Non-Malleable Reductions of Knowledge

Antonio Faonio and Lili Tong

EURECOM, Sophia Antipolis, France
{antonio.faonio,lili.tong}@eurecom.fr

Abstract. Non-malleability for non-interactive zero-knowledge proofs requires that, given a proof for a statement, it is infeasible to derive a valid proof for a related statement without knowing a corresponding witness.

We introduce a modular framework for analyzing non-malleable reductions of knowledge (RoKs). A reduction of knowledge transforms the task of proving knowledge for a source relation into proving knowledge for a target relation, often simpler or more structured. RoKs is an extremely useful tool for compositions. We identify different settings in which the composition of two RoKs, and in particular two non-interactive RoKs obtained via the Fiat-Shamir transform, preserves simulation extractability, and thus non-malleability. Our framework isolates simple and concrete properties required from each component, including novel forms of zero knowledge and new security notions that are easier to verify than full simulation extractability. This yields a systematic toolbox for establishing non malleability in modular proof systems.

Finally, we illustrate the power of our approach by analyzing LaBRADOR (Beullens and Seiler, CRYPTO'23), a lattice-based proof system for R1CS. We provide the first analysis of its simulation extractability and, since LaBRADOR is not zero knowledge, we design a zero knowledge variant that preserves its practical efficiency and sublinear proof size. Our results show that non-malleability for advanced proof systems can be achieved modularly, significantly simplifying the security analysis.

Table of Contents

Non-Malleable Reductions of Knowledge . . . 1
Antonio Faonio and Lili Tong
1 Introduction . . . 3
1.1 Technical Overview . . . 4
1.2 Related Work . . . 6
2 Preliminaries . . . 7
2.1 Notation . . . 7
2.2 Ring-related Preliminaries . . . 7
2.3 Lattice and Discrete Gaussian Distribution . . . 8
2.4 Approximate norm check . . . 10
2.5 Hardness Assumptions . . . 10
3 Reduction of Knowledge . . . 10
3.1 Interactive Reductions of Knowledge . . . 10
The FS compiler. . . . 11
Basic Properties. . . . 12
3.2 Composition for RoK . . . 13
4 Zero-Knowledge for RoK . . . 13
4.1 On the Zero-Knowledge of the Composition . . . 14
5 Simulation Extractability for RoK . . . 16
5.1 Unique Response and Hard to Invert Properties . . . 17
6 Non-Malleable Composition of RoK . . . 21
7 zk-LaBRADot is Non-Malleable . . . 26
7.1 Hiding Double Commitment . . . 27
7.2 Commit-and-Prove LaBRADot . . . 28
7.3 Approximate Norm Check with partial Zero-Knowledge . . . 33
7.4 Reducing $\mathcal{F}'$ to $\mathcal{F}$ with partial zero-knowledge . . . 35
7.5 The non-linear-term RoK for the Committed Principal Relation . . . 36
7.6 The linear-term RoK for the Committed Principal Relation . . . 40
7.7 Putting All Together . . . 43
A Proof of Lemma 10 . . . 49
B Additional Material for Section 7 . . . 51
B.1 Hiding for (the double) commitment scheme . . . 51
B.2 Aggregation with Partial Zero-Knowledge . . . 54
C Fiat-Shamir-based Signature-of-Knowledge . . . 54

1 Introduction

Zero-knowledge proofs are fundamental cryptographic primitives that allow a prover to convince a verifier of the validity of a statement without revealing any additional information beyond its truth [12, 38]. While the security of (non-interactive) zero-knowledge proofs is well understood in a stand-alone setting, the situation becomes more subtle when an adversary has access to multiple proofs. In this setting, a malicious prover may attempt to manipulate existing proofs in order to produce a valid proof for a different statement without knowing a corresponding witness. This threat, known as a *malleability attack*, was first formalized by Sahai [53], who introduced the notion of simulation soundness. This notion was later strengthened to simulation extractability by Groth [39], which guarantees that even in the presence of simulated proofs, any successful forgery implies knowledge of a valid witness. simulation extractability implies non-malleability for proof of knowledge as defined by De Santis *et al.* [25]. The risks associated with malleability are not merely theoretical. In practice, malleability vulnerabilities have led to significant incidents, including attacks involving more than three hundred thousand Bitcoins [26].

Recent work has proposed several modular approaches for establishing simulation extractability of proof systems. For example, [27, 28, 30, 41] focus on the non malleability of the PIOP-to-zkSNARK transformation [8, 22, 29, 32]. Other works, such as [19, 24, 33–35, 45], analyze specific constructions [4, 17, 32, 55–57], extending or refining the framework originally introduced in Faust *et al.* [31]. Very recent work by Abdolmaleki *et al.* [2] studies simulation extractability in the context of proof carrying data [11]. Notably, the work of Campanelli, Faonio and Russo [19] proposes a framework for modularly composing simulation-extractable commit-and-prove zkSNARKs [20]. A key insight of this work is that when composing two zkSNARK proofs that share a common witness, and thus prove the conjunction of their respective statements, it is possible, under suitable conditions, to relax the security requirements of one of the components. In particular, one of the zkSNARKs only needs to satisfy knowledge soundness and witness indistinguishability, rather than full simulation extractability.This modular perspective shows that by decomposing a zkSNARK into subcomponents and identifying precisely which parts require a non malleability analysis and which do not, one can significantly simplify the task of proving non malleability for complex systems.

The elegant notion of Reduction of Knowledge (RoK) [42] was introduced as a natural framework for decomposing complex proof systems into smaller modules. A RoK generalizes the concept of a proof of knowledge. Instead of directly proving knowledge of a witness $\mathbb{w}_1$ for a statement $\mathbb{x}_1$ in a relation R_1, the prover and verifier jointly define a new instance $\mathbb{x}_2$, and the prover convinces the verifier by outputting a corresponding witness $\mathbb{w}_2$ with respect to a relation R_2. The new instance may be designed to be simpler or smaller according to some complexity measure, for example as in compressed Sigma-Protocols [5, 17], or it may be a randomized instance of the same relation, as in Sigma protocols [23, 54], or it can aggregate multiple instances of the same relation to one single instance as in the notion of folding scheme [13, 14, 16, 43, 44][1]. The work of Kothapalli and Parno [42] shows that if Π_1 is a RoK from R_1 to R_2 and Π_2 is a RoK from R_2 to R_3, denoted $\Pi_1 : \mathsf{R}_1 \to \mathsf{R}_2$ and $\Pi_2 : \mathsf{R}_2 \to \mathsf{R}_3$, then the natural composition $\Pi_2 \circ \Pi_1 : \mathsf{R}_1 \to \mathsf{R}_3$, which first executes Π_1 and then feeds its output to Π_2, is knowledge sound provided that both Π_1 and Π_2 are knowledge sound.

Our Contributions. We study the composition $\Pi_2 \circ \Pi_1$ in the context of simulation extractability and identify several settings under which this property is preserved. In particular, we focus on non-interactive RoKs obtained via the Fiat-Shamir transform, where simulation extractability can be defined in a natural way. We present a broad collection of sufficient conditions on Π_1 and Π_2 that ensure simulation extractability of the composed protocol $\Pi_2 \circ \Pi_1$. These conditions are summarized in Table 1, and a more detailed treatment appears in the technical overview in Section 1.1.

In slightly more detail, we distinguish two notions of zero-knowledge for RoKs and, consequently, two corresponding notions of simulation extractability, since these properties depend on the underlying zero-knowledge simulators. The first notion of simulation extractability is a natural adaptation of the standard definition for proofs of knowledge, whereas the second is specifically tailored to the RoK setting. For the latter notion, denoted $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ in Table 1, we prove a result in the spirit of Faust *et al.* [31], adapted to this weaker notion of simulation extractability; see Theorem 4. In this setting, we introduce a new notion that we call hard to invert, denoted HtoInv in Table 1. This notion can be understood as a natural weakening of the unique-response property defined in [31], and later extended to multi-round protocols in [24, 34].

[1] zkSNARKs based on polynomial interactive oracle proofs (PIOPs) could be seen as reductions of knowledge to evaluation of committed polynomials on random points.

	Π_1		Π_2		$\Pi_2 \circ \Pi_1$	
	ZK	SND	ZK	SND	ZK	SND
I	✓	SIM-EXT	-	KS + 0-UR	✓	SIM-EXT
II	partial	SIM-EXT$_{\mathsf{pZK}}$	✓	SIM-EXT	✓	SIM-EXT
III	partial	KS + HtoInv	✓	SIM-EXT	✓	SIM-EXT
IV	partial$^+$	KS	✓	SIM-EXT	✓	SIM-EXT
V	$*$	SIM-EXT$_{\mathsf{pZK}}$	$*$	SIM-EXT$_{\mathsf{pZK}}$	$*$	SIM-EXT$_{\mathsf{pZK}}$

Table 1: Possible settings for the RoK composition theorem, Theorem 5. For the last row, $*$ means that Π_1 and Π_2 can be either forms of zero-knowledge and the composed RoK inherits the best of the two properties. With partial$^+$ we mean that the partial zero-knowledge simulator does not need to reprogram the random oracle. The rows in light gray are corollaries of our composition theorem Theorem 5 and Theorem 4.

Our framework can significantly facilitate the analysis of simulation extractability for schemes based on RoKs. To support this claim, we provide, as case study, an analysis of simulation extractability for the lattice-based scheme LaBRADOR [9]. To the best of our knowledge, this constitutes the first analysis of simulation extractability for a practical scheme based on lattice assumptions. Moreover, since LaBRADOR is not zero-knowledge, we additionally propose a zero-knowledge variant of LaBRADOR that preserves its practical efficiency advantages.

1.1 Technical Overview

Throughout the paper, we adopt the following notation: for $i \in \{1, 2\}$, $\mathbb{x}_i$ denotes the *source* instance of Π_i, and $\mathbb{x}_{i+1}$ denotes its *target* instance. The source instance is the input to the RoK, whereas the target instance is the instance generated during the execution of the RoK.

The first naive observation is that when both Π_1 and Π_2 are simulation extractable then the composition $\Pi := \Pi_2 \circ \Pi_1$ maintains simulation extractability. The intuitive reason is that target instance for Π_1 and source instance for Π_2 binds together the two proofs. Notice this is not necessarily true for (conjunction) composition of standard PoKs as noticed in [19].

A more interesting observation is that if Π_1 is zero-knowledge, then the composition $\Pi_2 \circ \Pi_1$ is zero-knowledge even if Π_2 itself is not. Indeed, a zero-knowledge simulator for $\Pi_2 \circ \Pi_1$ can first run the simulator for Π_1 and then execute the honest prover of Π_2 using the simulated witness corresponding to its source instance. Since no simulated proofs of Π_2 appear in the adversary's view, one might hope to reduce security directly to the knowledge soundness of Π_2. However, this is not sufficient in general. The RoK Π_2 may allow one to derive a valid proof for a statement from an existing proof of the same statement, for example if the protocol is re-randomizable. To obtain simulation extractability, we therefore require that proofs be uniquely determined by the witness. We formalize this requirement as the notion of 0-unique response , which is a special case of the k-unique response notion introduced in [24], see row I of Table 1.

Our third observation is that when Π_2 is zero-knowledge, the zero-knowledge requirement on Π_1 can be relaxed. In particular, there are two natural ways to interpret the pair $(\pi_1, \mathbb{w}_2)$ output by Π_1. One can view it as a proof for the PoK naturally induced by Π_1, or alternatively, one can regard π_1 as the proof generated by Π_1 and $\mathbb{w}_2$ as the private output of the prover. The second perspective allows us to refine the notion of zero-knowledge for RoKs by requiring only the existence of a simulator that produces a valid RoK proof π_1, without necessarily simulating the corresponding witness.

We further observe that these two forms of zero-knowledge lie at opposite ends of a spectrum. A more granular notion can be obtained by parsing the target witness $\mathbb{w}_2$ as $(\mathbb{w}_2', \mathbb{w}_2'')$ and requiring the zero-knowledge simulator to produce only the tuple $(\pi_1, \mathbb{w}_2')$. To formalize this idea, we restate the target relation R_2 by defining an *extended* relation $\tilde{\mathsf{R}}_2$ such that $(\tilde{\mathbb{x}}, \mathbb{w}_2'') \in \tilde{\mathsf{R}}_2$ if and only if $(\mathbb{x}_2, (\mathbb{w}_2', \mathbb{w}_2'')) \in \mathsf{R}_2$, where $\tilde{\mathbb{x}} = (\mathbb{x}_2, \mathbb{w}_2')$. In other words, we syntactically move part of the witness into the statement. This simple syntactic transformation turns out to be quite useful and aligns well with the modular nature of RoKs. It allows different components of the witness to be masked and turned zero-knowledge in separate steps, so that these steps can be handled by different RoKs, as we later illustrate in the case of LaBRADOR.

Returning to simulation extractability, as mentioned in the first part of the introduction, the notion inherently depends on the zero-knowledge simulator being considered. We therefore define the simulation-extractability game in the presence of a partial zero-knowledge simulator, which is required to correctly

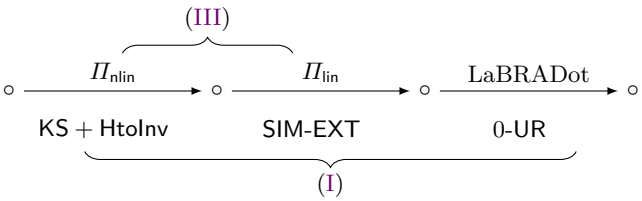


Fig. 1: Overview of the simulation-extractability analysis of ZK-LaBRADot, with references to the results in Table 1.

simulate π_1 but not necessarily the corresponding witness $\mathbb{w}_2$. In more detail, we design the game so that the second component of the simulator's output is ignored in simulation queries, namely the simulator might output both $\pi, \mathbb{w}_2$ but the simulation oracle returns only π. This leads to the notion of $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$, namely simulation extractability in the presence of a partial zero-knowledge simulator. This design choice has two consequences. First, the notion remains meaningful even when only a standard zero-knowledge simulator is available, since it can be turned into a partial one by discarding its second output. Second, the copy-paste attack is no longer trivially applicable. We can show that when Π_1 is $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ and Π_2 is simulation-extractable then their composition is so, see row II of Table 1.

We now analyze the relationship between $\mathsf{SIM\text{-}EXT}$ and $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$. Somewhat surprisingly, $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ is not strictly weaker than $\mathsf{SIM\text{-}EXT}$. In particular, in certain pathological cases where R_2 is in P, a RoK may satisfy $\mathsf{SIM\text{-}EXT}$ but fail to satisfy $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$; see Remark 1. We identify the missing ingredient needed to resolve this discrepancy as a new security property that we call hard-to-invert. In the hard-to-invert game, the adversary is asked to recover the corresponding witness $\mathbb{w}_2$ for one of the simulated proofs returned by the (partial) simulation oracle. This notion is closely related to the unique-response property introduced in [24, 31]. In the unique-response setting, the adversary is required to produce two accepting proofs for the same statement that share the same prefix, for example a tuple $(\pi_1, \mathbb{w}_2, \tilde{\mathbb{w}}_2)$ with $\mathbb{w}_2 \neq \tilde{\mathbb{w}}_2$ and both $\pi_1, \mathbb{w}_2$ and $\pi_1, \tilde{\mathbb{w}}_2$ being valid proofs. By analogy, hard-to-invert relates to unique-response for RoKs in the same way that preimage resistance relates to collision resistance for hash functions.[2]

Finally, we present two approaches to obtain $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$, see rows III, IV of Table 1. The first builds on the connection between unique-response and hard-to-invert. We show that if a scheme is partial zero-knowledge, knowledge sound and satisfies the hard-to-invert property, then it achieves $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$. The argument closely follows the original idea of [31]. Briefly, in the reduction to knowledge soundness, the simulator can be executed locally, and the hard-to-invert property guarantees that any forgery produced in the $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ game remains valid in the knowledge soundness game because the random oracle queries used to verify the forgery are not reprogrammed. The second approach is to assume that the partial zero-knowledge simulator does not reprogram the random oracle, achieving the same effect as the hard-to-invert property in the previous argument. A related observation was made independently and concurrently in [2] in the context of proof-carrying data. The assumption of a non-reprogramming partial zero-knowledge simulator is feasible in some cases. For example, in the Schnorr protocol, the partial zero-knowledge simulator only needs to sample a value g^z for a random mask z and does not need to compute the witness $cx + z$ corresponding to the target instance, where c denotes the challenge and x the secret.

ZK-LaBRADOR is Non-Malleable. LaBRADOR is an MSIS-based proof of knowledge. In particular, its core component, which we call LaBRADot[3], is a RoK that is recursively applied until the vectors in the target witness have constant dimension. LaBRADot is a RoK from (and to) dot-product equations with short solutions. Concretely, it proves knowledge of short-norm vectors $\vec{\boldsymbol{s}} = (\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_r) \in (\mathcal{R}_q^n)^r$ satisfying equations of the form:

$$\sum_{i,j=1}^{r} \boldsymbol{a}_{i,j} \langle \vec{\boldsymbol{s}}_i, \vec{\boldsymbol{s}}_j \rangle + \sum_{i=1}^{r} \langle \vec{\boldsymbol{\varphi}}_i, \vec{\boldsymbol{s}}_i \rangle - \boldsymbol{b} = 0.$$

The first result we establish is that LaBRADot, or more precisely a commit-and-prove adaptation of it, satisfies 0-unique-response. To prove this, we exploit its RoK-based structure. The proof of 0-unique-response closely follows the knowledge soundness analysis of LaBRADot. In particular, we show that all

[2] Continuing the analogy, the notion of weak unique-response [33] for R/PoKs is comparable to second-preimage resistance for hash functions.

[3] LaBRADOR stands for Lattice-Based Recursively Amortized Demonstration Of R1CS, while LaBRADot stands for Lattice-Based Recursively Amortized Demonstration of Dot-Product equations.

messages sent by the prover are either explicitly committed or implicitly bound to commitments. Any adversary from which we can extract a witness must generate its messages as an honest prover would, using the extracted witness to produce the forgery. Otherwise, the adversary would violate the binding property of the commitments. This implies that, for a fixed committed witness, there is a unique way to generate an accepting proof.

Unfortunately, we can only prove 0-unique response for LaBRADot assuming that the target witness has small ℓ_∞ norm. On the other hand, LaBRADot guarantees only that the source witness has bounded ℓ_2 norm. As a consequence, we cannot directly conclude that two recursive applications of LaBRADot preserve the 0-unique-response property. The issue stems from the fact that LaBRADot uses an Ajtai-based double-commitment scheme [14] that admits re-randomization. More precisely, the witness material $\vec{\boldsymbol{s}}$ is first committed into $\vec{\boldsymbol{t}}$, and then $\vec{\boldsymbol{t}}$ is decomposed into vectors $(\vec{\boldsymbol{t}_j})_j$ satisfying $\sum_j b^j \vec{\boldsymbol{t}_j} = \vec{\boldsymbol{t}}$, where each $\vec{\boldsymbol{t}_j}$ has ℓ_∞ norm bounded by b. These values are then committed again. To re-randomize the decomposition, one can add to the vectors $(\vec{\boldsymbol{t}_j})_j$ a low ℓ_2-norm vector (whose ℓ_∞ norm exceeds b), that belongs to the lattice $\Lambda = \left\{(\vec{\boldsymbol{x}_j})_j : \sum_j b^j \vec{\boldsymbol{x}_j} = 0\right\}$. We can sample efficiently from Λ using the techniques of [50][4].

We now describe the zero-knowledge RoK for LaBRADot. Since LaBRADot satisfies 0-unique-response, such a ZK-RoK must in particular be simulation-extractable (see row I of Table 1). We decompose the construction into two subcomponents, each of which is itself a RoK. See Fig. 1 for an overview of the composition. The first component, denoted Π_{nlin}, applies the folding technique from Nova [43] to randomize the witness $\vec{\boldsymbol{s}} = (\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_r)$. This technique, however, introduces additional variables corresponding to the cross terms between the original witness and the mask, as well as error terms arising from the mask. We collectively denote these new variables by the vector $\vec{\boldsymbol{u}}$. The relations among these variables can be expressed as linear equations, which are incorporated into the target instance.

The resulting target witness after Π_{nlin} has the form $(\vec{\boldsymbol{v}}, \vec{\boldsymbol{u}})$, where $\vec{\boldsymbol{v}}$ is zero-knowledge, and in fact it is of the form $\vec{\boldsymbol{v}} = \vec{\boldsymbol{z}} + \boldsymbol{\alpha}\vec{\boldsymbol{s}}$ for a challenge $\boldsymbol{\alpha}$ and mask $\vec{\boldsymbol{z}}$, but $\vec{\boldsymbol{u}}$ is not yet protected. In this setting, it suffices to prove that Π_{nlin} is partial zero-knowledge, meaning that the simulator can produce $(\pi_1, \vec{\boldsymbol{v}})$.

The zero-knowledge simulator reprograms the random oracle to simulate the commitment corresponding to $\vec{\boldsymbol{v}}$. In contrast, it can simulate the commitments to the cross-terms without sampling valid openings, since these commitments are not opened by the partial zero-knowledge simulator. This simplifies the proof of the hard-to-invert property, which requires the adversary to produce openings for the commitments to the cross-terms for a simulated proof. Thus proving that Π_{nlin} is $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ by the result of row III of Table 1. Recall that the component $\vec{\boldsymbol{u}}$ of the witness is subject only to linear constraints. Therefore, the second component Π_{lin} is a variation of a classical Σ-protocol for linear relations [5]. Proving $\mathsf{SIM\text{-}EXT}$ for such a RoK is fairly standard. Indeed, the unique-response property follows directly from the binding of the commitment sent in the first round, and the zero-knowledge property can be established using analogous arguments.

One detail omitted in the above description is that the protocols, as stated, would guarantee only *weak* knowledge soundness. Namely, the extractor may recover a ring element $\bar{\boldsymbol{a}}$ together with a committed witness $\vec{\boldsymbol{s}}$ such that $\|\bar{\boldsymbol{a}}\vec{\boldsymbol{s}}\|_2$ is small, without obtaining any direct bound on the norm of $\vec{\boldsymbol{s}}$ itself. To address this issue, we employ a zero-knowledge variant of the approximate norm-checking technique introduced by Lyubashevsky, Nguyen and Plançon [48][5]. Specifically, Π_{nlin} (resp. Π_{lin}) is defined as the composition $\Pi_{\mathsf{nlin}} := (\Pi_1 \circ \Pi_{\mathsf{zk\text{-}agg}'} \circ \Pi_{\mathsf{zk\text{-}norm}})$ (resp. with Π_2 in place of Π_1), where Π_1 (resp. Π_2) is the RoK described previously. We show that both $\Pi_{\mathsf{zk\text{-}norm}}$ and $\Pi_{\mathsf{zk\text{-}agg}'}$ satisfy non-reprogramming partial zero knowledge. As a consequence, their $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ properties follow directly from the result in row IV of Table 1. The simulation extractability of the composed protocols Π_{nlin} and Π_{lin} then follows modularly from the composition results in rows II and V of Table 1.

1.2 Related Work

Some related work on simulation extractability was already discussed in the first part of the introduction. Here we briefly summarize the works most relevant to our results. Several works studying simulation extractability of Fiat–Shamir-based proofs and arguments of knowledge, including [24, 33, 34, 45], follow a general methodology that originates from [31]. In these frameworks, knowledge soundness together with

[4] In Theorem 6, we turn this limitation into a feature: we show that this technique can be used to construct hiding double commitments without padding additional random ring elements into the witness.

[5] Additionally, we adapt the improved Johnson–Lindenstrauss-over-$\mathbb{Z}_q$ lemma used in LaBRADOR to the zero-knowledge setting, thereby strengthening the corresponding lemma from [48].

suitable forms of unique-response and zero-knowledge is sufficient to establish simulation extractability. These properties are typically k-zero-knowledge and k-unique-response [24], where the latter builds on ideas from [34], or variants such as weak unique-response [33].

On Partial Zero-Knowledge. Although not always formalized explicitly, the idea of simulating only a partial transcript in recursive constructions appears in several works. The closest to our setting is [43], where the authors define "zero-knowledge" for a reduction of knowledge in a way that corresponds to our notion of partial zero-knowledge. We make this distinction explicit because we view RoKs primarily as a tool for analyzing the simulation extractability of the associated PoKs. From this perspective, we define zero-knowledge for a RoK by directly inheriting the standard definition of zero-knowledge for PoK.

Very recently, and independently of our work, a partial zero-knowledge simulator that does not explicitly reprogram the random oracle has been used to establish simulation extractability for Proof-Carrying Data based on split-accumulation schemes by Abdolmaleki *et al.* [2]. The notion of split-accumulation scheme [16, 18] is closely related to the notion of Reduction of Knowledge.

Lattice-based Zero Knowledge. This work uses several lattice-based techniques. For example, rejection sampling by Lyubashevsky [46] is a standard tool to remove the dependency on the witness when applying a shift with masking values, which helps to achieve zero-knowledge. In addition, Lyubashevsky and Seiler [49] showed that if an element in some certain cyclotomic ring is short, then it is invertible. This result is useful both for building the challenge set and for sampling randomness when applying the leftover hash lemma [3, 15].

We also note that there have been several improvements since LaBRADOR. For instance, Klooß *et al.* [40] fixed a soundness gap in the knowledge extraction process. Moreover, Aardal *et al.* [1] proved that the Fiat–Shamir version of LaBRADOR satisfies knowledge soundness, and extended the construction from a 2-splitting ring to an l-splitting ring.

Recent lattice-based folding schemes build on ideas originating from LaBRADOR. This includes LatticeFold [13] and its refinement LatticeFold+ [14], as well as more recent constructions such as Neo [51] and Cyclo [36], which adapt these techniques in different directions to improve efficiency and expressiveness.

Finally, in concurrent work, Biasioli *et al.* [10] proposed a different approach to adding zero knowledge to LaBRADOR. Like our construction, their approach places a zero-knowledge layer before LaBRADOR, but their layer follows a different strategy from ours. We believe that proving their zero-knowledge layer to be $\mathsf{SIM\text{-}EXT}$ is an interesting open problem.

2 Preliminaries

2.1 Notation

The notation $(a, \cdot) \in \mathcal{I}$ is a shortcut meaning that there exists some value b such that $(a, b) \in \mathcal{I}$. Likewise, the notation $(a, \cdot) \leftarrow \mathcal{B}$ for an algorithm $\mathcal{B}$ means that the algorithm outputs a pair and we keep only the first component, ignoring the second one. We assume that all the algorithms, including the adversaries, simulators and extractors, are stateful machines and, given an algorithm $\mathcal{X} \in \{\mathcal{A}, \mathcal{B}, \dots\}$, we identify with $\mathsf{st}_{\mathcal{X}}$ its state.

The centered binomial distribution with a positive integer parameter κ, written as Bin_κ, is the distribution $\sum_{i=1}^{\kappa}(a_i - b_i)$, where $a_i, b_i \xleftarrow{\$} \{0,1\}$.

We use RO to denote a uniformly random function with domain $\{0,1\}^*$ (its codomain may vary depending on the application). Let $\mathcal{Q}$ be a set defining a partial function over $\{0,1\}^*$. We write $\mathsf{RO}[\mathcal{Q}]$ for the function that, on input x, outputs y if $(x, y) \in \mathcal{Q}$, and $\mathsf{RO}(x)$ otherwise.

We define an NP relation as a polynomial-time algorithm R that verifies triples consisting of public parameters pp, inputs $\mathbb{x}$, and witnesses $\mathbb{w}$. We use R interchangeably as both an algorithm and a set, and write $(\mathsf{pp}, \mathbb{x}, \mathbb{w}) \in \mathsf{R}$ to denote that the triple is accepted by R. The public parameters pp depend on the proof system and may include elements such as commitment keys and security parameters. To simplify notation, we sometimes omit pp when it is clear from the context.

2.2 Ring-related Preliminaries

Throughout this paper, we let $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$ be cyclotomic ring of degree d a power of 2. For a modulus q, let $\mathbb{Z}_q$ be the ring of integers mod q and $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$. As in LaBRADOR [9], we focus on the case where $\mathcal{R}_q$ splits into 2 residue fields.

We denote the CRT decomposition $\mathcal{R}_q \cong \mathcal{R}_1 \times \mathcal{R}_2$ and the projection onto each CRT slot by π_i for $i \in \{1, 2\}$. We denote polynomial vector in $\mathcal{R}_q^n$ by $\vec{\boldsymbol{a}}$, integer vector in $\mathbb{Z}_q^n$ by $\vec{a}$. For a vector $\vec{v}$, denote by v_i the i-th entry of $\vec{v}$. We denote matrices $A \in \mathbb{Z}_q^{m \times n}$ by capital letters. Similarly by boldface capital letters $\boldsymbol{A}$ for matrices between modules over the polynomial ring $\mathcal{R}_q$. Given a polynomial $\boldsymbol{a} \in \mathcal{R}_q$, we use lowercase $\vec{a}$ to denote its coefficient embedding in $\mathbb{Z}_q^d$. We denote with $\mathsf{ct}(\boldsymbol{a}) \in \mathbb{Z}_q$ the constant term of $\boldsymbol{a}$, namely if $\boldsymbol{a}(X) = \sum a_i X^i$ then $\mathsf{ct}(\boldsymbol{a}) = a_0$.

We recall the ℓ_p-norm of $\vec{v} \in \mathbb{Z}^n$ defined by $\|\vec{v}\|_p = (\sum_i |v_i|^p)^{\frac{1}{p}}$ for $p > 0$. Define the lift function $\mathrm{Lift} : \mathbb{Z}_q \to \mathbb{Z}$ choosing for each element in $\mathbb{Z}_q$ the unique representative in interval $[-\frac{q}{2}, \frac{q}{2})$, which induces ℓ_p-norm on $\mathbb{Z}_q$ by $\|\vec{v}\|_p := \|\mathrm{Lift}(\vec{v})\|_p$. We equip the polynomial ring $\mathcal{R}_q$ with ℓ_p-norm defined by $\|\boldsymbol{a}\|_p := \|\vec{a}\|_p$. Let $\mathcal{C} \subset \mathcal{R}_q$ be the set consisting of ring elements of norm bound τ, operator norm bound T. Moreover, denote by $\bar{\mathcal{C}}$ the difference of any pair of distinct elements from $\mathcal{C}$, elements in $\bar{\mathcal{C}}$ are units.

The following result describes concrete conditions for constructing such 2-splitting $\mathcal{R}_q$ and set $\mathcal{C}$.

Lemma 1 (Corollary 1.2, [49]). *Let $n \geq k > 1$ be powers of 2 and $p = 2k + 1 \pmod{4k}$ be a prime. Then the polynomial $X^n + 1$ factors as*

$$X^n + 1 \equiv \prod_{j=1}^{k} \left(X^{n/k} - r_j \right) \pmod{p}$$

for distinct $r_j \in \mathbb{Z}_p^*$ where $X^{n/k} - r_j$ are irreducible in the ring $\mathbb{Z}_p[X]$.

Furthermore, any $y \in \mathbb{Z}_p[X]/(X^n + 1)$ that satisfies either

$$0 < \|y\|_\infty < \frac{1}{\sqrt{k}} \cdot p^{1/k}$$

or

$$0 < \|y\| < p^{1/k}$$

has an inverse in $\mathbb{Z}_p[X]/(X^n + 1)$.

To prove inner product between integer vectors, one may use the following property on the Galois automorphism σ_{-1} defined by $X \mapsto X^{-1}$.

Lemma 2 (Lemma 2.4, [48]). *Let $\vec{a}, \vec{b} \in \mathbb{Z}_q^{nd}$, $\langle \vec{a}, \vec{b} \rangle = \mathsf{ct}(\langle \sigma_{-1}(\vec{\boldsymbol{a}}), \vec{\boldsymbol{b}} \rangle)$*

We use the following variant of Schwartz-Zippel lemma over $\mathcal{R}_q$, adapted to LaBRADOR setting:

Lemma 3 (Lemma 3.3 adapted, [1]). *Let $n \in \mathbb{N}$ and $f \in \mathcal{R}_q[X_1, \ldots, X_n]$ be a non-zero polynomial over $\mathcal{R}_q$ of total degree $\mathsf{deg}(f) \geq 0$. Let $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n$ be sampled independently and uniformly at random from $\mathcal{C}$. Then*

$$\Pr\left[f(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n) = 0\right] \leq \frac{\mathsf{deg}(f)}{|\mathcal{C}|}.$$

2.3 Lattice and Discrete Gaussian Distribution

The (unnormalised) Gaussian function over $\mathbb{R}^n$ with standard deviation $\sigma > 0$ and centre $\vec{c} \in \mathbb{R}^n$ is defined as

$$\rho_{\vec{c},\sigma}(\vec{x}) = \exp\left(-\frac{\|\vec{x} - \vec{c}\|^2}{2\sigma^2} \right).$$

When the centre is $\vec{c} = \vec{0}$, we omit it and simply write $\rho_\sigma(\vec{x})$.

For any countable set $A \subseteq \mathbb{R}^n$, define the Gaussian mass of A as $\rho_{\vec{c},\sigma}(A) = \sum_{\vec{x} \in A} \rho_{\vec{c},\sigma}(\vec{x})$. The discrete Gaussian distribution over a set $A \subseteq \mathbb{R}^n$ with centre $\vec{c}$ and parameter σ is defined as

$$D_{A,\sigma,\vec{c}}(\vec{x}) = \frac{\rho_{\vec{c},\sigma}(\vec{x})}{\rho_{\vec{c},\sigma}(A)}, \qquad \vec{x} \in A.$$

When $\vec{c} = \vec{0}$, we write $D_{A,\sigma}$.

An m-dimensional lattice Λ is a discrete additive subgroup of $\mathbb{R}^m$. For some $k \leq m$, called the *rank* of the lattice, Λ is generated as the set of all $\mathbb{Z}$-linear combinations of some k linearly independent basis vectors $B = \{\vec{b}_1, \ldots, \vec{b}_k\}$, namely, $\Lambda = \{B\vec{z} : \vec{z} \in \mathbb{Z}^k\}$. Throughout this work, we are only concerned with full-rank integer lattices, i.e., $\Lambda \subseteq \mathbb{Z}^m$ with $k = m$.

Lemma 4 (Adapted from [50, Lemma 2.6] and [52, Lemma 5.3]). *Let $\Lambda \subset \mathbb{R}^m$ be an m-dimensional lattice and $s, t > 0$, then*

$$\Pr\left(\|D_{\Lambda,s}\| < s\sqrt{m}\right) \leq 2^{-m},$$

$$\Pr\left(\|D_{\Lambda,s}\|_\infty > s \cdot t\right) \leq 2m \cdot e^{-\pi t^2}.$$

One may define discrete Gaussian over $\mathcal{R}$ induced by coefficient embedding $\mathcal{R}^n \to \mathbb{Z}^{nd}$, concretely, $D_\sigma(\vec{\boldsymbol{x}}) := \frac{\rho_\sigma(\vec{x})}{\rho_\sigma(\mathcal{R}^n)}$. We first describe rejection sampling on discrete Gaussian over $\mathcal{R}^n$ [46], a tool to remove dependency on witness from the distribution of proof element of the form a linear combination of the witness and a masking vector.

Algorithm 1 $\mathsf{Rej}(\boldsymbol{Z}, \boldsymbol{B}, \sigma, \rho)$

1: Sample $u \leftarrow \mathrm{Uniform}[0, 1)$.
2: Compute $t(\boldsymbol{Z}, \boldsymbol{B}) := \exp\left(\frac{-2\langle \boldsymbol{Z}, \boldsymbol{B}\rangle + \|\boldsymbol{B}\|^2}{2\sigma^2}\right)$
3: **if** $u > \rho^{-1} \cdot t(\boldsymbol{Z}, \boldsymbol{B})$ **then return** 0 ▷ reject
4: **else return** 1 ▷ accept
5: **end if**

Lemma 5 (Standard Gaussian Rejection Sampling, [46]). *Let $\boldsymbol{B} \in \mathcal{R}^{r \times n}$ be any matrix. Let $\sigma = \omega(\|\boldsymbol{B}\|_2\sqrt{\log n})$, there exists a constant ρ such that the procedure that samples a $\boldsymbol{Y} \leftarrow D_\sigma^{r \times n}$ and then returns the output of $\mathsf{Rej}(\boldsymbol{Z} := \boldsymbol{Y} + \boldsymbol{B}, \boldsymbol{B}, \sigma, \rho)$ outputs 1 happens with probability within $2^{-\omega(\log n)}/\rho$ of $1/\rho$. The distribution of $\boldsymbol{Z}$, conditioned on the output being 1, is within statistical distance of $2^{-\omega(\log n)}/\rho$ of $D_\sigma^{r \times n}$.*

Let $\Lambda \subseteq \mathbb{Z}^{nd} \subset \mathbb{R}^{nd}$ be a full rank lattice, fix $\vec{t} \in \mathbb{Z}^{nd}$. We are also interested in rejection sampling on $D_{\Lambda + \vec{t}, \vec{v}, \sigma}$ for outputting $D_{\Lambda + \vec{t}, \vec{t}, \sigma}$. For this reason, we adapt the standard gaussian rejection sampling above.

Lemma 6. *Let $\vec{\boldsymbol{b}} \in \mathcal{R}_q^n$ be any vector such that $\vec{t} - \vec{b} \in \Lambda$. Let $\sigma = \omega(\|\vec{\boldsymbol{b}}\|_2\sqrt{\log n})$, there exists a constant ρ such that the procedure that samples a $\vec{\boldsymbol{y}} \leftarrow D_{\Lambda,\sigma}$ and then returns the output of $\mathsf{Rej}(\vec{\boldsymbol{z}} := \vec{\boldsymbol{z}} + \vec{\boldsymbol{b}}, \vec{\boldsymbol{b}}, \sigma, \rho)$ outputs 1 happens with probability within $2^{-\omega(\log n)}/\rho$ of $1/\rho$. The distribution of $\vec{\boldsymbol{z}}$, conditioned on the output being 1, is within statistical distance of $2^{-\omega(\log n)}/\rho$ of $D_{\vec{\boldsymbol{t}} + \Lambda, \vec{t}, \sigma}$.*

The above lemma follows from the following lemma for bounding the success probability of Algorithm 1.

Lemma 7 (Generalised Lemma 4.5, [46]). *Let $\Lambda \subset \mathbb{R}^m$ be a lattice, for any $\vec{v}, \vec{c} \in \mathbb{Z}^m$, if $\sigma = \omega(\|v\|\sqrt{\log m})$, then*

$$\Pr\left[\, D_{\vec{c}+\Lambda,\vec{c},\sigma}(z)/D_{\vec{c}+\Lambda,\vec{v},\sigma}(z) = O(1)\ ;\ z \xleftarrow{\$} D_{\vec{c}+\Lambda,\vec{c},\sigma} \,\right] = 1 - 2^{-\omega(\log m)}.$$

This lemma is obtained by the following lemma that bounds the inner product of a varable from $D_{\vec{c}+\Lambda,\vec{c},\sigma}$ with any vector in $\mathbb{R}^m$.

Lemma 8 (Generalised Lemma 4.3, [46]). *Let $\Lambda \subset \mathbb{R}^m$ be a lattice, for any vector $\vec{v}, \vec{c} \in \mathbb{R}^m$ and any $\sigma, r > 0$,*

$$\Pr\left[\, |\langle \vec{z}, \vec{v}\rangle| > r\,;\ \vec{z} \leftarrow D_{\vec{c}+\Lambda,\vec{c},\sigma} \,\right] \ \leq\ 2e^{-\frac{(r-|\langle c,v\rangle|)^2}{2\|\vec{v}\|^2\sigma^2}}$$

Proof. We first notice that Lemma 4.3 in [46] generalises to $\vec{y} \leftarrow D_{\Lambda,\sigma}$. For $\vec{z} \leftarrow D_{\vec{c}+\Lambda,\vec{c},\sigma}$, variable $\vec{y} := \vec{z} - \vec{c}$ has distribution $D_{\Lambda,\sigma}$. Hence by triangular inequality, $\Pr\left[\,|\langle \vec{z}, \vec{v}\rangle| > r\right] \leq \Pr\left[\,|\langle \vec{y}, \vec{v}\rangle| > r - |\langle c, v\rangle|\right] \leq 2e^{-\frac{(r-|\langle c,v\rangle|)^2}{2\|\vec{v}\|^2\sigma^2}}$.

2.4 Approximate norm check

We use Johnson-Lindenstrauss projection to perform approximate norm check. For correctness, we rely on the following concentration lemma under the heuristic[6] substitution of Bin_1 with the normal distribution of variance 1/2 following [1].

Lemma* 9 (Generalised Lemma 2.8 from [48]). *Let $q \in \mathbb{N}$. For every $\lambda \in \mathbb{N}$, there exist constants $C_1 = C_1(\lambda)$ and $C_2 = C_2(\lambda)$ such that the following holds. Let $\vec{w} \in \mathbb{Z}^n$ with $\|\vec{w}\|_2 < b$ for some bound $b \leq q/C_1$,*

$$\Pr_{\Pi \leftarrow \mathsf{Bin}_1^{2\lambda \times n}} \left[\|\Pi \vec{w} \bmod q\|_2 > \sqrt{C_2} \|\vec{w} \bmod q\|_2 \right] \lesssim 2^{-\lambda},$$

where $\lesssim$ stresses the heuristic nature of the result.

For soundness, we rely on the following anti-concentration lemma:

Lemma* 10 (Generalised Lemma 2.9 from [48]). *Let $q \in \mathbb{N}$. For every $\lambda \in \mathbb{N}$, there exist constants $D_1 = D_1(\lambda)$ and $D_2 = D_2(\lambda)$ such that the following holds. Let $\vec{w} \in \mathbb{Z}_q^n$ with $\|\vec{w}\|_2 \geq b$ for some bound $b \leq q/D_1$ and let $\vec{y}$ be an arbitrary vector in $\mathbb{Z}_q^{2\lambda}$,*

$$\Pr_{\Pi \leftarrow \mathsf{Bin}_1^{2\lambda \times n}} \left[\|\Pi \vec{w} + \vec{y} \bmod q\|_2 < \sqrt{D_2} b \right] \lesssim 2^{-\lambda/2},$$

where $\lesssim$ stresses the heuristic nature of the result.

We give a proof of the lemma in Section A.

2.5 Hardness Assumptions

Definition 1 (Module-SIS Assumption). *Let $n, m, \beta \in \mathbb{N}$. The Module Short Integer problem $\mathsf{M\text{-}SIS}_{n;m,\beta}$ over $\mathcal{R}_q$ is defined as follows. Given $\boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m}$, find an $\vec{\boldsymbol{x}} \in \mathcal{R}_q^m$ such that $\boldsymbol{A}\vec{\boldsymbol{x}} = \vec{\boldsymbol{0}}$ and $0 < \|\vec{\boldsymbol{x}}\|_2 \leq \beta$. The M-SIS assumption states that no PPT adversary can solve this problem with non-negligible advantage.*

Weak Commitment Openings. Given an Ajtai commitment $\vec{\boldsymbol{v}} = \boldsymbol{A}\vec{\boldsymbol{w}} \in \mathcal{R}_q^\kappa$ with commitment key $\boldsymbol{A} \in \mathcal{R}_q^{\kappa \times n}$, an opening $\vec{\boldsymbol{w}}^* \in \mathcal{R}_q^n$ to $\vec{\boldsymbol{v}}$ is a short vector such that $\boldsymbol{A}\vec{\boldsymbol{w}}^* = \vec{\boldsymbol{v}}$. The notion of weak opening was introduced in [7]. A weak opening to $\vec{\boldsymbol{v}}$ is a tuple $(\bar{\boldsymbol{c}}, \vec{\boldsymbol{w}}^*) \in \bar{\mathcal{C}} \times \mathcal{R}_q^n$ such that $\boldsymbol{A}\vec{\boldsymbol{w}}^* = \vec{\boldsymbol{v}}$ and $\|\bar{\boldsymbol{c}}\vec{\boldsymbol{w}}^*\|_2 \leq \beta$. Note that given two weak openings $(\bar{\boldsymbol{c}}_1, \vec{\boldsymbol{w}}_1^*)$, $(\bar{\boldsymbol{c}}_2, \vec{\boldsymbol{w}}_2^*)$, one obtains an M-SIS solution for $\boldsymbol{A}$ of norm $4T\beta$ when $\bar{\boldsymbol{c}}_1 \neq \bar{\boldsymbol{c}}_2$, of norm 2β otherwise. Notice that, in general, we need not require $\bar{\boldsymbol{c}} \in \bar{\mathcal{C}}$: any ring element with low operator norm suffices. However, every such element appearing in our reductions belongs to $\bar{\mathcal{C}}$.

3 Reduction of Knowledge

3.1 Interactive Reductions of Knowledge

We define reduction of knowledge, the definition in this work is slightly different from [42], in the sense that the prover only outputs a proof together with a witness associated with the new relation. In case of public-coin verifier, this definition is equivalent to that in [42].

Definition 2 (Reduction of Knowledge). *Let R_1 and R_2 be two polynomial-time relations. A $(2\mu+1)$-move reduction of knowledge Π (in short, RoK) from R_1 to R_2, which we identify with the notation $\Pi : \mathsf{R}_1 \to \mathsf{R}_2$, consists of three PPT algorithms $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ denoting the generator, the prover, and the verifier:*

- *The algorithm* Setup *takes security parameter 1^λ and outputs public parameters* pp.

[6] We use the following convention: whenever a theorem or lemma uses the heuristic lemma, directly or indirectly, we add a $\star$ to its label so that readers can identify results based on the heuristic.

- *The prover and the verifier define a protocol* $(\mathbb{x}_2, \mathbb{w}_2), \mathtt{trns} \leftarrow \langle \mathsf{P}(\mathbb{w}_1), \mathsf{V}\rangle(\mathsf{pp}; \mathbb{x}_1)$*, where* $\mathbb{w}_1$ *is the private input of* P*,* pp *and* $\mathbb{x}_1$ *are the common inputs and* $\mathbb{x}_2$ *is the common output,* $\mathbb{w}_2$ *is the private output of the prover and* $\mathtt{trns}$ *is the transcription of the interaction. The protocol has* $\mu+1$ *messages sent by the prover to the verifier,* μ *messages from the verifier to the prover, and the prover speaks first.*

Whenever unnecessary, we omit $\mathtt{trns}$ from the output of the protocol execution $\langle \mathsf{P}(\mathbb{w}_1), \mathsf{V}\rangle(\mathsf{pp}; \mathbb{x}_1)$. Likewise, when clear from context, we omit pp from the common inputs.

A special case is the one-move reduction of knowledge, that is, a non interactive reduction of knowledge. We highlight the syntax of this case because it is the main object studied in this paper. More in detail, we consider non-interactive reduction of knowledge in the random oracle model. Briefly, in this case, we denote the single message from the prover to the verifier as π, we assume that the prover outputs both π and $\mathbb{w}_2$ and the verifier, upon input $\mathbb{x}_1, \pi$ outputs $\mathbb{x}_2$ (or an error message) and we assume that both the prover and verifier have oracle access to a ROM.

Definition 3 (Non-Interactive Reduction of Knowledge in the ROM). *Let* R_1 *and* R_2 *be two polynomial-time relations. A non-interactive reduction of knowledge, in short* NRoK*,* Π *from* R_1 *to* R_2 *in the ROM is a one-move reduction of knowledge with the following syntax:*

- *The prover* P *takes public parameters* pp*, oracle access to a ROM* RO *and a instance-witness pair* $(\mathsf{pp}, \mathbb{x}_1, \mathbb{w}_1) \in \mathsf{R}_1$*, generates a proof* π *and new witness* $\mathbb{w}_2$ *associated with* R_2*.*
- *The verifier* V *takes public parameters* pp*, an instance* $\mathbb{x}_1$ *associated in* R_1 *and a proof* π*, oracle access to* RO*, and it outputs* $\mathbb{x}_2$*.*

Moreover, we denote with $\bar{\mathsf{V}}$ *the* associated PoK verifier*, namely,*

$$\bar{\mathsf{V}}(\mathsf{pp}, \mathbb{x}_1, \pi, \mathbb{w}_2) = 1 \iff (\mathsf{pp}, \mathbb{x}_2, \mathbb{w}_2) \in \mathsf{R}_2 \wedge \mathsf{V}(\mathsf{pp}, \mathbb{x}, \pi) = \mathbb{x}_2 \neq \bot.$$

Reductions of knowledge satisfy (perfect) completeness, knowledge soundness, and possibly zero knowledge. We give the definitions of these properties only for the non interactive case. Since we consider the Fiat Shamir transform, completeness and knowledge soundness for the interactive case can be derived by the corresponding properties of the Fiat Shamir version of the reduction of knowledge. These properties are basically the equivalent of completeness and knowledge soundness for the for the associated PoK $\Sigma = (\mathsf{Setup}, \mathsf{P}, \bar{\mathsf{V}})$.

In addition, the interactive reductions of knowledge that we consider in this work satisfy public reducibility, which we define below. This notion is useful for defining the Fiat Shamir transform, which we present next.

Definition 4 (Public Reducibility). *Π satisfies public reducibility if there exists a deterministic polynomial-time algorithm f such that for any PPT adversary $\mathcal{A}$ and expected polynomial-time adversary P^*,*

$$\Pr\left[f(\mathsf{pp}, \mathbb{x}_1, \mathtt{trns}) = \mathbb{x}_2 \ : \ \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathbb{x}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pp}) \\ (\mathbb{x}_2, \mathbb{w}_2), \mathtt{trns} \leftarrow \langle \mathsf{P}^*(\mathsf{st}), \mathsf{V}\rangle(\mathsf{pp}; \mathbb{x}_1) \end{array} \right] = 1.$$

The FS compiler. To obtain a non-interactive reduction of knowledge from an (public-coin) interactive one, we may use the Fiat-Shamir transformation that is described below.

Definition 5 (Fiat-Shamir Transformation). *Let* $\Pi : \mathsf{R}_1 \to \mathsf{R}_2$ *be a public-coin* $(2\mu+1)$*-move interactive argument of knowledge that satisfies public reducibility with public-reducibility function* f*. Denote the transcript as* $\mathtt{trns} = (a_1, c_1, \ldots, a_\mu, c_\mu, a_{\mu+1}, \mathbb{w}_2)$*. The Fiat–Shamir transformation turns* Π *into a non-interactive reduction which we denote* $(\Pi)_{\mathsf{FS}} = (\mathsf{Setup}_{\mathsf{FS}}, \mathsf{P}_{\mathsf{FS}}, \mathsf{V}_{\mathsf{FS}})$ *in the random oracle model, where:*

- $\mathsf{Setup}_{\mathsf{FS}}$ *is the same as* Setup*.*
- *The prover* P_{FS}*, on input* $(\mathsf{pp}, \mathbb{x}_1, \mathbb{w}_1)$*, invokes* $\mathsf{P}(\mathbb{x}_1, \mathbb{w}_1)$*, and instead of asking the verifier for challenges* c_i *in round* i*, queries the random oracle to get*

$$c_i = \mathsf{RO}(\bar{a}_i), \quad \bar{a}_i = (\mathsf{pp} \| \mathbb{x}_1 \| a_1 \| \ldots \| a_i), \quad \forall i = 1, \ldots, \mu. \tag{1}$$

  P_{FS} *then outputs a non-interactive proof* $\pi = (a_1, \ldots, a_{\mu+1})$ *and a target witness* $\mathbb{w}_2$ *associated with* R_2*.*
- *The verifier* V_{FS} *on input* $(\mathsf{pp}, \mathbb{x}_1, \pi)$*, derives the challenges* c_i *by querying the random oracle as* P_{FS} *does, sets* $\mathtt{trns} \leftarrow (a_1, c_1, \ldots, c_\mu, a_{\mu+1})$*, and returns* $f(\mathsf{pp}, \mathbb{x}_1, \mathtt{trns})$*.*

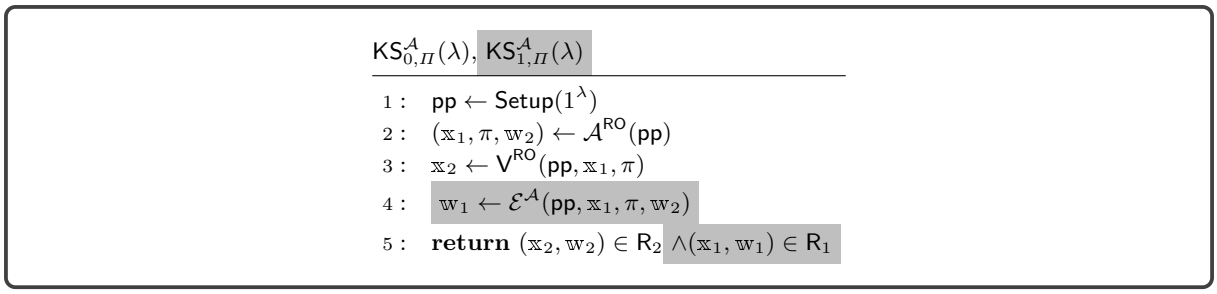


Fig. 2: KS security games. The extractor $\mathcal{E}$ has black-box access to the adversary $\mathcal{A}$; in particular, it implements the random oracle RO for $\mathcal{A}$ and can rewind $\mathcal{A}$ to any point in its execution using the same initial randomness. The KS_1 game additionally executes the gray-highlighted steps.

Basic Properties. We describe the main properties that a reduction of knowledge satisfies. As mentioned above, we define these properties for NRoK and then derive the corresponding properties for RoK using the Fiat–Shamir transform.

Definition 6 (Perfect Completeness). *A* NRoK Π *has perfect completeness if for any* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$*, any* $(\mathbb{x}_1, \mathbb{w}_1)$ *s.t* $(\mathsf{pp}, \mathbb{x}_1, \mathbb{w}_1) \in \mathsf{R}_1$*, any random oracle* RO*, we have* $\bar{\mathsf{V}}^{\mathsf{RO}}(\mathsf{pp}, \mathbb{x}_1, \pi, \mathbb{w}_2) = 1$ *where* $\pi, \mathbb{w}_2 \leftarrow \mathsf{P}^{\mathsf{RO}}(\mathsf{pp}, \mathbb{x}_1, \mathbb{w}_1)$*. Moreover, a interactive RoK* Π *has perfect completeness if* Π_{FS} *has perfect completeness.*

Without loss of generality, we focus on argument systems. Moreover, we require knowledge soundness to hold with respect to non-uniform polynomial-time adversaries.

Definition 7 (Knowledge Soundness). *A non-interactive RoK* $\Pi : \mathsf{R}_1 \to \mathsf{R}_2$ *is knowledge sound,* KS*, if there exists an extractor* $\mathcal{E}$ *running in expected polynomial time such that for every no-uniform PT adversary* $\mathcal{A}$*, the following probability is negligible in* λ*:*

$$\mathbf{Adv}^{\mathsf{KS}}(\Pi, \mathcal{E}, \mathcal{A})(\lambda) := \left| \Pr[\mathsf{KS}^{\mathcal{A}}_{0,\Pi}(\lambda)] - \Pr[\mathsf{KS}^{\mathcal{E},\mathcal{A}}_{1,\Pi}(\lambda)] \right|.$$

for games defined in Fig. 2. Moreover, an interactive RoK Π *is knowledge sound if* Π_{FS} *is knowledge sound.*

It is common to establish KS of a Fiat–Shamir-compiled NRoK by first proving $\vec{k}$-special soundness of the underlying interactive protocol and then invoking the following result by Attema *et al.* [6] to transfer this property to the non-interactive setting. We remark that, although the knowledge soundness of LaBRADOR under the Fiat–Shamir transform requires a more sophisticated predicate tree of transcripts, as explained in [1], for our zero-knowledge layer the standard distinctness predicate already suffices.

Definition 8 ($\vec{k}$-Special Soundness). *Let* $\mu \in \mathbb{N}$ *and let* Π *be an* $2\mu + 1$*-move* RoK*. We say that* Π *is* $\vec{k}$*-special sound for* $\vec{k} \in \mathbb{N}^\mu$ *if there exists an efficient extractor that, given* $K = \prod_{i=1}^{\mu} k_i$ *accepting transcripts* $(\mathtt{trns}_\ell)_{\ell \in [K]}$ *arranged in a tree structure satisfying the following properties, outputs a valid witness:*

- *The tree has depth* μ*.*
- *Each leaf is labeled with one of the accepting transcripts.*
- *Each edge is labeled with a challenge, out edges from the same node have different labels. In particular, edges at level* j *are labeled with challenges from the* j*-th challenge space of* RoK*.*
- *Each node at level* j *is labeled with a partial transcript of length* j*, and all transcripts labeling leaves in the subtree rooted at that node share this partial transcript as a common prefix.*

Theorem 1 (Theorem 2, [6]). *The Fiat-Shamir transformation of a* $(k_1, \ldots, k_\mu)$*-special-sound* $(2\mu + 1)$*-move interactive* RoK*, in which all challenges are sampled from a set* $\mathcal{C}$ *of size* N*, is knowledge sound with knowledge error*

$$\kappa_{\mathsf{FS}}(Q) = (Q + 1)\kappa,$$

where $\kappa := Er((k_1, \ldots, k_\mu), N) = 1 - \prod_{i=1}^{\mu}(1 - \frac{k_i - 1}{N})$ *is the knowledge error of the interactive* RoK*.*

Notice that the theorem above holds for any *deterministic* adversary, without restrictions on its computational complexity. It also extends to probabilistic adversaries: one can always fix the adversary's random coins to the string that maximizes its advantage.

3.2 Composition for RoK

Given $\Pi_1 : \mathsf{R}_1 \to \mathsf{R}_2$ and $\Pi_2 : \mathsf{R}_2 \to \mathsf{R}_3$ that share the same setup algorithm Setup, we consider the following composed reduction-of-knowledge, which we denote with $\Pi_2 \circ \Pi_1 : \mathsf{R}_1 \to \mathsf{R}_3$.

- The common input is $\mathsf{pp}, \mathbb{x}_1$ and private input for the prover is $\mathbb{w}_1$.
- Execute $(\mathbb{x}_2, \mathbb{w}_2) \leftarrow \langle \mathsf{P}_1(\mathbb{w}_1), \mathsf{V}_1 \rangle(\mathsf{pp}, \mathbb{x}_1)$.
- Execute $(\mathbb{x}_3, \mathbb{w}_3) \leftarrow \langle \mathsf{P}_2(\mathbb{w}_2), \mathsf{V}_2 \rangle(\mathsf{pp}, \mathbb{x}_2)$.
- The prover outputs $(\mathbb{x}_3, \mathbb{w}_3)$ and the verifier outputs $\mathbb{x}_3$.

The work of [42] showed that if Π_1 and Π_2 are knowledge sound then $\Pi_2 \circ \Pi_1$ is knowledge sound. We can define a similar composition for non-interactive RoKs. However, we need to be slightly more careful when defining composition between non-interactive RoKs. Given two non-interactive reductions of knowledge $(\Pi_1)_{\mathsf{FS}}$ and $(\Pi_2)_{\mathsf{FS}}$, obtained using the Fiat-Shamir transform on Π_1 and Π_2 respectively, we define their non-interactive composition as the Fiat-Shamir transform of $\Pi_2 \circ \Pi_1$. In symbols:

$$(\Pi_2)_{\mathsf{FS}} \circ (\Pi_1)_{\mathsf{FS}} := (\Pi_2 \circ \Pi_1)_{\mathsf{FS}}.$$

Notice that this composition is different from the naive composition of NRoKs. The reason is that the challenges for Π_2 in $(\Pi_2 \circ \Pi_1)_{\mathsf{FS}}$ are defined using the string $\mathbb{x}_1 \| \pi_1$, where $\mathbb{x}_1$ and π_1 are the instance and proof for Π_1. We can describe this composition in an equivalent formulation using the concept of *signature of knowledge* [21]. In particular, the composition above is equivalent to the procedure where we compose $(\Pi_1)_{\mathsf{FS}}$ with a non-interactive RoK $\Pi_2 : \mathsf{R}'_2 \to \mathsf{R}_3$ where the input instance to Π_2 is defined as $\mathsf{msg}, \mathbb{x}_2$ with $\mathsf{msg} = \mathbb{x}_1 \| \pi_1$ and where we define the *augmented relation* R'_2 as follows:

$$\forall \mathsf{msg} \in \{0,1\}^* : \mathsf{R}_2(\mathsf{pp}, \mathbb{x}_2, \mathbb{w}_2) \iff \mathsf{R}'_2(\mathsf{pp}, (\mathsf{msg}, \mathbb{x}_2), \mathbb{w}_2).$$

It is easy to see that the folklore variant of the Fiat–Shamir transform that includes the message msg in the hash turns an interactive reduction of knowledge $\Pi_2 \colon \mathsf{R}_2 \to \mathsf{R}_3$ into a non-interactive reduction of knowledge denoted by $(\Pi_2)_{\mathsf{FS}^*} \colon \mathsf{R}'_2 \to \mathsf{R}_3$. This construction is commonly called a *signature of knowledge* for R_2. It is straightforward to show that this transform preserves simulation extractability. More precisely, we view $(\Pi_2)_{\mathsf{FS}^*}$ as a *signature reduction of knowledge* from R_2 to R_3 . For more details, see Section C.

4 Zero-Knowledge for RoK

We now define zero knowledge for reductions of knowledge. We introduce a new definition called partial zero-knowledge, or partial-ZK for short. The main idea is that the zero knowledge simulator for the RoK only needs to simulate the proof π and does not need to simulate the private output $\mathbb{w}_2$.

Definition 9 (Notions of Zero Knowledge for RoK in the ROM). *We say that Π is (strong statistical) zero knowledge, written ZK (resp. partial zero-knowledge, written partial-ZK), if and only if there exists a polynomial time simulator $\mathcal{S}$ such that for every adversary $\mathcal{A}$ and for every sequence of parameters $(\mathsf{pp}_\lambda)_{\lambda \in \mathbb{N}}$ with $\mathsf{pp}_\lambda \leftarrow \mathsf{Setup}(1^\lambda)$, the advantage $\mathsf{Adv}^{\mathsf{ZK}}_{\mathcal{A},\mathcal{S}}(\lambda)$ (resp. $\mathsf{Adv}^{\mathsf{pZK}}_{\mathcal{A},\mathcal{S}}(\lambda)$) is negligible. For the partial version, the games for the advantage $\mathsf{Adv}^{\mathsf{pZK}}_{\mathcal{A},\mathcal{S}}(\lambda)$ are defined in the same way, except that the oracles do not run the commands shown in gray in Fig. 3. The advantage is defined as follows:*

$$\mathsf{Adv}^{\mathsf{ZK}}_{\mathcal{A},\mathcal{S}}(\lambda) := \left| \Pr\left[\mathcal{A}^{\mathsf{O}_{\mathsf{RO}}, \mathsf{O}'_{\mathcal{S}}}(\mathsf{pp}) = 1\right] - \Pr\left[\mathcal{A}^{\mathsf{RO}, \mathsf{O}_{\mathsf{P}}}(\mathsf{pp}) = 1\right] \right|$$

We assume the state of the simulator $\mathsf{st}_{\mathcal{S}}$ contains the partial function of the reprogrammed random oracle queries, in particular one can always parse $\mathsf{st}_{\mathcal{S}}$ as $(\mathcal{Q}, z)$ where $\mathcal{Q}$ is the partial function of reprogrammed oracle query and z is an arbitrary string. By abuse of notation, we denote $\mathsf{RO}[\mathsf{st}_{\mathcal{S}}]$ to be $\mathsf{RO}[\mathcal{Q}]$.

We remark that, although in the security experiments in Fig. 3 the partial-ZK simulator $\mathcal{S}$ is defined to output both $\tilde{\pi}$ and $\tilde{\mathbb{w}}_2$, we can assume that $\tilde{\mathbb{w}}_2$ can be an arbitrary string, for example the null string $\perp$. Indeed, as a direct consequence of the definition of partial-ZK, the value $\tilde{\mathbb{w}}_2$ does not need to be a valid witness for $\tilde{\mathbb{x}}_2 = \mathsf{V}(\mathsf{pp}, \mathbb{x}_1, \tilde{\pi})$.

Zero-Knowledge for Interactive RoK. Recall that the focus of this paper is to analyze non-interactive RoKs in the ROM. However, since virtually all RoKs we are aware of rely on the Fiat–Shamir transform to go from interactive to non-interactive, we slightly abuse terminology and say that an interactive RoK Π is zero-knowledge if its Fiat–Shamir transform Π_{FS} is zero-knowledge.

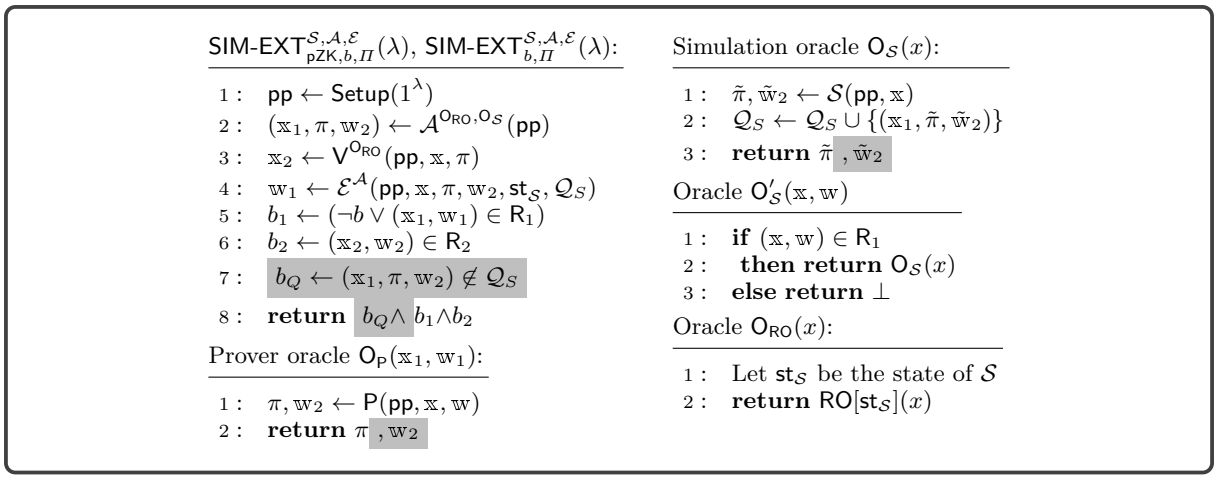


Fig. 3: The SIM-EXT games and the oracles for the zero-knowledge games, the games w.r.t. a partial-ZK simulator do not execute the gray-highlighted steps. We assume the state of the simulator $\mathsf{st}_\mathcal{S}$ contains the partial function of the reprogrammed random oracle queries.

A common recipe is that if the interactive RoK is honest-verifier zero-knowledge, then its Fiat–Shamir transform is zero-knowledge. However, we can introduce a more granular notion of honest-verifier zero-knowledge by assuming that the simulator needs to know the challenges only for the last k rounds [24, 34].

Let $\mathcal{S}$ be an honest-verifier zero-knowledge simulator for the interactive $2\mu+1$-move RoK Π, and let $k \in [\mu]$. We consider the following *k-canonical simulator associated with $\mathcal{S}$* for the non-interactive RoK $\Pi_{\mathsf{FS}} : \mathsf{R}_1 \to \mathsf{R}_2$:

- Parse the state $\mathsf{st}_\mathcal{S}$ as partial function $\mathcal{Q}$ (empty at first invocation).
- Sample $c_{k+1}, \ldots, c_\mu \xleftarrow{\$} \{0,1\}^\kappa$ uniformly at random.
- Run $\pi = (a_1, \ldots, a_{\mu+1}), \mathbb{w}_2 \leftarrow \mathcal{S}^{\mathsf{RO}[\mathcal{Q}]}(\mathbb{x}, (c_j)_{j \in [k+1,\mu]})$.
- Return $\pi, \mathbb{w}_2$, update the state $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(\bar{a}_j, c_j) : j > k\}$, where $\bar{a}_i$ defined as in Eq. (1) for the Fiat-Shamir transform.

Above we let $\mathcal{S}$ have oracle access to RO to define correctly the challenges for the first k rounds.

Definition 10. *We say that Π is k-zero-knowledge (resp. partial k-zero-knowledge), denoted k-ZK (resp. k-pZK), if there exists a PT algorithm $\mathcal{S}$ such that Π_{FS} is zero-knowledge (resp. partial k-zero-knowledge) where the simulator is the k-canonical simulator associated with $\mathcal{S}$. Moreover, we say that a $2\mu+1$-move Π is* non-reprogramming *partial zero-knowledge if it is μ-pZK.*

Notice that the notion of non-reprogramming applies only to partial zero-knowledge, as a non-reprogramming (strong) zero-knowledge simulator would break the knowledge soundness of the scheme.

4.1 On the Zero-Knowledge of the Composition

Before presenting our composition results, we note a simple syntactic transformation that can be applied to reductions of knowledge.

Definition 11. *Let $\tilde{\mathsf{R}}$ and R be relations. We say that $\tilde{\mathsf{R}}$ is an* extended relation associated *to R iff*

$$\tilde{\mathbb{x}} = (\mathbb{x}, \mathbb{w}'), \tilde{\mathbb{w}} \in \tilde{\mathsf{R}} \iff \mathbb{x}, \mathbb{w} = (\mathbb{w}', \tilde{\mathbb{w}}) \in \mathsf{R}.$$

An extended relation simply moves a part of the (private) witness into the (public) instance. Clearly, if Π_1 is a RoK from R_1 to R_2, then it is also a RoK from R_1 to $\tilde{\mathsf{R}}_2$, with only a minimal syntactic change. Namely, the prover sends $\mathbb{w}'$ to the verifier instead of outputting it. Similarly, with a simple syntactic transformation, if Π_2 is a RoK for $\mathsf{R}_2 \to \mathsf{R}_3$, then it is also a RoK for $\tilde{\mathsf{R}}_2 \to \mathsf{R}_3$. In this case, the verifier receives $\mathbb{w}'$ as part of its input (and its Fiat-Shamir transform does not include $\mathbb{w}'$ in the hashed values).

Notice that, while this syntactic change does not significantly affect the specification of the reduction of knowledge, the notions of zero-knowledge and partial zero-knowledge are quite sensitive to this change.

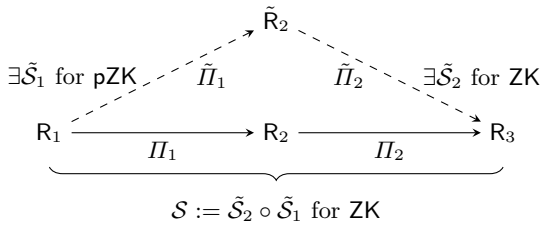


Fig. 4: A diagram illustrating Item 2 of Theorem 2. It shows that the natural composition of the zero-knowledge simulators for the RoKs to and from an extended relation $\tilde{\mathsf{R}}_2$ yields a valid zero-knowledge simulator for $\Pi_2 \circ \Pi_1$.

In particular, a partial zero-knowledge simulator for $\tilde{\Pi}_1 : \mathsf{R}_1 \to \tilde{\mathsf{R}}_2$ is more powerful than a partial zero-knowledge simulator for $\Pi_1 : \mathsf{R}_1 \to \mathsf{R}_2$, since it needs to additionally simulate $\mathbb{w}'$. Conversely, a zero-knowledge simulator for $\tilde{\Pi}_2 : \tilde{\mathsf{R}}_2 \to \mathsf{R}_3$ is less powerful than a zero-knowledge simulator for $\Pi_2 : \mathsf{R}_2 \to \mathsf{R}_3$, because it already receives $\mathbb{w}'$ as part of its input. Whenever it is clear from the context, we will simply say that Π_1 is zero-knowledge *for (the target relation)* $\tilde{\mathsf{R}}_2$ (resp. that Π_2 is zero-knowledge *for (the source relation)* $\tilde{\mathsf{R}}_2$), with the understanding that in the first case $\tilde{\mathsf{R}}_2$ is the target relation, whereas in the second case it is the source relation. The theorem below shows two composition results based on the zero-knowledge properties of the RoKs. In particular, it shows that we can compose the zero-knowledge simulators for $\tilde{\Pi}_1 : \mathsf{R}_1 \to \tilde{\mathsf{R}}_2$ and $\tilde{\Pi}_2 : \tilde{\mathsf{R}}_2 \to \mathsf{R}_3$ and obtain a zero-knowledge simulator for $\Pi_2 \circ \Pi_1$ (see also Fig. 4).

Theorem 2. *Let $\Pi := \Pi_2 \circ \Pi_1$ be the composition of $\Pi_1 : \mathsf{R}_1 \to \mathsf{R}_2$ and $\Pi_2 : \mathsf{R}_2 \to \mathsf{R}_3$.*

1. *If Π_1 is zero-knowledge then Π is zero-knowledge.*
2. *If there exists $\tilde{\mathsf{R}}_2$ an extended relation of R_2 such that Π_1 is partial zero-knowledge for $\tilde{\mathsf{R}}_2$ and Π_2 is zero-knowledge (resp. partial zero-knowledge) for $\tilde{\mathsf{R}}_2$ then Π is zero-knowledge (resp. partial zero-knowledge).*

Proof (Proof sketch). For the first part of the theorem, the construction is immediate: the simulator first invokes the simulator for Π_1, derives the intermediate instance $\mathbb{x}_2$ using the public reducibility function, and then runs the honest prover of Π_2. Since only the first component is simulated, indistinguishability follows directly from the zero-knowledge property of Π_1.

For the second part, the simulator combines the simulators $\tilde{\mathcal{S}}_1$ and $\tilde{\mathcal{S}}_2$ of $\tilde{\Pi}_1$ and $\tilde{\Pi}_2$, respectively. The main subtlety is that $\tilde{\Pi}_1$ provides only partial zero knowledge for the target relation $\tilde{\mathsf{R}}_2$, while $\tilde{\Pi}_2$ provides full zero knowledge for the same relation. The composed simulator first simulates a transcript of $\tilde{\Pi}_1$, producing a distribution identical to that of Π_1, and then invokes the simulator for $\tilde{\Pi}_2$ on the instance $\tilde{\mathbb{x}}_2 = (\mathbb{x}_2, \mathbb{w}'_2)$, producing a transcript distributed identically to that of Π_2 (see Fig. 4).

Proof. We start with the first case of the lemma. We define the simulator $\mathcal{S}$ of Π, let $\mathcal{S}_1$ be the zero-knowledge simulator for Π_1.

Simulator $\mathcal{S}(\mathsf{pp}, \mathbb{x}_1; \mathsf{st}_{\mathcal{S}})$:
1. Run $\mathcal{S}_1(\mathsf{pp}, \mathbb{x}_1; \mathsf{st}_{\mathcal{S}})$ which outputs $(\mathbb{x}, \pi_1, \mathbb{w}_1)$ and a new state $\mathsf{st}_{\mathcal{S}}$.
2. Run verifier $\mathsf{V}_1(\mathsf{pp}, \mathbb{x}_1, \pi_1)$ which outputs $\mathbb{x}_2$.
3. Run $(\mathsf{P}_1)_{\mathsf{FS}^*}(\mathsf{pp}, (\mathbb{x}_2, \pi_1), \mathbb{w}_2)$ which outputs $(\pi_2, \mathbb{w}_3)$.
4. Output $(\mathbb{x}, \pi_1 || \pi_2, \mathbb{w}_3)$ and the state $\mathsf{st}_{\mathcal{S}}$.

The indistinguishability of $\mathcal{S}$ follows directly from the property of $\mathcal{S}_1$, and therefore we omit the formal proof.

We prove the second case of the lemma when Π_2 is zero-knowledge, the case when Π_2 is partial-ZK is very similar. We define the simulator $\mathcal{S}$ of Π. There exist k_1 and k_2 such that Π_1 is k_1-pZK and Π_2 is k_2-ZK. Moreover, we can assume that $k_2 = 0$, because k_2-ZK simulator is also 0-ZK simulator for any k_2. Let $\tilde{\mathcal{S}}_1$ and $\tilde{\mathcal{S}}_2$ be the honest-verifier zero-knowledge simulators for $\tilde{\Pi}_1 : \mathsf{R}_1 \to \tilde{\mathsf{R}}_2$ and $\tilde{\Pi}_2 : \tilde{\mathsf{R}}_2 \to \mathsf{R}_1$. Notice that $\tilde{\mathcal{S}}_2$ takes in input $\mathbb{x}_2, w'_2$ where $\mathbb{w}_2 = w'_2, w''_2$ and outputs $\pi_2, \mathbb{w}_3$. Consider the following honest-verifier zero-knowledge simulator for the composed Π:

Simulator $(\mathcal{S}')^{\mathsf{RO}}(\mathsf{pp}, \mathbb{x}_1, c_{k_1+1}, \ldots, c_{r_1+r_2})$:
1. Run $(\pi_1, \mathbb{w}_1 = (w'_1, w''_2)) \leftarrow \mathcal{S}_1^{\mathsf{RO}}(\mathsf{pp}, \mathbb{x}_1, c_{k_1+1}, \ldots, c_{r_1})$.

2. Run the public reducibility function $\mathbb{x}_2 \leftarrow f_1(\mathsf{pp}, \mathbb{x}_1, \mathtt{trns})$ for Π_1 where $\mathtt{trns} = (a_1, c_1, \ldots, a_{r_1}, c_{r_1})$ and $\forall i \leq k_1 : c_i = \mathsf{RO}(\mathbb{x}_1, a_1, \ldots, a_i)$.
3. Run $\mathcal{S}_2(\mathsf{pp}, (\mathbb{x}_2, w_1'), c_{r_1+1}, \ldots, c_{r_1+r_2})$ which outputs $(\pi_2, \mathbb{w}_2)$.
4. Output $(\mathbb{x}, \pi_1 || \pi_2, \mathbb{w}_2)$.

We let $\mathcal{S}$ be the k_1-canonical simulator associated to $\mathcal{S}'$ for Π.

Let $\mathbf{H}_0$ be the output of an adversary $\mathcal{A}$ with oracle access to O_{RO} and $\mathsf{O}'_{\mathcal{S}}$ as in Definition 9. Consider the hybrid game $\mathbf{H}_1$ that is the same as $\mathbf{H}_0$ but where the simulation oracle $\mathsf{O}'_{\mathcal{S}}$ on input $(\mathbb{x}_1, \mathbb{w}_1)$ computes the following:

- $c_{r_1+1}, \ldots, c_{r_1+r_2} \xleftarrow{\$} \{0,1\}^\lambda$,
- $\pi_1, \mathbb{w}_2 \leftarrow \mathsf{P}_1^{\mathsf{RO}}(\mathsf{pp}, \mathbb{x}_1, \mathbb{w}_1)$,
- $\pi_2, \mathbb{w}_3 \leftarrow \mathcal{S}_2(\mathsf{pp}, (\mathbb{x}_2, w_2'), c_{r_1+1}, \ldots, c_{r_1+r_2})$,
- $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(\bar{a}_{r_1+j}, c_{r_1+j}) : j = 1, \ldots, r_2\}$,
- return $\pi_1 \| \pi_2, \mathbb{w}_3$ and set the state to $\mathcal{Q}$

Claim. There exists $\mathcal{A}_1$ such that $|\Pr[\mathbf{H}_0] - \Pr[\mathbf{H}_1]| \leq \mathsf{Adv}^{\mathsf{pZK}}_{\mathcal{A}_1, \tilde{\mathcal{S}}_1}(\lambda)$.

The adversary $\mathcal{A}_1$ has oracle to either $\mathsf{O}'_{\tilde{\mathcal{S}}_1}$ or $\mathsf{O}_{\mathsf{P}_1}$ runs $\mathcal{A}$ and answers its queries of the form $(\mathbb{x}_1, \mathbb{w}_1)$ by calling its oracle and obtaning π_1, w_2' and then running $\tilde{\mathcal{S}}_2$ on inputs $\mathbb{x}_2, w_2'$ and forwarding $\pi_1 \| \pi_2, \mathbb{w}_3$ to $\mathcal{A}$. Clearly, the hybrids are emulated perfectly.

Let $\mathbf{H}_1$ be the output of $\mathcal{A}$ with oracle access to O_{RO} and O_{P}.

Claim. There exists $\mathcal{A}_2$ such that $|\Pr[\mathbf{H}_0] - \Pr[\mathbf{H}_1]| \leq \mathsf{Adv}^{\mathsf{ZK}}_{\mathcal{A}_2, \tilde{\mathcal{S}}_2}(\lambda)$.

The adversary $\mathcal{A}_2$ has oracle to either $\mathsf{O}'_{\tilde{\mathcal{S}}_2}$ or $\mathsf{O}_{\mathsf{P}_2}$ runs $\mathcal{A}$ and answers its queries of the form $(\mathbb{x}_1, \mathbb{w}_1)$ by first running P_1 and then calling its oracle on $(\mathbb{x}_2, w_2')$ and obtaning $\pi_2, \mathbb{w}_3$ and forwarding $\pi_1 \| \pi_2, \mathbb{w}_3$ to $\mathcal{A}$. Clearly, the hybrids are emulated perfectly.

5 Simulation Extractability for RoK

We give two definitions of simulation extractability for RoKs. The first notion is equivalent to the classical notion of simulation extractability. We obtain it by extending the standard definition for proofs of knowledge to the syntax of RoKs.

We also introduce a weaker notion, tailored to partial zero-knowledge. Since simulation extractability depends on the zero-knowledge simulator, this variant considers adversaries that only see proofs π and are not given access to the corresponding simulated witnesses $\mathbb{w}_2$. We recall, that although the partial zero-knowledge simulator outputs $\pi, \tilde{\mathbb{w}}_2$, the latter value may be arbitrary, for example it could be $\perp$, as partial zero-knowledge does not require simulated proofs to be valid proofs for the associated proof of knowledge, but only that π_1 is distributed as in the real proofs.

Moreover, since the adversary does not see full simulated proofs, it cannot carry out the trivial copy–paste attack; more precisely, such an attack is no longer straightforward. For this reason, we no longer check whether the forged statement–proof pair appears in the set of queried statements and simulated proof pairs.

Definition 12 (Simulation Extractability). *Let $\Pi := (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ be a non-interactive reduction of knowledge for $\mathsf{R}_1 \to \mathsf{R}_2$. We say that Π is simulation extractable, $\mathsf{SIM\text{-}EXT}$ in short, (with respect to simulator $\mathcal{S}$) if there exists an EPT extractor $\mathcal{E}$ such that for every PPT adversary $\mathcal{A}$, the following probability is negligible in λ:*

$$\mathsf{Adv}^{\mathsf{SE}}(\Pi, \mathcal{S}, \mathcal{E}, \mathcal{A}) := \left| \Pr\left[\mathsf{SIM\text{-}EXT}^{\mathcal{S},\mathcal{A},\mathcal{E}}_{0,\Pi}(\lambda)\right] - \Pr\left[\mathsf{SIM\text{-}EXT}^{\mathcal{S},\mathcal{A},\mathcal{E}}_{1,\Pi}(\lambda)\right] \right|$$

for games defined in Fig. 3.

Moreover, we say that that Π is simulation-extractable w.r.t. partial-ZK, $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ in short, if the following is negligible in λ:

$$\mathsf{Adv}^{\mathsf{SE}}_{\mathsf{pZK}}(\Pi, \mathcal{S}, \mathcal{E}, \mathcal{A}) := \left| \Pr\left[\mathsf{SIM\text{-}EXT}^{\mathcal{S},\mathcal{A},\mathcal{E}}_{\mathsf{pZK},0,\Pi}(\lambda)\right] - \Pr\left[\mathsf{SIM\text{-}EXT}^{\mathcal{S},\mathcal{A},\mathcal{E}}_{\mathsf{pZK},1,\Pi}(\lambda)\right] \right|.$$

Let Π be a public-coin interactive reduction of knowledge, we say that Π is simulation extractable (resp. simulation extractable w.r.t. partial-ZK) if Π_{FS} is simulation extractable (resp. simulation extractable w.r.t. partial-ZK).

Game k-$\mathsf{UR}^{\mathcal{A}}_{\Pi_{\mathsf{FS}}}(\lambda)$

$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathbb{x}, (\pi, \mathbb{w}), (\pi', \mathbb{w}'), c) \leftarrow \mathcal{A}^{\mathsf{RO}}(\mathsf{pp})$
$\mathcal{Q}_c \leftarrow \{(\mathsf{pp} \| \mathbb{x} \| \pi_{|k}) \mapsto c\}$
$b_V \leftarrow \bar{\mathsf{V}}^{\mathsf{RO}[\mathcal{Q}_c]}(\mathsf{pp}, \mathbb{x}, \pi, \mathbb{w})$
$b'_V \leftarrow \bar{\mathsf{V}}^{\mathsf{RO}[\mathcal{Q}_c]}(\mathsf{pp}, \mathbb{x}, \pi', \mathbb{w}')$
$b \leftarrow (\pi, \mathbb{w}) \neq (\pi', \mathbb{w}') \wedge \pi_{|k} = \pi'_{|k}$
return $b_V \wedge b'_V \wedge b$

Game k-$\mathsf{HtoInv}^{\mathcal{A},\mathcal{S}}_{\Pi}(\lambda)$

$\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
$(\mathbb{x}, \pi, \mathbb{w}_2) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{RO}}, \mathsf{O}_{\mathcal{S}}}(\mathsf{pp})$
$b \leftarrow \exists \tilde{\pi} : (\mathbb{x}, \tilde{\pi}, \cdot) \in \mathcal{Q}_S \wedge \pi_{|k} = \tilde{\pi}_{|k}$
$b_V \leftarrow \bar{\mathsf{V}}^{\mathsf{O}_{\mathsf{RO}}}(\mathsf{pp}, v, \pi, \mathbb{w}_2)$
return $b_V \wedge b$

Fig. 5: Games k-UR and k-HtoInv. For the latter game, we consider the same oracles $\mathsf{O}_{\mathcal{S}}$ and O_{RO} of the $\mathsf{SIM\text{-}EXT}_w$ game, namely the oracles defined in Fig. 3 that does not execute the gray-highlighted steps.

It is easy to check that if $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ is SIM-EXT, in the reduction of knowledge sense, then the associated PoK $\Sigma = (\mathsf{Setup}, \mathsf{P}, \bar{\mathsf{V}})$ is simulation extractable in the standard sense.

SIM-EXT for zero-knowledge for extended relation. When considering extended relations, and in particular when the extended relation is the source relation, namely when $\tilde{\Pi} : \tilde{\mathsf{R}}_1 \to \mathsf{R}_2$, we can consider an adversary that has oracle access to the zero-knowledge simulator for $\tilde{\Pi}$, but must produce a forgery for $\Pi : \mathsf{R}_1 \to \mathsf{R}_2$. Namely, we are interested in the advantage $\mathsf{Adv}^{\mathsf{SE}}(\Pi, \tilde{\mathcal{S}}, \mathcal{E}, \mathcal{A})$, where $\tilde{\mathcal{S}}$ is the zero-knowledge simulator for $\tilde{\Pi}$.

5.1 Unique Response and Hard to Invert Properties

Previous work [24, 33, 34, 45] on the simulation extractability of Fiat–Shamir based proofs and arguments of knowledge follows a recipe that originated in [31]. In this approach, knowledge soundness together with two additional properties is enough to prove simulation extractability. The first of these two properties concerns the soundness of the scheme and is called unique response. The other property concerns the zero knowledge of the scheme. Here we recall these properties and put forward a new property, in line with unique-response, that is sufficient to prove simulation extractability w.r.t. partial zero-knowledge.

As useful notation, let Π be a FS-based NRoK derived from a $2\mu + 1$-move RoK, give a proof $\pi = (a_1, \ldots, a_\mu)$ for Π, for $k \leq \mu$, we denote with $\pi_{|k}$ the prefix $(a_1, \ldots, a_k)$ of π.

Definition 13 (k-Unique Response). *Let $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V}) : \mathsf{R}_1 \to \mathsf{R}_2$ be a $(2\mu + 1)$-move public-coin reduction of knowledge, with Π_{FS} its associated FS-transformed NRoK and $k \in [0, \mu]$. We say that Π_{FS} is k-unique response, in short k-UR, if for any PPT adversary $\mathcal{A}$, the probability $\mathbf{Adv}^{k\text{-UR}}_{\Pi_{\mathsf{FS}}}(\mathcal{A}) := \Pr\left[k\text{-}\mathsf{UR}^{\mathcal{A}}_{\Pi_{\mathsf{FS}}}(\lambda)\right] = \mathsf{negl}(\lambda)$ for the game defined in Fig. 5.*

It sometimes occurs that one analyzes the UR property on small reductions and is interested in the uniqueness of their composition.

Lemma 11. *Let $\Pi_1 : \mathsf{R}_1 \to \mathsf{R}_2$, $\Pi_2 : \mathsf{R}_2 \to \mathsf{R}_3$ be reductions of knowledge such that $(\Pi_1)_{\mathsf{FS}}$ is k-UR, $(\Pi_2)_{\mathsf{FS}}$ is 0-UR. Then it holds that $\Pi_{\mathsf{FS}} := (\Pi_2 \circ \Pi_1)_{\mathsf{FS}}$ is k-UR. Concretely, $\mathbf{Adv}^{k\text{-UR}}_{\Pi_{\mathsf{FS}},\mathcal{A}}(\lambda) \leq \mathbf{Adv}^{k\text{-UR}}_{(\Pi_1)_{\mathsf{FS}}}(\mathcal{A}) + \mathbf{Adv}^{0\text{-UR}}_{(\Pi_2)_{\mathsf{FS}}}(\mathcal{A})$.*

Proof (sketch). Let Π_1 be a $(2\mu_1 + 1)$-move RoK with $\mu_1 \geq k$. Let $\mathcal{A}$ be an adversary against k-UR of Π_{FS}, it outputs $(\mathbb{x}, (\pi = (\pi_1 \| \pi_2), \mathbb{w}_2), (\pi' = (\pi'_1 \| \pi'_2), \mathbb{w}'_2), c)$ such that the $\pi_{|k} = \pi'_{|k}$ and there exists a smallest $l > k$ such that $\pi_{|l} \neq \pi'_{|l}$. Now there are two cases. First assume that $l < \mu_1$, in this case one may emulate an adversary against k-UR of $(\Pi_1)_{\mathsf{FS}}$ with the same winning probability as $\mathcal{A}$. Next assume that $l \geq \mu_1$, in this case one may emulate an adversary against 0-UR of $(\Pi_2)_{\mathsf{FS}}$ with the same winning probability as $\mathcal{A}$.

The main theorem in [24] (Theorem 3.4) induces immediately the following statement on RoK:

Theorem 3. *Let $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V}) : \mathsf{R}_1 \to \mathsf{R}_2$ be a $(2\mu + 1)$-move public-coin reduction of knowledge, with Π_{FS} its associated FS-transformed NRoK and $k \in [1, \mu]$. Assume Π_{FS} satisfies KS, k-ZK and k-UR (for the same k). Then Π_{FS} satisfies SIM-EXT.*

Definition 14 (k-hard-to-invert RoK). *Given PT relations* R_1 *and* R_2*, and* $\Pi : \mathsf{R}_1 \to \mathsf{R}_2$ *a partial zero-knowledge* $(2\mu+1)$*-move public-coin reduction of knowledge. Let* $\mathcal{S}$ *be a simulator for partial-ZK of* Π*. We say that* Π_{FS} *is* k*-hard-to-invert,* k-HtoInv *in short, for* $k \le \mu$*, if for all PPT adversaries* $\mathcal{A}$*, the advantage* $\mathsf{Adv}^{k\text{-}\mathtt{HI}}_{\Pi,\mathcal{A}}(\lambda) := \Pr\left[k\text{-}\mathsf{HtoInv}^{\mathcal{A},\mathcal{S}}_{\Pi_{\mathsf{FS}}}(\lambda) = 1\right]$ *is negligible.*

At an intuitive level, the relation between hard to invert and unique response in reduction of knowledge is similar to the relation between one way functions and collision resistant hash functions. For unique response, the adversary must find two different suffixes that *collide* to the same prefix while forming valid proofs. In contrast, for hard to invert, the adversary must find a valid suffix given a prefix[7].

Our next theorem is in the same spirit as results showing that the unique-response property together with zero-knowledge and knowledge soundness imply simulation extractability after applying the Fiat–Shamir transform. In particular, prior work has shown that protocols satisfying a unique-response condition become simulation-extractable in the random-oracle model after Fiat–Shamir [24, 34]. Here we tailor such a result to the setting of simulation-extractability with respect to partial zero-knowledge. In this case, we show that the weaker notion of k-hard-to-invert (which is weaker than k-unique-response) is sufficient.

Theorem 4. *Let* $\Pi : \mathsf{R}_1 \to \mathsf{R}_2$ *be a partial zero-knowledge RoK with* $\mathcal{S}_{\mathsf{pZK}}$ *as a simulator. If* Π *satisfies* KS*,* k-pZK *and* k*-hard-to-invert, then* Π *is* $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$*. In particular, for any PT* $\mathcal{A}$ *and any extractor* $\mathcal{E}$*, there exist PT* $\mathcal{B}$ *and* $\mathcal{A}_{\mathsf{KS}}$ *such that*

$$\mathsf{Adv}^{\mathtt{SE}}_{\mathsf{pZK}}(\mathcal{S},\mathcal{E},\mathcal{A}) \le 2 \cdot \mathsf{Adv}^{k\text{-}\mathtt{HI}}_{\Pi,\mathcal{B}} + \mathbf{Adv}^{\mathtt{KS}}(\Pi,\mathcal{E},\mathcal{A}_{\mathsf{KS}})$$

Moreover, if $\mathcal{S}_{\mathsf{pZK}}$ *is non-reprogramming and* Π *is* KS*, then* Π *is* $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$*.*

Proof (Proof sketch). Let $\mathcal{A}$ be an adversary against $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$. The proof proceeds by separating the cases in which the adversary outputs a forgery whose verification queries hit a random-oracle point reprogrammed by the simulator from the cases in which they do not.

We first consider the general case. Starting from the $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ game with bit 0, we define a hybrid that aborts whenever the forgery $(\mathbb{x}_1, \pi, \mathbb{w}_2)$ has a k-prefix query $(\mathsf{pp}\|\mathbb{x}_1\|\pi_{|k})$ that was reprogrammed by the simulator. The probability of this abort event is bounded by the hard-to-invert property: indeed, such an event exactly means that the adversary has produced a valid witness for a proof whose relevant prefix comes from a simulated proof.

Conditioned on this bad event not occurring, the simulation can be embedded into a knowledge-soundness adversary. This adversary runs $\mathcal{A}$, answers its simulation queries by locally executing the partial zero-knowledge simulator, and forwards random-oracle queries consistently, using the programmed answers whenever necessary. When $\mathcal{A}$ outputs a forgery, the knowledge-soundness adversary outputs the same forgery, unless the forbidden reprogrammed-prefix event occurs.

The key point is that, in this hybrid, the verifier's random-oracle queries on the forged proof are not reprogrammed. Indeed, by the k-zero-knowledge structure of the simulator, reprogramming only occurs after the relevant prefix, and the hybrid explicitly excludes forgeries whose k-prefix was reprogrammed. Thus, the forgery is valid with respect to the real random oracle, and the hybrid is semantically equivalent to the knowledge-soundness game. By knowledge soundness, there exists an extractor that recovers a source witness for $\mathbb{x}_1$ whenever the adversary succeeds.

Finally, if the partial zero-knowledge simulator is non-reprogramming, the argument simplifies. In this case there is no bad event to exclude: the simulated random oracle and the real random oracle coincide everywhere. Hence any valid forgery produced in the $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ game is immediately a valid forgery for knowledge soundness. Running the knowledge-soundness extractor therefore directly yields the required extractor for $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$.

Proof. Let $\mathcal{A}$ be an adversary against $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$, let $\mathbf{H}_0$ be the game $\mathsf{SIM\text{-}EXT}^{\mathcal{S},\mathcal{A}_{\mathsf{pZK}}}_{\mathsf{pZK},0,\Pi_{\mathsf{FS}}}$, and let $\mathbf{H}_1$ be identical to $\mathbf{H}_0$, except that it returns 0 when $\exists(\mathsf{pp}\|\mathbb{x}_1\|\tilde{\pi}) \in \mathcal{Q} : \pi|_k = \tilde{\pi}|_k$ where $\mathsf{st}_{\mathcal{S}} = (\mathcal{Q}, z)$ and $(\mathbb{x}_1, \pi, \mathbb{w}_2)$ is the forgery of the adversary.

Consider the bad event $\mathsf{bad} := \mathbf{H}_0 \wedge \exists(\mathsf{pp}\|\mathbb{x}_1\|\tilde{\pi}) \in \mathcal{Q} : \pi|_k = \tilde{\pi}|_k$. It is easy to see that

$$|\Pr[\mathbf{H}_0] - \Pr[\mathbf{H}_1]| \le \Pr[\mathsf{bad}] \le \mathsf{Adv}^{k\text{-}\mathtt{HI}}_{\Pi,\mathcal{A}}(\lambda),$$

[7] We could define hard to invert property by sampling the prefix in different ways. For example, we could use the honest prover or the strong zero knowledge simulator. We choose to use the partial zero-knowledge simulator because our goal is to show that hard to invert is sufficient for $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$.

since $\mathcal{A}$ is a valid adversary for the hard-to-invert game and, moreover, the bad event corresponds to the winning condition of the hard-to-invert game.

Now, we define an adversary $\mathcal{A}_{\mathsf{KS}}$ against knowledge soundness as follows:

Adversary $\mathcal{A}_{\mathsf{KS}}^{\mathsf{RO}}(\mathsf{pp})$:

1. Initialize an empty set $\mathcal{Q}$ of programmed random oracle queries by $\mathcal{S}_{\mathsf{pZK}}$ and the initial state st for the simulator $\mathcal{S}$.
2. Run $\mathcal{A}_{\mathsf{pZK}}(\mathsf{pp})$
   Reply to its simulation queries as follows:
   - Upon simulation query x from $\mathcal{A}$ to $\mathcal{S}_{\mathsf{pZK}}$, run locally $(\pi, w) \leftarrow \mathcal{S}_{\mathsf{pZK}}(\mathsf{st}, \mathbb{x})$ and forward π. (W.o.l.g. we assume that any programmed RO queries are added in $\mathcal{Q}$.)

   Reply to its RO queries as follows:
   - If the query is in $\mathcal{Q}$, $\mathcal{A}_{\mathsf{KS}}$ forwards the programmed query to $\mathcal{A}$. Otherwise, $\mathcal{A}_{\mathsf{KS}}$ queries RO on the same thing and forwards the result to $\mathcal{A}$.

   Eventually, $\mathcal{A}$ outputs $(\mathbb{x}, \pi, \mathbb{w}_2)$.
3. If $(\mathsf{pp} \| \mathbb{x} \| \pi_{|k}) \in \mathcal{Q}$, abort;
4. Else output the forgery $(\mathbb{x}, \pi, \mathbb{w}_2)$.

We notice that $\mathbf{H}_1 = 1$ is equivalent to:

- $(\mathsf{pp} \| \mathbb{x}_1 \| \pi_{|k}) \notin \mathcal{Q}$,
- $\mathsf{V}^{\mathsf{RO}[\mathsf{st}_{\mathcal{S}}]}(\mathsf{pp}, \mathbb{x}_1, \pi) \to \mathbb{x}_2 \neq \bot \wedge (\mathbb{x}_2, \mathbb{w}_2) \in \mathcal{R}_2$

which means that:

- $\mathcal{A}_{\mathsf{KS}}$ does not abort,
- $\mathsf{V}^{\mathsf{RO}}(\mathsf{pp}, \mathbb{x}_1, \pi) \to \mathbb{x}_2 \neq \bot$, this is because $(\mathsf{pp}, \mathbb{x}_1, \pi)$ shares no common k-prefix with any simulated proof $\mathcal{A}$ queried, because of the previous item. Moreover, by k-ZK, the simulator does not reprogram the RO until the k-th round, which implies the $\mathsf{RO}[\mathsf{st}_{\mathcal{S}}]$ gives the same result as RO on the queries by the verifier on input $\mathbb{x}_1, \pi$. Namely, $\forall i : (\mathsf{pp} \| \mathbb{x}_1 \| \pi_{|i}) \notin \mathcal{Q}$.

Hence $\mathbf{H}_1$ and $\mathsf{KS}_0^{\mathcal{A}_{\mathsf{KS}}}$ are semantically equivalent. By assumption on knowledge soundness, there's exists a PT extractor $\mathcal{E}_{\mathsf{KS}}$ that, by interacting with $\mathcal{A}_{\mathsf{KS}}$, outputs a witness w such that $(\mathbb{x}, w) \in \mathcal{R}_1$ with overwhelming probability when $\mathcal{A}_{\mathsf{KS}}$ outputs a valid forgery.

From $\mathcal{A}_{\mathsf{KS}}$ and $\mathcal{E}_{\mathsf{KS}}$ we can define the extractor $\mathcal{E}$ for $\mathcal{A}$. Namely, $\mathcal{E}$ is the machine that runs $\mathcal{E}_{\mathsf{KS}}$ with input (or oracle access) to $\mathcal{A}_{\mathsf{KS}}$, recalling that the latter can be defined from $\mathcal{A}$. Let $\mathbf{H}_2$ be the same as $\mathbf{H}_1$ but it additionally runs $\mathcal{E}_{\mathsf{KS}}$ and outputs 0 if the extractor fails. Notice that $\mathbf{H}_3$ is semantically equivalent to $\mathsf{KS}_1^{\mathcal{A}_{\mathsf{KS}}}$, thus, because of the knowledge soundness of Π we have

$$|\Pr[\mathbf{H}_1] - \Pr[\mathbf{H}_2]| \leq \mathbf{Adv}^{\mathtt{KS}}(\Pi, \mathcal{E}_{\mathsf{KS}}, \mathcal{A}_{\mathsf{KS}}).$$

Finally, we define the last hybrid $\mathbf{H}_3$ to be the game $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK},1,\Pi}^{\mathcal{S},\mathcal{A},\mathcal{E}}$. The difference between the two hybrids is that $\mathbf{H}_2$ additionally check for the event bad, while $\mathbf{H}_3$ does not. Thus, by the same argument for $\mathbf{H}_0 \rightsquigarrow \mathbf{H}_1$ we have

$$|\Pr[\mathbf{H}_1] - \Pr[\mathbf{H}_2]| \leq \mathsf{Adv}_{\Pi,\mathcal{A}}^{k\text{-}\mathtt{HI}}(\lambda).$$

Now consider the case where the partial zero-knowledge simulator does not reprogram the random oracle. Similarly as before, let $\mathcal{A}$ be an adversary against $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$. Consider the following adversary against knowledge soundness:

Adversary $\mathcal{A}_{\mathsf{KS}}(\mathsf{pp})$

1. Emulate $\mathcal{A}$ and answer its simulation queries by running locally the simulator oracle $\mathcal{S}$, eventually it outputs forgery $(\mathbb{x}, \pi, \mathbb{w}_2)$
2. output $(\mathbb{x}, \pi, \mathbb{w}_2)$

Notice that since the simulator $\mathcal{S}$ does not reprogram the random oracle, we have $\bar{\mathsf{V}}^{\mathsf{RO}}(\mathsf{pp}, \mathbb{x}, \pi, \mathbb{w}_2) = 1$, thus the output of $\mathcal{A}_{\mathsf{KS}}$ is a valid forgery when $\mathcal{A}$'s output is, and $\mathsf{KS}_{0,\Pi}^{\mathcal{A}_{\mathsf{KS}}} \equiv \mathsf{SIM\text{-}EXT}_{\mathsf{pZK},0,\Pi}^{\mathcal{S},\mathcal{A}}$. As before, we define the extractor $\mathcal{E}$ to run $\mathcal{E}_{\mathsf{KS}}$ with $\mathcal{A}_{\mathsf{KS}}$, where the latter adversary can be defined using $\mathcal{A}$. It is easy to see that $\mathsf{KS}_{1,\Pi}^{\mathcal{A}_{\mathsf{KS}}} \equiv \mathsf{SIM\text{-}EXT}_{\mathsf{pZK},1,\Pi}^{\mathcal{S},\mathcal{A},\mathcal{E}}$. □

Remark 1. Notice that the hard to invert property is necessary in the previous theorem. Consider a reduction of knowledge $\Pi\colon \mathsf{R}_1 \to \mathsf{R}_2$ where R_2 is decidable in polynomial time and R_1 is NP hard. Even if we assume that Π satisfies $\mathsf{SIM\text{-}EXT}$ (and therefore knowledge soundness), we can still break $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$. The attack works as follows. The adversary queries a simulated proof for an arbitrary $\mathbb{x}_1$ in the language of R_1 and obtains $\tilde{\pi}$. Since R_2 is decidable in polynomial time, the adversary can compute a witness $\mathbb{w}_2$ for $\mathbb{x}_2 = f(\mathbb{x}_1, \tilde{\pi})$ and then output $(\mathbb{x}_1, \tilde{\pi}, \mathbb{w}_2)$. This is essentially a copy-paste attack, which is possible because R_2 is decidable in polynomial time. Hard-to-invert property states that such copy-paste attacks are not viable.

The previous remark shows that there exist RoKs that are $\mathsf{SIM\text{-}EXT}$ but not necessarily $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$. The next lemma shows that this somewhat unexpected gap can be filled thanks to the hard-to-invert property.

Lemma 12. *Let $\Pi : \mathsf{R}_1 \to \mathsf{R}_2$ be a $2\mu+1$-move zero-knowledge reduction of knowledge. If Π is $\mathsf{SIM\text{-}EXT}$ and μ-hard-to-invert, then it's $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$.*

Proof (Proof sketch). Let $\mathcal{S}$ be a zero-knowledge simulator for Π. Since a zero-knowledge simulator also induces a partial zero-knowledge simulator, we can use the same simulator in the $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ game.

The proof proceeds by relating any adversary $\mathcal{A}_{\mathsf{pZK}}$ against $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ to an adversary $\mathcal{A}$ against standard $\mathsf{SIM\text{-}EXT}$. The only obstruction is that, in the $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ game, the adversary sees only the partial simulated proof and may later output a full tuple $(\mathbb{x}, \pi, \mathbb{w}_2)$ corresponding to one of its own simulation queries[8]. We first rule out this case using the hard-to-invert property.

More precisely, we introduce a hybrid that aborts whenever the forgery $(\mathbb{x}, \pi, \mathbb{w}_2)$ belongs to the set $\mathcal{Q}_S$ of simulated proof-witness tuples generated by $\mathcal{S}$. The probability of this abort event is bounded by the hard-to-invert advantage, since such an event means that $\mathcal{A}_{\mathsf{pZK}}$ recovers a valid target witness for a simulated proof.

Conditioned on this event not occurring, we build an adversary $\mathcal{A}$ against standard $\mathsf{SIM\text{-}EXT}$ as follows. The adversary $\mathcal{A}$ internally runs $\mathcal{A}_{\mathsf{pZK}}$, forwards its simulation queries to the zero-knowledge simulator, but returns only the proof component to $\mathcal{A}_{\mathsf{pZK}}$, as in the partial zero-knowledge experiment. When $\mathcal{A}_{\mathsf{pZK}}$ outputs a forgery, $\mathcal{A}$ outputs the same forgery, unless it is one of the simulated tuples, in which case it aborts. By construction, this experiment is distributed exactly as the hybrid above.

Since Π is standard simulation-extractable, there exists an extractor $\mathcal{E}$ for $\mathcal{A}$. We then define an extractor $\mathcal{E}_{\mathsf{pZK}}$ for $\mathcal{A}_{\mathsf{pZK}}$ by running $\mathcal{E}$ on the wrapped adversary $\mathcal{A}$. This gives a valid source witness whenever the hybrid forgery is accepting and not a simulated tuple.

Proof. Let $\mathcal{S}$ be a zero-knowledge simulator for Π, notice that $\mathcal{S}$ is also a partial zero-knowledge simulator for Π. Let $\mathcal{A}_{\mathsf{pZK}}$ be an adversary against $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$, $\mathbf{H}_0$ be the game $\mathsf{SIM\text{-}EXT}^{\mathcal{S},\mathcal{A}_{\mathsf{pZK}}}_{\mathsf{pZK},0,\Pi_{\mathsf{FS}}}$ and let $\mathbf{H}_1$ be identical to $\mathbf{H}_0$, except that it returns 0 when $(\mathbb{x}, \pi, \mathbb{w}_2) \in \mathcal{Q}_S$.
Consider the bad event $\mathsf{bad} := \mathbf{H}_0 \wedge (\mathbb{x}, \pi, \mathbb{w}_2) \in \mathcal{Q}_S$. The adversary $\mathcal{A}_{\mathsf{pZK}}$ itself wins the hard-to-invert security experiment with probability $\Pr[\mathsf{bad}]$. Thus,

$$|\Pr[\mathbf{H}_0 = 1] - \Pr[\mathbf{H}_1 = 1]| \leq \mathsf{Adv}^{k\text{-}\mathtt{HI}}_{\Pi,\mathcal{A}_{\mathsf{pZK}}}(\lambda).$$

Consider the adversary against $\mathsf{SIM\text{-}EXT}$ of Π defined below:

Adversary $\mathcal{A}(\mathsf{pp})$:
1. Run $\mathcal{A}_{\mathsf{pZK}}(\mathsf{pp})$
2. Forward simulation oracle queries from $\mathcal{A}_{\mathsf{pZK}}$ to $\mathsf{O}_{\mathcal{S}}$, upon answer $(\pi, \mathbb{w}_2)$ from the oracle return π to $\mathcal{A}_{\mathsf{pZK}}$.
3. Forward random oracle queries from and to $\mathcal{A}_{\mathsf{pZK}}$.
4. If $(\mathbb{x}, \pi, \mathbb{w}_2) \in \mathcal{Q}_S$, abort;
5. Else output the forgery $(\mathbb{x}, \pi, \mathbb{w}_2)$.

We notice that $\mathbf{H}_1$ and $\mathsf{SIM\text{-}EXT}^{\mathcal{S},\mathcal{A},\mathcal{E}}_{0,\Pi}$ are semantically equivalent (for any $\mathcal{E}$, because the output bit does not depend on the extraction bit b_1 when $b = 0$).

By assumption on $\mathsf{SIM\text{-}EXT}$, there exists a PT extractor $\mathcal{E}$ that, by interacting with $\mathcal{A}$, outputs a witness w such that $(\mathbb{x}, w) \in \mathsf{R}_1$ with probability almost the same as $\mathbf{H}_1$ returning 1. Thus, we construct

[8] While, we recall that in the $\mathsf{SIM\text{-}EXT}$ game, the adversary is not allowed to output as its forgery a simulated proof.

a polynomial time extractor $\mathcal{E}_{\mathsf{pZK}}$ that runs $\mathcal{E}$ on $\mathcal{A}$, the extractor $\mathcal{E}_{\mathsf{pZK}}$ outputs exactly what $\mathcal{E}$ outputs. Notice, the adversary $\mathcal{A}$ is itself defined from $\mathcal{A}_{\mathsf{pZK}}$, thus $\mathcal{E}_{\mathsf{pZK}}$ needs only to know $\mathcal{A}_{\mathsf{pZK}}$ to run.

Finally we let $\mathbf{H}_2$ be the game that additionally runs $\mathcal{E}_{\mathsf{pZK}}$ with adversary $\mathcal{A}_{\mathsf{pZK}}$, namely, it computes $\mathbb{w}_1 \leftarrow \mathcal{E}_{\mathsf{pZK}}^{\mathcal{A}_{\mathsf{pZK}}}(\mathsf{pp}, \mathbb{x}, \pi, \mathbb{w}_2)$ and sets $b_1 \leftarrow (\mathbb{x}_1, \mathbb{w}_1) \in \mathsf{R}_2$, and finally outputs the conjunction of the output of $\mathbf{H}_1$ and b_2, namely $b_1 \wedge b_2$ (recall, $b_2 = (\mathbb{x}_2, \mathbb{w}_2) \in \mathsf{R}_2$). This hybrid is semantically equivalent to run the (standard) simulation-extractability game with $\mathcal{S}, \mathcal{A}_{\mathsf{pZK}}$ and $\mathcal{E}_{\mathsf{pZK}}$ and the bit $b = 1$, namely $\mathsf{SIM\text{-}EXT}_{1,\Pi}^{\mathcal{S},\mathcal{A}_{\mathsf{pZK}},\mathcal{E}_{\mathsf{pZK}}}$.

The last hybrid $\mathbf{H}_3$ is the same as $\mathbf{H}_2$ but we do not check anymore that $(\mathbb{x}_1, \pi, \mathbb{w}_2) \notin \mathcal{Q}_S$. Equivalently to the transition from $\mathbf{H}_0$ and $\mathbf{H}_1$, the hybrids are indistinguishable by hard-to-invert property. Finally notice that $\mathbf{H}_3$ is equivalent to the partial simulation-extractability game, namely $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK},1,\Pi}^{\mathcal{S},\mathcal{A}_{\mathsf{pZK}},\mathcal{E}_{\mathsf{pZK}}}$. By the triangular equality, there exist $\mathcal{E}$and $\mathcal{E}_{\mathsf{pZK}}$ such that for any PT $\mathcal{A}_{\mathsf{pZK}}$ there exists PT $\mathcal{A}$ such that:

$$\mathsf{Adv}_{\mathsf{pZK}}^{\mathtt{SE}}(\mathcal{S}_{\mathsf{pZK}}, \mathcal{E}_{\mathsf{pZK}}, \mathcal{A}_{\mathsf{pZK}}) \leq 2 \cdot \mathsf{Adv}_{\Pi,\mathcal{A}_{\mathsf{pZK}}}^{k\text{-}\mathtt{HI}}(\lambda) + \mathsf{Adv}^{\mathtt{SE}}(\mathcal{S}, \mathcal{E}, \mathcal{A}).$$

6 Non-Malleable Composition of RoK

Theorem 5. *Let $\Pi_1 : \mathsf{R}_1 \to \mathsf{R}_2$ and $\Pi_2 : \mathsf{R}_2 \to \mathsf{R}_3$ be two public-coin RoKs. We denote the composed RoK $\Pi := \Pi_2 \circ \Pi_1$. Then the following statements hold:*

1. *If Π_1 is zero-knowledge and simulation extractable, Π_2 is knowledge sound and* 0-UR*, then the composed RoK Π is zero-knowledge and simulation extractable.*
2. *If there exists $\tilde{\mathsf{R}}_2$ extended relation of R_2 such that if Π_1 is partial zero-knowledge for target relation $\tilde{\mathsf{R}}_2$ and partial simulation extractable, Π_2 is zero-knowledge for $\tilde{\mathsf{R}}_2$ for source relation and simulation extractable, then the composed RoK Π is zero-knowledge and simulation extractable.*
3. *If Π_1 and Π_2 are (partial) zero-knowledge and simulation extractable w.r.t. partial-ZK then the composed RoK Π is simulation extractable w.r.t. partial-ZK.*

Proof (Proof sketch).

First Statement. The simulator for Π is the same defined in the first case of Theorem 2, namely first run $\mathcal{S}_1$ and then run P_2. We first extract a witness for the intermediate instance using the knowledge soundness of Π_2. To do so, we build an adversary $\mathcal{A}_2$ against Π_2 that internally runs the adversary $\mathcal{A}$ against the composed protocol. Simulation queries of $\mathcal{A}$ are answered by locally running $\mathcal{S}_1$ and then the honest prover of Π_2. If $\mathcal{A}$ outputs a valid forgery for the composed protocol, then the suffix proof π_2 is a valid forgery for Π_2 with respect to the intermediate instance $\mathbb{x}_2$ produced by verifying π_1. Hence, by knowledge soundness of Π_2, an extractor $\mathcal{E}_2$ recovers a witness $\mathbb{w}_2$ for $\mathbb{x}_2$.

The next step rules out the case where the prefix proof π_1 together with the extracted witness $\mathbb{w}_2$ coincides with one of the simulated outputs of $\mathcal{S}_1$. If this happens, then the adversary has produced a different accepting continuation for the same intermediate witness. This yields two accepting proofs for Π_2 with the same source witness, contradicting the 0-unique-response property of Π_2. Thus the corresponding hybrid transition is bounded by the 0-UR advantage of Π_2.

Once this case is excluded, we can use the simulation extractability of Π_1. We build an adversary $\mathcal{A}_1$ against Π_1 that internally runs $\mathcal{A}$, answers its simulation queries using the simulator for Π_1, and completes them using the honest prover for Π_2. When $\mathcal{A}$ outputs a valid forgery, the extractor $\mathcal{E}_2$ first recovers $\mathbb{w}_2$ for the intermediate instance; then $\mathcal{A}_1$ outputs $(\mathbb{x}_1, \pi_1, \mathbb{w}_2)$ as a forgery for Π_1. Simulation extractability of Π_1 gives an extractor $\mathcal{E}_1$ that recovers a valid witness $\mathbb{w}_1$ for the original instance $\mathbb{x}_1$.

Combining the two extractors gives the extractor for the composed protocol: on input a forged proof $\pi_1 \| \pi_2$, it first runs $\mathcal{E}_2$ to extract the intermediate witness $\mathbb{w}_2$, and then runs $\mathcal{E}_1$ to extract the source witness $\mathbb{w}_1$. The final hybrid is therefore exactly the SIM-EXT game for the composed protocol with extraction enabled, proving the first statement.

Second Statement. The simulator for Π is the same defined in the second case of Theorem 2, namely first run $\tilde{\mathcal{S}}_1$ (to $\tilde{\mathsf{R}}_2$) and then run $\tilde{\mathcal{S}}_2$. The proof proceeds by hybrids, but now the suffix protocol Π_2 is simulated rather than honestly generated. The first transition uses the simulation extractability of the signature-of-knowledge derived using the Fiat-Shamir transform on Π_2: from any forgery for the composed protocol we obtain a forgery for Π_2 on the intermediate instance, with the prefix $\mathbb{x}_1 \| \pi_1$ treated as the tag or statement-dependent context. This yields an extractor $\mathcal{E}_2$ for the intermediate witness $\mathbb{w}_2$ and also enforces that the suffix forgery was not one of the simulated suffix proofs. The second transition then uses the partial simulation extractability of Π_1. Given the extracted $\mathbb{w}_2$, the prefix $(\mathbb{x}_1, \pi_1, \mathbb{w}_2)$ is

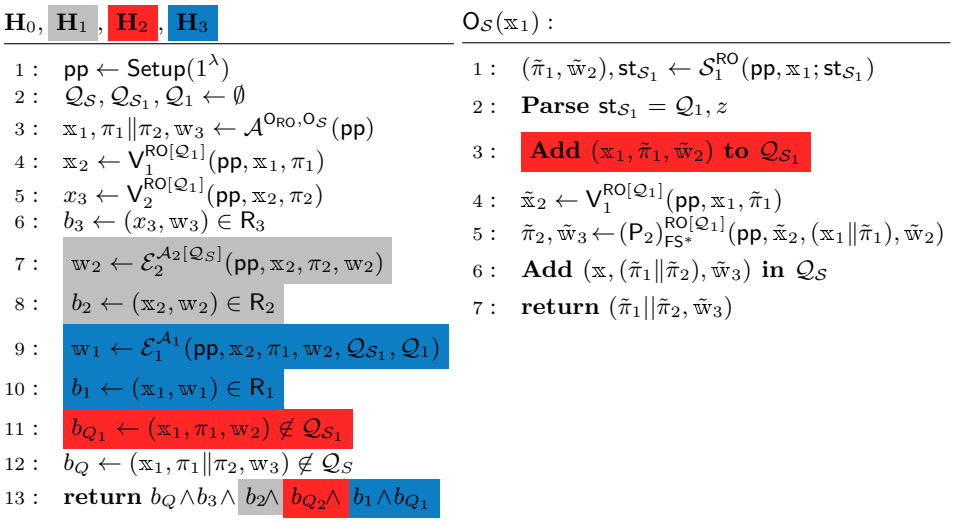


Fig. 6: Hybrid games for the proof of the first statement. $\mathbf{H}_0$ executes only non-colored operations, $\mathbf{H}_1$ additionally executes gray colored operations, $\mathbf{H}_2$ additionally executes red colored operations and $\mathbf{H}_3$ additionally executes blue colored operations.

a valid forgery for Π_1. Thus the extractor $\mathcal{E}_1$ for Π_1 recovers a source witness $\mathbb{w}_1$ for $\mathbb{x}_1$. As before, the extractor for the composed protocol is obtained by first running $\mathcal{E}_2$ and then $\mathcal{E}_1$.

The third statement follows by the same argument as the second one. The only difference is that the reduction to the suffix protocol uses partial simulation extractability for Π_2 as well. Consequently, the hybrid does not need the additional check excluding membership in the simulator query set for the suffix proof, since the partial simulation extractability game already ignores this part of the simulated output. The rest of the extractor construction and the hybrid argument are unchanged.

Proof. We prove the four statements.

First Statement. We prove that the transitions of games described in Fig. 6 are indistinguishable. Let $\mathcal{S}_1$ be the zero-knowledge simulator for Π_1 and let $\mathcal{S}$ be the simulator derived by $\mathcal{S}_1$ as in Theorem 2. For reference, we describe the simulator $\mathcal{S}$ in Fig. 6. Clearly, $\mathbf{H}_0$ is equivalent to $\mathsf{SIM\text{-}EXT}_{0,\Pi}^{\mathcal{S},\mathcal{A},\mathcal{E}}$ for any $\mathcal{E}$.

Let $\mathbf{H}_1$ be as defined in Fig. 6. Namely, $\mathbf{H}_1$ is identical to $\mathbf{H}_0$, except that it additionally runs the extractor of Π_2 against an adversary defined below and sets the bit b_2 according to the outcome of the knowledge-soundness experiment for Π_2.

Claim. There exist EPT extractor $\mathcal{E}_2$ and a non-uniform PT adversary $\mathcal{A}_2$ such that for any polynomial-sized set $\bar{\mathcal{Q}}_S$, let $\mathcal{A}_2[\bar{\mathcal{Q}}_S]$ be the adversary $\mathcal{A}_2$ with hardcoded the description of the set $\bar{\mathcal{Q}}_S$, we have:

$$|\Pr(\mathbf{H}_0[r_S] = 1 | \mathcal{Q}_S = \bar{\mathcal{Q}}_S) - \Pr(\mathbf{H}_1 = 1 || \mathcal{Q}_S = \bar{\mathcal{Q}}_S)| \leq \mathbf{Adv}^{\mathsf{KS}}(\Pi_2, \mathcal{E}_2, \mathcal{A}_2[\bar{\mathcal{Q}}_S])$$

Notice that, assuming Π_2 has negligible knowledge error, by averaging over all polynomial-sized sets $\mathcal{Q}_S$ we have

$$|\Pr(\mathbf{H}_0 = 1) - \Pr(\mathbf{H}_1 = 1)| \leq \Pr_{\mathcal{Q}_S}\left[\mathbf{Adv}^{\mathsf{KS}}(\Pi_2, \mathcal{E}_2, \mathcal{A}_2[\mathcal{Q}_S])\right] \leq \mathsf{negl}(\lambda).$$

Where the random variable $\mathcal{Q}_S$ in the probability above is sampled accordingly to the distribution defined by running $\mathcal{A}$ with the simulator $\mathcal{S}$ for Π.

Proof. The difference between the two hybrids is that in $\mathbf{H}_1$ we additionally extract from an adversary $\mathcal{A}_2[\bar{\mathcal{Q}}_S]$ a witness $\mathbb{w}_2$ and check that the extracted witness is valid for the forged instance. We define a PT adversary $\mathcal{A}_2[\bar{\mathcal{Q}}_S]$ against knowledge soundness of Π_2.

Adversary $\mathcal{A}_2^{\mathsf{RO}}[\bar{\mathcal{Q}}_S](\mathsf{pp})$:

1. Without loss of generality, we assume that $\bar{\mathcal{Q}}_S$ contains only valid simulated proofs. Hence, for every $(\mathbb{x}_1, \bar{\pi}) \in \bar{\mathcal{Q}}_S$, we can parse $\bar{\pi} = (a_1, c_1, \ldots, a_\mu, c_\mu)$ for some $\mu \in \mathbb{N}$, where the c_i are the challenges. For each i, either $c_i = \mathsf{RO}(\mathsf{pp} \| \mathbb{x}_1 \| a_1 \| \cdots \| a_i)$, in which case the simulated proof does not require programming the random oracle, or c_i is a value programmed to make the proof accepting. We then define $\mathcal{Q}_1$ to be the set of random-oracle queries programmed in this way and derived from the proofs in $\bar{\mathcal{Q}}_S$.

2. Initialize an empty set $\mathcal{Q}_{\mathcal{S}_1}$ of simulated proofs by $\mathcal{S}_1$, and the initial state $\mathsf{st}_{\mathcal{S}_1}$ for the simulator $\mathcal{S}_1$ initially to be empty.
3. Run the adversary $\mathcal{A}^{\mathsf{RO}[\mathcal{Q}_1]}(\mathsf{pp})$ and reply to its simulation queries as follows:
   - Upon the i-th simulation query $\mathbb{x}_1$ from $\mathcal{A}$, if the i-th element of $\bar{\mathcal{Q}}_S$ is $(\mathbb{x}_1, \bar{\pi}, \bar{\mathbb{w}}_3)$ return $\bar{\pi}, \bar{\mathbb{w}}_3$ to $\mathcal{A}$. Else, run locally $\mathcal{S}_1$, run the prover of Π_2, and add the tuple $(\mathbb{x}, (\tilde{\pi}_1 \| \tilde{\pi}_2), \mathbb{w}_3)$ in $\mathcal{Q}_{\mathcal{S}_1}$. In more details, run locally lines from 1 to 7 (excluding 3) of $\mathsf{O}_{\mathcal{S}}$ described in Fig. 6 assuming the set of simulated proofs fixed to $\bar{\mathcal{Q}}_{\mathcal{S}}$, unless the adversary never queried such input statement (this would happen only if the adversary has been rewound by $\mathcal{E}_2$ in step 9). Notice that, by the definition of $\bar{a}_j$ (see Eq. (1)) for any j the ROM query by Π_1 and Π_2 are separated.
   - Upon random oracle query, reply the query according to $\mathsf{RO}[\mathcal{Q}_1]$.
4. Eventually, $\mathcal{A}$ outputs $(\mathbb{x}_1, \pi_1 \| \pi_2, \mathbb{w}_3)$. If $(\mathbb{x}_1, \pi_1 \| \pi_2, \mathbb{w}_3) \in \mathcal{Q}_S$ output $\perp$, else output the forgery $(\mathbb{x}_2, \pi_2, \mathbb{w}_3)$ where $\mathbb{x}_2 \leftarrow \mathsf{V}_1^{\mathsf{RO}[\mathcal{Q}_1]}(\mathsf{pp}, \mathbb{x}_1, \pi_1)$.

By construction of $\mathcal{A}_2$, the game $\mathbf{H}_0$ conditioned on $\mathcal{Q}_S = \bar{\mathcal{Q}}_S$ is semantically equivalent to the knowledge-soundness game $\mathsf{KS}_{0,\Pi_2}^{\mathcal{A}_2[\bar{\mathcal{Q}}_S]}$. Indeed, KS_0 runs the adversary $\mathcal{A}_2[\bar{\mathcal{Q}}_S]$ once, and since $\mathcal{A}$ is deterministic, all simulation queries made by $\mathcal{A}$ in the emulation performed by $\mathcal{A}_2$ belong to the fixed set $\bar{\mathcal{Q}}_S$. Moreover, by state separation of the random-oracle queries of Π_1 and Π_2, any forgery produced by $\mathcal{A}$ that verifies with respect to $\mathsf{RO}[\mathcal{Q}_1]$, i.e., $\bar{\mathsf{V}}^{\mathsf{RO}[\mathcal{Q}_1]}(\mathsf{pp}, \mathbb{x}_1, (\pi, \mathbb{w}_3)) = 1$, also verifies with respect to the original random oracle RO when $\mathcal{Q}_S = \bar{\mathcal{Q}}_S$, namely $\bar{\mathsf{V}}^{\mathsf{RO}}(\mathsf{pp}, \mathbb{x}_1, (\pi, \mathbb{w}_3)) = 1$, where $\bar{\mathsf{V}}$ denotes the verifier of the associated PoK.

Similarly, $\mathbf{H}_1$ conditioned on $\mathcal{Q}_S = \bar{\mathcal{Q}}_S$ is semantically equivalent to $\mathsf{KS}_{1,\Pi_2}^{\mathcal{A}_2[\bar{\mathcal{Q}}_S]}$. As above, the emulated adversary $\mathcal{A}$ makes only simulation queries contained in $\bar{\mathcal{Q}}_S$, and therefore the output of $\mathcal{A}_2[\bar{\mathcal{Q}}_S]$ in the knowledge-soundness experiment coincides with the output of $\mathcal{A}$ in $\mathbf{H}_1$ conditioned on $\mathcal{Q}_S = \bar{\mathcal{Q}}_S$. Furthermore, when the extractor $\mathcal{E}_2$ interacts with $\mathcal{A}_2$ through oracle access, simulation queries are answered according to the fixed set $\bar{\mathcal{Q}}_S$, while any additional queries are answered exactly as $\mathcal{S}$ would in the sim-ext experiment for $\mathcal{A}$.

Hence, by setting $\mathcal{E}_2$ be the knowledge extractor of Π_2, the claim follows. □

The hybrid $\mathbf{H}_2$ is the same as the previous but additionally book-keeps the list of proofs simulated from $\mathcal{S}_1$ by populating the set $\mathcal{Q}_{\mathcal{S}_1}$, see line 3 of $\mathsf{O}_{\mathcal{S}}$ in Fig. 6. Notice if $(\mathbb{x}, \pi_1, \mathbb{w}_2) \in \mathcal{Q}_{\mathcal{S}_1}$ then $\exists \tilde{\pi}_2, \tilde{\mathbb{w}}_3 : (\mathbb{x}, \pi_1 \| \tilde{\pi}_2, \tilde{\mathbb{w}}_3) \in \mathcal{Q}_{\mathcal{S}}$ and $\tilde{\pi}_2, \tilde{\mathbb{w}}_3 = (\mathsf{P}_2)_{\mathsf{FS}^*}(\mathsf{pp}, \tilde{\mathbb{x}}_2, (\mathbb{x}_1 \| \pi_1), \mathbb{w}_2)$ (see lines 5,6 in Fig. 6).

Claim. There exists PT $\mathcal{B}$ such that:

$$|\Pr(\mathbf{H}_1 = 1) - \Pr(\mathbf{H}_2 = 1)| \leq \mathbf{Adv}^{\mathsf{0\text{-}UR}}(\Pi_2, \mathcal{B}).$$

Proof. We let the event $(\mathbb{x}_1, \pi_1, \mathbb{w}_2) \in \mathcal{Q}_{\mathcal{S}_1} \wedge (\mathbb{x}_1, \pi_1 \| \pi_2, \mathbb{w}_3) \notin \mathcal{Q}_{\mathcal{S}}$ be bad. We will show that $\Pr[\mathsf{bad}] \leq \mathbf{Adv}^{\mathsf{0\text{-}UR}}(\Pi_2, \mathcal{B})$ for an adversary $\mathcal{B}$ that we described below. Since $\mathbf{H}_1 \wedge \neg\mathsf{bad} \iff \mathbf{H}_2$ and $\mathbf{H}_2 \wedge \neg\mathsf{bad} \iff \mathbf{H}_2$, this implies the statement of the claim.

Adversary $\mathcal{B}(\mathsf{pp})$:
1. Initialize an empty set $\mathcal{Q}_{\mathcal{S}_1}$ of simulated proofs by $\mathcal{S}_1$, an empty set $\mathcal{Q}_1$ of programmed random oracle queries by $\mathcal{S}_1$.
2. Run the adversary $\mathcal{A}(\mathsf{pp})$ and reply to its simulation queries as follows:
   - Upon simulation query x from $\mathcal{A}$ to $\mathcal{S}$, run locally the simulator $\mathcal{S}$ (as in $\mathcal{A}_2$ of the previous claim).
   - Upon random oracle query, reply the query according to $\mathsf{RO}[\mathcal{Q}_1]$.
3. Eventually, $\mathcal{A}$ outputs $(\mathbb{x}_1, \pi_1 \| \pi_2, \mathbb{w}_3)$.
4. Run verifier $\mathsf{V}_1^{\mathsf{RO}[\mathcal{Q}_1]}(\mathsf{pp}, \mathbb{x}_1, \pi_1) \to \mathbb{x}_2$
5. Find in $\mathcal{Q}_{\mathcal{S}}$ simulated $(\mathbb{x}, \pi_1 || \tilde{\pi}_2, \tilde{\mathbb{w}}_3)$, abort if such query doesn't exist.
6. Output $(\mathbb{x}_2, (\pi_2, \mathbb{w}_3), (\tilde{\pi}_2, \tilde{\mathbb{w}}_3))$

Notice that both $(\pi_2, \mathbb{w}_3)$ and $(\tilde{\pi}_2, \tilde{\mathbb{w}}_3)$ are accepting proofs for $\mathbb{x}_2$ using $\mathsf{V}_2^{\mathsf{RO}[\mathcal{Q}_1]}$, moreover, by construction, none of the ROM queries needed to verify the proofs are in $\mathcal{Q}_1$ thus both proofs are accepting proof using $\mathsf{V}_2^{\mathsf{RO}}$. Thus, it suffices to show that the event bad implies $\mathcal{B}$ does not abort and $(\pi_2, \mathbb{w}_3) \neq (\tilde{\pi}_2, \tilde{\mathbb{w}}_3)$. Since $(\mathbb{x}_1, \pi_1, \mathbb{w}_2) \in \mathcal{Q}_{\mathcal{S}_1}$, there exists $(\mathbb{x}_1, \pi_1 || \tilde{\pi}_2, \tilde{\mathbb{w}}_3) \in \mathcal{Q}_{\mathcal{S}}$ and thus $\mathcal{B}$ does not abort. Moreover, since $(\mathbb{x}_1, \pi_1 || \tilde{\pi}_2, \tilde{\mathbb{w}}_3) \in \mathcal{Q}_{\mathcal{S}}$ and $(\mathbb{x}_1, \pi_1 || \pi_2, \mathbb{w}_3) \notin \mathcal{Q}_{\mathcal{S}}$ (because of bad), it must be that $(\pi_2, \mathbb{w}_3) \neq (\tilde{\pi}_2, \tilde{\mathbb{w}}_3)$. □

Let $\mathbf{H}_3$ be the same as $\mathbf{H}_2$ but where we additionally run an extractor $\mathcal{E}_1$ over an adversary $\mathcal{A}_1$ and we check that the extracted witness $\mathbb{w}_1$ is a valid witness for $\mathbb{x}_1$ (see lines 9,10,13 in Fig. 6).

Claim. There exist EPT extractor $\mathcal{E}_1$ and PT adversary $\mathcal{A}_1$ such that

$$|\Pr(\mathbf{H}_2 = 1) - \Pr(\mathbf{H}_3 = 1)| \leq \mathsf{Adv}^{\mathsf{SE}}(\Pi_1, \mathcal{S}_1, \mathcal{E}_1, \mathcal{A}_1)$$

Proof. We construct a PT adversary $\mathcal{A}_1$ against simulation-extractability of Π_1.

Adversary $\mathcal{A}_1^{\mathsf{O}_{\mathcal{S}_1}, \mathsf{O}_{\mathsf{RO}}}(\mathsf{pp})$:

1. Initialize an empty set $\mathcal{Q}_{\mathcal{S}}$ of simulated proofs.
2. Run the adversary $\mathcal{A}^{\mathsf{RO}}(\mathsf{pp})$ and reply to its queries as follows:
   - Upon simulation query $\mathbb{x}_1$, forward simulation query to $\mathsf{O}_{\mathcal{S}_1}$ which returns $\tilde{\pi}_1, \tilde{\mathbb{w}}_2$, compute a proof $\tilde{\pi}_2, \tilde{\mathbb{w}}_3$ as in lines 4,5 of the simulation oracle described in Fig. 6 and add the proof in $\mathcal{Q}_{\mathcal{S}}$, as in line 6.
3. Eventually $\mathcal{A}$ outputs the forgery $(\mathbb{x}_1, \pi_2, \mathbb{w}_2)$. Run extractor $\mathcal{E}_2^{\mathcal{A}_2[\mathcal{Q}_{\mathcal{S}}]}$ as in line 7 of Fig. 6, check bit $b_2 \leftarrow (\mathbb{x}_2, \mathbb{w}_2) \in \mathsf{R}_2$ as in line 8 and abort if $\neg b_2$. Return $(\mathbb{x}_1, \pi_1, \mathbb{w}_2)$.

Assuming that the extraction of $\mathcal{E}_2$ holds, whenever a forgery of $\mathcal{A}$ passes the verification algorithm for Π then the forgery of $\mathcal{A}_1$ passes the verification algorithm for Π_1. Moreover, for hybrid $\mathbf{H}_2$ to return 1 it is necessary that the extraction of $\mathcal{E}_2$ hold, and $\mathbf{H}_2$ checks that the forgery of $\mathcal{A}_1$ is not in $\mathcal{Q}_{\mathcal{S}_1}$ already. Thus $\mathsf{SIM\text{-}EXT}_{0,\Pi_1}^{\mathcal{S}_1,\mathcal{A}_1,\mathcal{E}_1}$ and $\mathbf{H}_2$ are semantically equivalent. By definition of $\mathbf{H}_3$, we have $\mathsf{SIM\text{-}EXT}_{1,\Pi_1}^{\mathcal{S}_1,\mathcal{A}_1,\mathcal{E}_1}$ and $\mathbf{H}_3$ are semantically equivalent. Thus we can conclude that the statement of the claim holds.

Claim. There exists EPT extractor $\mathcal{E}$ such that

$$\Pr(\mathbf{H}_3 = 1) = \Pr(\mathsf{SIM\text{-}EXT}_{1,\Pi}^{\mathcal{S},\mathcal{A},\mathcal{E}}(\lambda) = 1)$$

Proof. It suffices to construct such extractor algorithm $\mathcal{E}$.

Extractor $\mathcal{E}^{\mathcal{A}}(\mathsf{pp}, \mathbb{x}_1, \pi_1 || \pi_2, \mathbb{w}_2, \mathcal{Q}_{\mathcal{S}}, \mathcal{Q})$:

1. Run verifier $\mathbb{x}_1 \leftarrow \mathsf{V}_1^{\mathsf{RO}}(\mathsf{pp}, \mathbb{x}_1, \pi_1)$
2. Run extractor $\mathbb{w}_2 \leftarrow \mathcal{E}_2^{\mathcal{A}_2[\mathcal{Q}_S]}(\mathsf{pp}, \mathbb{x}_2, \pi_2, \mathbb{w}_3)$
3. Check $(\mathbb{x}_2, \mathbb{w}_2) \in \mathsf{R}_2$.
4. Define $\mathcal{Q}_{\mathcal{S}_1}$ from $\mathcal{Q}_{\mathcal{S}}$ by stripping off the Π_2 part.
5. Run extractor $\mathbb{w}_1 \leftarrow \mathcal{E}_1^{\mathcal{A}_1}(\mathsf{pp}, \mathbb{x}_1, \pi_1, \mathbb{w}_2, \mathcal{Q}_{\mathcal{S}_1}, \mathcal{Q})$
6. Output $\mathbb{w}_1$

Since the two games are semantically equivalent the claim holds. □

□

Second statement. We prove the transitions of games described in Fig. 7 are indistinguishable. Let $\mathcal{S}_1$ be the zero-knowledge simulator for $\Pi_1 : \mathsf{R}_1 \rightarrow \tilde{\mathsf{R}}_2$ and $\mathcal{S}_2$ be the zero-knowledge simulator for $\Pi_2 : \tilde{\mathsf{R}}_2 \rightarrow \mathsf{R}_3$ and let $\mathcal{S}$ be the simulator derived from $\mathcal{S}_1, \mathcal{S}_2$ as in Theorem 2. Clearly, $\mathbf{H}_0$ is equivalent to $\mathsf{SIM\text{-}EXT}_{0,\Pi}^{\mathcal{S},\mathcal{A},\mathcal{E}}(\lambda)$ for any $\mathcal{E}$.

Claim. There exist EPT extractor $\mathcal{E}_2$ and PT adversary $\mathcal{A}_2$ such that

$$|\Pr(\mathbf{H}_0 = 1) - \Pr(\mathbf{H}_1 = 1)| \leq \mathsf{Adv}^{\mathsf{SE}}((\Pi_2)_{\mathsf{FS}^*}, \mathcal{S}_2, \mathcal{E}_2, \mathcal{A}_2).$$

Proof. The hybrid $\mathbf{H}_1$ additionally extracts the witnesses from an adversary $\mathcal{A}_2$ and check the condition on b_{Q_2} (see Fig. 7 lines 6,7,8,1). We define a PT adversary $\mathcal{A}_2$ against simulation extractability of $(\Pi_2)_{\mathsf{FS}^*}$ (namely, the simulation extractability of the signature of knowledge scheme) as follows:

Adversary $\mathcal{A}_2^{\mathsf{RO}}(\mathsf{pp})$:

1. Initialize two empty sets $\mathcal{Q}_{\mathcal{S}_1}$, $\mathcal{Q}_{\mathcal{S}_2}$ of simulated proofs by $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively and empty set $\mathcal{Q}_1$, as in Fig. 7.
2. Run the adversary $\mathcal{A}^{\mathsf{RO}[\mathcal{Q}_1]}(\mathsf{pp})$ and reply to its queries as follow:
   - Upon simulation query x from $\mathcal{A}$ to $\mathcal{S}$, run locally the simulator $\mathcal{S}_1$, and the public-reducibility function, add $(\mathbb{x}, \pi_1)$ to $\mathcal{Q}_{\mathcal{S}_1}$, add any programmed RO queries to $\mathcal{Q}_1$. In more details, run locally the operations from 1 to 6 in $\mathsf{O}_{\mathcal{S}}$ in Fig. 7. Notice that, by the definition of the values $\bar{a}_j$ (see Eq. (1)) for any j the ROM queries by Π_1 and Π_2 are separated and thus we have $\mathcal{Q}_1 \cap \mathcal{Q}_2 = \emptyset$.
     Send query $(\tilde{\mathbb{x}}_2, \tilde{\mathbb{w}}_2')$ to its simulation oracle and forward the output to $\mathcal{A}$.
   - Upon random oracle query, reply the query according to $\mathsf{RO}[\mathcal{Q}_1]$.

$\mathbf{H}_0$, $\mathbf{H}_1$, $\mathbf{H}_2$

1 : $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$
2 : $\mathcal{Q}_S, \mathcal{Q}, \mathcal{Q}_1, \mathcal{Q}_2 \leftarrow \emptyset$
3 : $(\mathbb{x}_1, \pi_1 \| \pi_2, \mathbb{w}_3) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{RO}}, \mathsf{O}_{\mathcal{S}}}(\mathsf{pp})$
4 : $\mathbb{x}_2 \leftarrow \mathsf{V}_1^{\mathsf{RO}[\mathcal{Q}]}(\mathsf{pp}, \mathbb{x}_1, \pi_1)$
5 : $x_3 \leftarrow \mathsf{V}_2^{\mathsf{RO}[\mathcal{Q}]}(\mathsf{pp}, \mathbb{x}_2, \pi_2)$
6 : $b_3 \leftarrow (x_3, \mathbb{w}_3) \in \mathsf{R}_3$
7 : $\mathbb{w}_2 \leftarrow \mathcal{E}_2^{\mathcal{A}_2}(\mathsf{pp}, \tilde{\mathbb{x}}_2, \pi_2, \mathbb{w}_2)$
8 : $b_2 \leftarrow (\mathbb{x}_2, \mathbb{w}_2) \in \mathsf{R}_2$
9 : $b_{Q_2} \leftarrow (\mathbb{x}_2, (\mathbb{x}_1 \| \pi_1), \pi_2, \mathbb{w}_3) \notin \mathcal{Q}_{\mathcal{S}_2}$
10 : $\mathbb{w}_1 \leftarrow \mathcal{E}_1^{\mathcal{A}_1}(\mathsf{pp}, \mathbb{x}_2, \pi, \mathbb{w}_2)$
11 : $b_1 \leftarrow (\mathbb{x}_1, \mathbb{w}_1) \in \mathsf{R}_1$
12 : $b_Q \leftarrow (\mathbb{x}_1, \pi_1 \| \pi_2, \mathbb{w}_3) \notin \mathcal{Q}_S$
13 : **return** $b_Q \wedge b_3 \wedge b_2 \wedge b_{Q_2} \wedge b_1$

$\mathsf{O}_{\mathcal{S}}(\mathbb{x}_1)$:

1 : $\vec{c}_1 \leftarrow (c_j)_{j \in [k_1+1, r_1]} \xleftarrow{\$} \{0,1\}^\lambda$
2 : $(\tilde{\pi}_1, \tilde{\mathbb{w}}_2) \leftarrow \mathcal{S}_1^{\mathsf{RO}[\mathcal{Q}]}(\mathsf{pp}, \mathbb{x}_1, \vec{c}_1)$
3 : **Add** $\{(\bar{a}_j, c_j) : i \in [k_1+1, r_1]\}$ **to** $\mathcal{Q}_1$
4 : **Add** $(\mathbb{x}_1, \tilde{\pi}_1)$ **to** $\mathcal{Q}_{\mathcal{S}_1}$
5 : **let** $\tilde{\mathbb{w}}_2 = (\tilde{\mathbb{w}}_2', \tilde{\mathbb{w}}_2'')$
6 : $\tilde{\mathbb{x}}_2 \leftarrow f_1(\mathsf{pp}, \tilde{\mathbb{x}}_1, \tilde{\pi}_1)$
7 : $\vec{c}_2 \leftarrow (c_j)_{j \in [r_1+1, r_1+r_2]} \xleftarrow{\$} \{0,1\}^\lambda$
8 : $(\tilde{\pi}_2, \tilde{\mathbb{w}}_3) \leftarrow \mathcal{S}_2(\mathsf{pp}, (\tilde{\mathbb{x}}_2, \tilde{\mathbb{w}}_2'), \vec{c}_2)$
9 : **Add** $\{(\bar{a}_j, c_j) : i \in [r_1+1, r_2]\}$ **to** $\mathcal{Q}_2$
10 : **Add** $((\tilde{\mathbb{x}}_2, \tilde{\mathbb{w}}_2'), (\tilde{\mathbb{x}}_1 \| \tilde{\pi}_1), \tilde{\pi}_2, \tilde{\mathbb{w}}_3)$ **to** $\mathcal{Q}_{\mathcal{S}_2}$
11 : $\mathcal{Q} \leftarrow \mathcal{Q}_1 \cup \mathcal{Q}_2$
12 : **return** $(\tilde{\pi}_1 || \tilde{\pi}_2, \tilde{\mathbb{w}}_3)$

Fig. 7: Hybrid games for the proof of the second statement. $\mathbf{H}_0$ executes only non-colored operations, $\mathbf{H}_1$ additionally executes gray colored operations, $\mathbf{H}_2$ additionally executes blue colored operations. The values $\bar{a}_j$ correspond to $\mathsf{pp} \| \mathbb{x}_1 \| a_1, \ldots a_j$ as in Eq. (1).

3. Eventually, $\mathcal{A}$ outputs $(\mathbb{x}_1, \pi_1 \| \pi_2, \mathbb{w}_3)$, return the forgery $(\mathbb{x}_2, (\mathbb{x}_1 \| \pi_1), \pi_2, \mathbb{w}_3)$ where $\mathbb{x}_2 \leftarrow \mathsf{V}_1^{\mathsf{RO}[\mathcal{Q}_1]}(\mathsf{pp}, \mathbb{x}_1, \pi_1)$.

We notice that a valid forgery for $\mathcal{A}$ is a valid forgery for $\mathcal{A}_2$, in fact: (1) $(\mathbb{x}_1, \pi_1 \| \pi_2, \mathbb{w}_3) \notin \mathcal{Q}_{\mathcal{S}}$ implies that $(\mathbb{x}_2, (\mathbb{x}_1 \| \mathbb{x}_1), \pi_2, \mathbb{w}_3) \notin \mathcal{Q}_{\mathcal{S}_2}$ because there is an injective function mapping the tuples in $\mathcal{Q}_{\mathcal{S}}$ into tuples in $\mathcal{Q}_{\mathcal{S}_2}$ and thus $b_Q \Rightarrow b_{Q_2}$, and (2) if the verification passes for Π it must pass for Π_2.

Thus, the experiment $\mathsf{SIM\text{-}EXT}_{0,\Pi_2}^{\mathcal{S}_2,\mathcal{A}_2,\mathcal{E}_2}(\lambda)$ is equivalent to $\mathbf{H}_0$, and by definition of $\mathbf{H}_1$, we have $\mathsf{SIM\text{-}EXT}_{1,\Pi_2}^{\mathcal{S}_2,\mathcal{A}_2,\mathcal{E}_2}(\lambda)$ is equivalent to $\mathbf{H}_1$. □

Claim. There exist EPT extractor $\mathcal{E}_1$ and PT adversary $\mathcal{A}_1$ such that

$$|\Pr(\mathbf{H}_1 = 1) - \Pr(\mathbf{H}_2 = 1)| \leq \mathsf{Adv}_{\mathsf{pZK}}^{\mathtt{SE}}(\Pi_1, \mathcal{S}_1, \mathcal{E}_1, \mathcal{A}_1).$$

Proof. Similarly to the previous claim, $\mathbf{H}_2$ additionally run the extractor $\mathcal{E}_1$ of Π_1 on an adversary $\mathcal{A}_1$. We define such a PT adversary $\mathcal{A}_1$ against $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ of Π_1 as follows:

Adversary $\mathcal{A}_1^{\mathsf{RO}}(\mathsf{pp})$:

1. Initialize an empty set $\mathcal{Q}_{\mathcal{S}_2}$ of simulated proofs $\mathcal{S}_2$, empty set $\mathcal{Q}_2$ of programmed random oracle queries by $\mathcal{S}_2$.
2. Run the adversary $\mathcal{A}^{\mathsf{RO}[\mathcal{Q}_2]}(\mathsf{pp})$ and reply its queries as follows:
   - Upon simulation query $\mathbb{x}_1$ from $\mathcal{A}$ to, send query $\mathbb{x}_1$ to its own simulation oracle, let $\tilde{\pi}_1, (\tilde{\mathbb{w}}_2', \tilde{\mathbb{w}}_2'')$ be the output of the oracle, run locally the public-reducibility function, the simulator $\mathcal{S}_2$ and add the query to $\mathcal{Q}_2$. In more details, run locally the operations 5 to 12 of $\mathsf{O}_{\mathcal{S}}$ in Fig. 7.
   - Upon random oracle query, reply the query according to $\mathsf{RO}[\mathcal{Q}_2]$.
3. Eventually, $\mathcal{A}$ outputs $(x, \pi_1 \| \pi_2, \mathbb{w}_3)$.
4. Run extractor $\mathcal{E}_2^{\mathcal{A}_2}((\mathbb{x}_2, (\mathbb{x}_1, \pi_1)), \pi_2, \mathbb{w}_3)$ obtaining $\mathbb{w}_2$.
5. If $(\mathbb{x}_2, \mathbb{w}_2) \in \mathsf{R}_2$ output forgery $(\mathbb{x}, \pi_1, \mathbb{w}_2)$, else output $\perp$.

Assuming that the extraction of $\mathcal{E}_2$ holds, whenever a forgery of $\mathcal{A}$ passes the verification algorithm for Π then the forgery of $\mathcal{A}_1$ passes the verification algorithm for Π_1. Notice, since we reduce to $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$, any verifying forgery of $\mathcal{A}_1$ is a valid forgery for the game. Moreover, for hybrid $\mathbf{H}_1$ to return 1 it is necessary the extraction of $\mathcal{E}_2$ to hold. Thus $\mathsf{SIM\text{-}EXT}_{w,0,\Pi_1}^{\mathcal{S}_1,\mathcal{A}_1,\mathcal{E}_1}(\lambda)$ and $\mathbf{H}_1$ are equivalent. By definition of $\mathbf{H}_2$ we have that $\mathsf{SIM\text{-}EXT}_{w,1,\Pi_1}^{\mathcal{S}_1,\mathcal{A}_1,\mathcal{E}_1}(\lambda)$ and $\mathbf{H}_2$ are equivalent.

Claim. There exists PPT algorithm $\mathcal{E}$ such that

$$\Pr[\mathbf{H}_2 = 1] = \Pr\left[\mathsf{SIM\text{-}EXT}_{1,\Pi}^{\mathcal{S},\mathcal{A},\mathcal{E}}(\lambda) = 1\right].$$

Proof. It suffices to construct such extractor algorithm $\mathcal{E}$.

Extractor $\mathcal{E}^{\mathcal{A}}(\mathsf{pp}, \mathbb{x}_1, \pi_1 || \pi_2, \mathbb{w}_2)$:
1. Run verifier $\mathbb{x}_2 \leftarrow \mathsf{V}_1^{\mathsf{RO}}(\mathsf{pp}, \mathbb{x}_1, \pi_1)$
2. Run extractor $\mathbb{w}_2 \leftarrow \mathcal{E}_2^{\mathcal{A}_2}(\mathsf{pp}, \mathbb{x}_2, \mathbb{x}_1 \| \pi_1, \pi_2, \tilde{\mathbb{w}}_3)$
3. Run extractor $\mathbb{w}_1 \leftarrow \mathcal{E}_1^{\mathcal{A}_1}(\mathsf{pp}, \mathbb{x}_1, \pi_1, \mathbb{w}_2)$
4. Output $\mathbb{w}_1$

Clearly, the extractor $\mathcal{E}$ runs the operation described in lines 4,6,9 of $\mathbf{H}_2$, moreover the $\mathbf{H}_2$ performs all the checks described by the simulation extractability game for Π. Thus the two experiment are semantically equivalent. □

Third Statement. The proof of the third statement follows very similarly to the proof of the previous statement. The only exception is that $\mathcal{A}_2$ is an adversary against $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$; thus, the corresponding $\mathbf{H}_1$ does not need to check the bit $b_{Q_2'}$. □

7 zk-LaBRADot is Non-Malleable

We consider the main protocol of LaBRADOR [9] for dot-product constraints, referred below as LaBRADot. We define the principal relation R for LaBRADot.

The relation is (implicitly) parameterized by a rank $n \geq 1$, a multiplicity $r \geq 1$. An instance consists of a ℓ_2 norm bound β and a list of quadratic dot-product functions $\mathcal{F} = (f_k)_{k\in[K]}$ with signature $f : (R_q^n)^r \to R_q$ forming equations of the form

$$f(\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_r) = \sum_{i,j=1}^{r} \boldsymbol{a}_{i,j} \langle \vec{\boldsymbol{s}}_i, \vec{\boldsymbol{s}}_j \rangle + \sum_{i=1}^{r} \langle \vec{\boldsymbol{\varphi}}_i, \vec{\boldsymbol{s}}_i \rangle - \boldsymbol{b}, \tag{2}$$

where $\boldsymbol{a}_{i,j}, \boldsymbol{b} \in \mathcal{R}_q$ and $\vec{\boldsymbol{\varphi}}_i \in \mathcal{R}_q^n$. The matrix $(\boldsymbol{a}_{i,j})$ can be assumed to be symmetric without loss of generality, i.e. $\boldsymbol{a}_{i,j} = \boldsymbol{a}_{j,i}$. The witnesses $\vec{\boldsymbol{s}} = (\vec{\boldsymbol{s}}_j)_{j\in[r]}$ and $\forall j : \vec{\boldsymbol{s}}_j \in \mathcal{R}_q^n$ are short solutions for the system of equations defined by the functions over $\mathcal{R}_q$.

For efficiency, for some of these functions, we check the equations only for the constant terms. We call these *constant-term (dot-product) equations*. Thus, we consider two lists of equations.

$$((f_j)_{j\in[K]}, (f'_j)_{j\in[K']}, \beta), (\vec{\boldsymbol{s}}_j)_{j\in[r]} \in \mathsf{R} \iff \begin{array}{c} \forall i : f_i(\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_r) = 0 \\ \forall i : \mathsf{ct}(f'_i(\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_r)) = 0 \\ \sum_i \|\vec{\boldsymbol{s}}_i\|^2 \leq \beta^2 \end{array}$$

We sometimes omit β when it is clear from the context. We can naturally define a commit-and-prove version of the principal relation. We consider the double commitment scheme used in LaBRADot to define the commit-and-prove relation. In particular, let Com be the Ajtai commitment scheme. Given a matrix $\boldsymbol{A} \in \mathcal{R}_q^{\kappa\times n}$ which defines the commitment key, we define $\mathsf{Com}(\boldsymbol{A}, \vec{\boldsymbol{s}}) = \boldsymbol{A}\vec{\boldsymbol{s}}$. With a slight abuse of notation, we denote by $(\vec{\boldsymbol{s}}_j)_{j\in[r]}$ both a list of vectors and a matrix whose columns are the vectors $\vec{\boldsymbol{s}}_j$. Thus, $(\vec{\boldsymbol{t}}_j)_{j\in[r]} = \mathsf{Com}(\boldsymbol{A}, (\vec{\boldsymbol{s}}_j)_{j\in[r]})$ is well defined. Continuing this abuse of notation, given a list of matrices $(\boldsymbol{A}_i)_i$ and $(\vec{\boldsymbol{s}}_i)_i$, we define $\mathsf{Com}((\boldsymbol{A}_i)_i, (\vec{\boldsymbol{s}}_i)_i) = \sum_i \boldsymbol{A}_i \vec{\boldsymbol{s}}_i$.

We define Split_b to be the function that, on input a vector $\vec{\boldsymbol{t}}$, outputs the minimum list of vectors $(\vec{\boldsymbol{t}}_j)_{j\in[t]}$, with $t \leq \lceil \log q / \log b \rceil$, such that $\sum_j b^j \vec{\boldsymbol{t}}_j = \vec{\boldsymbol{t}}$ and $\|\vec{\boldsymbol{t}}_j\|_\infty \leq b/2$. We omit the subscript b whenever it is clear from the context.

Finally, given $\mathsf{ck} = (\mathsf{ck}_A, \mathsf{ck}_B) = (\vec{\boldsymbol{A}}, (\vec{\boldsymbol{B}}_{ij})_{i,j})$ where $\vec{\boldsymbol{B}}_{ij} \in \mathcal{R}_q^{\kappa_1\times\kappa}$ with the appropriate dimension for κ_1 , we define the double commitment

$$\mathsf{DCom}(\mathsf{ck}, \vec{\boldsymbol{s}}) := \mathsf{Com}(\mathsf{ck}_B, \mathsf{Split}(\mathsf{Com}(\mathsf{ck}_A, \vec{\boldsymbol{s}}))).$$

We omit the commitment keys whenever they are clear from the context. Sometimes it is useful to associate a slot to the committed vectors, which is why the matrices $\vec{\boldsymbol{B}}_{i,j}$ are double indexed. Given $I \subset [r]$, we write

$$\mathsf{DCom}(\mathsf{ck}_I, (\vec{\boldsymbol{s}}_i)_{i\in I}) = \sum_{i\in I} \mathsf{DCom}((\mathsf{ck}_A, \mathsf{ck}_{B,i}), \vec{\boldsymbol{s}}_i),$$

where $\mathsf{ck}_{B,i} = (\boldsymbol{B}_{i,j})_{j\in[\lceil \log q/\log b\rceil]}$ and $\mathsf{ck}_I = (\mathsf{ck}_A, (\mathsf{ck}_{B,i})_{i\in I})$.

Notice that DCom is not strictly (somewhat) homomorphic. In fact, the sum of the splits of two vectors $\vec{\boldsymbol{t}}$ and $\vec{\boldsymbol{t}}'$ is not necessarily the split of $\vec{\boldsymbol{t}} + \vec{\boldsymbol{t}}'$. However, we can define auxiliary opening material such that the commitment DCom is somewhat homomorphic with respect to the auxiliary opening material. Namely, given $\mathsf{c} = \mathsf{DCom}(\vec{\boldsymbol{s}})$, we define the auxiliary material to be $(\vec{\boldsymbol{t}}_j)_j = \mathsf{Split}(\mathsf{Com}(\vec{\boldsymbol{s}}))$. We then have that the double commitment scheme is somewhat homomorphic with respect to the auxiliary opening material and the opened value. We sometimes use the shorthand $\mathsf{c}, \vec{\boldsymbol{t}} \leftarrow \mathsf{DCom}(\vec{\boldsymbol{s}})$ to indicate that c denotes the double-commitment to $\vec{\boldsymbol{s}}$, while $\vec{\boldsymbol{t}} := (\vec{\boldsymbol{t}}_j)_j = \mathsf{Split}(\mathsf{Com}(\vec{\boldsymbol{s}}))$ represents the corresponding auxiliary opening values.

We define an opening algorithm Open, outputting a bit, for the double commitment scheme such that

$$\mathsf{Open}(\mathsf{ck}_I, \mathsf{c}, \beta, \gamma, (\vec{\boldsymbol{s}}_i)_{i\in I}, (\vec{\boldsymbol{t}}_{ij})_{i,j}) \iff \begin{cases} \sum_{i\in I} \mathsf{Com}(\mathsf{ck}_{B,i}, (\vec{\boldsymbol{t}}_{ij})_{i,j}) = \mathsf{c}, \\ \forall i \in I : \mathsf{Com}(\mathsf{ck}_A, \vec{\boldsymbol{s}}_i) = \vec{\boldsymbol{t}}_i, \\ \sum_j b^j \vec{\boldsymbol{t}}_{ij} = \vec{\boldsymbol{t}}_i, \\ \|\vec{\boldsymbol{t}}\| \le \gamma \wedge \sum_j \|\vec{\boldsymbol{s}}_j\|^2 \le \beta^2 \end{cases}$$

We define our commit-and-prove principal relation. The relation is additionally parameterized by the commitment key. Let $\mathbb{x} = ((\mathsf{c}_k, I_k, \gamma_k)_{k\in[C]}, \mathcal{F}, \mathcal{F}', \beta, \vec{\gamma})$ and $\mathbb{w} = ((\vec{\boldsymbol{s}}_j)_{j\in[r]}, (\vec{\boldsymbol{t}}_{ij})_{i,j})$. We define

$$(\mathbb{x}, \mathbb{w}) \in \hat{\mathsf{R}} \iff \begin{cases} I_k \subseteq [r], \forall k \neq k' : I_k \cap I_{k'} = \emptyset \\ \forall k : \mathsf{Open}(\mathsf{ck}_{I_k}, \mathsf{c}_k, \beta, \gamma_k, (\vec{\boldsymbol{s}}_i)_{i\in I_k}, (\vec{\boldsymbol{t}}_{ij})_{i\in I_k, j}) \\ ((\mathcal{F}, \mathcal{F}', \beta), (\vec{\boldsymbol{s}}_i)_{i\in[r]}) \in \mathsf{R} \end{cases}$$

Informally, we partition the vectors in the witness into $C+1$ sets, with the first C sets, namely $(\vec{\boldsymbol{s}}_i)_{i\in I_1}, \ldots, (\vec{\boldsymbol{s}}_j)_{j\in I_C}$, committed independently, and the last set, namely $\bar{I} \leftarrow [n] \setminus (\cup_{j\in[C]} I_j)$, uncommitted. Using separate slots for these commitments allows us to combine the sets by summing the commitments. Thus, if $((\mathsf{c}_1, I_1, \gamma_1), (\mathsf{c}_2, I_2, \gamma_2), \mathcal{F}, \mathcal{F}', \beta)$ is a valid instance, then the instance $((\mathsf{c}_1 + \mathsf{c}_2,\ I_1 \cup I_2, \max(\gamma_1, \gamma_2)), \mathcal{F}, \mathcal{F}', \beta)$ is also valid, and both instances share the same witness.

We omit the sets I_k and the values β and γ_k from the instance, simply writing $\mathbb{x} = ((\mathsf{c}_k)_{k\in[C]}, \mathcal{F}, \mathcal{F}')$, when they are clear from the context.

7.1 Hiding Double Commitment

Notice that DCom is not hiding. A simple way to achieve hiding is to concatenate to the committed vector a random vector $\vec{\boldsymbol{r}}$ sampled uniformly with bounded norm. However, applying this approach directly to the witness has two drawbacks. First, it increases the size of the witness. Second, it is not well suited for masking techniques that rely on (somewhat) homomorphic properties to randomize the witness for zero-knowledge purposes (for example, Lyubashevsky's rejection sampling). The issue is that the distribution of the auxiliary material depends in a non-trivial way on the committed values, whereas for masking we would prefer it to be independent of them.

We notice that we can randomize a double commitment in an alternative way. Namely, consider the lattice $\Lambda := \{\vec{\boldsymbol{z}} \in (\mathcal{R}_q^\kappa)^t \mid \sum_{j=0}^t \vec{\boldsymbol{z}}_j b^j = 0 \mod q\}$ and denote with $D_{\Lambda,\sigma}$ the lattice gaussian with standard deviation σ. We note that we can sample from $D_{\Lambda,\sigma}$ using standard techniques, for instance those of [50, Theorem 4.1]. Notice that we can interpret $\mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{t}}')$ with $\vec{\boldsymbol{t}}' \xleftarrow{\$} D_{\Lambda,\sigma}$ as a randomized double commitment to $\mathbf{0}$.

Thus, consider the following hiding version of the double commitment scheme DCom^*, that, upon input $(\vec{\boldsymbol{s}}_i)_{i\in I}$ for some $I \subset \mathbb{N}$ and commitment key ck, computes:

- Sample $(\vec{\boldsymbol{t}}_i)_{i\in I} \xleftarrow{\$} D_{\Lambda,\sigma}^{|I|t}$, commit $\mathsf{c}', \vec{\boldsymbol{t}}' \leftarrow \mathsf{DCom}(\mathsf{ck}_I, (\vec{\boldsymbol{s}}_i)_{i\in I})$
- Compute $\mathsf{c} \leftarrow \bar{\mathsf{c}} + \sum_{i\in I} \mathsf{Com}(\mathsf{ck}_{B,i}, \vec{\boldsymbol{t}}_i)$ and $\vec{\boldsymbol{t}} \leftarrow \vec{\boldsymbol{t}}' + (\vec{\boldsymbol{t}}_i)_i$.
- Output $(\mathsf{c}, \vec{\boldsymbol{t}})$.

As a shorthand, we sometimes let DCom^* output an additional auxiliary value $\vec{\boldsymbol{t}}'$. Looking ahead, this value will be used in conjunction with rejection sampling. In particular, when adding a (Gaussian) random mask $\vec{\boldsymbol{w}}$, for which we compute $\mathsf{c}_w, \vec{\boldsymbol{t}}_w, \vec{\boldsymbol{t}}_w' \leftarrow \mathsf{DCom}^*(\mathsf{ck}, \vec{\boldsymbol{w}})$, to a (double) committed witness $\vec{\boldsymbol{s}}$, we require two properties. First, the sum $\vec{\boldsymbol{w}} + \vec{\boldsymbol{s}}$ should be distributed as a Gaussian centered at zero, and hence be independent of $\vec{\boldsymbol{s}}$. Second, the sum $\vec{\boldsymbol{t}}_w + \vec{\boldsymbol{t}}$ should follow a Gaussian distribution centered at $\vec{\boldsymbol{t}}_w'$, and be independent of $\vec{\boldsymbol{t}}_s$.

As a shortcut, additionally, we let sometimes DCom^* output yet another auxiliary value $\vec{\boldsymbol{t}}'$. Looking ahead, this additional auxiliary value will be used when using rejection sampling, because when adding a (gaussian) random mask $\vec{\boldsymbol{w}}$ which, will be committed $\mathsf{c}_w, \vec{\boldsymbol{t}}_w, \vec{\boldsymbol{t}}_w' \leftarrow \mathsf{DCom}^*(\mathsf{ck}, \vec{\boldsymbol{w}})$, to a (double) committed witness $\vec{\boldsymbol{s}}$ we need both $\vec{\boldsymbol{z}} = \vec{\boldsymbol{w}} + \vec{\boldsymbol{s}}$ to be distributed as gaussian centered on zero and thus independent of $\vec{\boldsymbol{s}}$ and $\vec{\boldsymbol{t}}_w + \vec{\boldsymbol{t}}$ as a gaussian centered this time on $\vec{\boldsymbol{t}}_z' = \mathsf{Split}(\mathsf{Com}(\mathsf{ck}_A, \vec{\boldsymbol{z}}))$ and thus independent of $\vec{\boldsymbol{t}}_s$.

Theorem 6 (Hiding DCom^*). *Let $\sigma > \max\left\{(t-1)\sqrt{b^2+1} \cdot q^{\kappa_1/(\kappa t)} \cdot 2^{\lambda/(\kappa t d)}, 2(t-1)\sqrt{b^2+1}\right\}$ where $t = \lceil \log q / \log b \rceil$, the double-commitment scheme DCom^* with variance parameter σ and splitting parameter b is statistically hiding.*

Proof (sketch). We provide a proof sketch. This is essentially an application of [15, Corollary 6], which describes a generalisation of Leftover-Hash Lemma when the ring $\mathcal{R}_q$ is low-splitting. In more details, one needs to bound the term $\frac{q^{d\kappa_1}}{\rho_\sigma(\Lambda)}$, where the Gaussian mass $\rho_\sigma(\Lambda)$ can be lower bounded by estimating the number of lattice elements that lies in neighbourhood $\mathcal{B}(\sigma/2) \subset \mathbb{R}^{\kappa d t}$.

Proof. It suffices to argue that for $\vec{\boldsymbol{t}}' \xleftarrow{\$} D_{\Lambda,\sigma}$, $(\mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{t}}'), \mathsf{ck}_B)$ is close to uniform distribution over the same domain. We compute an upper bound on their statistical distance.

First we lower bound the gaussian mass $\rho_\sigma(\Lambda)$. Since $\Lambda = L^{\kappa d}$ where $L = \{x \in \mathbb{Z}^t : \langle x, g\rangle \equiv 0 \bmod q\}$, $g = (1, b, \ldots, b^{t-1})$, it suffices to compute $\rho_\sigma(L)$, we count the number of lattice elements with norm bounded by $\sigma/2$. Note that L is full rank with $t-1$ basis vectors $(v_i)_{i=0}^{t-1}$ of norm $\sqrt{b^2+1}$. For $z_i \in (-A, A) \subset \mathbb{Z}$, $x(z) = \sum_{i=1}^{t-1} z_i v_i$ lies in lattice L. Moreover, by triangular inequality, $\|x(z)\|_2 \le (t-1)A\sqrt{b^2+1}$. Take $A = \left\lfloor \frac{\sigma}{2(t-1)\sqrt{b^2+1}} \right\rfloor$, $\|x(z)\|_2 \le \sigma/2$. There are at least $(2A+1)^t$ such lattice elements, thus $\rho_\sigma(L) \ge e^{-\pi/4}\left(2\left\lfloor \frac{\sigma}{2(t-1)\sqrt{b^2+1}} \right\rfloor + 1\right)^{t-1}$.

Therefore we can compute collision probability $\mathsf{Coll}(D_{\Lambda,\sigma}) \le 2^{-H_\infty(D_{\Lambda,\sigma})} = \frac{1}{\rho_\sigma(\Lambda)} = \rho_\sigma(L)^{\kappa d}$. Applying [15, Corollary 6], one gets $SD(\mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{t}}'), \mathsf{Uniform}) \lesssim \frac{1}{2}\sqrt{|\mathcal{R}_q|^{\kappa_1}((\frac{(t-1)\sqrt{b^2+1}}{\sigma})^{\kappa t d} + 2\kappa t e^{-\pi r^2})}$ when $\sigma r < \frac{\sqrt{q}}{2\sqrt{2}}$.
Equivalently,

$$\max\{(t-1)\sqrt{b^2+1} \cdot q^{\kappa_1/(\kappa t)} \cdot 2^{\lambda/(\kappa t d)}, 2(t-1)\sqrt{b^2+1}\} < \sigma < \frac{\sqrt{q}}{2\sqrt{2}(\log q + \lambda)}$$

implies $SD(\mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{t}}'), \mathsf{Uniform}) \le \mathsf{negl}(\lambda)$.

7.2 Commit-and-Prove LaBRADot

We give a modular description of LaBRADot. In Fig. 11 in Section Bwe recall the full scheme LaBRADot from [9]. At a high level, the scheme consists of four steps, which we formalize as four RoKs, namely $\Pi_{\mathsf{nrm\text{-}chck}}, \Pi_{\mathsf{agg}'}, \Pi_{\mathsf{agg}}$, and Π_{amm}.

We note that the straightforward composition of our RoKs does not exactly coincide with the scheme described in Fig. 11. In particular, the authors of LaBRADot observe that the commitment $\boldsymbol{u}_1$, which in our description is sent in the first round of Π_{amm} (see below), can instead be sent earlier, since it depends only on the witness and not on the instance. We argue (see the proof of Theorem 7) that this difference in the message flow does not affect the analysis of 0-unique response. Indeed, sending a message earlier cannot violate this property (whereas sending it later potentially could). Intuitively, sending a message earlier can only reduce the adversary's ability to find diverging suffixes of the transcript.

In addition, we describe a commit-and-prove variant of LaBRADot, which can be obtained in a straightforward way from the original scheme. Concretely, we commit, as part of one of the proof elements, only to the witness components that are not yet committed. This step is also implicitly present in the original LaBRADot; however, in our setting we allow that part of the witness (or all of it) may already be committed.

In the first step of LaBRADot, which we identify with $\Pi_{\mathsf{nrm\text{-}chck}}$, the verifier samples[9] random challenge matrices $P_i \xleftarrow{\$} \mathsf{Bin}_1^{(2\lambda \times nd) \times r}$ where λ is a statistical security parameter. The prover responds with the projections $\vec{p}$ of the witness vectors $\vec{\boldsymbol{s}} = (\vec{\boldsymbol{s}}_j)_{j \in [n]}$ under the matrices P_i. These are Johnson–Lindenstrauss

[9] Recall that the probability mass function of Bin_1 assigns probability 1/2 to 0, and probability 1/4 to each of 1 and −1.

(JL) projections. As shown in [9], bounding the norm of these projections implies, with overwhelming probability in λ, a bound on the norm of the vectors $\vec{\boldsymbol{s}}$. The correctness of the projections can be expressed via linear functions f''_k whose validity is checked only on constant terms, namely $\mathsf{ct}(f''_k(\vec{\boldsymbol{s}})) = 0$, where

$$f''_k(\vec{\boldsymbol{s}}) = \sum_i \langle \sigma_{-1}(\vec{\boldsymbol{\pi}}_{ik}), \vec{\boldsymbol{s}}_i \rangle - p_k. \tag{3}$$

Above the vector $\vec{\boldsymbol{\pi}}_{ik} \in \mathcal{R}_q^n$ is defined by interpreting as a vector of polynomials the k-th row of the matrix P_i.

This RoK can be composed with any RoK with *relaxed knowledge soundness*, namely where the extraction algorithm can only output a weak opening of the commitments, to obtain a valid extracted witness. Formally, the approximate norm check is a RoK from $\hat{\mathsf{R}}$ to $\hat{\mathsf{R}}_{\mathsf{relaxed}}$, where the latter relation coincides with $\hat{\mathsf{R}}$ except that the norm constraint on a witness $\vec{\boldsymbol{s}}$ of the form $\|\vec{\boldsymbol{s}}\|_2 \leq \beta$ is replaced by a constraint of the form $\|\boldsymbol{a} \cdot \vec{\boldsymbol{s}}\|_2 \leq \beta$, for a value $\boldsymbol{a}$, defined by the extractor, that is typically a product of some challenge differences. We describe the RoK for approximate norm check, for simplicity we assume the instance contains only one commitment, extension to multiple commitments is trivial:

RoK $\Pi_{\mathsf{nrm\text{-}chck}}$ from $\hat{\mathsf{R}}$ to $\hat{\mathsf{R}}_{\mathsf{relaxed}}$

	Prover $\mathbb{w} = \vec{\boldsymbol{s}}, (\vec{\boldsymbol{t}}_j)_{j \in I}$		**Verifier** $\mathbb{x} = ((\mathsf{c}, I, \gamma), \mathcal{F}, \mathcal{F}', \tilde{\beta})$
1 :		$\xleftarrow{(P_i)_i}$	$(P_i)_{i \in [r]} \xleftarrow{\$} \mathsf{Bin}_1^{(2\lambda \times nd) \times r}$
2 :	$\vec{p} \leftarrow \sum_i P_i \cdot \tau(\vec{\boldsymbol{s}}_i) \in \mathbb{Z}_q^{2\lambda}$	$\xrightarrow{\vec{p}}$	**Check** $\|\vec{p}\|_2 \leq \sqrt{\lambda}\beta$
3 :			$\mathbb{x}' \leftarrow ((\mathsf{c}, I, \gamma), (\bar{\mathsf{c}}, \bar{I}, b), \mathcal{F}, \mathcal{F}' \cup (f''_k)_{k \in [2\lambda]}, \beta)$
4 :	**return** $\vec{\boldsymbol{s}}, (\vec{\boldsymbol{t}}_j)_{j \in [r]}$		**return** $\mathbb{x}'$

Above, the function τ is a coefficient embedding mapping $\mathcal{R}_q$ element to $\mathbb{Z}^d$ and $\tilde{\beta} = (\sqrt{\lambda / C_2})\beta$ where the constant C_2 depends on Lemma 9.

Next, two aggregation steps combine the functions f_k, the functions f'_k, and the newly introduced functions to check the JL projections (namely, the f''_k in the RoK $\Pi_{\mathsf{nrm\text{-}chck}}$), reducing their verification to a single aggregated function F.

In particular, the first aggregation reduces the constant-term dot-product equations to (standard) dot-product equations. To aggregate the constant-term dot-product equations LaBRADot needs challenges from $\mathbb{Z}_q$, thus, to obtain negligible soundness error, multiple challenges are used.

RoK $\Pi_{\mathsf{agg}'}$ from $\hat{\mathsf{R}}$ to $\hat{\mathsf{R}}$

	Prover $\mathbb{w} = (\vec{\boldsymbol{s}}, \vec{\boldsymbol{t}})$		**Verifier** $\mathbb{x} = (\mathsf{c}, \mathcal{F}, \mathcal{F}')$
1 :	**Let** $\forall j : f'_j(\vec{\boldsymbol{s}}) = q'_j(\vec{\boldsymbol{s}}) + b'_j$	$\xleftarrow{(\vec{\psi}_k)_k}$	$\forall k \in [\lambda'] : \vec{\psi}_k \xleftarrow{\$} \mathbb{Z}_q^{K'}$
2 :	$q''_k(\vec{X}) \leftarrow \sum_j \psi_{kj} q'_j(\vec{X})$		
3 :	$\boldsymbol{b}''_k \leftarrow q''_k(\vec{\boldsymbol{s}})$	$\xrightarrow{(\boldsymbol{b}''_k)_k}$	**Check** $\mathsf{ct}(\boldsymbol{b}''_k) = \sum_j \psi_{kj} b'_j$
4 :			$\forall k : f''_k(\vec{X}) := q''_k(\vec{X}) - \boldsymbol{b}''_k$
5 :	**return** $\mathbb{w}$		**return** $(\mathsf{c}, \mathcal{F} \cup (f''_k)_{k \in [\lambda']}, \emptyset)$

For efficiency, the constant terms $(b'_j)_j$ of the functions in $\mathcal{F}'$ belong to $\mathbb{Z}_q$ thus the prover must send the values $(\boldsymbol{b}''_k)_k$, as these cannot be computed by the verifier alone. As a consequence, this RoK is not zero-knowledge, not even in the partial sense. We show in Section 7.4 that this issue can be easily addressed by committing to additional randomness and by adding a dummy dot-product function and then running $\Pi_{\mathsf{agg}'}$.

The second aggregation step in LaBRADot is essentially identical to $\Pi_{\mathsf{agg}'}$. However, since the original dot-product equations are defined over $\mathcal{R}_q$, the aggregation uses a single challenge vector in $\mathcal{R}_q^K$ to combine the newly generated dot-product equations. Moreover, in this case the prover does not need to send the values $(\boldsymbol{b}''_k)_k$, as they can be computed directly by the verifier. We denote this RoK by Π_{agg}.

RoK Π_{agg} from $\hat{\mathsf{R}}$ to $\hat{\mathsf{R}}$

	Prover $\mathbb{w} = (\vec{\boldsymbol{s}}, \vec{\boldsymbol{t}})$		**Verifier** $\mathbb{x} = (\mathsf{c}, \mathcal{F}, \mathcal{F}')$
1 :		$\xleftarrow{\quad\vec{\boldsymbol{\alpha}}\quad}$	$\vec{\boldsymbol{\alpha}} \xleftarrow{\$} \mathcal{R}_q^K;\ \ F(\vec{X}) \leftarrow \sum_j \boldsymbol{\alpha}_j f_j(\vec{X})$
2 :	**return** $\mathbb{w}$		**return** $(\mathsf{c}, F, \mathcal{F}')$

To ensure type consistency when composing with the subsequent RoK, we observe that the previous aggregation RoKs can also be viewed as reductions from $\hat{\mathsf{R}}_{\mathsf{relaxed}}$ to $\hat{\mathsf{R}}_{\mathsf{relaxed}}$. Indeed, both RoKs leave the witness unchanged, and therefore a weak opening of the target witness implies a weak opening of the source witness, and vice versa.

Finally, an amortization step, which we formalize with the RoK Π_{amm}, maps the witness $\vec{\boldsymbol{s}}$ to a new witness $\vec{\boldsymbol{s}}' = (\bar{\boldsymbol{z}}, \vec{\boldsymbol{g}}, \vec{\boldsymbol{h}}, \vec{\boldsymbol{t}})$, where the vectors $\vec{\boldsymbol{g}} = (\boldsymbol{g}_{ijk})_{i,j,k}$ and $\vec{\boldsymbol{h}} = (\boldsymbol{h}_{ijk})_{i,j,k}$ are referred to as *garbage terms*. The value $\boldsymbol{g}_{ij} := \sum b^k \boldsymbol{g}_{ijk}$ are obtained as the dot products $\langle \vec{\boldsymbol{s}}_i, \vec{\boldsymbol{s}}_j \rangle$, while the values $\boldsymbol{h}_{i,j} := \sum b^k \boldsymbol{h}_{ijk}$ arise from the linear terms of the dot-product equations.[10] These values are committed using double commitments (called outer commitments in LaBRADot), producing c_g and $\boldsymbol{u}_2$ for $\vec{\boldsymbol{g}}$ and $\vec{\boldsymbol{h}}$, respectively.

The RoK commits to all the uncommitted witness material, defining $\bar{\mathsf{c}}$, and sets $\boldsymbol{u}_1 = \mathsf{c}_g + \bar{\mathsf{c}}$. Notice, this step is equivalent to the original LaBRADot protocol when $I = \emptyset$ and thus $\bar{I} = [r]$.

The amortization step reduces the verification of the freshly aggregated function F, together with the verification of the openings of the produced commitments, to the following functions:

$$\begin{aligned}
\tilde{f}_1(\vec{\boldsymbol{s}}') &:= \langle \bar{\boldsymbol{z}}, \bar{\boldsymbol{z}} \rangle - \sum_{i,j} \boldsymbol{g}_{ij} \boldsymbol{c}_i \boldsymbol{c}_j, \\
\tilde{f}_2(\vec{\boldsymbol{s}}') &:= \sum_i \langle \vec{\boldsymbol{\psi}}_i, \bar{\boldsymbol{z}} \rangle \boldsymbol{c}_i - \sum_{i,j} \boldsymbol{h}_{ij} \boldsymbol{c}_i \boldsymbol{c}_j, \\
\tilde{f}_3(\vec{\boldsymbol{s}}') &:= \sum_{i,j} \boldsymbol{a}_{i,j} \boldsymbol{g}_{ij} - \sum_i \boldsymbol{h}_{ii} + \boldsymbol{b}.
\end{aligned} \tag{4}$$

Here the values $\boldsymbol{a}_{ij}$, $\vec{\boldsymbol{\psi}}_i$, and $\boldsymbol{b}$ denote, respectively, the coefficients of the quadratic terms, linear terms, and constant term of the aggregated function F. In order to generate the target instance, the verifier generates additional functions $(\tilde{f}_k)_{k>3}$ that encode the checks:

- $\boldsymbol{A}\bar{\boldsymbol{z}} = \sum_i \boldsymbol{c}_i \vec{\boldsymbol{t}}_i$ where $\vec{\boldsymbol{t}}_i = \sum_j b^j \vec{\boldsymbol{t}}_{ij}$.
- $\mathsf{c} = \sum_{i \in I, j} \boldsymbol{B}_{i,j} \vec{\boldsymbol{t}}_{i,j}$, $\boldsymbol{u}_1 = \sum_{i \in \bar{I}, j} \boldsymbol{B}_{ij} \vec{\boldsymbol{t}}_{ij} + \sum_{i,j} \boldsymbol{C}_{ijk} \vec{\boldsymbol{g}}_{ijk}$ and $\boldsymbol{u}_2 = \sum_{i,j} \boldsymbol{D}_{ijk} \vec{\boldsymbol{h}}_{ijk}$, where $\mathsf{ck}_C = (\boldsymbol{C}_{ijk})_{i,j,k}$ and $\mathsf{ck}_C = (\boldsymbol{D}_{ijk})_{i,j,k}$ and the matrices $\boldsymbol{D}_{ijk}$ have κ_2 columns.

We define Π_{amm} as a RoK from $\hat{\mathsf{R}}_{\mathsf{relaxed}}$ to $\hat{\mathsf{R}}$ because we can only extract a weak opening for the source witness $\vec{\boldsymbol{s}}$ from a valid target witness $(\vec{\boldsymbol{z}}, \vec{\boldsymbol{g}}, \vec{\boldsymbol{h}}, \vec{\boldsymbol{t}})$. Notice that, with this choice of source relation, Π_{amm} only has low completeness error on the tuple from $\mathsf{R}' := \hat{\mathsf{R}} \cap \{(\mathbb{x}, \mathbb{w}) : |\mathcal{F}| = 1, \mathcal{F}' = \emptyset\}$. However, our goal is to analyze its properties under composition with the other RoKs, where it is only invoked on instance-witness tuples that lie in R'.

[10] Note that since we're only analysing SIM-EXT of one iteration LaBRADot, witness vector $\bar{\boldsymbol{z}}$ is not splitted for better readability. As a result, the ℓ_2 norm bound β' on target witness increases by at most a factor of b compared with the original β' in LaBRADot. Analysis of 0-UR in the case where $\bar{\boldsymbol{z}}$ is fully splitted follows from a similar argument.

RoK $\Pi^{\mathsf{g}}_{\mathsf{amm}}$ from $\hat{\mathsf{R}}_{\mathsf{relaxed}}$ to $\hat{\mathsf{R}}$ and $\mathsf{g} : \mathcal{L}(\hat{\mathcal{R}}) \to \mathbb{R}^r$

	Prover $\mathbb{w} = (\vec{\boldsymbol{s}}, \vec{\boldsymbol{t}}_I)$		**Verifier** $\mathbb{x} = ((\mathsf{c}, I, \gamma), \mathcal{F}, \beta)$
1 :	$\forall i, j : (\boldsymbol{g}_{ijk})_k \leftarrow \mathsf{Split}(\langle \vec{\boldsymbol{s}}_i, \vec{\boldsymbol{s}}_j \rangle)$		
2 :	$\mathsf{c}_g \leftarrow \mathsf{Com}(\mathsf{ck}_C, (\boldsymbol{g}_{ijk})_{i,j,k})$		
3 :	$\bar{\mathsf{c}}, \vec{\boldsymbol{t}}_{\bar{I}} \leftarrow \mathsf{DCom}(\mathsf{ck}_{\bar{I}}, (\vec{\boldsymbol{s}}_j)_{j \in \bar{I}})$		
4 :	$\boldsymbol{u}_1 \leftarrow \mathsf{c}_g + \bar{\mathsf{c}}$	$\xrightarrow{\vec{\boldsymbol{u}}_1}$	
5 :	$\vec{\boldsymbol{h}}_{ij} \leftarrow \frac{1}{2}(\langle \vec{\boldsymbol{\psi}}_i, \vec{\boldsymbol{s}}_i \rangle + \langle \vec{\boldsymbol{\psi}}_j, \vec{\boldsymbol{s}}_j \rangle)$		
6 :	$(\boldsymbol{h}_{ijk})_k \leftarrow \mathsf{Split}(\boldsymbol{h}_{ij})$		
7 :	$\boldsymbol{u}_2 \leftarrow \mathsf{Com}(\mathsf{ck}_D, (\boldsymbol{h}_{ijk})_{i,j,k})$	$\xrightarrow{\boldsymbol{u}_2}$	$\vec{\boldsymbol{c}} \xleftarrow{\$} \mathcal{C}^r$
8 :	$\bar{\boldsymbol{z}} \leftarrow \sum_i \boldsymbol{c}_i \vec{\boldsymbol{s}}_i$	$\xleftarrow{\vec{\boldsymbol{c}}}$	
9 :	**return** $\bar{\boldsymbol{z}}, \vec{\boldsymbol{g}}, \vec{\boldsymbol{h}}, \vec{\boldsymbol{t}} = (\vec{\boldsymbol{t}}_I, \vec{\boldsymbol{t}}_{\bar{I}})$		**return** $(\tilde{\mathcal{F}} = (\tilde{f}_k)_k, \emptyset, \vec{\beta}', \beta')$

This final RoK is parameterized by a function g, which defines the target ℓ_∞ norm bounds $\vec{\beta}' := \mathsf{g}(\mathbb{x})$. The motivation is that different choices of g may be appropriate depending on the setting in which LaBRADot is used.

We recall again that the original LaBRADot in [9] computes and sends $\boldsymbol{u}_1$ as part of the first message in the approximate norm check. To highlight this graphically, we send the commitment in a separate message in the description of Π_{amm}, although, since there is no challenge between two consecutive prover's messages, they could have been sent together. However, this does not affect our analysis of 0-unique response for LaBRADot, as discussed in the proof of Theorem 7.

Commit-and-Prove LaBRADot RoK Π^{g}, for a norm-bounding function g, is defined as $\Pi^{\mathsf{g}}_{\mathsf{amm}} \circ \Pi_{\mathsf{agg}} \circ \Pi_{\mathsf{agg}'} \circ \Pi_{\mathsf{nrm\text{-}chck}}$.

To prove 0-unique response for LaBRADot, we restrict the relation $\hat{\mathsf{R}}$ to a subset of instances whose witnesses are *unique*, in the sense defined below.

Definition 15. *A relation* R' *satisfies* witness uniqueness *(with respect to a setup algorithm* Setup*) if, for any polynomial-time adversary* $\mathcal{A}$*, the following probability is negligible:*

$$\Pr\left[(\mathbb{x}, \mathbb{w}_b) \in \mathsf{R}' \wedge \mathbb{w}_0 \neq \mathbb{w}_1 : (\mathbb{x}, \mathbb{w}_0, \mathbb{w}_1) \leftarrow \mathcal{A}(\mathsf{pp})\right],$$

where the probability is taken over the sampling of the public parameters pp *generated by* Setup*.*

For example, the subset of $\hat{\mathsf{R}}$ containing only instances where all the witness vectors are committed has witness uniqueness because of the binding of the commitment scheme.

Finally, we define a variant of the principal relation, denoted R^∞, which coincides with R except that in addition to the ℓ_2 norm constraint on the witness vectors, there is also ℓ_∞ norm constraint on (a subset of) witness vectors decided by g. Thus

$$\mathbb{x} = (\mathcal{F}, \mathcal{F}', \vec{\beta}, \beta), \mathbb{w} = (\vec{\boldsymbol{s}}_j)_j \in \mathsf{R}^\infty \iff (\mathcal{F}, \mathcal{F}', \beta), \mathbb{w} \in \mathsf{R} \wedge \forall j : \|\vec{\boldsymbol{s}}_j\|_\infty \leq \beta_j$$

Theorem 7. *For any* $\mathsf{R}' \subset \hat{\mathsf{R}}$ *satisfying witness uniqueness. Let* $\vec{\beta}' \leftarrow \mathsf{g}(\mathbb{x})$ *with* $\mathbb{x} = ((\mathsf{c}_k, I_k, \gamma_k)_k, \mathcal{F}, \mathcal{F}', \beta)$*, be defined as follows: set* $\beta'_1 := \beta\sqrt{\tau}$*, where* τ *bounds the norm of the challenge space; assign the indices corresponding to* $\vec{\boldsymbol{t}}_{I_k}$ *in* $\vec{\beta}'$ *the values* γ_k *for each* k*; and set all remaining indices to* b*.*

Assuming hardness of MSIS for rank κ *and* ℓ_2 *norm bound* $\max\{\beta' + \sqrt{r}T\tilde{\beta}, 4T\beta', \beta' + 4T\tilde{\beta}\}$*, for rank* κ_1 *and* ℓ_2 *norm bound* $2\beta'$*, commit-and-Prove LaBRADot* Π^{g} *restricted to source relation* R' *and with target relation* R^∞ *satisfies* 0-UR .

Proof (Proof Sketch). We provide some intuition for the proof of the theorem. First, notice that the witness uniqueness assumption is necessary. Intuitively, if two distinct witnesses were valid for the same instance, then an adversary running the honest prover algorithm on each witness could generate two different accepting proofs.

In the first part of the proof, we show that the first three RoKs produce target relations satisfying witness uniqueness. This is relatively straightforward, since in these RoKs the prover only sends values that are injective functions of the witness, and the source witness is itself unique.

The most interesting part is the proof of 0-UR for Π_{amm}. Here, witness uniqueness implies that $\vec{\boldsymbol{s}}$, $\vec{\boldsymbol{t}}_I$ (which opens c), and the values $\vec{\boldsymbol{t}}_i = \sum_j b^j \vec{\boldsymbol{t}}_{i,j}$ for $i \in \bar{I}$ are uniquely determined. Since $\|\vec{\boldsymbol{t}}_{i,j}\|_\infty \leq b$,

it follows that the vectors $\vec{\boldsymbol{t}}_{\bar{I}}$ are also unique. Moreover, by uniqueness of $\vec{\boldsymbol{s}}$ and inspection of the construction, the value $\bar{\boldsymbol{z}}$ is uniquely determined as well.

We then prove uniqueness of $\vec{\boldsymbol{g}}$ via a case analysis. If the two forked transcripts produced by the adversary contain the same value $\boldsymbol{u}_1$, then uniqueness of $\vec{\boldsymbol{t}}_I$ together with the binding of $\boldsymbol{u}_1$ (which follows from the MSIS assumption) implies uniqueness of $\vec{\boldsymbol{g}}$. Otherwise, since the equation $\tilde{f}_1$ from Eq. (4) must hold, the Schwartz–Zippel lemma implies that the values $\sum_k b^k \boldsymbol{g}_{ijk}$ are uniquely determined for every i, j. Since the entries of $\vec{\boldsymbol{g}}$ are bounded in ℓ_∞ norm by b, this in turn implies uniqueness of $\vec{\boldsymbol{g}}$. An analogous argument applies to $\vec{\boldsymbol{h}}$.

Proof. We prove 0-UR for the first three RoKs $\Pi_{\mathsf{nrm\text{-}chck}}$, $\Pi_{\mathsf{agg}'}$ and Π_{agg}. It follows from Lemma 11 that their composition remains 0-UR. At the end we argue uniqueness of commit-and-prove LaBRADot Π^{g} with target relation R^∞. For simplicity we just assume a single commitment c in the instance committing to $(\boldsymbol{s}_j)_{j\in[r]}$. The extension to multiple commitments is straightforward.

Claim. $\Pi_{\mathsf{nrm\text{-}chck}}$ satisfies 0-UR when restricted to source relation R.

Proof. Since the target witness is also a witness for the source instance, then by witness uniqueness of R, the RoK satisfies 1-UR . Assume it does not satisfy 0-UR, then, because of 1-UR, we have two $\vec{p}$ and $\vec{p}'$ where $\vec{p} \neq \vec{p}'$ such that, by Eq. (3), have:

$$\sum_i \langle \sigma_{-1}(\vec{\boldsymbol{\pi}}_{ik}), \vec{\boldsymbol{s}}_i \rangle = p_k \qquad\qquad \sum_i \langle \sigma_{-1}(\vec{\boldsymbol{\pi}}_{ik}), \vec{\boldsymbol{s}}_i \rangle = p'_k.$$

Which is in obvious contradiction.

Notice that since the source and target instance-witness pairs match $\Pi_{\mathsf{nrm\text{-}chck}}$ is 0-UR as a RoK from R' to R'.

Claim. $\Pi_{\mathsf{agg}'}$ satisfies 0-UR when restricted to source relation R'.

Proof. The proof is almost identical to the proof of the previous claim. The 1-UR comes from witness uniqueness of R', since the target witness and the source witness match. To reach a contradiction, assume it does not satisfy 0-UR and 1-UR. Then by definition of the RoK, we have two distinct vectors $(\boldsymbol{b}''_k)_k$ and $(\tilde{\boldsymbol{b}}''_k)_k$ where for any k:

$$q''(\vec{\boldsymbol{s}}) = \boldsymbol{b}''_k \qquad\qquad q''(\vec{\boldsymbol{s}}) = \tilde{\boldsymbol{b}}''_k$$

Which is an obvious contradiction.

We notice Π_{agg} satisfies 0-UR , since there is only one message from the verifier.

Now we argue 0-UR of LaBRADot Π^{g}. Let $\mathcal{A}$ be an adversary against 0-UR game of Π^{g}. Such adversary outputs a commitment c and two distinct accepting proofs $\pi^{(1)}, \pi^{(2)}$ where $\pi^{(1)}$ and $\pi^{(2)}$ are distinct valid proofs for instance $\mathbb{x} = (\mathsf{c}, \mathcal{F}, \mathcal{F}', \beta)$ in R'. Observe that such two proofs must agree on the prefix $(\pi_{\mathsf{nrm\text{-}chck}}, \pi_{\mathsf{agg}'}, \pi_{\mathsf{agg}})$ by 0-UR of the composition $\Pi_{\mathsf{agg}} \circ \Pi_{\mathsf{agg}'} \circ \Pi_{\mathsf{nrm\text{-}chck}}$. Hence let $\pi^{(1)}|_{\Pi_{\mathsf{amm}}}$ and $\pi^{(2)}|_{\Pi_{\mathsf{amm}}}$ be the suffixes of $\pi^{(1)}$ and $\pi^{(2)}$ containing only the Π_{amm} proofs, these are two distinct proofs for the same instance of form $\mathbb{x} = (\mathsf{c}, \{F\}, \emptyset, \beta)$ in R' computed using the public reducibility function of $\Pi_{\mathsf{agg}} \circ \Pi_{\mathsf{agg}'} \circ \Pi_{\mathsf{nrm\text{-}chck}}$.
We can build wrappers $\mathcal{A}_1$ and $\mathcal{A}_2$ around $\mathcal{A}$ that each internally runs $\mathcal{A}$, and $\mathcal{A}_i$ outputs $\mathbb{x}, \pi^{(i)}$. Both $\mathcal{A}_1$ and $\mathcal{A}_2$ are valid knowledge-soundness adversary. Let $\mathcal{E}$ be the knowledge soundness extractor of Π^{g}.

Consider the following event HonGen_i:

- Let $\vec{\boldsymbol{s}}^{(i)}, \vec{\boldsymbol{t}}_I^{(i)} \leftarrow \mathcal{E}^{\mathcal{A}_i}(\mathsf{pp}, \mathbb{x}, \pi^{(i)})$
- $\pi^h \leftarrow \Pi^{\mathsf{g}}.\mathsf{P}(\mathsf{pp}, \mathbb{x}, \vec{\boldsymbol{s}}^{(i)})$
- The event is true iff $\pi^h = \pi^{(i)}|_{\Pi_{\mathsf{amm}}}$ and $\pi^{(i)}|_{\Pi_{\mathsf{amm}}}$ verifies.

First we notice that $(\vec{\boldsymbol{s}}^{(1)}, \vec{\boldsymbol{t}}_I^{(1)}) = (\vec{\boldsymbol{s}}^{(2)}, \vec{\boldsymbol{t}}_I^{(2)})$ as otherwise we can break the witness uniqueness of R'. Moreover, we notice that the function from the instance and witness to the proofs defined by the prover of $(\Pi_{\mathsf{amm}})_{\mathsf{FS}}$ is an injective function for any assignment of the random oracle. Thus $\mathsf{HonGen}_1 \wedge \mathsf{HonGen}_2$ implies 0-UR of Π^{g}.

We show that for any $i = 1, 2$ the event HonGen_i holds with overwhelming probability. The analysis is identical for $i = 1$ and $i = 2$, thus we assume $i = 1$ and we omit indexing from now on.

Let $\pi^h = ((\boldsymbol{u}_1^h, \boldsymbol{u}_2^h), \vec{\boldsymbol{c}}^h, (\vec{\boldsymbol{z}}^h, \vec{\boldsymbol{g}}^h, \vec{\boldsymbol{h}}^h, \vec{\boldsymbol{t}}^h))$ and $\pi^{(1)}|_{\Pi_{\mathsf{amm}}} = ((\boldsymbol{u}_1, \boldsymbol{u}_2), \vec{\boldsymbol{c}}, (\vec{\boldsymbol{z}}, \vec{\boldsymbol{g}}, \vec{\boldsymbol{h}}, \vec{\boldsymbol{t}}))$ both being verifying proof for $\mathbb{x}$.

Note first that $\vec{\boldsymbol{t}}^h = \vec{\boldsymbol{t}}$. This is because the extracted source witness is interpolated of form $\vec{\boldsymbol{s}}_i = \frac{\vec{\boldsymbol{z}} - \vec{\boldsymbol{z}}'}{\boldsymbol{c}_i - \boldsymbol{c}'_i}$. From target relation check $\boldsymbol{A}\vec{\boldsymbol{z}} = \sum_i \boldsymbol{c}_i \sum_j b^j \vec{\boldsymbol{t}}_{ij}$ it holds that $\boldsymbol{A}\vec{\boldsymbol{s}}_i = \sum_j b^j \vec{\boldsymbol{t}}_{ij}$. Hence $\sum_j b^j \vec{\boldsymbol{t}}_{ij} = \sum_j b^j \vec{\boldsymbol{t}}^h_{ij}$. Moreover, from l_∞ norm check in target relation R^∞, we conclude that for each $i \in [r]$, $\vec{\boldsymbol{t}}^h_{ij} = \vec{\boldsymbol{t}}_{ij} = \mathsf{Split}(\boldsymbol{A}\vec{\boldsymbol{s}}_i)$. Let $\vec{\boldsymbol{t}}_i := \sum b^j \vec{\boldsymbol{t}_{ij}}^h$ for any $i \in [r]$.

Since both π and π^h are accepting, $\boldsymbol{A}\vec{\boldsymbol{z}} = \sum_j \boldsymbol{c}_j \vec{\boldsymbol{t}}_i = \boldsymbol{A}\vec{\boldsymbol{z}}^h$. Thus, if non-zero, $\vec{\boldsymbol{z}} - \vec{\boldsymbol{z}}^h$ forms a non-trivial MSIS solution of norm $\beta' + \sqrt{r}T\tilde{\beta}$. Assuming MSIS for rank κ and norm $\beta' + \sqrt{r}T\tilde{\beta}$ we conclude that $\vec{\boldsymbol{z}} = \vec{\boldsymbol{z}}^h$. We compute $\bar{\mathsf{c}} := \mathsf{Com}(\mathsf{ck}_{B,\bar{I}}, \vec{\boldsymbol{t}}_{\bar{I}})$. Let $\mathsf{c}_g^h := \boldsymbol{u}_1^h - \bar{\mathsf{c}}$ and $\mathsf{c}_g := \boldsymbol{u}_1 - \bar{\mathsf{c}}$.

If $\mathsf{c}_g^h = \mathsf{c}_g$ then by binding, and because $\vec{\boldsymbol{t}}^h_{\bar{I}} = \vec{\boldsymbol{t}}_{\bar{I}}$, we have $\vec{\boldsymbol{g}} = \vec{\boldsymbol{g}}^h$. Else, assuming $\boldsymbol{u}_1 \neq \boldsymbol{u}_1^h$ we reach a contradiction. In fact, by the dot-equation $\tilde{f}_1$ defined in Eq. (4) and $\vec{\boldsymbol{z}} = \vec{\boldsymbol{z}}^h$, we have

$$\sum_{i,j} (\boldsymbol{g}_{ij} - \boldsymbol{g}^h_{ij}) \boldsymbol{c}_i \boldsymbol{c}_j = 0$$

First notice that with overwhelming probability over the challenge $\vec{\boldsymbol{c}}$, the opening $\vec{\boldsymbol{g}}$ are fixed, namely we obtain the same opening material $\vec{\boldsymbol{g}}$ from the adversary for the commitment $\boldsymbol{u}_1$ independently of the challenge, otherwise we can break binding of the commitment scheme by finding two valid opening for $\boldsymbol{u}_1$. Secondly, by the Schwartz-Zippel lemma, sampling $\vec{\boldsymbol{c}}$ conditioned on the value $\vec{\boldsymbol{g}}$ being fixed we have $\forall i, j : \sum b^k \boldsymbol{g}_{ijk} = \sum b^k \boldsymbol{g}^h_{ijk}$ with overwhelming probability also, since the target instance-witness pair is in R^∞ we have $\|\vec{\boldsymbol{g}}\|_\infty \leq b$ and $\|\vec{\boldsymbol{g}}^h\|_\infty \leq b$, and thus they must be the same value as they are both well-formed base b decomposition digits.

The latter is in contradiction with $\boldsymbol{u}_1 \neq \boldsymbol{u}_1^h$.

Similarly, if $\boldsymbol{u}_2 = \boldsymbol{u}_2^h$ then by binding we have $\vec{\boldsymbol{h}} = \vec{\boldsymbol{h}}^h$. Otherwise, by the dot-equation $\tilde{f}_2$ defined in Eq. (4) and $\vec{\boldsymbol{g}} = \vec{\boldsymbol{g}}^h$, we have

$$\sum_{i,j} (\boldsymbol{h}_{ij} - \boldsymbol{h}^h_{ij}) \boldsymbol{c}_i \boldsymbol{c}_j = 0$$

Again, by binding , then by the Schwartz-Zippel lemma, and finally by the ∞-norm argument we have $\forall i, j : \boldsymbol{h}_{ij} = \boldsymbol{h}^h_{ij}$ with overwhelming probability, which is in contradiction with $\boldsymbol{u}_2 \neq \boldsymbol{u}_2^h$.

The final observation is that the original scheme (see Fig. 11) the prover sends $\boldsymbol{u}_1$ at the first round of the composed scheme. We can reduce an adversary $\mathcal{A}'$ for the 0-UR of the scheme in Fig. 11 to an adversary $\mathcal{A}$ of Π easily. The idea is that $\mathcal{A}$ runs $\mathcal{A}'$ internally and whenever $\mathcal{A}'$ sends a first message $(\mathbb{x}, \boldsymbol{u}_1)$ for commit-and-prove LaBRADot to RO, in order to obtain the first challenge, the reduction $\mathcal{A}$ simply forwards $\mathbb{x}$. $\mathcal{A}$ proceeds similarly for the other RO queries of $\mathcal{A}'$. When finally $\mathcal{A}'$ sends $(\mathbb{x}, \boldsymbol{u}_1, \vec{p}, (\boldsymbol{b}''_k)_k, \boldsymbol{u}_2)$ the reduction $\mathcal{A}$ the RO with the correct order $(\mathbb{x}, \vec{p}, (\boldsymbol{b}''_k)_k, \boldsymbol{u}_1, \boldsymbol{u}_2)$.

7.3 Approximate Norm Check with partial Zero-Knowledge

We give a partial zero-knowledge RoK performing approximate norm check. The RoK is $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$. We present such RoK in Fig. 8 , the idea is similar to the approximate range proof in [48] (see the full version [47, Section 5.1]), where approximate norm check is performed using a twist of the JL projections technique and zero-knowledge obtained using rejection sampling. Our version is adapted to $\hat{\mathsf{R}}$.

The target instance contains additional dot-product functions than the start, for checking the correct generation of projected witness.

Namely, in Fig. 8, let $\lambda = 2\lambda_1 + 2\lambda_2 + \kappa_1$, the functions f'_k are defined:

$$\begin{aligned} f'_k(\vec{\boldsymbol{s}}, \vec{\boldsymbol{y}}) &:= \textstyle\sum_i \langle \sigma_{-1}(\pi_i^{(k)}), \vec{\boldsymbol{s}}_i \rangle + \langle \boldsymbol{\delta}_k, \vec{\boldsymbol{y}}_1 \rangle - (\vec{z_1})_k && 0 < k \leq 2\lambda_1, \\ f'_k(\vec{\boldsymbol{s}}, \vec{\boldsymbol{y}}) &:= \textstyle\sum_{ij} \langle \sigma_{-1}((\pi'_{ij})^{(k)}), \vec{\boldsymbol{t}}_{ij} \rangle + \langle \boldsymbol{\delta}_k, \vec{\boldsymbol{y}}_2 \rangle - (\vec{z_2})_k && 2\lambda_1 < k \leq 2\lambda. \end{aligned} \tag{5}$$

The last κ_1 functions f'_k encode the equation $\mathsf{c}_y = \mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{y}})$. In particular, we parse $\vec{\boldsymbol{y}}$ as $(\vec{\boldsymbol{y}}_{ij})_{i,j}$ with $\vec{\boldsymbol{y}}_{ij} \in \mathcal{R}_q^{\kappa_1}$, and enforce the condition $\mathsf{c}_y - \sum_{i,j} \boldsymbol{B}_{ij} \vec{\boldsymbol{y}}_{ij} = 0$ coordinate-wise.

Note that in practice the dimension of witness is much larger than security parameter λ_1, λ_2, hence we embed an integer vector in $\mathbb{Z}_q^{2\lambda_i}$ to $\mathcal{R}_q^{2\lambda_i}$ naively, by regarding each integer coordinate as a constant term polynomial, rather than padding with zeros then coefficient embedding. Further optimizations are possible by packing the projections into higher-degree coefficients, but for simplicity we omit these optimizations here.

Approximate Norm-Check ZK-RoK $\Pi_{\mathsf{zk\text{-}norm}}$ from $\hat{\mathsf{R}}$ to $\hat{\mathsf{R}}_{\mathsf{relaxed}}$

Prover		**Verifier**
$\mathbb{w} = (\vec{\boldsymbol{s}}, \vec{\boldsymbol{t}})$		$\mathbb{x} = (\mathsf{c}, \mathcal{F}, \mathcal{F}', \beta)$
$\vec{y_1} \leftarrow D_{\sigma_1}^{2\lambda_1}, \vec{y_2} \leftarrow D_{\sigma_2}^{2\lambda_2}, \vec{\boldsymbol{r}} \xleftarrow{\$} U_\eta^{n_r}$		
$\mathsf{c}_y \leftarrow \mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{y}}),\ \vec{\boldsymbol{y}} := (\vec{\boldsymbol{y}}_1 \| \vec{\boldsymbol{y}}_2 \| \vec{\boldsymbol{r}})$	$\xrightarrow{\mathsf{c}_y}$	$\forall i \in [r] : \Pi_i \xleftarrow{\$} \mathsf{Bin}_1^{2\lambda_1 \times nd}$
$\vec{z_1} \leftarrow \sum_i \Pi_i \vec{s}_i + \vec{y_1}$	$\xleftarrow{(\Pi_i)_i, (\Pi'_{ij})_{i,j}}$	$\forall (i,j) \in [r] \times [t] : \Pi'_{ij} \xleftarrow{\$} \mathsf{Bin}_1^{2\lambda_2 \times \kappa d}$
$\vec{z_2} \leftarrow \sum_{ij} \Pi'_{ij} \vec{t_{ij}} + \vec{y_2}$		
Abort if $\exists i : \mathsf{Rej}(\vec{z_i}, \vec{z_i} - \vec{y_i}, \sigma_i, \rho) = 1$	$\xrightarrow{\vec{z}}$	$\forall i : \|\vec{z_i}\|_2 \overset{?}{<} \sigma_i \sqrt{4\lambda_i}$
return $(\vec{\boldsymbol{s}}, \vec{\boldsymbol{y}}), \vec{\boldsymbol{t}}$		**return** $(\mathsf{c}, \mathcal{F}, \mathcal{F}' \cup (f'_k)_k, \beta', \gamma)$

Fig. 8: The RoK for approximate norm check with partial zero-knowledge. We define U_η to be the distribution sampling uniformly random $\mathcal{R}_q$ element with infinity norm at most η. The functions f'_k are defined in Eq. (5). Formally, the source instance is $\mathbb{x} = ((\mathsf{c}_i, I_i, \gamma_i)_i, \mathcal{F}, \mathcal{F}', \beta, \gamma)$ and the target instance is $((\mathsf{c}_i, I_i, \gamma_i)_i, (\mathsf{c}_y, I_y), \mathcal{F}, \mathcal{F}' \cup (f'_k)_k, \beta', \gamma)$ where $I_y = [r+1, r+r_y]$, $r_y = \lceil (n_r + 2(\lambda_1 + \lambda_2))/n \rceil$ and $\beta' = \sqrt{\beta^2 + 4\lambda_1\sigma_1^2 + 4\lambda_2\sigma_2^2 + n_r\eta^2 + \gamma^2}$. Target witness $(\vec{\boldsymbol{s}}, \vec{\boldsymbol{y}}), \vec{\boldsymbol{t}}$ satisfies the functions in $\mathcal{F}, \mathcal{F}' \cup (f'_k)_k$ and $\mathsf{Open}(\mathsf{c}, \beta', \gamma, \vec{\boldsymbol{s}}, \vec{\boldsymbol{t}}) = 1$.

Theorem 8 ($\Pi_{\mathsf{zk\text{-}norm}}$). *The RoK $\Pi_{\mathsf{zk\text{-}norm}}$ is complete, knowledge sound, non-reprogramming partial zero-knowledge, and $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$.*

More precisely, let $\sigma_1, \sigma_2 \in \mathbb{R}$ be such that $\sigma_1 = \omega(\beta\sqrt{\log \lambda_1})$ and $\sigma_2 = \omega(\gamma\sqrt{\log \lambda_2})$. Then there exist constants $\rho_1, \rho_2, C_1, C_2 > 0$ such that completeness holds. The knowledge-soundness error is $2^{-\lambda_1} + 2^{-\lambda_2}$, moreover, the knowledge extractor outputs a witness $\vec{\boldsymbol{s}}$ with norm bounded by $\sigma_1\sqrt{4\lambda_1/D_2}$ and a witness $\vec{\boldsymbol{t}}$ with norm bounded by $\sigma_2\sqrt{4\lambda_2/D_2}$.

We first give a proof intuition. $\Pi_{\mathsf{zk\text{-}norm}}$ performs JL projection on witness $\vec{\boldsymbol{s}}$ and $\vec{\boldsymbol{t}}$. Completeness follows from the concentration lemma of JL projection, namely Lemma 9. For soundness, by target relation check the proof element $\vec{z}$ is forced to be well-formed, thus it follows from the anti-concentration lemma, namely Lemma 10, that the extracted witness is short. For partial zero-knowledge, since hiding Ajtai commitment is used, commitment c_y is close to uniform; since rejection sampling is used, proof element $\vec{z}$ is close to Gaussian.

Lemma 13 (Completeness). *Let $\sigma_1, \sigma_2 \in \mathbb{R}$ be such that $\sigma_1 = \omega(\beta\sqrt{\log \lambda_1})$, $\sigma_2 = \omega(\gamma\sqrt{\log \lambda_2})$. There exists constant $\rho_1, \rho_2, C_1, C_2 > 0$ such that the RoK $\Pi_{\mathsf{zk\text{-}norm}}$ is complete.*

Proof. We first analyse the aborting probability when P and V are honest. By Lemma 9, $\| \sum_i \Pi_i \vec{s}_i \|_2 \leq \sqrt{C_2}\beta$ with probability $1 - 2^{-\lambda_1}$, by rejection sampling, Lemma 5 there exists constant ρ_1 such that the aborting probability of P is within $2^{-\omega(\log \lambda_1)}$ of $1/\rho_1$. Moreover,by the tail bound, Lemma 4, $\vec{z}$ has norm bound $\sigma_1\sqrt{4\lambda_1}$ with probability $1 - 2^{-\lambda_1/2}$. Similarly for $\vec{\boldsymbol{t}}$. Function satisfiability follows by inspection and from Lemma 1. □

Lemma* 14 (Knowledge Soundness). *$\Pi_{\mathsf{zk\text{-}norm}}$ is knowledge sound with soundness error $2^{-\lambda_1} + 2^{-\lambda_2}$, extractor outputs witness $\vec{\boldsymbol{s}}$ of norm bound $\sigma_1 \cdot \sqrt{\frac{4\lambda_1}{D_2}}$, witness $\vec{\boldsymbol{t}}$ of norm bound $\sigma_2 \cdot \sqrt{\frac{4\lambda_2}{D_2}}$.*

Proof. Trivially, the extractor simply outputs the vectors $\vec{\boldsymbol{s}}$ defined by the adversary as part of the target witness. Thus, on an accepting transcript, by construction of the target instance the equations $(f_i)_{i \in [K]}$ hold for $\vec{\boldsymbol{s}}$.

Moreover, on an accepting transcript, one has that $\| \sum_i \Pi_i \vec{s}_i + \vec{y_1} \|_2 < \sigma_1\sqrt{4\lambda_1}$, by definition of the new dot-product functions $(f'_k)_k$ and the verifier's checks. By Lemma 10, $\| \vec{\boldsymbol{s}} \|_2 < \sigma_1\sqrt{\frac{4\lambda_1}{D_2}}$ with soundness error $2^{-\lambda_1}$. An identical argument applies to $\vec{\boldsymbol{t}}$. □

Lemma 15 (Partial zero-knowledge). *For $n_r > \frac{d\kappa_1 \log q + 2\lambda}{\log(2\eta+1)}$, $\Pi_{\mathsf{zk\text{-}norm}}$ is non-reprogramming partial zero-knowledge.*

Proof. We describe below a simulation algorithm that takes as input public parameters pp, family of dot product functions $(f_i)_{i\in[K]}$ and double commitment c_s.

$\underline{\mathcal{S}(\mathsf{pp};\mathbb{x})}$:

- Sample $\mathsf{c}_y \xleftarrow{\$} \mathcal{R}_q^{\kappa_1}$, $\vec{z_1} \leftarrow D_{\sigma_1}^{2\lambda_1}$, $\vec{z_2} \leftarrow D_{\sigma_2}^{2\lambda_2}$
- Obtain challenge $(\Pi_i)_{i\in[r]}, (\Pi'_{i,j})_{i\in[r],j\in[t]} \leftarrow \mathsf{RO}(\mathsf{pp}\|\mathbb{x}\|\mathsf{c}_y)$.
- Output $\pi := (\mathsf{c}_y; (\Pi_i); \vec{z_1}, \vec{z_2})$.

By Lemma 5, when prover doesn't abort, $\vec{z_i}$ has distribution within statistical distance of $2^{-\omega(\log \lambda_i)}$ of Gaussian.

Moreover, by hiding of Com^*, Lemma 32, c_y is distributed statistically close the commitment generated by the real proof. □

Corollary 1. *$\Pi_{\mathsf{zk\text{-}norm}}$ is $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$.*

The corollary readily follows from the previous lemmas and Theorem 4.

7.4 Reducing $\mathcal{F}'$ to $\mathcal{F}$ with partial zero-knowledge

We describe a partial zero-knowledge version of $\Pi_{\mathsf{agg}'}$ from LaBRADot. Similarly to $\Pi_{\mathsf{agg}'}$, the RoK $\Pi_{\mathsf{zk\text{-}agg}'}$ is a reduction of knowledge from $\hat{\mathsf{R}}$ to $\hat{\mathsf{R}}$ that maps a list $\mathcal{F}'$ of constant-term dot-product equations to $\lambda' = \lambda/\log q$ (standard) dot-product equations. Informally, the RoK executes $\Pi_{\mathsf{agg}'}$ on an instance-witness tuple augmented with dummy constant-term dot-equations and random witness polynomials that mask the higher-degree coefficients sent by the prover in $\Pi_{\mathsf{agg}'}$.

More precisely, the RoK $\Pi_{\mathsf{zk\text{-}agg}'}$, on input instance $\mathbb{x} = (\mathsf{c}, \mathcal{F}, \mathcal{F}')$ and witness $\mathbb{w} = (\vec{\boldsymbol{s}}, \vec{\boldsymbol{t}})$, proceeds as follows:

1. The prover samples $\lambda' = \lambda/\log q$ polynomials $(\boldsymbol{g}_k)_{k\in[\lambda']}$ such that $\boldsymbol{g}_k \in \mathcal{R}_q$, $\mathsf{ct}(\boldsymbol{g}_k) = 0$ and its higher order coefficients are sampled uniformly over $\mathbb{Z}_q$. It computes $(\boldsymbol{g}_{k,j})_j \leftarrow \mathsf{Split}_b(\boldsymbol{g}_k)$ for any k and sets $\vec{\boldsymbol{g}} = (\boldsymbol{g}_{k,j})_{k,j}$.
   It computes $\mathsf{c}_g, \vec{\boldsymbol{t}}_g \leftarrow \mathsf{DCom}^*(\mathsf{ck}_{[r+1,r+r_g]}, \vec{\boldsymbol{g}})$, where $r_g := \lceil(\lambda/b)/n\rceil$. (For practical applications, we may assume $n \gg \lambda/b$, thus $r_g = 1$).
2. The prover and verifier then execute $\Pi_{\mathsf{agg}'}$ on the instance $(\mathsf{c}, \mathsf{c}_g, \mathcal{F}, \mathcal{F}' \cup (f''_k)_{k\in[\lambda']})$ with witness $(\vec{\boldsymbol{s}}, \vec{\boldsymbol{g}}), (\vec{\boldsymbol{t}}, \vec{\boldsymbol{t}_g})$, where each function is defined as $f''_k(\vec{\boldsymbol{s}}, \vec{\boldsymbol{g}}) := \sum b^j \boldsymbol{g}_{k,j}$, that is, f''_k selects the k-th random mask. By construction, $\mathsf{ct}(f''_k(\vec{\boldsymbol{s}}, \vec{\boldsymbol{g}})) = 0$ for all k.

We give the full protocol in Section B.

Theorem 9. *The RoK $\Pi_{\mathsf{zk\text{-}agg}'}$ is complete, knowledge sound, non-reprogramming partial zero-knowledge and $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$. Concretely, $\Pi_{\mathsf{zk\text{-}agg}'}$ has soundness error $1/q^{\lambda'}$.*

Proof (sketch). We provide a proof sketch. Completeness follows immediately. For soundness, each new function satisfiability in target instance implies $\vec{\boldsymbol{s}}$ satisfies the constant term dot product functions in source instance with probability $1 - 1/q$ by Schwarz-Zippel lemma, hence negligible after λ' many iterations. For partial zero-knowledge, since hiding version of DCom is used, element c_g is close to uniform. Moreover, proof element $\boldsymbol{b}_k$ has uniformly random higher order coefficients. To this end, $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ follows from Theorem 4.

We provide the lemmas for knowledge soundness and partial zero-knowledge. $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ is implied by Theorem 4. Completeness follows immediately, as the prover only introduces equations that evaluate to zero on the newly added variables.

Lemma 16 (Partial Zero-knowledge). *$\Pi_{\mathsf{zk\text{-}agg}'}$ is non-reprogramming partial zero-knowledge.*

Proof. We describe below a simulation algorithm that takes as input public parameters pp, family of dot product functions $\mathcal{F}, \mathcal{F}'$ and double commitment c_s.

$\underline{\mathcal{S}(\mathsf{pp};\mathbb{x})}$:

- Sample $\mathsf{c}_g \xleftarrow{\$} \mathcal{R}_q^{\kappa_1}$
- Obtain challenge $(\vec{\psi_k})_{k\in[\lambda']} \leftarrow \mathsf{RO}(\mathsf{pp}\|\mathbb{x}\|\mathsf{c}_g)$
- Computes $b_{k0} = \sum_{j\in[K']} \psi_{kj} b'_j$ for $k \in [\lambda']$

– Sample $b_{ki} \xleftarrow{\$} \mathbb{Z}_q$ for $k \in [\lambda']$, $i \in [1, d-1]$, let $\boldsymbol{b}_k := \sum_{i=0}^{d-1} b_{ki} X^i$
– Output $\pi := (\mathsf{c}_g; (\overrightarrow{\psi_k})_k; (\boldsymbol{b}_k)_k)$.

To argue that a simulated transcript follows the same distribution as what an honest prover would output, one notice that since double commitment is hiding, proof element c_g is uniform in $\mathcal{R}_q^{\kappa_1}$. Since $\boldsymbol{g}_k$ has its higher order coefficients uniform in $\mathbb{Z}_q$, so is $\boldsymbol{b}_k$ because, for any assignment of the $\vec{\boldsymbol{s}}$, there exists a full-rank linear map mapping values $(\boldsymbol{b}_k)_k$ to $(\boldsymbol{g}_k)_k$. Moreover, its constant term coefficient is uniquely determined by verification equation.

Lemma 17 (Knowledge Soundness). *$\Pi_{\mathsf{zk\text{-}agg}'}$ satisfies knowledge soundness with soundness error $1/q^{\lambda'}$*

Proof. On a transcript that is accepting, it suffices to argue that $\vec{\boldsymbol{s}}$ satisfies functions in family $\mathcal{F}'$. Target relation check and verification check implies that

$$\forall k \in [\lambda'] : \sum_{j \in [K']} \psi_{k,j}(\mathsf{ct}(f'_j(\vec{\boldsymbol{s}}))) + \sum_{j \in [\lambda']} \psi_{k,(K'+j)} \mathsf{ct}(\boldsymbol{g}_k) = 0 \tag{6}$$

Applying Schwarz-Zippel lemma, for any k the probability that $\exists j : \mathsf{ct}(f'_j(\vec{\boldsymbol{s}})) \neq 0$ and Eq. (6) above holds for the index k is $1/q$. By independence, the probability $\exists j : \mathsf{ct}(f'_j(\vec{\boldsymbol{s}})) \neq 0$ and Eq. (6) holds is $1/q^{\lambda'}$.

7.5 The non-linear-term RoK for the Committed Principal Relation

We describe the first (partial) zero-knowledge RoK for LaBRADot, named Π_1 and described in Fig. 9, which masks the witness variables of the source instance but introduces extra unmasked variables.

We describe Π_1 as a RoK from $\hat{\mathsf{R}}$ to $\hat{\mathsf{R}}$. However, we can only establish relaxed knowledge soundness. Accordingly, in the security analysis we interpret Π_1 as a RoK from $\hat{\mathsf{R}}_{\mathsf{relaxed}}$ to $\hat{\mathsf{R}}$. Also note that, strictly speaking, this is a RoK from the subset of $\hat{\mathsf{R}}_{\mathsf{relaxed}}$ consisting of instances whose list of constant-term dot-product equations is empty. This is sufficient for our purposes, since in the composition this RoK is invoked only after $\Pi_{\mathsf{zk\text{-}agg}'}$. Formally, we define the subset $\mathsf{R}_1 \subseteq \hat{\mathsf{R}}_{\mathsf{relaxed}}$ such that $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}_1$ implies $\mathcal{F}' = \emptyset$.

The RoK Π_1 is described in Fig. 9. For simplicity, we assume that all witness vectors are committed in c. If this is not the case, Π_1 can be easily modified to additionally commit any remaining uncommitted vectors.

The RoK satisfies a stronger property than merely being a partial zero-knowledge RoK from $\hat{\mathsf{R}}_{\mathsf{relaxed}}$ to $\hat{\mathsf{R}}$. To formalize this, we introduce an extended target relation $\tilde{\mathsf{R}}_2$ and show that the RoK is partial zero-knowledge with respect to this extended relation.

The extended relation is defined as follows. Suppose that the source instance consists of K functions f_i and a witness composed of r vectors. By inspecting the RoK Π_1 in Fig. 9, we see that the target instance contains $\tilde{K} := 2K + \kappa_1$ functions and a witness consisting of $r + r_w$ vectors where $r_w := \lceil (4K + n_r) \log q / (n \log b) \rceil$, which we parse as two collections of vectors $\vec{\boldsymbol{v}}$ and $\vec{\boldsymbol{w}}$. We define the following extended relation:

$$\mathbb{x},\ \mathbb{w} = \left((\vec{\boldsymbol{v}}, \vec{\boldsymbol{w}}) \in (\mathcal{R}_q^n)^{r+r_w},\ \vec{\boldsymbol{t}_v}\right) \in \hat{\mathsf{R}}' \iff \left((\mathbb{x}, \vec{\boldsymbol{v}}, \vec{\boldsymbol{t}_v}),\ \vec{\boldsymbol{w}}\right) \in \tilde{\mathsf{R}}_2 \tag{7}$$

That is, we move the component $\vec{\boldsymbol{v}} \in (\mathcal{R}_q^n)^r, \vec{\boldsymbol{t}_v} \in (\mathcal{R}_q^{\kappa t})^r$ from the witness to the target instance, thus strengthening the partial zero-knowledge requirement (the simulator needs to simulate $\vec{\boldsymbol{v}}$ and $\vec{\boldsymbol{t}_v}$ as well). Note that the vector $\vec{\boldsymbol{w}}$ in the witness is not committed.

The first step of Π_1 is to process the instance. This processing is independent of the challenges, and we therefore refer to it as preprocessing. In particular, we define an algorithm PreP that, on input the instance $\mathbb{x} = ((\mathsf{c}, [r], \gamma), \mathcal{F} = (f_k)_{k \in [K]}, \emptyset, \beta)$, outputs a list of functions f'_k and f''_k and constant $\boldsymbol{b}_k$ for each $k \in [K]$ such that:

$$\forall k : f_k(\vec{\boldsymbol{s}}) = \underbrace{\sum_{i,j} \boldsymbol{a}_{kij} (\langle \vec{\boldsymbol{s}}_i, \vec{\boldsymbol{s}}_j \rangle) - \boldsymbol{s}'_k}_{f'_k(\vec{\boldsymbol{s}}, \boldsymbol{s}'_k)} + \underbrace{\boldsymbol{s}'_k + \sum_i \langle \vec{\boldsymbol{\rho}}_{ki}, \vec{\boldsymbol{s}}_i \rangle}_{f''_k(\vec{\boldsymbol{s}}, \boldsymbol{s}'_k)} + \boldsymbol{b}_k \tag{8}$$

The RoK Π_1 follows the ideas introduced in Nova [43]. We first sample a random vector $\vec{\boldsymbol{z}}$, evaluate the functions on this vector, and commit to their evaluations. As in Nova, we also commit to the cross terms

Non-linear-term ZK-RoK Π_1 from R_1 to $\hat{\mathsf{R}}$

	Prover		**Verifier**
2 :	Parse $\mathbb{w} = (\vec{\boldsymbol{s}}, \vec{\boldsymbol{t}})$,		
3 :	$(f'_k, f''_k, \boldsymbol{b}_k)_k \leftarrow \mathsf{PreP}(\mathbb{x})$		
4 :	$\vec{\boldsymbol{z}} = (\vec{\boldsymbol{z}}_i)_{i\in[r]} \leftarrow (D_\sigma^{nd})^r, \vec{\boldsymbol{r}} \xleftarrow{\$} U_\eta^{n_r}$		
5 :	$\mathsf{c}_z, \vec{\boldsymbol{t}_z}, \vec{\boldsymbol{t}_z}' \xleftarrow{\$} \mathsf{DCom}^*(\vec{\boldsymbol{z}})$		
6 :	**for** $k \in [K]$:		
7 :	$(\boldsymbol{s}'_{kj})_j \leftarrow \mathsf{Split}_{b_e}(f'_k(\vec{\boldsymbol{s}}, \boldsymbol{0}))$		
8 :	$(\boldsymbol{z}'_{kj})_j \leftarrow \mathsf{Split}_{b_e}(f'_k(\vec{\boldsymbol{z}}, \boldsymbol{0}))$		
9 :	$(\boldsymbol{z}''_{kj})_j \leftarrow \mathsf{Split}_{b_e}(f''_k(\vec{\boldsymbol{z}}, \boldsymbol{0}))$		
10 :	$\boldsymbol{e}_k \leftarrow \sum_{ij} \boldsymbol{a}_{kij}\langle\vec{\boldsymbol{s}}_i, \vec{\boldsymbol{z}}_j\rangle + \langle\vec{\boldsymbol{s}}_j, \vec{\boldsymbol{z}}_i\rangle$		
11 :	$(\boldsymbol{e}_{kj})_j \leftarrow \mathsf{Split}_{b_e}(\boldsymbol{e}_k)$		
12 :	**Let** $\vec{\boldsymbol{w}} = (\boldsymbol{s}'_{kj}, \boldsymbol{z}'_{kj}, \boldsymbol{z}''_{kj}, \boldsymbol{e}_{kj})_{k,j}, \vec{\boldsymbol{r}}$		
13 :	$\mathsf{c}_w \leftarrow \mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{w}})$	$\xrightarrow{\mathsf{c}_w, \mathsf{c}_z}$	$\boldsymbol{\alpha} \xleftarrow{\$} \mathcal{C}$
14 :	$(\vec{\boldsymbol{v}}, \vec{\boldsymbol{t}_v}) \leftarrow (\vec{\boldsymbol{z}}, \vec{\boldsymbol{t}_z}) + \boldsymbol{\alpha} \cdot (\vec{\boldsymbol{s}}, \vec{\boldsymbol{t}})$	$\xleftarrow{\boldsymbol{\alpha}}$	$\mathsf{c}_v \leftarrow \mathsf{c}_z + \boldsymbol{\alpha}\mathsf{c}$
15 :	Abort if $\mathsf{Rej}(\vec{\boldsymbol{v}}, \boldsymbol{\alpha}\vec{\boldsymbol{s}}, \sigma, \rho) = 1$		$\tilde{\mathcal{F}} \leftarrow (\tilde{f}_k)_{k\in[\tilde{K}]}$
16 :	Or $\mathsf{Rej}(\vec{\boldsymbol{t}_z} + \boldsymbol{\alpha}\vec{\boldsymbol{t}}, \boldsymbol{\alpha}\vec{\boldsymbol{t}} + \vec{\boldsymbol{t}_z}', \sigma', \rho) = 1$		$\mathbb{x}' \leftarrow ((\mathsf{c}_v, [r], \gamma_v), \tilde{\mathcal{F}}, \emptyset, \beta')$
17 :	**return** $(\vec{\boldsymbol{v}}, \vec{\boldsymbol{w}}), \vec{\boldsymbol{t}_v}$		**return** $\mathbb{x}'$

Fig. 9: The RoK for non-linear terms. This RoK masks the initial variables $(\vec{\boldsymbol{s}}_j)_{j\in[r]}$, but introduces additional variables $\vec{\boldsymbol{w}} \in (\mathcal{R}_q^n)^{r_w}$ that are not masked. Notice that in line 13, $\vec{\boldsymbol{w}}$ is parsed as a list of vectors in $\mathcal{R}_q^\kappa$.

$(\boldsymbol{e}_k)_k$. We then fold the two instances, the original instance corresponding to $\vec{\boldsymbol{s}}$ and the freshly sampled one corresponding to $\vec{\boldsymbol{z}}$, into a single instance.

The newly introduced variables, namely the evaluations of the functions on $\vec{\boldsymbol{z}}$ and the cross terms, are first split, because their norms are arbitrary big, and then committed all together using Com. We additionally include in this commitment a random vector $\vec{\boldsymbol{r}} \in \mathcal{R}_q^{n_r}$, where $n_r := \frac{\kappa_1 \log q + 2\lambda/d}{\log(2\eta+1)}$,of small norm η to ensure that the resulting commitment c_w is hiding (see Lemma 32). Recall, we define U_η to be the distribution sampling uniformly random $\mathcal{R}_q$ element with infinity norm at most η.

The target instance $\mathbb{x}'$ output by Π_1 is $((\mathsf{c}_v, [r], \gamma_v), (\tilde{f}_k)_{k\in[\tilde{K}]}, \emptyset, \beta')$ where $\tilde{K} := 2K + \kappa_1$, $\gamma_v := (T+1)b\sqrt{\kappa t d}$ and $\beta' := \sqrt{2nrd\sigma^2 + b_e^2 \cdot 4K\lceil \log q / \log b_e \rceil + \eta^2 n_r}$

Let $\vec{\boldsymbol{w}} = ((\boldsymbol{s}'_{kj}, \boldsymbol{z}'_{kj}, \boldsymbol{z}''_{kj}, \boldsymbol{e}_{kj})_{k,j}, \vec{\boldsymbol{r}})$, as in Fig. 9, for $k \in [K]$ we have

$$\tilde{f}_k(\vec{\boldsymbol{v}}, \vec{\boldsymbol{w}}) := f'_k(\vec{\boldsymbol{v}}, \sum_j b^j \boldsymbol{z}'_{kj} + \boldsymbol{\alpha}(\sum_j b^j \boldsymbol{e}_{kj}) + \boldsymbol{\alpha}^2(\sum_j b^j \boldsymbol{s}'_{kj})) \tag{9}$$

$$\tilde{f}_{K+k}(\vec{\boldsymbol{v}}, \vec{\boldsymbol{w}}) := f''_k(\vec{\boldsymbol{v}}, \boldsymbol{\alpha}(\sum_j b^j \boldsymbol{s}'_{kj})) + \boldsymbol{\alpha}\boldsymbol{b}_k - \sum_j b^j \boldsymbol{z}''_{kj} \tag{10}$$

Additionally, the last κ_1 functions, namely the function $(\tilde{f}_{2K+k})_{k\in[\kappa_1]}$, encode the linear equations $\mathsf{c}_w = \mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{w}})$.

Notice that $\Pi_1 : \hat{\mathsf{R}}_{\mathsf{relaxed}} \to \hat{\mathsf{R}}$ only has negligible completeness error on instance-witness tuples from $\hat{\mathsf{R}}$. For this reason, we define $\Pi_{\mathsf{nlin}} := (\Pi_1 \circ \Pi_{\mathsf{zk\text{-}agg}'} \circ \Pi_{\mathsf{zk\text{-}norm}})$ which is a RoK $\hat{\mathsf{R}} \to \hat{\mathsf{R}}$, and for which we can formally state completeness.

Theorem⋆ 10. *The RoK Π_{nlin} is complete, knowledge sound, partial zero-knowledge and $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$. Concretely, let $\sigma, \sigma', \sigma_1, \sigma_2 \in \mathbb{R}$ be such that $\sigma = \omega(T\beta\sqrt{\log n})$, $\sigma' = \omega(T\gamma\sqrt{\log \kappa t})$, $\sigma_1 = \omega(\beta\sqrt{\log \lambda_1})$ and $\sigma_2 = \omega(\gamma\sqrt{\log \lambda_2})$, there exist constants $\rho, \rho', \rho_1, \rho_2, C_1, C_2 > 0$ such that the RoK Π_{nlin} is complete. Assume Module SIS hardness for rank κ and norm $6T\beta'$, rank κ_1 and norm $\max\{6T\gamma_v, 2\beta', \beta' + \eta'\sqrt{2n_w d} + 1\}$, Π_{nlin} has knowledge-soundness error $2^{-\lambda_1} + 2^{-\lambda_2} + 1/q^{\lambda'} + 3/|\mathcal{C}|$, the knowledge extractor outputs a witness $\vec{\boldsymbol{s}}$ with norm bounded by $\sigma_1\sqrt{4\lambda_1/D_2}$ and a witness $\vec{\boldsymbol{t}}$ with norm bounded by $\sigma_2\sqrt{4\lambda_2/D_2}$. Moreover, Π_{nlin} satisfies $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$.*

Proof (sketch). We provide a proof sketch. Recall that $\Pi_{\mathsf{nlin}} := (\Pi_1 \circ \Pi_{\mathsf{zk\text{-}agg}'} \circ \Pi_{\mathsf{zk\text{-}norm}})$, we first argue that Π_1 is complete on instance-witness tuples from $\hat{\mathsf{R}}$ and $\tilde{\Pi}_1$ is partial zero-knowledge for the (target) extended relation $\tilde{\mathsf{R}}_2$. Completeness follows from the two rejection samplings Lemma 5 and Lemma 6. Partial zero-knowledge follows from rejection sampling and hiding commitments, moreover, the partial simulator would reprogram the challenge $\boldsymbol{\alpha}$. For knowledge soundness, note that since the challenge set $\mathcal{C}$ is subtractive, the extractor can extract weak opening $(\vec{\boldsymbol{s}}, \bar{\boldsymbol{\alpha}})$ to commitment c, where $\vec{\boldsymbol{s}}$ satisfies the functions in source instance by Schwarz-Zippel Lemma 3 with probability $1 - 3/|\mathcal{C}|$. Then extractor in $\Pi_{\mathsf{zk\text{-}norm}}$ shows that $\vec{\boldsymbol{s}}$ has norm bounded by $\sigma_1\sqrt{4\lambda_1/D_2}$ with overwhelming probability. Similarly for $\vec{\boldsymbol{t}}$. For $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$, by Theorem 4, it suffices to argue $\tilde{\Pi}_1$ satisfies 1-HtoInv, which follows from the observation that since c_w is uniformly random, from an adversary against 1-HtoInv one can emulate an adversary against MSIS rank κ_1 and norm $2(\gamma_w + \sigma\sqrt{2n_w d})$. □

We prove Theorem 10 through a sequence of lemmas, one for each property stated in the theorem.

Lemma 18. *Let $\sigma, \sigma' \in \mathbb{R}$ be such that $\sigma = \omega(T\beta\sqrt{\log n})$, $\sigma' = \omega(T\gamma\sqrt{\log \kappa t})$, there exist constants $\rho, \rho' > 0$ such that the RoK Π_{nlin} is complete.*

Proof. Notice that $(\Pi_{\mathsf{zk\text{-}agg}'} \circ \Pi_{\mathsf{zk\text{-}norm}})$ is complete and the target instance and target witness are of the form $(\mathsf{c}, \beta, (f_k)_k), (\vec{\boldsymbol{s}}_j)_j$.

We analyse the aborting probability when P and V are honest and assuming that the source instance and source witness are of this form. Since $\|\boldsymbol{\alpha}\vec{\boldsymbol{s}}\|_2 \leq T\beta$, by rejection sampling, Lemma 5, there exists constant ρ such that the aborting probability of P is within $2^{-\omega(\log n)}$ of $1/\rho$. Moreover, by the tail bound for discrete gaussian, Lemma 4, $\vec{\boldsymbol{v}}$ has norm bound $\beta_v = \sigma\sqrt{2nrd}$, $\vec{\boldsymbol{t}}_v$ has norm bound $\gamma_v = \sigma'\sqrt{2\kappa trd}$ with overwhelming probability.

By (somewhat) homomorphic property we have that since $\mathsf{c}_z, \vec{\boldsymbol{t}}_z = \mathsf{DCom}(\mathsf{ck}_{[1:r]}, \vec{\boldsymbol{z}})$ and $\mathsf{c}, \vec{\boldsymbol{t}} = \mathsf{DCom}(\mathsf{ck}_{[1:r]}, (\vec{\boldsymbol{s}}_j)_j)$ then $\mathsf{Open}(\mathsf{ck}_{[1:r]}, \mathsf{c}_z + \boldsymbol{\alpha}\mathsf{c}, \vec{\boldsymbol{z}} + \boldsymbol{\alpha}\vec{\boldsymbol{s}}, \vec{\boldsymbol{t}}_z + \boldsymbol{\alpha}\vec{\boldsymbol{t}}) = 1$. Moreover, by inspection $\|\vec{\boldsymbol{w}}\|_\infty \leq \max(b, \eta)$. Hence with overwhelming probability $\|v\|_2^2 + \|w\|_2^2 \leq 2nrd\sigma^2 + b_e^2 \cdot 4K\lceil \log q / \log b_e \rceil + \eta^2 n_r = \beta^2$.

We verify that the equations $\tilde{f}_i$,for $i \in [2K]$, hold. We set

$$\boldsymbol{z}'_k = \sum_j b^j \boldsymbol{z}'_{kj}, \qquad \boldsymbol{z}'_k = \sum_j b^j \boldsymbol{z}''_{kj}, \qquad \boldsymbol{s}'_k = \sum_j b^j \boldsymbol{s}'_{kj}. \tag{11}$$

For $k \in [K]$ we have

$$\begin{aligned}
\tilde{f}_i(\vec{\boldsymbol{v}}, \vec{\boldsymbol{w}}) &= \textstyle\sum_{i,j} \boldsymbol{a}_{kij} \langle \vec{\boldsymbol{v}}_i, \vec{\boldsymbol{v}}_j \rangle - (\boldsymbol{z}'_k + \boldsymbol{\alpha}\boldsymbol{e}_k + \boldsymbol{\alpha}^2 \boldsymbol{s}'_k) \\
&= \boldsymbol{\alpha}^2 (\textstyle\sum_{i,j} \boldsymbol{a}_{kij} \langle \vec{\boldsymbol{s}}_i, \vec{\boldsymbol{s}}_j \rangle - \boldsymbol{s}'_k) + \boldsymbol{\alpha} (\textstyle\sum_{i,j} \boldsymbol{a}_{kij} (\langle \vec{\boldsymbol{s}}_i, \vec{\boldsymbol{z}}_j \rangle + \langle \vec{\boldsymbol{z}}_i, \vec{\boldsymbol{s}}_j \rangle) - \boldsymbol{e}_k) \\
&+ (\textstyle\sum_{i,j} \boldsymbol{a}_{kij} \langle \vec{\boldsymbol{z}}_i, \vec{\boldsymbol{z}}_j \rangle - \boldsymbol{z}'_k) \\
&= 0.
\end{aligned}$$

Where in the first equation we simply open up the definition of $\vec{\boldsymbol{v}}$ and computed all the terms, while in the second equation we applied the definitions of $\boldsymbol{s}'_k, \boldsymbol{e}_k$ and $\boldsymbol{z}_k$. For $k \in [K]$ we have

$$\begin{aligned}
\tilde{f}_{K+k}(\vec{\boldsymbol{v}}, \vec{\boldsymbol{w}}) &= f''_k(\vec{\boldsymbol{v}}, \boldsymbol{\alpha}\boldsymbol{s}'_k) + \boldsymbol{\alpha}\boldsymbol{b}_k - \boldsymbol{z}''_k \\
&= \sum_i \langle \vec{\boldsymbol{\rho}}_{ki}, \vec{\boldsymbol{z}}_i \rangle + \boldsymbol{\alpha}(\boldsymbol{s}'_k + \boldsymbol{b}_k) - \boldsymbol{z}''_k \\
&= \boldsymbol{\alpha}(\sum_i \langle \vec{\boldsymbol{\rho}}_{ki}, \vec{\boldsymbol{s}}_i \rangle + \boldsymbol{s}'_k + \boldsymbol{b}_k) + \sum_i \langle \vec{\boldsymbol{\rho}}_{ki}, \vec{\boldsymbol{z}}_i \rangle - \boldsymbol{z}''_k \\
&= 0
\end{aligned}$$

Where the last equation holds because the source instance belongs to the relation thus $\sum_i \langle \vec{\boldsymbol{\rho}}_{ki}, \vec{\boldsymbol{s}}_i \rangle + \boldsymbol{s}'_k + \boldsymbol{b}_k = 0$ and by the definition of $\boldsymbol{z}''_k$. Finally notice that $\tilde{f}_k$ for $k \in [K]$ are of degree at most quadratic and $\tilde{f}_k$ for $k \in [K+1, 2K]$ are of degree at most linear. Moreover, the functions can be rewritten using the notation from Eq. (2) by defining $r_w = \lceil (4K + \kappa + \lambda) \log q / (n \log b) \rceil$ additional witness vectors, in which we arrange the new variables contained in $\vec{\boldsymbol{w}}$. We then define vectors $\vec{\boldsymbol{\rho}}_{i,k}$ for $i \in [r+1, r+r_w]$, whose coefficients are appropriately assigned to the corresponding new variables according to $\tilde{f}_k$.

Lemma 19. *Π_1 is partial zero-knowledge for the (target) extended relation $\tilde{\mathsf{R}}_2$. Moreover, it satisfies* 1-pZK.

Proof. We need to describe a simulator for the associated $\tilde{\Pi}_1 : \hat{\mathsf{R}} \to \tilde{\mathsf{R}}_2$ where the prover sends, as part of the last message, the witness material $\vec{\boldsymbol{v}}, \vec{\boldsymbol{t}}_v$. We describe below a simulation algorithm that takes as input public parameters pp, and instance $\mathbb{x}$ containing commitment c.

$\mathcal{S}(\mathsf{pp}; \mathbb{x})$:

- Sample $\mathsf{c}_w \xleftarrow{\$} \mathcal{R}_q^{\kappa_1}$, $\vec{\boldsymbol{v}} \leftarrow D_\sigma^{nrd}$,
- Compute $\mathsf{c}_v, \vec{\boldsymbol{t}}_v \xleftarrow{\$} \mathsf{DCom}^*(\mathsf{ck}, \vec{\boldsymbol{v}})$
- Sample $\boldsymbol{\alpha} \xleftarrow{\$} \mathcal{C}$, compute $\mathsf{c}_z \leftarrow \mathsf{c}_v - \boldsymbol{\alpha}\mathsf{c}$
- Reprogram $\mathsf{RO}(\mathsf{pp}, \mathbb{x}; \mathsf{c}_w, \mathsf{c}_z) := \boldsymbol{\alpha}$
- Output $\pi := (\mathsf{c}_w, \mathsf{c}_z; \boldsymbol{\alpha}; \vec{\boldsymbol{v}}, \vec{\boldsymbol{t}}_v)$.

We analyse the distribution of each simulated proof element:

- c_w: By the way we sample $\vec{\boldsymbol{r}}$, and by the choice of the length $n_r = (\kappa_1 \log q + (4\lambda/d))/\log(2\eta + 1)$ and infinity norm η for the random vector $\vec{\boldsymbol{r}}$ we can apply hiding Ajtai commitment Lemma 32, c_w is statistically close to uniform distribution on $\mathcal{R}_q^{\kappa_1}$.
- $\vec{\boldsymbol{v}}$: By Lemma 5, when prover does not abort, $\vec{\boldsymbol{v}}$ has distribution within statistical distance of $2^{-\omega(\log n)}$ of discrete Gaussian.
- $\vec{\boldsymbol{t}}_v$: First notice that both $\vec{\boldsymbol{t}}_v$ and $\boldsymbol{\alpha}\vec{\boldsymbol{t}} + \vec{\boldsymbol{t}}_z'$ lie in the co-set $\Lambda + \vec{\boldsymbol{t}}_v'$ where $\vec{\boldsymbol{t}}_v' = \mathsf{Split}(\mathsf{Com}(\vec{\boldsymbol{v}}))$. By Lemma 5, when prover does not abort, the random variable $\vec{\boldsymbol{t}}_v$, as sampled by the prover, is distributed statistically close to a lattice Gaussian over $D_{\vec{\boldsymbol{t}}_v' + \Lambda, \vec{\boldsymbol{t}}_v', \sigma'}$, which follows the same distribution as computing the auxiliary information of an hiding double commitment of $\vec{\boldsymbol{v}}$.
- c_z: By construction, $\mathsf{Open}(\mathsf{c}_v, \vec{\boldsymbol{v}}, \vec{\boldsymbol{t}}_v) = 1$, fixing a uniformly random challenge $\boldsymbol{\alpha}$, there exists a unique c_z satisfying $\mathsf{c}_v = \mathsf{c}_z + \boldsymbol{\alpha}\mathsf{c}$.

□

Lemma 20. *$\tilde{\Pi}_1$ satisfies* 1-HtoInv *assuming hardness of MSIS for rank κ_1 and norm $\beta' + \eta'\sqrt{2n_w d} + 1$ where $\eta' := \lceil 2^{\frac{\kappa_1 \log q + 2\lambda/d}{n_w} - 1} \rceil$.*

Remark 2. For fixed κ_1, q, λ, and d, we have $\eta' \leq 2^{\frac{\kappa_1 \log q + 2\lambda/d}{n_w} - 1} + 1$, and hence $\eta'\sqrt{2n_w d} = O(\sqrt{n_w})$ as n_w grows.

Proof. Let $\mathcal{A}^{\mathsf{O_{RO}}, \mathsf{O_S}}$ be an adversary against 1-HtoInv, we emulate an adversary $\mathcal{B}$, solving any inhomogeneous Module-SIS instance $(\boldsymbol{B}, \mathsf{c_{sis}}, \gamma)$ (where $\boldsymbol{B} = (\boldsymbol{B}_{i,j})_{i,j}$ uniformly random from the space of commitment keys ck_B, $\mathsf{c_{sis}} \xleftarrow{\$} \mathcal{R}_q^{\kappa_1}$, $\gamma = \beta' + \eta'\sqrt{2n_w d}$) with the same probability as $\mathcal{A}$ winning.

$\mathcal{B}(\boldsymbol{B}, \mathsf{c_{sis}}, \gamma)$

- Initialise empty table $\mathcal{T}$.
- Sample commitment key ck where we set $\mathsf{ck}_B := (\boldsymbol{B}_{ij})_{i,j} = \boldsymbol{B}$.
- Run $\mathcal{A}^{\mathsf{O_{RO}}, \mathsf{O_S}}(\mathsf{ck})$ and simulate the random oracle.
- Upon simulation query of form $\mathbb{x} = (\mathsf{c}, \beta, \gamma, \mathcal{F}, \emptyset)$, run the following simulator $\mathcal{S}'$ (we underline the differences with $\mathcal{S}$):
  - $\mathsf{c}_w \leftarrow \mathsf{c_{sis}} + \mathsf{c}_e$ where $\mathsf{c}_e \leftarrow \mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{e}})$ and $\vec{\boldsymbol{e}} \xleftarrow{\$} U_{\eta'}^{n_w}$ where we recall $U_{\eta'}^{n_w}$ is the uniform distribution over the hypercube of n_w-dimensional vectors of ring elements with infinity-norm at most η'.
  - Sample $\vec{\boldsymbol{v}} \leftarrow D_\sigma^{nrd}$,
  - Compute $\mathsf{c}_v, \vec{\boldsymbol{t}}_v \xleftarrow{\$} \mathsf{Com}^*(\mathsf{ck}, \vec{\boldsymbol{v}})$
  - Sample $\boldsymbol{\alpha} \xleftarrow{\$} \mathcal{C}$, compute $\mathsf{c}_z \leftarrow \mathsf{c}_v - \boldsymbol{\alpha}\mathsf{c}$
  - Reprogram $\mathsf{RO}(\mathsf{pp}, \mathbb{x}; \mathsf{c}_w, \mathsf{c}_z) := \boldsymbol{\alpha}$
  - Add the tuple $(\mathbb{x}, \mathsf{c}_w, \vec{\boldsymbol{e}})$ to $\mathcal{T}$.
  - Output $\pi := (\mathsf{c}_w, \mathsf{c}_z; \boldsymbol{\alpha}; \vec{\boldsymbol{v}}, \vec{\boldsymbol{t}}_v)$.
- Eventually $\mathcal{A}$ outputs a proof $(\mathsf{c}_w, \mathsf{c}_z; \boldsymbol{\alpha}; \vec{\boldsymbol{v}}, \vec{\boldsymbol{t}}_v, \vec{\boldsymbol{w}})$ for some $\mathbb{x}$ queried to the simulation oracle. Let $\vec{\boldsymbol{e}}$ such that $(\mathbb{x}, \mathsf{c}_w, \vec{\boldsymbol{e}}) \in \mathcal{T}$, the reduction $\mathcal{B}$ outputs $\vec{\boldsymbol{w}} - \vec{\boldsymbol{e}}$.

We notice that $\mathcal{B}$ provides $\mathcal{A}$ with identical view, this is because by the leftover hash lemma, Lemma 31,both c_w returned by the simulator $\mathcal{S}$ and the same value returned by $\mathcal{S}'$ are statistically close to uniform distribution. By construction, $\mathcal{A}$ winning implies

$$\mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{w}} - \vec{\boldsymbol{e}}) = \mathsf{c_{sis}} \wedge \|\vec{\boldsymbol{w}} - \vec{\boldsymbol{e}}\|_2 \leq \beta' + \eta'\sqrt{2n_w d}$$

Thus, the equation above implies $\mathcal{B}$ finding a short non-zero solution $(\vec{\boldsymbol{w}} - \vec{\boldsymbol{e}}, 1)$ for MSIS instance $((\boldsymbol{B}, -\mathsf{c_{sis}}), \gamma + 1)$. □

Lemma 21 (Special Soundness of Π_1). *Assume Module SIS hard for rank κ and norm $6T\beta'$, rank κ_1 and norm $\max\{6T\gamma_v, 2\beta'\}$, then $\Pi_1 : \hat{\mathsf{R}}_{\mathsf{relaxed}} \to \hat{\mathsf{R}}$ is computational 3-special sound.*

Proof. We consider three accepting transcripts and let $\boldsymbol{\alpha}^{(i)}$ be the challenge of the i-th transcript for $i \in [3]$. The transcript $\mathtt{trns}^{(i)}$ is of the form $(\mathsf{c}_w, \mathsf{c}_z, \boldsymbol{\alpha}^{(i)}, \vec{\boldsymbol{v}}^{(i)}, \vec{\boldsymbol{w}}^{(i)}, \vec{\boldsymbol{t}}^{(i)})$. Clearly, $\vec{\boldsymbol{w}} := \vec{\boldsymbol{w}}^{(1)} = \vec{\boldsymbol{w}}^{(2)} = \vec{\boldsymbol{w}}^{(3)}$ because they are all valid openings for c_w and as otherwise we would break the binding of Com. Thus this must hold with overwhelming probability. As we have done in Lemma 18, we can define the values $\boldsymbol{s}'_k, \boldsymbol{z}'_k, \boldsymbol{z}''_k$ as in Eq. (11).

Interpolating the points $(\boldsymbol{\alpha}^{(1)}, \vec{\boldsymbol{t}}^{(1)})$ and $(\boldsymbol{\alpha}^{(2)}, \vec{\boldsymbol{t}}^{(2)})$ we can define two vectors $\vec{\boldsymbol{t}}^{\mathcal{E}}_s, \vec{\boldsymbol{t}}^{\mathcal{E}}_z$ such that for $i = 1, 2$:

$$\boldsymbol{\alpha}^{(i)} \vec{\boldsymbol{t}}^{\mathcal{E}}_s + \vec{\boldsymbol{t}}^{\mathcal{E}}_z = \vec{\boldsymbol{t}}^{(i)}$$

Moreover, $\bar{\boldsymbol{\alpha}} := (\boldsymbol{\alpha}^{(1)} - \boldsymbol{\alpha}^{(2)})$, $\vec{\boldsymbol{t}}^{\mathcal{E}}_s$ is a weak opening for c. Namely, we have that $\mathsf{c} = \mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{t}}^{\mathcal{E}}_s)$ and $\|\bar{\boldsymbol{\alpha}} \vec{\boldsymbol{t}}^{\mathcal{E}}_s\|_2 \leq 2\gamma_v$. And $(\bar{\boldsymbol{\alpha}}, \vec{\boldsymbol{t}}^{\mathcal{E}}_z)$ is a weak opening of c_z such that $\|\bar{\boldsymbol{\alpha}} \vec{\boldsymbol{t}}^{\mathcal{E}}_s\|_2 \leq 2T\gamma_v$.

Thus $\vec{\boldsymbol{t}}^{(i)} = \boldsymbol{\alpha}^{(i)} \vec{\boldsymbol{t}}^{\mathcal{E}}_s + \vec{\boldsymbol{t}}^{\mathcal{E}}_z$ for all $i \in [3]$, as otherwise $\bar{\boldsymbol{\alpha}}(\vec{\boldsymbol{t}}^{(3)} - (\boldsymbol{\alpha}^{(3)} \vec{\boldsymbol{t}}^{\mathcal{E}}_s + \vec{\boldsymbol{t}}^{\mathcal{E}}_z))$ forms an non-zero MSIS solution of rank κ_1 and norm $6T\gamma_v$.

Let $\vec{B} = \vec{I} \otimes (b^i)_i$ where $\vec{I}$ has r rows, with the same argument we just used before, we can define $\vec{\boldsymbol{s}}^{\mathcal{E}}$ and $\vec{\boldsymbol{z}}^{\mathcal{E}}$, using interpolation on $(\boldsymbol{\alpha}^{(1)}, \vec{\boldsymbol{v}}^{(1)})$ and $(\boldsymbol{\alpha}^{(2)}, \vec{\boldsymbol{v}}^{(2)})$. They are weak opening for $\vec{B} \cdot \vec{\boldsymbol{t}}^{\mathcal{E}}_s$ and $\vec{B} \cdot \vec{\boldsymbol{t}}^{\mathcal{E}}_z$ respectively such that for all $j \in [r]$, $\|\bar{\boldsymbol{\alpha}} \vec{\boldsymbol{s}}^{\mathcal{E}}_j\|_2 \leq 2\beta'$ and $\|\bar{\boldsymbol{\alpha}} \vec{\boldsymbol{z}}^{\mathcal{E}}_j\|_2 \leq 2T\beta'$. By hardness of MSIS for rank κ and norm $6T\beta'$ we have that for $i = 1, 2, 3$:

$$\vec{\boldsymbol{v}}^{(i)} = \boldsymbol{\alpha}^{(i)} \vec{\boldsymbol{s}}^{\mathcal{E}} + \vec{\boldsymbol{z}}^{\mathcal{E}}$$

Thus from the verification equation on $\tilde{f}_k$ for any $k \in [K]$, we have that the following polynomial $P_k(X)$ has three roots:

$$\begin{aligned} P_k(X) = X^2(\textstyle\sum_{i,j} \boldsymbol{a}_{kij} \langle \vec{\boldsymbol{s}}^{\mathcal{E}}_i, \vec{\boldsymbol{s}}^{\mathcal{E}}_j \rangle - \boldsymbol{s}'_k) \\ + X(\textstyle\sum_{i,j} \boldsymbol{a}_{kij} (\langle \vec{\boldsymbol{s}}^{\mathcal{E}}_i, \vec{\boldsymbol{z}}^{\mathcal{E}}_j \rangle + \langle \vec{\boldsymbol{z}}^{\mathcal{E}}_i, \vec{\boldsymbol{s}}^{\mathcal{E}}_j \rangle) - \boldsymbol{e}_k) \\ + (\textstyle\sum_{i,j} \boldsymbol{a}_{kij} \langle \vec{\boldsymbol{z}}^{\mathcal{E}}_i, \vec{\boldsymbol{z}}^{\mathcal{E}}_j \rangle - \boldsymbol{z}'_k). \end{aligned}$$

See Lemma 18 (completeness of Π_1) for more details on how we derived the polynomial. Here the variables $\boldsymbol{s}'_k, \boldsymbol{e}_k$ and $\vec{\boldsymbol{z}}'_k$ are those defined from $\vec{u}$ output by the adversary. Moreover, the roots are different in each of the CRT slots (since these roots are challenge elements and differences of distinct challenge elements are invertible in $\mathcal{R}_q$), thus the polynomial $P_k(X)$ is quadratic and has three different roots in each of the CRT slots which means that the polynomial P_k is the zero polynomial. This implies that $\boldsymbol{s}'_k = \sum_{i,j} \boldsymbol{a}_{kij} \langle \vec{\boldsymbol{s}}_i, \vec{\boldsymbol{s}}_j \rangle$.

With the same argument we can show that the polynomial $P_{K+k}(X)$ for $k \in [K]$, described below, are linear but have three different roots (over all the CRT slots) thus they are zero polynomials.

$$X \left(\sum_i \langle \vec{\boldsymbol{\rho}}_{ki}, \vec{\boldsymbol{s}}_i \rangle + \boldsymbol{s}'_k + \boldsymbol{b}_k \right) + \sum_i \langle \vec{\boldsymbol{\rho}}_{ki}, \vec{\boldsymbol{z}}_i \rangle - \boldsymbol{z}''_k$$

Which implies that $\sum_i \langle \vec{\boldsymbol{\rho}}_{ki}, \vec{\boldsymbol{s}}_i \rangle + \sum_{i,j} \boldsymbol{a}_{kij} \langle \vec{\boldsymbol{s}}_i, \vec{\boldsymbol{s}}_j \rangle + \boldsymbol{b}_k = 0$ for any $k \in [K]$. For opening of the starting commitment, we note that $\mathsf{Open}(\mathsf{ck}, \mathsf{c}, \vec{\boldsymbol{s}}^{\mathcal{E}}, \vec{\boldsymbol{t}}^{\mathcal{E}}_s) = 1$, $\|\bar{\boldsymbol{\alpha}} \vec{\boldsymbol{t}}^{\mathcal{E}}_s\|_2 \leq 2\gamma_v$ and for all $j \in [r]$, $\|\bar{\boldsymbol{\alpha}} \vec{\boldsymbol{s}}^{\mathcal{E}}_j\|_2 \leq 2\beta'$. □

Lemma* 22. *Π_{nlin} satisfies $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ w.r.t. the target relation $\tilde{\mathsf{R}}_2$.*

Proof. The previous two lemmas, together with Theorem 1, imply that $(\Pi_1)_{\mathsf{FS}}$ satisfies KS, 1-pZK for the extended (target) relation $\tilde{\mathsf{R}}_2$ and 1-HtoInv, thus by Theorem 4 it satisfies $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ w.r.t. the target relation $\tilde{\mathsf{R}}_2$.

Moreover, $\Pi_{\mathsf{zk\text{-}agg}'} \circ \Pi_{\mathsf{zk\text{-}norm}}$ is $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ thus the composition Π_{nlin} is $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ by Theorem 5. □

7.6 The linear-term RoK for the Committed Principal Relation

We can refine the target relation of the RoK Π_{nlin}. In fact, we observe that: (1) the variables are divided in two sets of indexes $I_v = [1, r]$ and $I_w = [r+1, r+r_w]$ for some $r_w \in \mathbb{N}$, (2) the variables whose indexes are I_u appear only in linear functions . Simplifying, we can parse the target $\mathbb{x}, \mathbb{w}$ of Π_{nlin} as:

$$\mathbb{x} = ((\mathsf{c}_v, [r], \gamma_v), \mathcal{F}, \mathcal{F}', \vec{\beta}), \mathbb{w} = \left((\vec{\boldsymbol{s}}_j)_j, \vec{\boldsymbol{t}} \right) = (\vec{\boldsymbol{v}}, \vec{\boldsymbol{w}}), \vec{\boldsymbol{t}}_v$$

Linear-term ZK-RoK Π_2 from R_2 to $\hat{\mathsf{R}}$

Prover		**Verifier**
Parse $\mathbb{w} = (\vec{\boldsymbol{v}}, \vec{\boldsymbol{w}}), \vec{\boldsymbol{t}}_v$		Parse $\mathbb{x} = (\mathsf{c}_v, f, (\beta, \gamma))$
$\vec{\boldsymbol{r}} \xleftarrow{\$} U_\eta^{n_r}, \mathsf{c}_w \leftarrow \mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{w}} \Vert \vec{\boldsymbol{r}})$,		
$\vec{\boldsymbol{z}} \xleftarrow{\$} D_\sigma^{(n_w+n_r)\cdot d}, \bar{\boldsymbol{z}} \leftarrow l(\vec{\boldsymbol{z}}), \mathsf{c}_z \leftarrow \mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{z}})$	$\xrightarrow{\mathsf{c}_w, \mathsf{c}_z, \bar{\boldsymbol{z}}}$	$\boldsymbol{\alpha} \xleftarrow{\$} \mathcal{C}$
$\vec{\boldsymbol{u}} \leftarrow \vec{\boldsymbol{z}} + \boldsymbol{\alpha} \cdot (\vec{\boldsymbol{w}} \Vert \vec{\boldsymbol{r}})$	$\xleftarrow{\boldsymbol{\alpha}}$	
Abort if $\mathsf{Rej}(\vec{\boldsymbol{u}}, \boldsymbol{r}\vec{\boldsymbol{w}}, \sigma, \rho) = 1$		
return $(\vec{\boldsymbol{v}}, \vec{\boldsymbol{u}}), \vec{\boldsymbol{t}}_v$		**return** $(\mathsf{c}_v, \tilde{\mathcal{F}}, \beta')$

Fig. 10: The RoK for linear terms. This RoK masks the variables $\vec{\boldsymbol{w}}$. We refer the reader to the main text for the description of the target instance.

We can define relation $\mathsf{R}_2 \subset \hat{\mathsf{R}}$ which formalize the target instances of Π_{nlin}.

$$(\mathbb{x}, \mathbb{w}) \in \mathsf{R}_2 \iff \begin{cases} (\mathbb{x}, \mathbb{w}) \in \hat{\mathsf{R}}, \\ \forall k : \exists q_k, l_k, \boldsymbol{b}_k : f_k(\vec{\boldsymbol{v}}, \vec{\boldsymbol{w}}) = q_k(\vec{\boldsymbol{v}}) + l_k(\vec{\boldsymbol{w}}) + \boldsymbol{b}_k, \\ \mathsf{deg}(l_k) \leq 1, \end{cases} \tag{12}$$

Namely, the variables in $\vec{\boldsymbol{v}}$ might appear in quadratic and linear terms in q_k, but the variables in $\vec{\boldsymbol{w}}$ appear only in linear terms in l_k. It is easy to see that Π_{nlin} is indeed a RoK from $\hat{\mathsf{R}}$ to R_2.

Similarly, we refine the extended relation $\tilde{\mathsf{R}}_2$ from Eq. (7) by introducing a more granular version, in which the left-hand side of Eq. (7) requires $(\mathbb{x}, \mathbb{w}) \in \mathsf{R}_2$. It is easy to verify that $\tilde{\Pi}_{\mathsf{nlin}}$ is indeed a RoK from $\hat{\mathsf{R}}$ to $\tilde{\mathsf{R}}_2$.

We present the RoK Π_{lin}, which is obtained as the composition $(\Pi_2 \circ \Pi_{\mathsf{agg}} \circ \Pi_{\mathsf{zk\text{-}agg}'} \circ \Pi_{\mathsf{zk\text{-}norm}})$, where Π_2 is the RoK described in Fig. 10.

The RoK Π_2 is a standard Σ-protocol for linear relations, described as a RoK from R_2 to $\hat{\mathsf{R}}$ in Fig. 10. As in the case of Π_1 from previous section, we can only establish relaxed knowledge soundness for Π_2. Accordingly, in the security analysis we interpret Π_2 as a RoK from $(\mathsf{R}_2)_{\mathsf{relaxed}}$ to $\hat{\mathsf{R}}$, where, similarly to Section 7.5, we define $(\mathsf{R}_2)_{\mathsf{relaxed}}$ to coincide with R_2 except that, on the right-hand side of Eq. (12), we only require $(\mathbb{x}, \mathbb{w}) \in \hat{\mathsf{R}}_{\mathsf{relaxed}}$. Also, we can define $(\tilde{\mathsf{R}}_2)_{\mathsf{relaxed}}$ by relaxing the norm condition for the witness in the extended relation $\tilde{\mathsf{R}}_2$.

In Lemma 27 we show that $(\Pi_{\mathsf{agg}} \circ \Pi_{\mathsf{zk\text{-}agg}'} \circ \Pi_{\mathsf{zk\text{-}norm}})$ can be interpreted as a RoK from R_2 to $(\mathsf{R}_2)_{\mathsf{relaxed}}$, and, in particular, due to the aggregation performed by Π_{agg} and $\Pi_{\mathsf{zk\text{-}agg}'}$, the target instance of this RoK contains a single dot-product function $f(\vec{\boldsymbol{v}}, \vec{\boldsymbol{w}})$. Accordingly, Π_2 operates on a single dot-product function f.

The dot-product functions $\tilde{\mathcal{F}} = (\tilde{f}_k)_{k \in [\kappa_1 + 1]}$ in the target instance are defined as follows. The first κ_1 functions are linear and jointly enforce the relation $\mathsf{c}_u = \mathsf{Com}(\mathsf{ck}_{B,I_u}, \vec{\boldsymbol{u}})$, where $\mathsf{c}_u := \mathsf{c}_z + \boldsymbol{\alpha}\mathsf{c}_w$, corresponding to the verifier's check in the underlying Σ-protocol for linear relation. The final function is defined as

$$\tilde{f}_{\kappa_1+1}(\vec{\boldsymbol{v}}, \vec{\boldsymbol{u}}) := \boldsymbol{\alpha} \cdot q(\vec{\boldsymbol{v}}) + l(\vec{\boldsymbol{u}}) + \boldsymbol{b} - \bar{\boldsymbol{z}}.$$

The norm bound β' is set to $\sqrt{\beta^2 + 2dn_w\sigma^2 + n_r\eta^2 + \gamma^2}$ Notice that we could obtain a tighter norm bound by having $\Pi_{\mathsf{zk\text{-}norm}}$ apply the JL projection separately to $\vec{\boldsymbol{v}}$, $\vec{\boldsymbol{w}}$, and $\vec{\boldsymbol{t}}$. We do not pursue this optimization for readability.

Similar to the previous RoK Π_1, the size n_r of the random vector $\vec{\boldsymbol{r}}$ is set to $\frac{\kappa_1 \log q + 2\lambda/d}{\log(2\eta+1)}$ so that Lemma 32 applies.

Theorem⋆ 11. *The RoK Π_{lin} is complete, knowledge sound, zero-knowledge and* SIM-EXT. *Concretely, let $\sigma, \sigma_1, \sigma_2 \in \mathbb{R}$ be such that $\sigma = \omega(T\beta\sqrt{\log n_w})$, $\sigma_1 = \omega(\beta\sqrt{\log \lambda_1})$ and $\sigma_2 = \omega(\gamma\sqrt{\log \lambda_2})$, there exist constants $\rho, \rho_1, \rho_2, C_1, C_2 > 0$ such that the RoK Π_{lin} is complete. Assume Module SIS hardness for rank κ and norm $2\beta'$, rank κ_1 and norm $2\beta'$, Π_{lin} has knowledge-soundness error $2^{-\lambda_1} + 2^{-\lambda_2} + 1/q^{\lambda'} + 2/|\mathcal{C}|$,*

the knowledge extractor outputs a witness $\vec{\boldsymbol{s}}$ *with norm bounded by* $\sigma_1\sqrt{4\lambda_1/D_2}$ *and a witness* $\vec{\boldsymbol{t}}$ *with norm bounded by* $\sigma_2\sqrt{4\lambda_2/D_2}$.

Proof (sketch). The argument is very similar to Theorem 10.

Lemma 23. *Let* $\sigma = \omega(T\beta\sqrt{\log n_w})$. *There exists constant* $\rho > 0$ *such that the RoK* Π_2 *is complete.*

Proof. Target instance satisfiability is by design. It suffices to analyse the aborting probability for honest P and V. By rejection sampling, Lemma 5, there exists constant ρ such that the aborting probability of P is within $2^{-\omega(\log n_w)}$ of $1/\rho$. Moreover, by the tail bound, Lemma 4, $\vec{\boldsymbol{u}}$ has norm bound $\beta_u = \sigma\sqrt{2n_w d}$ with overwhelming probability. Hence $\|\vec{\boldsymbol{v}}\|_2^2 + \|\vec{\boldsymbol{u}}\|_2^2 + \|\vec{\boldsymbol{t}}_v\|_2^2 \leq \beta^2 + 2dn_w\sigma^2 + \gamma^2$ holds with at least the same probability.

Lemma 24 (Special Soundness of Π_2). *Π_2 is computational 2-special sound assuming hardness of MSIS for rank κ and norm $2\beta'$, rank κ_1 and norm $2\beta'$.*

Proof. For $i \in \{1, 2\}$, the transcript $\mathtt{trns}^{(i)}$ is of the form $(\mathsf{c}_w, \mathsf{c}_z, \bar{\boldsymbol{z}}; \boldsymbol{r}^{(i)}; \vec{\boldsymbol{v}}, \vec{\boldsymbol{u}}^{(i)}, \vec{\boldsymbol{t}_v})$. The vectors $\vec{\boldsymbol{v}}, \vec{\boldsymbol{t}_v}$ must be the same in both transcripts under the assumption of MSIS for rank κ and norm $2\beta'$, rank κ_1 and norm $2\beta'$. We interpolate the two points $(\boldsymbol{r}^{(1)}, \vec{\boldsymbol{u}}^{(1)}), (\boldsymbol{r}^{(2)}, \vec{\boldsymbol{u}}^{(2)})$ and define two vectors $\vec{\boldsymbol{w}}^{\mathcal{E}}, \vec{\boldsymbol{z}}^{\mathcal{E}}$ such that for $i = 1, 2$:

$$\boldsymbol{\alpha}^{(i)}\vec{\boldsymbol{w}}^{\mathcal{E}} + \vec{\boldsymbol{z}}^{\mathcal{E}} = \vec{\boldsymbol{u}}^{(i)} \tag{13}$$

Since these two transcripts are accepting, $\mathsf{Com}(\mathsf{ck}, \vec{\boldsymbol{u}}^{(i)}) = \mathsf{c}_u$ for all i. Sub in the interpolation above we get $\mathsf{Com}(\mathsf{ck}, \vec{\boldsymbol{w}}^{\mathcal{E}}) = \mathsf{c}_w$.
From the verification equation, using Eq. (13), we obtain a linear polynomial $P(X)$ on $\mathcal{R}_q$ with two distinct roots where:

$$P(X) := (q(\vec{\boldsymbol{v}}) + l(\vec{\boldsymbol{w}}^{\mathcal{E}}) + \boldsymbol{b})X + (l(\vec{\boldsymbol{z}}^{\mathcal{E}}) - \bar{\boldsymbol{z}}).$$

Using the same argument as in Lemma 21 based on invertibility of pairwise differences of challenge elements, we have $q(\vec{\boldsymbol{v}}) + l(\vec{\boldsymbol{w}}^{\mathcal{E}}) + \boldsymbol{b} = \boldsymbol{0}$. □

Lemma 25. *Let* $\tilde{\Pi}_2 : (\tilde{\mathsf{R}}_2)_{\mathsf{relaxed}} \to \hat{\mathsf{R}}$ *be the RoK associated to the extended source relation* $\tilde{\mathsf{R}}_2$, $\tilde{\Pi}_2$ *satisfies zero-knowledge and in particular* 1-ZK.

Proof. We describe below a simulation algorithm that takes as input an instance $\mathbb{x}$ and additional input a partial witness $(\vec{\boldsymbol{v}}, \vec{\boldsymbol{t}_v})$:

$\underline{\mathcal{S}(\mathsf{pp}; \mathbb{x}, \vec{\boldsymbol{v}})}$:

- Sample $\boldsymbol{\alpha} \xleftarrow{\$} \mathcal{C}$, $\vec{\boldsymbol{u}} \xleftarrow{\$} D_\sigma^{(n_w+n_r)\cdot d}$,
- Sample $\mathsf{c}_u \xleftarrow{\$} \mathcal{R}_q^{\kappa_1}$,
- Compute $\mathsf{c}_z \leftarrow \mathsf{c}_u - \boldsymbol{\alpha}\mathsf{c}_w$,
- Set $\bar{\boldsymbol{z}} \leftarrow l(\vec{\boldsymbol{u}}) - \boldsymbol{\alpha}(q(\vec{\boldsymbol{v}}) + \boldsymbol{b})$,
- Reprogram $\mathsf{RO}(\mathsf{pp}, \mathbb{x}; \mathsf{c}_u, \mathsf{c}_z, \bar{\boldsymbol{z}}) := \boldsymbol{\alpha}$
- Output $\pi = (\mathsf{pp}, \mathbb{x}; \mathsf{c}_u, \mathsf{c}_z, \bar{\boldsymbol{z}}; \vec{\boldsymbol{\alpha}}; \vec{\boldsymbol{v}}, \vec{\boldsymbol{t}_v}, \vec{\boldsymbol{u}})$

Correctness is by design, it suffices to inspect the distribution of each proof element.

- $\vec{\boldsymbol{u}}$, thanks to rejection sampling Lemma 5, the prover samples $\vec{\boldsymbol{u}}$ from a distribution that is statistically close to $D_\sigma^{(n_w+n_r)\cdot d}$. While the simulator samples directly from the latter distribution. Thus the two random variables are indistinguishable.
- For c_u, by hiding of the commitment scheme and our choice of $\vec{\boldsymbol{r}}$, Lemma 32, is statistically close to those computed by the prover.
- c_z and $\bar{\boldsymbol{z}}$ are uniquely defined once the other random variables are set.

Lemma 26. *$\tilde{\Pi}_2$ satisfies* 1-UR *assuming hardness of MSIS for rank κ and norm $2\beta'$.*

Proof (informal). For fixed partial transcript $\mathbb{x}; \mathsf{c}_w, \mathsf{c}_z, \bar{\boldsymbol{z}}; \boldsymbol{\alpha}$, if an adversary can output two distinct valid $\vec{\boldsymbol{u}}^{(1)}$ and $\vec{\boldsymbol{u}}^{(2)}$ of norm bound β_u, both open $\mathsf{c}_u := \mathsf{c}_z + \boldsymbol{\alpha} \cdot \mathsf{c}_w$, then clearly it is breaking MSIS for rank κ and norm $2\beta'$. □

Lemma 27. *Π_{lin} is a complete RoK from R_2 to $\hat{\mathsf{R}}$.*

Proof. Since $\Pi_{\mathsf{zk\text{-}norm}}$ is a RoK from $\hat{\mathsf{R}}$, we can restrict its source instance-witness pairs to $\mathsf{R}_2 \subset \hat{\mathsf{R}}$. Under this restriction, it follows from inspection that $\Pi_{\mathsf{zk\text{-}norm}}$ is in fact a RoK from R_2 to $(\mathsf{R}_2)_{\mathsf{relaxed}}$: it only introduces additional witness variables $\vec{\boldsymbol{y}_1}, \vec{\boldsymbol{y}_2}, \vec{\boldsymbol{r}}$ and new linear functions in $\mathcal{F}'$, and these variables appear exclusively in linear terms.

Similarly, $\Pi_{\mathsf{zk\text{-}agg}'}$ is a RoK from $(\mathsf{R}_2)_{\mathsf{relaxed}}$ to $(\mathsf{R}_2)_{\mathsf{relaxed}}$. It introduces additional witness variables $\vec{\boldsymbol{g}}$ and new linear functions in $\mathcal{F}$, and again all new variables appear only in linear terms.

The protocol $\Pi_{\mathsf{agg}'}$ is also a RoK from $(\mathsf{R}_2)_{\mathsf{relaxed}}$ to $(\mathsf{R}_2)_{\mathsf{relaxed}}$. It does not introduce new variables, and it defines a single function f as a linear combination of the functions in the source instance, thereby preserving the structural constraints of R_2.

Finally, Π_2 is a RoK from a subset of $(\mathsf{R}_2)_{\mathsf{relaxed}}$ whose instances have only one dot-product function. The source relation for Π_2 matches the target relation of $\Pi_{\mathsf{agg}'}$.

Thus, composing these four RoKs yields the desired result.

Lemma* 28. *$\tilde{\Pi}_{\mathsf{lin}}$ satisfies* SIM-EXT *with respect to source relation $\tilde{\mathsf{R}}_2$ and target relation $\hat{\mathsf{R}}$.*

Proof. Lemmas 24 to 26 imply that $\tilde{\Pi}_2$ satisfies KS, 1-ZK and 1-UR , thus by Theorem 3, $\tilde{\Pi}_2$ satisfies SIM-EXT.

Moreover, by Lemmas 15 and 16 $\Pi' := (\Pi_{\mathsf{agg}} \circ \Pi_{\mathsf{zk\text{-}agg}'} \circ \Pi_{\mathsf{zk\text{-}norm}})$ is pZK with source relation $\hat{\mathsf{R}}$ and thus also with source relation $\tilde{\mathsf{R}}_2$ and is $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ thanks to Lemma 16 and Corollary 1. Thus $\tilde{\Pi}_{\mathsf{nlin}}$ is ZK with source relation $\tilde{\mathsf{R}}_2$ and by Theorem 5 is SIM-EXT with source relation $\tilde{\mathsf{R}}_2$.

□

7.7 Putting All Together

We are ready to state our main theorem on the simulation extractability of zero-knowledge LaBRADot.

Theorem* 12. *Define* one-iteration *zk-LaBRADot as the proof of knowledge $\Pi^{\mathsf{g}} \circ \Pi_{\mathsf{lin}} \circ \Pi_{\mathsf{nlin}}$ with g set as in Theorem 7. Then* one-iteration *zk-LaBRADot is simulation-extractable assuming MSIS for rank κ and norm $\max\{6T\beta_{\mathsf{nlin}}(\beta), 2\beta_{\mathsf{lin}}(\beta_{\mathsf{nlin}}), \beta_{\mathsf{g}}(\beta_{\mathsf{lin}}) + \sqrt{r}T\tilde{\beta}, 4T\beta_{\mathsf{g}}(\beta_{\mathsf{lin}}), \beta_{\mathsf{g}}(\beta_{\mathsf{lin}}) + 4T\tilde{\beta}\}$, for rank κ_1 and norm $\max\{6T\gamma_v, 2\beta_{\mathsf{nlin}}(\beta), 2\beta_{\mathsf{lin}}(\beta_{\mathsf{nlin}}), 2\beta_{\mathsf{g}}(\beta_{\mathsf{lin}})\}$ where β_{g} denotes β' in Section 7.2, β_{nlin} denotes β' in Section 7.5, β_{lin} denotes β' in Section 7.6.*[11]

Proof. By Lemma 22, Π_{nlin} is $\mathsf{SIM\text{-}EXT}_{\mathsf{pZK}}$ with respect to the extended target relation $\tilde{\mathsf{R}}_2$. Moreover, by Lemma 28, Π_{lin} is SIM-EXT with respect to the extended source relation $\tilde{\mathsf{R}}_2$ Therefore, by our main composition theorem (Theorem 5), the composition $\Pi_{\mathsf{ZK}} := \Pi_{\mathsf{lin}} \circ \Pi_{\mathsf{nlin}}$ is SIM-EXT. Define polynomial-time recognizable relation $\mathsf{R}'((\mathbb{x}, \mathsf{aux}); \mathbb{w}) \iff (\mathbb{x}; \mathbb{w}) \in \hat{\mathsf{R}} \wedge f_{\mathsf{ZK}}(\mathsf{aux}) = \mathbb{x}$, where f_{ZK} denotes the public reducibility function of Π_{ZK}. Consider the two RoKs Π'_{ZK} and $(\Pi^{\mathsf{g}})'$, which are almost identical to Π_{ZK} and Π^{g}, respectively, except that the target relation of Π'_{ZK} and the source relation of $(\Pi^{\mathsf{g}})'$ are both R'. This can formally be obtained by slightly modifying the public-reducibility function for Π'_{ZK} to include the proof π as part of the target instance and by slightly modifying the verification procedure for $(\Pi^{\mathsf{g}})'$ to reuse the public-reducibility function. Notice that these modifications are syntactic adaptations that ensure R' is both efficiently recognizable and, as we show next, has witness uniqueness. Moreover, $\Pi^{\mathsf{g}} \circ \Pi_{\mathsf{ZK}}$ is SIM-EXT if and only if $(\Pi^{\mathsf{g}})' \circ \Pi'_{\mathsf{ZK}}$ is SIM-EXT, since a proof π for an instance $\mathbb{x}$ verifies in the former composition if and only if it verifies for the same instance in the latter composition.

We show that R' has witness uniqueness assuming MSIS for rank κ and norm $2\beta_{\mathsf{lin}}$, for rank κ_1 and norm $2\beta_{\mathsf{lin}}$. In fact, the target instances and witness are of the form $\mathbb{x} = (\mathsf{c}_v, \mathcal{F}, \beta_{\mathsf{lin}})$ and $\mathbb{w} = (\vec{\boldsymbol{v}}, \vec{\boldsymbol{u}}), \vec{\boldsymbol{t}_v}$, where $\vec{\boldsymbol{v}}$ is committed in c_v, thus finding two different witnesses that are different in the values $\vec{\boldsymbol{v}}, \vec{\boldsymbol{t}_v}$ breaks the binding of the commitment scheme. Moreover, the remaining witness $\vec{\boldsymbol{u}}$, by inspection of Fig. 10, is committed in a (single) Ajtai commitment value c_u which is set by the equations in $\mathcal{F}$. Thus, binding of commitment scheme implies uniqueness of $\vec{\boldsymbol{u}}$.

Finally, by Theorem 7, LaBRADot Π^{g} satisfies 0-UR with source relation R' since the relation has witness uniqueness , and so does $(\Pi^{\mathsf{g}})'$. Applying the main theorem once more, we conclude that $\Pi^{\mathsf{g}} = \Pi^{\mathsf{g}} \circ (\Pi_{\mathsf{lin}} \circ \Pi_{\mathsf{nlin}})$ is simulation extractable.

[11] $\beta_{\mathsf{nlin}}(\beta)$ denotes target norm bound function β' in Π_{nlin} evaluating at source norm bound β. Similarly for $\beta_{\mathsf{lin}}(\beta_{\mathsf{nlin}})$ and $\beta_{\mathsf{g}}(\beta_{\mathsf{lin}})$.

Acknowledgements

The first author was supported by the French National Research Agency (ANR) through the CTRL-M project (ANR-25-CE39-5707) and by France 2030 through the ANR project TRUSTINCloudS (ANR-23-PECL-0009). The second author was fully supported by the French National Research Agency (ANR) through the CTRL-M project (ANR-25-CE39-5707).

References

[1] M. A. Aardal et al. "Aggregating Falcon Signatures with LaBRADOR". In: *CRYPTO 2024, Part I*. Ed. by L. Reyzin and D. Stebila. Vol. 14920. LNCS. Springer, Cham, Aug. 2024, pp. 71–106. DOI: 10.1007/978-3-031-68376-3_3 (cit. on pp. 7, 8, 10, 12, 49).

[2] B. Abdolmaleki et al. *On the Simulation-Extractability of Proof-Carrying Data*. Cryptology ePrint Archive, Report 2025/2037. 2025. URL: https://eprint.iacr.org/2025/2037 (cit. on pp. 3, 5, 7).

[3] S. Agrawal et al. "Discrete Gaussian Leftover Hash Lemma over Infinite Domains". In: *ASIACRYPT 2013, Part I*. Ed. by K. Sako and P. Sarkar. Vol. 8269. LNCS. Springer, Berlin, Heidelberg, Dec. 2013, pp. 97–116. DOI: 10.1007/978-3-642-42033-7_6 (cit. on p. 7).

[4] A. Arun, S. T. V. Setty, and J. Thaler. "Jolt: SNARKs for Virtual Machines via Lookups". In: *EUROCRYPT 2024, Part VI*. Ed. by M. Joye and G. Leander. Vol. 14656. LNCS. Springer, Cham, May 2024, pp. 3–33. DOI: 10.1007/978-3-031-58751-1_1 (cit. on p. 3).

[5] T. Attema and R. Cramer. "Compressed Σ-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics". In: *CRYPTO 2020, Part III*. Ed. by D. Micciancio and T. Ristenpart. Vol. 12172. LNCS. Springer, Cham, Aug. 2020, pp. 513–543. DOI: 10.1007/978-3-030-56877-1_18 (cit. on pp. 3, 6).

[6] T. Attema, S. Fehr, and M. Klooß. "Fiat-Shamir Transformation of Multi-round Interactive Proofs". In: *TCC 2022, Part I*. Ed. by E. Kiltz and V. Vaikuntanathan. Vol. 13747. LNCS. Springer, Cham, Nov. 2022, pp. 113–142. DOI: 10.1007/978-3-031-22318-1_5 (cit. on p. 12).

[7] T. Attema, V. Lyubashevsky, and G. Seiler. "Practical Product Proofs for Lattice Commitments". In: *CRYPTO 2020, Part II*. Ed. by D. Micciancio and T. Ristenpart. Vol. 12171. LNCS. Springer, Cham, Aug. 2020, pp. 470–499. DOI: 10.1007/978-3-030-56880-1_17 (cit. on p. 10).

[8] E. Ben-Sasson et al. "Aurora: Transparent Succinct Arguments for R1CS". In: *EUROCRYPT 2019, Part I*. Ed. by Y. Ishai and V. Rijmen. Vol. 11476. LNCS. Springer, Cham, May 2019, pp. 103–128. DOI: 10.1007/978-3-030-17653-2_4 (cit. on p. 3).

[9] W. Beullens and G. Seiler. "LaBRADOR: Compact Proofs for R1CS from Module-SIS". In: *CRYPTO 2023, Part V*. Ed. by H. Handschuh and A. Lysyanskaya. Vol. 14085. LNCS. Springer, Cham, Aug. 2023, pp. 518–548. DOI: 10.1007/978-3-031-38554-4_17 (cit. on pp. 4, 7, 26, 28, 29, 31, 49).

[10] B. Biasioli et al. *A Toolkit for Succinct Lattice-Based Zero Knowledge Proofs*. Cryptology ePrint Archive, Paper 2026/1289. 2026. URL: https://eprint.iacr.org/2026/1289 (cit. on p. 7).

[11] N. Bitansky et al. "Recursive composition and bootstrapping for SNARKS and proof-carrying data". In: *45th ACM STOC*. Ed. by D. Boneh, T. Roughgarden, and J. Feigenbaum. ACM Press, June 2013, pp. 111–120. DOI: 10.1145/2488608.2488623 (cit. on p. 3).

[12] M. Blum, P. Feldman, and S. Micali. "Non-Interactive Zero-Knowledge and Its Applications (Extended Abstract)". In: *20th ACM STOC*. ACM Press, May 1988, pp. 103–112. DOI: 10.1145/62212.62222 (cit. on p. 3).

[13] D. Boneh and B. Chen. "LatticeFold: A Lattice-Based Folding Scheme and Its Applications to Succinct Proof Systems". In: *ASIACRYPT 2025, Part III*. Ed. by G. Hanaoka and B.-Y. Yang. Vol. 16247. LNCS. Springer, Singapore, Dec. 2025, pp. 330–362. DOI: 10.1007/978-981-95-5099-9_11 (cit. on pp. 3, 7).

[14] D. Boneh and B. Chen. "LatticeFold+: Faster, Simpler, Shorter Lattice-Based Folding for Succinct Proof Systems". In: *CRYPTO 2025, Part VII*. Ed. by Y. T. Kalai and S. F. Kamara. Vol. 16006. LNCS. Springer, Cham, Aug. 2025, pp. 327–361. DOI: 10.1007/978-3-032-01907-3_11 (cit. on pp. 3, 6, 7).

[15] K. Boudgoust and O. Lapiha. *Leftover Hash Lemma(s) Over Cyclotomic Rings*. Cryptology ePrint Archive, Report 2025/1080. 2025. URL: https://eprint.iacr.org/2025/1080 (cit. on pp. 7, 28, 51, 53).

[16] B. Bünz and B. Chen. "Protostar: Generic Efficient Accumulation/Folding for Special-Sound Protocols". In: *ASIACRYPT 2023, Part II*. Ed. by J. Guo and R. Steinfeld. Vol. 14439. LNCS. Springer, Singapore, Dec. 2023, pp. 77–110. DOI: 10.1007/978-981-99-8724-5_3 (cit. on pp. 3, 7).

[17] B. Bünz et al. "Bulletproofs: Short Proofs for Confidential Transactions and More". In: *2018 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2018, pp. 315–334. DOI: 10.1109/SP.2018.00020 (cit. on p. 3).

[18] B. Bünz et al. "Recursive Proof Composition from Accumulation Schemes". In: *TCC 2020, Part II*. Ed. by R. Pass and K. Pietrzak. Vol. 12551. LNCS. Springer, Cham, Nov. 2020, pp. 1–18. DOI: 10.1007/978-3-030-64378-2_1 (cit. on p. 7).

[19] M. Campanelli, A. Faonio, and L. Russo. "SNARKs for Virtual Machines Are Non-malleable". In: *EUROCRYPT 2025, Part IV*. Ed. by S. Fehr and P.-A. Fouque. Vol. 15604. LNCS. Springer, Cham, May 2025, pp. 153–183. DOI: 10.1007/978-3-031-91134-7_6 (cit. on pp. 3, 4).

[20] M. Campanelli, D. Fiore, and A. Querol. "LegoSNARK: Modular Design and Composition of Succinct Zero-Knowledge Proofs". In: *ACM CCS 2019*. Ed. by L. Cavallaro et al. ACM Press, Nov. 2019, pp. 2075–2092. DOI: 10.1145/3319535.3339820 (cit. on p. 3).

[21] M. Chase and A. Lysyanskaya. "On Signatures of Knowledge". In: *CRYPTO 2006*. Ed. by C. Dwork. Vol. 4117. LNCS. Springer, Berlin, Heidelberg, Aug. 2006, pp. 78–96. DOI: 10.1007/11818175_5 (cit. on p. 13).

[22] A. Chiesa et al. "Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS". In: *EUROCRYPT 2020, Part I*. Ed. by A. Canteaut and Y. Ishai. Vol. 12105. LNCS. Springer, Cham, May 2020, pp. 738–768. DOI: 10.1007/978-3-030-45721-1_26 (cit. on p. 3).

[23] R. Cramer, I. Damgård, and B. Schoenmakers. "Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols". In: *CRYPTO'94*. Ed. by Y. Desmedt. Vol. 839. LNCS. Springer, Berlin, Heidelberg, Aug. 1994, pp. 174–187. DOI: 10.1007/3-540-48658-5_19 (cit. on p. 3).

[24] Q. Dao and P. Grubbs. "Spartan and Bulletproofs are Simulation-Extractable (for Free!)" In: *EUROCRYPT 2023, Part II*. Ed. by C. Hazay and M. Stam. Vol. 14005. LNCS. Springer, Cham, Apr. 2023, pp. 531–562. DOI: 10.1007/978-3-031-30617-4_18 (cit. on pp. 3–7, 14, 17, 18).

[25] A. De Santis et al. "Robust Non-interactive Zero Knowledge". In: *CRYPTO 2001*. Ed. by J. Kilian. Vol. 2139. LNCS. Springer, Berlin, Heidelberg, Aug. 2001, pp. 566–598. DOI: 10.1007/3-540-44647-8_33 (cit. on p. 3).

[26] C. Decker and R. Wattenhofer. "Bitcoin Transaction Malleability and MtGox". In: *ESORICS 2014, Part II*. Ed. by M. Kutylowski and J. Vaidya. Vol. 8713. LNCS. Springer, Cham, Sept. 2014, pp. 313–326. DOI: 10.1007/978-3-319-11212-1_18 (cit. on p. 3).

[27] A. Faonio, D. Fiore, and L. Russo. "Real-World Universal zkSNARKs are Non-Malleable". In: *ACM CCS 2024*. Ed. by B. Luo et al. ACM Press, Oct. 2024, pp. 3138–3151. DOI: 10.1145/3658644.3690351 (cit. on p. 3).

[28] A. Faonio and L. Russo. "Sumcheck-Based zkSNARKs are Non-malleable". In: *Advances in Cryptology – CRYPTO 2026*. Springer Nature Switzerland, 2026, pp. 35–61. DOI: 10.1007/978-3-032-35424-2_2 (cit. on p. 3).

[29] A. Faonio et al. "Structure-Preserving and Re-randomizable RCCA-Secure Public Key Encryption and Its Applications". In: *ASIACRYPT 2019, Part III*. Ed. by S. D. Galbraith and S. Moriai. Vol. 11923. LNCS. Springer, Cham, Dec. 2019, pp. 159–190. DOI: 10.1007/978-3-030-34618-8_6 (cit. on p. 3).

[30] A. Faonio et al. "From Polynomial IOP and Commitments to Non-malleable zkSNARKs". In: *TCC 2023, Part III*. Ed. by G. N. Rothblum and H. Wee. Vol. 14371. LNCS. Springer, Cham, Nov. 2023, pp. 455–485. DOI: 10.1007/978-3-031-48621-0_16 (cit. on p. 3).

[31] S. Faust et al. "On the Non-malleability of the Fiat-Shamir Transform". In: *INDOCRYPT 2012*. Ed. by S. D. Galbraith and M. Nandi. Vol. 7668. LNCS. Springer, Berlin, Heidelberg, Dec. 2012, pp. 60–79. DOI: 10.1007/978-3-642-34931-7_5 (cit. on pp. 3, 5, 6, 17).

[32] A. Gabizon, Z. J. Williamson, and O. 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 (cit. on p. 3).

[33] C. Ganesh et al. "Fiat-Shamir Bulletproofs are Non-Malleable (in the Algebraic Group Model)". In: *EUROCRYPT 2022, Part II*. Ed. by O. Dunkelman and S. Dziembowski. Vol. 13276. LNCS. Springer, Cham, May 2022, pp. 397–426. DOI: 10.1007/978-3-031-07085-3_14 (cit. on pp. 3, 5–7, 17).

[34] C. Ganesh et al. "What Makes Fiat-Shamir zkSNARKs (Updatable SRS) Simulation Extractable?" In: *SCN 22*. Ed. by C. Galdi and S. Jarecki. Vol. 13409. LNCS. Springer, Cham, Sept. 2022, pp. 735–760. DOI: 10.1007/978-3-031-14791-3_32 (cit. on pp. 3, 6, 7, 14, 17, 18).

[35] C. Ganesh et al. "Fiat-Shamir Bulletproofs are Non-malleable (in the Random Oracle Model)". In: *Journal of Cryptology* 38.1 (Jan. 2025), p. 11. DOI: 10.1007/s00145-024-09525-2 (cit. on p. 3).

[36] A. Garreta et al. *Cyclo: Lightweight Lattice-based Folding via Partial Range Checks*. Cryptology ePrint Archive, Paper 2026/359. 2026. URL: https://eprint.iacr.org/2026/359 (cit. on p. 7).

[37] C. Gentry, S. Halevi, and V. Lyubashevsky. "Practical Non-interactive Publicly Verifiable Secret Sharing with Thousands of Parties". In: *EUROCRYPT 2022, Part I*. Ed. by O. Dunkelman and S. Dziembowski. Vol. 13275. LNCS. Springer, Cham, May 2022, pp. 458–487. DOI: 10.1007/978-3-031-06944-4_16 (cit. on p. 49).

[38] S. Goldwasser, S. Micali, and C. Rackoff. "The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract)". In: *17th ACM STOC*. ACM Press, May 1985, pp. 291–304. DOI: 10.1145/22145.22178 (cit. on p. 3).

[39] J. Groth. "Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures". In: *ASIACRYPT 2006*. Ed. by X. Lai and K. Chen. Vol. 4284. LNCS. Springer, Berlin, Heidelberg, Dec. 2006, pp. 444–459. DOI: 10.1007/11935230_29 (cit. on p. 3).

[40] M. Klooß et al. "RoK, Paper, SISsors Toolkit for Lattice-Based Succinct Arguments - (Extended Abstract)". In: *ASIACRYPT 2024, Part V*. Ed. by K.-M. Chung and Y. Sasaki. Vol. 15488. LNCS. Springer, Singapore, Dec. 2024, pp. 203–235. DOI: 10.1007/978-981-96-0935-2_7 (cit. on p. 7).

[41] M. Kohlweiss, M. Pancholi, and A. Takahashi. "How to Compile Polynomial IOP into Simulation-Extractable SNARKs: A Modular Approach". In: *TCC 2023, Part III*. Ed. by G. N. Rothblum and H. Wee. Vol. 14371. LNCS. Springer, Cham, Nov. 2023, pp. 486–512. DOI: 10.1007/978-3-031-48621-0_17 (cit. on p. 3).

[42] A. Kothapalli and B. Parno. "Algebraic Reductions of Knowledge". In: *CRYPTO 2023, Part IV*. Ed. by H. Handschuh and A. Lysyanskaya. Vol. 14084. LNCS. Springer, Cham, Aug. 2023, pp. 669–701. DOI: 10.1007/978-3-031-38551-3_21 (cit. on pp. 3, 10, 13).

[43] A. Kothapalli, S. Setty, and I. Tzialla. "Nova: Recursive Zero-Knowledge Arguments from Folding Schemes". In: *CRYPTO 2022, Part IV*. Ed. by Y. Dodis and T. Shrimpton. Vol. 13510. LNCS. Springer, Cham, Aug. 2022, pp. 359–388. DOI: 10.1007/978-3-031-15985-5_13 (cit. on pp. 3, 6, 7, 36).

[44] A. Kothapalli and S. T. V. Setty. "HyperNova: Recursive Arguments for Customizable Constraint Systems". In: *CRYPTO 2024, Part X*. Ed. by L. Reyzin and D. Stebila. Vol. 14929. LNCS. Springer, Cham, Aug. 2024, pp. 345–379. DOI: 10.1007/978-3-031-68403-6_11 (cit. on p. 3).

[45] H. Lipmaa. "Plonk is Simulation Extractable in ROM Under Falsifiable Assumptions". In: *TCC 2025, Part IV*. Ed. by B. Applebaum and H. (Lin. Vol. 16271. LNCS. Springer, Cham, Dec. 2025, pp. 3–36. DOI: 10.1007/978-3-032-12290-2_1 (cit. on pp. 3, 6, 17).

[46] V. Lyubashevsky. "Lattice Signatures without Trapdoors". In: *EUROCRYPT 2012*. Ed. by D. Pointcheval and T. Johansson. Vol. 7237. LNCS. Springer, Berlin, Heidelberg, Apr. 2012, pp. 738–755. DOI: 10.1007/978-3-642-29011-4_43 (cit. on pp. 7, 9).

[47] V. Lyubashevsky, N. K. Nguyen, and M. Plancon. *Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General*. Cryptology ePrint Archive, Report 2022/284. 2022. URL: https://eprint.iacr.org/2022/284 (cit. on p. 33).

[48] 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*. Ed. by Y. Dodis and T. Shrimpton. Vol. 13508. LNCS. Springer, Cham, Aug. 2022, pp. 71–101. DOI: 10.1007/978-3-031-15979-4_3 (cit. on pp. 6, 8, 10, 33, 49).

[49] V. Lyubashevsky and G. Seiler. "Short, Invertible Elements in Partially Splitting Cyclotomic Rings and Applications to Lattice-Based Zero-Knowledge Proofs". In: *EUROCRYPT 2018, Part I*. Ed. by J. B. Nielsen and V. Rijmen. Vol. 10820. LNCS. Springer, Cham, Apr. 2018, pp. 204–224. DOI: 10.1007/978-3-319-78381-9_8 (cit. on pp. 7, 8).

[50] D. Micciancio and C. Peikert. "Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller". In: *EUROCRYPT 2012*. Ed. by D. Pointcheval and T. Johansson. Vol. 7237. LNCS. Springer, Berlin, Heidelberg, Apr. 2012, pp. 700–718. DOI: 10.1007/978-3-642-29011-4_41 (cit. on pp. 6, 9, 27).

[51] W. Nguyen and S. Setty. *Neo: Lattice-based folding scheme for CCS over small fields and pay-per-bit commitments*. Cryptology ePrint Archive, Report 2025/294. 2025. URL: https://eprint.iacr.org/2025/294 (cit. on p. 7).

[52] C. Peikert. "Limits on the Hardness of Lattice Problems in ℓ_p Norms". In: *Computational Complexity* 17.2 (2008), pp. 300–351. DOI: 10.1007/s00037-008-0251-3 (cit. on p. 9).

[53] A. Sahai. "Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security". In: *40th FOCS*. IEEE Computer Society Press, Oct. 1999, pp. 543–553. DOI: 10.1109/SFFCS.1999.814628 (cit. on p. 3).

[54] C.-P. Schnorr. "Efficient Identification and Signatures for Smart Cards". In: *CRYPTO'89*. Ed. by G. Brassard. Vol. 435. LNCS. Springer, New York, Aug. 1990, pp. 239–252. DOI: 10.1007/0-387-34805-0_22 (cit. on p. 3).

[55] S. Setty. "Spartan: Efficient and General-Purpose zkSNARKs Without Trusted Setup". In: *CRYPTO 2020, Part III*. Ed. by D. Micciancio and T. Ristenpart. Vol. 12172. LNCS. Springer, Cham, Aug. 2020, pp. 704–737. DOI: 10.1007/978-3-030-56877-1_25 (cit. on p. 3).

[56] S. T. V. Setty, J. Thaler, and R. S. Wahby. "Unlocking the Lookup Singularity with Lasso". In: *EUROCRYPT 2024, Part VI*. Ed. by M. Joye and G. Leander. Vol. 14656. LNCS. Springer, Cham, May 2024, pp. 180–209. DOI: 10.1007/978-3-031-58751-1_7 (cit. on p. 3).

[57] R. S. Wahby et al. "Doubly-Efficient zkSNARKs Without Trusted Setup". In: *2018 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2018, pp. 926–943. DOI: 10.1109/SP.2018.00060 (cit. on p. 3).

A Proof of Lemma 10

We prove the following anti-concentration lemma. The proof follows the same strategy of [48] but it starts from an improved version of the lemma from [9].

Lemma⋆ 10 (Generalised Lemma 2.9 from [48]). *Let $q \in \mathbb{N}$. For every $\lambda \in \mathbb{N}$, there exist constants $D_1 = D_1(\lambda)$ and $D_2 = D_2(\lambda)$ such that the following holds. Let $\vec{w} \in \mathbb{Z}_q^n$ with $\|\vec{w}\|_2 \geq b$ for some bound $b \leq q/D_1$ and let $\vec{y}$ be an arbitrary vector in $\mathbb{Z}_q^{2\lambda}$,*

$$\Pr_{\Pi \leftarrow \mathsf{Bin}_1^{2\lambda \times n}} \left[\|\Pi \vec{w} + \vec{y} \bmod q\|_2 < \sqrt{D_2} b \right] \lesssim 2^{-\lambda/2},$$

where $\lesssim$ stresses the heuristic nature of the result.

First we prove an equivalent of Lemma 4.2 of [9] for the Bin_2 distribution and security parameter $\lambda = 256$. Similarly to [1], we postulate an asymptotic version, and provide a script, `jlbin2.py` which we implemented starting from the script `jl.py` from [1] that computes the bounds for different security parameters.

We stress the heuristic nature of both lemmas. The assumption H2 in the lemmas below is taken from Lemma 2.8 [48]. Following [37], we set the values for H1 and H3 replacing the Bin_2 distribution with scaled chi-square distribution with variance 1, matching the variance of centered binomial distribution with $\kappa = 2$[12].

Lemma⋆ 29 (JL lemma for Bin_2). *Let $q \in \mathbb{N}$ and let $d \geq 1$. Assume the following heuristic tail bounds for every fixed vector $\vec{x} \in [-q/2, q/2]^d$:*

$$\Pr_{\vec{\pi} \xleftarrow{\$} \mathsf{Bin}_2^d} [|\langle \vec{\pi}, \vec{x} \rangle| > 19.29 \|\vec{x}\|_2] \lesssim 2^{-266}, \tag{H1}$$

$$\Pr_{\Pi \xleftarrow{\$} \mathsf{Bin}_2^{512 \times d}} \left[\|\Pi \vec{x}\|_2^2 < 26 \|\vec{x}\|_2^2 \right] \lesssim 2^{-256}, \tag{H2}$$

$$\Pr_{\Pi \xleftarrow{\$} \mathsf{Bin}_2^{512 \times d}} \left[\|\Pi \vec{x}\|_2^2 > 1357 \|\vec{x}\|_2^2 \right] \lesssim 2^{-256}. \tag{H3}$$

Then, for every vector $\vec{w} \in [-q/2, q/2]^d$ such that $\|\vec{w}\|_2 \geq b$ and $b \leq \frac{q}{640}$ we have

$$\Pr_{\Pi \xleftarrow{\$} \mathsf{Bin}_2^{512 \times d}} \left[\|\Pi \vec{w} \bmod q\|_2 < \sqrt{26}\, b \right] \lesssim 2^{-256}.$$

Proof. We follow the proof strategy of Lemma 4.2 in LaBRADOR and split into three cases. Let $T := \frac{q}{640}$. By assumption, $b \leq T$.

Case 1: $\|\vec{w}\|_2 < q/40$. Assume that $\|\Pi \vec{w} \bmod q\|_2 < \sqrt{26}\, b$. If no coordinate of $\Pi \vec{w}$ wraps modulo q, then

$$\|\Pi \vec{w}\|_2 < \sqrt{26}\, b \leq \sqrt{26}\, \|\vec{w}\|_2,$$

which, by the heuristic lower-tail bound (H2), happens with probability less than 2^{-256}.

Otherwise, some row $\vec{\pi}_i$ of Π wraps modulo q. Then for some non-zero integer k,

$$\langle \vec{\pi}_i, \vec{w} \rangle \in [kq - \sqrt{26}\, b,\ kq + \sqrt{26}\, b].$$

Hence $|\langle \vec{\pi}_i, \vec{w} \rangle| > q - \sqrt{26}\, b \geq q - \sqrt{26}\, \frac{q}{640}$. Since $1 - \frac{\sqrt{26}}{640} > 0.992$, and since $\|\vec{w}\|_2 < q/40$, we get $|\langle \vec{\pi}_i, \vec{w} \rangle| > 39.68\, \|\vec{w}\|_2 > 19.29\, \|\vec{w}\|_2$. By (H1), this happens for a fixed row with probability less than 2^{-266}. Taking a union bound over the $512 = 2^9$ rows gives $512 \cdot 2^{-266} = 2^{-257}$. Therefore, in Case 1,

$$\Pr_{\Pi} \left[\|\Pi \vec{w} \bmod q\|_2 < \sqrt{26}\, b \right] < 2^{-256} + 2^{-257}.$$

[12] We compute this value using a python script `jlbin2.py` which we implemented starting from the script `jl.py` from [1] which can be found at https://github.com/dfaranha/aggregate-falcon. Note that the code in `jlbin2.py` was generated with the support of LLM. Our code can be found at https://gitlab.eurecom.fr/faonio/zk-labradot.

Case 2: $\|\vec{w}\|_\infty \geq q/160$**.** Let i be an index such that $|w_i| \geq \frac{q}{160}$. Let $\vec{\pi}$ be one row of Π. We condition on all entries of $\vec{\pi}$ except π_i. Write $a := \sum_{j \neq i} \pi_j w_j \pmod q$, where the result is represented in $[-q/2, q/2]$.

We claim that

$$\Pr_{\pi_i \xleftarrow{\$} \mathsf{Bin}_2} [|a + \pi_i w_i \bmod q| \leq T] \leq \frac{1}{2}.$$

Indeed, the possible values of π_i are $-2, -1, 0, 1, 2$. Since $|w_i| \geq q/160 = 4T$, an interval of radius T modulo q cannot contain two adjacent values among

$$a - 2w_i,\ a - w_i,\ a,\ a + w_i,\ a + 2w_i \pmod q.$$

Thus the set of values of π_i that can make $|a + \pi_i w_i \bmod q| \leq T$ contains no adjacent elements of $\{-2, -1, 0, 1, 2\}$. Therefore each row is large, meaning $|\langle \vec{\pi}, \vec{w} \rangle \bmod q| > T$, with probability at least $1/2$.

Notice that $\sum_{j=0}^{57} \binom{512}{j} < 2^{256}$. Thus, except with probability less than 2^{-256}, at least 58 rows are large. In that case,

$$\|\Pi \vec{w} \bmod q\|_2 > \sqrt{58}\, T.$$

Since $b \leq T$ and $58 > 26$, we have $\sqrt{58}\, T > \sqrt{26}\, b$. Therefore, in Case 2,

$$\Pr_{\Pi} \left[\|\Pi \vec{w} \bmod q\|_2 < \sqrt{26}\, b \right] < 2^{-256}.$$

Case 3: $\|\vec{w}\|_2 \geq q/40$ **and** $\|\vec{w}\|_\infty < q/160$**.** As in the proof of the lemma in LaBRADOR, we construct a vector $\vec{v}$ from $\vec{w}$ by repeatedly setting coordinates to zero until $\|\vec{v}\|_2 < \frac{q}{40}$.

Since each removed coordinate has absolute value less than $q/160$, the last removal can decrease the squared norm by less than $q^2/160^2$. Therefore the final vector $\vec{v}$ satisfies $\|\vec{v}\|_2^2 \geq \frac{q^2}{40^2} - \frac{q^2}{160^2} = \frac{15q^2}{160^2}$. Equivalently, $\frac{\sqrt{15}}{160} q \leq \|\vec{v}\|_2 < \frac{q}{40}$. Set $M := \frac{160}{\sqrt{15}}$, then $\frac{q}{M} \leq \|\vec{v}\|_2 < \frac{q}{40}$. Moreover, $v_j = w_j$ or $v_j = 0$ for every j, and the supports of $\vec{v}$ and $\vec{w} - \vec{v}$ are disjoint. Hence, for a row $\vec{\pi}$ of Π,

$$\langle \vec{\pi}, \vec{w} \rangle = \langle \vec{\pi}, \vec{v} \rangle + \langle \vec{\pi}, \vec{w} - \vec{v} \rangle,$$

and the two summands are independent.

Let $S := \langle \vec{\pi}, \vec{v} \rangle$. For $\pi_j \xleftarrow{\$} \mathsf{Bin}_2$, we have

$$\mathbb{E}[\pi_j] = 0, \qquad \mathbb{E}[\pi_j^2] = 1, \qquad \mathbb{E}[|\pi_j|^3] = \frac{3}{2}.$$

Thus the summand $\pi_j v_j$ has variance v_j^2 and third absolute moment $\frac{3}{2}|v_j|^3$. Applying the Berry–Esseen theorem with constant 0.52, and letting Z be a normal random variable with mean 0 and variance $\|\vec{v}\|_2^2$, we get

$$\sup_x |\Pr[S \leq x] - \Pr[Z \leq x]| \leq 0.52 \cdot \frac{3}{2} \frac{\|\vec{v}\|_\infty}{\|\vec{v}\|_2}.$$

Since $\|\vec{v}\|_\infty \leq \|\vec{w}\|_\infty < \frac{q}{160}$ and $\|\vec{v}\|_2 \geq \frac{\sqrt{15}}{160} q$, we obtain

$$\sup_x |\Pr[S \leq x] - \Pr[Z \leq x]| \leq 0.52 \cdot \frac{3}{2} \cdot \frac{1}{\sqrt{15}} < 0.202.$$

Denote this Berry–Esseen error by $\Delta := 0.52 \cdot \frac{3}{2} \cdot \frac{1}{\sqrt{15}} < 0.202$.

Now fix any value $a \in [-q/2, q/2]$. We upper bound the probability that

$$|S + a \bmod q| < \sqrt{26}\, b.$$

First, because $\|\vec{v}\|_2 < \frac{q}{40}$, we have $\frac{q}{2} > 20\|\vec{v}\|_2 > 19.29\|\vec{v}\|_2$. Therefore, by (H1),

$$\Pr[|S| > q/2] < 2^{-266}.$$

Conditioned on $|S| \leq q/2$, the bad event can only occur when $S \in G_a$, where

$$G_a := \left\{ x \in [-q/2, q/2] : |x + a \bmod q| < \sqrt{26}\, b \right\}.$$

The set G_a has measure at most $2\sqrt{26}\,b \le 2\sqrt{26}\,T = 2\sqrt{26}\frac{q}{640}$. The density of Z is pointwise at most $\frac{1}{\sqrt{2\pi}\,\|\vec{v}\|_2} \le \frac{M}{\sqrt{2\pi}\,q}$. Therefore $\Pr[Z \in G_a] \le \frac{2\sqrt{26}\,q/640}{\sqrt{2\pi}\,q/M} = \frac{2\sqrt{26}\,M}{640\sqrt{2\pi}} < 0.263$. Either G_a or its complement in $[-q/2, q/2]$ is an interval, so the Berry–Esseen bound gives

$$\Pr[S \in G_a] \le \Pr[Z \in G_a] + 2\Delta < 0.263 + 2 \cdot 0.202 < 0.666.$$

Hence, for every fixed value of $\langle \vec{\pi}, \vec{w} - \vec{v} \rangle \bmod q$, one row is bad with probability at most $0.666 + 2^{-266}$. The rows are independent, so

$$\Pr_{\Pi}\left[\|\Pi\vec{w} \bmod q\|_2 < \sqrt{26}\,b\right] \le (0.666 + 2^{-266})^{512} < 2^{-300} < 2^{-256}.$$

□

We now state the lemma for arbitrary constants C_1' and C_2'.

Lemma 30. *Let $q \in \mathbb{N}$. For every $\lambda \in \mathbb{N}$, there exist constants $C_1' = C_1'(\lambda)$ and $C_2' = C_2'(\lambda)$ such that the following holds. Let $\vec{w} \in \mathbb{Z}^n$ with $\|\vec{w}\|_2 < b$ for some bound $b \le q/C_1'$,*

$$\Pr_{\Pi \xleftarrow{\$} \mathsf{Bin}_2^{2\lambda \times n}}\left[\|\Pi\vec{w} \bmod q\|_2 > \sqrt{C_2'}\|\vec{w} \bmod q\|_2\right] \lesssim 2^{-\lambda},$$

where $\lesssim$ stresses the heuristic nature of the result.

We now prove the Lemma 10.

Proof. We set $D_1 := C_1'$ and $D_2 := 2C_2'$. Suppose for contradiction that for some $\vec{w}$ and $\vec{y}$ we have:

$$\Pr_{\Pi \leftarrow \mathsf{Bin}_1^{2\lambda \times m}}[\|\Pi\vec{w} + \vec{y} \bmod q\|_2 < \sqrt{C_2'}\|\vec{w} \mod q\|_2] \ge 2^{-\lambda/2}$$

This implies that

$$\Pr_{\Pi_1, \Pi_2 \leftarrow \mathsf{Bin}_1^{2\lambda \times m}}\begin{bmatrix}\|\Pi_1\vec{w} + \vec{y} \bmod q\|_2 < \sqrt{C_2'}\|\vec{w} \mod q\|_2 \wedge \\ \|\Pi_2\vec{w} + \vec{y} \bmod q\|_2 < \sqrt{C_2'}\|\vec{w} \mod q\|_2\end{bmatrix} \ge 2^{-\lambda}$$

Since $\Pi_1 - \Pi_2$ is distributed as $\mathsf{Bin}_2^{2\lambda \times m}$, the above implies that

$$\Pr_{\Pi \leftarrow \mathsf{Bin}_2^{2\lambda \times m}}[\|\Pi\vec{w} + \vec{y} \bmod q\|_2 < \sqrt{C_2'}\|\vec{w} \mod q\|_2] \ge 2^{-\lambda}$$

which is in contradiction with Lemma 30.

B Additional Material for Section 7

B.1 Hiding for (the double) commitment scheme

Lemma 31 (2-splitting case of Corollary 4, [15]). *Assume the ring setting for LaBRADOR as described in Section 2.2. Let m, n, η be positive integers such that $\eta < q/2$. Recall $U_\eta = \{\boldsymbol{x} \in \mathcal{R} : \|\boldsymbol{x}\|_\infty < \eta\}$. We define*

$$\mathcal{P}' = \{(\boldsymbol{A}\vec{\boldsymbol{x}} \bmod q, \boldsymbol{A}) \mid \boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m},\ \vec{\boldsymbol{x}} \xleftarrow{\$} U_\eta^m\},$$

and

$$\mathcal{U}' = \{(\vec{\boldsymbol{u}}, \boldsymbol{A}) \mid \boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m},\ \vec{\boldsymbol{u}} \xleftarrow{\$} \mathcal{R}_q^n\}.$$

Then, it holds

$$\mathrm{SD}(\mathcal{P}', \mathcal{U}') \le \frac{1}{2}\sqrt{\left(\left(\frac{q^n}{(2\eta+1)^m}\right)^{\frac{d}{2}} + 1\right)^2 - 1}$$

Lemma 32 (Hiding Com). *Assume the ring setting for LaBRADOR as described in Section 2.2. Let $n_r, m, \kappa, \kappa_1, \eta$ be positive integers such that $\eta, b < q/2$, commitment key $\boldsymbol{B}_i \xleftarrow{\$} \mathcal{R}_q^{\kappa_1 \times \kappa t}$ for any i, messages $\vec{\boldsymbol{m}_0}, \vec{\boldsymbol{m}_1} \in \mathcal{R}_q^m$. Define $U_\eta = \{\boldsymbol{x} \in \mathcal{R} : \|\boldsymbol{x}\|_\infty < \eta\}$. We define*

$$\mathcal{P} = \{(\mathsf{Com}(\mathsf{ck}, \vec{\boldsymbol{m}} \| \vec{\boldsymbol{r}}), (\boldsymbol{B}_i)_i) \mid \forall i : \boldsymbol{B}_i \xleftarrow{\$} \mathcal{R}_q^{\kappa_1 \times \kappa t},\ \vec{\boldsymbol{r}} \xleftarrow{\$} U_\eta^{n_r}\},$$

Then, as long as $n_r > \frac{\kappa_1 \log q + 2\lambda/d}{\log(2\eta+1)}$, we have $\mathrm{SD}(\mathcal{P}, \mathcal{U}) \le 2^{-\lambda}$ holds for any messages $\vec{\boldsymbol{m}}$, where $\mathcal{U}$ is the uniform distribution over the same support of $\mathcal{P}$.

Proof. Notice that $(\mathsf{Com}(\mathsf{ck}, \vec{\boldsymbol{m}} \| \vec{\boldsymbol{r}}), (\boldsymbol{B}_i)_i)$ contains the additive term of the form $\boldsymbol{A}\vec{\boldsymbol{r}}$, where $\boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{\kappa_1 \times n_r}$. Applying Lemma 31 one gets that this term has statistical distance to uniform distribution over $\mathcal{R}_q^{\kappa_1}$ upper bounded by $\frac{1}{2}\sqrt{\left(\left(\frac{q^{\kappa_1}}{(2\eta+1)^{n_r}}\right)^{\frac{d}{2}}+1\right)^2-1} < 2^{-\lambda}$ when $n_r > \frac{\kappa_1 \log q + 2\lambda/d}{\log(2\eta+1)}$.

Applying iteratively Lemma 31, one may compute the statistical distance between the $\mathsf{DCom}_{\mathsf{ck},b}$ of an uniform short vector and uniform distribution:

Lemma 33 (Hiding DCom). *Assume the ring setting for LaBRADOR as described in Section 2.2. Let $m, \kappa, \kappa_1, \eta, b$ be positive integers such that $\eta, b < q/2$, commitment key $\mathsf{ck}_A \xleftarrow{\$} \mathcal{R}_q^{\kappa \times n}$, $\mathsf{ck}_B \xleftarrow{\$} \mathcal{R}_q^{\kappa_1 \times \kappa t}$, a message $\vec{\boldsymbol{m}} \in \mathcal{R}_q^n$. Let $U_\eta = \{\boldsymbol{x} \in \mathcal{R} : \|\boldsymbol{x}\|_\infty < \eta\}$. We define*

$$\mathcal{P}' = \{(\mathsf{DCom}(\mathsf{ck}, b; \vec{\boldsymbol{r}}, \vec{\boldsymbol{m}}), \mathsf{ck}_A, \mathsf{ck}_B) \mid \boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{\kappa \times n}, (\boldsymbol{B}_{ij}) \xleftarrow{\$} \mathcal{R}_q^{\kappa_1 \times \kappa t}, \vec{\boldsymbol{r}} \xleftarrow{\$} U_\eta^{n_r}\},$$

and

$$\mathcal{U}' = \{(\vec{\boldsymbol{u}}, \mathsf{ck}_A, \mathsf{ck}_B) \mid \boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{\kappa \times n}, (\boldsymbol{B}_{ij}) \xleftarrow{\$} \mathcal{R}_q^{\kappa_1 \times \kappa t}, \vec{\boldsymbol{u}} \xleftarrow{\$} \mathcal{R}_q^{\kappa_1}\}.$$

Then, it holds

$$\mathrm{SD}(\mathcal{P}', \mathcal{U}') \le \frac{1}{2}\left(\sqrt{\left(\left(\frac{q^{\kappa}}{(2\eta+1)^{n}}\right)^{\frac{d}{2}}+1\right)^2-1} + \sqrt{\left(\left(\frac{q^{\kappa_1}}{(2b+1)^{\kappa t}}\right)^{\frac{d}{2}}+1\right)^2-1}\right)$$

Proof. Denote by $\mathcal{U}(A)$ the uniform distribution over set A. We first analyse $\mathsf{DCom}(\mathsf{ck}, b; \vec{\boldsymbol{r}})$. Let random variable $X := \mathsf{Com}(\mathsf{ck}_A, \vec{\boldsymbol{r}})$, $Y := \mathsf{Split}_b(X)$, $Z := \mathsf{Com}(\mathsf{ck}_B, \mathcal{U}(U_b^{\kappa t}))$. Applying Lemma 31, we have

$$\mathrm{SD}(X, \mathcal{U}(\mathcal{R}_q^{\kappa})) < \frac{1}{2}\sqrt{\left(\left(\frac{q^{\kappa}}{(2\eta+1)^{n}}\right)^{\frac{d}{2}}+1\right)^2-1}.$$

Now since $\mathsf{Split}_b : \mathcal{R}_q^{\kappa} \to U_b^{\kappa t}$ is bijective, $\mathrm{SD}(Y, \mathcal{U}(U_b^{\kappa t})) = \mathrm{SD}(X, \mathcal{U}(\mathcal{R}_q^{\kappa}))$. Moreover, because linear maps (or more generally deterministic functions) do not increase statistical distance, we have

$$\mathrm{SD}(Z, \mathsf{DCom}(\mathsf{ck}, b, \vec{\boldsymbol{r}})) \le \mathrm{SD}(Y, \mathcal{U}(U_b^{\kappa t})) = \mathrm{SD}(X, \mathcal{U}(\mathcal{R}_q^{\kappa})).$$

Applying again Lemma 31, we have

$$\mathrm{SD}(Z, \mathcal{U}(\mathcal{R}_q^{\kappa_1})) < \frac{1}{2}\sqrt{\left(\left(\frac{q^{\kappa_1}}{(2b+1)^{\kappa t}}\right)^{\frac{d}{2}}+1\right)^2-1}.$$

Hence by triangular inequality we obtain that:

$$\mathrm{SD}(\mathsf{DCom}(\mathsf{ck}, b, \vec{\boldsymbol{r}}), \mathcal{U}(\mathcal{R}_q^{\kappa_1})) \le \mathrm{SD}(X, \mathcal{U}(\mathcal{R}_q^{\kappa})) + \mathrm{SD}(Z, \mathcal{U}(\mathcal{R}_q^{\kappa_1}))$$

Finally, we note that hiding DCom is defined in such a way that $\mathsf{DCom}(\mathsf{ck}, b; \vec{\boldsymbol{r}}, \vec{\boldsymbol{m}}) := \mathsf{DCom}(\mathsf{ck}, b; \vec{\boldsymbol{r}}) + \mathsf{DCom}(\mathsf{ck}, b; \vec{\boldsymbol{m}})$. Since shifting by a constant is bijective and preserves statistical distance, we conclude the proof. □

Corollary 2. *Adopting the same notation as in Lemma 33. Let randomness $\vec{\boldsymbol{r}}$ be sampled uniformly from $U_\beta^{n_r}$, then $n_r > \frac{\kappa \log q}{\log(2\beta+1)} + \frac{2\lambda+1}{d}\frac{2}{\log(2\beta+1)}$ and $\kappa > \kappa_1 + \frac{2\lambda+1}{d}\frac{2}{\log(2b+1)}$ guarentees $\mathsf{DCom}(\mathsf{ck}, b, (\vec{\boldsymbol{m}}, \vec{\boldsymbol{r}}))$ is $2^{-\lambda}$-close to uniform distribution.*

Proof. Let $\epsilon = 2^{-\lambda}$, to have $\sqrt{\left(\left(\frac{q^{\kappa}}{(2\beta+1)^{n_r}}\right)^{\frac{d}{2}}+1\right)^2-1} < \epsilon$, re-arrange one obtains

$$n_r > \frac{\kappa \log q - \frac{2}{d}(\log(\sqrt{\epsilon^2+1}-1)}{\log(2\beta+1)}$$

By Talor expansion, $\log(\sqrt{\epsilon^2+1}-1) \ge \log(\frac{\epsilon^2}{2} - \frac{\epsilon^4}{8}) = 2\log\epsilon + \log(4-\epsilon^2) - 3 > -2\lambda - 1$. Hence by taking $n_r > \frac{\kappa \log q}{\log(2\beta+1)} + \frac{2\lambda+1}{d}\frac{2}{\log(2\beta+1)}$ we bound this term by ϵ. Similarly for the other term, note that $t \approx \frac{\log q}{\log b}$. □

Theorem 6 (Hiding DCom^*). *Let $\sigma > \max\left\{(t-1)\sqrt{b^2+1}\cdot q^{\kappa_1/(\kappa t)}\cdot 2^{\lambda/(\kappa t d)}, 2(t-1)\sqrt{b^2+1}\right\}$ where $t = \lceil \log q/\log b\rceil$, the double-commitment scheme DCom^* with variance parameter σ and splitting parameter b is statistically hiding.*

Proof. It suffices to argue that for $\vec{\boldsymbol{t}}' \xleftarrow{\$} D_{\Lambda,\sigma}$, $(\mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{t}}'), \mathsf{ck}_B)$ is close to uniform distribution over the same domain. We compute an upper bound on their statistical distance.

First we lower bound the gaussian mass $\rho_\sigma(\Lambda)$. Since $\Lambda = L^{\kappa d}$ where $L = \{x \in \mathbb{Z}^t : \langle x, g\rangle \equiv 0 \bmod q\}$, $g = (1, b, \ldots, b^{t-1})$, it suffices to compute $\rho_\sigma(L)$, we count the number of lattice elements with norm bounded by $\sigma/2$. Note that L is full rank with $t-1$ basis vectors $(v_i)_{i=0}^{t-1}$ of norm $\sqrt{b^2+1}$. For $z_i \in (-A, A) \subset \mathbb{Z}$, $x(z) = \sum_{i=1}^{t-1} z_i v_i$ lies in lattice L. Moreover, by triangular inequality, $\|x(z)\|_2 \le (t-1)A\sqrt{b^2+1}$. Take $A = \left\lfloor \frac{\sigma}{2(t-1)\sqrt{b^2+1}} \right\rfloor$, $\|x(z)\|_2 \le \sigma/2$. There are at least $(2A+1)^t$ such lattice elements, thus $\rho_\sigma(L) \ge e^{-\pi/4}\left(2\left\lfloor \frac{\sigma}{2(t-1)\sqrt{b^2+1}}\right\rfloor + 1\right)^{t-1}$.

Therefore we can compute collision probability $\mathsf{Coll}(D_{\Lambda,\sigma}) \le 2^{-H_\infty(D_{\Lambda,\sigma})} = \frac{1}{\rho_\sigma(\Lambda)} = \rho_\sigma(L)^{\kappa d}$. Applying [15, Corollary 6], one gets $SD(\mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{t}}'), \mathsf{Uniform}) \lesssim \frac{1}{2}\sqrt{|\mathcal{R}_q|^{\kappa_1}((\frac{(t-1)\sqrt{b^2+1}}{\sigma})^{\kappa t d} + 2\kappa t e^{-\pi r^2})}$ when $\sigma r < \frac{\sqrt{q}}{2\sqrt{2}}$.
Equivalently,

$$\max\{(t-1)\sqrt{b^2+1}\cdot q^{\kappa_1/(\kappa t)}\cdot 2^{\lambda/(\kappa t d)}, 2(t-1)\sqrt{b^2+1}\} < \sigma < \frac{\sqrt{q}}{2\sqrt{2}(\log q + \lambda)}$$

implies $SD(\mathsf{Com}(\mathsf{ck}_B, \vec{\boldsymbol{t}}'), \mathsf{Uniform}) \le \mathsf{negl}(\lambda)$.

LaBRADot Π from $\hat{\mathcal{R}}$ to $\mathcal{R}$

	Prover		**Verifier**
1 :	**Let** $\mathbb{w} = (\vec{\boldsymbol{s}}_i)_i, (\vec{\boldsymbol{t}}_{ij})_{i,j}$		**Let** $\mathbb{x} = ((\mathsf{c}_k, I_k, \gamma_k), \mathcal{F}, \mathcal{F}', \vec{\beta})$
2 :	$\forall i,j : (\boldsymbol{g}_{ijk})_k \leftarrow \mathsf{Split}(\langle \vec{\boldsymbol{s}}_i, \vec{\boldsymbol{s}}_j\rangle)$		
3 :	$\vec{\boldsymbol{u}}_1 \leftarrow \mathsf{Com}(\mathsf{ck}_C, (\boldsymbol{g}_{ijk})_{i,j,k})$	$\xrightarrow{\vec{\boldsymbol{u}}_1}$	$(P_i)_{i\in[r]} \xleftarrow{\$} \mathsf{Bin}_1^{(2\lambda\times nd)\times r}$
4 :		$\xleftarrow{(P_i)_i}$	
5 :	$\vec{\boldsymbol{p}} \leftarrow \sum_i P_i \cdot \tau(\vec{\boldsymbol{s}}_i) \in \mathbb{Z}_q^{2\lambda}$	$\xrightarrow{\vec{\boldsymbol{p}}}$	**Check** $\|\vec{\boldsymbol{p}}\|_2 \le \sqrt{\lambda}\beta$
6 :	$q''_k(\vec{\boldsymbol{s}}) := \sum_l \psi_{kl}\cdot q'_l(\vec{\boldsymbol{s}})$	$\xleftarrow{(\vec{\psi}_k, \vec{\omega}_k)_k}$	$\forall k \in [\lambda'] : \vec{\omega}_k \xleftarrow{\$} \mathbb{Z}_q^{2\lambda}$
7 :	$l''_k(\vec{\boldsymbol{s}}) := \sum_l \psi_{kl}\cdot l'_l(\vec{\boldsymbol{s}}) + \sum_{i,j}\omega_{kj}\langle \sigma_{-1}(\vec{\boldsymbol{\pi}}_i), \vec{\boldsymbol{s}}_i\rangle$		$\forall k \in [\lambda'] : (\vec{\psi}_k)_k \xleftarrow{\$} (\mathbb{Z}_q^{K'})^{\lambda'}$,
8 :	**Parse** $l''_k(\vec{\boldsymbol{s}}) = \sum_i \langle \vec{\psi}''_{ki}, \vec{\boldsymbol{s}}_i\rangle$		
9 :	$\boldsymbol{b}''_k \leftarrow f''_k(\vec{\boldsymbol{s}})$	$\xrightarrow{(\boldsymbol{b}''_k)_k}$	$\vec{\boldsymbol{\alpha}} \xleftarrow{\$} \mathcal{R}_q^K, \vec{\boldsymbol{\beta}} \xleftarrow{\$} \mathcal{R}_q^{\lambda'}$
10 :	$\vec{\boldsymbol{\psi}}_i \leftarrow \sum_k \boldsymbol{\alpha}_k \vec{\boldsymbol{\psi}}_{ik} + \sum_k \vec{\boldsymbol{\beta}}_k \vec{\boldsymbol{\psi}}''_{ki}$	$\xleftarrow{\vec{\boldsymbol{\alpha}}, \vec{\boldsymbol{\beta}}}$	
11 :	$(\boldsymbol{h}_{ijk})_k \leftarrow \mathsf{Split}(\frac{1}{2}(\langle \vec{\boldsymbol{\psi}}_i, \vec{\boldsymbol{s}}_i\rangle + \langle \vec{\boldsymbol{\psi}}_j, \vec{\boldsymbol{s}}_j\rangle))$		
12 :	$\boldsymbol{u}_2 \leftarrow \mathsf{Com}(\mathsf{ck}_D, (\boldsymbol{h}_{ijk})_{i,j,k})$	$\xrightarrow{\boldsymbol{u}_2}$	$\vec{\boldsymbol{c}} \xleftarrow{\$} \mathcal{C}^r$
13 :	$\bar{\boldsymbol{z}} \leftarrow \sum_i \boldsymbol{c}_i \vec{\boldsymbol{s}}_i$	$\xleftarrow{\vec{\boldsymbol{c}}}$	
14 :	**return** $\bar{\boldsymbol{z}}, \vec{\boldsymbol{g}}, \vec{\boldsymbol{h}}, \vec{\boldsymbol{t}}$		**return** $((\tilde{f}_k)_k, \emptyset, \tilde{\beta})$

Fig. 11: commit-and-prove LaBRADot . τ is a coefficient embedding mapping. We set $\lambda' = \lambda/\log q$. Functions $(\tilde{f}_k)_k$ as in Eq. (4).

B.2 Aggregation with Partial Zero-Knowledge

RoK $\Pi_{\mathsf{zk\text{-}agg}'}$ from $\hat{\mathcal{R}}$ to $\hat{\mathcal{R}}$

	Prover $\mathbb{w} = (\vec{\boldsymbol{s}}, \vec{\boldsymbol{t}})$		**Verifier** $\mathbb{x} = (\mathsf{c}, \mathcal{F}, \mathcal{F}')$
1 :	$\forall k \in [\lambda'] : \boldsymbol{g}_k \xleftarrow{\$} \mathcal{R}_q \cap \mathsf{ct} = 0$		
2 :	$\mathsf{c}_g, \vec{\boldsymbol{t}}_g \leftarrow \mathsf{DCom}^*(\mathsf{ck}, (\boldsymbol{g}_k)_k)$	$\xrightarrow{\mathsf{c}_g}$	
3 :	$\forall k \in [\lambda'] : \boldsymbol{b}_k \leftarrow \boldsymbol{g}_k + \sum_j \psi_{kj} q'_j(\vec{\boldsymbol{s}})$	$\xleftarrow{(\vec{\psi}_k)_k}$	$\forall k \in [\lambda'] : \vec{\psi}_k \xleftarrow{\$} \mathbb{Z}_q^{K'}$
4 :		$\xrightarrow{(\boldsymbol{b}_k)_k}$	**Check** $\mathsf{ct}(\boldsymbol{b}_k) = \sum_j \psi_{kj} \mathsf{ct}(\boldsymbol{b}'_j)$
5 :			$\forall k : f''_k(\vec{X}) := \sum_j \psi_{kj} q'_j(\vec{\boldsymbol{s}}) + \langle \delta_k, \vec{\boldsymbol{g}} \rangle - \boldsymbol{b}_k$
6 :	**return** $\vec{\boldsymbol{s}} = (\vec{\boldsymbol{s}}_j)_j, (\boldsymbol{g}_k)_k, \vec{\boldsymbol{t}}_g, \vec{\boldsymbol{t}}$		**return** $(\mathsf{c}, \mathcal{F} \cup (f''_k)_{k \in [\lambda']}, \emptyset)$

C Fiat-Shamir-based Signature-of-Knowledge

Definition 16 (Signature Reduction of Knowledge). *Let* R_1 *and* R_2 *be two polynomial-time relations. We say* $\Pi : \mathsf{R}_1 \to \mathsf{R}_2$ *is a signature reduction of knowledge,* SRoK *in short, if* Π *is a complete, zero-knowledge and simulation extractable reduction of knowledge for* $\mathsf{R}'_1 \to \mathsf{R}_2$ *where the augmented starting relation* R'_1 *is such that:*

$$\forall \mathsf{msg} \in \{0,1\}^\lambda : \mathsf{R}_1(\mathsf{pp}, x, w) \iff \mathsf{R}'_1(\mathsf{pp}, (\mathsf{msg}, x), w),$$

where msg *is referred to as the signed message.*

Throughout this section we denote by $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ an interactive argument of knowledge.

Definition 17 (Fiat–Shamir* Transformation). *The Fiat–Shamir* transformation turns an interactive argument* Π *into a non-interactive argument which we denote* $\Pi_{\mathsf{FS}^*} = (\mathsf{Setup}, \mathsf{P}', \mathsf{V}')$ *in the random oracle model, where:*

- Setup *is the same as that of* Π*.*
- *The prover* P'*, on input* $(\mathsf{pp}, \mathsf{msg}, x; w)$*, proceeds as in the Fiat–Shamir transformation, except that for each* RO *query, instead of querying* $(\mathsf{pp} \| x \| \pi|_i)$*, it queries* $(\mathsf{pp} \| \mathsf{msg} \| x \| \pi|_i)$*.*
- P' *outputs* (msg, x, π)*.*
- *The verifier* V'*, on input* $(\mathsf{pp}, \mathsf{msg}, x, \pi)$*, proceeds as in the Fiat–Shamir transformation, except that for each* RO *query, instead of querying* $(\mathsf{pp} \| x \| \pi|_i)$*, it queries* $(\mathsf{pp} \| \mathsf{msg} \| x \| \pi|_i)$*.*

We notice that FS^* is preserving zero-knowledge: let $\mathcal{S}$ be a simulator of Π, we construct simulator $\mathcal{S}'$ of Π_{FS^*}

Simulator $\mathcal{S}'$

1. Initialise two empty sets $\mathcal{T}, \mathcal{T}'$. Upon receiving simulation queries of form $(\mathsf{pp}, \mathsf{msg}, x)$, locally run simulator $\mathcal{S}$ on (pp, x). Upon $\mathcal{S}$ adding reprogrammed RO queries of form $(\mathsf{pp} \| x \| \text{rest})$ to $\mathcal{T}$, add queries $(\mathsf{pp} \| \mathsf{msg} \| x \| \text{rest})$ with the same reprogrammed value to $\mathcal{T}'$.
2. Eventually $\mathcal{S}(\mathsf{pp}, x) \to (x, \pi)$. $\mathcal{S}'$ Outputs (msg, x, π)

Lemma 34. *Let* Π *be an interactive proof of knowledge. If* Π_{FS} *is* SIM-EXT*, then* Π_{FS^*} *is a signature of knowledge. Concretely, for any* Π_{FS^*} *sim-ext adversary* $\mathcal{A}$ *exists* Π_{FS} *sim-ext adversary* $\mathcal{B}$ *such that* $\mathsf{Adv}^{\mathsf{SE}}(\Pi_{\mathsf{FS}^*}, \mathcal{S}', \mathcal{E}, \mathcal{A}) \leq q \cdot \mathsf{Adv}^{\mathsf{SE}}(\Pi_{\mathsf{FS}}, \mathcal{S}, \mathcal{E}, \mathcal{B})$*, where* $q = \mathsf{poly}(\lambda)$ *denotes an upper bound on the number of* RO *queries made by* $\mathcal{A}$*.*

Proof. Let $\mathcal{A}$ be an adversary against the SIM-EXT game for Π_{FS^*}. We construct an adversary $\mathcal{B}$ against the SIM-EXT game for Π_{FS} by a standard random-oracle guessing argument. Before returning its forgery, we let $\mathcal{A}$ make all random-oracle queries that the verifier makes for that forgery. This modification does not change the distribution of the game, since these are only additional queries to the same random oracle.

Adversary $\mathcal{B}(\mathsf{pp})$:

1. Choose $i^* \xleftarrow{\$} [q]$, initialize an empty local random-oracle table for each message, and run $\mathcal{A}(\mathsf{pp})$.
2. Answer the first $i^* - 1$ random-oracle queries of $\mathcal{A}$ by lazy sampling in the corresponding local table.
3. On the i^*-th query of the form $\mathsf{pp}\|\mathsf{msg}^*\|z$, designate msg^* as the distinguished message and return the answer of $\mathcal{B}$'s random oracle on $\mathsf{pp}\|z$.
4. On every later query $\mathsf{pp}\|\mathsf{msg}\|z$, return the answer of $\mathcal{B}$'s random oracle on $\mathsf{pp}\|z$ if $\mathsf{msg} = \mathsf{msg}^*$ and otherwise answer from the local table for msg.
5. Answer a simulation query (msg, x) by emulating $\mathcal{S}'$ using the table for msg.
6. If $\mathsf{msg} = \mathsf{msg}^*$, forward x to the simulation oracle $\mathcal{S}$ and map each programmed point $\mathsf{pp}\|z$ to $\mathsf{pp}\|\mathsf{msg}^*\|z$.
7. If $\mathsf{msg} \neq \mathsf{msg}^*$, run the same simulation locally using the local table for msg.
8. Upon receiving (msg, x, π) from $\mathcal{A}$, output (x, π) if $\mathsf{msg} = \mathsf{msg}^*$ and abort otherwise.

The local tables are extended consistently with all random-oracle values and all reprogrammed points produced before the distinguished message is selected. Consequently, this construction also handles simulation queries made before the distinguished message is selected. Thus, by lazy sampling, the view of $\mathcal{A}$ is distributed exactly as in the $\mathsf{SIM\text{-}EXT}$ game for Π_{FS^*}.

Let (msg, x, π) be the output of $\mathcal{A}$, and let j be the index of its first random-oracle query in the namespace $\mathsf{pp}\|\mathsf{msg}\|\cdot$. Such an index exists by the preliminary verifier queries. If $i^* = j$, then $\mathsf{msg}^* = \mathsf{msg}$, and the random-oracle answers used to verify (msg, x, π) are precisely the answers used to verify (x, π) in the Π_{FS} game. Moreover, the simulation queries with message msg are mapped exactly to the simulation queries recorded by $\mathcal{S}$. Hence, whenever $\mathcal{A}$ produces a valid fresh forgery, $\mathcal{B}$ produces a valid fresh forgery (x, π). Since i^* is uniform and independent of the view of $\mathcal{A}$, the event $i^* = j$ occurs with probability $1/q$. Thus:

$$\mathsf{Adv}^{\mathtt{SE}}(\Pi_{\mathsf{FS}^*}, \mathcal{S}', \mathcal{E}, \mathcal{A}) \leq q \cdot \mathsf{Adv}^{\mathtt{SE}}(\Pi_{\mathsf{FS}}, \mathcal{S}, \mathcal{E}, \mathcal{B}).$$

Corollary 3. *Let $\Pi : \mathsf{R}_1 \to \mathsf{R}_2$ be an interactive reduction of knowledge. If Π_{FS} is $\mathsf{SIM\text{-}EXT}$, then Π_{FS^*} is a signature reduction of knowledge.*