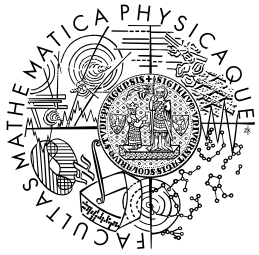


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Bc. Kateřina Bžatková

Multivariate polynomial commitment schemes

Computer Science Institute of Charles University

Supervisor of the master thesis: Mgr. Pavel Hubáček, Ph.D.

Study programme: Mathematics

Study branch: Mathematics for Information Technologies

Prague 2022

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

I would like to thank my supervisor Pavel Hubáček for his long-term support, patience, valuable advice, and help with any problems. I would like to thank my family for being there for me during the entire period of my studies.

Title: Multivariate polynomial commitment schemes

Author: Bc. Kateřina Bžatková

Institute: Computer Science Institute of Charles University

Supervisor: Mgr. Pavel Hubáček, Ph.D., Computer Science Institute of Charles University

Abstract: This thesis focuses on polynomial commitment schemes – cryptographic protocols that allow committing to a polynomial and, subsequently, proving the correctness of evaluations of the committed polynomial at requested points. As our main results, we present new schemes that enable committing to multivariate polynomials and efficiently proving the correctness of evaluations at multiple points. As the main technical tools for our constructions, we use theorems from abstract algebra related to ideals of polynomial rings and some group-theoretic properties. Compared to the state-of-the-art that inspired our work, our main contribution is the improved communication complexity achieved by our protocol.

Keywords: polynomial commitment schemes, arguments of knowledge, Hilbert's weak Nullstellensatz, Gröbner basis

Contents

Introduction

What is a polynomial commitment scheme?

Generally, a commitment scheme is a fundamental component in many cryptographic protocols, where some *statement* needs to be committed by one participant and then verified by the other. The very common and direct are applications of polynomial commitments in zero-knowledge protocols, where the essence is Prover's effort not to reveal any additional information about the statement, just persuading about Prover's knowledge. Zero-knowledge protocols are, for example, used in authentication systems or Zcach cryptocurrency.

In this thesis, we consider more specific *polynomial* commitment schemes based on the polynomial statement, as the name indicates.

The commitment scheme involves two participants: a Prove and a Verifier, typically, we write $\mathcal{V}$. Briefly speaking, Prover is responsible for committing to a *polynomial*. However, the own polynomial keeps secret and does not reveal the polynomial to Verifier.

Via an interactive protocol, the Prover tries to prove the knowledge of the secretly committed polynomial. At the end of the communication, Verifier decides if he has been convinced or not. In the notation of the commitment scheme, we use accept or reject as the Verifier's response.

The polynomial commitment scheme comprises three parts: Setup, Commit, and Open. In the Setup part, all prerequisites, in our case algebraic objects such a as prime field, groups with group generators, and a pairing function between groups, necessary for the remaining part of the polynomial commitment scheme, are established. The elements in Setup are defined to ensure that the commitment scheme is secure under some security assumptions. In this thesis, we assume the Q-DLOG security assumption, precisely defined later Section 2.5, a generalization of the standard discrete logarithm assumption. In this work, the Setup part is performed by an external trusted party. In the following Commit part, only the Prover participates, where under defined scheme creates and publishes *commitment* to a secret polynomial. The Open part is an interactive protocol between Prover and Verifier, where Verifier decides if "believes" that Prover knows the secret polynomial corresponding to the commitment or not. We can say the Open part is a verification process of Prover honesty.

The polynomial commitment scheme's primary purpose is to evaluate committed polynomials in any requested points.

Let f be a polynomial over a finite field $\mathbb{F}$. At first, the Prover, following the Commit protocol, commits to some polynomial by computing and publishing the polynomial commitment. Let z be an element from $\mathbb{F}$. Then Prover knowing the committed polynomial, can output $y = f(z)$. Polynomial commitment schemes must ensure that whenever the Verifier outputs accept in the Open protocol, the value y output by the Prover is the real evaluation of the committed polynomial f in point z.

First, we want to create a protocol where the honest Prover convinces Verifier with high probability. If Prover knows the polynomial to which the commitment was created and proceeds the interactive protocol properly, Verifier outputs accept. We call this property completeness.

Secondly, we must ensure that if Prover convinces Verifier, Prover doesn't know the committed polynomial with a negligible probability. We say protocol has knowledge soundness property. Knowledge soundness assumption indicates that Commit has to be some irreversible operation since dishonest Prover cannot be able to determine committed polynomials after the commitment is established.

Related work

Polynomial commitment schemes are already a well known and studied area in the cryptographic literature (Bünz et al. [2019],Boneh et al. [2020], Kate et al. [2010]). This thesis merges approaches from two works in this area.

Bünz et al. [2019] presented a technique of building polynomial commitment schemes based on a multi-round interactive protocol. The security assumption used in their work is called the Adaptive root assumption, based on the difficulty of computing l-th root of random group element G, where l is a randomly chosen prime number. In the Setup part, a group of unknown order with a random group element is established.

The polynomial commitment is built in the following way. Polynomial over a finite field $\mathbb{F}$ is uniquely encoded by evaluating polynomial in sufficient large $\mathbb{F}$ element and then multiply by group element G. The security assumption guarantees that the polynomial cannot be recovered from the released commitment.

The Open part is a multi-round interactive protocol, where the degree of the evaluated polynomial decreases in each round. The number of rounds depends on the degree of the committed polynomial.

This thesis mainly extends work by Boneh et al. [2020], where the authors used another technique for building a polynomial commitment scheme focused on reducing Prover's communication complexity.

In Boneh et al. [2020], the authors suggested computing in two additive groups, for which they require the Q-DOG assumption. For efficient computation, their protocol uses a mapping called *pairing*, which efficiently maps the element from two additive groups $\mathbb{G}_1, \mathbb{G}_2$ to another multiplicative group $\mathbb{G}_t$. For additive groups, the authors suggested the Q-DLOG assumption, generalized of the discrete logarithm assumption.

In the Setup part, *structured reference strings* denoted by SRS are constructed and limited by the parameter Q corresponding to the parameter from the Q-DLOG assumption. SRS contains all possible monomials that could be included in the committed polynomials. The encoding is performed by group elements from groups $\mathbb{G}_1, \mathbb{G}_2$. Based on the structure of SRS, the authors prove the completeness and knowledge soundness property for interactive proofs of their Open part.

The commitment is created using SRS. By the coefficients of the polynomial, which is supposed to be committed, the Prover combines encoded monomials

from SRS and outputs a single element of $\mathbb{G}_1$.

During the Open part, the Prover uses the same method to compute the combination of SRS for other polynomials. At the end of the Open protocol, the Verifier checks some identity between elements computed by SRS. The computation is efficient by using defined pairing.

Our results

However, the polynomial commitment scheme from Boneh et al. [2020] is only for univariate polynomials. Our scheme extends their work and provides a scheme for multivariate polynomials.

The scheme exploits ideal theory from abstract algebra with Hilbert's weak Nullstellensatz as the main theorem. For instance, other claims and algorithms participating in our protocol are Buchberger's algorithm for computation of a Gröbner basis, claims appearing in Gröbner basis theory, and the Multivariate Division algorithm.

Our work is focused on achieving a small size of proof, we can also say Prover's communication complexity. Since Bünz et al. [2019] also defined the scheme for multivariate polynomials, a comparison with our scheme suggests itself.

In protocol by Bünz et al. [2019], the proof size depends on the number of rounds in Open protocol. We outline that the complexity is $\mathcal{O}(\mu \cdot \log(d)\mathbb{G})$, where d is the total degree of the polynomial, μ is the number of polynomial variables, and $\mathbb{G}$ denotes single operation in the group defined in Setup.

Our work reduces the size of Prover's communication to only one group element in the case we consider the scheme for univariate polynomial and to $\mathcal{O}(\mu\mathbb{G})$ in the case of the multivariate polynomial. The communication complexity comparison is more precisely described in Chapter 5.

1. Ideal Theory

This chapter introduces a part of algebra used in our construction of polynomial commitment schemes.

1.1 Hilbert's weak Nullstellensatz

One of the essential theorems for our work is the weak version of Hilbert's Nullstellensatz from in Boneh et al. [2020] (Theorem 3.15.), which allows us to convert the problem of checking the correctness of polynomial evaluations to verifying whether a polynomial belongs to an ideal. First, we present some standard algebraic definitions.

Definition 1 (Ideal polynomial ring)**.**
Let $\mathbb{F}$ be a field, $R = \mathbb{F}[X_1, \ldots, X_\mu]$ a polynomial ring. A subset $I \subseteq R$ is an ideal, if it satisfies:

i) $0 \in I$,

ii) If $f_1, f_2 \in I$, *then* $f_1 + f_2 \in I$,

iii) If $f \in I$ *and* $r \in R$, *then* $r \cdot f \in I$.

Definition 2 (Ideal generated by a finite set)**.** *Let $R = \mathbb{F}[X_1, \ldots, X_\mu]$ be a polynomial ring, I an ideal in R, and $g_1, \ldots, g_s$ polynomials in R. We say the ideal I is generated by $g_1, \ldots, g_s$, denote by $I = \langle g_1, \ldots, g_s \rangle$, if it holds that:*

$$I = \{f |\ f = h_1 g_1 + \cdots + h_s g_s, h_i \in R\}$$

Theorem 1 (Hilbert's weak Nullstellensatz)**.** *Let $\mathbb{F}$ be a field, $\alpha_1, \ldots, \alpha_\mu \in \mathbb{F}$ and $R = \mathbb{F}[X_1, \ldots, X_\mu]$ a polynomial ring. Then, $\langle X_1 - \alpha_1, \ldots, X_\mu - \alpha_\mu \rangle$ is a maximal ideal, and, for each polynomial $f \in R$, the following holds:*

$$f \in \langle (X_1 - \alpha_1), \ldots, (X_\mu - \alpha_\mu) \rangle \Leftrightarrow f(\alpha_1, \ldots \alpha_\mu) = 0.$$

Hilbert's weak Nullstellensatz allows us to use an ideal identity for verifying the correctness of polynomial evaluation. Let $r \in \mathbb{F}$ be an alleged value of polynomial at evaluation point $\bar{z} = z_1, \ldots, z_\mu$, i.e., $r = f(z_1, \ldots z_\mu)$. Then, we need to verify whether $f(z_1, \ldots z_\mu) - r = 0$. By using Hilbert's weak Nullstellensatz, it is equivalent if $f(X_1, \ldots X_\mu) - r$ could be generated by the basis $(X_1 - z_1, \ldots, X_\mu - z_\mu)$, i.e., if there exist $h_1, \ldots h_\mu \in F[X_1, \ldots, X_\mu]$ s.t.

$$f(X_1, \ldots, X_\mu) - r = \sum_{i=1}^{\mu} h_i(X_1, \ldots, X_\mu)(X_i - z_i).$$

1.1.1 Multiple evaluation points

More generally, Hilbert's weak Nullstellensatz can be used to verify the correctness of evaluation at multiple points. For $\bar{\alpha} = (\alpha_1, \ldots, \alpha_\mu) \in \mathbb{F}^\mu$, denote by I_α the ideal $(X_1 - \alpha_1, \ldots, X_\mu - \alpha_\mu)$.

Using Hilbert's weak Nullstellensatz (Theorem 1) for a multivariate polynomial $f \in \mathbb{F}[X_1, \ldots, X_\mu]$ and evaluation points $\bar{\alpha}^{(1)}, \ldots, \bar{\alpha}^{(t)} \in \mathbb{F}^\mu$, we have

$$\forall i \in [t] \ : \ f(\bar{\alpha}^{(i)}) = 0 \quad \Leftrightarrow \quad \forall i \in [t] \ : \ f \in I_{\bar{\alpha}^{(i)}} \quad \Leftrightarrow \quad f \in \bigcap_{i \in [t]} I_{\bar{\alpha}^{(i)}}. \tag{1.1}$$

For shortened notation, by $[t]$ we denote the set of values $\{1, \ldots, t\}$.

Denote by I the intersection of the above ideals $I_{\bar{\alpha}^{(i)}}$, i.e.,

$$I := \bigcap_{i \in [t]} I_{\bar{\alpha}^{(i)}}.$$

For all $i \in [t]$, the ideals $I_{\bar{\alpha}^{(i)}}$ are maximal in $\mathbb{F}[X_1, \ldots, X_\mu]$ and, thus, pairwise comaximal. Since, for all pairs of comaximal ideals J_1 and J_2, we have the equality $J_1 \cap J_2 = J_1 J_2$, we can express I as follows

$$I = \bigcap_{i \in [t]} I_{\bar{\alpha}^{(i)}} = \prod_{i \in [t]} I_{\bar{\alpha}^{(i)}}.$$

Since we know how to construct the ideal I, we can derive the structure of the ideal I and then precisely express its size.

The size of basis of the ideal I depends on parameters μ and t. Specifically, its elements are all possible combinations of products of polynomials $(X_j - \alpha_j{}^{(i)})$. Consider all possible mappings $\sigma : [t] \to [\mu]$ and denote the k-th of these mappings by σ_k.

Then, we can express the k-th basis element of the ideal I as follows:

$$g_k = \prod_{i=1}^{t} \left(X_{\sigma_k(i)} - \alpha^{(i)}_{\sigma_k(i)} \right). \tag{1.2}$$

Since the number of all possible mappings σ is μ^t, the number of basis polynomials is also μ^t, i.e., for some $g_1, \ldots, g_{\mu^t} \in \mathbb{F}[X_1, \ldots, X_\mu]$

$$I = \langle g_1, \ldots, g_{\mu^t} \rangle.$$

We extend using Hilbert's weak Nullstellensatz for the case we want to verify the correctness of polynomial evaluation in more points.

Let $r \in \mathbb{F}_{<t}[X_1, \ldots X_\mu]$ be polynomial which should satisfy $f(\bar{\alpha}^{(i)}) = r(\bar{\alpha}^{(i)})$ for all evaluating points $\bar{\alpha}^{(1)}, \ldots, \bar{\alpha}^{(t)}$. We limit the degree of polynomial r to t. It follows from the fact each interpolation polynomial with t evaluation point has a degree at most t.

Using the relation (1.1), we convert the verification of polynomial equality to verify whether their difference belongs to the ideal I.

Specifically, instead of proving $f(\bar{\alpha}^{(i)}) - r(\bar{\alpha}^{(i)}) = 0$ we can use the following corollary of Hilbert's weak Nullstellensatz (Theorem 1) for more evaluation points.

Corollary 1. *Let $f \in \mathbb{F}[X_1, \ldots, X_\mu]$, $\bar{\alpha}^{(1)}, \ldots, \bar{\alpha}^{(t)} \in \mathbb{F}^\mu$ are evaluating points, and ideal $I = (g_1, \ldots, g_{\mu^t})$, where $g_k = \prod_{i=1}^{t} \left(X_{\sigma_k(i)} - \alpha^{(i)}_{\sigma_k(i)} \right)$.*
Let r be a polynomial in $\mathbb{F}_{<t}[X_1, \ldots, X_\mu]$. For each $j \in [t]$ following holds:

$$f(\bar{\alpha}^{(j)}) - r(\bar{\alpha}^{(j)}) = 0 \Leftrightarrow f(\bar{\alpha}^{(j)}) - r(\bar{\alpha}^{(j)}) \in I$$

$$\Leftrightarrow \exists \ h_1, \ldots, h_\mu \in \mathbb{F}[X_1, \ldots, X_\mu] : f - r = \sum_{i=1}^{\mu^t} h_i g_i.$$

1.1.2 Computing polynomials h_i

Corollary 1 guarantees the existence of polynomials h_i from decomposition of polynomial $f - r$. In this section, we establish how these polynomials can be computed. For this purpose, we use the Multivariate Division Algorithm 1.

Before presenting the complete algorithm, we recall some necessary definitions and notations from the Gröbner basis theory based on Adams [1994].

We denote $\mathbb{T}^n = \{x_1^{e_1} \cdots x_n^{e_n} | e_1, \ldots, e_n \in \mathbb{N}_0\}$.

Remark (Monomial ordering). Monomial ordering over $\mathbb{T}^n$ is a linear ordering $<$ over set $\mathbb{T}^n$ satisfying:

i) $1 < x^\epsilon$, for all $x^\epsilon \in \mathbb{T}^n \setminus \{1\}$

ii) if $x^\alpha < x^\beta$, then $x^\alpha x^\gamma < x^\beta x^\gamma$, for all $x^\alpha, x^\beta, x^\gamma \in \mathbb{T}^n$.

Remark. Let $f = a_n x_1^{n_1} \cdots x_\mu^{n_\mu} + \cdots + a_1 x_1 \cdots x_\mu + a_0$ be a multivariate polynomial. Then we denote *total degree* the largest sum of the exponents in one single monomial.
We denote $lt(f) = a_n x_1^{n_1} \ldots x_\mu^{n_\mu}$ the *leading term*, i.e. the term with largest sum of exponent, in case of equality the defined ordering decides.
We denote $lm(f) = x_1^{n_1} \cdots x_\mu^{n_\mu}$ the *leading monomial*, i.e the leading term without the coefficient.

Definition 3. *Given $f, g, h \in \mathbb{F}[X_1, \ldots, X_\mu]$, with $g \neq 0$, we say that f reduces to h mod g written*

$$f \to^g h$$

iff $lm(g)$ divides non-zero term X that appears in f and

$$h = f - \frac{X}{lt(g)} g.$$

Definition 4. *Let $f, h \in \mathbb{F}[X_1, \ldots, X_\mu]$ and $F \subseteq \mathbb{F}[X_1, \ldots, X_\mu] \setminus \{0\}$.*
$F = \{f_1, \ldots, f_s\}$. We say that f reduces to h modulo F, denoted

$$f \to^F h$$

iff there exists a sequence of indices $i_1, \ldots i_t \in \{1, \ldots, s\}$ and a sequence of polynomials $h_1, \ldots, h_{t-1}$ such that

$$f \to^{f_{i_1}} h_1 \to^{f_{i_2}} h_2 \ldots \to^{f_{i_{t-1}}} h_{t-1} \to^{f_{i_t}} h$$

Definition 5 (reduced polynomial)**.** *Let $G \subseteq \mathbb{F}[x_1, \ldots, x_\mu] \setminus \{0\}$. Polynomial $\rho \in \mathbb{F}[x_1, \ldots, x_\mu]$ is called reduced with respect to G, if for every $g \in G$ none of the terms included in ρ is divisible by $lm(g)$. In other words, ρ cannot be reduced modulo G.*

Now we cover all prerequisites to introduce an important algorithm, Multivariate Division Algorithm. (Adams [1994] Algorithm 1.5.1)

Algorithm 1 Multivariable Division Algorithm

INPUT: $f, G = \{g_1, \ldots, g_s | g_i \in \mathbb{F}[X_1, \ldots, X_\mu]\}$
OUTPUT: $h_1, \ldots h_s,\ \rho \in \mathbb{F}[X_1, \ldots, X_\mu]$, where ρ is reduced mod G,
$f = h_1 g_1 + \cdots + h_s g_s + \rho,\ lm(f) = max\{lm(h_i g_i), \rho | i = 1, \ldots, s\}$

1: $h_1 := 0, \ldots h_s := 0, \rho := 0, q := f$
2: **while** $q \neq 0$ **do**
3: **if** $\exists\ i \in 1, \ldots s, st. lm(g_i) | lm(q)$ **then**
4: choose min i (according to given monomial ordering) st. $lm(g_i) | lm(q)$
5: $h_i := h_i + \frac{lt(q)}{lt(g_i)}$
6: $q := q - \frac{lt(q)}{lt(g_i)} g_i$
7: **else** $\rho = \rho + lt(q), q = q - lt(q)$
8: **end if**
9: **end while**
10: RETURN $h_1, \ldots h_s, \rho$

Generally, G could be any ideal basis and the algorithm outputs a decomposition $f = h_1 g_1 + \cdots + h_s g_s + \rho$, where ρ is a non-zero polynomial. However, for our purpose in the polynomial commitment scheme, we need to work with a more specific basis called Gröbner basis, by Adams [1994] (Definition 1.6.1).

Definition 6 (Gröbner basis)**.** *Let $I \subseteq \mathbb{F}[x_1, \ldots, x_\mu]$ be an ideal.*
$G = \{g_1, \ldots, g_t\} \subseteq I \setminus \{0\}$. G is called Gröbner basis of ideal I if, for every $f \in I, f \neq 0$, there is a $g_i \in G$ s.t. $lm(g_i) | lm(f)$.

To solve the problem of finding Gröbner basis of a given ideal, we use a well-known Buchberger's algorithm Adams [1994] (Algorithm 1.7.1). The algorithm gets a set of polynomials determining the ideal as an input and outputs a Gröbner basis of this ideal. For our purpose, we need the algorithm to be deterministic. In the general version, the algorithm is introduced as non-deterministic when there is some randomness in choosing pair of polynomials from a given set. We can remove randomness by establishing that polynomials are chosen under a given monomial ordering.

Let's write down a known estimated complexity of the algorithm:

$$d^{2^{\mu + o(1)}},$$

where d is *total degree* of input multivariate polynomials, μ is number of variables.

The following equivalence from Adams [1994] (Theorem 1.6.2) states how a Gröbner basis is related with a polynomial decomposition output by Algorithm 1 (*Multivariate Division Algorithm*).

Proposition 1. *Let $0 \neq I \subseteq \mathbb{F}[x_1, \ldots, x_\mu], G = \{g_1, \ldots, g_s\} \subseteq I, g_1, \ldots g_s \neq 0$. Let $\mathbb{T}^n$ be a monomial ordering. Then the following conditions are equivalent.*

i) G is a Gröbner basis of ideal I

ii) $f \in I \Leftrightarrow f \rightarrow^G 0$

iii) $f \in I$ if and only if $f = \sum_{i=1}^{s} h_i g_i$, $h_1, \ldots, h_s \in \mathbb{F}[x_1, \ldots, x_\mu]$ and $lm(f) = \max\{lm(h_i) lm(g_i) | i = 1, \ldots, s\}$

Suppose that a Gröbner basis of a given ideal is already computed. The following lemma states that using Algorithm 1 (Multivariable Division Algorithm), we obtain the required polynomials h_i.

Lemma 2. *Let $I = \langle G \rangle$ be an ideal $\subseteq \mathbb{F}[X_1, \ldots, X_\mu]$ and G be a Gröbner basis of ideal I. Multivariable Division Algorithm (Algorithm 1) with an input $G = \{g_1, \ldots, g_s\}$ outputs $f = \sum_{i=1}^{s} h_i g_i$ iff $f \in I$.*

Proof. The *Multivariate Division Algorithm* generally outputs a decomposition of polynomial $f = \sum_{i=1}^{s} h_i g_i + \rho$, where r is modulo G, in the other words $f \to^G \rho$.

By previous Proposition 1, since G is a Gröbner basis of the ideal I, $f \in I \Leftrightarrow f \to^G 0$. It implies $f \in I \Leftrightarrow r = 0$. Finally, $f \in I$ iff *Multivariate Division Algorithm* outputs $f = \sum_{i=1}^{s} h_i g_i,\ (\rho = 0)$. □

Summary . Let f be a polynomial in which we compute an evaluation and r be a polynomial which satisfies $f(\bar{\alpha}^{(i)}) = r(\bar{\alpha}^{(i)})$ for all $i \in [t]$. The evaluation points $\bar{\alpha}^{(i)}$ determine the ideal I by $I = (g_1, \ldots, g_k)$, where $g_k = \prod_{i=1}^{t} \left(X_{\sigma_k(i)} - \alpha^{(i)}_{\sigma_k(i)}\right)$. Using Buchberger's algorithm, we compute Gröbner basis $\tilde{g_1}, \ldots, \tilde{g_k}$ of the ideal I. By Corollary 1, the polynomial $f - r \in I$. Finally by Lemma 2, using the Multivariate Division Algorithm (Algorithm 1) with polynomial $f - r$ and Gröbner basis as inputs we obtain required polynomials h_i.

The following example demonstrates the necessity of calculating Gröbner basis. *Multivariate Division Algorithm* cannot be directly run for any basis of the ideal I since it will not be ensured that even for $f - r \in I$, the algorithm outputs $\sum_{i=1}^{s} h_i g_i$.

Example. Let $\mathbb{F} = \mathbb{Z}_{17}$, and all polynomials are considering in polynomial ring $\mathbb{Z}_{17}[X_1, X_2]$. Let $\bar{\alpha_1}, \bar{\alpha_2}$ be evaluating points and I_1, I_2 are corresponding ideals:

$$\bar{\alpha_1} = (2, 3), I_1 = \langle X_1 - 2, X_2 - 3 \rangle$$
$$\bar{\alpha_2} = (1, 4), I_2 = \langle X_1 - 1, X_2 - 4 \rangle$$

Using equation 1.2 we compute the intersection of I_1, I_2.

$$\begin{aligned} I &= \langle (X_1 - 2)(X_1 - 1), (X_1 - 2)(X_2 - 4), (X_1 - 3)(X_1 - 1), (X_2 - 3)(X_2 - 4) \rangle \\ &= \langle X_1^2 - 3X_1 + 2,\ X_1X_2 - 2X_2 - 4X_1 + 8,\ X_1X_2 - 3X_1 - X_2 + 3, \\ &\quad X_2^2 - 7X_2 + 12 \rangle \end{aligned}$$

We can consider these polynomials as basis g_1, g_2, g_3, g_4 of ideal I. However, we cannot directly use *Multivariable Division Algorithm* to successfully compute a decomposition of some polynomial f.

Let compute with the following polynomial:

$$f - r = (X_1X_2 - 3X_1 - X_2 + 3) - (X_1X_2 - 2X_2 - 4X_1 + 8) = X_1 + X_2 - 5$$

This polynomial is obviously an element of I because it is a linear combination of two generating polynomials g_3, g_2. Since none of the leading monomials of generating polynomials could divide polynomial f, *Multivariate Division Algorithm* outputs nonzero residual polynomial ρ. Thus, the algorithm does no give the required decomposition.

2. Preliminaries and Definitions

2.1 Commitment scheme

For an introduction, write down a general definition of the commitment scheme. Later in this chapter, after all the necessary prerequisites, we also define a polynomial commitment scheme.

Security parameter The security of the protocol has to be computed against some given parameter. We call this parameter *security parameter*, and denote by λ.

By convention, we use the unary representation 1^λ for inputs of algorithms. Probabilistic polynomial-time algorithms (PPT) run in polynomial time with respect to the size of inputs. Since we consider an input 1^λ, PPT algorithms run in λ polynomial-time.

Generally, in cryptography, it is sufficient to prove that some probability is negligible in the security parameter λ.

Definition 7. *A function $\rho : \mathbb{N} \to \mathbb{R}$ is negligible, if for every integer $c > 0$ there exists an integer n_c such that for all $x > n_c$*

$$|\rho(x)| < \frac{1}{x^c}.$$

At this point, we can write a general definitions of *commitment scheme* with required property *binding*, both presented in Bünz et al. [2019](Definition 4).

Definition 8 (Commitment scheme)**.** *A commitment scheme Γ is a tuple (Setup, Commit, Open) of PPT algorithms where:*

- $\mathsf{Setup}\ (1^\lambda) \to \mathsf{pp}$ *generates public parameters* pp
- $\mathsf{Commit}\ (pp, x) \to (c)$ *takes a secret message x and outputs a public commitment c.*
- $\mathsf{Open}\ (pp, c, x) \to b \in \{0, 1\}$ *verifies the opening of commitment c to the secret message x.*

  The Open is an interactive argument of knowledge.

Generally, *commitment scheme* is required to be *binding*. It means no efficient adversary can create a commitment for two different messages.

Definition 9 (Binding)**.**
A commitment scheme Γ is binding if for all PPT adversaries $\mathcal{A}$:

$$\Pr\left[b_0 = b_1 \neq 0 \wedge x_0 \neq x_1 \quad : \quad \begin{array}{l} \mathsf{pp} \leftarrow Setup(1^\lambda) \\ (c, x_0, x_1) \leftarrow \mathcal{A}(\mathsf{pp}) \\ b_0 \leftarrow \mathsf{Open}(\mathsf{pp}, c, x_0) \\ b_1 \leftarrow \mathsf{Open}(\mathsf{pp}, c, x_1) \end{array} \right] \leq \mathsf{negl}(\lambda)$$

2.1.1 Interactive argument of knowledge

According to the definition of a commitment scheme, Open part is an interactive argument of knowledge. The interactive argument of knowledge is a more demanding interactive protocol. Generally, in the interactive protocol, a Prover convinces a Verifier that c is the commitment to a message m, while the message m remains a Prover's secret.

Furthermore, in an interactive argument of knowledge, if Prover convinces Verifier, Prover truly knows the secret message m with high probability.

We have to establish a new term relation $\mathcal{R}$. Generally, let x be a statement and w be a witness. Then the relation $\mathcal{R}$ is a set of pairs (x, w) which satisfies the defined condition for $\mathcal{R}$.

For instance, let $x = (a, b)$, where a, b are elements in $\mathbb{F}$ and w be a polynomial $f \in \mathbb{F}[X_1, \ldots, X_\mu]$. Then we can define relation $\mathcal{R}$ in following way: pair (x, w) is in $\mathcal{R}$ iff $b = f(a)$.

Now we can introduce the interactive argument of knowledge more formally from Bünz et al. [2019](Definition 1).

Definition 10 (Interactive argument of knowledge)**.**
Let $\mathcal{P}, \mathcal{V}$ be PPT (probabilistic polynomial-time) interactive algorithms.
Let Setup$(1^\lambda) \to$ (pp) denote a non-interactive algorithm with given security parameter λ outputting public parameters pp*.*

Both $\mathcal{P}$ and $\mathcal{V}$ have access to pp *and both are given statement x. Prover has on input w in addition. Let $\langle \mathcal{P}(pp, x, w), \mathcal{V}(pp, x) \rangle$ denote the output of $\mathcal{V}$ after its interaction with $\mathcal{P}$. Verifier outputs* accept *or* reject*.*

The triple (Setup*, $\mathcal{P}, \mathcal{V}$) is called an argument of knowledge for relation $\mathcal{R}$ if protocols satisfy completeness and knowledge soundness conditions.*

By Verifier's output accept, we understand that Prover convinces Verifier. Otherwise, reject means that Verifier has not been convinced.

To complete the Definition 10, we have to define *completeness* and *knowledge soundness*. We use the definition from Bünz et al. [2019](Definition 1).

Definition 11 (Completeness)**.**
Interactive protocol has property completeness if every honest Prover $\mathcal{P}$ given $(x, w) \in \mathcal{R}$ convinces Verifier (i.e. $\langle \mathcal{P}(pp, x, w), \mathcal{V}(pp, x) \rangle =$ accept *) with probability one.*

AGM model In this thesis, we work in the Algebraic Group Model (AGM) of Fuchsbauer et al. [2018] (section 1.2). The AGM puts more restrictions on the adversaries than a Standard Model. Informally, whenever an algebraic adversary computes some group element, it also has to uncover the group representation of the element with respect to its input.

Definition 12 (Algebraic adversary)**.**
Whenever algebraic adversary $\mathcal{A}$ outputs an element $g \in \mathbb{G}$, it also outputs a vector $\bar{v} = (v_1, \ldots, v_n) \in \mathbb{F}^n$ such that $g = \bar{v}^T \bar{l} = \prod_{i=1}^{l} v_i l_i$, where $l_1, \ldots, l_n$ is the list of all group elements from $\mathbb{G}$ that are given to the adversary.

We introduce the *knowledge soundness* property (Boneh et al. [2020], Definition 2.3.), where we suppose an algebraic adversary participates.

Definition 13 (Knowledge soundness in AGM)**.**
Interactive protocol has knowledge soundness in AGM if there exists an efficient algorithm E such that for any algebraic adversary $\mathcal{A}$ the probability of adversary's winning of the following game is $\mathsf{negl}\ (\lambda)$ *over the randomness of $\mathcal{A}$.*

1. *A chooses input x and plays the role of $\mathcal{P}$ in the protocol with input x.*
2. *Given access to outputs of algebraic adversary $\mathcal{A}$ (including the vector of representation of group elements), E outputs w dependent on this vector.*
3. *$\mathcal{A}$ wins if*
   - *$\mathcal{V}$ outputs* accept *at the end of the protocol and*
   - *$(x, w) \notin \mathcal{R}$*

2.2 Group operation

This thesis suggests a polynomial commitment scheme, which security is based on a group assumption. Furthermore, we work in more groups and we need to efficiently switch between them. We start with efficient group mapping called a *pairing*. The *pairing* naturally maps elements from two additive groups to another multiplicative group.

Definition 14 (Pairing)**.** *Let $\mathbb{G}_1, \mathbb{G}_2$ be two additive cyclic groups of prime order q. Let $\mathbb{G}_t$ be a multiplicative cyclic group of the same order q. The natural mapping between groups $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t$ is called pairing, satisfying the following properties:*

- *$e(g_1, g_2) = g_t$, where g_1, g_2, g_t are group generators of $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t$,*
- *Bilinearity: $\forall x, y \in \mathbb{F}_q^*, \forall g_1 \in \mathbb{G}_1, \forall g_2 \in \mathbb{G}_2 : e(x \cdot g_1, y \cdot g_2) = e(g_1, g_2)^{xy}$,*
- *Non-degeneracy: $e(g_1, g_2) \neq 1 \in \mathbb{G}_t$,*
- *Computability: Efficient algorithm computing e exists.*

Remark. Later in this work, we use a shorter notation for some of these operations. Let $x \in \mathbb{F}, g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2, \{g_{t_1}, g_{t_2}\} \in \mathbb{G}_t$ as above. For $e(g_1, g_2)$, we use $g_1 \odot g_2$. For operation $x \cdot g_1$, we use $[x]_1 = x \cdot g_1$, similarly $[x]_2 = x \cdot g_2$.

Remark. Since we suppose to use both additive and multiplicative groups, the terms multiplication have a different meaning. Let $x \in \mathbb{F}, g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2$, as above, and $\{g_{t_1}, g_{t_2}\} \in \mathbb{G}_t$. The operation $x \cdot g_1$ we call multiplication in $\mathbb{G}_1$. Similarly the operation $x \cdot g_2$ we call multiplication in $\mathbb{G}_2$. The operation $g_{t_1} \cdot g_{t_2}$ we call multiplication in $\mathbb{G}_t$.

2.3 Public parameters

All public parameters are generated in Setup algorithm, an initial non-interactive protocol in the polynomial commitment scheme. The Setup is performed by a

third trusted party, which means Prover, not even Verifier, could participate in Setup.

Generally, Setup takes security parameter as an input and outputs public parameters, denoted by pp. In other words, public parameters depend on the security parameter λ.

Specifically in this thesis, Setup outputs a representation of a finite field $\mathbb{F}$ of order $q = 2^{\omega(\lambda)}$, representations of two additive cyclic groups $\mathbb{G}_1,\ \mathbb{G}_2$ of order q, one multiplicative cyclic group $\mathbb{G}_t$ of order q, together with groups generators $g_1 \in \mathbb{G}_1,\ g_2 \in \mathbb{G}_2,\ g_t \in \mathbb{G}_t$ and group pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_t$ satisfying conditions from Definition 14.

To conclude, we have $\mathsf{pp} = (\mathbb{F}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, e)$.

2.4 Structured reference strings

In this work, the schemes are based on Structured Reference Strings (SRS), presented in Gabizon et al. [2019] (Section 2.2.). These strings are also constructed in Setup. The Setup algorithm choose uniformly $x_1, \ldots, x_\mu$ from $\mathbb{F}$, takes group elements g_1, g_2 and returns two structured reference string defined in the following way.

Definition 15 (Structured reference strings)**.**
Let $x_1, \ldots, x_\mu$ be elements in a finite field $\mathbb{F}$ and $f_1(X_1, \ldots, X_\mu), \ldots, f_l(X_1, \ldots, X_\mu)$ be polynomials over $\mathbb{F}$. Let g_1 be an element of $\mathbb{G}_1$ and g_2 an element of $\mathbb{G}_2$. Then we define μ-variables structured reference strings $\mathbf{srs_1}, \mathbf{srs_2}$ with degree Q as follows:

$$\mathbf{srs_1} = \Big\{[f_1(x_1^{d_1}, \ldots, x_\mu^{d_\mu})]_1, \ldots, [f_l(x_1^{d_1}, \ldots, x_\mu^{d_\mu})]_1, \sum_{i=1}^{\mu} d_i < Q\Big\}$$

$$\mathbf{srs_2} = \Big\{[f_1(x_1^{d_1}, \ldots, x_\mu^{d_\mu})]_2, \ldots, [f_l(x_1^{d_1}, \ldots, x_\mu^{d_\mu})]_2, \sum_{i=1}^{\mu} d_i < Q\Big\}$$

In other words, the degree Q restricts the total degree of polynomials appearing in the above SRS.

Generally, SRS can be determined by any polynomials $f(X)$. In our work, we use simple monomials. Specifically, SRS are described in schemes in sections 3.1, 4.1.

At this point, we can write the inputs and outputs of Setup algorithm.
Let λ is the security parameter, $\{d,\ \mu\} = poly(\lambda)$. In the Setup algorithm, pp and $\mathbf{srs_1}, \mathbf{srs_2}$ are generated:

- Setup $(1^\lambda, \mu,\ Q) \to (pp, \mathbf{srs_1}, \mathbf{srs_2})$,
  where $\mathsf{pp} = (\mathbb{F}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, e)$ are the public parameters.
  $\mathbf{srs_1}, \mathbf{srs_2}$ are μ-variables structured reference strings with degree Q satisfying Definition 15.

Generation of pp with SRS requires work from an external trusted party, which can be optimized as follows. Setup could be divided into two parts. In the first part, pp with groups $\mathbb{G}_1, \mathbb{G}_2$ and a finite field $\mathbb{F}$ are created. The second part uses pp and generates SRS by choosing random elements from $\mathbb{F}$. We emphasize that one generation of pp could be reused for more SRS generations.

2.5 Security assumptions

Once we have SRS defined, we can describe the security assumption used in our schemes. We consider completeness and knowledge soundness under a group property called the Q-DLOG assumption, presented in Gabizon et al. [2019] (Definition 2.1.). In the following definition, we define the Q-DLOG assumption for two additive groups.

Definition 16 (Q-DLOG assumption)**.**
Let Setup *be initial protocol generating* pp *and* $\mathbf{srs_1}, \mathbf{srs_2}$ *as above. Fix a positive integer* Q*. Additive groups* $\mathbb{G}_1, \mathbb{G}_2$ *have* Q*-DLOG assumption if for every polynomial algebraic adversary* $\mathcal{A}$ *following holds:*

$$\Pr\left[x \in \{x_1, \ldots, x_\mu\} \quad : \quad \begin{array}{l} \mathsf{pp}, \mathbf{srs_1}, \mathbf{srs_2} \leftarrow \mathit{Setup}(1^\lambda, \mu, Q) \\ x \in \mathbb{F} \leftarrow \mathcal{A}(\mathit{pp}, \mathbf{srs_1}, \mathbf{srs_2}) \end{array} \right] \leq \mathsf{negl}(\lambda)$$

Since we consider working in the Algebraic group model, we have to assume the algebraic adversary in the definition. Emphasize that this requirement does not change the strength of the definition.

Informally we can understand the Q-DLOG assumption as the complexity of computing a root of a given group element g (in our case group generator). For simplicity, suppose $\mu = 1$. Even the adversary knows a sequence $g_1, x \cdot g_1, x^2 \cdot g_1, \ldots, x^Q \cdot g_1$, it is hard to compute the unknown element $x \in \mathbb{F}$ in a polynomial time in λ.

2.5.1 Ideal and real pairing check

Finally, we have to mention a relation between two types of polynomial identity checks. One way to check whether the equality of some polynomials holds is to compare all their corresponding coefficients. We say we have to perform the *ideal check*. However, in our protocol, whole polynomials with all coefficients are not sent. Instead of that, the polynomials are represented using the SRS as group elements. Hence, we need to check the polynomial identity by checking whether the equation of their group representation is correct. For this purpose, we use the *real pairing check*. Both *ideal check* and *real pairing check* are introduced in Boneh et al. [2020] (Section 2.2).

Definition 17 (real pairing check)**.** *Let* $\mathbf{a}, \mathbf{b}$ *be the vectors of* $\mathbb{F}$ *elements whose encoding in* $\mathbb{G}_1, \mathbb{G}_2$ *an algebraic adversary* $\mathcal{A}$ *outputs during protocol. Real pairing check has the form:*

$$(\mathbf{a} \cdot T_1) \cdot (T_2 \cdot \mathbf{b}^T) = 0,$$

where T_1, T_2 *are matrices over* $\mathbb{F}$*.*

Matrices T_1, T_2 define the form of the equation. In the corresponding ideal check, we verify if the equation holds for the actual polynomials, which coefficients are derived from vectors $\mathbf{a}, \mathbf{b}$.

Definition 18 (ideal check)**.** *Let* $\mathbf{a}, \mathbf{b}$ *be vectors from real paring check definition. Let* $f_{i,l(x)}$ *be the* l*-th polynomial in* $\mathbf{srs_i}$ *and* $\bar{v}$ *a vector over* $\mathbb{F}$*. Then* $a_j =$

$\sum v_l f_{i,l}(x) = R_{i,j}(x)$ *for* $R_{i,j}(X) = \sum v_l f_{i,l}(X)$. *Denote, for* $i \in \{1,2\}$ *the vector of polynomials* $R_i = (R_{i,j})_j$. *The corresponding ideal check has the form*

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

At this point, we show that the replacement of the *ideal check* by *real pairing check* in the protocol is legitimate. The following proposition (presented in Gabizon et al. [2019], Lemma 2.2) states if the *ideal check* does not hold, the *real pairing check* passes with a negligible probability.

Proposition 2. *Consider the Q-DLOG assumption for additive groups* $\mathbb{G}_1, \mathbb{G}_2$. *Given a polynomial algebraic adversary* $\mathcal{A}$ *participating in a protocol - based on degree Q SRS. The probability that* $\mathcal{A}$ *can pass the real pairing check is larger by, at most, an additive* $\mathsf{negl}(\lambda)$ *than the probability of the corresponding ideal check holds.*

Even the replacement from *real pairing check* to *ideal check* is done in natural way, the notation can be confusing. We illustrate the definitions in the following example.

Example (real pairing check and corresponding ideal check).
During the protocol, the algebraic adversary outputs the following values:

$$[a_1]_1 = [x_1 + x_2]_1, [a_2]_1 = [2x_1^2 + x_2]_1, [a_3]_1 = [3x_1x_2]_1,$$
$$[b_1]_2 = [x_1^2]_2, [b_2]_2 = [x_2]_2, [b_3]_2 = [x_2^3]_2.$$

Then the vector $\mathbf{a}, \mathbf{b}$ are

$$\mathbf{a} = (x_1 + x_2, 2x_1^2 + x_2, 3x_1x_2),$$
$$\mathbf{b} = (x_1^2, x_2, x_2^3).$$

Let T_1, T_2 be matrices determine the form of the equation. The *real pairing check* has the form:

$$(x_1 + x_2, 2x_1^2 + x_2, 3x_1x_2) \cdot T_1 \cdot T_2 (x_1 + x_2, 2x_1^2 + x_2, 3x_1x_2)^T = 0$$

At this point, we derive the corresponding ideal check. We can derive the SRS elements and the vector $\bar{v}$ which includes the polynomial coefficients. We illustrate that only for a_2:

$$a_2 = 2x_1^2 + x_2 \Rightarrow \bar{v} = (2,1), f_{1,1} = x_1^2, f_{1,2} = x_2$$
$$\Rightarrow R_{1,2}(X) = 2X_1^2 + X_2.$$

Overall, we have two vectors of polynomials, each corresponding to one **srs**.

$$R_1 = (X_1 + X_2, 2X_1^2 + X_2, 3X_1X_2)$$
$$R_2 = (X_1^2, X_2, X_2^3).$$

The *ideal check* has the form:

$$(X_1 + X_2, 2X_1^2 + X_2, 3X_1X_2) \cdot T_1 \cdot T_2 (X_1 + X_2, 2X_1^2 + X_2, 3X_1x_2)^T \equiv 0$$

2.6 Schwartz–Zippel lemma

Our protocol is designed so that Prover does not reveal whole polynomials created during the protocol because the goal is to decrease Prover's communication complexity.

However, Prover has to be able to convince Verifier about some polynomial identity - that one polynomial is equal to another. In this case, Schwartz–Zippel lemma becomes useful. Under some probability, two polynomials are equal if their evaluation in random points is equal. We can slightly change the statement that the difference between two polynomials is equal to zero (i.e., two polynomials are identical) if the evaluation in a random point is equal to zero.

Precisely, the probability is estimated in the following standard lemma, presented in Coretti et al. [2018] (Lemma 37).

Proposition 3 (Schwartz–Zippel lemma)**.** *Let $\mathbb{F}$ be a finite field, $f \in \mathbb{F}[X_1, \ldots, X_\mu]$ be a non-zero polynomial of total degree $d \geq 0$. Let S be a finite subset of $\mathbb{F}$ and let $r_1, r_2, \ldots, r_\mu$ be selected at random independently and uniformly form S. Then*

$$Pr[f(r_1, r_2, \ldots, r_\mu) = 0] \leq \frac{d}{|S|}.$$

2.7 Polynomial commitment scheme

A polynomial commitment scheme is a commitment scheme (Definition 8), where the secret message x is a multivariate polynomial.

The Open protocol is more specific since not only the correctness of commitment is verified. Additionally, the protocol includes checking the polynomial evaluation at a given point. The input of Open protocol is extended by challenging value $\alpha \in \mathbb{F}$ and Prover's offered value $y \in \mathbb{F}$. In the Open protocol, Prover has to convince Verifier about equality $y = f(\alpha)$, where f is the committed polynomial.

In our thesis, we consider *polynomial commitment scheme* with structured reference strings (SRS), similarly to the scheme presented in Boneh et al. [2020] (Definition 2.3.).

Definition 19 (Polynomial commitment scheme - based on SRS)**.**
A polynomial commitment scheme Γ - based on SRS is a tuple (Setup, Commit, Open).

- Setup $(1^\lambda,\ \mu,\ Q) \rightarrow (\mathsf{pp}\ , SRS)$ *generates public parameters and public structured reference strings.*
- Commit $(SRS,\ f(X_1, \ldots, X_\mu)) \rightarrow c$
  takes a secret polynomial $f(X_1, \ldots, X_\mu) \in \mathbb{F}_{<d}[X_1 \ldots, X_\mu]$ and outputs a public commitment $c \in \mathbb{G}_1$.
- Open *is an interactive protocol between a PPT prover $\mathcal{P}$ and verifier $\mathcal{V}$. Only $\mathcal{P}$ knows $f(X_1, \ldots, X_\mu)$. Both $\mathcal{P}$ and $\mathcal{V}$ are given:*
  1. *integers t, d*
  2. pp, *SRS*

3. $c \in \mathbb{G}_1$ *an alleged commitment to* f
4. $\bar{\alpha}^{(1)}, \ldots, \bar{\alpha}^{(t)} \in \mathbb{F}^{\mu}$
5. $r \in \mathbb{F}_{<t}[X_1, \ldots, X_\mu]$, *the polynomials allegedly satisfying*

$$r(\bar{\alpha}^{(i)}) = f(\bar{\alpha}^{(i)}) \ \forall i \in [t].$$

At the end of the protocol $\mathcal{V}$ *outputs* accept *or* reject.

Open *is an interactive argument of knowledge with respect to the following relation:*

$$\mathcal{R} = \Big\{ \langle (c, r, \bar{\alpha}^{(1)}, \ldots \bar{\alpha}^{(t)}, d), f(X_1, \ldots, X_\mu) \rangle : f \in \mathbb{F}_{<d}[X_1, \ldots X_\mu], \\ \forall i : f(\bar{\alpha}^{(i)}) = r(\bar{\alpha}^{(i)}), c = [f(x_1, \ldots, x_\mu]_1)] \Big\}. \tag{2.1}$$

A polynomial commitment scheme is evaluation binding if no efficient Adversary can convince Verifier that the committed polynomial $f(X_1, \ldots X_\mu)$ evaluates to different values $r_0 \neq r_1$ in the same point $\bar{\alpha} \in \mathbb{F}^{\mu}$. We write a definition from Bünz et al. [2019], where the *evaluation binding* is defined informally.

Definition 20 (Evaluation binding)**.** *A polynomial commitment scheme* Γ *- based on SRS, is evaluation binding if for all PPT adversaries* $\mathcal{A}$*:*

$$\Pr \left[\begin{array}{l|ll} b_0 = b_1 \neq \mathsf{reject} & : & (\mathsf{pp}, \mathbf{srs_1}, \mathbf{srs_2}) \leftarrow Setup(1^\lambda) \\ \wedge \ r_0 \neq r_1 & & (c, r_0, r_1) \leftarrow \mathcal{A}(\mathsf{pp}, \mathbf{srs_1}, \mathbf{srs_2}, \bar{\alpha}) \\ & & b_0 \leftarrow \mathsf{Open}(\mathsf{pp}, c, \bar{\alpha}, r_0, f(\bar{X})) \\ & & b_1 \leftarrow \mathsf{Open}(\mathsf{pp}, c, \bar{\alpha}, r_1, f(\bar{X})) \end{array} \right] \leq \mathsf{negl}(\lambda).$$

Since Open is an interactive argument of knowledge, it satisfies the *completeness* and *knowledge soundness* conditions. Below, we specify these conditions for the *Polynomial commitment scheme*.

Completeness of polynomial commitment scheme
Let $d,\ t = poly(\lambda), \bar{\alpha}^{(1)}, \ldots, \bar{\alpha}^{(t)} \in \mathbb{F}^{\mu}, f \in \mathbb{F}_{<d}[X_1, \ldots, X_\mu], r \in \mathbb{F}_{<t}[X_1, \ldots X_\mu]$, $c \in \mathbb{G}_1$, $\mathcal{R}$ be the relation defined in 2.1. The *polynomial commitment scheme* has *completeness* if the following holds: If $\mathcal{P}$ proceeds the protocol correctly with the values $\langle (c, r, \bar{\alpha}^{(1)}, \ldots, \bar{\alpha}^{(t)}, d), f \rangle \in \mathcal{R}$, $\mathcal{V}$ outputs accept with probability one.

Knowledge soundness of polynomial commitment scheme
Polynomial commitment scheme has knowledge soundness if there exists an efficient PPT algorithm E such that for any algebraic adversary $\mathcal{A}$ the probability of $\mathcal{A}$ winning the following game is negl (λ) over the randomness of PPT $\mathcal{A}$ and Setup $(1^\lambda, d, t)$.

1. Given d, t and $(pp, SRS) =$ Setup $(1^\lambda, \mu, Q)$, $\mathcal{A}$ outputs $c \in \mathbb{G}_1$.
2. E given access to outputs of algebraic adversary $\mathcal{A}$ (including the vector $\bar{v}$ of linear combination $c = \sum_{i=1}^{s} v_i l_i$, where $l_i \in SRS$), outputs a witness polynomial $f \in \mathbb{F}[X_1, \ldots, X_\mu]$ with coefficients included in vector $\bar{v}$.

3. $\mathcal{A}$ outputs $\bar{\alpha}^{(1)}, \ldots, \bar{\alpha}^{(t)} \in \mathbb{F}^{\mu}, r \in \mathbb{F}_{<t}[X_1, \ldots, X_{\mu}]$.
4. $\mathcal{A}$ takes the part of $\mathcal{P}$ in the protocol Open with inputs $c, \bar{\alpha}^{(1)}, \ldots, \bar{\alpha}^{(t)}, r$.
5. $\mathcal{A}$ wins if
   - $\mathcal{V}$ outputs accept at the and of protocol and
   - $\langle (c, r, \bar{\alpha}^{(1)}, \ldots, \bar{\alpha}^{(t)}, d), f \rangle \notin \mathcal{R}$.

The knowledge soundness is a strong property. In the following theorem, we show that knowledge soundness implies the scheme is evaluation binding.

Theorem 3. *Let* Γ *be a binding polynomial commitment scheme - based on SRS. If* Γ *has knowledge soundness, then* Γ *is also evaluation binding.*

Proof.
To prove the theorem by contradiction, we suppose that an algebraic $\mathcal{A}$ breaks the evaluation binding by outputting $(c, \bar{\alpha}, r_0), (c, \bar{\alpha}, r_1)$, where $r_0 \neq r_1$. Since the knowledge soundness holds, there exists an efficient PPT algorithm E that extracts polynomials $f_0(\bar{X}) \neq f_1(\bar{X})$, which were committed to c in Commit part and used in Open protocol by the adversary $\mathcal{A}$. The polynomial inequality $f_0(\bar{X}) \neq f_1(\bar{X})$ follows from $f_0(\bar{\alpha}) = r_0 \neq r_1 = f(\bar{\alpha})$, two polynomial evaluated differently in same point cannot be equal. But $f_0(\bar{X}) \neq f_1(\bar{X})$ is a contradiction with the assumption that Γ is the biding commitment scheme, i.e. two different polynomials cannot be committed to the same value c. □

3. First Scheme for One Evaluation Point

In this chapter, we present a polynomial commitment scheme, where the Prover commits to some multivariate polynomial f over a finite field $\mathbb{F}$ and afterward evaluates the polynomial in a given point $\bar{\alpha} \in \mathbb{F}^{\mu}$.

In Open interactive protocol the Prover convinces the Verifier about correctness of the value $f(\bar{\alpha})$ w.r.t the commitment. This scheme verifies the correctness of the evaluated value in only one point per one Open protocol.

The scheme is designed so that the valid commitment cannot be computed to any polynomials from $\mathbb{F}[X_1, \ldots, X_{\mu}]$. One scheme parameter is an integer d, which upper bounds the total degree of the committed polynomial f. Hence, we have restrictions on the degree of SRS. In the Setup algorithm, parameter d is taken as an input determining the SRS degree (according to the previous notation $Q = d$). We need to consider a Q-DLOG assumption (Definition 16) for the groups $\mathbb{G}_1, \mathbb{G}_2$, where the parameter $Q = d$. The formal description of our first scheme is given in the next section.

3.1 The scheme for one evaluating point

1. Setup $(1^{\lambda}, \mu, d) = (\mathsf{pp}, \mathbf{srs_1}, \mathbf{srs_2})$,
   where $\mathsf{pp} = (\mathbb{F}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, e)$ and $\mathbf{srs_1}, \mathbf{srs_2}$ are defined as follows:

$$\mathbf{srs_1}, \mathbf{srs_2} = \left\{ [x_1^{d_1} x_2^{d_2} \cdots x_{\mu}^{d\mu}]_1, [1]_2, [x_1]_2, \ldots, [x_{\mu}]_2,\ d_i \geq 0, \sum_{i=1}^{\mu} d_i < d \right\},$$

   where $x_1, x_2, \ldots, x_{\mu} \in \mathbb{F}$ are chosen uniformly at random.

2. Commit $(f, \mathbf{srs_1}, \mathbf{srs_2}) = [f(x_1, \ldots, x_{\mu})]_1$

3. Open $(\mathsf{pp}, \mathbf{srs_1}, \mathbf{srs_2},\ d,\ c,\ \bar{\alpha} = (\alpha_1, \ldots, \alpha_{\mu}) \in \mathbb{F}^{\mu},\ r \in \mathbb{F})$

   a) $\mathcal{P}$ computes polynomials $h_1, h_2, \ldots, h_{\mu} \in \mathbb{F}[X_1, \ldots, X_{\mu}]$ s.t.

$$f(X_1, \ldots, X_{\mu}) - r = \sum_{i=1}^{\mu} h_i(X_1, \ldots, X_{\mu})(X_i - \alpha_i).$$

   Using $\mathbf{srs_1}$, $\mathcal{P}$ computes and sends to $\mathcal{V}$ group elements $[h_1(x_1, \ldots, x_{\mu})]_1, \ldots, [h_{\mu}(x_1, \ldots, x_{\mu})]_1$.

   b) Using $\mathbf{srs_2}$, $\mathcal{V}$ computes $[(x_i - \alpha_i)]_2$
   $\mathcal{V}$ return accept if and only if

$$(c - [r]_1) \odot [1]_2 = \prod_{i=1}^{\mu} \Big([h_i(x_1, \ldots, x_{\mu})]_1 \odot [(x_i - \alpha_i)]_2 \Big). \tag{3.1}$$

3.2 Getting polynomials h_i

In the Open part of the protocol, $\mathcal{P}$ needs to compute polynomials $h_1, \ldots, h_\mu$ s.t.

$$f(X_1, \ldots, X_\mu) - r = \sum_{i=1}^{\mu} h_i(X_1, \ldots, X_\mu)(X_i - \alpha_i).$$

In this case, the computation of polynomials h_i is more straightforward than described in Section 1.1.2. According to Section 1.1.2, $\mathcal{P}$ has to compute Gröbner basis of an ideal I given by an evaluation point $\bar{\alpha}$. However, we can simplify the procedure by noticing that Gröbner basis of the ideal I is already known. The established ideal is $I = (X_1, -\alpha_1, \ldots, X_\mu - \alpha_\mu)$. The following lemma states that $(X_1 - \alpha_1, \ldots, X_\mu - \alpha_\mu)$ is a Gröbner basis indeed.

Lemma 4. *Let $G = (g_1, \ldots, g_\mu |\ g_i = X_i - \alpha_i, \alpha_i \in \mathbb{F})$, Let $I = \langle G \rangle$ be an ideal in $\mathbb{F}[X_1, \ldots, X_\mu]$. Then G is a Gröbner basis of the ideal I.*

Proof. We can use the Definition 6 of Gröbner basis. Let $f \in I, f \neq 0$ be a non-constant polynomial. Since $lm(g_1) = X_1, \ldots, \ lm(g_\mu) = X_\mu$, there certainly exists $g_i \in G$ s.t. $lm(g_i) | lm(f)$. Thus, polynomials in G are already a Gröbner basis of the ideal I. □

Therefore, $\mathcal{P}$ simply uses Algorithm 1 (*Multivariate Division Algorithm*) with Gröbner basis G as the input, and gets the required polynomials $h_1, \ldots, h_\mu$.

3.3 Completeness of the first protocol

Consider the prescribed $\mathcal{P}$. Then $f(\bar{\alpha}) = r$, c is a correct commitment to $f(X_1, \ldots, X_\mu)$ and $\mathcal{P}$ follows the Open protocol correctly with these values.

By Hilbert's weak Nullstellensatz (Theorem 1), the first assumption, $f(\bar{\alpha}) = r$ implies $f(X_1, \ldots, X_\mu) - r \in I$, where $I = \langle X_1 - \alpha_1, \ldots, X_\mu - \alpha_\mu \rangle$ and the polynomial $f - r$ could be generated by a basis of ideal I. Furthermore, by Lemma 4, $\langle X_1 - \alpha_1, \ldots, X_\mu - \alpha_\mu \rangle$ is Gröbner basis of the ideal I.

Therefore, according to lemma 4, $\mathcal{P}$ can compute $h_1, \ldots h_\mu \in \mathbb{F}[X_1, \ldots, X_\mu]$, s.t. $f - r = \sum_{i=1}^{\mu} h_i(X_i - \alpha_i)$ directly using Algorithm 1 (*Multivariate Division Algorithm*). We write the following observation:

$$f - r = \sum_{i=1}^{\mu} h_i(X_i - z_i) \Rightarrow f(x_1, \ldots, x_\mu) - r = \sum_{i=1}^{\mu} h_i(x_1, \ldots, x_\mu)(x_i - \alpha_i), \quad (3.2)$$

where $x_1, \ldots, x_\mu$ are elements used in $\mathbf{srs_1}, \mathbf{srs_2}$.

In the next step, $\mathcal{P}$ sends Commit $(h_i, \mathbf{srs_1}, \mathbf{srs_2})$ to $\mathcal{V}$. Since c is a correct commitment, $c = [f(x_1, \ldots, x_\mu)]_1$. At the end of the protocol, $\mathcal{V}$ checks if

$$(c - [r]_1) \odot [1]_2 = \prod_{i=1}^{\mu} \Big([h_i(x_1, \ldots, x_\mu)]_1 \odot [(x_i - \alpha_i)]_2 \Big).$$

Since $\mathcal{P}$ passes the protocol correctly:

$$
\begin{aligned}
(c - [r]_1) \odot [1]_2 &= ([f(x_1, \ldots, x_\mu)]_1 - [r]_1) \odot [1]_2 \\
&= (f(x_1, \ldots, x_\mu) \cdot g_1 - r \cdot g_1) \odot 1 \cdot g_2 \\
&= ((f(x_1, \ldots, x_\mu) - r) \cdot g_1) \odot g_2 \\
&= g_t^{f(x_1,\ldots,x_\mu)-r} \\
&= g_t^{\sum_{i=1}^{\mu} h_i(x_1,\ldots,x_\mu)(x_i-\alpha_i)} \\
&= \prod_{i=1}^{\mu} \Big([h_i(x_1, \ldots, x_\mu)] \cdot g_1 \odot [(x_i - \alpha_i)] \cdot g_2 \Big),
\end{aligned}
$$

where the fourth and sixth equalities follow from the bilinearity of the pairing. The fifth equality follows from (3.2). To conclude, $\mathcal{V}$ outputs accept with probability one.

3.4 Knowledge soundness of the first protocol

We show that there exists an extractor E as needed to satisfy Definition 13. We suppose Setup generates two additive groups $\mathbb{G}_1, \mathbb{G}_2$ satisfying Q-DLOG assumption (Definition 16) with parameter $Q = d$. Since $\mathcal{A}$ is algebraic adversary, whenever $\mathcal{A}$ outputs $c \in \mathbb{G}_1$, E given access to linear combination $c = \sum_{i=1}^{s} v_i \cdot l_i$, where $\bar{v} = v_1, \ldots, v_s$ is coefficient vector provided by algebraic $\mathcal{A}$ and $l_1, \ldots, l_s$ is list of all elements in $\mathbf{srs_1}$.

E outputs an extracted witness polynomial

$$
f = \sum_{i=0}^{s} v_i X^i.
$$

In the next step, $\mathcal{A}$ determines $\bar{\alpha} \in \mathbb{F}^\mu, r \in \mathbb{F}$ s.t. $f(\bar{\alpha}) \neq r$. We show that with these values $\mathcal{V}$ outputs accept with only negl (λ) probability.

For simplicity, we shorten notation $H(X_1, \ldots, X_\mu)$ to $H(\bar{X})$ and $H(x_1, \ldots, x_\mu)$ to $H(\bar{x})$ for any following polynomials.

Now $\mathcal{A}$ outputs some $[H_1(\bar{x})]_1, \ldots [H_\mu(\bar{x})]_1$, $H_i \in \mathbb{F}_{<d}[\bar{X}]$. However, according to Hilbert's weak Nullstellensatz (Theorem 1), the following holds:

$$
\begin{aligned}
f(\bar{\alpha}) - r \neq 0 \Leftrightarrow f(\bar{X}) - r &\notin \langle (X_1 - z_1) \cdots (X_\mu - z_\mu) \rangle \\
&\Leftrightarrow \nexists\ H_i(\bar{X}) \text{ s.t } f(\bar{X}) - r = \sum_{i=1}^{\mu} H_i(\bar{X})(X_i - z_i).
\end{aligned} \tag{3.3}
$$

$\mathcal{V}$ verifies validity of real pairing check at point $\bar{x}$, i.e,

$$
([f(\bar{x})]_1 - [r]_1) \odot [1]_2 = \prod_{i=1}^{\mu} ([H_i(\bar{x})]_1 \odot [x_i - \alpha_i]_2).
$$

Under the Q-DLOG assumption for groups $\mathbb{G}_1, \mathbb{G}_2$, we can use the proposition of probability bounding of real and ideal checks.

By Proposition 2, it suffices to upper bound the probability of the corresponding ideal check:

$$f(\bar{X}) - r = \sum_{i=1}^{\mu} H_i(\bar{X})(X_i - \alpha_i).$$

According to eq. (3.3), polynomials H_i passing the ideal check do not exist. Using Proposition 2, the real check passes with probability $\mathsf{negl}\,(\lambda)$.

3.5 Summary

Theorem 5. *Assuming* Setup *generates additive groups* $\mathbb{G}_1, \mathbb{G}_2$ *for which the Q-DLOG assumption holds. Then the scheme defined in section 3.1 is a Polynomial commitment scheme - based on SRS, where* Open *part is an interactive argument of knowledge with respect to the relation* $\mathcal{R}$*, defined in 2.1.*

Proof. The scheme includes all three parts Setup, Commit, Open from definition Polynomial commitment scheme (Definition 19). For Open to be an interactive argument of knowledge, we have to prove completeness (Definition 11) and knowledge soundness (Definition 13). The completeness is proved in Section 3.3. The knowledge soundness under the Q-DLOG assumption for $\mathbb{G}_1, \mathbb{G}_2$ is proved in Section 3.4. □

3.5.1 Binding

In this section we show that our suggested scheme from Section 3.1 satisfies *binding* and *evaluation binding* properties. An algebraic adversary cannot commit to two different polynomials with the same commitment (*binding*). An algebraic adversary cannot convince the Verifier that the committed polynomial is evaluated in the same point to different values (*evaluation binding*).

Theorem 6.
Let Setup *from polynomial commitment scheme defined in section 3.1 output* $\mathbb{G}_1, \mathbb{G}_2$ *satisfying Q-DLOG assumption. Then the scheme is binding (Definition 9).*

Proof. To prove the theorem by contradiction, we suppose that the scheme is not binding. Hence, an algebraic adversary $\mathcal{A}$ convinces the Verifier in Open protocol with values $(\mathsf{pp}, \mathbf{srs_1}, \mathbf{srs_2}, c, f_0(\bar{X}))$ and $(\mathsf{pp}, \mathbf{srs_1}, \mathbf{srs_2}, c, f_1(\bar{X}))$, where $f_0(\bar{X}) \neq f_1(\bar{X})$.

The commitment is equal for both polynomials, so we have $[f_0(\bar{x})]_1 = [f_1(\bar{x})]_1$. We write the last equality in more detail.

$$\begin{aligned}
[f_0(x_1, \dots x_\mu)]_1 = [f_1(x_1, \dots x_\mu)]_1 \Rightarrow \\
f_0(x_1, \dots x_\mu) \cdot g_1 = f_1(x_1, \dots x_\mu) \cdot g_1 \Rightarrow \\
(f_0(x_1, \dots x_\mu) - f_1(x_1, \dots x_\mu)) \cdot g_1 = 0.
\end{aligned}$$

$g_1 \neq 0$, since g_1 is a generator of group $\mathbb{G}_1$ which has order q. Hence, we have

$$\begin{aligned}
f_0(x_1, \dots x_\mu) - f_1(x_1, \dots x_\mu) = 0 \text{ or} \\
f_0(x_1, \dots x_\mu) - f_1(x_1, \dots x_\mu) = k \cdot q, \text{ for } k \in \mathbb{N}.
\end{aligned}$$

It implies that polynomial $f_0(\bar{X}) - f_1(\bar{X})$ or $f_0(X_1, \dots X_\mu) - f_1(X_1, \dots X_\mu) - k \cdot q$ has a non trivial root. Both of these polynomials have the total degree at most Q, so $\mathcal{A}$ can find nontrivial factor $x_1, \dots, x_\mu$ which is in contradiction with the Q-DLOG assumption. □

Theorem 7. *Assuming* Setup *generates additive groups* $\mathbb{G}_1, \mathbb{G}_2$ *for which Q-DLOG assumption holds. The polynomial commitment scheme defined in the section 3.1 is evaluation binding (definition 20).*

Proof. The knowledge soundness property of Open protocol is shown in the section 3.4. Then we can use Theorem 3, which states that if a polynomial commitment scheme is binding (Theorem 6) and has knowledge soundness, then the polynomial commitment scheme is also evaluation binding. □

3.6 Scheme complexity

First, we introduce a standard auxiliary technical proposition.

Proposition 4. *Let* $f \in \mathbb{F}[X_1, \dots, X_\mu]$ *be a* μ*-variate polynomial of total degree* $d \in \mathbb{N}$*. The maximum number of terms appearing in polynomial* f *is* $\binom{\mu+d}{d}$.

Proof. We need to compute the sum of the total number of terms of degrees exactly $0, \dots, d$. We look at this task as a combinatorial problem. Let $i \in \{0, \dots, d\}$ be the current degree. We must divide i powers among exponents to μ parts. This we can express as the combinatorial number $\binom{i+\mu-1}{\mu-1}$. Since we need all the terms at degree $0, \dots, d$, we obtain the following sum:

$$\sum_{i=0}^{d} \binom{i+\mu-1}{\mu-1} = \frac{(d+1)\binom{\mu+d}{\mu-1}}{\mu} = \frac{(d+1)\frac{(\mu+d)!}{(\mu-1)!(d+1)!}}{\mu}$$
$$= \frac{(d+1)(\mu+d)!}{(\mu-1)!\mu(d+1)d!} = \frac{(\mu+d)!}{\mu!d!} = \binom{\mu+d}{\mu}.$$ □

Theorem 8. *The polynomial commitment scheme defined in the Section 3.1 has the following properties:*

i) $\mathbf{srs_1}, \mathbf{srs_2}$ *consist of* $\binom{\mu+d}{\mu}$ *elements of* $\mathbb{G}_1$ *and* $\mu + 1$ *elements of* $\mathbb{G}_2$.

ii) For integer $n \leq d$ *and* $f \in \mathbb{F}_{<n}[X_1, \dots, X_\mu]$*, computing* Commit $(f, \mathbf{srs_1}, \mathbf{srs_2})$ *requires* $\binom{\mu+n}{\mu}$ *multiplications in* $\mathbb{G}_1$.

iii) Prover $\mathcal{P}$ *sends* μ *elements of* $\mathbb{G}_1$.

iv) Verifier $\mathcal{V}$ *computes one multiplication in* $\mathbb{G}_1$*, one addition in* $\mathbb{G}_1$*,* μ *multiplications in* $\mathbb{G}_2$*,* $\mu + 1$ *pairings and* μ *multiplications in* $\mathbb{G}_t$.

Proof.

i) According to Proposition 4, the number of terms with total degree $0, \ldots, d$ is $\binom{\mu+d}{\mu}$. To create $\mathbf{srs_1}$ we need to multiply each of the terms by g_1.

Hence, to compute $\mathbf{srs_1}$, it is required to do $\binom{\mu+d}{\mu}$ multiplication in $\mathbb{G}_1$.

Since $\mathbf{srs_2}$ consists of $[1]_2, [x_1]_2, \ldots, [x_\mu]_2$, it is simply required to do $\mu + 1$ multiplications in $\mathbb{G}_2$.

ii) According to Proposition 4, any polynomial f contains at most $\binom{\mu+n}{\mu}$ terms. For each term exists exactly one corresponding $\mathbb{G}_1$ element in precomputed $\mathbf{srs_1}$. For example for term $3X_1X_2$ corresponding $[x_1x_2]_1$ and it remains to compute $3 \cdot [x_1x_2]_1$. The computation of $\mathsf{Commit}\ (f, \mathbf{srs_1}, \mathbf{srs_2})$ requires $\binom{\mu+n}{\mu}$ multiplication in $\mathbb{G}_1$.

iii) Prover $\mathcal{P}$ has to send $[h_1(x_1, \ldots, x_\mu)]_1, \ldots, [h_\mu(x_1, \ldots, x_\mu)]_1$ i.e. μ elements of $\mathbb{G}_1$

iv) Verifier $\mathcal{V}$ computes μ multiplications in $\mathbb{G}_2$ for computing $[(x_i - z_i)]_2$ and, since r is a single element in $\mathbb{F}$, one multiplication in $\mathbb{G}_1$ for $[r]_1$. $\mathcal{V}$ computes one addition in $\mathbb{G}_1$, $\mu + 1$ pairings and μ multiplications in $\mathbb{G}_t$ to verify eq. (3.1) at the end of the protocol. □

4. Second Scheme for Multiple Evaluation Points

In this chapter, we present a polynomial commitment scheme, where Open protocol verifies the correctness of a polynomial evaluation in multiple points.

The scheme requires more input parameters than the previous scheme from the section 3.1. The restriction of a total degree of a committed polynomial is the same parameter d. Additionally, the scheme depends on a parameter t, the number of evaluating points.

In our scheme, we use the following notation: $D = \mu^t - 1$. In Setup, we have more specific restrictions to the particular degree of monomials appearing in SRS, including both parameters d, D. The limitation of SRS total degree is $d + D$. Thus, we have to consider a Q-DLOG assumption (Definition 16) for the groups $\mathbb{G}_1, \mathbb{G}_2$, where the parameter $Q = d + D$.

4.1 The scheme for multiple evaluating points

1. Setup $(1^\lambda,\ \mu,\ d,\ D) \coloneqq (\mathsf{pp}, \mathbf{srs_1}, \mathbf{srs_2})$,
   where $\mathsf{pp} = (\mathbb{F}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, g_1, g_2, g_t, e)$ and $\mathbf{srs_1}, \mathbf{srs_2}$ are defined as follows:

$$\mathbf{srs_1}, \mathbf{srs_2} = \Big\{ [x_1^{d_1} \cdots x_\mu^{d_\mu}]_1,\ d_i \geq 0, \sum_{i=1}^{\mu} d_i < d,$$
$$[x_1^{d_1} \cdots x_\mu^{d_\mu} \cdot w^{d_w}]_1,\ d_i \geq 0,\ d_w < D, \sum_{i=1}^{\mu} d_i < d,$$
$$[1]_2, \ldots, [x_\mu]_2, [w]_2. \Big\}$$

2. Commit $(f, \mathbf{srs_1}, \mathbf{srs_2}) \coloneqq [f(x_1, \ldots, x_\mu)]_1$ (identical to the first scheme)

3. Open $(\mathsf{pp}, \mathbf{srs_1}, \mathbf{srs_2}, D, d, c, \bar{\alpha}^{(1)}, \ldots, \bar{\alpha}^{(t)} \in \mathbb{F}^\mu, r \in \mathbb{F}_{<t}[X_1, \ldots, X_\mu])$

   a) Both $\mathcal{P}$ and $\mathcal{V}$ compute a Gröbner basis of an ideal I and combine the basis polynomials into one polynomial $B \in \mathbb{F}[X_1, \ldots, X_\mu, W]$:

      i)
      $$I \coloneqq \langle \tilde{b}_1, \ldots, \tilde{b}_{\mu^t} \rangle, \text{ where } \tilde{b}_k \coloneqq \prod_{i=1}^{t} \left(X_{\sigma_k(i)} - \alpha^{(i)}_{\sigma_k(i)} \right).$$

      ii) $(b_1, \ldots, b_{\mu^t}) \leftarrow$ (Buchberger's algorithm on $(\tilde{b}_1, \ldots, \tilde{b}_{\mu^t})$)

      iii)
      $$B \coloneqq \sum_{i=0}^{D} b_i W^i.$$

   b) $\mathcal{P}$ computes polynomials h_i satisfying:

$$f(\bar{X}) - r(\bar{X}) = \sum_{i=0}^{D} h_i(\bar{X}) b_i(\bar{X}),$$

and combines them to one polynomial $H \in \mathbb{F}[X_1, \ldots, X_\mu, W]$:

i) $h_1, \ldots, h_{\mu^t} \leftarrow$(Multivariate Division Algorithm on $(f - r, b_1, \ldots, b_{\mu^t})$)

ii)
$$H := \sum_{i=0}^{D} h_i(\bar{X}) W^{D-i}$$

c) $\mathcal{P}$ computes $H_L,\ H_R$, satisfying:

$$H(\bar{X}, W) = H_L(\bar{X}, W) + H_R(\bar{X}, W) W^{D/2},$$

where $H_L,\ H_R$ are polynomials with variable W in exponent $\leq D/2$

d) $\mathcal{P}$ computes f_L, f_R satisfying:

$$(H_L(\bar{X}, W) + H_R(\bar{X}, W) W^{D/2}) \cdot B(\bar{X}, W) = f_L(\bar{X}, W) + f_M(\bar{X}, W) + f_R(\bar{X}, W) W^{D+1},$$

where f_L, f_R are polynomials with variable W in exponent $< D$, f_M is polynomial with variable W in exponent $= D$

e) $\mathcal{P}$ sends commitments to H_L, H_R, f_L, f_R to $\mathcal{V}$:

$$\mathcal{P} \xrightarrow{\mathsf{Commit}(H_L, \mathbf{srs_1}, \mathbf{srs_2}), \mathsf{Commit}(H_R, \mathbf{srs_1}, \mathbf{srs_2})} \mathcal{V}$$

$$\mathcal{P} \xrightarrow{\mathsf{Commit}(f_L, \mathbf{srs_1}, \mathbf{srs_2}), \mathsf{Commit}(f_R, \mathbf{srs_1}, \mathbf{srs_2})} \mathcal{V}$$

f) $\mathcal{V}$ sends randomly chosen $\bar{\gamma} \in \mathbb{F}^{\mu+1}$ to $\mathcal{P}$

$$\mathcal{P} \xleftarrow{\bar{\gamma} = (\gamma_1, \ldots, \gamma_\mu, \omega)} \mathcal{V}$$

g) $\mathcal{P}$ sends to $\mathcal{V}$ values $H_L(\bar{\gamma}), H_R(\bar{\gamma}), f_L(\bar{\gamma}), f_R(\bar{\gamma}), f(\bar{\gamma})$.

h) $\mathcal{P}$ and $\mathcal{V}$ run the Open of the protocol from section 3.1 with one evaluating point $\bar{\gamma}$, where Open runs separately for each H_L, H_R, f_L, f_R with commitments from part e), for f with commitment from Commit part.

i) $\mathcal{V}$ verifies:

$$(H_L(\bar{\gamma}) + H_R(\bar{\gamma}) \omega^{D/2}) B(\bar{\gamma}) = f_L(\bar{\gamma}) + (f(\bar{\gamma}) - r(\bar{\gamma})) \omega^D + f_r(\bar{\gamma}) \omega^{D+1}. \quad (4.1)$$

j) $\mathcal{V}$ outputs accept only if equation 4.1 is correct and in step h) outputs accept in all Open subprotocols.

4.2 About the protocol

The difference is to cope with more evaluating points. Hence, the computation of the ideal I is more complicated.

By to Section 1.2, the number of polynomials in the basis of the ideal I is μ^t. In this case, $\mathcal{P}$ would have to send all $h_1, \ldots, h_{\mu^t}$. In the protocol, we avoid this inconvenience by combining basis polynomials to one polynomial B by adding a new variable W:

$$B = \sum_{i=0}^{D} b_i W^i.$$

Vice versa the decomposition of the polynomials b_i is uniquely determined by B. Similarly to the first protocol, $\mathcal{P}$ still has to compute polynomials h_i s.t.

$$f(\bar{X}) - r(\bar{X}) = \sum_{i=0}^{D} h_i(\bar{X}) b_i(\bar{X}).$$

Polynomials h_i are also combined to one polynomial H with one extra variable in the following way:

$$H = \sum_{i=0}^{D} h_i W^{D-i}.$$

The polynomial H orders the elements in the opposite way than polynomial B. This allows us to use the form of product $B \cdot H$. Additionally, $\mathcal{P}$ divides $H(\bar{X})$ to two parts, according to the degree in variable W. This detail is more precise discussed next.

$$\begin{aligned}(H_L(\bar{X}, W) + H_R(\bar{X}, W) W^{D/2}) \cdot B(\bar{X}, W) = \\ f_L(\bar{X}, W) + (f - r)(\bar{X}, W) W^D + f_R(\bar{X}, W) W^{D+1}\end{aligned} \tag{4.2}$$

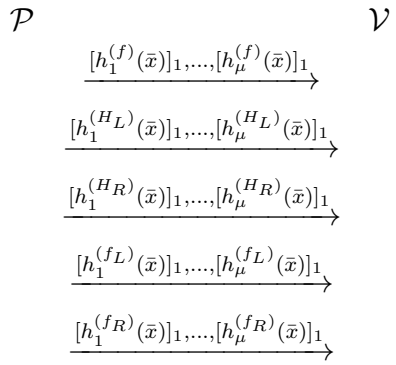
The term with variable W to the power D exponent is exactly the polynomial $(f - r)(\bar{X})$.

$$f(\bar{X}) - r(\bar{X}) = \sum_{i=0}^{D} b_i h_i.$$

Hence, $\mathcal{P}$ convinces $\mathcal{V}$ about the knowledge and correctness of the above polynomials appearing in eq. (4.2). However, while avoiding to send the whole polynomials. For this purpose, we use the first protocol from section 3.1, $\mathcal{P}$ commits to each polynomial and convinces $\mathcal{V}$ about the correctness of values $H_L(\bar{\gamma}), H_R(\bar{\gamma}), f_L(\bar{\gamma}), f_R(\bar{\gamma}), f(\bar{\gamma})$.

$\mathcal{P}$ computes commitments to H_L, H_R, f_L, f_R. The commitment to f is already known. Then, $\mathcal{V}$ sends a random value $\bar{\gamma} \in \mathbb{F}^{\mu+1}$ and obtains a Prover's answer with $H_L(\bar{\gamma}), H_R(\bar{\gamma}), f_L(\bar{\gamma}), f_R(\bar{\gamma}), f(\bar{\gamma})$.

By following Open part form the first protocol, $\mathcal{P}$ has to compute and sends $[h_1(\bar{X})]_1, \ldots, [h_{\mu+1}(\bar{X})]_1$ for all committed polynomials (in total $5*(\mu+1)$ elements are sent by the Prover).



We emphasize that only the Open part from the previous scheme is used. The values $[h_i(\bar{x})]_1$ are computed by $\mathbf{srs_1}$ from the scheme for multiple evaluation points (Section 4.1).

At the end of each Open subprotocol, $\mathcal{V}$ returns accept or reject.

Finally, Verifier evaluates the Equation (4.2) in challenge point $\bar{\gamma}$ to be convinced about the correctness of the equation.

4.2.1 SRS structure

In this section, we explain the correctness of SRS generated in the Setup.

Using $\mathbf{srs_1}$, $\mathcal{P}$ computes commitments to $f(\bar{X})$, $H_L(\bar{X}, W), H_R(\bar{X}, W)$, $f_L(\bar{X}, W), f_R(\bar{X}, W)$.

At first, a total degree of $f(\bar{X})$ is upper bounded by d from the protocol assumption. Hence, for creating the commitment to polynomial f, we need:

$$\mathbf{srs_1} = [x_1^{d_1} \cdots x_\mu^{\mu}]_1, d_i \geq 0, \sum_{i=1}^{\mu} d_i < d.$$

Next, we estimate the total degree of polynomials $H_L(\bar{X}, W), H_R(\bar{X}, W)$, $f_L(\bar{X}, W), f_R(\bar{X}, W)$.

We can estimate the total degree of polynomial $r(\bar{X})$, as r is an interpolation polynomial for each variable separately. Since the interpolation is computed for t evaluation points, the polynomial has at most degree t in each variable. Overall, we can upper bound the total degree of polynomial r by $\mu \cdot t$.

The difference of two polynomials preserves the total degree. Polynomial $f(\bar{X})$ has total degree d, polynomial r has total degree $\mu \cdot t$. Hence, $(f - r)(\bar{X})$ has total degree $\max\{d, \mu \cdot t\}$.

Using the *Multivariate Division Algorithm*(Algorithm 1), $\mathcal{P}$ computes polynomials $h_i(\bar{X})$. From the algorithm we have the following estimate:

$$lm(f - r) = \max\{lm(b_i)lm(h_i)|\ i = 1, \ldots, \mu^t\}.$$

Thus the total degree of $h_1, \ldots, h_{\mu^t}$ is also $\max\{d, \mu \cdot t\}$.

Polynomials H_L, H_R preserve the total degree in variables $X_1, \ldots, X_\mu$. Polynomials f_L, f_R are part of the decomposition of $H \cdot B$ consisting of terms $b_i h_j$, for some $i, j \in \{1, \ldots, \mu^t\}$. We can use the estimate as:

$$lm(f - r) = \max\{lm(b_i)lm(h_i)|\ i = \{1, \ldots, \mu^t\}.$$

Thus, each term appearing in $H \cdot B$ has a total degree bounded by $\max\{d, \mu \cdot t\}$, so f_L, f_R have the same limitation.

Additionally, we have to describe exponents in variable W. As the next section explains, we have to upper bound SRS exponents in variable W by $D = \mu^t - 1$.

In conclusion, to compute commitments to H_L, H_R, f_L, f_R we need to have the following form of $\mathbf{srs_1}$:

$$\mathbf{srs_1} = \{[x_1^{d_1} \cdots x_\mu^{d\mu} \cdot w^{d_w}]_1,\ d_i \geq 0,\ d_w < D, \sum_{i=1}^{\mu} d_i < \max\{d, \mu \cdot t\}\}$$

In step h), both parties run the Open part of protocol from Section 3.1, where $\mathcal{P}$ has to compute $[h_i^{(H_L)}]_1$, $[h_i^{(H_R)}]_1$, $[h_i^{(f_L)}]_1$, $[h_i^{(f_R)}]_1$, for $i = \{1, \ldots, \mu\}$. From the structure of the protocol follows that the above defined $\mathbf{srs_1}$ suffices to compute these values.

Finally, $\mathcal{V}$ has to use $\mathbf{srs_2}$ for computing $(x_1 - \gamma_1), \ldots, (x_\mu - \gamma_\mu), (w - \omega)$. Hence, $\mathbf{srs_2}$ consist of

$$\mathbf{srs_2} = \{[1]_2, [x_1]_2, \ldots, [x_\mu]_2, [w]_2\}.$$

4.2.2 Omitting the limitation of degree in variable W

This section explains why we require the limitation of the degree in element w given in the SRS. The discussed part of SRS is

$$\mathbf{srs_1} = \Big\{[x_1^{d_1} \cdots x_\mu^{d\mu} \cdot w^{d_w}]_1,\ d_i \geq 0,\ d_w < D, \sum_{i=1}^{\mu} d_i < d\Big\}.$$

SRS is constructed so that only polynomials including terms with variable W in degree at most $D - 1$ can be committed correctly.

The necessity of this requirement is shown in the following example.

Example. Assume the protocol with the "unlimited" SRS, i.e., it is possible to compute a commitment to every polynomial, even including monomials where W has a degree larger than D.

Let $f = X_1^2 - X_1 + 3$ be the polynomial which does not satisfy $f(\bar{\alpha}) = r$. Following parametres are established before the Open part of the protocol or could be computed by both parties ($\mathcal{P}$ cannot change these values during Open protocol): $D = 2$, $r = 5$, $B = (b_1, \ldots, b_D)$, the Gröbner basis of the ideal I. Nevertheless, $\mathcal{P}$ can choose any polynomial H and compute $H \cdot B$. For example, $\mathcal{P}$ chooses:

$$H \cdot B = (X_1 + X_2) + (X_1^3)W + (X_2^2)W^2 + (X_1X_2)W^3 + (X_1X_2^2)W^4.$$

If $\mathcal{P}$ participates honestly in the rest of protocol, $\mathcal{P}$ sends commit-ments to $f_L = (X_1 + X_2) + (X_1^3)$ and $f_R = (X_1 X_2) + (X_1 X_2^2)W$. Then $\mathcal{V}$ outputs reject at the end of the protocol, because the equation 4.1 would not be correct.

However, if we allow to commit to a term including W with a degree larger than D, $\mathcal{P}$ can change the middle term according to committed polynomial f.

We have $f - r = X_1^2 - X_1 - 2$. Then $\mathcal{P}$ can compute polynomials $f_L = (X_1 + X_2) + (X_1^3)W + (X_2^2 - X_1^2 + X_1 + 2)W^2, f_R = (X_1 X_2) + (X_1 X_2^2)W$ and send their commitments. The left part of the equation stays the same, only the decomposition to f_L, f_R has been changed:

$$H \cdot B = f_L + X_1^2 - X_1 - 2 + f_R W^3.$$

If $\mathcal{P}$ follows the rest of the protocol correctly, these commitments are correct, and $\mathcal{V}$ output accept after each Open subprotocols.
The equation

$$H(\bar{\gamma})B(\bar{\gamma}) = f_L(\bar{\gamma}) + (f(\bar{\gamma}) - r(\bar{\gamma}))w^D + f_r(\bar{\gamma})w^{D+1}$$

is also correct, because polynomials f_L, f_R have been modified in the way that both sides of the equation are equal. In conclusion, $\mathcal{P}$ can pass the protocol with any polynomial f.

In particular, the SRS limitation of the degree in variable W, makes the decomposition of $H \cdot B$ to $f_L, f_M = (f - r), f_R$ unique.

4.3 Completeness

Suppose the prescribed $\mathcal{P}$ proceeds correctly during the Open protocol.
The Prover's honesty implies the following:

i) commitment is correct, i.e, Commit $(f, \mathbf{srs_1}, \mathbf{srs_2}) = [f(x_1, \dots x_\mu)]_1$

ii) $\forall i \in [t] : f(\bar{\alpha}^{(i)}) = r(\bar{\alpha}^{(i)})$.

By Hilbert's weak Nullstellensatz (Theorem 1) and eq. (1.1) the second assumption implies that

$$\forall i \in [t] : f(X_1, \dots X_\mu) - r(X_1, \dots X_\mu) \in I_{\bar{\alpha}^{(i)}} = \langle (X_1 - \alpha_1^{(i)}), \dots, (X_\mu - \alpha_\mu^{(i)}) \rangle$$
$$\Rightarrow f(X_1, \dots X_\mu) - r(X_1, \dots X_\mu) \in I = \bigcap_{i \in [t]} I_{\bar{\alpha}^{(i)}}.$$

It implies that there exists the decomposition

$$f(X_1, \dots X_\mu) - r(X_1, \dots X_\mu) = \sum_{i=1}^{\mu^t} h_i g_i,$$

where g_i are basis polynomials of ideal I and h_i are polynomials in $\mathbb{F}[X_1, \dots, X_\mu]$.

To achieve this composition, in the section Open a) Prover $\mathcal{P}$ computes the Gröbner basis of ideal I. First, the Prover computes

$$I \coloneqq \langle \tilde{b}_1, \dots, \tilde{b}_{\mu^t} \rangle, \text{ where } \tilde{b}_k \coloneqq \prod_{i=1}^{t} \left(X_{\sigma_k(i)} - \alpha_{\sigma_k(i)}^{(i)} \right).$$

Using Buchberger's algorithm, $\mathcal{P}$ computes polynomials $b_1, \ldots, b_{\mu^t}$ which is Gröbner basis of the ideal I.

In the section Open b), Prover $\mathcal{P}$ uses the *Multivariate Division Algorithm* (Algorithm 1) with Gröbner basis $b_1, \ldots, b_{\mu^t}$ as the algorithm input and gets the polynomials $h_1, \ldots, h_{\mu^t}$ as the result. By Lemma 2, $\sum_{i=1}^{\mu^t} h_i b_i$ is the required decomposition of polynomial $f - r$.

Prover $\mathcal{P}$ combines both sequences of polynomials to

$$B = \sum_{i=0}^{D} b_i W^i, \; H = \sum_{i=0}^{D} h_i W^{D-i}$$

and computes the product: $H(X_1, \ldots, X_\mu, W) \cdot B(X_1, \ldots, X_\mu, W)$. Since we have the following expression of the terms in the product $H \cdot B$

$$\begin{aligned} i \leq D &: \sum_{j=0}^{i} b_j h_{D+j-i} W^i \\ i \geq D &: \sum_{j=0}^{2D-i} h_j b_{D+j-i} W^i \\ \Rightarrow i = D &: \sum_{j=0}^{D} b_j h_j W^D = f(X_1 \ldots, X_\mu) - r(X_1 \ldots, X_\mu) W^D, \end{aligned}$$

we can divide the product into three parts

$$H \cdot B = f_L + (f - r) W^D + f_R \cdot W^{D+1}, \tag{4.3}$$

where $f_L, \; f_R \in \mathbb{F}[X_1, \ldots, X_\mu, W]$ contain terms of degree at most $D - 1$ in variable W. This allows the Prover to compute Commit $(f_L, \mathbf{srs_1}, \mathbf{srs_2})$, Commit $(f_R, \mathbf{srs_1}, \mathbf{srs_2})$.

After dividing $H(\bar{X}, W)$ to part $H_L(\bar{X}, W), H_R(\bar{X}, W)$ via

$$H(\bar{X}, W) = H_L(\bar{X}, W) + H_R(\bar{X}, W) W^{D/2}, \tag{4.4}$$

both H_L, H_R are polynomials of degree at most $D - 1$ in variable W. Hence, $\mathcal{P}$ can compute Commit $(H_L, \mathbf{srs_1}, \mathbf{srs_2})$, Commit $(H_R, \mathbf{srs_1}, \mathbf{srs_2})$.

Prover follows Open from the commitment scheme (section 3.1) with one evaluation point $\bar{\gamma} = (\gamma_1, \ldots, \gamma_\mu, \omega) \in \mathbb{F}^{\mu+1}$ and polynomials f_L, f_R, H_L, H_R, f.

Since Open protocol for one evaluation point has completeness (proved in section 3.3), Verifier outputs accept at the end of each Open protocol.

The correctness of the equation

$$(H_L(\bar{\gamma}) + H_R(\bar{\gamma}) \omega^{D/2}) B(\bar{\gamma}) = f_L(\bar{\gamma}) + (f(\bar{\gamma}) - r(\bar{\gamma})) \omega^D + f_r(\bar{\gamma}) \omega^{D+1}$$

follows directly from the correctness of equations 4.3 and 4.4.

4.4 Knowledge soundness

We show that there exists an extractor E as needed to satisfy Definition 13. Assume an algebraic adversary outputs a commitment $c \in \mathbb{G}_1$. An algorithm E given access to a linear combination $c = \sum_{i=1}^{s} v_i l_i$, where $\bar{v} = v_1, \ldots, v_s$ is coefficient vector provided by algebraic $\mathcal{A}$ and $l_1, \ldots, l_s$ is list of all elements in $\mathbf{srs_1}$.Then E outputs witness polynomial

$$f = \sum_{i=0}^{s} v_i X^i.$$

However $\mathcal{A}$ is a cheating adversary, so $(f, c, \bar{\alpha}^{(1)}, \ldots, \bar{\alpha}^{(\mu)}, r) \notin \mathcal{R}$ for some $i \in [t]$. Hence

$$f(\bar{\alpha}^{(i)}) \neq r(\bar{\alpha}^{(i)}), \text{for some } i \in [t]. \tag{4.5}$$

The Adversary $\mathcal{A}$ runs the Open protocol with values $(d, c, \bar{\alpha}^{(1)}, \ldots, \bar{\alpha}^{(t)}, r)$ and tries to convince $\mathcal{V}$ to decide accept. We show that it comes with probability at most $\mathsf{negl}(\lambda)$.

The ideal I is uniquely determined by evaluating points $\bar{\alpha}^{(1)}, \ldots, \bar{\alpha}^{(t)}$. Gröbner basis $b_1, \ldots b_{\mu^t}$ of the ideal I is computed by both $\mathcal{A}$ and $\mathcal{V}$ in the protocol part a). Thus, $\mathcal{A}$ has to proceed in the rest of Open with these $b_1, \ldots, b_{\mu^t}$.

At this point, suppose that the adversary $\mathcal{A}$ computes polynomials H_L, H_R, f_L, f_R satisfying the equation 4.2:

$$\begin{aligned}(H_L(\bar{X}, W) + H_R(\bar{X}, W)W^{D/2}) \cdot B(\bar{X}, W) = \\ f_L(\bar{X}, W) + (f - r)(\bar{X}, W)W^D + f_R(\bar{X}, W)W^{D+1}.\end{aligned}$$

Let $H(\bar{X}, W) \coloneqq H_L(\bar{X}, W) + H_R(\bar{X}, W)W^{D/2}$. And denote the terms in $H(\bar{X}, W)$ by h_i as follows: $H(\bar{X}, W) = \sum_{i=0}^{D} h_i W^{D-i}$.

Since SRS upper bounds the degree of variable W by $D - 1$, we can uniquely compare monomials including W^D:

$$\sum_{i=0}^{D} h_i b_i W^D = (f - r)(\bar{X})W^D \Rightarrow \sum_{i=0}^{D} h_i b_i = (f - r)(\bar{X})$$

Polynomials $b_1, \ldots, b_{\mu^t}$ form a basis of ideal I. Hence $(f - r)(\bar{X}) \in I$, what is in a contradiction with the equation 4.5. It implies, that polynomials H_L, H_R, f_L, f_R satisfying eq. (4.2) do not exist.

Nevertheless, $\mathcal{A}$ outputs commitments to (H_L, H_r, f_L, f_R). Verifier $\mathcal{V}$ sends a random challenge $\bar{\gamma} \in \mathbb{F}^{\mu+1}$ and using Open protocol from the first scheme (Section 3.1), the adversary $\mathcal{A}$ convinces $\mathcal{V}$ that $H_L(\gamma), H_R(\gamma), f_L(\gamma), f_R(\gamma), f(\gamma)$ are correct evaluation of polynomials H_L, H_R, f_L, f_R, f.

From knowledge soundness of the first protocol (Theorem 5), $\mathcal{A}$ convinces $\mathcal{V}$, where

$$\begin{aligned}(H_L, c_{HL}, \gamma, H_L(\gamma)) \notin \mathcal{R}, or \\ (H_R, c_{HR}, \gamma, H_R(\gamma)) \notin \mathcal{R}, or \\ (f_L, c_{f_L}\gamma, f_L(\gamma)) \notin \mathcal{R}, or \\ (f_R, c_{f_R}, \gamma, f_R(\gamma)) \notin \mathcal{R}, or \\ (f, c_f, \gamma, f(\gamma)) \notin \mathcal{R},\end{aligned}$$

with only $\mathsf{negl}(\lambda)$ probability.

In the final step, $\mathcal{V}$ verifies the correctness of equation Equation (4.1) with values $H(\bar{\gamma}), f(\bar{\gamma}), f_L(\bar{\gamma}), f_R(\bar{\gamma}), f(\bar{\gamma})$.
To show that knowledge soundness holds for the entire Open protocol, we have to estimate the probability that the Verifier outputs accept, even if the equation 4.2 doesn't hold.

We can use Schwartz–Zippel lemma (Proposition 3). Let

$$f_{SZ} \coloneqq (H_L + H_R W^{D/2}) \cdot B - f_L - (f - r)W^D - f_R W^{D+1}$$

be a polynomial from the equation 4.1. The total degree of f_{SZ} is at most $d \cdot 2D$. Let $S = \mathbb{F}$ be subset of $\mathbb{F}$, in this case equal to $\mathbb{F}$.

In protocol step f) a point $\bar{\gamma} = (\gamma_1, \ldots, \gamma_\mu, \omega)$ is randomly chosen from S. Then using Schwartz–Zippel lemma we obtain

$$\Pr[f_{SZ}(\gamma_1, \ldots, \gamma_\mu, \omega) = 0] \leq \frac{d \cdot 2D}{|\mathbb{F}|}.$$

We have estimation for parameters d, D and $|\mathbb{F}|$ by security parameter λ.

$$\begin{aligned} d, \mu &= poly(\lambda) \\ D &= \mu^t - 1 \\ |\mathbb{F}| &= 2^{\omega(\lambda)} \end{aligned}$$

If we suppose t is $poly(\lambda)$, the fraction

$$\frac{d \cdot 2D}{|\mathbb{F}|} = \frac{2d(\mu^t - 1)}{|\mathbb{F}|} = \frac{poly(\lambda)^{poly(\lambda)}}{2^{\omega(\lambda)}} = \mathsf{negl}(\lambda).$$

Furthermore, if we limit the parameter t to be a constant, we can suppose a smaller size of the finite field $|\mathbb{F}| = \lambda^{\omega(1)}$, and we have the following estimation:

$$\frac{d \cdot 2D}{|\mathbb{F}|} = \frac{2d(\mu^t - 1)}{|\mathbb{F}|} = \frac{poly(\lambda)}{\lambda^{\omega(1)}} = \mathsf{negl}(\lambda).$$

In both case $\Pr[f_{SZ}(\gamma_1, \ldots \gamma_\mu, \omega) = 0] \leq \mathsf{negl}(\lambda)$. Thus, the Verifier $\mathcal{V}$ accepts the equation 4.1 with only probability negl (λ).

4.5 Summary

Theorem 9. *Assuming* Setup *generates additive groups $\mathbb{G}_1, \mathbb{G}_2$ for which the Q-DLOG assumption holds. Then the scheme defined in the section 4.1 is a Polynomial commitment scheme - based on SRS, where* Open *part is an interactive argument of knowledge with respect to the relation $\mathcal{R}$, defined in 2.1.*

Proof. The scheme includes all three parts Setup, Commit, Open from the definition Polynomial commitment scheme (Definition 19). To be Open an interactive argument of knowledge, we have to prove completeness (Definition 11) and knowledge soundness (Definition 13).

The completeness is proved in Section 4.3 and the knowledge soundness supposing the Q-DLOG assumption for $\mathbb{G}_1, \mathbb{G}_2$ is proved in Section 4.4. Note that the Q-DLOG assumption is not directly used in Section 4.4, but the proof refers to Section 3.4, where the Q-DLOG assumption is essential. □

4.5.1 Binding

Since the Commit part is the same as Commit in the scheme for one evaluating point (defined in Section 3.1), we just refer to Section 3.5.1.

Theorem 10. *Let* Setup *from polynomial commitment scheme defined in section 4.1 outputs* $\mathbb{G}_1, \mathbb{G}_2$ *satisfying Q-DLOG assumption. Then the scheme is binding (Definition 9).*

The proof of this theorem is identical to the proof that a polynomial commitment scheme for one evaluation point is biding (Theorem 6).

Theorem 11. *Assuming* Setup *generates additive groups* $\mathbb{G}_1, \mathbb{G}_2$ *for which Q-DLOG assumption holds. The polynomial commitment scheme defined in section 4.1 is evaluation binding (Definition 20).*

Proof. The knowledge soundness property is shown in the section 4.4. Then we can use Theorem 3, which states that if a polynomial commitment scheme is binding and has knowledge soundness property, then the polynomial commitment scheme is also evaluation binding. □

4.6 Scheme complexity

Theorem 12. *The polynomial commitment scheme defined in the section 4.1 has the following properties:*

i) $\mathbf{srs_1}, \mathbf{srs_2}$ *consist of* $D \cdot \binom{\mu+d}{\mu}$ $\mathbb{G}_1$ *elements and* $\mu + 2$ *elements of* $\mathbb{G}_2$.

ii) For integer $n \leq d$ *and* $f \in \mathbb{F}_{<n}[X_1, \ldots, X_\mu]$.
Computation Commit $(f, \mathbf{srs_1}, \mathbf{srs_2})$ *requires* $\binom{\mu+n}{\mu}$ *multiplications in* $\mathbb{G}_1$.

iii) Prover $\mathcal{P}$ *sends* $(4 + 5 \cdot \mu)$ *elements of* $\mathbb{G}_1$ *and* 5 *elements of* $\mathbb{F}$.

Proof.

i) Additionally to SRS from the first scheme (section 3.1), each element in $\mathbf{srs_1}$ is multiplied by $w^0, w, w^2, \ldots, w^{D-1}$. Hence, we have $D \cdot \binom{\mu+d}{\mu}$ elements in $\mathbb{G}_1$. $\mathbf{srs_2}$ is simply consist of $[1]_2, [x_1]_2, \ldots, [x_\mu]_2, [w]_2$, i.e., $\mu + 2$ elements of $\mathbb{G}_2$.

ii) Commit part is the same for both schemes. We can use the same argument as in Theorem 8, section ii).

iii) During Open protocol, Prover sends commitments to H_L, H_R, f_L, f_R, i.e., four $\mathbb{G}_1$ elements. Then Prover runs five times Open protocol from the scheme with one evaluating points. By Theorem 8 section iii), in total we have $5 \cdot \mu$ elements from $\mathbb{G}_1$. The Prover also has to send $f(\bar{\gamma}), f_R(\bar{\gamma}), f_L(\bar{\gamma}), H_R(\bar{\gamma})$, $H_L(\bar{\gamma})$, i.e., 5 elements of $\mathbb{F}$. □

5. Prover's Communication Complexity

This chapter provides a comparison of the Prover's communication complexity across the other related polynomial commitment schemes presented by Bünz et al. [2019] and Boneh et al. [2020].

In the first table, it is assumed the scheme is performed with only univariate polynomials.

Table 5.1: Prover's communication complexity in Open protocol with univariate polynomials ($\mu = 1$)

	1 evaluating point	t evaluating points
This work	$1\mathbb{G}_1$	$9\mathbb{G}_1 + 5\mathbb{F}$
Bünz et al. [2019]	$2log(d)\mathbb{G} + 2\log(d)\mathbb{Z}_p$	$2\log(d)\mathbb{G} + (t+1)\log(d)\mathbb{Z}_p$
Boneh et al. [2020]	$1\mathbb{G}_1$	$1\mathbb{G}_1$

The second comparison is focused on schemes with multivariate polynomials. The scheme presented by Boneh et al. [2020] is missing since the variant with multivariate polynomials is not suggested.

Table 5.2: Prover's communication complexity in Open protocol with multivariate polynomials

	1 evaluating point	t evaluating points
This work	$\mu\mathbb{G}_1$	$(4 + 5\mu)\mathbb{G}_1 + 5\mathbb{F}$
Bünz et al. [2019]	$\mu 2\log(d)\mathbb{G} + \mu 2\log(d)\mathbb{Z}_p$	$\mu 2\log(d)\mathbb{G} + \mu(t+1)\log(d)\mathbb{Z}_p$

5.1 Building a multivariate scheme from a univariate scheme

In this section, we discuss a possibility of an extension of Boneh et al. [2020] scheme, towards a multivariate version.

Bünz et al. [2019] presented an idea to transform a commitment scheme for polynomials with one variable into a version with multivariate polynomials. The idea is based on running the Open protocol for univariate polynomials in many rounds. Scheme present Open protocol multivariate polynomials which uses Open subprotocol for univariate polynomials.

Starting with μ-multivariate polynomial, in each round, one variable is reduced. In the next step, the same Open protocol is running again with a new polynomial with $\mu - 1$ variables. This process repeats until one variable remains. In the a simple protocol for univariate polynomial is running, and Verifier outputs accept or reject.

We present the technique from Bünz et al. [2019] in more detail. The crucial point is using an appropriate encoding, which uniquely encodes each variable.

Suggested encoding of polynomial f is $f(q)$, an evaluation of polynomial f in sufficiently large $\mathbb{F}$ element q.

In the encoding for multivariate polynomials, each variable is raised to the power of q. The range of the powers for each variable is different with no overlap.

$$\mathsf{Enc}(f(X_1, \ldots, X_\mu)) = f(q_1, \ldots, q_\mu),$$

where $q_i = q^{(d+1)^(i-1)}$, for each $i \in [\mu]$ and $f(X_1, \ldots, X_\mu)$ has degree d in each variable.
In addition, suggesting encoding is suitable for Commit part, since commitment is computed in following way:

$$\mathsf{Commit}(f(X_1, \ldots, X_\mu)) = g^{f(q_1, \ldots, q_\mu)},$$

where g is a group element. The encoding satisfies the property:

$$\mathsf{Commit}(f(X_1, \ldots, X_\mu)) = \mathsf{Commit}(\mathsf{Enc}(f(X_1, \ldots, X_\mu))).$$

In each round of Open protocol for multivariate polynomials, one variable is reduced. More precisely, in the first step polynomial is encoded in each variable except last one.

$$\hat{f}(X_\mu) \coloneqq f(q_1, \ldots, q_{\mu-1}, X_\mu)$$

Using Open subprotocol for univariate polynomial, the commitment of $\hat{f}(X_\mu)$ is verified with respect to the last variable. In the case Verifier accepts, the polynomial is backward decoded to $\mu - 1$ variables polynomial.

$$\tilde{f}(X_1, \ldots, X_{\mu-1}) = \mathsf{Dec}(\bar{f}),$$

where $\bar{f}$ is single element in $\mathbb{F}$, since $\bar{f}$ is the value in the end of the protocol for univariate polynomial, where the degree of polynomial is decreasing to a constant polynomial.

Important is that decoding of $\bar{f}$ is unique under defined range of coefficients, so the polynomial $\tilde{f}(X_1, \ldots, X_{\mu-1})$ is uniquely determined.

In the next round, Open protocol runs with polynomial $\tilde{f}(X_1, \ldots, X_{\mu-1})$.

Step by step, all variables except one are reduced. The final step is running protocol for univariate polynomial, where in the end, Verifier outputs accept or reject. In conclusion, the protocol for multivariate running the protocol for univariate polynomials for each variable separately using appropriate encoding.

A natural question is whether the same technique could be used directly for the Boneh et al. [2020] scheme. In that case, the same technique allows us to obtain Prover's communication complexity for multivariate polynomials equal to Prover's communication complexity for univariate polynomials multiplied by μ, e.g., $\mu\mathbb{G}_1$.

It is not necessary to introduce the Boneh et al. [2020] scheme deeply, because this work used the same Commit algorithm.

$$\mathsf{Commit}(f(X_1, \ldots, X_\mu)) = [f(x_1, \ldots, x_\mu)]_1$$

We point out that compared to the previous protocol Bünz et al. [2019], elements $x_1, \ldots, x_\mu$ are secret for both participant Prover and Verifier.

In addition, secret elements $x_1, \ldots, x_\mu$ are independent of each other. In comparison to encoding in Bünz et al. [2019] protocol, where $q_1, \ldots, q_\mu$ are each a power to predefined q.

Even though it still holds that

$$\mathsf{Commit}(f(X_1, \ldots, X_\mu)) = \mathsf{Commit}(f(x_1, x_2, \ldots, x_{\mu-1}, X_\mu),$$

the value $f(x_1, x_2, \ldots, x_{\mu-1}, X_\mu)$ cannot be computed in the Open protocol by none of the participants as in the Bünz et al. [2019]. For the same reason, it cannot be performed by any of the participant decoding $\mathsf{Dec}([f(x_1, \ldots, x_\mu)]_1)$.

Hence, we cannot fix a variable in the same way as in the Bünz et al. [2019] protocol, and the technique of Bünz et al. [2019] is not applicable for Boneh et al. [2020] protocol.

Conclusion

We presented new polynomial commitment schemes, which extend the previously studied scheme from Boneh et al. [2020] and enable Prover to commit to multivariate polynomials.

In the first scheme (defined in section 3.1), Verifier can verify polynomial evaluation only in one evaluation point. The Open protocol of the first scheme also served as an essential part of the Open protocol in the second polynomial commitment scheme.

The second scheme (defined in section 4.1) is designed for more evaluation points. Once the Prover commits to a polynomial f, Verifier obtains the polynomial r, equal to the committed polynomial f in all evaluating points. After that, the correctness of the obtained polynomial could be verified by running a single Open protocol.

Our work is mainly focused on reducing Prover's communication. Since Boneh et al. [2020] did not present a scheme for multivariate polynomials, we compare the Prover's communication complexity during Open protocol with another studied protocol presented in Bünz et al. [2019]. In their protocol, Prover has to send $\mu \cdot 2\log(d)$ group elements with $(t+1)\log(d)$ element from a finite field. In our work, Prover's communication is reduced to $4+5\mu$ group elements and 4 elements from a finite field.

We present proofs that both schemes satisfy properties *completeness* and *knowledge soundness*. However, the *knowledge soundness* is shown in the Algebraic group model, which gives more restrictions to the adversary than the Standard model. One open question is how the *knowledge soundness* could be proved in the Standard model, where an adversary does not have to recover the representation of group elements.

Boneh et al. [2020] presented scheme which enables committing to more than one polynomials. It is done by adding a random element from the finite field, which uniquely combines committed polynomials. Our scheme could be possibly extended in a similar way.

Bibliography

Philippe Loustaunau William W. Adams. *An introduction to Groebner bases*. Graduate Studies in Mathematics, Vol 3. American Mathematical Society, Praha, 1994. ISBN 08-2183-804-0; 9780821838044.

Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. Efficient polynomial commitment schemes for multiple points and polynomials. *IACR Cryptol. ePrint Arch.*, page 81, 2020. URL `https://eprint.iacr.org/2020/081`.

Benedikt Bünz, Ben Fisch, and Alan Szepieniec. Transparent snarks from DARK compilers. *IACR Cryptol. ePrint Arch.*, page 1229, 2019. URL `https://eprint.iacr.org/archive/2019/1229/1582704065.pdf`.

Sandro Coretti, Yevgeniy Dodis, and Siyao Guo. Non-uniform bounds in the random-permutation, ideal-cipher, and generic-group models. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 693–721. Springer, 2018. doi: 10.1007/978-3-319-96884-1_23. URL `https://doi.org/10.1007/978-3-319-96884-1_23`.

Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 33–62. Springer, 2018. doi: 10.1007/978-3-319-96881-0_2. URL `https://doi.org/10.1007/978-3-319-96881-0_2`.

Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. *IACR Cryptol. ePrint Arch.*, page 953, 2019. URL `https://eprint.iacr.org/2019/953`.

Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 177–194. Springer, 2010. doi: 10.1007/978-3-642-17373-8_11. URL `https://doi.org/10.1007/978-3-642-17373-8_11`.