbb{G}_2$	$k\mathbb{G}_1,\ (k+n)\mathbb{G}_2$
proof size	$\lvert\mathbf{B}\rvert\mathbb{G}_1$	$n\mathbb{G}_1$
commit size	$1\mathbb{G}_1$	$1\mathbb{G}_1$
commit	$(d-1)\mathbb{G}_1^+,\ d\mathbb{G}_1^\times$	$(d-1)\mathbb{G}_1^+,\ d\mathbb{G}_1^\times$
opening	$\mathcal{O}(nk\log k + \lvert\mathbf{B}\rvert d\log d\log\lvert\lvert P\rvert\rvert + k + (n+1)\cdot 2k\cdot \binom{n+2k-1}{2k}^{\omega})\mathbb{F}$, $\mathcal{O}(\lvert\mathbf{B}\rvert\cdot d)\mathbb{G}_1^+,\ \mathcal{O}(\lvert\mathbf{B}\rvert\cdot d)\mathbb{G}_1^\times$	$\mathcal{O}(nk\log k + k + nd\log d\log\lvert\lvert P\rvert\rvert)\mathbb{F}$, $\mathcal{O}(nd)\mathbb{G}_1^+,\ \mathcal{O}(nd)\mathbb{G}_1^\times$
verification	$\mathcal{O}((n+1)k\log k + \lvert\mathbf{B}\rvert d\log d\log\lvert\lvert R\rvert\rvert + (n+1)\cdot 2k\cdot \binom{n+2k-1}{2k}^{\omega})\mathbb{F}$, $\mathcal{O}(k^n)\mathbb{G}_1^+,\ \mathcal{O}(k^n)\mathbb{G}_1^\times,\ \mathcal{O}(\lvert\mathbf{B}\rvert\cdot k^n)\mathbb{G}_2^\times,$ $\mathcal{O}(\lvert\mathbf{B}\rvert\cdot k^n)\mathbb{G}_2^\times,\ (\lvert\mathbf{B}\rvert+1)\mathcal{P}$	$\mathcal{O}((n+1)k\log k)\mathbb{F}$, $k\mathbb{G}_1^+,\ (k+1)\mathbb{G}_1^\times,$ $\mathcal{O}(nk)\mathbb{G}_2^+,\ \mathcal{O}(nk)\mathbb{G}_2^\times,$ $(n+1)\mathcal{P}$

Table 1. Complexity of the Boomy protocol for a multivariate polynomial in $\mathbb{F}[X_1,\ldots,X_n]$ of degree bounded by d_i in each variable X_i. d denotes the maximum number of terms in the polynomial: $d := \prod_{i=1}^{n} d_i$. Lines commit, opening and verification present the complexity of their computations, with $\mathbb{G}_i^+$ and $\mathbb{G}_i^\times$ respectively denoting addition and scalar multiplication in $\mathbb{G}_i$, $\mathbb{F}$ denoting multiplications in $\mathbb{F}$, $\mathcal{P}$ denoting the pairing operation, $\lvert\mathbf{B}\rvert$ the size of the Gröbner basis, $\lvert\lvert P\rvert\rvert$ being the product of the maximum coefficients of the polynomial in each variable and ω the linear algebra constant. These notations are detailed in Section 2.

We note that the complexity of Boomy is better than the complexity consisting of doing k openings with the protocol in [23] and aggregating them using a uniformly random linear combination as in [4]. To our knowledge, Boomy is the first proposal of a protocol allowing the batch opening of a multivariate polynomial based on pairing-friendly groups, paving the way for new applications in the field of verifiable computation. We first introduce preliminaries in Section 2, present the intuition, the construction, the proof of security and the complexity of Boomy in Section 3. We then analyze special cases in Section 4. In Section 5, we explore the potential impact Boomy can have on different applications like verifiable computation, proof of data availability or verifiable information dispersal (VID).

2 Preliminaries

2.1 Notations

Throughout this paper, λ will denote the security parameter, $\mathbb{F}$ a finite field of super-polynomial size $\lambda^{\omega(1)}$ and $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ groups of the same size $\lambda^{\omega(1)}$ that allow the construction of a non-degenerated pairing function $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ computable in polynomial time over λ. $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ will be written additively.

It will be implicitly assumed that before all else, an algorithm is executed to output $(\mathbb{F}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2, G_T)$ given λ as input, such that G_1 and G_2 are uniformly randomly chosen generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively, with $e(G_1, G_2) = G_T$ a generator of $\mathbb{G}_T$.

These generated parameters are also implicitly given as inputs for every algorithm. All adversaries will be supposed probabilistic polynomial time (PPT) algorithms. $negl(\lambda)$ will denote the set of all negligible functions over λ, i.e., all functions lower than $1/p(\lambda)$ for all polynomials p evaluated in λ.

Finally, we will abbreviate vectors with bold letters (e.g., $\mathbf{X} := [X_1, \ldots, X_n]$), the cardinal of a set using the absolute value symbols, elements of the group $\mathbb{G}_i : i \in \{1, 2, T\}$ by using $[\alpha]_i = \alpha \cdot G_i$ and will often write the non-zero integer-set up to n as $[n] := \{1, \ldots, n\}$. We will also use bold notation for ideals $\mathbf{I}$ and algebraic affine varieties $\mathbf{V}$.

2.2 Cryptographic Assumptions

The security of Boomy, like [23], relies on the discrete logarithm (DL), the t-SDH [3] and the t-SBDH [15] assumptions.

Definition 1. DL Assumption. *For any $\tau \in \mathbb{F}^*$. Given the tuple $([1]_1, [\tau]_1) \in \mathbb{G}_1^2$ and for every PPT adversary $\mathcal{A}$, $\Pr[\mathcal{A}([1]_1, [\tau]_1) = \tau)] = negl(\lambda)$.*

Definition 2. t-SDH Assumption. *For any $\tau \in \mathbb{F}^*$. Given the tuple $([1]_1, [\tau]_1, \ldots, [\tau^l]_1) \in \mathbb{G}_1^{l+1}$ and for every PPT adversary $\mathcal{A}$, $\Pr\left[\mathcal{A}([1]_1, [\tau]_1, \ldots, [\tau^l]_1) = \left(c, \left[\frac{1}{\tau+c}\right]_1\right)\right] = negl(\lambda)$ for any $c \in \mathbb{F}\backslash\{-\tau\}$.*

Definition 3. t-SBDH Assumption. *For any $\tau \in \mathbb{F}^*$. Given the tuple $([1]_1, [\tau]_1, \ldots, [\tau^l]_1) \in \mathbb{G}_1^{l+1}$ and for every PPT adversary $\mathcal{A}$, $\Pr\left[\mathcal{A}([1]_1, [\tau]_1, \ldots, [\tau^l]_1) = \left(c, \left[\frac{1}{\tau+c}\right]_T\right)\right] = negl(\lambda)$ for any $c \in \mathbb{F}\backslash\{-\tau\}$.*

2.3 Multivariate Polynomial Commitment

We define a multivariate polynomial commitment as a scheme which follows our extension of the definition given by [18]. More precisely, we expand their definition to several variables and on a batch of evaluation points.

Definition 4. *A multivariate polynomial commitment scheme consists of four algorithms:* Setup*,* Commit*,* Open *and* Verify *such that:*

– $\mathsf{Setup}(\mathbf{d}, k)$: generates a verifier key vk and a prover key pk. The prover key can be used to commit, or to open on a set containing at most k evaluation points, a multivariate polynomial of degree at most d_i in variable X_i. The verifier key is used in the verification of an opening generated with pk.
– $\mathsf{Commit}(P, pk)$: generates a polynomial commitment cm of the polynomial P using the prover key pk.
– $\mathsf{Open}(P, k, (\mathbf{a_i})_{i\in[k]}, pk)$: generates the proof $\boldsymbol{\pi}$ of the evaluation of $P \in \mathbb{F}[X_1, \dots, X_n]$ at the k points $(\mathbf{a_i})_{i\in[k]}$ using pk (with $\mathbf{a_i} = [a_{i,1}, \dots, a_{i,n}]$).
– $\mathsf{Verify}(k, (\mathbf{a_i})_{i\in[k]}, \mathbf{z}, \boldsymbol{\pi}, cm, vk)$: verifies that indeed $\forall i \in [k], z_i = P(\mathbf{a_i})$, i.e., $\mathbf{z}$ are the correct evaluations of the polynomial P at $(\mathbf{a_i})$, P being represented indirectly by its commitment cm. It outputs *accept* if it holds and *reject* otherwise.

Definition 5. *A multivariate polynomial commitment scheme is considered secure if the four properties (correctness, polynomial binding, evaluation binding and hiding) hold:*

– **Correctness***: For all* $P \in \mathbb{F}[\mathbf{X}]$ *of degree at most* d_i *in variable* X_i*, and for all* $(\mathbf{a_i})_{i\in[k]} \in (\mathbb{F}^n)^k$*:*

$$\Pr\left[\begin{array}{r} (pk, sk) \leftarrow \mathsf{Setup}(\mathbf{d}, k) \\ cm \leftarrow \mathsf{Commit}(P, pk) \\ \boldsymbol{\pi} \leftarrow \mathsf{Open}(P, k, (\mathbf{a_i}), pk) \end{array} : \mathsf{Verify}(k, (\mathbf{a_i}), (P(\mathbf{a_i})), \boldsymbol{\pi}, cm, vk) = Accept\right] = 1$$

– **Polynomial Binding***: For all PPT adversary* $\mathcal{A}$*:*

$$\Pr\left[\begin{array}{c} (pk, sk) \leftarrow \mathsf{Setup}(\mathbf{d}, k),\ (cm, P(\mathbf{X}), (\mathbf{a_i}), P'(\mathbf{X}), (\boldsymbol{a'_i}), \boldsymbol{\pi}, \boldsymbol{\pi'}) \leftarrow \mathcal{A}(pk) : \\ \mathsf{Verify}(k, (\mathbf{a_i}), (P(\mathbf{a_i})), \boldsymbol{\pi}, cm, vk) = Accept \wedge \\ \mathsf{Verify}(k, (\boldsymbol{a'_i}), (P'(\boldsymbol{a'_i})), \boldsymbol{\pi'}, cm, vk) = Accept \wedge \\ P(\mathbf{X}) \neq P'(\mathbf{X}) \end{array}\right] = negl(\lambda)$$

– **Evaluation Binding***: For all PPT adversary* $\mathcal{A}$*, it is selectively secure if:*

$$\Pr\left[\begin{array}{c} (\mathbf{a_i}) \leftarrow \mathcal{A}(),\ (pk, sk) \leftarrow \mathsf{Setup}(\mathbf{d}, k),\ (cm, \mathbf{z}, \boldsymbol{\pi}, \boldsymbol{z'}, \boldsymbol{\pi'}) \leftarrow \mathcal{A}(pk) : \\ \mathsf{Verify}(k, (\mathbf{a_i}), \mathbf{z}, \boldsymbol{\pi}, cm, vk) = Accept \wedge \\ \mathsf{Verify}(k, (\mathbf{a_i}), \boldsymbol{z'}, \boldsymbol{\pi'}, cm, vk) = Accept \wedge \\ \exists i \in [k] : z_i \neq z'_i \end{array}\right] = negl(\lambda)$$

else it is adaptively secure if:

$$\Pr\left[\begin{array}{c} (pk, sk) \leftarrow \mathsf{Setup}(\mathbf{d}, k),\ (cm, (\mathbf{a_i}), \mathbf{z}, \boldsymbol{\pi}, \boldsymbol{z'}, \boldsymbol{\pi'}) \leftarrow \mathcal{A}(pk) : \\ \mathsf{Verify}(k, (\mathbf{a_i}), \mathbf{z}, \boldsymbol{\pi}, cm, vk) = Accept \wedge \\ \mathsf{Verify}(k, (\mathbf{a_i}), \boldsymbol{z'}, \boldsymbol{\pi'}, cm, vk) = Accept \wedge \\ \exists i \in [k] : z_i \neq z'_i \end{array}\right] = negl(\lambda)$$

– **Computational Hiding**: *For all polynomial $P \in \mathbb{F}[\mathbf{X}]$, given at most $d := (\prod_i d_i) - 1$ proven evaluation points distributed over one or more batches, a PPT adversary $\mathcal{A}$ cannot determine the value of a new evaluation point with probability more than $negl(\lambda)$.*
In other words, if $(pk, vk) \leftarrow \mathsf{Setup}(\mathbf{d}, k)$, $cm \leftarrow \mathsf{Commit}(P, pk)$ and given k points $(\mathbf{a_i})_{i\in[k]}$ such that $k < d$, even if a PPT adversary $\mathcal{A}$ has access to several batch openings proving those k evaluations, accepted by Verify, they cannot determine $P(\boldsymbol{\alpha})$ with probability more than $negl(\lambda)$ for all $\boldsymbol{\alpha}$ that are not in the given set of evaluation points.

3 Multivariate Polynomial Commitment for Multiple Points

3.1 Intuition

In [23], a prover wants to prove that a chosen polynomial $P \in \mathbb{F}[X_1, \ldots, X_n]$ evaluates to z at point $\mathbf{a} = (a_1, \ldots, a_n)$. To do that, the authors based their protocol on the fact that the polynomial P can be divided by several quotients that nullify at this point $\mathbf{a}$. They proved that

$$\forall i \in [n], \exists Q_i \in \mathbb{F}[\mathbf{X}] : P(\mathbf{X}) = \sum_{i=1}^{n} Q_i(\mathbf{X}) \cdot (X_i - a_i) + r \tag{1}$$

This can be explained by simply observing that the polynomial reduction of a polynomial $P \in \mathbb{F}[X_1, \ldots, X_n]$ by the polynomial $(X_1 - a_1)$ leads to a remainder polynomial in $\mathbb{F}[X_2, \ldots, X_n]$ and thus, that the successive reduction by all the polynomials $(X_i - a_i)$, for $i \in [n]$, leads to a constant remainder which is equal to $P(\mathbf{a})$.

In the univariate case, to define batch openings of points $\mathbf{a} = [a_1, \ldots, a_k]$, [18] divides the polynomial P by $\prod_{i=1}^{k}(X - a_i)$ and shows that the remainder polynomial (of degree lower than k) is the Lagrange polynomial interpolation of the points $[(a_1, P(a_1)), \ldots, (a_k, P(a_k)]$. This polynomial can thus be rebuilt by the verifier from the given evaluation points.

To extend this scheme to the multivariate case, it is thus necessary to define some divisor polynomials built from the points $(\mathbf{a_1}, \ldots, \mathbf{a_k})$ that lead to a remainder polynomial that can be rebuilt by the verifier from the points and their evaluations.

The natural extension of univariate polynomial division to the multivariate case is defined by the Gröbner basis theory [11], if we want to have a unique remainder. The polynomials of the Gröbner basis then play the role of the divisor, the reduction operation corresponds to the polynomial pseudo-division and the corresponding affine algebraic variety is equal to the set composed of the common roots of the basis.

From the Gröbner basis theory, the underlying reason explaining why the construction of [23] works, is that the polynomials $f_i(\mathbf{X}) = X_i - a_i$ define an affine algebraic variety which is exactly $\{\mathbf{a}\}$. Moreover, they also form the reduced Gröbner basis of the Ideal generated by (f_i), since each of them is of degree 1 in a different variable and their leading coefficient is 1. The remainder of the reduction of P by (f_i) is then a constant polynomial always equal to $P(\mathbf{a})$.

To generalize their protocol over a batch of points $(\mathbf{a_i})_{i\in[k]}$, we reused the above reason to construct a set of polynomials that defines an affine algebraic variety that equals $\{\mathbf{a_i} : i \in [k]\}$. To make the remainder unique, we computed a Gröbner basis $\mathbf{B}$ of the ideal generated by this set of polynomials. This generalizes Eq. 1 as

$$P(\mathbf{X}) = \sum_{i=1}^{n} Q_i(\mathbf{X}) \cdot B_i(\mathbf{X}) + R(\mathbf{X})$$

The verifier has to recover $R(\mathbf{X})$, which is not trivial with more than one point. To do that, we propose computing one possible interpolation of the batch of points. Since in $\mathbb{F}[\mathbf{X}] \big/ \langle \mathbf{B} \rangle$ the polynomial P is equal to R or to any other interpolation of the points $(\mathbf{a_i}, P(\mathbf{a_i}))_{i\in[k]}$, the verifier only has to reduce their interpolated polynomial by the ideal $\langle \mathbf{B} \rangle$ to recover R.

3.2 Polynomials Defining an affine algebraic variety

In this section, we present a method enabling the construction of a set of polynomials defining an affine algebraic variety $\mathbf{V}$ given the set of points of $\mathbf{V}$. Suppose that we want to construct the polynomials defining $\mathbf{V} := \{\mathbf{a_i} \in \mathbb{F}^n : i \in [k]\}$. First, we calculate the n polynomials that define the n algebraic affine varieties in each dimension independently: $\forall i \in [n], S_i := \{a_{j,i}; \forall j \in [k]\}$ and $f_i(X_i) := \prod_{j \in S_i}(X_i - j)$. Those polynomials together define the affine algebraic variety of the "grid" (the Cartesian product) formed by each tuple that has each coordinate in common with any $\mathbf{a_i}$. We interpolate a polynomial that goes through all the points of the "grid" with value 0 if they are in $\{\mathbf{a_i} : i \in [k]\}$ and 1 otherwise. This polynomial can be interpolated one variable at a time using Lagrange polynomial interpolations.

This construction directly leads to the following property:

Theorem 1. *For any $n \in \mathbb{N}$, we can always construct a set containing at most $n+1$ polynomials in $\mathbb{F}[X_1, \ldots, X_n]$ that defines an affine algebraic variety that equals a finite set of k given points $(\mathbf{a_i})_{i\in[k]} \in (\mathbb{F}^n)^k$.*

Example 1. To illustrate the construction, let us consider the the algebraic variety equal to $\{(2,5),(2,10),(7,2),(7,5)\}$ and the polynomial ring $\mathbb{F}[X_1, X_2]$ with $\mathbb{F} := \mathbb{Z} \big/ 97\mathbb{Z}$. We show how to construct the set of three polynomials $\{f_1, f_2, f_3\}$ defining this variety.

First we compute f_1 and f_2 as $f_1(X_1, X_2) := (X_1 - 2) \cdot (X_1 - 7)$ and $f_2(X_1, X_2) := (X_2 - 2) \cdot (X_2 - 5) \cdot (X_2 - 10)$. There, the variety defined by these two polynomials is the Cartesian product of $\{1, 2\}$ and $\{2, 5, 10\}$, *i.e.* $\mathbf{V}(f_1, f_2) = \{(2,2),(2,5),(2,10),(7,2),(7,5),(7,10)\}$. Now, we want to remove the points $(2,2)$ and $(7,10)$. To do that, we build the interpolating polynomial that equals 1 at $(2,2)$ and $(7,10)$, and 0 at the other points, one variable at a time. For that, we first compute the Lagrange polynomials h_1 and h_2 in X_2 that interpolate $\{(2,1),(5,0),(10,0)\}$ and $\{(2,0)(5,0),(10,1)\}$ respectively. We obtain $h_1(X_2) = 93X_2^2+60X_2+91$ and $h_2(X_2) = 17X_2^2+75X_2+73$. Then we can define $f_3(X_1,X_2)$ as the interpolation in X_1 of the points $\{(2,h_1(X_2)),(7,h_2(X_2)\}$ and we obtain $f_3(X_1,X_2) = -6X_1X_2^2 - 32X_1X_2 + 19X_1 - 10X_1^2 + 6X_2 + 26$.

3.3 Main protocol

For a multivariate polynomial $P(\mathbf{X}) \in \mathbb{F}[X_1,\dots,X_n]$ of degree at most d_i in variable X_i, a prover wants to convince a verifier, except with probability less than $negl(\lambda)$, that for a chosen set of k points $(\mathbf{a_i}) \in (\mathbb{F}^n)^k$,

$$P(\mathbf{a_i}) = z_i \text{ for } i \in [k]$$

Below, we present the Boomy protocol composed of the four algorithms:

- $\mathsf{Setup}(\mathbf{d},k)$: uniformly randomly chose $\tau \in \mathbb{F}$. Output $pk := \{[\prod_{i=1}^n \tau_i^{\alpha_i}]_1 : \alpha_i \in [d_i]\}$ and $vk = \{[\prod_{i=1}^n \tau_i^{\alpha_i}]_1 : \alpha_i \in [k]\} \cup \{[\prod_{i=1}^n \tau_i^{\alpha_i}]_2 : \alpha_i \in [k]\}$.
- $\mathsf{Commit}(P,pk)$: return $cm = [P(\boldsymbol{\tau})]_1$.
- $\mathsf{Open}(P,k,(\mathbf{a_i})_{i\in[k]},pk)$:
  1. Compute the polynomials $\mathbf{f}$ defining the affine algebraic variety composed of $(\mathbf{a_i})_{i\in[k]}$.
  2. For a predefined monomial order, compute a Gröbner basis $\mathbf{B}$ of the ideal generated by $\mathbf{f}$.
  3. Reduce the polynomial P with each B_i to recover the "quotients" Q_i and the "remainder" R such that
  $$P(\mathbf{X}) = \sum_{i=1}^{|\mathbf{B}|}(B_i(\mathbf{X}) \cdot Q_i(\mathbf{X})) + R(\mathbf{X})$$
  4. Compute and return the proof $\boldsymbol{\pi} := ([Q_i(\boldsymbol{\tau})]_1)_{i\in|\mathbf{B}|}$ composed of each "quotient" evaluated in $\boldsymbol{\tau}$.
- $\mathsf{Verify}(k,(\mathbf{a_i})_{i\in[k]},\mathbf{z},\boldsymbol{\pi},cm,vk)$:
  1. Compute the same polynomials $\mathbf{f}$ defining the affine algebraic variety that is equal to the set composed of $(\mathbf{a_i})_{i\in[k]}$.
  2. Compute the same Gröbner basis $\mathbf{B}$ of the ideal generated by $\mathbf{f}$.
  3. Compute by interpolation a polynomial $R'(\mathbf{X})$ that equals P on the points $(\mathbf{a_i})_{i\in[k]}$. Recover $R(\mathbf{X})$ by reducing $R'(\mathbf{X})$ with $\mathbf{B}$. Evaluate R and each B_i in $\boldsymbol{\tau}$ to obtain $[R(\boldsymbol{\tau})]_1$ and $[B_i(\boldsymbol{\tau})]_2$.
  4. Accept if:
  $$e(cm - [R(\boldsymbol{\tau})]_1, [1]_2) = \sum_{i=1}^{|\mathbf{B}|} e(\pi_i, [B_i(\boldsymbol{\tau})]_2)$$
  otherwise reject.

Theorem 2. *The Boomy protocol is a selectively secure multivariate polynomial commitment scheme (as defined in Section 2.3) under the assumptions presented in Section 2.2.*

A proof of Theorem 2 is provided in the next Section 3.4.

3.4 Security Analysis

In this section, we prove that Boomy's construction is secure under the selective security model and explain why its security can also be proven in the algebraic group model of [13]. To prove the polynomial binding, the evaluation binding and the hiding properties, we will build a simulator $\mathcal{S}$ that can break a given a t-SBDH, t-SDH or DL problem in PPT with probability more than $negl(\lambda)$ if it has access to an adversary that can break one of these properties in PPT with probability more than $negl(\lambda)$.

Correctness: The correctness directly follows from the reduction of P by a Gröbner basis $\mathbf{B}$ giving quotients Q_i and remainder R.

$$\begin{aligned}
Accept \leftarrow \mathsf{Verify}() &\iff e(cm - [R(\boldsymbol{\tau})]_1, [1]_2) = \sum_{i=1}^{|\mathbf{B}|} e(\pi_i, [B_i(\boldsymbol{\tau})]_2) \\
&\iff e([P(\boldsymbol{\tau})]_1 - [R(\boldsymbol{\tau})]_1, [1]_2) = \sum_{i=1}^{|\mathbf{B}|} e([Q_i(\boldsymbol{\tau})]_1, [B_i(\boldsymbol{\tau})]_2) \\
&\iff [P(\boldsymbol{\tau}) - R(\boldsymbol{\tau})]_T = \sum_{i=1}^{|\mathbf{B}|} [B_i(\boldsymbol{\tau}) \cdot Q_i(\boldsymbol{\tau})]_T \\
&\iff P(\boldsymbol{\tau}) = \left(\sum_{i=1}^{|\mathbf{B}|} B_i(\boldsymbol{\tau}) \cdot Q_i(\boldsymbol{\tau}) \right) + R(\boldsymbol{\tau})
\end{aligned}$$

Polynomial Binding: The simulator $\mathcal{S}$ first crafts the trusted setup of Boomy using the elements $([1]_1, [t]_1, \dots, [t^l]_1)$ of the t-SBDH problem. It can do this by uniformly randomly picking r_i and s_i in $\mathbb{F}$ and fixing $\tau_i := r_i \cdot t + s_i$ for $i \in \{2, \dots, n\}$ and $\tau_1 = t$. Since each τ_i is a polynomial in t, $\mathcal{S}$ can craft vk and pk without knowing t. Suppose that a PPT adversary $\mathcal{A}$ can craft $P(\mathbf{X})$ and $Q(\mathbf{X})$ such that $P \neq Q$ and $[P(\boldsymbol{\tau})]_1 = [Q(\boldsymbol{\tau})]_1$, then we have:

$$[P(\boldsymbol{\tau})]_1 - [Q(\boldsymbol{\tau})]_1 = [P(\boldsymbol{\tau}) - Q(\boldsymbol{\tau})]_1 = [(P - Q)(\boldsymbol{\tau})]_1 = [0]_1$$

It follows that $\boldsymbol{\tau}$ is a non-trivial root of the polynomial $P - Q$. The simulator recovers the polynomials P and Q from $\mathcal{A}$ and computes $P - Q$. It crafts the polynomial $Z(X) := (P - Q)(\mathbf{X})$ by replacing each variable X_i with $r_i \cdot X + s_i$. Then we have $Z(t) = P - Q(\boldsymbol{\tau}) = 0$. $\mathcal{S}$ can recover t in PPT with the same probability as $\mathcal{A}$ by factorizing Z [27], breaking the t-SDH assumption.

Evaluation Binding: The simulator $\mathcal{S}$ first asks the adversary $\mathcal{A}$ to commit on the challenge point $(\mathbf{a_i})_{i \in [k]}$ at which it will forge a valid batch opening containing at least one incorrect evaluation. Suppose that the incorrect evaluation happens at least at $\boldsymbol{a_\eta}$ with the false value z'_η for the rest of the proof. $\mathcal{S}$ then crafts the trusted setup of Boomy using the elements $([1]_1, [t]_1, \dots, [t^l]_1)$ of the t-SBDH problem. It can do this by uniformly randomly picking r_i and s_i in $\mathbb{F}$ that verify $a_{\eta,i} = r_i \cdot a_{\eta,1} + s_i$ and fixing $\tau_i := r_i \cdot t + s_i$ for $i \in [n]\backslash\{\eta\}$ and $\tau_\eta = t$. Since each τ_i is a polynomial in t, $\mathcal{S}$ can craft vk and pk without knowing t. $\mathcal{S}$ then calls $\mathcal{A}$ to recover $cm, \boldsymbol{\pi}, \boldsymbol{\pi'}, \mathbf{z}, \boldsymbol{z'}, (\mathbf{a_i})_{i\in[k]}$ in PPT with probability more than $negl(\lambda)$ such that both $\mathsf{Verify}(k, (\mathbf{a_i})_{i\in[k]}, \mathbf{z}, \boldsymbol{\pi}, cm)$ and $\mathsf{Verify}(k, (\mathbf{a_i})_{i\in[k]}, \boldsymbol{z'}, \boldsymbol{\pi'}, cm)$ output *Accept* and $\mathbf{z} \neq \boldsymbol{z'}$. We will use $\mathbf{B}$ to denote the elements of the Gröbner basis used during the Boomy protocol, $R(\mathbf{X})$ and $R'(\mathbf{X})$ to denote the remainders of the reduction of the interpolated polynomial of $(\mathbf{a_i}, z_i)$ and $(\mathbf{a_i}, z'_i)$ by $\mathbf{B}$. Since the verification holds, we have

$$e(cm - [R(\boldsymbol{\tau})]_1, [1]_2) = \sum_{i=1}^{|\mathbf{B}|} e(\pi_i, [B_i(\boldsymbol{\tau})]_2)$$

$$e(cm - [R'(\boldsymbol{\tau})]_1, [1]_2) = \sum_{i=1}^{|\mathbf{B}|} e(\pi'_i, [B_i(\boldsymbol{\tau})]_2)$$

$\mathcal{S}$ defines $\delta(\mathbf{X}) := R'(\mathbf{X}) - R(\mathbf{X}) \neq 0$ (because $R'(\boldsymbol{a_\eta}) - R(\boldsymbol{a_\eta}) = z'_\eta - z_\eta \neq 0$). It follows that

$$\begin{aligned} e(cm - [R(\boldsymbol{\tau})]_1, [1]_2) - e(cm - [R'(\boldsymbol{\tau})]_1, [1]_2) &= \sum_{i=1}^{|\mathbf{B}|} (e(\pi_i, [B_i(\boldsymbol{\tau})]_2) - e(\pi'_i, [B_i(\boldsymbol{\tau}]_2)) \\ \iff e([R'(\boldsymbol{\tau})]_1 - [R(\boldsymbol{\tau})]_1, [1]_2) &= \sum_{i=1}^{|\mathbf{B}|} e(\pi_i - \pi'_i, [B_i(\boldsymbol{\tau})]_2) \\ \iff e([\delta(\boldsymbol{\tau})]_1, [1]_2) &= \sum_{i=1}^{|\mathbf{B}|} e(\pi_i - \pi'_i, [B_i(\boldsymbol{\tau})]_2) \end{aligned} \tag{2}$$

But since $B_i \in \mathbf{I}(\mathbf{V}(\{\mathbf{a_i}, i \in [k]\})) \subset \mathbf{I}(\mathbf{V}(\{\boldsymbol{a_\eta}\})), \forall i \in [|\mathbf{B}|], \forall j \in [n], \exists Q_{i,j}(\mathbf{X}) \in F[X_1, \dots, X_n]$ such that $B_i(\mathbf{X}) := \sum_{j=1}^{n}(X_j - a_{\eta,j}) \cdot Q_{i,j}(\mathbf{X})$, $\mathcal{S}$ can do the same with $\delta(\mathbf{X})$ but $\delta \notin \mathbf{I}(\mathbf{V}(\{\boldsymbol{a_\eta}\}))$ so that δ will have a remainder different from the zero polynomial. Since the quotients are all degree 1 in each variable, the remainder will be in $\mathbb{F}$. So $\forall j \in [n], \exists D_j(\mathbf{X}) \in F[\mathbf{X}]$ such that $\delta(\mathbf{X}) := \sum_{j=1}^{n}(X_j - a_{\eta,j}) \cdot D_j(\mathbf{X}) + d$. With $d \in \mathbb{F}\backslash\{0\}$. Eq. 2 can be rewritten when replacing π and π' by $[p]_1$ and $[p']_1$ respectively as

$$
\begin{aligned}
\left[\sum_{j=1}^{n}(\tau_j - a_{\eta,j}) \cdot D_j(\boldsymbol{\tau}) + d\right]_T &= \left[\sum_{i=1}^{|\mathbf{B}|}(p_i - p'_i) \cdot \sum_{j=1}^{n}(\tau_j - a_{\eta,j}) \cdot Q_{i,j}(\boldsymbol{\tau})\right]_T \\
\iff [d]_T &= \left[\sum_{j=1}^{n}(\tau_j - a_{\eta,j}) \cdot \left(-D_j(\boldsymbol{\tau}) + \sum_{i=1}^{|\mathbf{B}|}(p_i - p'_i) \cdot Q_{i,j}(\boldsymbol{\tau})\right)\right]_T \\
\iff [d]_T &= \left[(t - a_{\eta,1}) \cdot \sum_{j=1}^{n} -r_j \cdot D_j(\boldsymbol{\tau}) + r_j \cdot \sum_{i=1}^{|\mathbf{B}|}(p_i - p'_i) \cdot Q_{i,j}(\boldsymbol{\tau})\right]_T \\
\iff \left[\frac{1}{t - a_{\eta,1}}\right]_T &= \left[d^{-1} \cdot \sum_{j=1}^{n} r_j \cdot \left(-D_j(\boldsymbol{\tau}) + \sum_{i=1}^{|\mathbf{B}|}(p_i - p'_i) \cdot Q_{i,j}(\boldsymbol{\tau})\right)\right]_T \\
\iff \left[\frac{1}{t - a_{\eta,1}}\right]_T &= d^{-1} \cdot \sum_{j=1}^{n} r_j \cdot \left(e([-D_j(\boldsymbol{\tau})]_1, [1]_2) + \sum_{i=1}^{|\mathbf{B}|} e(\pi_i - \pi'_i, [Q_{i,j}(\boldsymbol{\tau})]_2)\right)
\end{aligned}
\tag{3}
$$

Therefor, $\mathcal{S}$ can break the t-SBDH problem returning $\left(-a_{\eta,1}, \left[\frac{1}{t-a_{\eta,1}}\right]_T\right)$ in PPT using Eq. 3 with the same probability as $\mathcal{A}$.

Computational Hiding: The simulator $\mathcal{S}$ can break the DL problem $([1]_1, [\alpha]_1)$ if it has access to an adversary $\mathcal{A}$ able to break the computational hiding property. To do that, $\mathcal{S}$ first crafts the trusted setup of the Boomy protocol using uniformly randomly picked $\tau_i \in \mathbb{F}$ for $i \in [n]$ to obtain vk and pk. $\mathcal{S}$ uniformly randomly takes k evaluations $(\mathbf{a_i}, z_i)_{i\in[k]}$ of a polynomial P such that $\forall j > 1, a_{i,j} = 0$. $\mathcal{S}$ supposes that $P(\mathbf{0}) = \alpha$, which is the solution of the discrete logarithm problem. The simulator can find a polynomial that verifies $P(\mathbf{a_i}) = z_i$ using Lagrange polynomial interpolation in X_1 and then multiplies it by $(X_1 - \alpha)$ to obtain P. P is then a polynomial of one variable and the same argument as in [18] can be given: $\mathcal{S}$ computes the commit $P(\boldsymbol{\tau})$ using the DL problem and the trusted setup, it also computes the proofs and gives them to the adversary $\mathcal{A}$ who outputs the polynomial P in PPT with probability more than $negl(\lambda)$. $\mathcal{S}$ can recover α by evaluating P at $\mathbf{0}$ and therefore break the DL problem with probability more than $negl(\lambda)$ in PPT.

Note on the Algebraic Group Model Note that these proofs can easily be done in the Algebraic Group Model (AGM) presented in [13] through the proof of the knowledge soundness property and the "real pairing check" described in [4]. The correctness, the polynomial binding and the hiding properties have the same proof in this model. The evaluation binding holds if the knowledge soundness also holds. Indeed, since Verify outputted *Accept* for the evaluation of a polynomial

P at $(\mathbf{a_i})$ to false values $\boldsymbol{z'}$ represented by the polynomial $R'(\mathbf{X})$ (the remainder of the reduction), the adversary in the AGM can produce polynomials $Q_i(\mathbf{X})$ such that

$$e([P(\boldsymbol{\tau})]_1 - [R'(\boldsymbol{\tau})]_1, [1]_2) = \sum_{i=1}^{|\mathbf{B}|} e([Q_i(\boldsymbol{\tau})]_1, [B_i(\boldsymbol{\tau})]_2)$$

Then the probability of success of this "real pairing" verification is bounded by the probability of success of the ideal check: $P(\mathbf{X}) - R'(\mathbf{X}) \equiv \sum_{i=1}^{|\mathbf{B}|} Q_i(\mathbf{X}) \cdot B_i(\mathbf{X})$. But since $\boldsymbol{z'}$ is not the real value of P on $(\mathbf{a_i})$, we have $P(\mathbf{a_i}) - z'_i \neq 0$ for at least one $i \in [k]$. However, according to Hilbert's weak Nullstellensatz [11], this means that $P(\mathbf{X}) - R'(\mathbf{X}) \notin \langle \mathbf{G} \rangle$ which implies that $\nexists\, Q_i \in \mathbb{F}[\mathbf{X}]$, such that $P(\mathbf{X}) - R'(\mathbf{X}) = \sum_{i=1}^{|\mathbf{B}|} Q_i(\mathbf{X}) \cdot B_i(\mathbf{X})$, which contradicts the ideal check. This proves that the knowledge soundness holds and then that the Boomy protocol is secure in the AGM.

3.5 Complexity Analysis

In this section, we analyze the complexity of Boomy. In the case where $n = 1$ we get the same complexity as [18] and in the case where $n \geq 1$ and $k = 1$ we obtain the same complexity as [23].

In the following, we expose the complexity evaluations of Boomy for a polynomial P of bounded degree d_i in each variable X_i. $d := \prod_{i \in [n]} d_i$ is then the maximum number of terms in P. k denotes the maximum number of points supported during the Open algorithm.

The prover key pk is composed of elements enabling the commitment and proof of polynomials of maximum degrees d_i in variable X_i. Hence, its size is d elements of $\mathbb{G}_1$. The verifier key, however, enables the verification of proofs produced with pk. It results that vk has to support the computation of the evaluation of the Gröbner basis and of the polynomial that is interpolating the remainder. Since all those polynomials are at most of degree k in each variable, vk is composed of k^n elements of $\mathbb{G}_1$ for the evaluation of the remainder and k^n elements of $\mathbb{G}_2$ for the evaluation of the polynomials of the Gröbner basis.

The proof is composed of the evaluations of the Gröbner basis $\mathbf{B}$ and then is of size $|\mathbf{B}|$ elements of $\mathbb{G}_1$ while the commitment is the evaluation of P only representing one element of $\mathbb{G}_1$.

The computation complexity of the commitment is reduced to the evaluation of P in $\mathbb{G}_1$. Since P is composed of at most d terms, it is necessary to do d scalar multiplications in $\mathbb{G}_1$ and $d - 1$ additions in $\mathbb{G}_1$.
The opening is composed of several steps:

- The computation of the polynomials defining the affine algebraic variety. It can be done in $\mathcal{O}(nk \log k)$ multiplications in $\mathbb{F}$ using the construction of Section 3.2.

– The computation of a Gröbner basis related to the ideal generated by the previous polynomials. It can be done in $\mathcal{O}((n+1) \cdot 2k \cdot \binom{n+2k-1}{2k}^{\omega})$ multiplications in $\mathbb{F}$ with $2 \leq \omega \leq 3$ the linear algebra constant of matrix inversion [2].
– The reduction of P by the Gröbner basis. It can be bounded by $\mathcal{O}(|\mathbf{B}| \cdot d \log d \log ||P|| + k)$ multiplications in $\mathbb{F}$ with $||P||$ being the product of the maximum coefficients of P in each variable [26].
– The evaluation of the quotients using pk. Since the quotients are of degree bounded by the degree of P, they need less than $|\mathbf{B}| \cdot d$ scalar multiplications in $\mathbb{G}_1$ and less than $|\mathbf{B}|(d-1)$ additions in $\mathbb{G}_1$.

The verification is composed of the following steps:

– The computation of the same polynomials defining the affine algebraic variety. It can be done in $\mathcal{O}(nk \log k)$ multiplications in $\mathbb{F}$.
– The computation of the same Gröbner basis related to the ideal generated by the previous polynomials. It can be done in $\mathcal{O}((n+1) \cdot 2k \cdot \binom{n+2k-1}{2k}^{\omega})$ multiplications in $\mathbb{F}$.
– One interpolation of the remainder using the evaluation points. It can be done in $\mathcal{O}(nk \log k)$ multiplications in $\mathbb{F}$.
– The reduction of the computed remainder by the Gröbner basis. It can be bounded by $\mathcal{O}(|\mathbf{B}| \cdot d \log d \log ||R||)$ multiplications in $\mathbb{F}$ with $||R||$ being the product of the maximum coefficients of the remainder in each variable.
– The evaluation of the remainder using vk. It can be done in less than k^n scalar multiplications in $\mathbb{G}_1$ and less than $k^n - 1$ additions in $\mathbb{G}_1$ because the remainder is of degree bound by k in each variable.
– The evaluation of the polynomials of the Gröbner basis using vk. It can be done in less than $|\mathbf{B}| \cdot k^n$ scalar multiplications in $\mathbb{G}_2$ and less than $|\mathbf{B}| \cdot (k^n - 1)$ additions in $\mathbb{G}_2$.
– The $|\mathbf{B}|+1$ evaluation of the pairing function.

The complexity of Boomy is summarized in Table 2 below where, in the lines commit computation, opening computation and verification computation, $\mathbb{G}_i^+$ and $\mathbb{G}_i^\times$ denote addition and scalar multiplication (in additive notation) in $\mathbb{G}_i$, $\mathbb{F}$ denotes multiplications in $\mathbb{F}$. We will denote the pairing operation with $\mathcal{P}$.

Note that trusted setups of univariate polynomial commitments can be computed using a multi-party protocol, reinforcing its security by distributing the knowledge of the secret among multiple entities. An adversary would have to corrupt every participant to recover the secret τ of the trusted setup [19]. Moreover, univariate polynomial commitments can be adapted and reused without a loss of security for the Boomy protocol, as proposed for [23] by the authors of [30]. This directly enables the use of trusted setups generated from multi-party protocols without the need to redo the heavy computation associated.

	Complexities
pk size	$d\mathbb{G}_1$
vk size	$(k^n)\mathbb{G}_1,\ (k^n)\mathbb{G}_2$
proof size	$\lvert\mathbf{B}\rvert\mathbb{G}_1$
commit size	$1\mathbb{G}_1$
commit computation	$(d-1)\mathbb{G}_1^+,\ d\mathbb{G}_1^\times$
opening computation	$\mathcal{O}(nk\log k + \lvert\mathbf{B}\rvert d\log d\log\lvert\lvert P\rvert\rvert +$ $k + (n+1)\cdot 2k\cdot \binom{n+2k-1}{2k}^\omega)\mathbb{F},$ $\mathcal{O}(\lvert\mathbf{B}\rvert\cdot d)\mathbb{G}_1^+,\ \mathcal{O}(\lvert\mathbf{B}\rvert\cdot d)\mathbb{G}_1^\times$
verification computation	$\mathcal{O}((n+1)k\log k + \lvert\mathbf{B}\rvert d\log d\log\lvert\lvert R\rvert\rvert +$ $(n+1)\cdot 2k\cdot \binom{n+2k-1}{2k}^\omega)\mathbb{F},\ \mathcal{O}(k^n)\mathbb{G}_1^+,$ $\mathcal{O}(k^n)\mathbb{G}_1^\times,\ \mathcal{O}(\lvert\mathbf{B}\rvert\cdot k^n)\mathbb{G}_2^+,\ \mathcal{O}(\lvert\mathbf{B}\rvert\cdot k^n)\mathbb{G}_2^\times,\ (\lvert\mathbf{B}\rvert+1)\mathcal{P}$

Table 2. Complexity of Boomy in the general case

4 Special Cases

4.1 Cartesian Product

In the construction of the polynomials that define the affine algebraic variety presented in Section 3.2, n univariate polynomials are built first. The variety corresponding to these first n polynomials is the Cartesian product of the roots of the univariate polynomial. The last interpolation polynomial added in the construction is used to "filter out" some of the points of the Cartesian product (see the Example in Section 3.2). Therefore, if the variety is by construction a Cartesian product, this last polynomial is useless (since it evaluates to the zero polynomial adding no element to the ideal).

The first n polynomials directly form the reduced Gröbner basis since each of them is a univariate polynomial in different variables and their leading coefficient is one. It follows that the proof contains exactly n elements and the complexity of the opening computation is then reduced compared to the general case.

4.2 Points Distinct in One Dimension

In the case where, for a given dimension j, all of the j-coordinates of the evaluation points are different, the proof can be reduced to n elements. The computation of a Gröbner basis can also be simplified because it can be obtained by only computing Lagrange polynomial interpolations in one dimension. This case may occur in many applications enabling drastic reductions in verifier and prover computations.

Claim. For any dimension n, for any set of points $\{\mathbf{a_i} \in \mathbb{F}^n : i \in [k]\}$ such that $\exists m \in [n] : \forall (i,j)$ with $i \neq j$ we have $a_{i,m} \neq a_{j,m}$, we can build the reduced Gröbner basis of the ideal of the variety $\{\mathbf{a_i} \in \mathbb{F}^n : i \in [k]\}$ simply by using Lagrange polynomial interpolations.

Proof. Let $n \in \mathbb{N}$, fix $\mathbf{V} := \{\mathbf{a_i} \in \mathbb{F}^n : i \in [k]\}$ such that $\exists m \in [n] : \forall (i,j), i \neq j \Rightarrow a_{i,m} \neq a_{j,m}$. Let $\mathbf{I}(\mathbf{V})$ denote the ideal of the polynomials vanishing at the points of $\mathbf{V}$. As [11] did with two variables, using Lagrange polynomial interpolations, we will compute the $n-1$ intermediate polynomials $h_i(X_m)$ for each $i \neq m$:

$$h_i(X_m) := \sum_{j=1}^{k} a_{j,i} \prod_{l \neq j} \frac{X_m - a_{l,m}}{a_{j,m} - a_{l,m}}$$

Let $f(X_m) := \prod_{i=1}^{k}(X_m - a_{i,m})$.
First, let us show that:

$$\begin{aligned}\mathbf{I}(\mathbf{V}) = &\langle X_0 - h_0(X_m), X_1 - h_1(X_m), \ldots, X_{m-1} - h_{m-1}(X_m), \\ &f(X_m), X_{m+1} - h_{m+1}(X_m), \ldots, X_n - h_n(X_m)\rangle\end{aligned}$$

1. It is straightforward to prove the inclusion $\langle X_0 - h_0(X_m), X_1 - h_1(X_m), \ldots, X_{m-1} - h_{m-1}(X_m), f(X_m), X_{m+1} - h_{m+1}(X_m), \ldots, X_n - h_n(X_m)\rangle \subseteq \mathbf{I}(\mathbf{V})$. Indeed, $f(X_m)$ vanishes by construction at $\{a_{i,m} : i \in [k]\}$. It is also the case for any $X_i - h_i(X_m)$ at $\{a_{j,i}; j \in [k]\}$ because $\forall j \in [k], h_i(a_{j,m}) = a_{j,i}$.
2. For the other inclusion, we will suppose that the polynomial $P(\mathbf{X})$ vanishes at $\{\mathbf{a_i} : i \in [k]\}$. Giving the monomial order $X_i > X_{i+1}$ and $X_m < X_i$ (so we have $X_0 > X_1 > \cdots > X_{m-1} > X_{m+1} > \cdots > X_n > X_m$), the leading term of f is X_m^k and the leading term of $X_i - h_i(X_m)$ is X_i. It follows that the reduction of P by f and $(X_i - h_i(X_m))_{i \in [n] \setminus \{m\}}$ has a polynomial composed of the monomials $1, \ldots, X_m^{k-1}$ as remainder. Thus, the remainder R is a univariate polynomial of degree at most $k-1$ or the zero polynomial. Since $\exists (Q_1, \ldots, Q_n) \in \mathbb{F}[\mathbf{X}]$ such that

$$P(\mathbf{X}) = Q_m(\mathbf{X}) \cdot f(X_m) + \sum_{i=1, i \neq m}^{n} Q_i(\mathbf{X}) \cdot (X_i - h_i(X_m)) + R(\mathbf{X})$$

it follows that

$$R(\mathbf{X}) = P(\mathbf{X}) - Q_m(\mathbf{X}) \cdot f(X_m) - \left(\sum_{i=1, i \neq m}^{n} Q_i(\mathbf{X}) \cdot (X_i - h_i(X_m)) \right)$$

Since P, f and $X_i - h_i(X_m)$ vanish at $\{\mathbf{a_i} : i \in [k]\}$, R also vanishes at those different k points. Since R is a univariate polynomial of degree at most $k-1$ with k distinct roots, R is the zero polynomial, proving that P is in $\langle X_0 - h_0(X_m), X_1 - h_1(X_m), \ldots, X_{m-1} - h_{m-1}(X_m), f(X_m), X_{m+1} - h_{m+1}(X_m), \ldots, X_n - h_n(X_m)\rangle$. Since P can be any element of $\mathbf{I}(\mathbf{V})$, we have the second inclusion.

Secondly, let us prove that $\mathbf{B} := \{X_i - h_i(X_m) : i \in [k]\backslash\{m\}\} \cup \{f(X_m)\}$ is the reduced Gröbner basis of $\mathbf{I}(\mathbf{V})$.

1. We have previously shown that for any $P \in \mathbf{I}(\mathbf{V})$, its remainder of the reduction by $\mathbf{B}$ is the zero polynomial. It follows that the leading term of P is divisible by the leading term of elements of $\mathbf{B}$ which proves that $\mathbf{B}$ is a Gröbner basis of $\mathbf{I}(\mathbf{V})$.
2. By construction, the leading coefficient of any element of $\mathbf{B}$ is 1. It is clear that with the order $X_0 > X_1 > \cdots > X_{m-1} > X_{m+1} > \cdots > X_n > X_m$, the leading term of $X_i - h_i(X_m)$ which is X_i does not divide any term of $f(X_m)$ or $X_j - h_j(X_m)$ with $i \neq j$. And it is also clear that the leading term of $f(X_m)$ which is X_m^k does not divide any term of $X_i - h_i(X_m)$ (because $\forall i \in [k]\backslash\{m\}, X_i > X_m$ so the leading term is X_i). It follows that $\mathbf{B}$ is the reduced Gröbner basis of $\mathbf{I}(\mathbf{V})$.

Corollary 1. *Using the construction given in Section 3.2, fix* $\mathbf{B}$ *the reduced Gröbner basis. The remainder of any* $P \in \mathbb{F}[X_1, \ldots, X_n]$ *by the reduction of* $\mathbf{B}$ *is a polynomial in one variable.*

Proof. This proposition is trivial since in the construction of Section 3.2, the leading terms of the polynomials of the Gröbner basis $\mathbf{B}$ are X_i if $i \neq m$ or X_m^k, therefore the remainder is composed of the monomials $1, X_m, \ldots, X_m^{k-1}$.

It follows that the remainder $R(\mathbf{X})$ can be directly computed once again using a Lagrange polynomial interpolation in variable X_m. In this case, the interpolated polynomial is already the remainder of its reduction by $\mathbf{B}$ making this step of computation unnecessary and the verifier key vk reducible to only k elements. The complexity is summarized in Table 3 below.

	Complexities
pk size	$d\mathbb{G}_1$
vk size	$k\mathbb{G}_1,\ (k+n)\mathbb{G}_2$
proof size	$n\mathbb{G}_1$
commit size	$1\mathbb{G}_1$
commit computation	$(d-1)\mathbb{G}_1^+,\ d\mathbb{G}_1^\times$
opening computation	$\mathcal{O}(nk\log k + nd\log d\log\|\|P\|\|{+}k)\mathbb{F}$, $\mathcal{O}(nd)\mathbb{G}_1^+,\ \mathcal{O}(nd)\mathbb{G}_1^\times$
verification computation	$\mathcal{O}((n+1)k\log k)\mathbb{F}$, $k\mathbb{G}_1^+,\ (k+1)\mathbb{G}_1^\times,\ \mathcal{O}(nk)\mathbb{G}_2^+$, $\mathcal{O}(nk)\mathbb{G}_2^\times,\ (n+1)\mathcal{P}$

Table 3. Complexity of Boomy when the evaluation points are distinct on at least one dimension

5 Applications

5.1 Verifiable Computation

Verifiable computations are being used more and more in several fields, such as cloud computing to ensure the correct behavior of an external server [29], or blockchain to improve scalability [25]. Verifiable computation is the main area of application of [23], as far as we know. This is a very special case of multivariate polynomial commitment, as the programs that are verified are often represented as a series of univariate quadratic polynomials [14]. It can therefore also be represented as a bivariate polynomial where the degree of one of its two variables is at most two [9]. We believe that Boomy can be applied to these techniques to build proofs of several evaluation points at the same time, enabling new ways to make proof aggregations, accelerating their protocols or reducing their communication complexity.

5.2 Data Availability Sampling

One of the main challenges in blockchain is the scalability issue. To address this issue, [1] proposes making light clients able to verify the availability and authenticity of block data. They based their approach on the proof of erasure codes, more exactly on two-dimensional (or more) Reed-Solomon codes to make data availability sampling [16]. Later, the Ethereum blockchain planned to make this protocol a core component of their sharding protocol and create a new transaction metadata type called blobs [7] based on polynomial commitments. However, they switched their paradigm from fraud proofs of [1] to validity proofs, i.e., using polynomial commitments. The proto-Danksharding upgrade of the blockchain is based on the commitment protocol in [18] to commit each line as a univariate polynomial. The two-dimensional encoded data can then be verified in batches of 16 evaluations, deriving the commitments of the second dimension extension using the homomorphic properties of the [18] polynomial commitment. Each validator of the blockchain, that acts as the light clients described in [1], will then have to verify two rows and two columns of encoded data.

We claim that their scheme can benefit from our polynomial commitment protocol which would reduce the size of communications, one of the new challenges that have emerged following Ethereum's sharding proposal. Indeed, by considering the block of data one bivariate polynomial, Boomy reduces the size of the commitment to only one element. Using our special case of Section 4.1, it is possible to provide proofs with two elements for certain subsets of elements in rows or columns. Note that two proofs of elements of the same row (or column) can share one element among them using the correct monomial ordering. Therefore, it is possible to factorize these elements to reduce the global size of the proofs. It is also possible, using the special case described in Section 4.2, to split each column or row into random elements of each row (using the case where points are all different in one variable) or column respectively (using the case where points form a Cartesian product because each column contains 16 elements per row) in the manner described in [5]. This has the advantage of having a better distribution of the data in the network reducing the number of required online validators to reconstruct the data from shards. All these improvements can help to cut down communication complexity, tackling the major network challenges posed by sharding [20].

5.3 Verifiable Information Dispersal

Verifiable information dispersal (VID) [24, 8] is rather close to data availability sampling. The aim is slightly different since it focuses on securely distributing data and ensuring its integrity. In the data availability sampling used in Ethereum, validators ensure that, at a given time, the data are available and correct. Instead VID disperses the data among peers, each one receiving a shard, and provides a guarantee with each shard that it came from the same piece of information and is correct. Blockchains' scalability can also benefit from this protocol: rollups and validiums (a validium being a rollup which stores its data off-chain) can claim that a committee has received the correct data and made it available [22]. VID is used by the committee to ensure the correct reception and integrity of the data, its members certify this via a threshold signature sent to the blockchain.

Boomy can be beneficial to VID by reducing communication complexity, paving the way for new protocols for storage-constrained systems like blockchains.

References

1. Al-Bassam, M., Sonnino, A., Buterin, V., Khoffi, I.: Fraud and data availability proofs: Detecting invalid blocks in light clients. pp. 279–298 (2021).

https://doi.org/10.1007/978-3-662-64331-0˙15
2. Bardet, M., Faugere, J.C., Salvy, B.: On the Complexity of the F5 Gröbner basis Algorithm. arXiv preprint arXiv:1312.1655 (2013). https://doi.org/10.48550/arXiv.1312.1655
3. Boneh, D., Boyen, X.: Short signatures without random oracles. pp. 56–73 (2004). https://doi.org/10.1007/978-3-540-24676-3˙4
4. 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
5. Boneh, D., Nikolaenko, V.: Data availability sampling and danksharding: An overview and a proposal for improvements. a16zcrypto. April (2023), https://a16zcrypto.com/posts/article/an-overview-of-danksharding-and-a-proposal-for-improvement-of-das/
6. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. pp. 315–334 (2018). https://doi.org/10.1109/SP.2018.00020
7. Buterin, V.: Proto-danksharding faq. HackMD. July (2022), https://notes.ethereum.org/@vbuterin/proto_danksharding_faq
8. Cachin, C., Tessaro, S.: Asynchronous verifiable information dispersal. In: 24th IEEE Symposium on Reliable Distributed Systems (SRDS'05). pp. 191–201. IEEE (2005). https://doi.org/10.1109/RELDIS.2005.9
9. Chen, B., Bünz, B., Boneh, D., Zhang, Z.: HyperPlonk: Plonk with linear-time prover and high-degree custom gates. pp. 499–530 (2023). https://doi.org/10.1007/978-3-031-30617-4˙17
10. Chiesa, A., Hu, Y., Maller, M., Mishra, P., Vesely, P., Ward, N.P.: Marlin: Preprocessing zkSNARKs with universal and updatable SRS. pp. 738–768 (2020). https://doi.org/10.1007/978-3-030-45721-1˙26
11. Cox, D., Little, J., O'Shea, D., Sweedler, M.: Ideals, varieties, and algorithms. American Mathematical Monthly **101**(6), 582–586 (1994). https://doi.org/10.1007/978-3-319-16721-3
12. Damgård, I.: Commitment schemes and zero-knowledge protocols. In: School organized by the European Educational Forum. pp. 63–86. Springer (1998). https://doi.org/10.1007/3-540-48969-X_3
13. Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. pp. 33–62 (2018). https://doi.org/10.1007/978-3-319-96881-0˙2
14. 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
15. Guo, F., Mu, Y., Chen, Z.: Identity-based encryption: How to decrypt multiple ciphertexts using a single decryption key. pp. 392–406 (2007). https://doi.org/10.1007/978-3-540-73489-5˙22
16. Hall-Andersen, M., Simkin, M., Wagner, B.: Foundations of Data Availability Sampling. Cryptology ePrint Archive, Report 2023/1079 (2023), https://eprint.iacr.org/2023/1079
17. Hopwood, D., Bowe, S., Hornby, T., Wilcox, N., et al.: Zcash protocol specification. GitHub: San Francisco, CA, USA (2016)
18. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-size commitments to polynomials and their applications. pp. 177–194 (2010). https://doi.org/10.1007/978-3-642-17373-8˙11
19. Kohlweiss, M., Maller, M., Siim, J., Volkhov, M.: Snarky ceremonies. pp. 98–127 (2021). https://doi.org/10.1007/978-3-030-92078-4˙4

20. Król, M., Ascigil, O., Rene, S., Rivière, E., Pigaglio, M., Peeroo, K., Stankovic, V., Sadre, R., Lange, F.: Data Availability Sampling in Ethereum: Analysis of P2P Networking Requirements. arXiv preprint arXiv:2306.11456 (2023). https://doi.org/10.48550/arXiv.2306.11456
21. Lee, J.: Dory: Efficient, transparent arguments for generalised inner products and polynomial commitments. pp. 1–34 (2021). https://doi.org/10.1007/978-3-030-90453-1˙1
22. Nazirkhanova, K., Neu, J., Tse, D.: Information dispersal with provable retrievability for rollups. In: Proceedings of the 4th ACM Conference on Advances in Financial Technologies. pp. 180–197 (2022). https://doi.org/10.1145/3558535.3559778
23. Papamanthou, C., Shi, E., Tamassia, R.: Signatures of correct computation. pp. 222–242 (2013). https://doi.org/10.1007/978-3-642-36594-2˙13
24. Rabin, M.O.: Efficient dispersal of information for security, load balancing, and fault tolerance. Journal of the ACM (JACM) **36**(2), 335–348 (1989). https://doi.org/10.1145/62044.62050
25. Thibault, L.T., Sarry, T., Hafid, A.S.: Blockchain scaling using rollups: A comprehensive survey. IEEE Access (2022). https://doi.org/10.1109/ACCESS.2022.3200051
26. Van Der Hoeven, J.: On the complexity of multivariate polynomial division. In: Applications of Computer Algebra: Kalamata, Greece, July 20–23 2015. pp. 447–458. Springer (2017). https://doi.org/10.1007/978-3-319-56932-1_28
27. Von Zur Gathen, J., Gerhard, J.: Modern computer algebra. Cambridge university press (2013). https://doi.org/10.1017/CBO9781139856065
28. Xie, T., Zhang, Y., Song, D.: Orion: Zero knowledge proof with linear prover time. pp. 299–328 (2022). https://doi.org/10.1007/978-3-031-15985-5˙11
29. Yu, X., Yan, Z., Vasilakos, A.V.: A survey of verifiable computation. Mobile Networks and Applications **22**, 438–453 (2017). https://doi.org/10.1007/s11036-017-0872-3
30. Zapico, A., Buterin, V., Khovratovich, D., Maller, M., Nitulescu, A., Simkin, M.: Caulk: Lookup arguments in sublinear time. pp. 3121–3134 (2022). https://doi.org/10.1145/3548606.3560646