A Brief Survey of Two Recent Polynomial Commitment Schemes from Lattices

Minuk Ban and Hyung Tae Lee
School of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea
{apolo0430, hyungtaelee}@cau.ac.kr

***Abstract*—A polynomial commitment scheme (PCS) enables a prover to commit to a polynomial and later prove the correctness of its evaluation without revealing the polynomial. Although discrete logarithm-based PCSs offer succinct proofs, they are not quantum-safe. Lattice-based PCSs provide post-quantum security and additive homomorphism, making them suitable for applications such as zero-knowledge proofs and secure multi-party computation. In this article, we review two recent lattice-based PCSs, Greyhound and HyperWolf, both relying on the Module-SIS assumption but differing in target polynomial classes and proof techniques. In particular, Greyhound achieves a smaller proof size $O(\log \log N)$ through folding and LaBRADOR proofs, while HyperWolf supports univariate and multilinear polynomials with lower verifier cost $O(\log N)$ using hypercube evaluation.**

***Index Terms*—Polynomial commitment schemes, Post-quantum cryptography, Lattice-based cryptography**

I. Introduction

A polynomial commitment scheme (PCS) is a cryptographic primitive that allows a prover to commit to a degree-bounded polynomial f and later produce a succinct proof π to convince a verifier that a claimed evaluation $y = f(x)$ is correct, without revealing f in its entirety. PCS has become a fundamental building block in numerous cryptographic protocols, including verifiable secret sharing (VSS) [1], secure multi-party computation (MPC) [2], and zero-knowledge succinct non-interactive arguments of knowledge (zkSNARKs) [3]–[5].

The concept of PCS was first introduced in [6], where the authors proposed a construction for univariate polynomials based on the discrete logarithm (DL) assumption. Since then, significant progress has been made in the extension of PCS to multilinear [5], [7] and multivariate polynomials [8]–[10], still under the DL assumption. PCSs based on DL enjoy constant or sublinear proof size and verification cost, but are vulnerable to quantum attacks.

An alternative approach is to construct PCS from error-correcting codes (ECCs) [11], [12], which are believed to remain secure against quantum adversaries. However, such constructions typically do not preserve homomorphic properties, limiting their applicability in protocols that require efficient proof composition.

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2023-00229400, 50%) and the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2021-NR058311, 50%). H. T. Lee is the corresponding author.

Lattice-based PCS provides both post-quantum security and (additive) homomorphism, due to the natural homomorphic structure of underlying lattices. Many PCSs based on lattices have been proposed [13]–[16], offering various trade-offs in proof size, efficiency, and supported polynomial classes. In this work, we focus on the recently proposed lattice-based PCSs, Greyhound [14] and [15], which rely on the same lattice assumption and employ similar techniques for polynomial evaluation. Our study compares their design choices, underlying assumptions, and performance characteristics.

II. Preliminaries

Notation. Let λ be the security parameter. For $n \in \mathbb{N}$, we define $[n] := \{0, ..., n-1\}$. Let q be an odd prime and define the ring of integers modulo q as $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$. For a power of two d, $\mathbb{R}_q := \mathbb{Z}_q[X]/(X^d + 1)$ be the ring of integers of the $2d$-th cyclotomic field where $\mathbb{Z}_q[X]$ is the ring of polynomials over $\mathbb{Z}_q$. $\mathbb{R}_q^\times$ denotes the set of invertible elements in $\mathbb{R}_q$. For a set S, $s \xleftarrow{\$} S$ denotes that s is sampled uniformly at random from S.

Bold symbols (e.g., $\mathbf{a}, \vec{\mathbf{a}}, \mathbf{A}$) denote elements, vectors, or matrices over $\mathbb{R}_q$, while their non-bold counterparts (e.g., $a, \vec{a}, A$) denote corresponding objects over $\mathbb{Z}_q$. For $\mathbf{f} = \sum_{i=0}^{d-1} f_i X^i \in \mathbb{R}_q$, $(\mathbf{f}) = f_0$ denotes the constant term of $\mathbf{f}$. For a ring element $\mathbf{f} \in \mathbb{R}_q$ and a ring vector $\vec{\mathbf{f}} \in \mathbb{R}_q^n$, their ℓ_p-norms are defined as $\|\mathbf{f}\|_p := \left(\sum_{i=0}^{d-1} |f_i|^p\right)^{1/p}$ and $\|\vec{\mathbf{f}}\|_p := \left(\sum_{i=0}^{n-1} \|\mathbf{f}_i\|_p^p\right)^{1/p}$. Throughout the paper, unless otherwise specified, we refer to the ℓ_2-norm as $\|\cdot\|$.

A. Gadget Matrix

For a positive integer $b \geq 2$, we define the gadget vector $\vec{g}_b^\intercal = (1, b, b^2, ..., b^{\delta-1})$, where $\delta = \lceil \log_b q \rceil$. Then, the gadget matrix $G_{b,n}$ is defined as $G_{b,n} := I_m \otimes \vec{g}_b^\intercal$ where I_m is the $m \times m$ identity matrix and $A \otimes B$ denotes the Kronecker product between matrices A and B. We define the inverse function $G_{b,n}^{-1} : \mathbb{R}_q^{n \times m} \to \mathbb{R}_q^{\delta n \times m}$ which decomposes each entry with respect to the base b. For a matrix $\mathbf{A} \in \mathbb{R}_q^{n \times m}$, $\hat{\mathbf{A}} = G_{b,n}^{-1}(\mathbf{A})$ denotes the decomposition of $\mathbf{A}$. We have $G_{b,n} G_{b,n}^{-1}(\mathbf{A}) = \mathbf{A}$ and $\|\hat{\mathbf{A}}_i\| \leq \frac{a}{2}\sqrt{\delta n}$ for each column $\hat{\mathbf{A}}_i$ of $\hat{\mathbf{A}}$.

B. Conjugation Automorphism

Let $\sigma_{-1} : \mathbb{R}_q \to \mathbb{R}_q$ be the conjugation automorphism, which maps X to X^{-1}. Given $\mathbf{a} \in \mathbb{R}_q$, the automorphism of $\mathbf{a}$ is defined as

$$\sigma_{-1}(\mathbf{a}) = \sum_{i=0}^{d-1} a_i X^{-i} = a_0 - \sum_{i=1}^{d-1} a_i X^{d-i}.$$

For vectors $\vec{\mathbf{a}}, \vec{\mathbf{b}} \in \mathbb{R}_q^n$, if their coefficients are represented as $\vec{a}, \vec{b} \in \mathbb{Z}_q^{nd}$, then the inner product between $\vec{a}$ and $\vec{b}$ is equal to $(\langle \sigma_{-1}(\vec{\mathbf{a}}), \vec{\mathbf{b}} \rangle)$.

C. Interactive Proof

Let $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$ be a ternary relation. For a triple $(\quad, x, w) \in \mathcal{R}$, we refer to as the public parameter, x as the statement, and w as the witness for x. We denote $\mathcal{R}(\quad, x) := \{w : (\quad, x, w) \in \mathcal{R}\}$.

Definition II.1 (Interactive Proof System)**.** *An interactive proof system $\Pi = (\mathcal{S}, \mathcal{P}, \mathcal{V})$ for a relation $\mathcal{R}$ is an interactive protocol consisting of three probabilistic polynomial-time (PPT) algorithms: the setup algorithm $\mathcal{S}$, prover $\mathcal{P}$ and verifier $\mathcal{V}$. The goal of the system is to enable $\mathcal{P}$ to convince $\mathcal{V}$ that it possesses a witness w for a given statement x. Both $\mathcal{P}$ and $\mathcal{V}$ take as public input the statement x and the public parameters , while $\mathcal{P}$ additionally holds a private witness $w \in \mathcal{R}(\quad, x)$. At the end of the protocol, $\mathcal{V}$ accepts the claim of $\mathcal{P}$ if $(\quad, x, w) \in \mathcal{R}$ holds; otherwise, it rejects.*

We write an interactive protocol between $\mathcal{P}$ and $\mathcal{V}$ to obtain the communication transcript tr and the $\mathcal{V}$'s decision bit b as $(tr, b) \leftarrow \langle \mathcal{P}(\quad, x, w), \mathcal{V}(\quad, x) \rangle$. If $\mathcal{V}$ accepts the claim of $\mathcal{P}$, we set $b = 1$, and $b = 0$ otherwise.

Definition II.2 (Completeness)**.** *An interactive proof system $\Pi = (\mathcal{S}, \mathcal{P}, \mathcal{V})$ for the relation $\mathcal{R}$ achieves completeness with completeness error $\epsilon(\lambda)$ if, for every valid instance $(\quad, x, w) \in \mathcal{R}$, the probability that $\mathcal{V}$ rejects (i.e., outputs $b = 0$) is at most $\epsilon(\lambda) +$ (λ).*

We say that Π achieves perfect completeness when the completeness error $\epsilon(\lambda) = 0$.

Definition II.3 (Knowledge Soundness)**.** *An interactive proof system $\Pi = (\mathcal{S}, \mathcal{P}, \mathcal{V})$ for the relation $\mathcal{R}$ is said to be knowledge sound with knowledge error $\varepsilon(\lambda)$ if there exists an expected PPT extractor $\mathcal{E}$ such that, for any stateful PPT adversary $\mathcal{P}^*$, the probability that $\mathcal{P}^*$ generates a tuple (x, w^*) and convinces $\mathcal{V}$ to accept (i.e., $b = 1$), while the extractor $\mathcal{E}$ fails to find a witness w with $(\quad, x, w) \in \mathcal{R}$, is at most $\varepsilon(\lambda) +$ (λ).*

The extractor $\mathcal{E}$ is given black-box oracle access to the (malicious) prover $\mathcal{P}^*$ and may rewind it to any point in the interaction.

D. Polynomial Commitment Scheme

The PCS is a cryptographic primitive that allows a prover to commit to a polynomial f (typically through its coefficients), and later prove the correctness of its evaluation at a chosen point. Throughout the paper, we focus on PCSs in the interactive setting.

Definition II.4 (Polynomial Commitment Scheme (PCS))**.** *The PCS for $f \in \mathbb{Z}_q^{<N}[X]$ with slack space $\mathcal{SL}$ consists of the following four PPT algorithms:*

- $(1^\lambda) \to$ *: On input the security parameter λ, output the public parameter .*
- $(\quad, f) \to (\quad, \quad)$*: On input and a polynomial f, output a commitment and a decommitment state .*
- $(\quad, \quad, f, \quad, c) \to 0/1$*: On input , a commitment , a polynomial f, a decommitment state , and a relaxation factor $c \in \mathcal{SL}$, output a bit indicating whether is a valid commitment to f under .*
- $(\quad, \quad, x, y; (f, \quad))$*: An interactive protocol between prover $\mathcal{P}$ and verifier $\mathcal{V}$, where the common input is , a commitment , an evaluation point x, and a value y. $\mathcal{P}$ additionally holds f and .*

The PCS must satisfy evaluation completeness, weak binding, and knowledge soundness, formally defined below.

Definition II.5 (Evaluation Completeness)**.** *A PCS $=$ (, , ,) satisfies evaluation completeness with completeness error $\epsilon(\lambda)$ if for every polynomial f and any evaluation point $x \in \mathbb{Z}$, the probability that the verifier rejects valid proof from $\mathcal{P}$ for $y = f(x)$ is $\epsilon(\lambda) +$ (λ).*

Definition II.6 (Weak Binding)**.** *A PCS $=$ (, , ,) satisfies weak binding if for every PPT adversary $\mathcal{A}$, the probability that $\mathcal{A}$ generates two distinct and valid tuples $(f, \quad, c)$ and $(f', \quad', c')$ such that both tuples open successfully to the same commitment under the same public parameters is negligible.*

Definition II.7 (Knowledge Soundness)**.** *A PCS $=$ (, , ,) is knowledge sound with knowledge error $\varepsilon(\lambda)$ if for every stateful PPT adversary $\mathcal{P}^*$, the probability that $\mathcal{V}$ accepts the proof of $\mathcal{P}^*$ for $f(x) = y$ where either is not a commitment to f or y is not the evaluation of f at x is $\varepsilon(\lambda) +$ (λ).*

E. Inner and Outer Commitment with Lattice Problems

First, we recall the standard module short integer solution (Module-SIS) assumption [17].

Definition II.8 (Module-SIS)**.** *We say that the module short integer solution assumption $_{n,m,q,\beta}$ holds if for any PPT adversary $\mathcal{A}$, the following holds:*

$$\Pr\left[\mathbf{A}\vec{\mathbf{z}} = \vec{\mathbf{0}} \wedge 0 < \|\vec{\mathbf{z}}\| \leq \beta \,\middle|\, \begin{matrix} \mathbf{A} \leftarrow \mathbb{R}^{n \times m} \\ \vec{\mathbf{z}} \leftarrow \mathcal{A}(\mathbf{A}) \end{matrix}\right] \leq \quad (\lambda).$$

Next, we recall the inner and outer commitments from [18]. Let λ be the security parameter, and let $n, m, r, b, q \in \mathbb{Z}$ be positive integers. Denote by $\bar{\beta}, \bar{\gamma}, \bar{\tau} > 0$ the security-related norm bounds, and set $\delta = \lceil \log_b q \rceil$. Given the public parameter $= (\mathbf{A} \in \mathbb{R}_q^{\kappa \times m}, \mathbf{B} \in \mathbb{R}_q^{\kappa \times \kappa\delta n})$, we define the commitment to a matrix $\mathbf{S} = \{\vec{\mathbf{s}}_0, ..., \vec{\mathbf{s}}_{n-1}\} \in \mathbb{R}_q^{m \times n}$ in two steps:

1. Compute the *inner commitments* $t_{in,i} := \mathbf{A}\vec{s}_i \in \mathbb{R}_q^{\kappa}$.
2. Compute the *outer commitment* $t_{out} := \mathbf{B}\,\hat{t}_{in}$ where $\hat{t}_{in} := (G_{b,\kappa}^{-1}(t_{in,i}))_{i\in[n]}$.

A weak opening for the commitment t_{out} is a tuple $(\vec{s}_i, \hat{t}_{in,i}, c_i)_{i\in[n]}$ which satisfies all the following conditions:

$$\|c_i\vec{s}_i\| \le \bar{\beta}, \|c_i\| \le \bar{\tau}, c_i \in \mathbb{R}_q^{\times}, \mathbf{A}\vec{s}_i = G_{b,\kappa}\,\hat{t}_{in,i},$$

$$\mathbf{B}\begin{bmatrix}\hat{t}_{in,0}\\ \vdots \\ \hat{t}_{in,n-1}\end{bmatrix} = t_{out} \text{ and } \left\|\begin{bmatrix}\hat{t}_{in,0}\\ \vdots \\ \hat{t}_{in,n-1}\end{bmatrix}\right\| \le \bar{\tau}.$$

The above inner and outer commitment scheme satisfies the binding property under the Module-SIS assumption.

Lemma II.1. *[14, Lemma 2.11] There is a deterministic algorithm, that given two weak openings $(\vec{s}_i, \hat{t}_{in,i}, c_i)_{i\in[n]}$ and $(\vec{s}'_i, \hat{t}'_{in,i}, c'_i)_{i\in[n]}$ for the commitment t_{out} such that $\vec{s}_i \neq \vec{s}'_i$ for some $i \in [n]$, outputs a vector $\vec{z} \in \mathbb{R}_q^{m+\kappa\delta z}$ such that $[\mathbf{A}|\mathbf{B}]\vec{z} = \mathbf{0}$ and $0 \le \|\vec{z}\| \le \max(4\bar{\tau}\bar{\beta}, 2\bar{\gamma})$.*

III. Greyhound

Greyhound [14] is an interactive PCS for a univariate polynomial $f(X) = \sum_{i=0}^{N-1} f_i X^i$. The key idea of Greyhound for efficient proof of univariate polynomial evaluation is folding. Let N be a degree of polynomial f that can be represented by $N = mn$ for positive integers m, n. Then, we can write

$$y = \vec{a}_0^{\mathsf{T}} \cdot \begin{bmatrix} f_0 & f_m & \cdots & f_{(n-1)m} \\ f_1 & f_{m+1} & \cdots & f_{(n-1)m+1} \\ \vdots & \vdots & \ddots & \vdots \\ f_{m-1} & f_{2m-1} & \cdots & f_{nm-1} \end{bmatrix} \cdot \vec{a}_1, \quad (1)$$

where $\vec{a}_0^{\mathsf{T}} = [1, x, x^2, \cdots, x^{m-1}]$ and $\vec{a}_1^{\mathsf{T}} = [1, x^m, x^{2m}, \cdots, x^{(n-1)m}]$. Whereas the overall commitment scheme operates over $\mathbb{R}_q$, the above decomposition is represented over $\mathbb{Z}_q$. Thus, we need to convert elements from $\mathbb{Z}_q$ to $\mathbb{R}_q$.

In [19], the authors demonstrate how to convert the proof of polynomial evaluations from $\mathbb{Z}_q$ to $\mathbb{R}_q$. Suppose that the degree of a polynomial N is divisible by the ring dimension d. Then

$$y = \sum_{i=0}^{N-1} f_i x^i = \sum_{i=0}^{N/d-1} \left(\sum_{j=0}^{d-1} f_{id+j} x^j \right) \cdot (x^d)^i.$$

Suppose that two elements over $\mathbb{R}_q$ are defined as

$$\mathbf{x} = \sum_{j=0}^{d-1} x^j X^j, \ \mathbf{f}_i = \sum_{j=0}^{d-1} f_{id+j} X^j \text{ for } i \in [N/d-1]$$

by conjugation automporphism. Then, as shown in [20], y is equal to the constant term of the following expression:

$$\mathbf{y} = \sum_{i=0}^{N/d-1} \sigma_{-1}(\mathbf{x}) \cdot \mathbf{f}_i \cdot (x^d)^i.$$

A. Building Block for Greyhound

At the end of , the verifier needs to check the relation

$$\mathcal{R} := \{(\mathbf{P}, \vec{\mathbf{h}}, \bar{\gamma}), \vec{\mathbf{z}}) : \mathbf{P}\vec{\mathbf{z}} = \vec{\mathbf{h}} \wedge \|\vec{\mathbf{z}}\| \le \bar{\gamma}\}$$

is satisfied where

$$\mathbf{p} := \begin{bmatrix} \mathbf{D} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} & \mathbf{0} \\ \vec{\mathbf{a}}_1^{\mathsf{T}} G_{b_1,n} & \mathbf{0} & \mathbf{0} \\ \vec{\mathbf{c}}^{\mathsf{T}} G_{b_1,n} & \mathbf{0} & -\vec{\mathbf{a}}_0^{\mathsf{T}} \\ \mathbf{0} & \vec{\mathbf{c}} \otimes G_{b_1,n} & -\mathbf{A} \end{bmatrix}, \vec{\mathbf{h}} := \begin{bmatrix} \vec{\mathbf{v}} \\ t_{out} \\ \sigma_{-1}(\mathbf{x})^{-1} \cdot \mathbf{y} \\ 0 \\ \mathbf{0} \end{bmatrix},$$

$$\bar{\gamma} := \sqrt{b_1^2(\kappa+1)\delta_1 nd + (n\tau b_0)^2 \delta_0 md} \text{ and } \vec{\mathbf{z}} := \begin{bmatrix} \hat{\vec{\mathbf{w}}} \\ \hat{t}_{in} \\ \vec{\mathbf{s}} \end{bmatrix}.$$

To prove knowledge of a short vector $\vec{\mathbf{z}}$ satisfying the relation $\mathcal{R}$, Greyhound employs the LaBRADOR proof system [18]. For the detailed construction of LaBRADOR, we refer the reader to [18].

B. Construction of Greyhound

Before presenting the detailed construction of Greyhound, we introduce its parameters in Table I.

Table I
PARAMETERS OF GREYHOUND

Notation	Description
q	prime modulus
N	number of coefficients
d	ring dimension
m, n	folding parameter
κ	height of matrices $\mathbf{A}, \mathbf{B}, \mathbf{D}$
b_0, b_1	decomposition base
δ_0, δ_1	$\delta_0 = \lceil \log_{b_0} q \rceil, \delta_1 = \lceil \log_{b_1} q \rceil$
τ	ℓ_1-norm of a challenge
$\bar{\gamma}$	ℓ_2-norm of $\vec{\mathbf{z}}$
$\bar{\beta}$	ℓ_2-norm bound of witness
$\mathcal{C}$	challenge space

Now, we describe the details of Greyhound.

- $(1^{\lambda}) \rightarrow$
  1. $\mathbf{A} \xleftarrow{\$} \mathbb{R}_q^{\kappa \times \delta_0 m}$
  2. $\mathbf{B} \xleftarrow{\$} \mathbb{R}_q^{\kappa \times \kappa\delta_1 n}$
  3. $\mathbf{D} \xleftarrow{\$} \mathbb{R}_q^{\kappa \times \delta_1 n}$
  4. **return** $:= (\mathbf{A}, \mathbf{B}, \mathbf{D})$
- $(\ , f \in \mathbb{Z}_q^{<N}[X]) \rightarrow (\ , \ )$
  1. $f(X) := \sum_{i=0}^{N-1} f_i X_i$
  2. **for** $i = 0, 1, ..., N/d - 1$:
  3. $\quad \mathbf{f}_i := \sum_{j=0}^{d-1} f_{id+j} X^j \in \mathbb{R}_q$
  4. **for** $i = 0, ..., n-1$:
  5. $\quad \vec{\mathbf{f}}_i^{\mathsf{T}} := (\mathbf{f}_{im}, ..., \mathbf{f}_{(i+1)m-1}) \in \mathbb{R}_q^m$
  6. $\quad \vec{\mathbf{s}}_i := G_{b_0,m}^{-1}(\vec{\mathbf{f}}_i)$
  7. $\quad t_{in,i} := \mathbf{A}\vec{\mathbf{s}}_i$
  8. $\quad \hat{t}_{in,i} := G_{b_1,\kappa}^{-1}(t_{in,i})$
  9. $\hat{t}_{in} := (\hat{t}_{in,i})_{i\in[n]}$
  10. $t_{out} := \mathbf{B}\,\hat{t}_{in}$
  11. $:= (\vec{\mathbf{s}}_i, \hat{t}_{in,i})_{i\in[n]}$
  12. **return** $(\ t_{out}, \ )$

- $(\quad, \quad, f, \quad, (c_i)_{i\in[n]}) \to 0/1$
  1. $f(X) := \sum_{i=0}^{N-1} f_i X_i$
  2. **for** $i = 0, 1, ..., N/d - 1$:
  3. $\quad \mathbf{f}_i := \sum_{j=0}^{d-1} f_{id+j} X^j \in \mathbb{R}_q$
  4. **for** $i = 0, ..., n-1$:
  5. $\quad \vec{\mathbf{f}}_i^{\mathsf{T}} := (\mathbf{f}_{im}, ..., \mathbf{f}_{(i+1)m-1}) \in \mathbb{R}_q^m$
  6. $\quad$ **if** $G_{b_0,m}\vec{\mathbf{s}}_i \neq \vec{\mathbf{f}}_i \vee \mathbf{A}\vec{\mathbf{s}}_i \neq G_{b_{1,\kappa}} \hat{} \ _{in,i}$:
  7. $\quad\quad$ **return** 0
  8. $\quad$ **if** $\|c_i \cdot \vec{\mathbf{s}}_i\| > \bar{\beta} \vee \|c_i\|_1 > \bar{\tau} \vee c_i \notin \mathbb{R}_q^\times$:
  9. $\quad\quad$ **return** 0
  10. $\hat{} \ _{in} := (\hat{} \ _{in,i})_{i\in[n]}$
  11. **if** $\| \hat{} \ _{in}\| > \bar{\gamma} \vee \mathbf{B} \hat{} \ _{in} \neq \quad$:
  12. $\quad$ **return** 0
  13. **return** 1
- $(\quad, \quad, x, y; (f, \quad))$
  1. $f(X) := \sum_{i=0}^{N-1} f_i X_i$
  2. $\mathbf{x} = \sum_{j=0}^{d-1} x^j X^j$
  3. $\vec{\mathbf{a}}_0^{\mathsf{T}} = [1, x^d, x^{2d}, ..., x^{(m-1)d}] G_{b_0,m}$
  4. $\vec{\mathbf{a}}_1^{\mathsf{T}} = [1, x^{md}, x^{2md}, ..., x^{(n-1)md}]$
  5. $\mathcal{P}$ computes:
  6. $\quad$ **for** $i = 0, 1, ..., N/d - 1$:
  7. $\quad\quad \mathbf{f}_i := \sum_{j=0}^{d-1} f_{id+j} X^j \in \mathbb{R}_q$
  8. $\quad \mathbf{y} := \sum_{i=0}^{N/d-1} \sigma_{-1}(\mathbf{x}) \cdot \mathbf{f}_i \cdot (x^d)^i$
  9. $\quad \vec{\mathbf{w}}^{\mathsf{T}} := \vec{\mathbf{a}}_0^{\mathsf{T}} [\vec{\mathbf{s}}_0 | \cdots | \vec{\mathbf{s}}_{n-1}]$
  10. $\quad \hat{\vec{\mathbf{w}}} := G_{b_1,n}^{-1}(\vec{\mathbf{w}})$
  11. $\quad \vec{\mathbf{v}} := \mathbf{D}\hat{\vec{\mathbf{w}}}$
  12. $\mathcal{P} \to \mathcal{V} : (\mathbf{y}, \vec{\mathbf{v}})$
  13. $\mathcal{V} \to \mathcal{P} : \vec{\mathbf{c}} \in \mathcal{C}^n$
  14. $\mathcal{P}$ computes $\vec{\mathbf{s}} := [\vec{\mathbf{s}}_0 | \cdots | \vec{\mathbf{s}}_{n-1}] \vec{\mathbf{c}}$.
  15. $\mathcal{P}$ and $\mathcal{V}$ define the instance-witness pair as described in Section III-A.
  16. $\mathcal{V}$ checks whether $\quad(\mathbf{y}) \stackrel{?}{=} y$.
  17. $\mathcal{P}$ and $\mathcal{V}$ execute LaBRADOR to prove $(x, w) \in \mathcal{R}$.

Theorem III.1. *Greyhound satisfies evaluation completeness, weak binding and knowledge soundness under the Module-SIS assumption.*

Proof. Greyhound inherently satisfies evaluation completeness, and Lemma II.1 ensures weak binding. For knowledge soundness, we modify the evaluation protocol so that the prover directly outputs $\mathbf{z}$. Then, there exists an extractor that obtains a valid witness with soundness error $n/|\mathcal{C}|$. The extractor either finds a short solution to $[\mathbf{B}|\mathbf{D}]$ or recovers together with $(c_i)_{i\in[n]}$ within the prescribed norm bounds. From these, we reconstruct $\mathbf{f}_i = G_{b_0,m}\vec{\mathbf{s}}_i$ and extract $f \in \mathbb{Z}_q^{<N}[X]$ such that $f(x) = y$. □

IV.

[15] is an interactive PCS for univariate and multilinear polynomials with a lattice assumption. The key idea is similar to Greyhound, but the difference is a dimension and a target polynomial. uses a generalized technique to evaluate both univariate and multilinear polynomials by k-dimensional hypercybe.

Let N be the number of coefficients in a polynomial f. represents polynomial evaluation into a $u_{k-1} \times \cdots u_0$ hypercube where $N = \prod_{i=0}^{k-1} u_i$. In the multilinear case, let ℓ denote the number of evaluation points and we then define $N = 2^\ell$. Consequently, unlike Greyhound, we need k-evaluation vectors $\vec{a}_0, ..., \vec{a}_{k-1}$ where $|a_i| = u_i$ for $i \in [k]$. For a univariate polynomial $f(X) := \sum_{i=0}^{N-1} f_i X^i$, we define the auxiliary vectors $\vec{a}_i = (1, x^{\prod_{j=0}^{i-1} u_j}, x^{2\prod_{j=0}^{i-1} u_j}, ..., x^{(u_i-1)\prod_{j=0}^{i-1} u_j})$. For a multivariate polynomial $f(X_0, ..., X_{\ell-1}) := f_0 + f_1 X_0 + f_2 X_1 + \cdots + f_{N-1} X_0 X_1, ..., X_{\ell-1}$, we define $\vec{a}_i = \bigotimes_{j=\sum_{k=0}^{i} \log u_k - 1}^{\sum_{k=0}^{i-1} \log u_k} (1, x_j)$, and thus we have $\bigotimes_{i=k-1}^{0} \vec{a}_i = \vec{x}$. optimizes the evaluation process introduced in [12]. This process applies to both univariate and multilinear polynomial evaluations, and can be expressed uniformly as

$$y = \begin{bmatrix} f_0 & f_1 & \cdots & f_{u_0-1} \\ f_{u_0} & f_{u_0+1} & \cdots & f_{2u_0-1} \\ \vdots & \vdots & \ddots & \vdots \\ f_{(u_1-1)u_0} & f_{(u_1-1)u_0+1} & \cdots & f_{u_1 u_0-1} \end{bmatrix} \cdot \vec{a}_0 \cdot \vec{a}_1. \quad (2)$$

The difference from Greyhound lies in the order of the coefficients: in Eq. (1), the coefficients of N are arranged by column, whereas in Eq. (2), they are arranged by row. By Eq. (2), we can define a generalized evaluation process for k-dimensional setting as

$$y = (\mathcal{F}(\cdots(\mathcal{F}(\mathcal{F}([F]) \cdot \vec{a}_0) \cdot \vec{a}_1) \cdot \vec{a}_2) \cdots) \cdot \vec{a}_{k-1}. \quad (3)$$

Here, $[F]$ denotes a k-dimensional hypercube of size $u_{k-1} \times u_{k-2} \cdots \times u_0$. For each $i \in [k]$, the function $\mathcal{F}$ maps a $(k-i)$-dimensional hypercube of size $u_{k-1} \times u_{k-2} \cdots \times u_i$ to a $(k-i-1)$-dimensional hypercube matrix of size $(u_{k-1} \times \cdots \times u_{i+1}) \times u_i$, where a hypercube matrix simply refers to flattening a higher-dimensional hypercube into a 2D matrix, with $(u_{k-1} \cdots u_{i+1})$ rows and u_i columns.

A. Building Block for

At the end of , the verifier needs to check the following relation is satisfied:

$$\mathcal{R}_k = \left\{ \begin{matrix} x = (\quad^{(k)}, \quad^{(k)}_{out}, \mathbf{y}^{(k)}, (\vec{a}_j)_{j\in[k]}); \\ w = (\mathbf{s}^{(k)}, (D(\mathbf{s}_i^{(k-1)}, \quad^{(k-1)}_{in,i}) \end{matrix} \right\}.$$

Here, the function D is a flattening function which maps an input hypercube to a one-dimensional vector by sequentially concatenating all its edges.

The prover aims to convince the verifier that it knows a witness corresponding to the coefficients of a polynomial satisfying $y = f(x)$. For efficiency, the proof protocol for the relation $\mathcal{R}_k$ in leverages the Johnson-Lindenstrauss (JL) lemma together with the sum-check protocol. The JL lemma says that a random linear projection of the original vector approximately preserves its ℓ_2-norm. By the JL lemma, the prover just reveals the projected vector instead of the entire vector to convince the verifier. Below is the modular variant of the JL lemma.

Lemma IV.1 (Modular Johnson-Lindenstrauss Variant)**.** *Let $q \in \mathbb{N}$, and let $\mathcal{D}$ be a distribution on $\{-1, 0, 1\}$ such that $\mathcal{D}(-1) = \mathcal{D}(1) = 1/4$ and $\mathcal{D}(0) = 1/2$. For every vector $\vec{a} \in \mathbb{Z}_q^n$ with $\|\vec{a}\| \leq b$ and $b \leq q/125$, we have:*

$$\Pr_{M \leftarrow \mathcal{D}^{256 \times n}}[\|M\vec{a} \bmod q\|^2 < 30b^2] \lessapprox 2^{-128}.$$

Next, the key idea of sum-check protocol lies in the use of randomness and reduction. In each round r, the verifier samples a random challenge $\vec{C}^{(k-r)} \in \mathcal{C}^{u_{k-r-1}}$ and sends it to the prover. Using this challenge, the prover and the verifier reduce the relation $\mathcal{R}_{k-r}$ to $\mathcal{R}_{k-r-1}$, which corresponds to verifying the evaluation up to the $(k-r)$-th dimension described in Eq. (3). Repeating this process for k rounds allows the verifier to ultimately confirm the correctness of the computation.

More concretely, Eq. (3) can be represented as an inner product using conjugation automorphism:

$$y = \quad (\mathbf{y}^{(k)}) = \quad (\langle \quad ^{(k-1)}, \vec{a}_{k-1} \rangle)$$

where

$$^{(k-1)} = \mathcal{F}(\cdots(\mathcal{F}(\mathbf{s}^{(k-1)}) \cdot \sigma_{-1}(M_R(\vec{a}_0))) \cdot \vec{a}_1) \cdots) \cdot \vec{a}_{k-2}.$$

Here, $M_R : \mathbb{Z}_q^{nd} \to \mathbb{R}_q^n$ denotes a mapping function. Similarly, $^{(k-1)}$ can be reduced to $^{(k-2)}$ from the relation

$$^{(k-1)} = \langle \quad ^{(k-2)}, \vec{a}_{k-2} \rangle.$$

Thus, by iterating this process round by round, the prover and verifier progressively reduce the evaluation problem. We omit the complete protocol of the proof protocol for proving $\mathcal{R}_k$.

B. Construction of

Before presenting the detailed construction of , we summarize its parameters in Table II.

Table II
PARAMETERS OF HyperWolf

Notation	Description
q	prime modulus
N	number of coefficients
d	ring dimension
u	length of auxiliary vectors
k	dimension of the coefficient hypercube
κ	height of matrices $\mathbf{A}, \mathbf{B}$
b_0, b_1	decomposition base
δ_0, δ_1	$\delta_0 = \lceil \log_{b_0} q \rceil, \delta_1 = \lceil \log_{b_1} q \rceil$
τ	norm of a challenge
T	operation norm bound of a challenge
$\beta^{(k-r)}$	norm bound of witness in round r
$\mathcal{C}$	challenge space

Now, we describe the details of . The lengths of auxiliary vectors are set uniformly as $u_{k-1} = u_{k-2} = \cdots = u_0 = u$ and we define $N = u^k d = 2^\ell$.

- $(1^\lambda) \to$
  1. $\mathbf{A} \xleftarrow{\$} \mathbb{R}_q^{\kappa \times u\delta_0}$
  2. $\mathbf{B} \xleftarrow{\$} \mathbb{R}_q^{\kappa \times \kappa\delta_1}$
  3. **return** $:= (\mathbf{A}, \mathbf{B})$

- $(\quad, f \in \mathbb{Z}_q^{<N}[X]$ or $\mathbb{Z}_q[X_0, X_1, ..., X_{\ell-1}]) \to (\quad, \quad)$
  1. represent f as $\vec{f} = (f_0, f_1, ..., f_{N-1})$
  2. **for** $i = 0, 1, ..., u^k - 1$:
  3. $\quad \mathbf{f}_i := \sum_{j=0}^{d-1} f_{id+j} X^j \in \mathbb{R}_q$
  4. $\vec{\mathbf{f}}^\mathsf{T} := (\mathbf{f}_0, \mathbf{f}_1, ..., \mathbf{f}_{u^k-1}) \in \mathbb{R}_q^{u^k}$
  5. $\vec{\mathbf{s}} := G^{-1}_{b_0, u^k}(\vec{\mathbf{f}})$
  6. parse $\vec{\mathbf{s}}$ as a k-dimensional hypercube $\mathbf{s}^{(k)}$ and let $(\mathbf{s}_i^{(k-1)})_{i \in [u]}$ be the slices of $\mathbf{s}^{(k)}$ along the k-th dimension.
  7. $\mathbf{A}^{(k)} = \vec{1}^\mathsf{T}_{u^{k-2}} \otimes \mathbf{A} \in \mathbb{R}_q^{\kappa \times u^{k-1} b_0}$
  8. **for** $i = 0, ..., u-1$:
  9. $\quad _{in,i} := \mathbf{A}^{(k)} D(\mathbf{s}_i^{(k-1)})$
  10. $\mathbf{B}^{(k)} = \vec{1}^\mathsf{T}_u \otimes \mathbf{B} \in \mathbb{R}_q^{\kappa \times u\kappa b_1}$
  11. $\quad _{out} := \mathbf{B}^{(k)} G^{-1}_{b_1, u\kappa}((\quad _{in,i})_{i \in [u]})$
  12. $\quad := (\mathbf{s}^{(k)}, (D(\mathbf{s}_i^{(k-1)}), \quad _{in,i})_{i \in [u]})$
  13. **return** $(\quad _{out}, \quad)$

- $(\quad, \quad, f, \quad, (c_i)_{i \in [n]}) \to 0/1$
  1. represent f as $\vec{f} = (f_0, f_1, ..., f_{N-1})$
  2. **for** $i = 0, 1, ..., u^k - 1$:
  3. $\quad \mathbf{f}_i := \sum_{j=0}^{d-1} f_{id+j} X^j \in \mathbb{R}_q$
  4. $\vec{\mathbf{f}}^\mathsf{T} := (\mathbf{f}_0, \mathbf{f}_1, ..., \mathbf{f}_{u^k-1}) \in \mathbb{R}_q^{u^k}$
  5. $\vec{\mathbf{s}} := G^{-1}_{b_0, u^k}(\vec{\mathbf{f}})$
  6. **if** $(D(\mathbf{s}_i^{(k-1)}))_{i \in [u]} \neq D(\mathbf{s}^{(k)}) \vee G_{b_0, u^k}(D(\mathbf{s}^{(k)})) \neq \vec{\mathbf{f}}$:
  7. $\quad$ **return** 0
  8. **for** $i = 0, ..., u-1$:
  9. $\quad$ **if** $\mathbf{A}^{(k)} D(\mathbf{s}_i^{(k-1)}) \neq _{in,i} \vee \mathbf{B}^{(k)} G^{-1}_{b_1, u\kappa}((\quad _{in,i})_{i \in [u]}) \neq \quad _{out}$:
  10. $\quad\quad$ **return** 0
  11. $\quad$ **if** $\|c_i \cdot D(\mathbf{s}_i^{(k-1)})\| \geq \bar{\beta} \vee \|c_i\| \geq \bar{\tau} \vee c_i \notin \mathbb{R}_q^\times$:
  12. $\quad\quad$ **return** 0
  13. **return** 1

- $(\quad, \quad, x$ (or $\vec{x} \in \mathbb{Z}_q^\ell), y; (f, \quad))$
  1. **if** $f(X) \in \mathbb{Z}_q^{<N}[X]$, for $i = 1, ..., k-1$: $\vec{a}_i^\mathsf{T} = (1, x^{u^i}, x^{2u^i}, ..., x^{(u-1)u^i})$ and $\vec{a}_0^\mathsf{T} = (1, x, x^2, ..., x^{ud-1})$
  2. **if** $f(X) \in \mathbb{Z}_q[X_0, X_1, ..., X_{\ell-1}]$, for $i = 1, ..., k-1$: $\vec{a}_i^\mathsf{T} = \bigotimes_{j=(i+1)\log u - 1}^{i \log u}(1, x_j)$ and $\vec{a}_0^\mathsf{T} = \bigotimes_{j=\log u + \log d - 1}^{0}(1, x_j)$
  3. $\vec{a}_0 \leftarrow G_{b_0, ud}(\vec{a}_0)$
  4. $\mathcal{P}$ computes $\mathbf{y} = \langle \quad ^{(k)}, \vec{a}_{k-1} \rangle$.
  5. $\mathcal{P}$ and $\mathcal{V}$ define the instance-witness pair as described in Section IV-A.
  6. $\mathcal{P}$ and $\mathcal{V}$ execute the protocol to prove $(x, w) \in \mathcal{R}_k$.
  7. $\mathcal{V}$ checks whether $\quad (\mathbf{y}) \stackrel{?}{=} y$.

Theorem IV.2. *satisfies evaluation completeness, weak binding and knowledge soundness under the Module-SIS assumption.*

Proof. satisfies evaluation completeness naturally via its construction. For each round $r \in [k-1]$, given $(u+1)$ accepting transcripts, one can compute a

weak opening and verify the inner/outer commitment binding, norm bound $\beta^{(k-r)}$, and folded inner product relation $\langle \quad^{(k-r)}, \vec{a}_{k-r-1} \rangle \stackrel{?}{=} \mathbf{y}^{(k)}$ to extract a valid witness $w_r \in \mathcal{R}_{k-r}(x_r)$. The probability that the extracted witness violates norm bound each round is at most $2^{-128} + q^{-d/2} \leq 2^{-127}$. Therefore, the extractor can extract a valid witness with a soundness error $\epsilon = 2^{-126}$ and a norm bound $\sqrt{128/30}$.

Also, applying the union bound over $(k-1)$ rounds, the total knowledge soundness error is bounded by $\frac{(k-1)u}{|\mathcal{C}|} + 2^{-126}(k-1)$. □

V. Comparison

Both Greyhound [14] and [15] are lattice-based polynomial commitment schemes constructed under the Module-SIS assumption, providing post-quantum security and additive homomorphism with transparent setup. Despite using the same lattice assumption, their target polynomial classes and evaluation, and proof techniques differ substantially.

The most notable difference lies in the evaluation technique. Both PCSs use the folding technique for polynomial evaluation: Greyhound targets univariate polynomials using a coefficient matrix, whereas generalizes to both univariate and multilinear polynomials via the k-dimensional hypercube.

Table III
Comparison of Greyhound and HyperWolf

Scheme	commit. size	Proof size	Prover Cost	Verifier Cost
Greyhound	$O(1)$	$O(\log \log N)$	$O(N)$	$O(\sqrt{N})$
HyperWolf	$O(1)$	$O(\log N)$	$O(N)$	$O(\log N)$

In terms of complexity, both PCSs achieve a constant commitment size $O(1)$ and the same asymptotic prover cost $O(N)$, while differing in the proof size and verifier cost. The proof size of Greyhound is $O(\log \log N)$, which is smaller than HyperWolf's $O(\log N)$, but the difference is not very significant. In contrast, the verifier cost shows a large gap: $O(\log N)$ for HyperWolf and $O(\sqrt{N})$ for Greyhound, which is due to the difference in dimensions. The overall comparison is summarized in Table III.

VI. Conclusion

In this survey, we review two recent lattice-based PCSs, Greyhound and , where both are based on Module-SIS assumption but use a different technique. While the Greyhound achieves a smaller proof size via a coefficient matrix and LaBRADOR proofs for a univariate polynomial, achieves a lower verifier cost via the k-dimensional coefficient hypercube for both univariate and multilinear polynomials.

There are several directions for future work that we would like to pursue. Previous schemes have mainly focused on univariate or multilinear polynomials. A natural next step is to construct a lattice-based PCS that supports general multivariate polynomials. In addition, we aim to investigate whether alternative lattice assumptions beyond Module-SIS can lead to more efficient or versatile PCSs. Finally, we are interested in designing a fully homomorphic PCS, which would significantly broaden the applicability of PCS in cryptographic protocols.

References

[1] M. Backes, A. Datta, and A. Kate, "Asynchronous computational VSS with reduced communication complexity," in *CT-RSA 2013*, ser. LNCS, E. Dawson, Ed., vol. 7779. Springer, 2013, pp. 259–276.

[2] R. Bhadauria, C. Hazay, M. Venkitasubramaniam, W. Wu, and Y. Zhang, "Private polynomial commitments and applications to MPC," in *PKC 2023, Part II*, ser. LNCS, A. Boldyreva and V. Kolesnikov, Eds., vol. 13941. Springer, 2023, pp. 127–158.

[3] B. Bünz, B. Fisch, and A. Szepieniec, "Transparent snarks from DARK compilers," in *EUROCRYPT 2020, Part I*, ser. LNCS, A. Canteaut and Y. Ishai, Eds., vol. 12105. Springer, 2020, pp. 677–706.

[4] A. Chiesa, Y. Hu, M. Maller, P. Mishra, P. Vesely, and N. P. Ward, "Marlin: Preprocessing zksnarks with universal and updatable SRS," in *EUROCRYPT 2020, Part I*, ser. LNCS, A. Canteaut and Y. Ishai, Eds., vol. 12105. Springer, 2020, pp. 738–768.

[5] R. S. Wahby, I. Tzialla, A. Shelat, J. Thaler, and M. Walfish, "Doubly-efficient zksnarks without trusted setup," in *2018 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2018, pp. 926–943.

[6] A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," in *ASIACRYPT 2010*, ser. LNCS, M. Abe, Ed., vol. 6477. Springer, 2010, pp. 177–194.

[7] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *2018 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2018, pp. 315–334.

[8] Y. Zhang, D. Genkin, J. Katz, D. Papadopoulos, and C. Papamanthou, "vsql: Verifying arbitrary SQL queries over dynamic outsourced databases," in *2017 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2017, pp. 863–880.

[9] ——, "A zero-knowledge version of vsql," *IACR Cryptol. ePrint Arch.*, p. 1146, 2017.

[10] J. Lee, "Dory: Efficient, transparent arguments for generalised inner products and polynomial commitments," in *TCC 2021, Part II*, ser. LNCS, K. Nissim and B. Waters, Eds., vol. 13043. Springer, 2021, pp. 1–34.

[11] T. Xie, Y. Zhang, and D. Song, "Orion: Zero knowledge proof with linear prover time," in *CRYPTO 2022, Part IV*, ser. LNCS, Y. Dodis and T. Shrimpton, Eds., vol. 13510. Springer, 2022, pp. 299–328.

[12] A. Golovnev, J. Lee, S. T. V. Setty, J. Thaler, and R. S. Wahby, "Brakedown: Linear-time and post-quantum snarks for R1CS," *IACR Cryptol. ePrint Arch.*, p. 1043, 2021.

[13] G. Fenzi, H. Moghaddas, and N. K. Nguyen, "Lattice-based polynomial commitments: Towards asymptotic and concrete efficiency," *J. Cryptol.*, vol. 37, no. 3, p. 31, 2024.

[14] N. K. Nguyen and G. Seiler, "Greyhound: Fast polynomial commitments from lattices," in *CRYPTO 2024, Part X*, ser. LNCS, L. Reyzin and D. Stebila, Eds., vol. 14929. Springer, 2024, pp. 243–275.

[15] L. Zhang, S. Gao, and B. Xiao, "Hyperwolf: Efficient polynomial commitment schemes from lattices," *IACR Cryptol. ePrint Arch.*, p. 922, 2025.

[16] V. Cini, G. Malavolta, N. K. Nguyen, and H. Wee, "Polynomial commitments from lattices: Post-quantum security, fast verification and transparent setup," in *CRYPTO 2024, Part X*, ser. LNCS, L. Reyzin and D. Stebila, Eds., vol. 14929. Springer, 2024, pp. 207–242.

[17] A. Langlois and D. Stehlé, "Worst-case to average-case reductions for module lattices," *Des. Codes Cryptogr.*, vol. 75, no. 3, pp. 565–599, 2015.

[18] W. Beullens and G. Seiler, "Labrador: Compact proofs for R1CS from module-sis," in *CRYPTO 2023, Part V*, ser. LNCS, H. Handschuh and A. Lysyanskaya, Eds., vol. 14085. Springer, 2023, pp. 518–548.

[19] M. R. Albrecht, G. Fenzi, O. Lapiha, and N. K. Nguyen, "SLAP: succinct lattice-based polynomial commitments from standard assumptions," in *EUROCRYPT 2024, Part VI*, ser. LNCS, M. Joye and G. Leander, Eds., vol. 14656. Springer, 2024, pp. 90–119.

[20] V. Lyubashevsky, N. K. Nguyen, and M. Plançon, "Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general," in *CRYPTO 2022, Part II*, ser. LNCS, Y. Dodis and T. Shrimpton, Eds., vol. 13508. Springer, 2022, pp. 71–101.