

Note on Extractability of the PST Polynomial Commitment Scheme

Janno Siim  and Pritam Pal 

University of Tartu, Tartu, Estonia

Abstract. A recent work by Belohorec et al. (Crypto, 2025) shows that the well-known PST multivariate polynomial commitment scheme is black-box extractable under falsifiable assumptions. They show that a minimally modified (extended) PST is extractable under an assumption $\text{ARSDH}(n)$, and that the canonical PST is extractable under an assumption $\text{GARSDH}(n)$. Both of these assumptions are new and more specialized than the original ARSDH assumption proposed by Lipmaa et al. (Eurocrypt, 2024) to prove black-box extractability of the univariate KZG polynomial commitment. A natural question is whether these assumptions are actually stronger than the original ARSDH assumption. We answer this negatively: we show that both $\text{ARSDH}(n)$ and $\text{GARSDH}(n)$ are equivalent to the original ARSDH assumption. Secondly, we point out a gap in the proof that canonical PST is extractable under $\text{GARSDH}(n)$ assumption.

Keywords: Polynomial commitment schemes · Falsifiable assumptions · PST commitment scheme · ARSDH assumption

1 Introduction

A polynomial commitment scheme [KZG10] allows one to commit to a polynomial f succinctly and to later open it to $f(\alpha)$ for an arbitrary evaluation point α . The opening phase includes a proof π that convinces the verifier that the evaluation is indeed correct. Polynomial commitment schemes have turned out to be crucial for constructing efficient SNARKs [CHM⁺20, GWC19, CBBZ23, ZCF24], but have also found use in other important applications such as Data Availability Sampling (DAS) [HASW24] and Verkle trees [Kus18] that are now part of Ethereum ecosystem.

In case of SNARKs, it is important that the polynomial commitment scheme is extractable, i.e., in the security proof, it must be possible to efficiently extract the committed polynomial f from a commitment C and a valid opening proof π . Some polynomial commitment schemes have an interactive opening phase, which makes it possible to extract the polynomial f using rewinding techniques. Polynomial commitment schemes with a non-interactive opening phase, such as the KZG polynomial commitment scheme [KZG10] and the (canonical) PST commitment scheme [PST13], were until recently only known to be extractable under non-falsifiable assumptions¹ or in idealized group models [CHM⁺20]. Both of these have been widely criticised [GK16], and there are known (although contrived) protocols secure in idealized group models that are insecure in the real world [Den02, Zha22].

A breakthrough was achieved by Lipmaa, Parisella, and Siim [LPS24] who showed that the KZG polynomial commitment scheme [KZG10] is computationally special sound (which

E-mail: jannosiim@gmail.com (Janno Siim), iampritampal23@gmail.com (Pritam Pal)

¹Recall, an assumption is falsifiable if it can be presented as an interactive game between an efficient challenger and an adversary.



is a specific flavor of extractability) under a falsifiable assumption, the Adaptive Rational Strong Diffie-Hellman (ARSDH, see [Definition 1](#)) assumption. Their main technique was to use the evaluation point α for rewinding. ARSDH is a minor generalization of the RSDH assumption [GR19] and the more established Strong Diffie-Hellman (SDH) assumption [BB04]. Since then, ARSDH has been used to prove the security of various pairing-based protocols [LPS25, CGKY26, Lip25, WZ25, CADLT25, Sii26, XTP⁺26].

A recent work by Belohorec, Dvůrák, Hoffmann, Hubáček, Masková, and Pastýřík [BDH⁺25] further extends the result of [LPS24] to the PST polynomial commitment scheme [PST13]. Whereas KZG allows commitments to univariate polynomials, PST is a natural generalization of KZG that allows commitments to multivariate polynomials. Multivariate polynomial commitment schemes (and in particular, multilinear polynomial commitment schemes) have recently come to prominence since it is known how to achieve a linear-time SNARK prover using them [CBBZ23].

The work of Belohorec et al. [BDH⁺25] studies two quite similar flavors of PST. The first one, which they call *canonical PST*, is what is usually considered to be the PST polynomial commitment scheme, and it has a non-interactive opening proof. The second one they call the *extended PST*, and it has an interactive opening proof. Effectively, they are the same protocol, but in the extended PST, elements of the canonical opening proof are sent round by round.

To prove the extractability of the extended PST, they propose a new ARSDH(n) assumption (see [Definition 2](#)), where n corresponds to the number of variables in the committed polynomial. This is a multivariate generalization of the original ARSDH assumption, where $\text{ARSDH}(1) \equiv \text{ARSDH}$. To prove the extractability of the canonical PST, they propose the Generalized ARSDH(n) assumption ($\text{GARSDH}(n)$, see [Definition 3](#)), which is a more complicated variant of ARSDH(n). Both ARSDH(n) and $\text{GARSDH}(n)$ are falsifiable, but significantly more specialized than the original ARSDH assumption proposed by Lipmaa et al. [LPS24]. Belohorec et al. [BDH⁺25] provide analysis of both assumptions in the Algebraic Group Model (AGM) [FKL18] and in the Generic Group Model (GGM) [Sho97] to justify the validity of these assumptions. Subsequently ARSDH(n) also appears in [BHS26, BHKM26].

1.1 Our contribution

Our short note consists of two contributions, which clarify the extractability of the PST polynomial commitment scheme.

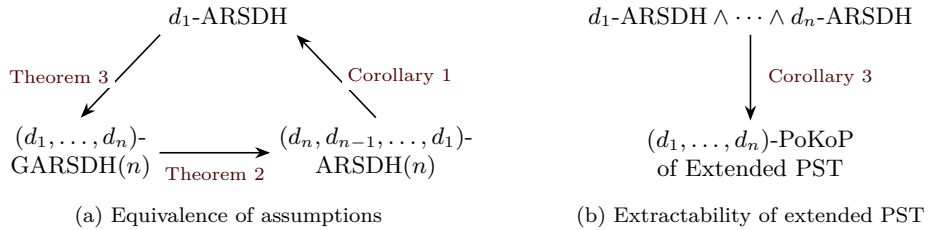


Figure 1: Overview of our results, for $\prod_{i=1}^n (d_i + 1) \in \text{poly}(\lambda)$. Arrows denote implications between assumptions or security properties.

Contribution 1: Reductions to ARSDH. Surprisingly, we show that both ARSDH(n) and $\text{GARSDH}(n)$ are equivalent to the original ARSDH assumption (even if $n > 1$). In fact, if there exist an adversary $\mathcal{A}$ that breaks the ARSDH(n) (resp. $\text{GARSDH}(n)$) assumption with probability ε and with the running time t , then one can construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ as a subroutine and breaks ARSDH with the same probability

ε and $\mathcal{B}$'s computation is dominated by t and $\mathcal{O}(d^n)$ scalar multiplications, where n is the number of variables and d is the degree bound for individual variables. In the case of the PST polynomial commitment scheme, d^n must always have polynomial size since the commitment key also contains d^n group elements. Thus, the reductions are efficient.

As we will see, the reductions are rather straightforward. Both reductions rely on the observation that the reduction can pick trapdoors for the first $n - 1$ variables locally and then use the adversary $\mathcal{A}$ to break the ARSDH assumption for the last variable. Figure 1 summarizes our reductions and their consequence for the extended PST.

Contribution 2: Flaw in the Security Proof. By reducing ARSDH(n) to standard ARSDH, we can immediately conclude that the extended PST is extractable under the original ARSDH assumption.

However, unfortunately, the same is not true for the canonical PST. We point out two gaps in the proof of Theorem 5 in [BDH⁺25] that canonical PST is extractable under GARSDH(n). To the best of our knowledge, there is no simple way to recover the result. One can still prove the extractability of the canonical PST under a falsifiable assumption, but the only suitable assumption we know is essentially tautological: canonical PST is special sound.

2 Preliminaries

We denote the security parameter by λ . PPT stands for probabilistic polynomial time. For a set A , we write $a \leftarrow \$ A$ to denote that a is chosen uniformly at random from A . We write $f(\lambda) \approx_\lambda 0$ to denote that a function f is negligible in λ . We write $f(\lambda) = \text{poly}(\lambda)$ to denote that a function f is asymptotically bounded by polynomial. For integers $a \leq b$, we write $[a..b] := \{a, a + 1, \dots, b\}$. We denote by $\mathbb{Z}_p[X_1, \dots, X_n]$ the ring of polynomials in n variables with coefficients in $\mathbb{Z}_p$. For a finite set $\mathcal{S}$, we denote the vanishing polynomial by $Z_{\mathcal{S}}(X) := \prod_{s \in \mathcal{S}} (X - s)$.

2.1 Bilinear Groups

Let Pgen be a bilinear group generator that takes as input the security parameter 1^λ and outputs $\text{gp} = (p, \bullet, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_T)$, where p is a prime, $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ are additive groups of order p with generators $\mathcal{P}_1, \mathcal{P}_2$, and $\mathcal{P}_T$, respectively, and $\bullet : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a non-degenerate bilinear map. Namely, for all $a, b \in \mathbb{Z}_p$, we have $(a\mathcal{P}_1) \bullet (b\mathcal{P}_2) = (ab)\mathcal{P}_T$. In this work, we consider asymmetric bilinear groups, i.e., $\mathbb{G}_1 \neq \mathbb{G}_2$, and there is no efficient isomorphism between them.

We will use the notation $[a]_1 := a \cdot \mathcal{P}_1$ for $a \in \mathbb{Z}_p$, and similarly for $[a]_2$ and $[a]_T$. We assume that all algorithms get the group parameters gp as an input without explicitly mentioning it.

2.2 ARSDH Family of Assumption

We first recall the original d -ARSDH assumption from [LPS24].

Definition 1 (d -ARSDH). Let Pgen be a bilinear group generator and $d \in \text{poly}(\lambda)$. We say that the d -Adaptive Rational Strong Diffie-Hellman (ARSDH) assumption holds with respect to bilinear group generator Pgen if for any PPT $\mathcal{A}$,

$$\Pr \left[\begin{array}{l} |S| = d + 1 \wedge \\ [\psi]_T \neq [0]_T \wedge \\ [\chi]_1 \bullet [1]_2 = [\psi]_1 \bullet [Z_S(t)]_2 \end{array} : \begin{array}{l} \text{gp} \leftarrow \text{Pgen}(1^\lambda), t \leftarrow \$ \mathbb{Z}_p, \\ \text{ck} \leftarrow (\{[t^i]_1\}_{i=0}^d, [1, t]_2) \\ ([\chi]_1, [\psi]_1, \mathcal{S} \subseteq \mathbb{Z}_p) \leftarrow \mathcal{A}(\text{ck}) \end{array} \right] \approx_\lambda 0,$$

where $Z_S(X) := \prod_{s \in \mathcal{S}} (X - s)$.

$\text{Game}_{\text{Pgen}, \mathcal{A}, d_1, \dots, d_n}^{\text{ARSDH}(n)}(\lambda)$ $\begin{aligned} &\text{gp} \leftarrow \text{Pgen}(1^\lambda), t_1, \dots, t_n \leftarrow \$\mathbb{Z}_p; \\ &\text{ck} \leftarrow (\{[t_1^{i_1} \dots t_n^{i_n}]_1\}_{i_1, \dots, i_n=0}^{d_1, \dots, d_n}, \{[t_i]_2\}_{i=1}^n); \\ &([\chi]_1, [\psi]_1, \{[\varphi_i]_1, \alpha_i\}_{i=1}^{n-1}, \mathcal{S}) \leftarrow \mathcal{A}(\text{ck}); \\ &\quad // \text{ winning conditions for } \mathcal{A}: \\ &b_1 \leftarrow \mathcal{S} \subseteq \mathbb{Z}_p \wedge \mathcal{S} = d_n + 1; \\ &b_2 \leftarrow [\chi]_T \neq \sum_{i=1}^{n-1} [\varphi_i]_1 \bullet [t_i - \alpha_i]_2; \\ &b_3 \leftarrow [\chi]_1 \bullet [1]_2 = \sum_{i=1}^{n-1} [\varphi_i]_1 \bullet [t_i - \alpha_i]_2 + [\psi]_1 \bullet [Z_{\mathcal{S}}(t_n)]_2; \\ &\text{return } b_1 \wedge b_2 \wedge b_3; \end{aligned}$

Figure 2: The game for $(d_1, \dots, d_n)$ -ARSDH(n) assumption.

If the set $\mathcal{S}$ is fixed in advance (the assumption is non-adaptive), then one obtains the d -RSDH assumption [GR19]. If $\chi = Z_{\mathcal{S} \setminus \{a\}}(t)$ for some $a \in \mathcal{S}$, then $\psi = \chi / Z_{\mathcal{S}}(t) = 1/(t - a)$, which is the regular SDH assumption [BB04].

2.2.1 Generalized ARSDH Assumptions

Recent work [BDH⁺25] proposes two n -variate generalizations of the original ARSDH assumption: ARSDH(n) and GARSDH(n) assumptions. We recall them in the following.

Definition 2 ($(d_1, \dots, d_n)$ -ARSDH(n)). Let Pgen be a bilinear group generator and $d_1, \dots, d_n \geq 0$ such that $\prod_{i=1}^n (d_i + 1) \in \text{poly}(\lambda)$ ². We say that the $(d_1, \dots, d_n)$ -ARSDH(n) assumption holds with respect to bilinear group generator Pgen if for any PPT $\mathcal{A}$,

$$\Pr \left[\text{Game}_{\text{Pgen}, \mathcal{A}, d_1, \dots, d_n}^{\text{ARSDH}(n)}(\lambda) \right] \approx_\lambda 0,$$

where $\text{Game}_{\text{Pgen}, \mathcal{A}, d_1, \dots, d_n}^{\text{ARSDH}(n)}(\lambda)$ is defined in Figure 2.

It is easy to see that by setting $n = 1$, $(d_1, \dots, d_n)$ -ARSDH(n) collapses to the original d_1 -ARSDH assumption. The second assumption in [BDH⁺25] is the GARSDH(n) assumption and it is defined as follows.

Definition 3 (GARSDH(n)). Let Pgen be a bilinear group generator and $d_1, \dots, d_n \geq 0$ such that $\prod_{i=1}^n (d_i + 1) \in \text{poly}(\lambda)$. Then we say that the $(d_1, \dots, d_n)$ -Generalized ARSDH(n) assumption holds for Pgen , if for all PPT adversary $\mathcal{A}$,

$$\Pr \left[\text{Game}_{\text{Pgen}, \mathcal{A}, d_1, \dots, d_n}^{\text{GARSDH}(n)}(\lambda) \right] \approx_\lambda 0,$$

where $\text{Game}_{\text{Pgen}, \mathcal{A}, d_1, \dots, d_n}^{\text{GARSDH}(n)}(\lambda)$ is defined in Figure 3.

Remark 1. We follow similar notation in Figure 3 as [BDH⁺25], but let us slightly clarify it in the case when $j = 1$. When $j = 1$, the set $\mathcal{S}_j^{(i_1, \dots, i_{j-1})}$ is simply denoted as $\mathcal{S}_1 := \mathcal{S}_1^{()}$ and the property $|\mathcal{S}_1| = d_1 + 1$ is checked in the condition b_1 on the line 8. (This might not be immediately clear looking at the formula for b_1 .) Similarly, we denote $\varphi_1 := \varphi_1^{()}$. Finally, when $j = 1$, the sum $\sum_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}} [\varphi_j^{(i_1, \dots, i_{j-1})}]_1 \bullet [Z_{\mathcal{S}_j^{(i_1, \dots, i_{j-1})}}(t_j)]_2$ (see the line 10 in the condition b_3) should be interpreted as $[\varphi_1]_1 \bullet [Z_{\mathcal{S}_1}(t_1)]_2$. Understanding these subtleties is important in the proof of the reduction Theorem 3.

²Note that [BDH⁺25] only requires that $d_1, \dots, d_n \in \text{poly}(\lambda)$ while not specifying a constraint for n . We constrain the product $\prod_{i=1}^n (d_i + 1)$ as it determines the size of ck . We do the same in Definition 3.

$\text{Game}_{\text{Pgen}, \mathcal{A}, d_1, \dots, d_n}^{\text{GARSDH}(n)}(\lambda)$	
1 :	$\text{gp} \leftarrow \text{Pgen}(1^\lambda), t_1, \dots, t_n \leftarrow \mathbb{Z}_p;$
2 :	$\text{ck} \leftarrow (\{[t_1^{i_1} \dots t_n^{i_n}]_1\}_{i_1, \dots, i_n=0}^{d_1, \dots, d_n}, \{[t_i]_2\}_{i=1}^n);$
3 :	$\mathcal{A}(\text{ck})$ outputs:
4 :	(a) $[\chi]_1,$
5 :	(b) for all $j \in [1, n], \{\mathcal{S}_j^{(i_1, \dots, i_{j-1})}\}_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}}$ where $\mathcal{S}_j^{(i_1, \dots, i_{j-1})} \subseteq \mathbb{Z}_p,$
6 :	(c) for all $j \in [1, n], \{[\varphi_j^{(i_1, \dots, i_{j-1})}]_1\}_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}};$
7 :	// winning conditions for $\mathcal{A}$:
8 :	$b_1 \leftarrow \bigwedge_{j=1}^n \bigwedge_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}} (\mathcal{S}_j^{(i_1, \dots, i_{j-1})} = d_j + 1);$
9 :	$b_2 \leftarrow [\chi]_T \neq \sum_{j=2}^n \sum_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}} [\varphi_j^{(i_1, \dots, i_{j-1})}]_1 \bullet [Z_{\mathcal{S}_j^{(i_1, \dots, i_{j-1})}}(t_j)]_2;$
10 :	$b_3 \leftarrow [\chi]_T = \sum_{j=1}^n \sum_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}} [\varphi_j^{(i_1, \dots, i_{j-1})}]_1 \bullet [Z_{\mathcal{S}_j^{(i_1, \dots, i_{j-1})}}(t_j)]_2;$
11 :	return $b_1 \wedge b_2 \wedge b_3;$

Figure 3: The game for $(d_1, \dots, d_n)$ -GARSDH(n) assumption.

Knowing the above notational conventions, it is straightforward to check that GARSDH(1) is equivalent to ARSDH assumption. The test on the line 8 only checks that $|\mathcal{S}_1| = d_1 + 1$, the test on the line 9 checks that $[\chi]_T \neq [0]_T$ (the sum is empty), and the test on the line 10 checks that $[\chi]_T = [\varphi_1]_1 \bullet [Z_{\mathcal{S}_1}(t_1)]_2$.

3 Overview of PST

We give a brief overview of the PST polynomial commitment scheme [PST13] and the result of Belohorec et al. [BDH⁺25] to set the stage for our contribution. Readers familiar with both works can safely skip to the next section.

We discuss the two variants of multivariate PST polynomial commitment scheme as described in [BDH⁺25]. Both are a generalization of the univariate KZG polynomial commitment scheme [KZG10] to multivariate polynomials. The canonical PST is the variant with a non-interactive opening phase and the extended PST is the variant with an interactive opening phase.

The commitment key generation and commitment algorithms are the same for both variants. PST allows commitments to n -variate polynomials $f(X_1, \dots, X_n)$ where the individual degrees of each variable is bounded by parameters $\vec{d} = (d_1, d_2, \dots, d_n)$ respectively. The commitment key generation algorithm PST.KGen takes as input the degree bounds and the pairing description. It then samples n trapdoor $t_1, \dots, t_n \leftarrow \mathbb{Z}_p$ and computes $[M(t_1, \dots, t_n)]_1$ for each n -variate monomial $M(X_1, \dots, X_n) = X_1^{i_1} \dots X_n^{i_n}$, where $0 \leq i_j \leq d_j$ for all $j \in [1..n]$. Eventually it outputs a commitment key ck which contains the pairing description gp , the group elements $[M(t_1, \dots, t_n)]_1$ for all n -variate monomials $M(X_1, \dots, X_n)$ and the group elements $[t_i]_2$ for all $i \in [1..n]$. The commitment algorithm PST.Com takes as an input the commitment key ck and a polynomial $f(X_1, \dots, X_n)$ and outputs the commitment $\text{Com} = [f(t_1, \dots, t_n)]_1$, which is a linear combination of the $\mathbb{G}_1$ group elements in ck . See Figure 4 for the detailed description of PST.KGen and PST.Com.

3.1 Canonical PST

The opening phase of canonical PST [PST13] uses the following polynomial decomposition lemma. For any $f(X_1, \dots, X_n) \in \mathbb{Z}_p[X_1, \dots, X_n]$ and any $(x_1, \dots, x_n) \in \mathbb{Z}_p^n$, it holds that

$$f(X_1, \dots, X_n) = \sum_{i=1}^n q_i(X_i, \dots, X_n)(X_i - x_i) + f(x_1, \dots, x_n), \text{ where}$$

$$q_i(X_i, \dots, X_n) = \frac{f(x_1, \dots, x_{i-1}, X_i, \dots, X_n) - f(x_1, \dots, x_i, X_{i+1}, \dots, X_n)}{(X_i - x_i)}.$$

Note that $q_i(X_i, \dots, X_n)$ is a polynomial in $\mathbb{Z}_p[X_1, \dots, X_n]$ for all $1 \leq i \leq n$. The main intuition behind the PST security is that if the identity $f(X_1, \dots, X_n) = \sum_{i=1}^n q_i(X_i, \dots, X_n)(X_i - x_i) + z$ holds, then it immediately follows that $f(x_1, \dots, x_n) = z$.

In the opening phase the prover receives the evaluation point $(x_1, \dots, x_n)$ and computes the value $z = f(x_1, \dots, x_n)$ and the opening proof $\pi = (\pi_1, \dots, \pi_n)$, where $\pi_i = [q_i(t_i, \dots, t_n)]_1$. The verifier uses pairings to check that the decomposition lemma's identity holds on the commitments: $(\text{Com} - [z]_1) \bullet [1]_2 = \sum_{i=1}^n \pi_i \bullet [t_i - x_i]_2$. See Figure 4.

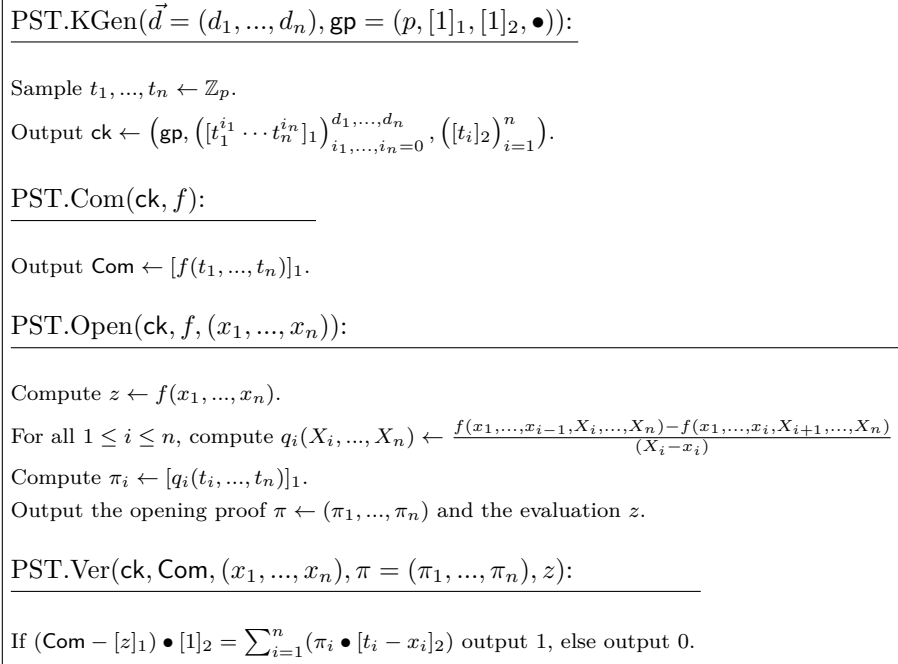


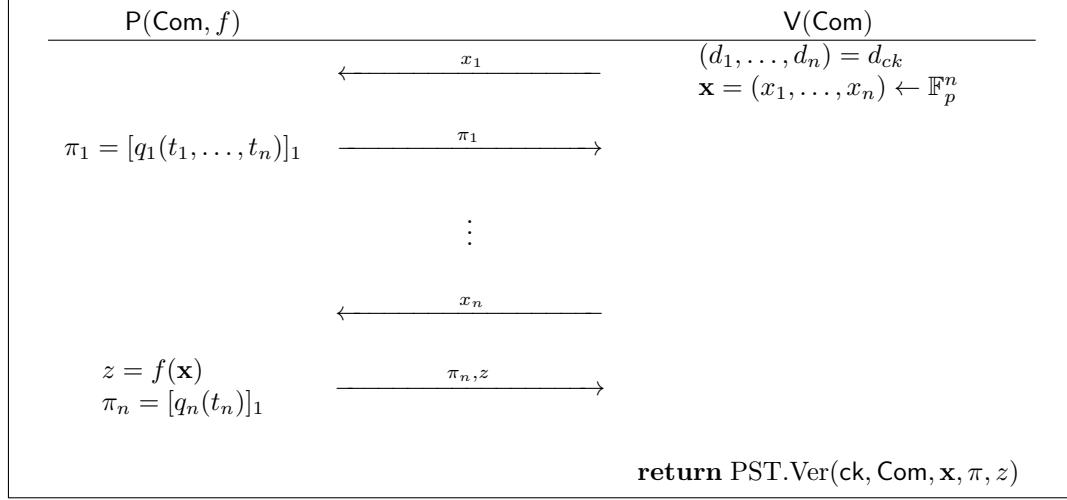
Figure 4: Canonical PST PCS

3.2 Extended PST

The extended PST is an interactive argument $(\mathbf{P}, \mathbf{V})$ for the relation

$$\mathcal{R}_{\text{ck}}^{\text{PST}} = \left\{ (C, f) : C = \text{PST.Com}(\text{ck}, f) \wedge \deg(f) \leq \vec{d} \right\}.$$

It is based on the observation that each quotient polynomial $q_i(X_i, \dots, X_n)$ depends only on the first i coordinates of the evaluation point $\mathbf{x} = (x_1, \dots, x_n)$. PST.KGen, PST.Com, and PST.Ver are identical to the canonical PST. Now, the interactive opening algorithm

**Figure 5:** Extended PST opening protocol

is a n -round protocol. In i -th round P receives i -th coordinate x_i from the verifier and replies with $\pi_i = [q_i(t_i, \dots, t_n)]_1$ where

$$q_i(X_i, \dots, X_n) = \frac{f(x_1, \dots, x_{i-1}, X_i, \dots, X_n) - f(x_1, \dots, x_i, X_{i+1}, \dots, X_n)}{(X_i - x_i)}.$$

In n -th round P receives x_n and sends $\pi_n = [q_n(t_n)]_1$ and the evaluation of $f(X_1, \dots, X_n)$ at $(x_1, \dots, x_n)$ i.e. $z = f(x_1, \dots, x_n)$. In the end, the verifier receives the same proof elements as in the canonical PST and performs the same verification. See Figure 5 for the detailed description of the extended PST opening protocol.

3.3 Results of Belohorec et al.

We briefly summarize the results of Belohorec et al. [BDH⁺25] on the extractability of PST. They define an extractability notion called proof of knowledge of a polynomial (PoKoP). In short, a polynomial commitment scheme is a PoKoP if the opening phase is a proof of knowledge of the committed polynomial f . In particular, the evaluation points $x_1, \dots, x_n$ are used for rewinding by an expected polynomial-time extractor **Ext** to extract the committed polynomial f from a commitment **Com** and a valid opening proof π . We refer the reader to [BDH⁺25] for the precise definition of PoKoP. The authors proved that both versions of PST are proofs of knowledge of a polynomial (PoKoP) under the ARSDH(n) and GARSDH(n) assumptions, respectively. We state their main results below.

Theorem 1 ([BDH⁺25], Theorem 5). *For $n \in \mathbb{N}$, let $\vec{d} = (d_1, \dots, d_n) \in \text{poly}(\lambda)$ be a vector of degree bounds and **Pgen** be a bilinear group generator.*

1. *If the ARSDH(n) assumption holds for **Pgen** then the extended PST is a PoKoP with respect to **Pgen** and $\vec{d}$.*
2. *If the GARSDH(n) assumption holds for **Pgen** then both the extended and canonical PST are PoKoPs with respect to **Pgen** and $\vec{d}$.*

In Section 5 we will point out gaps in the proof of Item 2 of Theorem 1. To the best of our knowledge, that result does not hold and there is no simple way to recover it.

4 Reductions

We start by slightly reformulating the d -ARSDH assumption. It is easy to see that d -ARSDH assumption (see [Definition 1](#)) is equivalent to the following variant, where the condition $[\psi]_T \neq [0]_T$ is replaced by $[\chi]_T \neq [0]_T$.

Definition 4 (d -ARSDH*). Let $d \in \text{poly}(\lambda)$. We say that the d -ARSDH* assumption holds with respect to bilinear group generator Pgen if for any PPT $\mathcal{A}$,

$$\Pr \left[\begin{array}{l} |\mathcal{S}| = d + 1 \wedge \\ [\chi]_T \neq [0]_T \wedge \\ [\chi]_1 \bullet [1]_2 = [\psi]_1 \bullet [Z_{\mathcal{S}}(t)]_2 \end{array} : \begin{array}{l} \text{gp} \leftarrow \text{Pgen}(1^\lambda), t \leftarrow \$_{\mathbb{Z}_p}, \\ \text{ck} \leftarrow (\{[t^i]_1\}_{i=0}^d, [1, t]_2) \\ ([\chi]_1, [\psi]_1, \mathcal{S} \subseteq \mathbb{Z}_p) \leftarrow \mathcal{A}(\text{ck}) \end{array} \right] \approx_\lambda 0.$$

Lemma 1. *The d -ARSDH and d -ARSDH* assumptions are equivalent.*

Proof. **ARSDH** $\Rightarrow$ **ARSDH***. Suppose $\mathcal{A}$ outputs $([\chi]_1, [\psi]_1, \mathcal{S})$ that breaks the d -ARSDH* assumption. Then, $\chi \neq 0$, which implies that $\psi \neq 0$. Thus, $\mathcal{A}$ also breaks d -ARSDH assumption.

ARSDH* $\Rightarrow$ **ARSDH**. Suppose $\mathcal{A}$ outputs $([\chi]_1, [\psi]_1, \mathcal{S})$ that breaks the d -ARSDH assumption. The same $\mathcal{A}$ will also break d -ARSDH* assumption unless $\chi = 0$. If $\chi = 0$, but $\psi \neq 0$, then $\psi Z_{\mathcal{S}}(t) = 0$ implies $Z_{\mathcal{S}}(t) = 0$. This is only possible if the trapdoor $t \in \mathcal{S}$. We can then construct an adversary $\mathcal{B}$ that breaks the d -ARSDH* assumption as follows.

- $\mathcal{B}$ gets ck as an input and runs $\mathcal{A}(\text{ck})$ that outputs $([\chi]_1, [\psi]_1, \mathcal{S})$.
- If $\chi \neq 0$, $\mathcal{B}$ outputs $([\chi]_1, [\psi]_1, \mathcal{S})$.
- Otherwise, $\mathcal{B}$ recovers the trapdoor t from the set $\mathcal{S}$. It picks any set $\mathcal{S}' \subseteq \mathbb{Z}_p$ of size $d + 1$ such that $t \notin \mathcal{S}'$ and outputs $(Z_{\mathcal{S}'}(t) \cdot [1]_1, [1]_1, \mathcal{S}')$.

It is easy to confirm that $\mathcal{B}$ breaks d -ARSDH* assumption with the same probability as $\mathcal{A}$ breaks d -ARSDH assumption. $\square$

Since the reductions between ARSDH and ARSDH* are tight, in the following we will use the ARSDH* assumption as the definition of ARSDH assumption.

We also recall the following (and slightly paraphrased) result from [\[BDH⁺25, Lemma 4\]](#).

Lemma 2 ([\[BDH⁺25\]](#)). *Let $d_1, \dots, d_n \geq 0$ such that $\delta(n) := \prod_{i=1}^n (d_i + 1) \in \text{poly}(\lambda)$. If the $(d_1, \dots, d_n)$ -ARSDH(n) assumption holds, then $(d_2, \dots, d_n)$ -ARSDH(n) also holds.*

It immediately gives us the following useful corollary.

Corollary 1 (ARSDH(n) $\Rightarrow$ ARSDH). *Let $d_1, \dots, d_n \geq 0$ such that $\delta(n) := \prod_{i=1}^n (d_i + 1) \in \text{poly}(\lambda)$. If the $(d_1, \dots, d_n)$ -ARSDH(n) assumption holds, then d_n -ARSDH also holds.*

4.1 GARSDH(n) $\Rightarrow$ ARSDH(n)

As a first step, we show that GARSDH(n) implies ARSDH(n) for appropriately chosen parameters. Thus, GARSDH(n) is at least as strong as ARSDH(n) for the same n for degree parameters are shown in the next theorem.

Theorem 2 (GARSDH(n) $\Rightarrow$ ARSDH(n)). *Let $d_1, \dots, d_n \geq 0$ such that $\delta(n) := \prod_{i=1}^n (d_i + 1) \in \text{poly}(\lambda)$. Additionally fix $D_1, \dots, D_{n-1} \geq 0$ such that $\delta'(n) = (d_1 + 1) \prod_{i=1}^{n-1} (D_i + 1) \in \text{poly}(\lambda)$. If the $(d_1, \dots, d_n)$ -GARSDH(n) assumption holds, then the $(D_1, \dots, D_{n-1}, d_1)$ -ARSDH(n) assumption holds.*

Proof. Let $\mathcal{A}$ be a t -time adversary which can win the ARSDH(n) game with a non-negligible probability ϵ . We construct a PPT adversary $\mathcal{B}$ that wins the GARSDH(n) game. $\mathcal{B}$ works as follows,

1. It gets the commitment key $\text{ck} = \{\{[t_1^{i_1} \cdots t_n^{i_n}]_1\}_{i_1, \dots, i_n=0}^{d_1, \dots, d_n}, \{[t_i]_2\}_{i=1}^n\}$.
2. $\mathcal{B}$ sets $D_n = d_1$ and ensures that $\delta'(n) = (d_1 + 1) \prod_{i=1}^{n-1} (D_i + 1) \in \text{poly}(\lambda)$. Samples $T_1, \dots, T_{n-1} \leftarrow_{\$} \mathbb{Z}_p$ and computes

$$\text{ck}' = \{\{[T_1^{i_1} \cdots T_n^{i_n}]_1\}_{i_1, \dots, i_n=0}^{D_1, \dots, D_n}, \{[T_i]_2\}_{i=1}^n\}.$$

Note that $[T_n^{i_n}]_1 \leftarrow [t_1^{i_n}]_1$ for all $i_n \in [0..D_n]$.

3. $\mathcal{B}$ runs $\mathcal{A}(\text{ck}')$ and gets $([\tilde{\chi}]_1, [\psi]_1, \{[\tilde{\varphi}_i]_1, \alpha_i\}_{i=1}^{n-1}, \mathcal{S}_n)$.
4. $\mathcal{B}$ randomly samples $\mathcal{S}_j^{(i_1, \dots, i_{j-1})} \subset \mathbb{Z}_p$ for $j \in [2, n]$ and $i_1 \in [0..d_1], \dots, i_{j-1} \in [0..d_{j-1}]$ such that $|\mathcal{S}_j^{(i_1, \dots, i_{j-1})}| = d_j + 1$. Sets $[\varphi_j^{(i_1, \dots, i_{j-1})}]_1 \leftarrow [0]_1$ for all $j \in [2, n]$.
5. Computes $[\chi]_1 = [\tilde{\chi}]_1 - \sum_{i=1}^{n-1} (T_i - \alpha_i)[\tilde{\varphi}_i]_1$ and store $\mathcal{S}_1 \leftarrow \mathcal{S}_n$ and store $[\varphi_1]_1 \leftarrow [\psi]_1$
6. $\mathcal{B}$ outputs $([\chi]_1, \{\{\mathcal{S}_j^{(i_1, \dots, i_{j-1})}\}_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}}, \{[\varphi_j^{(i_1, \dots, i_{j-1})}]_1\}_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}}\}_{j=1}^n)$.

Now, if $\mathcal{A}$ wins the $(D_1, \dots, D_n)$ -ARSDH(n) game, we have $|\mathcal{S}_n| = D_n + 1 = d_1 + 1$, $[\tilde{\chi}]_T \neq \sum_{i=1}^{n-1} [\tilde{\varphi}_i]_1 \bullet [T_i - \alpha_i]_1$, and $[\tilde{\chi}]_T = \sum_{i=1}^{n-1} [\tilde{\varphi}_i]_1 \bullet [T_i - \alpha_i]_1 + [\psi]_1 \bullet [Z_{\mathcal{S}_n}(T_n)]_2$. This implies that

1. $|\mathcal{S}_j^{(i_1, \dots, i_{j-1})}| = d_j + 1$ for $j \in [1..n]$ and $i_1 \in [0..d_1], \dots, i_{j-1} \in [0..d_{j-1}]$,
2. $[\chi]_T \neq \sum_{j=2}^n \sum_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}} [\varphi_j^{(i_1, \dots, i_{j-1})}]_1 \bullet [Z_{\mathcal{S}_j^{(i_1, \dots, i_{j-1})}}(t_j)]_2$,
3. $[\chi]_T = [\varphi_1]_1 \bullet [Z_{\mathcal{S}_1}(t_1)]_2 + \sum_{j=2}^n \sum_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}} [\varphi_j^{(i_1, \dots, i_{j-1})}]_1 \bullet [Z_{\mathcal{S}_j^{(i_1, \dots, i_{j-1})}}(t_j)]_2$.

The adversary $\mathcal{B}$ wins the GARSDH(n) game with the same probability as $\mathcal{A}$.

The running time of $\mathcal{B}$:

1. To generate the ck' , $\mathcal{B}$ performs $T_2^{i_1} \cdots T_n^{i_{n-1}} [t_1^{i_n}]_1$ for $i_1 \in [0..D_1], \dots, i_{n-1} \in [0..D_{n-1}]$, $i_n \in [0..d_1]$ i.e. $(d_1 + 1) \prod_{j=1}^{n-1} (D_j + 1)$ scalar multiplications i.e. $\mathcal{O}(\delta'(n))$ scalar multiplications.
2. $\mathcal{B}$ runs $\mathcal{A}$ which takes time t .
3. The time for the random sampling of the sets $\mathcal{S}_j^{i_1, \dots, i_{j-1}}$ from $\mathbb{Z}_p$ with $|\mathcal{S}_j^{(i_1, \dots, i_{j-1})}| = d_j + 1$ i.e. $\prod_{i=1}^j (d_i + 1)$ random samplings for $j \in [2..n]$.
4. $\mathcal{O}(n)$ scalar multiplications to compute $[\chi]_1 = [\tilde{\chi}]_1 - \sum_{i=1}^{n-1} (T_i - \alpha_i)[\tilde{\varphi}_i]_1$.

□

4.2 ARSDH $\implies$ Generalized ARSDH(n)

Now we show the main result of the paper. Standard d_1 -ARSDH implies $(d_1, \dots, d_n)$ -GARSDH(n). In fact, they are equivalent and reductions exist in both directions. Our technique is based on the observation that the reduction can sample $t_2, \dots, t_n$ locally.

Theorem 3 ($\text{ARSDH} \Leftrightarrow \text{GARSDH}(n)$). *Let $d_1, \dots, d_n \geq 0$ such that $\delta(n) := \prod_{i=1}^n (d_i + 1) \in \text{poly}(\lambda)$. Then, the d_1 -ARSDH assumption is equivalent to the $(d_1, \dots, d_n)$ -GARSDH(n) assumption.*

Proof. d_1 -ARSDH $\Rightarrow$ $(d_1, \dots, d_n)$ -GARSDH(n): Let $\mathcal{A}$ be a t -time adversary of the GARSDH(n) game with winning probability ε . We show that $\mathcal{A}$ can be transformed to an adversary $\mathcal{B}$ that wins in the ARSDH game with the same probability. The construction of $\mathcal{B}$ is as follows:

1. It gets as an input $\text{ck} = \{\{[t_1^i]_1\}_{i=0}^{d_1}, [1, t_1]_2\}$, where $t_1 \leftarrow \$ \mathbb{Z}_p$.
2. $\mathcal{B}$ samples $t_2, \dots, t_n \leftarrow \$ \mathbb{Z}_p$ and computes

$$\text{ck}' = \{\{[t_1^{i_1} \dots t_n^{i_n}]_1\}_{i_1, \dots, i_n=0}^{d_1, \dots, d_n}, \{[t_i]_2\}_{i=1}^n\}.$$

3. $\mathcal{B}$ runs $\mathcal{A}(\text{ck}')$ and gets
 - (a) $[\tilde{\chi}]_1$,
 - (b) for all $j \in [1, n]$, $\{\tilde{\mathcal{S}}_j^{(i_1, \dots, i_{j-1})}\}_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}}$ where $\tilde{\mathcal{S}}_j^{(i_1, \dots, i_{j-1})} \subseteq \mathbb{Z}_p$,
 - (c) for all $j \in [1, n]$, $\{[\tilde{\varphi}_j^{(i_1, \dots, i_{j-1})}]_1\}_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}}$.
4. $\mathcal{B}$ computes $[\chi]_1 = [\tilde{\chi}]_1 - \sum_{j=2}^n \sum_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}} Z_{\tilde{\mathcal{S}}_j^{(i_1, \dots, i_{j-1})}}(t_j) [\tilde{\varphi}_j^{(i_1, \dots, i_{j-1})}]_1$, $[\psi]_1 \leftarrow [\tilde{\varphi}_1]_1$ (see Remark 1 for notation) and outputs $([\chi]_1, [\psi]_1, \tilde{\mathcal{S}}_1)$

According to Definition 3 if adversary $\mathcal{A}$ wins the GARSDH(n) game, then

1. $|\tilde{\mathcal{S}}_j^{(i_1, \dots, i_{j-1})}| = d_j + 1$ for all $j \in [1, n]$,
2. $[\tilde{\chi}]_T \neq \sum_{j=2}^n \sum_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}} [\tilde{\varphi}_j^{(i_1, \dots, i_{j-1})}]_1 \bullet [Z_{\tilde{\mathcal{S}}_j^{(i_1, \dots, i_{j-1})}}(t_j)]_2$,
3. $[\tilde{\chi}]_T = \sum_{j=1}^n \sum_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}} [\tilde{\varphi}_j^{(i_1, \dots, i_{j-1})}]_1 \bullet [Z_{\tilde{\mathcal{S}}_j^{(i_1, \dots, i_{j-1})}}(t_j)]_2$.

which implies that

1. $|\tilde{\mathcal{S}}_1| = d_1 + 1$, (see Remark 1 for notation)
2. $[\chi]_T = [\tilde{\chi}]_T - \sum_{j=2}^n \sum_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}} [\tilde{\varphi}_j^{(i_1, \dots, i_{j-1})}]_1 \bullet [Z_{\tilde{\mathcal{S}}_j^{(i_1, \dots, i_{j-1})}}(t_j)]_2 \neq [0]_T$,
3. $[\chi]_T = [\tilde{\varphi}_1]_1 \bullet [Z_{\tilde{\mathcal{S}}_1}(t_1)]_2$.

Thus, according to Definition 1 $\mathcal{B}$ wins the d_1 -ARSDH game with ε probability.

The running time of $\mathcal{B}$:

1. To generate the ck' , $\mathcal{B}$ performs $t_2^{i_2} \dots t_n^{i_n} [t_1^{i_1}]_1$ for $i_1 \in [0..d_1], \dots, i_n \in [0..d_n]$ i.e. $\prod_{j=1}^n (d_j + 1)$ scalar multiplications i.e. $\mathcal{O}(\delta(n))$ scalar multiplications.
2. $\mathcal{B}$ runs $\mathcal{A}$ which takes time t .
3. At the last step $\mathcal{B}$ returns

$$[\chi]_1 = [\tilde{\chi}]_1 - \sum_{j=2}^n \sum_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}} Z_{\tilde{\mathcal{S}}_j^{(i_1, \dots, i_{j-1})}}(t_j) [\tilde{\varphi}_j^{(i_1, \dots, i_{j-1})}]_1$$

i.e. $\sum_{j=2}^n \prod_{i=1}^{j-1} (d_i + 1) \approx \mathcal{O}(\delta(n))$ multi-scalar multiplications and asymptotically the same number of polynomial evaluations $Z_{\tilde{\mathcal{S}}_j^{(i_1, \dots, i_{j-1})}}(t_j)$.

$(d_1, \dots, d_n)$ -GARS DH(n) $\Rightarrow$ d_1 -ARSDH: On the other hand, this reduction is again similar to the reduction of GARS DH(n) to GARS DH($n - 1$) as given in [BDH⁺25]. From an adversary $\mathcal{B}$ of d_1 -ARSDH game, we can construct the adversary $\mathcal{A}$ of GARS DH(n) game. The construction of $\mathcal{A}$ is as follows,

1. Gets as input $\text{ck} = \{\{[t_1^{i_1} \dots t_n^{i_n}]_1\}_{i_1, \dots, i_n=0}^{d_1, \dots, d_n}, \{[t_i]_2\}_{i=1}^n\}$.
2. Sets $\text{ck}' = \{\{[t_1^{i_1}]_1\}_{i_1=0}^{d_1}, [1, t_1]_2\}$.
3. $\mathcal{A}$ runs $\mathcal{B}(\text{ck}')$ and gets $([\chi]_1, [\psi]_1, \mathcal{S}_1)$.
4. $\mathcal{A}$ randomly samples $\mathcal{S}_j^{(i_1, \dots, i_{j-1})} \subset \mathbb{Z}_p$ for $j \in [2, n]$ and $i_1 \in [0..d_1], \dots, i_{j-1} \in [0..d_{j-1}]$ such that $|\mathcal{S}_j^{(i_1, \dots, i_{j-1})}| = d_j + 1$. Sets $[\varphi_j^{(i_1, \dots, i_{j-1})}]_1 \leftarrow [0]_1$ for all $j \in [2, n]$ and $[\varphi_1]_1 = [\psi]_1$.
5. $\mathcal{A}$ outputs

$$\left([\chi]_1, \{\{\mathcal{S}_j^{(i_1, \dots, i_{j-1})}\}_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}}, \{[\varphi_j^{(i_1, \dots, i_{j-1})}]_1\}_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}}\}_{j=1}^n \right).$$

Now, if $\mathcal{B}$ wins the d_1 -ARSDH game, we have $|\mathcal{S}_1| = d_1 + 1$, $[\chi]_T \neq [0]_T$, and $[\chi]_T = [\psi]_1 \bullet [Z_{\mathcal{S}_1}(t_1)]_2$. This implies that

1. $|\mathcal{S}_j^{(i_1, \dots, i_{j-1})}| = d_j + 1$ for $j \in [1..n]$ and $i_1 \in [0..d_1], \dots, i_{j-1} \in [0..d_{j-1}]$,
2. $[\chi]_T \neq \sum_{j=2}^n \sum_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}} [\varphi_j^{(i_1, \dots, i_{j-1})}]_1 \bullet [Z_{\mathcal{S}_j^{(i_1, \dots, i_{j-1})}}(t_j)]_2$,
3. $[\chi]_T = [\varphi_1]_1 \bullet [Z_{\mathcal{S}_1}(t_1)]_2 + \sum_{j=2}^n \sum_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}} [\varphi_j^{(i_1, \dots, i_{j-1})}]_1 \bullet [Z_{\mathcal{S}_j^{(i_1, \dots, i_{j-1})}}(t_j)]_2$.

The adversary $\mathcal{A}$ wins the GARS DH(n) game with the same probability as $\mathcal{B}$.

The running time of $\mathcal{A}$: The running time of the adversary $\mathcal{A}$ is dominated (besides running $\mathcal{B}$) by the random sampling of the sets $\mathcal{S}_j^{(i_1, \dots, i_{j-1})}$ from $\mathbb{Z}_p$ with $|\mathcal{S}_j^{(i_1, \dots, i_{j-1})}| = d_j + 1$ i.e. $\prod_{i=1}^j (d_i + 1)$ random samplings for $j \in [2..n]$. $\square$

As an immediate consequence of [Theorem 2](#) and [Theorem 3](#), we have the following corollary.

Corollary 2. *Let $\vec{d} = (d_1, \dots, d_n)$ be such that $\delta(n) := \prod_{i=1}^n (d_i + 1) \in \text{poly}(\lambda)$. Then, the d_n -ARSDH assumption implies the $(d_1, \dots, d_n)$ -ARSDH(n) assumption.*

Proof. For the forward implication, apply [Theorem 3](#) to the reversed degree vector $(d_n, \dots, d_1)$. This gives the corresponding GARS DH(n) assumption from d_n -ARSDH. Then [Theorem 2](#) gives $(d_1, \dots, d_n)$ -ARSDH(n).

Conversely, let $\mathcal{A}$ break d_n -ARSDH. Given an n -variate commitment key, restrict it to powers of t_n , run the adversary to obtain $([\chi]_1, [\psi]_1, \mathcal{S})$, and set $[\varphi_i]_1 = [0]_1$ and $\alpha_i = 0$ for all $i \in [1..n - 1]$. The winning conditions then coincide, with the same success probability. $\square$

Finally, we can conclude the following result.

Corollary 3. *Let $\vec{d} = (d_1, \dots, d_n)$ be such that $\delta(n) := \prod_{i=1}^n (d_i + 1) \in \text{poly}(\lambda)$. If d_i -ARSDH assumption holds for all $i \in [1..n]$, then the extended PST is a PoKoP with respect to Pgen and $\vec{d}$.*

Proof. This is almost an immediate consequence of [Corollary 2](#) and [Item 1](#) in [Theorem 1](#). One can observe that the proof of the latter result in [\[BDH⁺25\]](#) is an inductive proof that requires a series of assumptions:

- $(d_1, \dots, d_n)$ -ARSDH(n),
- $(d_1, \dots, d_{n-1})$ -ARSDH($n-1$),
- ...
- (d_1) -ARSDH(1) $\equiv$ d_1 -ARSDH.

Based on [Corollary 2](#), all these assumptions are implied by the assumptions d_n -ARSDH, d_{n-1} -ARSDH, ..., d_1 -ARSDH, respectively. Thus, our conclusion follows. $\square$

One would expect that a similar result holds for the canonical PST, but this turns out to not be the case, as we will explain in the next section.

5 Issues with the Reduction of Canonical PST to GARSDH

After careful study of the canonical PST reduction to GARSDH(n) assumption [\[BDH⁺25\]](#), we found a gap in their proof which does not seem to be easily fixable. Namely, the results in question are Lemma 5 and Theorem 5 in the full version of the paper [\[BDH⁺25\]](#). Both Lemma 5 and Theorem 5 are needed in order to show that if the GARSDH(n) assumption holds, then the canonical PST is computationally special sound (precursor notion for proving PoKoP).

For simplicity, we will focus on the case of $n = 2$. In this case, the special soundness adversary is given as an input a commitment key $\text{ck} = (\{[t_1^{i_1} t_2^{i_2}]_1\}_{i_1, i_2=0}^{d_1, d_2}, [t_1, t_2]_2)$ and it must output transcripts

$$([c]_1, x_1^{(i_1)}, x_2^{(i_1, i_2)}, [\pi_1^{(i_1, i_2)}]_1, [\pi_2^{(i_1, i_2)}]_1, z_{i_1, i_2}) \quad (1)$$

that are accepted by the verifier for all $i_1 \in [0..d_1]$ and $i_2 \in [0..d_2]$, i.e., it holds that,

$$[c - z_{i_1, i_2}]_1 \bullet [1]_2 = [\pi_1^{(i_1, i_2)}]_1 \bullet [t_1 - x_1^{(i_1)}]_2 + [\pi_2^{(i_1, i_2)}]_1 \bullet [t_2 - x_2^{(i_1, i_2)}]_2 .$$

The adversary wins if the extractor is unable to find a polynomial $L(X_1, X_2)$ of degree at most (d_1, d_2) such that $[c]_1 = [L(t_1, t_2)]_1$.

Background on Interpolation. Before continuing, we briefly review some basic facts about Lagrange interpolation. Let $A = \{a_0, a_1, \dots, a_d\} \subseteq \mathbb{Z}_p$ be a set of distinct elements. We denote by $\ell_i^A(X)$ the i th Lagrange polynomial over the set A , which is defined as

$$\ell_i^A(X) = \prod_{j=0, j \neq i}^d \frac{X - a_j}{a_i - a_j} = \omega_i^A Z_A(X) / (X - a_i) , \quad (2)$$

where $\omega_i^A = \frac{1}{Z_{A \setminus a_i}(a_i)} = \frac{1}{\prod_{j \in [0..d], j \neq i} (a_i - a_j)}$. The Lagrange basis polynomial $\ell_i^A(X)$ is the unique polynomial of degree at most d such that $\ell_i^A(a_j) = 1$ if $i = j$ and $\ell_i^A(a_j) = 0$ if $i \neq j$.

Given a set of output points $z_0, z_1, \dots, z_d \in \mathbb{Z}_p$, their interpolation polynomial over the set A is defined as $L(X) = \sum_{i=0}^d z_i \ell_i^A(X)$. This gives the lowest degree polynomial such that $L(a_i) = z_i$ for all $i \in [0..d]$. As a useful special case, interpolation of a constant 1 polynomial can be expressed as

$$\sum_{i=0}^d \ell_i^A(X) = 1 . \quad (3)$$

The same ideas can be extend to the multivariate case. Using elements from the transcripts in (1), we can define sets $\mathcal{S} = \{x_1^{(i_1)}\}_{i_1=0}^{d_1}$ and $\mathcal{S}_2^{(i_1)} = \{x_2^{(i_1, i_2)}\}_{i_2=0}^{d_2}$ for $i_1 \in [0..d_1]$. Then,

$$L(X, Y) = \sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} z_{i_1, i_2} \ell_{i_1}^{\mathcal{S}}(X) \ell_{i_2}^{\mathcal{S}_2^{(i_1)}}(Y) .$$

gives us the lowest degree polynomial such that $L(x_1^{(i_1)}, x_2^{(i_1, i_2)}) = z_{i_1, i_2}$ for all $i_1 \in [0..d_1]$ and $i_2 \in [0..d_2]$. This is precisely how the special soundness extractor in [BDH⁺25] recovers the polynomial $L(X_1, X_2)$ from the transcripts.

5.1 Issue with Lemma 5

The goal of the batching lemma [BDH⁺25, Lemma 5] is to take all of the transcripts in Equation (1) and batch them into a single equation which could then be used to break the GARS DH(n) assumption. Let $\mathcal{S} = \{x_1^{(i_1)}\}_{i_1=0}^{d_1}$ and $\mathcal{S}_2^{(i_1)} = \{x_2^{(i_1, i_2)}\}_{i_2=0}^{d_2}$ for $i \in [0..d_1]$. Let $L(X_1, X_2) = \sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} z_{i_1, i_2} \ell_{i_1}^{\mathcal{S}}(X_1) \ell_{i_2}^{\mathcal{S}_2^{(i_1)}}(X_2)$ be the interpolation polynomial over z_{i_1, i_2} . They show that

$$\begin{aligned} c - L(t_1, t_2) &\stackrel{Eq.(3)}{=} c \cdot \left(\sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} \ell_{i_1}^{\mathcal{S}}(t_1) \ell_{i_2}^{\mathcal{S}_2^{(i_1)}}(t_2) \right) - \sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} z_{i_1, i_2} \ell_{i_1}^{\mathcal{S}}(t_1) \ell_{i_2}^{\mathcal{S}_2^{(i_1)}}(t_2) \\ &= \sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} (c - z_{i_1, i_2}) \ell_{i_1}^{\mathcal{S}}(t_1) \ell_{i_2}^{\mathcal{S}_2^{(i_1)}}(t_2) \\ &= \sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} \left(\pi_1^{(i_1, i_2)}(t_1 - x_1^{(i_1)}) + \pi_2^{(i_1, i_2)}(t_2 - x_2^{(i_1, i_2)}) \right) \ell_{i_1}^{\mathcal{S}}(t_1) \ell_{i_2}^{\mathcal{S}_2^{(i_1)}}(t_2) \\ &\stackrel{Eq.(2)}{=} \sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} \pi_1^{(i_1, i_2)} w_{i_1}^{\mathcal{S}} Z_{\mathcal{S}}(t_1) \ell_{i_2}^{\mathcal{S}_2^{(i_1)}}(t_2) + \\ &\quad \sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} \pi_2^{(i_1, i_2)} \ell_{i_1}^{\mathcal{S}}(t_1) w_{i_2}^{\mathcal{S}_2^{(i_1)}} Z_{\mathcal{S}_2^{(i_1)}}(t_2) \\ &= Z_{\mathcal{S}}(t_1) \varphi_1 + \sum_{i_1=0}^{d_1} Z_{\mathcal{S}_2^{(i_1)}}(t_2) \varphi_2^{(i_1)} , \end{aligned}$$

where $\varphi_1 = \sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} \pi_1^{(i_1, i_2)} w_{i_1}^{\mathcal{S}} \ell_{i_2}^{\mathcal{S}_2^{(i_1)}}(t_2)$ and $\varphi_2^{(i_1)} = \sum_{i_2=0}^{d_2} \pi_2^{(i_1, i_2)} \ell_{i_1}^{\mathcal{S}}(t_1)$.

More precisely, the batching lemma for the bivariate case states that the following equation holds:

$$[c - L(t_1, t_2)]_T = [\varphi_1]_1 \bullet [Z_{\mathcal{S}}(t_1)]_2 + \sum_{i_1=0}^{d_1} [\varphi_2^{(i_1)}]_1 \bullet [Z_{\mathcal{S}_2^{(i_1)}}(t_2)]_2 ,$$

This is technically correct, but in the reduction the adversary must be able to efficiently compute the required elements. Specifically, it must be able to efficiently compute $[\chi]_1 = [c]_1 - [L(t_1, t_2)]_1$, $[\varphi_1]_1$, $[\varphi_2^{(i_1)}]_1$, $\mathcal{S} = \{x_1^{(i_1)}\}_{i_1=0}^{d_1}$, $\mathcal{S}_2^{(i_1)} = \{x_2^{(i_1, i_2)}\}_{i_2=0}^{d_2}$ for all $i_1 \in [0..d_1]$ to break the GARS DH(2). Note that $[\chi]_1$ can be computed easily using the commitment key ck as $[L(t_1, t_2)]_1 = \sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} z_{i_1, i_2} [\ell_{i_1}^{\mathcal{S}}(t_1) \ell_{i_2}^{\mathcal{S}_2^{(i_1)}}(t_2)]_1$. However, it is not possible to efficiently compute, say $[\varphi_2^{(i_1)}]_1$, which contains terms such as $[\pi_2^{(i_1, i_2)}]_1 \ell_{i_1}^{\mathcal{S}}(t_1)$. The adversary knows t_1 only as a group element from the commitment key ck . This is the first gap we identified.

For the sake of completeness, we also recall the generalized version of the batching lemma [BDH⁺25, Lemma 5] for n variables. The generalized lemma says that for $d_1, \dots, d_n \in \text{poly}(\lambda)$ and $(t_1, \dots, t_n) \in \mathbb{Z}_p^n$, let

$$\mathcal{T} = \left\{ \left([c]_1, \left(x_1^{(i_1)}, \dots, x_n^{(i_1, \dots, i_n)} \right), \left(\pi_1^{(i_1, \dots, i_n)}, \dots, \pi_n^{(i_1, \dots, i_n)} \right), z^{(i_1, \dots, i_n)} \right) \right\}_{i_1, \dots, i_n=0}^{d_1, \dots, d_n} \quad (4)$$

be a set of accepting transcripts such that for all $i_1 \in [d_1], \dots, i_n \in [d_n]$,

$$[c - z^{i_1, \dots, i_n}]_1 = \sum_{j=1}^n [\pi_j^{i_1, \dots, i_n}]_1 \bullet [t_j - x_j^{(i_1, \dots, i_n)}]_2$$

Now they define:

1. $S_j^{(i_1, \dots, i_{j-1})} = \left\{ x_j^{(i_1, \dots, i_j)} \right\}_{i_j=0}^{d_j}$
2. $[\varphi_j^{(i_1, \dots, i_{j-1})}]_1 = \sum_{i_j, \dots, i_n=0}^{d_j, \dots, d_n} \prod_{k \neq j} \ell_{i_k}^{S_k^{(i_1, \dots, i_{j-1})}}(t_k) \omega_{i_j}^{S_j^{(i_1, \dots, i_{j-1})}} [\varphi_j^{(i_1, \dots, i_n)}]_1$
3. $L(X_1, \dots, X_n) = \sum_{i_1, \dots, i_n=0}^{d_1, \dots, d_n} z^{(i_1, \dots, i_n)} \prod_{k=1}^n \ell_{i_k}^{S_k^{(i_1, \dots, i_{k-1})}}(X_k)$

Then the lemma states that the following holds:

$$[c - L(t_1, \dots, t_n)]_1 \bullet [1]_2 = \sum_{j=1}^n \sum_{i_1, \dots, i_{j-1}=0}^{d_1, \dots, d_{j-1}} [\varphi_j^{(i_1, \dots, i_{j-1})}]_1 \bullet [Z_{S_j^{(i_1, \dots, i_{j-1})}}(t_j)]_2$$

In the general case, it is not possible to efficiently compute the following elements in $\mathbb{G}_1$ using only the commitment key ck ,

$$[\varphi_j^{(i_1, \dots, i_{j-1})}]_1 = \sum_{i_j, \dots, i_n=0}^{d_j, \dots, d_n} \prod_{k \neq j} \ell_{i_k}^{S_k^{(i_1, \dots, i_{j-1})}}(t_k) \omega_{i_j}^{S_j^{(i_1, \dots, i_{j-1})}} [\varphi_j^{(i_1, \dots, i_n)}]_1 .$$

Though the batching lemma itself is true, it cannot be used in [BDH⁺25, Theorem 5] to obtain an efficient reduction.

5.2 Issue in Theorem 5

Even if the issue in Lemma 5 could be overcome, there is another issue in the proof of Theorem 5. In this theorem, the goal is to show that if $[c]_1 \neq [L(t_1, t_2)]_1$, then it is possible to break the GARS DH(n) assumption. The adversary gets ck as an input and outputs transcripts as in Equation (1). First, they apply the batching lemma to get

$$[c - L(t_1, t_2)]_T = [\varphi_1]_1 \bullet [Z_S(t_1)]_2 + \sum_{i_1=0}^{d_1} [\varphi_2^{(i_1)}]_1 \bullet [Z_{S_2^{(i_1)}}(t_2)]_2 ,$$

as above. Next they apply the GARS DH(2) assumption, where $\chi = c - L(t_1, t_2)$. Unless the adversary breaks the GARS DH(2) assumption, it must hold that

$$[c - L(t_1, t_2)]_T = \sum_{i_1=0}^{d_1} [\varphi_2^{(i_1)}]_1 \bullet [Z_{S_2^{(i_1)}}(t_2)]_2 . \quad (5)$$

This indeed seems to work (assuming we ignore the issue in Section 5.1). However, now they claim that one can apply the GARS DH(1) assumption to Equation (5) to conclude that $c - L(t_1, t_2) = 0$. This does not seem to be possible in any straightforward way. The equation for the GARS DH(1) assumption should be of the form $[\chi]_T = [\varphi]_1 \bullet [Z_S(t)]_2$. In fact, it is easy to find a counterexample to show that Equation (5) does not always imply $c - L(t_1, t_2) = 0$. Set $\varphi_2^{(0)} = 1$, $\varphi_2^{(1)} = -1$ and $\varphi_2^{(i)} = 0$ for all $i \geq 2$. Let us denote $Z_{S_2^{(0)}}(Y) = X^{d_2+1} + f(Y)$ and $Z_{S_2^{(1)}}(Y) = X^{d_2+1} + g(Y)$, where $\deg(f), \deg(g) \leq d_2$. It is now easy to see that

$$c - L(t_1, t_2) = f(t_2) - g(t_2) .$$

Since $f(X) - g(X)$ is a non-zero polynomial when $S_2^{(0)} \neq S_2^{(1)}$, it follows that $c - L(t_1, t_2) \neq 0$. Otherwise, one could simply find the roots of $f(X) - g(X)$ to break the discrete logarithm assumption.

$\text{Game}_{\text{Pgen}, \mathcal{A}, d_1, \dots, d_n}^{\text{TAUTPST}(n)}(\lambda)$
1 : $\text{gp} \leftarrow \text{Pgen}(1^\lambda), t_1, \dots, t_n \leftarrow \mathbb{Z}_p$; 2 : $\text{ck} \leftarrow (\{[t_1^{i_1} \dots t_n^{i_n}]_1\}_{i_1, \dots, i_n=0}^{d_1, \dots, d_n}, \{[t_i]_2\}_{i=1}^n)$; 3 : $\mathcal{T} \leftarrow \mathcal{A}(\text{ck})$, where $\mathcal{T} = \{([c]_1, \dots)\}$ is a tree of transcripts as in Equation (4); 4 : Define $L(X_1, \dots, X_n)$ as in Item 3 in Section 5.1; 5 : $b_1 \leftarrow$ all transcripts in $\mathcal{T}$ accept $\wedge [c]_1 \neq [L(t_1, \dots, t_n)]_1$; 6 : return b_1 ;

Figure 6: The game for $(d_1, \dots, d_n)$ -Tautological PST assumption.

5.3 Trivial Fix

We do not know how to recover the result of Theorem 5 in [BDH⁺25]. On the one hand, it might not be too important since the extended PST is essentially as efficient as the canonical PST, and we already showed that the extended PST is extractable under the standard ARSDH assumption. On the other hand, since canonical PST has a non-interactive opening phase (and there are not many PCS with this property), it would be nice to prove its extractability as well under some falsifiable assumption.

We point out that one can always make the following tautological (but still falsifiable) assumption which is sufficient.

Definition 5 (Tautological PST Assumption). Let Pgen be a bilinear group generator and $d_1, \dots, d_n \geq 0$ such that $\prod_{i=1}^n (d_i + 1) \in \text{poly}(\lambda)$. Then we say that the $(d_1, \dots, d_n)$ -Tautological PST assumption holds for Pgen , if for all PPT adversary $\mathcal{A}$,

$$\Pr \left[\text{Game}_{\text{Pgen}, \mathcal{A}, d_1, \dots, d_n}^{\text{TAUTPST}(n)}(\lambda) \right] \approx_\lambda 0 ,$$

where $\text{Game}_{\text{Pgen}, \mathcal{A}, d_1, \dots, d_n}^{\text{TAUTPST}(n)}(\lambda)$ is defined in Figure 6.

This assumption is clearly falsifiable (it is a game between an efficient adversary and an efficient challenger), and it effectively states that the canonical PST has computational special soundness, which is exactly what Theorem 5 in [BDH⁺25] aims to prove. It is a simple exercise to show as a sanity check that the Tautological PST assumption also holds in the generic group model. We leave it as an open problem to prove the extractability of the canonical PST under some weaker falsifiable assumption.

Acknowledgement

Janno Siim was funded by the European Union and Estonian Research Council via grant PSG1167 and project TEM-TA119. Pritam Pal was funded by the Estonian Research Council via project PRG2531.

References

- [BB04] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer, Berlin, Heidelberg, May 2004. doi: [10.1007/978-3-540-24676-3_4](https://doi.org/10.1007/978-3-540-24676-3_4).
- [BDH⁺25] Juraj Belohorec, Pavel Dvůrák, Charlotte Hoffmann, Pavel Hubáček, Kristýna Masková, and Martin Pastyřík. On extractability of the KZG family of

- polynomial commitment schemes. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 584–616. Springer, Cham, August 2025. doi:[10.1007/978-3-032-01887-8_19](https://doi.org/10.1007/978-3-032-01887-8_19).
- [BHKM26] Juraj Belohorec, Pavel Hubáček, Aleksi Kalsta, and Kristýna Mašková. CHOPIN: Optimal pairing-based multilinear polynomial commitments from bivariate KZG. Cryptology ePrint Archive, Paper 2026/480, 2026. URL: <https://eprint.iacr.org/2026/480>.
- [BHS26] Juraj Belohorec, Pavel Hubáček, and Dominik Stejskal. On lifting AGM security to AGM with oblivious sampling. Cryptology ePrint Archive, Paper 2026/201, 2026. URL: <https://eprint.iacr.org/2026/201>.
- [CADLT25] Rutchathon Chairattana-Apirom, Nico Döttling, Anna Lysyanskaya, and Stefano Tessaro. Everlasting anonymous rate-limited tokens. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part VI*, volume 16250 of *LNCS*, pages 435–468. Springer, Singapore, December 2025. doi:[10.1007/978-981-95-5119-4_14](https://doi.org/10.1007/978-981-95-5119-4_14).
- [CBBZ23] Binyi Chen, Benedikt Bünz, Dan Boneh, and Zhenfei Zhang. HyperPlonk: Plonk with linear-time prover and high-degree custom gates. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part II*, volume 14005 of *LNCS*, pages 499–530. Springer, Cham, April 2023. doi:[10.1007/978-3-031-30617-4_17](https://doi.org/10.1007/978-3-031-30617-4_17).
- [CGKY26] Alessandro Chiesa, Ziyi Guan, Christian Knabenhans, and Zihan Yu. On the fiat–shamir security of succinct arguments from functional commitments. In Nadia Heninger and Mike Rosulek, editors, *Advances in Cryptology – CRYPTO 2026*, pages 189–213, Cham, 2026. Springer Nature Switzerland. doi:[10.1007/978-3-032-35424-2_7](https://doi.org/10.1007/978-3-032-35424-2_7).
- [CHM⁺20] Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Psi Vesely, and Nicholas P. Ward. Marlin: Preprocessing zkSNARKs with universal and updatable SRS. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 738–768. Springer, Cham, May 2020. doi:[10.1007/978-3-030-45721-1_26](https://doi.org/10.1007/978-3-030-45721-1_26).
- [Den02] Alexander W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 100–109. Springer, Berlin, Heidelberg, December 2002. doi:[10.1007/3-540-36178-2_6](https://doi.org/10.1007/3-540-36178-2_6).
- [FKL18] Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Cham, August 2018. doi:[10.1007/978-3-319-96881-0_2](https://doi.org/10.1007/978-3-319-96881-0_2).
- [GK16] Shafi Goldwasser and Yael Tauman Kalai. Cryptographic assumptions: A position paper. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 505–522. Springer, Berlin, Heidelberg, January 2016. doi:[10.1007/978-3-662-49096-9_21](https://doi.org/10.1007/978-3-662-49096-9_21).
- [GR19] Alonso González and Carla Ràfols. Shorter pairing-based arguments under standard assumptions. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 728–757. Springer, Cham, December 2019. doi:[10.1007/978-3-030-34618-8_25](https://doi.org/10.1007/978-3-030-34618-8_25).

- [GWC19] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Report 2019/953, 2019. URL: <https://eprint.iacr.org/2019/953>.
- [HASW24] Mathias Hall-Andersen, Mark Simkin, and Benedikt Wagner. FRIDA: Data availability sampling from FRI. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VI*, volume 14925 of *LNCS*, pages 289–324. Springer, Cham, August 2024. doi:10.1007/978-3-031-68391-6_9.
- [Kus18] John Kuszmaul. Verkle trees. Technical report, MIT, 2018. URL: <https://math.mit.edu/research/highschool/primes/materials/2018/Kuszmaul.pdf>.
- [KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 177–194. Springer, Berlin, Heidelberg, December 2010. doi:10.1007/978-3-642-17373-8_11.
- [Lip25] Helger Lipmaa. Plonk is simulation extractable in ROM under falsifiable assumptions. In Benny Applebaum and Huijia (Rachel) Lin, editors, *TCC 2025, Part IV*, volume 16271 of *LNCS*, pages 3–36. Springer, Cham, December 2025. doi:10.1007/978-3-032-12290-2_1.
- [LPS24] Helger Lipmaa, Roberto Parisella, and Janno Siim. Constant-size zk-SNARKs in ROM from falsifiable assumptions. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part VI*, volume 14656 of *LNCS*, pages 34–64. Springer, Cham, May 2024. doi:10.1007/978-3-031-58751-1_2.
- [LPS25] Helger Lipmaa, Roberto Parisella, and Janno Siim. On knowledge-soundness of plonk in ROM from falsifiable assumptions. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VII*, volume 16006 of *LNCS*, pages 362–395. Springer, Cham, August 2025. doi:10.1007/978-3-032-01907-3_12.
- [PST13] Charalampos Papamanthou, Elaine Shi, and Roberto Tamassia. Signatures of correct computation. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 222–242. Springer, Berlin, Heidelberg, March 2013. doi:10.1007/978-3-642-36594-2_13.
- [Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT’97*, volume 1233 of *LNCS*, pages 256–266. Springer, Berlin, Heidelberg, May 1997. doi:10.1007/3-540-69053-0_18.
- [Sii26] Janno Siim. The art of linearization: - from a KZG’s trick to a general commitment framework. In Shi Bai and Edoardo Persichetti, editors, *PKC 2026, Part II*, volume 16552 of *LNCS*, pages 455–487. Springer, Cham, May 2026. doi:10.1007/978-3-032-26734-4_15.
- [WZ25] Benedikt Wagner and Arantxa Zapico. Proving the security of PeerDAS without the AGM. Cryptology ePrint Archive, Report 2025/1683, 2025. URL: <https://eprint.iacr.org/2025/1683>.
- [XTP⁺26] Jiajun XIN, Xiangnan TIAN, Christodoulos PAPPAS, Yongjin Huang, and Dimitris PAPADOPOULOS. Gryphes: Hybrid proofs for modular snarks with applications to zkrollups. *PoPETs/PETS*, 2026(2):198–214, June 2026. 26th Privacy Enhancing Technologies Symposium (PETS 2026), PETS 2026 ; Conference date: 20-07-2026 Through 25-07-2026. URL: <https://petsymposium.org/2026/>, doi:10.56553/POPETS-2026-0044.

- [ZCF24] Hadas Zeilberger, Binyi Chen, and Ben Fisch. BaseFold: Efficient field-agnostic polynomial commitment schemes from foldable codes. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part X*, volume 14929 of *LNCS*, pages 138–169. Springer, Cham, August 2024. [doi:10.1007/978-3-031-68403-6_5](https://doi.org/10.1007/978-3-031-68403-6_5).
- [Zha22] Mark Zhandry. To label, or not to label (in generic groups). In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 66–96. Springer, Cham, August 2022. [doi:10.1007/978-3-031-15982-4_3](https://doi.org/10.1007/978-3-031-15982-4_3).