Adaptive Special Soundness
A new framework for knowledge extraction with applications to Kilian's protocol

Thomas Attema[1,2], Michael Klooß[3], Russell W. F. Lai[4], and Pavlo Yatsyna[5]

[1] CWI, Cryptology Group
[2] TNO, Applied Cryptography & Quantum Applications
thomas.attema@tno.nl
[3] Karlsruhe Institute of Technology
klooss@mail.informatik.kit.edu
[4] Aalto University
russell.lai@aalto.fi
[5] Charles University, Faculty of Mathematics and Physics
p.yatsyna@matfyz.cuni.cz

Abstract. The knowledge soundness of an interactive proof is often proven via various "special soundness (SS)" frameworks which separate knowledge extraction into: (1) extracting accepting transcripts conforming to some structure; (2) recovering a witness from structure-conforming transcripts. However, even access-structure SS [TCC'23], one of the most general, does not handle proof systems employing probabilistic tests, a protocol design technique common in more advanced and especially succinct proof systems. The reason is that its extraction strategy decides whether a challenge is "useful" from the challenges used in previous transcripts alone, ignoring the prover messages. While predicate SS [CRYPTO'24] captures probabilistic tests, it does not capture access structures and forgoes the combinatorial nature and simplicity of SS.

We introduce *adaptive special soundness (AdSS)*, which generally captures extraction strategies that *adaptively* pick challenges for the prover taking the history of entire transcripts into account. AdSS in particular captures protocols employing probabilistic tests, formally generalises access-structure SS, and provides a simpler interface than predicate SS while retaining the same expressiveness under mild conditions. We provide knowledge extractors for AdSS protocols with optimal knowledge error κ and almost optimal tradeoffs between extraction probability and runtime.

To demonstrate the utility of AdSS, we provide a new (knowledge) soundness analysis of Kilian's succinct argument, resolving an open problem in [TCC'24]. In passing, we tighten the existing soundness error bound and unify the analyses in the strict and expected polynomial time regimes. We also provide a range of other examples, from simple ones showing how AdSS captures prior notions, to a new analysis of lattice-based Bulletproofs with unstructured challenges, which improves modularity and covers certain rings beyond $\mathbb{Z}$ compared to [TCC'23].

Table of Contents

Adaptive Special Soundness A new framework for knowledge extraction with applications to Kilian's protocol . . . 1

Thomas Attema, Michael Klooß, Russell W. F. Lai, and Pavlo Yatsyna

1 Introduction . . . 4
- 1.1 Special Soundness and its Limitations . . . 4
- 1.2 Our Contributions . . . 5
  - Adaptive Special Soundness. . . . 5
  - Composition of Extractors. . . . 6
  - Tight (Knowledge) Soundness Analysis of Kilian's Protocol. . . . 6
  - Modular Analysis of Lattice-based Bulletproofs. . . . 6
- 1.3 Related Work . . . 6
  - Special Soundness Notions Generalised by AdSS. . . . 6
  - Analysis of Kilian's Protocol and IOP-Based Arguments. . . . 7
  - Analyses of Lattice-based Proof Systems. . . . 8
- 1.4 Organisation . . . 8

2 Technical Overview . . . 9
- 2.1 Adaptive Special Soundness . . . 9
  - Useful Challenge Structures (UCS). . . . 9
  - Adaptive Special Soundness (AdSS). . . . 10
  - Examples. . . . 10
  - Almost Optimal Extractor. . . . 10
- 2.2 How to use $\mathcal{U}$-AdSS . . . 10
- 2.3 An Illustrative Example: Amortisation . . . 11
- 2.4 Main Application - Kilian's Succinct Argument . . . 12
- 2.5 Nested Composition of $\mathcal{U}$-AdSS Extractors . . . 12
- 2.6 Future Work and Open Questions . . . 13

3 Preliminaries . . . 13
- 3.1 Computation . . . 13
- 3.2 Interactive Proofs . . . 14
- 3.3 Interactive Reductions of Knowledge . . . 16

4 Adaptive Special Soundness . . . 16
- 4.1 Base Case . . . 17
- 4.2 Nested Composition . . . 18

5 Analysis of Kilian's Succinct Argument . . . 20
  - The Protocol. . . . 21
  - (Knowledge) Soundness Analysis. . . . 21
  - UCS for Kilian. . . . 21

References . . . 23

A Conjunctive Composition of UCS . . . 26

B Examples of UCS . . . 27
- B.1 Transcript-Agnostic Examples . . . 27
- B.2 A Basic Lattice Example . . . 29

C Knowledge Extractors for AdSS Protocols . . . 30
- C.1 Abstract Adversaries and Extractors . . . 30
- C.2 Notation and Conventions . . . 31
- C.3 Extractor Construction . . . 32
- C.4 Auxiliary Lemmata . . . 33
- C.5 Two Chain Rules . . . 34
- C.6 Analysis of the Extractor . . . 35

D AND-Composition of Abstract Extractors . . . 41

E Nested Composition of Abstract Extractors . . . 43

E.1 Nested Composition . . . 43
E.2 Additional Preliminaries . . . 44
E.3 Generic Success Bounds for Nested Composition . . . 44
E.4 Multi-round protocols, Traditional and Nested Composition . . . 45
E.5 Nested Composition of UCS extractors . . . 46
F Examples of Nested Composition of Extractors . . . 48
F.1 (Tree-)Special Soundness . . . 48
F.2 Self-Composition with 2-Special Soundness . . . 49
F.3 Coordinate-wise k-special soundness (CWSS) from nested composition . . . 49
G Lattice-based Bulletproofs . . . 50
G.1 Overview . . . 50
Existing Results. . . . 50
Our Goal. . . . 51
Tree Representation of Transcripts. . . . 51
Constructing the UCS. . . . 52
$\mathcal{U}$-Adaptive Special Soundness Analysis. . . . 53
G.2 Algebraic Number Theory Background . . . 53
G.3 Protocol Description . . . 54
G.4 UCS and Witness Extractor for Lattice Bulletproofs . . . 55
Goal. . . . 55
UCS and Witness Extractor. . . . 56
G.5 Knowledge Soundness Analysis . . . 58
H Multilinear Composite Schwartz-Zippel over Rings . . . 66
H.1 Main Theorem . . . 66
H.2 Proof of Inductive Step . . . 68
H.3 Coset-Counting Constants . . . 70
H.4 Inverse Bound . . . 73
H.5 On the Ratio of Disproportionately Long Elements . . . 75
I Notions Related to Adaptive Special Soundness . . . 77
I.1 Almost Special Soundness . . . 77
I.2 Predicate Special Soundness . . . 78
I.3 Other Notions Related to AdSS . . . 80

1 Introduction

Interactive proof systems are ubiquitous cryptographic primitives allowing a prover to convince a verifier about the veracity of mathematical statements. The soundness of a proof system asserts that no malicious prover can convince an honest verifier of a false statement, except with a small probability. A stronger variant, known as knowledge soundness (KS), postulates the existence of a knowledge extractor, which can *efficiently obtain a witness* when given oracle access to any convincing prover. Interactive proofs that are knowledge sound are called *proofs of knowledge* (PoKs). They are important in cases where a witness trivially exists (but is presumably hard to find), e.g. for proving knowledge of a discrete logarithm (DL) or a short integer solution (SIS).

The design and analysis of knowledge extractors from scratch is often challenging, even for simple and natural proof systems. Therefore frameworks (such as special soundness) are employed, and a general research direction is to improve expressiveness and guarantees of such framework. This work presents adaptive special soundness (AdSS), a new framework which is expressive enough to capture a wide range of extraction strategies. Indeed, AdSS seems to handle all applications of comparable frameworks.

1.1 Special Soundness and its Limitations

Special Soundness. A tested framework for knowledge extractors is via notions called special soundness (SS) and its various generalisations. In such a framework, the knowledge extractor consists of two parts:

1. A (transcript) extractor[6] $\mathcal{E}$ which aims to find a list of accepting transcripts (i.e. messages exchanged between prover and verifier) given black-box (rewinding) access to a potentially malicious prover.
2. A witness extractor $\mathcal{W}$ which when given a list of transcripts satisfying some desirable properties is guaranteed to output a valid witness.

The property stating that a witness extractor $\mathcal{W}$ exists is typically called X-SS, where X depends on the properties of the transcripts that are desired by $\mathcal{W}$. For instance, the classic 2-SS for public-coin 3-move protocols (a.k.a. Σ-protocols) refers to the following property: Given $k = 2$ accepting transcripts (a, c, z) and (a, c', z') sharing the same first prover message a but having distinct challenges $c \neq c'$, $\mathcal{W}$ is guaranteed to output a witness. The property is naturally generalisable to k-special soundness for any $k \geq 2$, requiring k distinct challenges. It has been generalised to arbitrary access structures in [AFR23], which we refer to as "access structure special soundness", stylised as Γ-SS in [AFR23] for an access structure Γ, which e.g. relaxes the distinctness requirement on challenges. Analysis of multi-round protocols uses $(k, k, \ldots, k)$-SS a.k.a. tree-SS [BCCGP16] (or a Γ-SS variant [AFR23]), where the latter considers the situation of having a k-ary tree of accepting transcripts with nodes labelled by prover messages and edges by distinct challenges.

The appeal of the special soundness framework is that, once it is proven that any X-SS proof system is also knowledge sound, then designers of proof systems can focus on proving that their constructions are X-SS, which is often a much simpler task. Moreover, various additional results allow common variations of proofs systems to be handled in a modular manner. For instance, sequential or parallel compositions, and non-interactive variants obtained via the Fiat-Shamir transformation. As before, a theorem proving that any X-SS proof system behaves well under any of these variations can be reused to automatically obtain security proofs of the corresponding variations of any X-SS proof system.

Limitations: Capturing Probabilistic Tests. A major limitation of access-structure SS [AFR23] (and all its specialisations) is that it does not handle proof systems which employ probabilistic tests well. This is unfortunate since probabilistic tests commonly occur in more advanced and especially succinct proof systems. While more recent generalisations such as almost [BF23] and predicate [AABKT24] special soundness also aim to tackle this problem, they come with other limitations (cf. Sections 1.3 and I). To illustrate the limitation in more detail, consider following proof system template:

1. Suppose that the witness wit is committed in a binding commitment and a separate PoK is used to prove the well-formedness of the commitment.
2. The verifier sends as a challenge randomness r for a probabilistic test Test_r (e.g. the random projection tests in [CGKR22; GHL22; BS23] for checking whether the witness is a short vector).

[6] Since in this work we will mostly be dealing with transcript extractors, we simply call them "extractors" when there is no ambiguity.

3. The prover returns $\mathsf{Test}_r(\mathsf{wit})$ and the verifier checks that $\mathsf{Test}_r(\mathsf{wit})$ is of "good quality" (where good depends on the context).
4. The prover and the verifier engage in a sound (but not knowledge sound) interactive proof with the verifier to prove the well-formedness of $\mathsf{Test}_r(\mathsf{wit})$.

Let us sketch an (ad-hoc) construction of a knowledge extraction for the above proof system. First, suppose it runs the knowledge extractor of the PoK to obtain a *candidate* witness wit^*. What should it do if wit^* is not of good quality? Intuitively, we would like to argue that, since the prover passes a sound probabilistic test, the candidate witness wit^* should be of good quality. Unfortunately, the soundness of the probabilistic test requires *independent* randomness, or equivalently, an (a priori) *fixed* witness. But the witness wit^* is extracted in an (adversarially influenced) process and can be correlated with the probabilistic test.

A natural and typical solution to the above issue is to modify the knowledge extractor, such that whenever wit^* is of bad quality, the prover is run again on fresh challenges to extract another candidate witness $\mathsf{wit}^\dagger$. Intuitively, we can argue as follows: If $\mathsf{wit}^* \neq \mathsf{wit}^\dagger$, then the binding property of the commitment scheme is broken — we call this a *binding break*. If $\mathsf{wit}^* = \mathsf{wit}^\dagger$, then the randomness of the probabilistic test was independent of wit^*, and the soundness guarantee applies.

Unfortunately, the above modification cannot be captured via access-structure SS, and thus, an ad-hoc and technical analysis becomes necessary. The problem occurs, because access-structure SS decides whether a list of transcripts is desirable or not only depending on their *challenges* but not the entire transcripts. Meanwhile, the modified knowledge extractor must *examine entire transcripts* to derive a candidate witness wit^*, in order to decide whether it is of good quality.

1.2 Our Contributions

To overcome the limitations of previous notions of special soundness, we introduce a new concept: *adaptive special soundness* (AdSS). Notably, AdSS encompasses not only classical systems such as Kilian-style protocols [Kil92], but also recent succinct proof systems, including LaBRADOR [BS23] and lattice-based Bulletproofs with unstructured challenges [BF23]. Below, we provide a summary of our contributions and defer more detailed definitions and results to the technical overview (Section 2).

Adaptive Special Soundness. At a high level, we replace the access structure Γ by a function called a *useful challenge structure (UCS)* $\mathcal{U}$, which given transcripts $\tau_1, \ldots, \tau_\ell$ outputs a subset $\mathcal{U}(\tau_1, \ldots, \tau_\ell) \subseteq \mathfrak{C}$ of the challenge set. Similar to how k-SS is generalised by Γ-SS, i.e. from deciding usefulness by checking whether the challenges $c_1, \ldots, c_k$ are distinct to checking $\Gamma(c_1, \ldots, c_k)$, $\mathcal{U}$-adaptive SS further generalises by deciding whether a challenge c is in the set $\mathcal{U}(\tau_1, \ldots, \tau_\ell)$, which may depend not only on previous challenges but also entire transcripts. Analogous to the measures κ_Γ and t_Γ associated to Γ-SS [AFR23], we can define the measures $\kappa(\mathcal{U})$ and $\mathsf{depth}(\mathcal{U})$ for $\mathcal{U}$-adaptive SS which control the knowledge error and runtime of the extractor that we construct for the following theorem:

Theorem 1.1 ((Informal, cf. Theorem 4.5) AdSS $\implies$ KS). *Let Π be a 3-move public-coin interactive proof (i.e. a Σ-protocol). Suppose $\mathcal{U}$ is a "natural" useful challenge structure and Π is $\mathcal{U}$-AdSS. Then Π is knowledge sound with knowledge error $\kappa(\mathcal{U})$ and runtime overhead*[7] $\frac{\mathsf{depth}(\mathcal{U})}{1-\kappa(\mathcal{U})}$.

Our result holds for "natural" UCSs which admit certain monotonicity, bounded depth, decidability and sampleability properties (cf. Definition 4.4). AdSS formally generalises access-structure SS in the following sense: For any Γ there exists a $\mathcal{U}_\Gamma$ such that our extractor achieves the same knowledge error $\kappa_\Gamma = \kappa(\mathcal{U}_\Gamma)$ as in [AFR23], which is optimal as evidenced by a matching cheating strategy. While the extractor in [AFR23] runs the prover $t_\Gamma = \mathsf{depth}(\mathcal{U}_\Gamma)$ times in expectation, which is often optimal[8], our extractor requires at most $\frac{\mathsf{depth}(\mathcal{U}_\Gamma)}{1-\kappa(\mathcal{U}_\Gamma)}$, which is almost optimal. See the technical overview for more details.

[7] The runtime overhead roughly refers to the ratio between the runtime of the knowledge extractor and that of the adversary.

[8] Extracting from *honest* provers may need t_Γ transcripts, e.g. for k-special soundness.

Composition of Extractors. The standard approach for analysing (sequential) compositions of Σ-protocols turns out to be insufficient for composing AdSS extractors in general. The issue that arises is subtle and connected with probabilistic tests which span over multiple rounds, e.g. a multivariate Schwartz–Zippel test or sum-checks, where each round one variable is randomly chosen and fixed. To handle this, we provide an abstract notion of *nested composition* of extractors that is slightly more powerful than traditional composition yet retains (almost) the same composition guarantees for knowledge error and runtime:

Theorem 1.2 ((Informal, cf. Theorem 4.12) Nested AdSS $\implies$ KS). *Let Π be a $(2\mu+1)$-move public-coin interactive proof. Suppose $\mathcal{U} = (\mathcal{U}_1, \ldots, \mathcal{U}_\mu)$ is a sequence of "natural" useful challenge structure and Π is $\mathcal{U}$-AdSS. Then Π is knowledge sound with knowledge error $\kappa = 1 - \prod_{i=1}^{\mu}(1-\kappa(\mathcal{U}_i)) \leq \sum_{i=1}^{\mu}\kappa(\mathcal{U}_i)$ and runtime overhead $\prod_{i=1}^{\mu}\frac{\mathsf{depth}(\mathcal{U}_i)}{1-\kappa}$.*

As an additional contribution, we introduce an AND-composition of extractors, which generalise coordinate-wise special soundness (CWSS) [FMN24] for AdSS, see Sections 1.3 and 2 for more details.

Tight (Knowledge) Soundness Analysis of Kilian's Protocol. Already in 1992, Kilian demonstrated how to transform a probabilistically checkable proof (PCP) into a public-coin interactive proof [Kil92]. Due to the use of probabilistic tests (i.e. PCPs), analysing the soundness of Kilian's protocol using special soundness techniques has remained challenging and, until now, suboptimal [CDGSY24].

To close this gap, we show that Kilian's protocol satisfies $\mathcal{U}$-adaptive SS for a suitable choice of $\mathcal{U}$. This allows us to derive concrete soundness guarantees: if the underlying PCP is sound (resp. knowledge sound), then Kilian's protocol inherits ordinary (resp. knowledge) soundness. The resulting knowledge error has a matching attack (assuming "worst-case" PCP string can be efficiently constructed). Somewhat surprisingly, our analysis improves upon the best known soundness error bounds [CDGSY24] for Kilian's protocol, thereby already illustrating the practical power and generality of the adaptive special soundness framework. We refer to Section 1.3 for a detailed comparison with the analysis in [CDGSY24] and Section 5 for the details of our analysis.

Modular Analysis of Lattice-based Bulletproofs. The lattice-based Bulletproofs protocol [BLNS20; AL21; ACK21; BF23] is the lattice-analogue of the Bulletproofs protocol [BCCGP16; BBBP+18] in the discrete logarithm setting. Due to technical challenges unique to the lattice setting, earlier works [BLNS20; AL21; ACK21] based on tree-SS focused on instantiations with polynomial-size structured challenge sets and hence only achieved inverse-polynomial knowledge error. Bünz and Fisch [BF23] were the first to analyse the instantiation over $\mathbb{Z}$ with challenge set $[0, 2^\lambda)$ via a new notion that they call *almost special soundness* (AMSS). However, the definition of AMSS is highly complex and tailored for protocols built from certain deterministic commitment schemes, making it difficult to use as a general-purpose tool for analysing other proof systems.

Towards simplifying and modularising the analysis of lattice-based Bulletproofs, we show that the μ-round lattice-based Bulletproofs protocol satisfies $\mathcal{U}$-adaptive SS for a sequence $\mathcal{U} = (\mathcal{U}_1, \ldots, \mathcal{U}_\mu)$ of useful challenge structures. At a high level, our analysis abstracts out the core idea in [BF23] of viewing the lattice Bulletproofs protocol as a nested probabilistic test (more specifically a multivariate Schwartz–Zippel test), where the prover's response is delayed to the last round of the protocol. By appealing to our result on nested composition of extractors, we obtain a much simpler knowledge error bound of $\kappa \leq \mu \cdot (\frac{3}{|\mathfrak{C}|} + \varepsilon_{\mathsf{SZ}})$, where $\mathfrak{C}$ is the challenge set and $\varepsilon_{\mathsf{SZ}}$ is the failure probability of the Schwartz–Zippel test. Further, we note that our analysis covers instantiations over rings of integers of imaginary quadratic fields with class number 1, a slightly large class of rings than $\mathbb{Z}$ considered in [BF23].

1.3 Related Work

Our results connect to prior work along several dimensions, which we discuss separately below.

Special Soundness Notions Generalised by AdSS. Our AdSS notion formally generalises the notion of access-structure special soundness [AFR23] which in turn is a generalisation of k-special soundness. AdSS also captures the extraction strategies behind almost special soundness [BF23] and predicate special soundness [AABKT24], though a fully formal comparison with either is subtle; we discuss this below and

in Section I. Coordinate-wise special soundness [FMN24] (CWSS) is best viewed as a mode of extractor composition, and we define AND-composition in Section D. A visualisation of these relations is in Fig. 1. Below, and in Sections B, D and I, we explain these relations in more detail.



Fig. 1. Relations between special soundness notions. Arrows indicate inclusions (where nested and AND-composition are implicitly allowed). The dashed arrow indicates that an inclusion for a large subclass is known; the dotted arrow indicates that all known applications are captured, but no formal inclusion is known.

Access-structure special soundness [ACK21; AF22; Att23; AFR23; AFKR26]. Our extractor generalises the extractor from [AFKR26] from Γ-special soundness [AFR23] to adaptive special soundness. In terms of knowledge error and expressiveness, our results encompass those for Γ-special soundness [AFKR26] for interactive proof systems.

Almost special soundness [BF23]. Almost special soundness (AMSS) [BF23] was designed to analyse a specific class of folding protocols, and its very definition is tied to a deterministic commitment scheme. As a consequence, AMSS is not a general-purpose tool. For instance, it does not naturally apply to protocols with non-deterministic provers, and its syntax and semantics are largely incomparable to adaptive special soundness, see Section I.1 for details. AMSS also comes with no composition theorem and must be applied to a protocol as a whole. Our AdSS-based re-analysis of lattice-based Bulletproofs is simpler and more modular, leveraging our nested composition result.

Predicate special soundness [AABKT24]. The independent work [AABKT24] introduces *predicate special soundness*, which bears similarities to almost special soundness and adaptive special soundness. It considers three predicates: A challenge predicate, a property predicate and a binding predicate. Extraction is guaranteed to work if a tree of transcripts satisfies all predicates; if the binding predicate is violated, a witness for a special "binding relation" is extracted; the probability that neither occurs is the knowledge error. This setup allows analysing probabilistic tests, and [AABKT24] uses it specifically to analyse (Fiat–Shamir transformed) Labrador [BS23]. Under mild restrictions on the predicates (roughly, efficiency and monotonicity), adaptive special soundness essentially encompasses predicate special soundness (see Section I.2); in this regime, protocols analysed via predicate special soundness benefit from our improved knowledge extractors and composition results. The two notions are otherwise, in general, incomparable: predicate special soundness *tracks* failure events after the fact, whereas adaptive special soundness *steers* the extractor towards a guaranteed break from the outset. Relatedly, it is unclear whether predicate special soundness handles cut-and-choose protocols such as Kilian's argument without an increased runtime overhead and/or knowledge error.

Coordinate-wise special soundness (CWSS) [FMN24]. The notion of *coordinate-wise* special soundness provides a framework for extraction techniques that were used implicitly in prior works, e.g. [BBCP+18]. In Section D, we provide an AND composition result for extractors, which captures and generalises the underlying idea of CWSS and is, in particular, applicable to adaptive special soundness.

Analysis of Kilian's Protocol and IOP-Based Arguments. Interactive arguments based on PCPs and IOPs [Kil92; BCS16] are usually analysed in idealised models, e.g. the random oracle. These models enable so-called *straight-line* extractors, which avoid rewinding and simplify knowledge soundness analyses.

Removing such idealisations and analysing these protocols in the standard model, particularly using special soundness frameworks, has long remained challenging. The primary difficulty lies in handling

probabilistic checks without incurring a superpolynomial blow-up in the extractor's runtime. Recent work by Attema, Fehr, and Resch [AFR23] made initial progress in bridging this gap. They analysed a specific IOP-based protocol using access-structure special soundness. However, their result only yields a quasi-polynomial-time extractor.

We extend this line of work by introducing *adaptive special soundness*, further narrowing the gap between PCP/IOP-based constructions and special soundness analyses. Specifically, we show how Kilian's PCP-based protocol can be analysed in our framework, yielding a *tight* knowledge error while maintaining expected polynomial-time extraction without random oracle assumptions. Extending this methodology to general multi-round IOPs remains open.

Chiesa et al. [CDGSY24] revisited Kilian's protocol in the standard model and improved the best-known soundness error bounds. They give two separate reductions: a *strict* polynomial-time one with soundness error $\sigma_{\mathrm{PCP}} + \sigma_{\mathrm{VC}} + \varepsilon$ and runtime proportional to $1/\varepsilon$, and an *expected* polynomial-time one with the same error but runtime proportional only to $\log(1/\varepsilon)$. Here σ_{PCP} is the soundness error of the PCP, σ_{VC} the (position) binding error of the vector commitment (VC), and $0 < \varepsilon \leq 1$ arbitrary. The strict-time reduction extends to knowledge soundness, but the expected-time one does not: it splits into an *inefficient* reduction to the PCP and an efficient one to the vector commitment, and hence yields no knowledge extractor. Obtaining an expected-time knowledge soundness result with the better dependency on ε is explicitly left open in [CDGSY24].

Applying AdSS to Kilian's protocol resolves this open problem: we obtain a single expected polynomial-time knowledge extractor with soundness error $\sigma_{\mathrm{PCP}} + \sigma_{\mathrm{VC}}$, i.e. without the additive ε term, immediately implying knowledge soundness when the PCP is knowledge sound and the VC is secure. Applying Markov's inequality reintroduces an error term $0 < \varepsilon \leq 1$ and runtime proportional to $1/\varepsilon$, recovering the strict-time guarantees of [CDGSY24].

Analyses of Lattice-based Proof Systems. A large body of works (e.g. [BDLOP18; YAZX+19; BLS19; ESLL19; ALS20; BLNS20; ENS20; LNS21; AL21; ACK21; LNP22; CLM23; BF23; KLNO24]) has been devoted to studying the task of proving knowledge of a vector $\mathbf{x}$ satisfying relations of the form

$$\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q \qquad \text{and} \qquad \|\mathbf{x}\| \leq \alpha$$

or more generally bounded-norm satisfiability of polynomial equations over $\mathcal{O}_q$. In particular, the Bulletproofs protocol [BCCGP16; BBBP+18], which is a succinct argument for arithmetic circuit satisfiability in the discrete logarithm setting, has been adapted to the lattice setting by Bootle et al. [BLNS20]. Analyses of the lattice Bulletproofs protocol have been improved in subsequent works [AL21; ACK21; CLM23; BF23]. Notably, [ACK21] derives an optimal knowledge error for tree-special sound interactive proofs, matching the trivial cheating probability. All these works except [BF23] analysed soundness of (lattice) Bulletproofs via (tree-)special soundness.

Bünz and Fisch [BF23] were the first to analyse lattice Bulletproofs over a large, unstructured challenge set (in the $\mathbb{Z}$ setting), rather than the structured polynomial-size challenge sets used by the tree-special-soundness analyses above; this is what enables a negligible knowledge error. Part of this work both simplifies and generalises this strategy, extending it to the ring setting via the notion of adaptive special soundness. Concretely, our analysis differs from that of [BF23] on two axes. The first is modularity: it factors through adaptive special soundness and the nested composition theorem (Theorem 4.12) rather than through a compiler tied to a deterministic commitment scheme. The second is coverage: it also applies over the rings of integers of imaginary quadratic fields, whereas [BF23] is stated for $\mathbb{Z}$ only. Our knowledge error bound is also slightly tighter in that it does not contain the size of the transcript tree as a multiplicative factor.

1.4 Organisation

The main goal of this paper is to introduce AdSS as a new tool to analyse interactive proof systems. In Section 2, we give a detailed overview of the new AdSS notion, its implication to knowledge soundness (KS), how to use it to analyse protocols, an amortisation example, and the nested composition of its extractors. We then provide rigorous definitions and the formal AdSS-to-KS claims in Sections 3 and 4 and a semi-formal analysis of Kilian's succinct argument in Section 5. The content in these main sections should be sufficient for an argument system designer to use AdSS to analyse their protocol.

Due to space constraint, we have to defer the analysis of the extractors and their composition, as well as other examples of AdSS-based analysis, to the appendix. In Section A we define a conjunctive composition of UCSs and state its implication to KS. Section B contains yet more examples of UCS, including a simple lattice-based example which serves as a warmup to the lattice-based Bulletproofs analysis in Section G. In Sections C to E we present the analysis of the knowledge extractors of AdSS protocols and their conjunctive and nested compositions, respectively. Since nested composition is a complicated concept, we provide some examples in Section F to ease understanding. Explaining the lattice Bulletproofs protocol and its analysis requires recalling extensive background on lattices and algebraic number theory. We provide these details in the self-contained Sections G and H. Finally, in Section I, we compare AdSS to related notions such as PSS and AMSS.

2 Technical Overview

We first give in Section 2.1 a high-level overview of $\mathcal{U}$-adaptive special soundness (AdSS) parametrised by a useful challenge structure $\mathcal{U}$ and how to use it in Section 2.2. We then turn our focus to analysing Kilian's succinct argument in the AdSS framework in Section 5.

2.1 Adaptive Special Soundness

For concreteness, consider a 3-move public-coin Σ-protocol with challenge space $\mathfrak{C}$. Let $\mathcal{A}$ be an abstract adversary, i.e. $\mathcal{A}(c)$ outputs either an *accepting* transcript $\tau = (a, c, z)$ or $\perp$. We write $\mathsf{challof}(\tau) = c$ for the challenge of a transcript τ. We propose a general framework for showing that the Σ-protocol is knowledge sound via a notion that we call $\mathcal{U}$-adaptive special soundness parametrised by a *useful challenge structure (UCS)* $\mathcal{U}$.

Useful Challenge Structures (UCS). A UCS $\mathcal{U}$ is a function which inputs a sequence of transcripts $T = (\tau_1, \dots \tau_i)$ and outputs a subset $\mathcal{U}(T)$ of $\mathfrak{C}$. A UCS $\mathcal{U}$ is *useful* in the sense that there exists another algorithm, called the witness extractor $\mathcal{W}$, that is guaranteed to find a satisfying witness $\mathsf{wit} \leftarrow \mathcal{W}(T)$ if $\mathcal{U}(T) = \emptyset$, i.e. T is considered *complete* by $\mathcal{U}$ and no more challenges are needed. It effectively acts as an interface between the witness extractor $\mathcal{W}$ and a transcript extractor $\mathcal{E}$ (to be constructed):

- Given the current list of transcripts T, the set $\mathcal{U}(T)$ tells $\mathcal{E}$ which challenges (and hence which transcripts) to seek next.
- Once $\mathcal{E}$ obtains a useful transcript τ, the set of useful challenges is updated to $\mathcal{U}(T\|\tau)$, guiding the next steps of the extraction.
- When $\mathcal{U}(T) = \emptyset$, the witness extractor $\mathcal{W}$ can efficiently compute a witness from the list T, i.e. the transcript extractor $\mathcal{E}$ has completed its task.

For our analysis, we require $\mathcal{U}$ to be monotonic, i.e. $\mathcal{U}(T\|\tau) \subseteq \mathcal{U}(T)$, so that the set of useful challenges only shrinks when more transcripts are available. Intuitively, monotonicity formalises the idea that as the extractor gathers more transcripts, fewer challenges remain useful, i.e. the extractor (or rather $\mathcal{U}(T)$) is steadily narrowing down the set of challenges it still needs to complete the extraction. In this way, the extractor moves closer to its goal of collecting enough transcripts to reconstruct a witness. Moreover, we call a sequence $(\tau_1, \dots, \tau_\ell)$ a $\mathcal{U}$-chain, if $\mathsf{challof}(\tau_i) \in \mathcal{U}(\tau_1, \dots, \tau_{i-1})$ for all $i = 2, \dots, \ell$. Transcript sequences generated by our rewinding-based transcript extractor will always be $\mathcal{U}$-chains.

Two important measures of a UCS $\mathcal{U}$ are:

- The *depth* $\mathsf{depth}(\mathcal{U})$ is the length of the longest complete $\mathcal{U}$-chain.
- The *knowledge error* $\kappa(\mathcal{U}) = 1 - \min_T \frac{|\mathcal{U}(T)|}{|\mathfrak{C}|} = \max_T \frac{|\mathfrak{C} \setminus \mathcal{U}(T)|}{|\mathfrak{C}|}$ where T ranges over incomplete $\mathcal{U}$-chains, i.e. $\mathcal{U}(T) \neq \emptyset$. In other words, $\kappa(\mathcal{U})$ is the relative size of the *largest useless challenge set* $\mathfrak{C} \setminus \mathcal{U}(T)$.

Naturally, the efficiency of our transcript extractor $\mathcal{E}$ depends on the *depth* $\mathsf{depth}(\mathcal{U})$ of the function $\mathcal{U}$: Since $\mathcal{E}$ always try to extend a $\mathcal{U}$-chain $T = (\tau_1, \dots, \tau_{j-1})$ to a longer $\mathcal{U}$-chain, i.e. the challenge $c_j = \mathsf{challof}(\tau_j)$ lies in $\mathcal{U}(\tau_1, \dots, \tau_{j-1})$, until eventually $\mathcal{U}(T) = \emptyset$, $\mathsf{depth}(\mathcal{U})$ measures the length of the longest

chain of transcripts that can be generated by the transcript extractor. In short, the larger $\mathsf{depth}(\mathcal{U})$ is, the more transcripts may be needed to extract a witness.

Formally, as we explain below, this interaction between $\mathcal{U}$ and $\mathcal{E}$ is captured by the notion of $\mathcal{U}$-adaptive special soundness and the construction of $\mathcal{E}$ whose existence implies the knowledge soundness of $\mathcal{U}$-AdSS protocols.

Adaptive Special Soundness (AdSS). Let $\mathcal{R}$ be an NP-relation and $\mathcal{U}$ be a UCS. We say that a 3-move public-coin protocol Π is *$\mathcal{U}$-adaptive special soundness* if there exists a witness extraction algorithm $\mathcal{W}$, which on input stmt and a complete $\mathcal{U}$-chain $T = (\tau_1, \ldots, \tau_\ell)$ of accepting Π-transcripts outputs a witness wit for stmt. By considering chains of UCSs $\mathcal{U} = (\mathcal{U}_1, \ldots, \mathcal{U}_\mu)$, we can similarly define the notion of $\mathcal{U}$-adaptive special soundness for $(2\mu + 1)$-move public-coin proof systems.

Examples. As a simple example, we can capture k-special soundness over challenge set $\mathfrak{C}$ with a UCS $\mathcal{U}_{k\text{-}\mathsf{SS}}$, by setting

$$\mathcal{U}_{k\text{-}\mathsf{SS}}(\tau_1, \ldots, \tau_\ell) = \begin{cases} \mathfrak{C} \setminus \{\mathsf{challof}(\tau_i)\}_{i=1}^{\ell} & |\{\mathsf{challof}(\tau_i)\}_{i=1}^{\ell}| < k \\ \emptyset & |\{\mathsf{challof}(\tau_i)\}_{i=1}^{\ell}| \geq k. \end{cases}$$

We find $\mathsf{depth}(\mathcal{U}) = k$ and $\kappa(\mathcal{U}) = \frac{k-1}{|\mathfrak{C}|}$. We can similarly capture access structure special soundness and recover the same values as defined in [AFR23]. See Section B for details and more examples. In Section 2.3 we give a simple example of how AdSS can be used to improve the analysis of the standard amortisation technique of Σ-protocols, and in Section 5 an elaborated example of using AdSS to analyse Kilian's succinct argument.

Almost Optimal Extractor. We generalise the extractor from [AFKR26], which itself builds on [AF22; AFR23], to handle *useful challenge structures* rather than access structures. Conveniently, we can re-use both extractor and analysis from [AFKR26] with only minor changes, since it internally works with useful challenge sets (denoted $\mathcal{U}_\Gamma$) and it suffices to verify that the analysis extends to the setting where those sets may depend on prior accepting *transcripts* (instead of merely challenges).

Let us recall the construction. The extractor $\mathcal{E}$ operates by iteratively extending the list of transcripts T until $\mathcal{U}(T) = \emptyset$, i.e. until a witness can be efficiently extracted from T. To grow T, the extractor samples challenges from $\mathcal{U}(T)$ until it obtains the next accepting transcript. However, without additional safeguards, this process would yield an unbounded expected runtime. To address this, a coin flip is introduced: if the coin lands heads, the extractor aborts and fails. This mechanism allows us to balance success probability against expected runtime. By carefully selecting the coin's probability of returning heads, we obtain an extractor with optimal knowledge error and near-optimal expected runtime, as stated in Theorems 1.1 and 4.5. The technical details of this knowledge extractor and its analysis are provided in Section C.3.

We call our extractor *almost optimal*, since its running time is roughly $\frac{\mathsf{depth}(\mathcal{U})}{1-\kappa(\mathcal{U})}$ times the expected runtime of the adversary, and it must recover $\mathsf{depth}(\mathcal{U})$ transcripts in the worst case, hence run the adversary that often. Thus, it is only a factor $\frac{1}{1-\kappa(\mathcal{U})}$ larger than optimal. For the knowledge error, our extractor recovers the same error as [ACK21; AFR23] for k-special soundness and Γ-special soundness [AFR23], which are known to be optimal by (trivial) attack strategies.

2.2 How to use $\mathcal{U}$-AdSS

A typical setting where AdSS applies (but Γ-SS not) is a proof of knowledge for a relation $\mathcal{R}$ that (1) proves knowledge of a witness wit committed in a binding commitment, and (2) applies a σ-sound probabilistic test $\varphi(\mathsf{wit}, c)$ (for challenge $c \in \mathfrak{C}_\varphi$) that asserts that $\Phi(\mathsf{wit}) = 1$ for a predicate Φ. In more detail, let $\mathcal{R}$ be a relation such that $(\mathsf{stmt}, \mathsf{wit}) \in \mathcal{R}$ if and only if (1) $(\mathsf{stmt}, \mathsf{wit}) \in \mathcal{R}_{\mathsf{com}}$, where $\mathcal{R}_{\mathsf{com}}$ ensures wit committed by (a commitment that is part of) stmt, and (2) $(\mathsf{stmt}, \mathsf{wit}) \in \mathcal{R}_\Phi$ if and only if $\Phi(\mathsf{wit}) = 1$. The target relation is $\mathcal{R} = \mathcal{R}_{\mathsf{com}} \cap \mathcal{R}_\Phi$.

Some concrete examples for Φ and φ are:

- $\Phi(\mathsf{wit})$ may check a system of linear (or polynomial) equations and $\varphi(\mathsf{wit}, c)$ may be an aggregation/batch check. We discuss this further in Section 2.3.

– $\Phi(\mathsf{wit})$ may assert (euclidean) shortness, and $\varphi(\mathsf{wit}, c)$ may be a short random linear projection (i.e., Johnson–Lindenstrauss predicate). We discuss this further in Section B.2.

In any case, if $\Phi(\mathsf{wit}) = 1$, then $\varphi(\mathsf{wit}, c) = 1$ for all c (by correctness). However, if $\Phi(\mathsf{wit}) = 0$, then $\varphi(\mathsf{wit}, c) = 1$ for at most a σ-fraction of $c \in \mathfrak{C}_\varphi$.

To analyse the protocol, we start (in practice implicitly) by designing a UCS for $\mathcal{R}_{\mathsf{com}}$ to extract the commitment by $(\mathcal{U}_{\mathsf{com}}, \mathcal{W}_{\mathsf{com}})$-AdSS. This can be as simple as k-special soundness, or some elaborate bespoke UCS. As our next step, we extend this UCS to handle failures. The intuition is simple: If $\Phi(\mathsf{wit}') = 0$, then search for (transcripts with) challenges c such that $\varphi(\mathsf{wit}', c) = 0$. As such c do not fool the probabilistic test, an accepting transcript cannot use wit' internally, and this should yield a commitment binding break. However, the precise way this is formalised depends on the protocol. In Section 5, we demonstrate this for Kilian's succinct argument. As a rule of thumb, we arrive at the following blueprint for a UCS $\mathcal{U}$ based on the sketched approach. Given a sequence $T = (\tau_1, \ldots, \tau_i)$ of accepting transcripts, compute $\mathcal{U}(T)$ as follows:

– If T is an incomplete $\mathcal{U}_{\mathsf{com}}$-chain, return $\mathcal{U}_{\mathsf{com}}(T)$.
– If a prefix T' of $T = T' \| T''$ is a complete $\mathcal{U}_{\mathsf{com}}$-chain, then let $\mathsf{wit}' = \mathcal{W}'(T')$ and perform the following:
  • If $\Phi(\mathsf{wit}') = 1$ or $\mathsf{wit}'' = \mathcal{W}(\mathsf{pred}(T') \| T'') \neq \bot$, return $\emptyset$. (Here, $\mathsf{pred}(T')$ is the prefix of T', omitting the last transcript.)
  • Else, return
$$\big(\mathcal{U}_{\mathsf{com}}(T') \cap \{c \mid \varphi(\mathsf{wit}', c) = 0\}\big) \setminus \mathcal{U}_{\mathsf{com}}(\mathsf{pred}(T') \| T'')$$
  (This step must be adapted to the concrete protocol.)
– If T is neither of the above, it is not a $\mathcal{U}$-chain, and we set $\mathcal{U}(T) \coloneqq \emptyset$.

Clearly, $\mathcal{U}$ is monotonic and if T is a complete $\mathcal{U}$-chain then by construction, one can extract either (1) a witness wit for $\mathcal{R}$ or (2) a pair of witnesses $\mathsf{wit}', \mathsf{wit}''$ and where $\mathsf{wit}'' = \mathcal{W}(\mathsf{pred}(T') \| T'')$ used an accepting transcript τ'' in T'' with $\varphi(\mathsf{wit}', c'') = 0$. In typical cases, condition (2) implies that a binding break was found, which concludes the extraction. A typical reasoning is as follows: The witness wit' is sufficient to predict how an honest prover replies, and the "Else"-path is chosen so that the verifier would *reject* any transcript with a challenge in that path — this motivates restricting to c with $\varphi(\mathsf{wit}', c) = 0$. As a consequence, an accepting transcript (of a malicious prover) must necessarily differ from what could be predicted. And this leads to a binding break. In fact, this is a general theme for constructing UCSs: Restrict to challenges which are "unpredictable", i.e., where the extractor will learn an accepting response which it could not generate itself. The analysis of Kilian's protocol in Section 5 follows exactly this pattern.

To conclude the recipe, we would need to analyse the properties $\kappa(\mathcal{U})$ and $\mathsf{depth}(\mathcal{U})$. Fortunately, these are often easily related to $\kappa(\mathcal{U}_{\mathsf{com}})$ and $\mathsf{depth}(\mathcal{U}_{\mathsf{com}})$. Once these measures are computed, we can invoke Theorem 4.5 to obtain a knowledge extractor $\mathcal{E}$ with knowledge error $\kappa(\mathcal{U})$ and runtime overhead $\frac{\mathsf{depth}(\mathcal{U})}{1-\kappa(\mathcal{U})}$.

2.3 An Illustrative Example: Amortisation

To demonstrate the utility of our notion of adaptive special soundness (AdSS), we present a simple yet informative example. While this example does not showcase the full power of AdSS — prior techniques already achieve the optimal knowledge error — it does highlight a concrete improvement: AdSS reduces the expected runtime of the knowledge extractor by a polynomial factor.

We consider a well-known amortisation technique from Σ-protocol theory. A prover wishes to convince a verifier that a committed vector $\mathbf{x} \in \mathbb{F}^n$ satisfies m linear constraints $\langle \mathbf{a}_1, \mathbf{x} \rangle = b_1, \ldots, \langle \mathbf{a}_m, \mathbf{x} \rangle = b_m$, where $\mathbf{a}_i \in \mathbb{F}^n$ and $b_i \in \mathbb{F}$ for all i. The naive Σ-protocol for this problem incurs communication complexity linear in the number of constraints m. However, a standard amortisation technique can reduce this become independent of m, by combining the m constraints into a single one:

$$\left\langle \sum_{i=1}^{m} \mathbf{a}_i c^{i-1}, \mathbf{x} \right\rangle = \sum_{i=1}^{m} b_i c^{i-1}, \tag{1}$$

for a uniformly random challenge $c \in \mathbb{F}$ selected by the verifier.

For simplicity, let us ignore the initial message of the prover, and assume that the response is just $\mathbf{x}$ and that the verifier checks Eq. (1) directly. In practice, this check is masked by the initial message

to achieve HVZK (and extraction will require an additional transcript to recover the mask), or it is delegated to a (succinct) subprotocol for checking a single linear constraint on the committed $\mathbf{x}$.[9] Now, it is well known that this amortisation yields an m-special sound protocol: if Eq. (1) holds for m distinct values of c, then it follows that all original constraints $\langle \mathbf{a}_i, \mathbf{x} \rangle = b_i$ must hold. This implies a knowledge error of $(m-1)/|\mathbb{F}|$, which is optimal. The standard knowledge extractor for this protocol has expected runtime proportional to m, as it must collect m accepting transcripts.

In contrast, by applying our AdSS framework, it suffices to show that the protocol is $\mathcal{U}$-adaptive special sound where

$$\mathcal{U}(T) = \begin{cases} \mathfrak{C} & T = \emptyset \\ \mathfrak{C} \setminus \left\{ c \in \mathfrak{C} : \left\langle \sum_{i=1}^{m} \mathbf{a}_i c^{i-1}, \mathbf{x}^* \right\rangle = \sum_{i=1}^{m} b_i c^{i-1} \right\} & T = (\tau) \\ \emptyset & \text{otherwise} \end{cases}$$

and $\mathbf{x}^*$ denotes the candidate witness derived from τ. This leads to an extractor that requires at most $\mathsf{depth}(\mathcal{U}) = 2$ accepting transcripts. The key insight is that (1) constitutes a probabilistic test, precisely the type of setting AdSS is designed to handle efficiently. After obtaining the first accepting transcript τ, the extractor derives a candidate witness. This witness either satisfies all m constraints, i.e. $\mathcal{U}((\tau)) = \emptyset$, or it satisfies (1) for at most $m-1$ challenges c, i.e. $\kappa(\mathcal{U}) = (m-1)/|\mathbb{F}|$. Finding any second accepting transcript for a challenge c' outside this small set implies a commitment equivocation. Thus, only two transcripts suffice to extract a valid witness or break the binding property of the commitment scheme. This efficiency gain relies critically on the adaptive nature of the special soundness: the extractor can tailor its second challenge based on the observed candidate witness.

In summary, both Γ-SS and AdSS establish knowledge soundness of this protocol with the same, optimal knowledge error $(m-1)/|\mathbb{F}|$. The difference is qualitative: AdSS requires at most 2 accepting transcripts (3 if the protocol is masked for HVZK), whereas Γ-SS may need up to m, which is reflected in a better bound on the expected runtime of the extractor. In the larger context where the linear check is delegated to a subprotocol, extraction requires *trees* of transcripts, and the size of these trees grows more slowly under AdSS than under Γ-SS. While the practical impact in this particular case may be limited, the example refines the analysis of a well-established technique and offers initial insight into the advantages of AdSS.

2.4 Main Application - Kilian's Succinct Argument

As our main application of AdSS, we re-analyse Kilian's succinct argument [Kil92], which compiles a PCP into an interactive argument using a (succinct) vector commitment (VC): the prover commits to a PCP string π, and the verifier's challenge ρ determines a set of indices $I(\rho)$ at which π is opened. Following the analysis template presented in Section 2.2, we let $\mathcal{W}(T)$ output either a VC position binding break, if the VC openings in T are inconsistent, or a partial PCP string $\widetilde{\pi}$ obtained by aggregating the openings in T. In the latter case, the probabilistic test underlying $\mathcal{U}(T)$ is the PCP verifier itself, i.e. $\varphi_{\widetilde{\pi}}(\rho) = \mathsf{PCP.Vfy}(\widetilde{\pi}_{I(\rho)}, I(\rho))$. A further challenge is then useful precisely when $\varphi_{\widetilde{\pi}}(\rho) = 0$, so that any new accepting transcript must either open a previously unopened index of $\widetilde{\pi}$ or break position binding. By Theorem 4.5, the resulting knowledge extractor matches the knowledge error of the underlying PCP and improves over the best known bounds. We refer to Section 5 for the full analysis.

2.5 Nested Composition of $\mathcal{U}$-AdSS Extractors

Until now, we only considered 3-move protocols with a single challenge. However, efficient (succinct) arguments are often multi-round protocols. While sequential composition results for extractors of special sound protocols are well-known [AC20; ACK21; AF22; AFK22; AFR23; AFK23] (a.k.a. tree-special soundness) and translate to UCS, we need a slightly stronger composition theorem to fully exploit the power of UCS to choose challenges which *ensure* a certain property.

[9] This simplified setting occurs implicitly in multi-round protocols when extraction uses trees of transcripts and tree special soundness. For example, it occurs in succinct arguments without (HV)ZK. For protocols with HVZK, such as standard Σ-protocols, our notion of nested composition of extractors allows us to treat it similarly: First extract pairs of transcripts that always yield a candidate $\mathbf{x}$. Then argue about/extract from this intermediate adversary, which *does* output $\mathbf{x}$ as in our simplified setting.

Recall that, e.g. in the analysis of Kilian's succinct argument (Section 5), the way to handle a probabilistic test ϕ is to target accepting transcripts with $\phi_{\mathsf{wit}'}(c) = 0$, and therefore obtain binding breaks. In a multi-round setting, $\phi_{\mathsf{wit}'}$ may depend not only on challenge c_i of round i, but on *all* challenges $(c_1, \ldots, c_\mu)$. For instance, for μ-fold composition, we are searching for accepting challenges with $\varphi_{\mathsf{wit}'}(c_1, \ldots, c_\mu) = 0$. Indeed, for our Bulletproofs analysis, this is a critical ingredient. Unfortunately, in the "traditional" composition theorem, extractors are only aware of the i-th round challenge space $\mathfrak{C}_i$, which is insufficient.

To fill this gap, we define and analyse what we call *nested composition* of extractors. Under mild conditions, we obtain a knowledge extractor whose knowledge error and runtime are almost identical to "traditional" extractor composition for multi-round protocols. In the below we briefly outline how nested composition is defined.

The first naive idea towards nested composition is to pass the candidate wit' to the child extractor, so that it can define its useful challenges depending on wit'. Unfortunately, this is incompatible with the usual techniques of analysing the composition of extractors.[10] Fortunately, we observe that, in all of our applications, it suffices if the parent extractor chooses the *initial challenge* of its child in such a way that $\varphi_{\mathsf{wit}'}(c_1, \ldots, c_\mu) = 0$ holds — the child need not be aware of wit' for this. Thus, if the i-th round extractor chooses c_i and the initial challenge $c_{i+1}, \ldots, c_\mu$ for the $(i+1)$-th round extractor, we can still model the required extraction strategies.

In our formal analysis in Section E, we define nested composition using the language of abstract adversaries and abstract extractors — algorithms which take a challenge as input. With this, extractor and adversary now have identical interfaces, and the application of an extractor $\mathcal{E}_2$ to adversary $\mathcal{A}$ yields another adversary $\mathcal{A}_2 = \mathcal{E}_2^{\mathcal{A}}$ which outputs as its "transcript" a whole sequence of transcripts, and to which another extractor may be applied, and so on. This makes nested composition simple to state, analyse and apply, at the cost of some formalisation overhead.

2.6 Future Work and Open Questions

In this work, we presented AdSS, AND-composition and nested composition and their analysis for interactive proof systems. However, we leave the analysis of these tools in the (non-interactive) Fiat–Shamir setting to future work. As our basic extractor for AdSS is almost identical to [AFKR26], for which Fiat–Shamir security is shown, there is little doubt that this is possible. However, in [AFKR26], only traditional composition of extractors is treated, and extending this to nested and AND composition requires a careful analysis.

A second open question that is left to future work is related to compiled multi-round IOPs. While AdSS is more powerful (as seen with Kilian's argument), a straightforward multi-round extraction still suffers from superpolynomial runtime growth, analogous to [AFR23]. Overcoming this problem would finally provide expected polynomial-time extractors for knowledge soundness compiled IOPs in a modular way.

3 Preliminaries

We write $[k] = \{1, \ldots, k\}$. For tuples A, B, we write $A \preceq B$, if A is a prefix of B, and $A \prec B$ if $A \preceq B \wedge A \neq B$; we define $A \succeq B$ and $A \succ B$ analogously. For a set S, we write $S^* \coloneqq \bigcup_{i=1}^{\infty} S^i$ for the set of all finite sequences of elements of S. For a sequence $T \in S^*$, write $|T|$ for the number of S elements in T. For $\mathbf{x} \in \mathbb{Z}^n$, we denote its infinity norm by $\|\mathbf{x}\| = \max_{i \in [n]} |x_i|$. If $T = (\tau_1, \ldots, \tau_\ell)$ is a tuple, we write $T \| \tau \coloneqq (\tau_1, \ldots, \tau_\ell, \tau)$ for the concatenation of T and τ.

3.1 Computation

By convention, algorithms output $\perp$ on input elements not from the expected domain. We write $\mathsf{time}_A(A)$ or just $\mathsf{time}(A)$ to denote the time, i.e. computation steps, spent executing A. If A is probabilistic $\mathsf{time}(A)$ is a random variable (for uniformly chosen uniform random tape of A). We write $\mathsf{time}_A(B^A)$ for the

[10] At a high level, the problem is that we now consider a *family* of adversaries $\mathcal{B}_{\mathsf{wit}'} = \mathcal{E}^{\mathcal{A}}(\mathsf{wit}')$ and wit' changes during extraction. The usual "cancellation" tricks in the analysis of expected time of rewinding ("$\varepsilon \cdot \frac{1}{\varepsilon} = 1$") do not work well with this, and the proof breaks down. It does seem possible to handle this approach by slightly weakening security guarantees. While we believe it is worthwhile to study such alternative modes of composition, given the length of this work, we pursue a more standard approach.

overall time spent in executing the oracle A, i.e. not counting the algorithm B. Similarly, we write $\mathsf{queries}_{\mathsf{A}}(B^A)$ to denote the number of queries to A (from B).

We often use "efficient algorithms" to refer to algorithms which run in probabilistic (or expected) polynomial time in its input length. A relation $\mathcal{R} : \mathbb{X} \times \mathbb{W} \to \{0,1\}$ is a deterministic function which maps a statement-witness tuple $(\mathsf{stmt}, \mathsf{wit}) \in \mathbb{X} \times \mathbb{W}$ to a bit.[11] We write $\mathsf{wit} \in \mathcal{R}(\mathsf{stmt})$ if $\mathcal{R}(\mathsf{stmt}, \mathsf{wit}) = 1$. We assume that all statements $\mathsf{stmt} \in \mathbb{X}$ have the same description length which can be denoted symbolically by $|\mathsf{stmt}|$.

Throughout, we consider families of objects/algorithms $(a_{\mathsf{stmt}})_{\mathsf{stmt} \in \mathbb{X}}$ (e.g. challenge sets, adversaries and extractors) parametrised by $\mathsf{stmt} \in \mathbb{X}$. We always assume that these families are *(PPT-)uniform*, i.e. there is a PPT Turing machine M with $M(\mathsf{stmt}) = a_{\mathsf{stmt}}$.

3.2 Interactive Proofs

We recall the notions of (public-coin) interactive proofs, its completeness and its knowledge soundness. We use the widespread simplification, that it is possible to sample uniformly random challenge c from a challenge space $\mathfrak{C}$.[12] Formally, the notion of public coin protocol generalises to any challenge distribution that has an efficient sampling algorithm that is also efficiently *reverse samplable*.[13,14]

Definition 3.1 ((Public-Coin) Interactive Proofs). *Let $\mathbb{X}$ be a statement space, $\mathbb{W}$ be a witness space and $\mathfrak{C}$ be a challenge set. A $(2\mu+1)$-move (public-coin) interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ is a tuple of probabilistic polynomial time interactive algorithms with the following structure:*

- *The prover $\mathcal{P}(\mathsf{stmt}, \mathsf{wit})$ inputs a statement $\mathsf{stmt} \in \mathbb{X}$ and a witness $\mathsf{wit} \in \mathbb{W}$ and initiates the interaction with the verifier $\mathcal{V}$ with message a_0.*
- *The verifier $\mathcal{V}(\mathsf{stmt}; c)$ inputs a statement $\mathsf{stmt} \in \mathbb{X}$ and a uniformly random challenge $(c_1, \ldots, c_\mu) = c \leftarrow_\$ \mathfrak{C} = \prod_{i=1}^{\mu} \mathfrak{C}_i$. Note here that $\mu = \mu(\mathsf{stmt})$ may depend on stmt.*
- *For $\mu = \mu(\mathsf{stmt})$ rounds, upon receiving a prover's $(i-1)$-th message, $\mathcal{V}(\mathsf{stmt}; c)$ responds with the i-th challenge c_i.*
- *Define a (full) transcript τ of an interaction as $(\mathsf{stmt}; a_0, c_1, a_1, \ldots, c_\mu, a_\mu)$, and partial transcripts as prefixes. For brevity, we omit stmt when it is clear from the context.*
- *After the interaction, the verifier's decision bit is deterministically computed as $b = \mathcal{V}(\mathsf{stmt}, \tau)$. As a shorthand for transcript and (verifier) output of the protocol, we write $(\tau, b) \leftarrow \langle \mathcal{P}(\mathsf{stmt}, \mathsf{wit}), \mathcal{V}(\mathsf{stmt}) \rangle$, where τ or b are omitted if irrelevant. The prover has no output.*

A 3-move (public-coin) interactive proof is also called a Σ-protocol.

Unless specified otherwise, all of our results assume public coin IPs.

Remark 3.2. While Definition 3.1 defines public-coin interactive proofs without a setup, all our results hold with respect to an extended definition of public-coin interactive proofs with a setup.

Next, we recall completeness and knowledge soundness.

Definition 3.3 (Completeness). *An interactive proof Π is complete for relation $\mathcal{R} \subseteq \mathbb{X} \times \mathbb{W}$ with completeness error $\gamma \colon \mathbb{X} \to [0,1]$ if for any $(\mathsf{stmt}, \mathsf{wit}) \in \mathcal{R}$, it holds that $\Pr[\langle \mathcal{P}(\mathsf{stmt}, \mathsf{wit}), \mathcal{V}(\mathsf{stmt}) \rangle = 1] \geq 1 - \gamma(\mathsf{stmt})$ with probability taken over the randomness of $\mathcal{P}$ and $\mathcal{V}$.*

We define a more fine-grained notion of knowledge soundness from which the standard notion follows.

[11] More formally, a relation is a family of functions $\{\mathcal{R}_\lambda : \mathbb{X}_\lambda \times \mathbb{W}_\lambda \to \{0,1\}\}$ parametrised by some size parameter (often the security parameter) λ. To simplify notation, we keep this size parameter implicit.

[12] Formally, in *strict* poly-time with binary coins, the statistical distance may be lower-bounded by $2^{-\mathsf{poly}}$. As is commonly done, we sweep this term under the rug. In *expected* time (i.e. for our extractors), exact sampling is possible.

[13] Reverse sampling for Sample asserts that, given some $c \in \mathfrak{C}$, one can sample a uniform r with $c = \mathsf{Sample}(r)$; it extends naturally to families $\mathfrak{C}_{\mathsf{stmt}}$. As our extractors are expected time algorithms and handle expected time adversaries, we may assume that (reverse) sampling is possible *exactly* in expected time. See also Remark C.10.

[14] Our results are strong enough to handle the random tape r directly, but it is inconvenient, e.g. for challenge set $\mathfrak{C} = \mathbb{Z}_p$ to always specify a sampling algorithm (which then becomes part of the analysis).

Definition 3.4 (Relaxed Knowledge Soundness). *Let $\Pi = (\mathcal{P}, \mathcal{V})$ be an interactive proof for $\mathbb{X} \times \mathbb{W}$ and $\widetilde{\mathcal{R}} \subseteq \mathbb{X} \times \widetilde{\mathbb{W}}$ be a relation. Suppose $\mathcal{E} \colon \mathbb{X} \to \widetilde{\mathbb{W}}$ is an expected polynomial time oracle algorithm, i.e. $\mathbb{E}[\mathsf{time}(\mathcal{E}^{\mathcal{A}}(x))] \leq \mathsf{poly}(|x|)$ where oracle queries count as one step. Moreover, let $f \colon \mathbb{X} \times [0,1] \to [0,1]$ be a function. We call $\mathcal{E}$ an **extractor** for Π for relation $\widetilde{\mathcal{R}}$ with **knowledge function** f, if for all $\mathsf{stmt} \in \mathbb{X}$ and all (not necessarily efficient) $\mathcal{P}^*$ with $\varepsilon(\mathcal{P}^*, \mathsf{stmt}) = \Pr[b = 1 \mid b \leftarrow \langle \mathcal{P}^*(\mathsf{stmt}), \mathcal{V}(\mathsf{stmt}) \rangle]$, it holds that*

$$\Pr[(\mathsf{stmt}; \mathcal{E}^{\mathcal{P}^*}(\mathsf{stmt})) \in \widetilde{\mathcal{R}}] \geq f_{\mathsf{stmt}}(\epsilon(\mathcal{P}^*, \mathsf{stmt}))$$

where $f_{\mathsf{stmt}}(x) \coloneqq f(\mathsf{stmt}, x)$.

*We say Π is **knowledge sound** with knowledge error $\kappa \colon \mathbb{X} \to [0,1]$, if there exists an extractor $\mathcal{E}$ and a positive polynomial q, such that $\mathcal{E}$ is an extractor (for relation $\widetilde{\mathcal{R}} \coloneqq \mathcal{R}$) for Π with knowledge function $f(\mathsf{stmt}, x) = \frac{x - \kappa(|\mathsf{stmt}|)}{q(|\mathsf{stmt}|)}$, i.e. if*

$$\Pr[(\mathsf{stmt}; \mathcal{E}^{\mathcal{P}^*}(\mathsf{stmt})) \in \widetilde{\mathcal{R}}] \geq \frac{\varepsilon(\mathcal{P}^*, \mathsf{stmt}) - \kappa(\mathsf{stmt})}{q(|\mathsf{stmt}|)}$$

where $\varepsilon(\mathcal{P}^*, \mathsf{stmt})$ is defined as above.

Example 3.5 (Two knowledge functions). Let $\kappa \colon \mathbb{X} \to [0,1]$ be a knowledge error. We call knowledge functions of the form $f_{\mathsf{stmt}}(\varepsilon) = \frac{\varepsilon - \kappa(\mathsf{stmt})}{1 - \kappa(\mathsf{stmt})}$ **optimal-type**. Note that $f_{\mathsf{stmt}}(\varepsilon) > 0$ if $\varepsilon > \kappa(\mathsf{stmt})$ and $f_{\mathsf{stmt}}(1) = 1$, which best possible given only a knowledge error to specify f. Similarly, we call $f_{\mathsf{stmt}}(\varepsilon) = \varepsilon - \kappa(\mathsf{stmt})$ **additive-type** (that is, denominator is 1). The naming reflects how the two types behave under composition (cf. Section E).

The common approach for measuring the (expected) runtime of a knowledge extractor (cf. Definition 3.4), is to count the (expected) number of oracle calls it makes to the prover $\mathcal{P}^*$ as a single step. However, for extractors constructed via (nested) composition, where an intermediate extractors acts as an oracle for other extractors, the expected runtime of an oracle (i.e. another knowledge extractor) may depend on the specific challenge at which it is queried. To address this, we adopt the notion of *runtime overhead* (adapted from [Klo21; Klo23]) which generalises the count of oracle queries by incorporating the dependency of runtime on the queried challenge. It is well-known and used in various forms, especially when viewed as a kind of Rényi divergence.

Definition 3.6 (Runtime Overhead). *Let $\mathcal{E}$ be an extractor. We say that $\mathcal{E}$ has **runtime overhead** ρ, if for every algorithm $\mathcal{A}$, we have*

$$\mathbb{E}[\mathsf{time}_{\mathcal{A}}(\mathcal{E}^{\mathcal{A}}(C))] \leq \rho \cdot \mathbb{E}[\mathsf{time}_{\mathcal{A}}(\mathcal{A}(C))] \,.$$

More generally[15], if $\mathcal{E}$ has oracle access to another oracle-algorithm, then we say that $\mathcal{E}$ has runtime overhead ρ (w.r.t. $\mathcal{A}$), if for every oracle-algorithm $\mathcal{B}$ and algorithm $\mathcal{A}$, we have

$$\mathbb{E}[\mathsf{time}_{\mathcal{A}}(\mathcal{E}^{\mathcal{B}^{\mathcal{A}}}(C))] \leq \rho \cdot \mathbb{E}[\mathsf{time}_{\mathcal{A}}(\mathcal{B}^{\mathcal{A}}(C))] \,.$$

For the convenience of defining adaptive special soundness in Section 4 and analysing its knowledge extractors in Sections C and E, we introduce various shorthands for accepting transcripts and the challenges used to produce them.

Definition 3.7 (Transcripts). *Let Π be an $(2\mu + 1)$-move interactive proof with statement space $\mathbb{X}$ and challenge space $\mathfrak{C} = \prod_{i=1}^{\mu} \mathfrak{C}_i$, where $\mathcal{C}_i$ denotes the i-th challenge space.*

- ***(Set of) accepting transcripts** ($\mathcal{T}_{\Pi,\mathsf{stmt}}, \mathcal{T}_{\Pi}$): For any statement $\mathsf{stmt} \in \mathbb{X}$, write $\mathcal{T}_{\Pi,\mathsf{stmt}}$ for the set of all accepting transcripts and $\mathcal{T}_{\Pi} \coloneqq \bigcup_{\mathsf{stmt} \in \mathbb{X}} \mathcal{T}_{\Pi,\mathsf{stmt}}$ for the union of all accepting transcripts over all statements. In what follows, we consider only accepting transcripts.*
- ***Challenge of transcript** $\mathsf{challof}_S(\tau)$: For a (partial) transcript $\tau = (\mathsf{stmt}; a_0, c_1, \ldots, a_\mu)$, we let $\mathsf{challof}_S(\tau) = (c_i)_{i \in S}$. As shorthands, we define $\mathsf{challof}(\tau) = \mathsf{challof}_{[\mu]}(\tau) = c$, and $\mathsf{challof}_i = \mathsf{challof}_{\{i\}}(\tau) = c_i$.*

[15] The claim without $\mathcal{B}$ usually implies the one with $\mathcal{B}$, with the same proof.

$\underline{\mathsf{AdSS}_{\Pi,\mathcal{U},\mathcal{W},\mathcal{E}}(\mathsf{stmt})}$

$T \leftarrow \mathcal{E}^{\mathcal{U}}(\mathsf{stmt})$
assert $\mathsf{Complete}_{\Pi,\mathcal{U}}(\mathsf{stmt}, T) = 1$
return $\mathcal{W}(\mathsf{stmt}, T) \notin \mathcal{R}(\mathsf{stmt})$

$\underline{\mathsf{Complete}_{\Pi,\mathcal{U}}(\mathsf{stmt}, T)}$

assert $\mathcal{U}(\mathsf{stmt}, T) = \emptyset$
assert $\mathsf{Chain}_{\Pi,\mathcal{U}}(\mathsf{stmt}, T)$
return $\top$

$\underline{\mathsf{Chain}_{\Pi,\mathcal{U}}(\mathsf{stmt}, T)}$

assert $T \in \mathcal{T}^{*}_{\Pi,\mathsf{stmt}}$
$\ell := |T|$
parse $T = (\tau_1, \ldots, \tau_\ell)$
for $k \in [\ell]$
  assert $\mathsf{challof}(\tau_k) \in \mathcal{U}(\mathsf{stmt}, (\tau_1, \ldots, \tau_{k-1}))$
return $\top$

Fig. 2. Adaptive special soundness experiment and subroutines.

- ***Predecessor and last of sequence of transcripts*** $\mathsf{pred}, \mathsf{last}$**.** *Let $T = (\tau_1, \ldots, \tau_\ell)$ be a sequence of transcripts. We write $\mathsf{pred}(T) = (\tau_1, \ldots, \tau_{\ell-1})$ and $\mathsf{last}(T) = \tau_\ell$ for the predecessor and last of T, respectively. For the empty sequence, we define $\mathsf{pred}(\emptyset) = \mathsf{last}(\emptyset) = \emptyset$.*
- ***i-nested transcripts*** $\tau^{(i)} \in \mathcal{T}^{(i)}_{\Pi,\mathsf{stmt}}$*: A 0-nested transcript is simply a transcript. The set of 0-nested transcripts is denoted $\mathcal{T}^{(0)}_{\Pi,\mathsf{stmt}} = \mathcal{T}_{\Pi,\mathsf{stmt}}$. For any $i \in [\mu]$, an i-nested transcript is a sequence of $(i-1)$-nested transcripts such that all transcripts in $\mathsf{flatten}(\tau^{(i)})$ is constant under $\mathsf{challof}_{[\mu-i]}$. That is, $\tau^{(i)} = (\tau_1^{(i-1)}, \ldots, \tau_\ell^{(i-1)})$ where*

$$\mathsf{challof}_{[\mu-i]}(\tau^{(i)}) := \mathsf{challof}_{[\mu-i]}(\tau) = \mathsf{challof}_{[\mu-i]}(\tau')$$

  for all $\tau, \tau' \in \mathsf{flatten}(\tau^{(i)})$. The set of i-nested transcripts is denoted $\mathcal{T}^{(i)}_{\Pi,\mathsf{stmt}} \subseteq (\mathcal{T}^{(i-1)}_{\Pi,\mathsf{stmt}})^*$. For each $i \in \{0\} \cup [\mu]$, define $\mathcal{T}^{(i)}_{\Pi} := \bigcup_{\mathsf{stmt} \in \mathbb{X}} \mathcal{T}^{(i)}_{\Pi,\mathsf{stmt}}$.

3.3 Interactive Reductions of Knowledge

For completeness, we relate (public-coin) interactive proofs to the interactive reductions (of knowledge) framework [KP23]. We deviate from the original framework to simplify and streamline definitions.

Definition 3.8 (Reduction of knowledge (RoK)). *A interactive reduction from $\mathcal{R}_1$ to $\mathcal{R}_2$ is a modified (public-coin) interactive proof Π, where the final output of the algorithms is redefined as follows: For inputs $\mathsf{stmt}_1 \in \mathbb{X}_1$ and $(\mathsf{stmt}_1, \mathsf{wit}_1) \in \mathcal{R}_1$, we let $(\tau, \mathsf{wit}_2, \mathsf{stmt}_2) \leftarrow \langle \mathcal{P}(\mathsf{stmt}_1, \mathsf{wit}_1), \mathcal{V}(\mathsf{stmt}_1; c)\rangle$ denote the transcript, prover output, and verifier output, respectively.*

For an interactive reduction Π, we define the induced interactive proof $\mathsf{IP}[\Pi]$ by running Π except that

- *the prover attaches wit_2 to its final message and outputs nothing;*
- *the verifier computes $\mathsf{stmt}_2 = \mathcal{V}(\mathsf{stmt}_1, \tau)$ and outputs $b = [(\mathsf{stmt}_2, \mathsf{wit}_2) \in \mathcal{R}_2]$.*

We say Π has a property (P) if $\mathsf{IP}[\Pi]$ has property (P), In particular, Π is knowledge sound with knowledge error κ (a.k.a. a reduction of knowledge) if $\mathsf{IP}[\Pi]$ is knowledge sound with the same error.

4 Adaptive Special Soundness

We introduce the notion of $(\mathcal{U}, \mathcal{W})$-adaptive special soundness (AdSS) for an interactive proof system Π over a challenge set $\mathfrak{C}$ for a relation $\mathcal{R} : \mathbb{X} \times \mathbb{W} \to \{0, 1\}$. On the one hand, AdSS generalises the access-structure special soundness notion introduced in [AFR23]. On the other hand, in subsequent sections, we will show that, similar to access-structure special soundness, AdSS also implies knowledge soundness via a knowledge extractor with optimal success probability.

4.1 Base Case

We first describe the base case of $(\mathcal{U}, \mathcal{W})$-AdSS which is likely sufficient for analysing 3-move interactive proofs. The parameters $(\mathcal{U}, \mathcal{W})$ defined relative to an interactive proof system Π are algorithms with the following semantics:

- $\mathcal{U} : \mathbb{X} \times \mathcal{T}_\Pi \to 2^{\mathfrak{C}}$ is an algorithm, called the useful challenge structure (UCS) associated to $\mathcal{W}$, that maps accepting transcripts of Π to subsets of the challenge set of Π.
- $\mathcal{W} : \mathbb{X} \times \mathcal{T}_\Pi \to \mathbb{W}$ is an algorithm, called the witness extractor, that maps a statement $\mathsf{stmt} \in \mathbb{X}$ and a sequence of transcripts T of Π to a witness $\mathsf{wit} \in \mathbb{W}$ or a failure symbol $\perp$.

Relative to a UCS $\mathcal{U}$, we can define the notion of a chain of transcripts, and further whether a chain is complete or incomplete.

Definition 4.1 (Chains of Transcripts and their Completeness.). *Let Π be a public-coin interactive proof system with a challenge set $\mathfrak{C}$. Let $\mathcal{U} : \mathbb{X} \times \mathcal{T}_\Pi \to 2^{\mathfrak{C}}$ be a deterministic polynomial time algorithm.*

- ***$\mathcal{U}$-chain*** *$T \in \mathsf{Chains}_{\Pi,\mathsf{stmt}}(\mathcal{U})$. A sequence of transcripts $T = (\tau_1, \ldots, \tau_k) \in \mathcal{T}_{\Pi,\mathsf{stmt}}$ is a **$\mathcal{U}$-chain**, denoted $T \in \mathsf{Chains}_{\Pi,\mathsf{stmt}}(\mathcal{U})$, if $\mathsf{Chain}_{\Pi,\mathcal{U}}(\mathsf{stmt}, T) = 1$ where $\mathsf{Chain}_{\Pi,\mathcal{U}}$ is defined in Fig. 2.*
- ***Incomplete $\mathcal{U}$-chain*** *$T \in \mathsf{IncompChains}_{\Pi,\mathsf{stmt}}(\mathcal{U})$. A $\mathcal{U}$-chain $T \in \mathcal{T}_{\Pi,\mathsf{stmt}}$ is complete if $\mathsf{Complete}_{\Pi,\mathcal{U}}(\mathsf{stmt}, T) = 1$, and incomplete otherwise, denoted $T \in \mathsf{IncompChains}_{\Pi,\mathsf{stmt}}(\mathcal{U})$.*
- ***V-respecting $\mathcal{U}$-chain.*** *For a predicate $\mathsf{V} \colon \mathcal{T}_\Pi \to \{0,1\}$, a $\mathcal{U}$-chain $T = (\tau_1, \ldots, \tau_k)$ is **V-respecting** if every transcript τ_i in T satisfies V.*

In words, a $\mathcal{U}$-chain T is a sequence of transcripts such that for each prefix $T' \preceq T$, the challenge of the last transcript in T' is in the useful set of the preceding transcripts, i.e. $\mathsf{challof}(\mathsf{last}(T')) \in \mathcal{U}(\mathsf{stmt}, \mathsf{pred}(T'))$. Intuitively, $\mathcal{U}$ and $\mathcal{W}$ should be thought of as a pair of complementary algorithms that work together to extract the witness of a statement from a sequence of transcripts, where the task of extracting transcripts from a prover $\mathcal{P}^*$ is further outsourced to a transcript extractor $\mathcal{E}$. Given a $\mathcal{U}$-chain T, $\mathcal{U}(T)$ are the challenges that $\mathcal{W}$ wishes to interrogate the $\mathcal{P}^*$ on given the currently collected list of transcripts T, i.e. $\mathcal{U}(T)$ are challenges considered "useful" by $\mathcal{W}$. A $\mathcal{U}$-chain T is complete when there are no more useful challenges, i.e. $\mathcal{U}(\mathsf{stmt}, T) = \emptyset$, or in other words T is already sufficient for $\mathcal{W}$ to extract the witness. We formalise this intuition as the notion of adaptive special soundness. An interactive proof system Π is said to be $(\mathcal{U}, \mathcal{W})$-AdSS if the witness extractor $\mathcal{W}$, on input a statement stmt and transcripts T produced according to the specification of the UCS $\mathcal{U}$, i.e. a complete $\mathcal{U}$-chain, produces a valid witness wit of the given statement stmt.

Definition 4.2 (Adaptive Special Soundness). *Let $\mathcal{R} : \mathbb{X} \times \mathbb{W} \to \{0,1\}$ be a relation. Let Π be a public-coin interactive proof system with a challenge set $\mathfrak{C}$. Let $\mathcal{U} : \mathbb{X} \times \mathcal{T}_\Pi \to 2^{\mathfrak{C}}$ and $\mathcal{W} : \mathbb{X} \times \mathcal{T}_\Pi \to \mathbb{W}$ be deterministic polynomial time algorithms. We say that Π is $(\mathcal{U}, \mathcal{W})$-adaptive special sound for $\mathcal{R}$ if for all statements $\mathsf{stmt} \in \mathbb{X}$ and for all probabilistic extractors $\mathcal{E}$ it holds that $\Pr[\mathsf{AdSS}_{\Pi,\mathcal{U},\mathcal{W},\mathcal{E}}(\mathsf{stmt}) = 1] = 0$ with probability taken over the randomness of $\mathcal{E}$, where the experiment $\mathsf{AdSS}_{\Pi,\mathcal{U},\mathcal{W},\mathcal{E}}$ is defined in Fig. 2.*[16] *In this case, we also say that $\mathcal{U}$ is the useful challenge structure associated with the witness extractor $\mathcal{W}$. We say that Π is $\mathcal{U}$-adaptive special sound for $\mathcal{R}$ if it is $(\mathcal{U}, \mathcal{W})$-adaptive special sound for some witness extractor $\mathcal{W}$.*

Note that the notion of a UCS $\mathcal{U}$ does not describe explicitly how to produce the transcripts T satisfying $\mathsf{Complete}_{\Pi,\mathcal{U}}(\mathsf{stmt}, T) = 1$. Rather, the purpose of the UCS $\mathcal{U}$ is to guide a transcript extractor $\mathcal{E}_\mathcal{U}$, to be constructed in the proof of Theorem 4.5 below, to produce such transcripts. By composing $\mathcal{E}_\mathcal{U}$ and $\mathcal{W}$, we obtain a knowledge extractor $\mathcal{E}_\Pi$ of Π, proving the knowledge soundness of the latter. The quality of the knowledge extractor $\mathcal{E}_\Pi$ will depend on the measures $\kappa(\mathcal{U})$ and $\mathsf{depth}(\mathcal{U})$, which we define below.

Definition 4.3 (Measures of UCS). *Let $\mathbb{X}$ be a set of statements. Let Π be a public-coin interactive proof system with a challenge set $\mathfrak{C}$. Let $\mathcal{U} : \mathbb{X} \times \mathcal{T}_\Pi \to 2^{\mathfrak{C}}$ be a deterministic polynomial time algorithms. Define the measures*

$$\kappa(\mathcal{U}) := \max_{\mathsf{stmt}, T \in \mathsf{IncompChains}_{\Pi,\mathsf{stmt}}(\mathcal{U})} \Pr[c \notin \mathcal{U}(\mathsf{stmt}, T) \mid c \leftarrow_\$ \mathfrak{C}],$$

[16] Equivalently, we say that Π is $(\mathcal{U}, \mathcal{W})$-adaptive special sound for $\mathcal{R}$ if for all statement $\mathsf{stmt} \in \mathbb{X}$ and for all sequence of accepting transcripts $T \in \mathcal{T}^*_{\Pi,\mathsf{stmt}}$ satisfying $\mathsf{Complete}_{\Pi,\mathcal{U}}(\mathsf{stmt}, T) = 1$, it holds that $\mathcal{W}(\mathsf{stmt}, T) \in \mathcal{R}(\mathsf{stmt})$, where $\mathsf{Complete}_{\Pi,\mathcal{U}}$ is defined in Fig. 2.

$$\mathsf{depth}(\mathcal{U}) := \max_{\mathsf{stmt}, T \in \mathsf{Chains}_{\Pi,\mathsf{stmt}}(\mathcal{U})} |T|$$

where $\mathsf{IncompChains}_{\Pi,\mathsf{stmt}}(\mathcal{U})$ *and* $\mathsf{Chains}_{\Pi,\mathsf{stmt}}(\mathcal{U})$ *are defined in Definition 4.1.*

In words, $\kappa(\mathcal{U})$ measures the maximum ratio between the number of "useless" challenges and the number of all possible challenges, maximised over all possible statements and all sequences of transcripts produced according to the specification of $\mathcal{U}$; $\mathsf{depth}(\mathcal{U})$ measures the maximum length of a sequence of transcripts produced according to the specification of $\mathcal{U}$ over all possible statements.

We are particularly interested in UCSs which are "natural", which means that they are monotonic, have bounded depth and output challenge subsets whose memberships are efficiently decidable and which are efficiently sampleable.

Definition 4.4 (Natural UCS). *Let* $\mathcal{R} : \mathbb{X} \times \mathbb{W} \to \{0,1\}$ *be a relation. Let* Π *be a public-coin interactive proof system with a challenge set* $\mathfrak{C}$*. Let* $\mathcal{U} : \mathbb{X} \times \mathcal{T}_\Pi \to 2^{\mathfrak{C}}$ *be a deterministic polynomial time algorithm. We say that* $\mathcal{U}$ *is natural if the following properties hold:*

- *Monotonicity: For any* $\mathsf{stmt} \in \mathbb{X}$ *and* $T \in \mathsf{Chains}_{\Pi,\mathsf{stmt}}(\mathcal{U})$*, it holds that* $\mathcal{U}(\mathsf{stmt}, T) \subseteq \mathcal{U}(\mathsf{stmt}, \mathsf{pred}(T))$*.*
- *Bounded depth: It holds that* $\mathsf{depth}(\mathcal{U}) \leq \mathsf{poly}(|\mathsf{stmt}|)$*.*
- *Efficiently decidable: There exists a polynomial time algorithm which, on input* $(\mathsf{stmt}, T, c) \in \mathbb{X} \times \mathcal{T}^*_{\Pi,\mathsf{stmt}} \times \mathfrak{C}$ *decides whether* $c \in \mathcal{U}(\mathsf{stmt}, T)$*.*
- *Efficiently sampleable: There exists a PPT algorithm which, on input* $(\mathsf{stmt}, T) \in \mathbb{X} \times \mathcal{T}^*_{\Pi,\mathsf{stmt}}$*, samples uniformly at random from* $\mathcal{U}(\mathsf{stmt}, T)$*.*

With respect to the measures $\kappa(\mathcal{U})$ and $\mathsf{depth}(\mathcal{U})$, Theorem 4.5 states that, for natural choices of $\mathcal{U}$, all $(\mathcal{U}, \mathcal{W})$-adaptive special sound protocols are also knowledge sound with an optimal knowledge extractor.

Theorem 4.5 (AdSS $\implies$ KS). *Let* $\mathcal{R} : \mathbb{X} \times \mathbb{W} \to \{0,1\}$ *be a relation. Let* Π *be a public-coin interactive proof system with a challenge set* $\mathfrak{C}$*. Let* $\mathcal{U} : \mathbb{X} \times \mathcal{T}_\Pi \to 2^{\mathfrak{C}}$ *and* $\mathcal{W} : \mathbb{X} \times \mathcal{T}_\Pi \to \mathbb{W}$ *be deterministic polynomial time algorithms. Suppose that* $\mathcal{U}$ *is natural (Definition 4.4) and* Π *is* $(\mathcal{U}, \mathcal{W})$*-adaptive special sound for* $\mathcal{R}$ *(Definition 4.2). Then* Π *is knowledge sound for* $\mathcal{R}$ *with knowledge error* $\kappa = \kappa(\mathcal{U})$*. Specifically, there exists a knowledge extractor* $\mathcal{E}$ *with runtime overhead* $\frac{\mathsf{depth}(\mathcal{U})}{1-\kappa(\mathcal{U})}$ *(Definition 4.3) such that for all statement* $\mathsf{stmt} \in \mathbb{X}$ *and all prover* $\mathcal{P}^*$ *it succeeds with probability*

$$\Pr\big((\mathsf{stmt}; \mathcal{E}^{\mathcal{P}^*}(\mathsf{stmt})) \in \mathcal{R}\big) \geq \frac{\epsilon(\mathcal{P}^*, \mathsf{stmt}) - \kappa(\mathcal{U})}{1 - \kappa(\mathcal{U})}.$$

We prove Theorem 4.5 in Section C, where it is restated for convenience.

Remark 4.6 (Per-statement knowledge error and depth). In Definition 4.3 and Theorem 4.5 we considered the knowledge error $\kappa(\mathcal{U})$ and the depth $\mathsf{depth}(\mathcal{U})$ to be functions independent of the statement stmt. We could also consider more fine-grained notions of per-statement knowledge error and depth. The statement and proof of Theorem 4.5 can be refined accordingly.

Remark 4.7 (Convention when defining UCSs $\mathcal{U}$*).* For Theorem 4.5, how $\mathcal{U}$ is defined for $T \notin \mathsf{Chains}(\mathcal{U})$ is irrelevant. We adopt the convention that $\mathcal{U}(T) = \emptyset$ for $T \notin \mathsf{Chains}(\mathcal{U})$ to conform with monotonicity.

4.2 Nested Composition

While the notion of AdSS is well-defined for interactive proofs with any number of moves, it is defined mostly for the purpose of analysing 3-move protocols. To capture more complex extraction strategies, e.g. for $(2\mu+1)$-move interactive proofs, we extend the AdSS definition to allow $\mathcal{U} = (\mathcal{U}_1, \ldots, \mathcal{U}_\mu)$ to be a sequence of UCSs, and $\mathcal{W}$ to instead take as input a μ-nested transcript. We call this notion μ-nested AdSS, which generalises e.g. the notion of $(k_1, \ldots, k_\mu)$-special soundness. Intuitively, a μ-nested $\mathcal{U} = (\mathcal{U}_1, \ldots, \mathcal{U}_\mu)$ should be seen an instruction to the transcript extractor to extract the transcripts using a nested composition of the individual extractors for $\mathcal{U}_1, \ldots, \mathcal{U}_\mu$, similar to how the transcript extractor for a $(k_1, \ldots, k_\mu)$-special sound protocol is expected to produce a tree of transcripts. As for the base case, the nested AdSS is well-defined for interactive proofs with any number of moves, but it is mostly suited for analysing $(2\mu+1)$-move protocols. To establish the terminologies, we first generalise Definition 4.1 to chains of nested transcripts.

$\underline{\mathsf{AdSS}^{\mathtt{NEST},\mu}_{\Pi,\mathcal{U},\mathcal{W},\mathcal{E}}(\mathsf{stmt}) \text{ where } \mathcal{U}=(\mathcal{U}_1,\ldots,\mathcal{U}_\mu)}$

$\tau^{(\mu)} \leftarrow \mathcal{E}^{\mathcal{U}}(\mathsf{stmt})$

assert $\mathsf{Complete}^{\mathtt{NEST},1}_{\Pi,\mathcal{U}}(\mathsf{stmt},\tau^{(\mu)})=1$

return $\mathcal{W}(\mathsf{stmt},\tau^{(\mu)}) \notin \mathcal{R}(\mathsf{stmt})$

$\underline{\mathsf{Complete}^{\mathtt{NEST},i}_{\Pi,(\mathcal{U}_i,\ldots,\mathcal{U}_\mu)}(\mathsf{stmt},\tau^{(\mu-i+1)}) \text{ where } i\in[\mu]}$

assert $\mathcal{U}_i(\mathsf{stmt},\tau^{(\mu-i+1)})=\emptyset$

assert $\mathsf{Chain}^{\mathtt{NEST},i}_{\Pi,\mathcal{U}_i}(\mathsf{stmt},\tau^{(\mu-i+1)})$

if $i<\mu$

  foreach $\tau^{(\mu-i)} \in \tau^{(\mu-i+1)}$

    assert $\mathsf{Complete}^{\mathtt{NEST},i+1}_{\Pi,(\mathcal{U}_{i+1},\ldots,\mathcal{U}_\mu)}(\mathsf{stmt},\tau^{(\mu-i)})$

return $\top$

$\underline{\mathsf{Chain}^{\mathtt{NEST},i}_{\Pi,\mathcal{U}_i}(\mathsf{stmt},\tau^{(\mu-i+1)}) \text{ where } i\in[\mu]}$

assert $\tau^{(\mu-i+1)} \in \mathcal{T}^{(\mu-i+1)}_{\Pi,\mathsf{stmt}}$

$\ell := |\tau^{(\mu-i+1)}|$

parse $\tau^{(\mu-i+1)}=(\tau_1^{(\mu-i)},\ldots,\tau_\ell^{(\mu-i)})$

// Since $\tau^{(\mu-i+1)} \in \mathcal{T}^{(\mu-i+1)}_{\Pi,\mathsf{stmt}}$, we have, for all $k\in[\ell]$, $\mathsf{challof}_{[i-1]}(\tau^{(\mu-i+1)})=\mathsf{challof}_{[i-1]}(\tau_k^{(\mu-i)})$.

for $k\in[\ell]$

  assert $\mathsf{challof}_{[\mu]\setminus[i-1]}(\tau_k^{(\mu-i)}) \in \mathcal{U}_i(\mathsf{stmt},(\tau_1^{(\mu-i)},\ldots,\tau_{k-1}^{(\mu-i)}))$

return $\top$

Fig. 3. Nested adaptive special soundness experiment and subroutines.

Definition 4.8 (Chains of Nested Transcripts and their Completeness.). *Let $\mathcal{R}:\mathbb{X}\times\mathbb{W}\to\{0,1\}$ be a relation. Let $\mu\in\mathbb{N}$ and Π be a public-coin interactive proof system with a challenge set $\mathfrak{C}\cong\mathfrak{C}_1\times\ldots\times\mathfrak{C}_\mu$. Let $\mathcal{U}=(\mathcal{U}_i)_{i=1}^{\mu}$ where $\mathcal{U}_i:\mathbb{X}\times\mathcal{T}_\Pi^{(\mu-i+1)}\to 2^{\prod_{j=i}^{\mu}\mathfrak{C}_j}$ be deterministic polynomial time algorithms.*

- *$(\mathcal{U}_i,\ldots,\mathcal{U}_\mu)$-**chain** $\tau^{(\mu-i+1)}\in\mathsf{Chains}^{(i)}_{\Pi,\mathsf{stmt}}(\mathcal{U})$. A $(\mu-i+1)$-nested transcript $\tau^{(\mu-i+1)}\in\mathcal{T}^{(\mu-i+1)}_{\Pi,\mathsf{stmt}}$ is a $(\mathcal{U}_i,\ldots,\mathcal{U}_\mu)$-**chain**, denoted $\tau^{(\mu-i+1)}\in\mathsf{Chains}^{(i)}_{\Pi,\mathsf{stmt}}(\mathcal{U})$, if $\mathsf{Chain}^{(i)}_{\Pi,\mathcal{U}_i}(\mathsf{stmt},\tau^{(\mu-i+1)})=1$ where $\mathsf{Chain}^{(i)}_{\Pi,\mathcal{U}_i}$ is defined in Fig. 3.*
- ***Incomplete** $(\mathcal{U}_i,\ldots,\mathcal{U}_\mu)$-**chain** $\tau^{(\mu-i+1)}\in\mathsf{IncompChains}^{(i)}_{\Pi,\mathsf{stmt}}(\mathcal{U})$. A $(\mathcal{U}_i,\ldots,\mathcal{U}_\mu)$-chain $\tau^{(\mu-i+1)}$ is complete if $\mathsf{Complete}^{(i)}_{\Pi,(\mathcal{U}_i,\ldots,\mathcal{U}_\mu)}(\mathsf{stmt},\tau^{(\mu-i+1)})=1$, and incomplete otherwise, denoted $\tau^{(\mu-i+1)}\in\mathsf{IncompChains}^{(i)}_{\Pi,\mathsf{stmt}}(\mathcal{U})$.*
- ***Height.** The **height** of a $(\mathcal{U}_i,\ldots,\mathcal{U}_\mu)$-chain $\tau^{(\mu-i+1)}$ is $\mu-i+1$.*
- *V-**respecting chain.** For a predicate $\mathsf{V}\colon\mathcal{T}_\Pi\to\{0,1\}$, a $(\mathcal{U}_i,\ldots,\mathcal{U}_\mu)$-chain $\tau^{(\mu-i+1)}$ is V-**respecting** if every transcript in $\tau^{(\mu-i+1)}$ satisfies V.*

Definition 4.9 (μ-Nested Adaptive Special Soundness). *Let $\mathcal{R}:\mathbb{X}\times\mathbb{W}\to\{0,1\}$ be a relation. Let $\mu\in\mathbb{N}$ and Π be a public-coin interactive proof system with a challenge set $\mathfrak{C}\cong\mathfrak{C}_1\times\ldots\times\mathfrak{C}_\mu$. Let $\mathcal{U}=(\mathcal{U}_i)_{i=1}^{\mu}$ where $\mathcal{U}_i:\mathbb{X}\times\mathcal{T}_\Pi^{(\mu-i+1)}\to 2^{\prod_{j=i}^{\mu}\mathfrak{C}_j}$ and $\mathcal{W}:\mathbb{X}\times\mathcal{T}_\Pi^{(\mu)}\to\mathbb{W}$ be deterministic polynomial time algorithms. We say that Π is $(\mathtt{NEST}[\mathcal{U}],\mathcal{W})$-adaptive special sound for $\mathcal{R}$ if for all statement $\mathsf{stmt}\in\mathbb{X}$ and for all probabilistic extractors $\mathcal{E}$ it holds that $\Pr[\mathsf{AdSS}^{\mathtt{NEST},\mu}_{\Pi,\mathcal{U},\mathcal{W},\mathcal{E}}(\mathsf{stmt})=1]=0$ with probability taken over the randomness of $\mathcal{E}$, where the experiment $\mathsf{AdSS}^{\mathtt{NEST},\mu}_{\Pi,\mathcal{U},\mathcal{W},\mathcal{E}}$ is defined in Fig. 3. In this case, we also say that $\mathcal{U}$ is the sequence of useful challenge structures associated with the witness extractor $\mathcal{W}$. We say that Π is $\mathtt{NEST}[\mathcal{U}]$-adaptive special sound for $\mathcal{R}$ if it is $(\mathtt{NEST}[\mathcal{U}],\mathcal{W})$-adaptive special sound for some witness extractor $\mathcal{W}$.*

Note that Definition 4.9 is more general than Definition 4.2 in that the latter can be recovered from the former by setting $\mu=1$. We therefore regard Definition 4.9 as the formal definition of adaptive special soundness. As for the simple case, the purpose of the nested UCS $\mathcal{U}=(\mathcal{U}_i)_{i=1}^{\mu}$ is to guide a transcript extractor $\mathcal{E}_{\mathcal{U}}$, which is a nested composition of smaller transcript extractors $(\mathcal{E}_{\mathcal{U}_i})_{i=1}^{\mu}$, to be constructed in the proof of Theorem 4.12 below, to produce such transcripts. To measure the quality of $\mathcal{E}_{\mathcal{U}}$, we extend Definition 4.4 for the measures of UCSs to the nested case.

Definition 4.10 (Measures of Nested UCS). *Let $\mathbb{X}$ be a set of statements. Let Π be a public-coin interactive proof system with a challenge set $\mathfrak{C} \cong \mathfrak{C}_1 \times \ldots \times \mathfrak{C}_\mu$. Let $\mathcal{U}_i : \mathbb{X} \times \mathcal{T}_\Pi^{(\mu-i+1)} \to 2^{\prod_{j=i}^{\mu} \mathfrak{C}_j}$ be a deterministic polynomial time algorithms for some $i \in [\mu]$. Define the measures*

$$\kappa(\mathcal{U}) := 1 - \prod_{i=1}^{\mu}(1 - \kappa^{(i)}(\mathcal{U})) \qquad \text{and} \qquad \mathsf{depth}(\mathcal{U}) := \prod_{i=1}^{\mu} \mathsf{depth}^{(i)}(\mathcal{U})$$

where

$$\kappa^{(i)}(\mathcal{U}) := \max_{\substack{\mathsf{stmt}\in\mathbb{X} \\ \tau^{(\mu-i+1)}\in\mathsf{IncompChains}^{(i)}_{\Pi,\mathsf{stmt}}(\mathcal{U})}} \Pr\Big[c \notin \mathcal{U}_i(\mathsf{stmt}, \tau^{(\mu-i+1)}) \;\Big|\; c \leftarrow_{\$} \prod_{j=i}^{\mu} \mathfrak{C}_j\Big],$$

$$\mathsf{depth}^{(i)}(\mathcal{U}) := \max_{\substack{\mathsf{stmt}\in\mathbb{X} \\ \tau^{(\mu-i+1)}\in\mathsf{Chains}^{(i)}_{\Pi,\mathsf{stmt}}(\mathcal{U})}} |\tau^{(\mu-i+1)}|$$

where $\mathsf{IncompChains}^{(i)}_{\Pi,\mathsf{stmt}}(\mathcal{U})$ and $\mathsf{Chains}^{(i)}_{\Pi,\mathsf{stmt}}(\mathcal{U})$ are defined in Definition 4.8.

Definition 4.11 (Natural Nested UCS). *Let $\mathcal{R} : \mathbb{X} \times \mathbb{W} \to \{0,1\}$ be a relation. Let $\mu \in \mathbb{N}$ and Π be a public-coin interactive proof system with a challenge set $\mathfrak{C} \cong \mathfrak{C}_1 \times \ldots \times \mathfrak{C}_\mu$. Let $\mathcal{U} = (\mathcal{U}_i)_{i=1}^{\mu}$ where $\mathcal{U}_i : \mathbb{X} \times \mathcal{T}_\Pi^{(\mu-i+1)} \to 2^{\prod_{j=i}^{\mu} \mathfrak{C}_j}$ be deterministic polynomial time algorithms. We say that $\mathcal{U}$ is natural if the following properties hold:*

- *Monotonicity: For any $\mathsf{stmt} \in \mathbb{X}$, $i \in [\mu]$ and $\tau^{(\mu-i+1)} \in \mathsf{Chains}^{(i)}_{\Pi,\mathsf{stmt}}(\mathcal{U})$, it holds that $\mathcal{U}_i(\mathsf{stmt}, \tau^{(\mu-i+1)}) \subseteq \mathcal{U}_i(\mathsf{stmt}, \mathsf{pred}(\tau^{(\mu-i+1)}))$.*
- *Bounded depth: It holds that $\mathsf{depth}(\mathcal{U}) \leq \mathsf{poly}(|\mathsf{stmt}|)$.*
- *Efficiently decidable: There exists a polynomial time algorithm which, on input $(\mathsf{stmt}, i, \tau^{(\mu-i+1)}, c) \in \mathbb{X} \times [\mu] \times \mathcal{T}_{\Pi,\mathsf{stmt}}^{(\mu-i+1)} \times \prod_{j=i}^{\mu} \mathfrak{C}_j$ decides whether $c \in \mathcal{U}_i(\mathsf{stmt}, \tau^{(\mu-i+1)})$.*
- *Efficiently sampleable: There exists a PPT algorithm which, on input $(\mathsf{stmt}, i, \tau^{(\mu-i+1)}) \in \mathbb{X} \times [\mu] \times \mathcal{T}_{\Pi,\mathsf{stmt}}^{(\mu-i+1)}$, samples uniformly at random from $\mathcal{U}_i(\mathsf{stmt}, \tau^{(\mu-i+1)})$.*

Similar to the simple case, Theorem 4.12 states that, for natural choices of $\mathcal{U} = (\mathcal{U}_i)_{i=1}^{\mu}$, all $(\mathcal{U}, \mathcal{W})$-adaptive special sound protocols are also knowledge sound with an optimal knowledge extractor. The proof is deferred to Section E.

Theorem 4.12 (μ-Nested AdSS $\implies$ KS). *Let $\mathcal{R} : \mathbb{X} \times \mathbb{W} \to \{0,1\}$ be a relation. Let Π be a public-coin interactive proof system with a challenge set $\mathfrak{C} \cong \mathfrak{C}_1 \times \ldots \times \mathfrak{C}_\mu$. Let $\mathcal{U} = (\mathcal{U}_i)_{i=1}^{\mu}$ where $\mathcal{U}_i : \mathbb{X} \times \mathcal{T}_\Pi^{(\mu-i+1)} \to 2^{\prod_{j=i}^{\mu} \mathfrak{C}_j}$ and $\mathcal{W} : \mathbb{X} \times \mathcal{T}_\Pi^{(\mu)} \to \mathbb{W}$ be deterministic polynomial time algorithms. Suppose that $\mathcal{U}$ is natural (Definition 4.11) and Π is $(\mathtt{NEST}[\mathcal{U}], \mathcal{W})$-adaptive special sound for $\mathcal{R}$ (Definition 4.9). Then Π is knowledge sound for $\mathcal{R}$ with knowledge error $\kappa = \kappa(\mathcal{U})$. Specifically, there exists a knowledge extractor $\mathcal{E}$ with runtime overhead $\frac{\mathsf{depth}(\mathcal{U})}{1-\kappa(\mathcal{U})}$ (Definition 4.10) such that for all statement $\mathsf{stmt} \in \mathbb{X}$ and all prover $\mathcal{P}^*$ it succeeds with probability*

$$\Pr\big((\mathsf{stmt}; \mathcal{E}^{\mathcal{P}^*}(\mathsf{stmt})) \in \mathcal{R}\big) \geq \frac{\epsilon(\mathcal{P}^*, \mathsf{stmt}) - \kappa(\mathcal{U})}{1 - \kappa(\mathcal{U})}.$$

5 Analysis of Kilian's Succinct Argument

We now turn to our main application of adaptive special soundness: the analysis of Kilian's succinct argument [Kil92]. Somewhat surprisingly, our framework yields a knowledge extractor that improves upon the best-known bounds for (knowledge) soundness, highlighting the potential of AdSS even in well-studied settings. Instead of presenting the analysis fully formally — which requires recalling formal definitions of probabilistically checkable proofs (PCP) and succinct vector commitments (VC) and distracts from the main point — we outline Kilian's construction and describe its AdSS-based analysis at a high yet sufficiently precise level so that a formal analysis can be derived without much difficulty.

The Protocol. Kilian's approach [Kil92] takes as input a PCP and compiles it into an interactive argument, by using a computationally position binding succinct VC scheme, such as a (hash-based) Merkle tree commitment scheme. So let PCP be a PCP for some NP-relation $\mathcal{R}$; we write $(\mathsf{stmt}, \mathsf{wit}) \in \mathcal{R}$ for statement and witness, respectively. We let q denote the query complexity, N the length, r the randomness complexity and Σ the alphabet of PCP, where q, N, r may depend on stmt. Further, let VC be a (succinct) vector commitment, e.g. a Merkle tree commitment. Then the compiled interactive argument $(\mathcal{P}, \mathcal{V}) = \mathsf{Kilian}[\mathsf{PCP}, \mathsf{VC}]$ is defined as:

- $\mathcal{P}(\mathsf{stmt}, \mathsf{wit}) \to \mathcal{V}(\mathsf{stmt})$: Prepare a PCP string $\pi \leftarrow \mathsf{PCP.Prove}(\mathsf{stmt}, \mathsf{wit})$, and commitment vc with decommitment d as $(vc, d) \leftarrow \mathsf{VC.Com}(\pi)$. Send vc.
- $\mathcal{V} \to \mathcal{P}$: Sample $\rho \leftarrow_\$ \{0,1\}^r$ uniformly and let $I = \mathsf{PCP.Query}(\mathsf{stmt}; \rho)$ be the set of q query indices $I \subseteq \{1, \dots N\}$. Send I.
- $\mathcal{P} \to \mathcal{V}$: Send $z_I = ((\pi_i)_{i \in I}, d_I)$, where d_I denotes vc decommitment information for the indices in I.
- $\mathcal{V}$: Receive $z_I = ((\pi_i)_{i \in I}, d_I)$. Accept if $\mathsf{VC.VfyOpen}(vc, (\pi_i)_{i \in I}, I, d_I) = 1$ and $\mathsf{PCP.Vfy}(\mathsf{stmt}, (\pi_i)_{i \in I}, I) = 1$. That is, the opening d_I of vc at indices I to $(\pi_i)_{i \in I}$ is valid, and the PCP answers were convincing. Else reject.

For the analysis, let $\mathcal{A}$ be an abstract deterministic malicious prover, i.e. $\mathcal{A}(I)$ outputs $\perp$ or *accepting* transcripts (vc, I, z_I), with $z_I = (\pi_I, d_I)$ and fixed vc. To declutter notation, we omit stmt and vc, and write $I(\rho) = \mathsf{PCP.Query}(\mathsf{stmt}; \rho)$. Moreover, for $\pi \in (\Sigma \cup \{\perp\})^N$, we write $\pi_I = \pi_{i \in I}$ and let $\mathsf{PCP.Vfy}(\pi_I, I)$ reject if π_I contains $\perp$.

Remark 5.1. While challenges in the original protocol are $I \subseteq \{1, \dots N\}$, we will set $\mathfrak{C} = \{0,1\}^r$ and treat the randomness ρ as the challenge. This is needed, because our results assume uniformly distributed challenges (which $I = I(\rho)$ may not be). Additionally, this showcases how to apply our techniques to non-trivial challenge distributions.

(Knowledge) Soundness Analysis. Our goal is to show that if the underlying PCP is sound (respectively, knowledge sound), then the compiled argument is computationally sound (respectively, knowledge sound). While knowledge soundness is a strictly stronger and typically more challenging property to establish, our analysis is essentially the same in both cases.

Specifically, we define the UCS $\mathcal{U}$ such that $\mathcal{U}(T) = \emptyset$ if and only if the list of transcripts T either contains a VC position binding break or if it defines a (partial) PCP string with large enough success probability. This UCS captures the fact that we aim to extract a PCP string that passes verification with sufficiently high probability. From there, we proceed as follows:

- In the case of *ordinary soundness*, this leads to a contradiction: for false statements no such PCP string exists, so the extractor must have obtained a commitment equivocation.
- In the case of *knowledge soundness*, we apply the PCP knowledge extractor to the extracted PCP string to recover a valid witness.

Assuming such a UCS exists, it follows that the protocol $(\mathcal{P}, \mathcal{V}) = \mathsf{Kilian}[\mathsf{PCP}, \mathsf{VC}]$ is $\mathcal{U}$-adaptive special sound.

Remark 5.2. We adopt the following standard notion of knowledge soundness for PCPs [Val08]. Specifically, we say that a PCP system is *strongly σ-extractable* if there exists a deterministic polynomial-time witness recovery algorithm W such that, for any partial PCP string $\widetilde{\pi}$, if $\mathsf{W}(\widetilde{\pi}) = \perp$ (i.e. no valid witness is recovered), then:

$$\Pr_{\rho}[\mathsf{PCP.Vfy}(\widetilde{\pi}_{I(\rho)}, I(\rho)) = 1] \leq \sigma\,.$$

This captures the notion that any PCP string that causes acceptance with probability greater than σ must encode a valid witness.

UCS for Kilian. We now proceed to define $\mathcal{U}$. Let $T = (\tau_1, \dots, \tau_\ell)$ be a sequence of accepting transcripts. Define $\widetilde{\pi}(T) \in (\Sigma \cup \{\perp\})^N$ as the partial PCP string derived from T, where $\widetilde{\pi}_i(T) = \perp$ if index i was never opened in any transcript of T. We distinguish two cases:

– **Case 1: Inconsistent openings.** If any index i is opened to at least 2 different values in T, then a position binding violation has occurred. We set $\mathcal{U}(T) = \emptyset$.
– **Case 2: Consistent openings.** In this case, $\widetilde{\pi}(T)$ is well-defined. A challenge ρ is considered *not useful* if the verifier accepts the partial PCP string under challenge $I(\rho)$, i.e. if $\mathsf{PCP.Vfy}(\widetilde{\pi}_{I(\rho)}(T), I(\rho)) = 1$, since we can already construct an accepting transcript for this challenge. Conversely, if the verifier rejects, this reveals new information: the challenge either opens previously unopened indices or forces the adversary to equivocate the commitment. Thus, we define:

$$\mathcal{U}(T) = \{\rho \mid \mathsf{PCP.Vfy}(\widetilde{\pi}_{I(\rho)}(T), I(\rho)) = 0\}\,.$$

Let us now analyse the key properties of $\mathcal{U}$:

– **Monotonicity.** For $T_i = (\tau_1, \ldots, \tau_i)$ and $T_{i+1} = T_i \| \tau_{i+1}$, we have $\mathcal{U}(T_{i+1}) \subseteq \mathcal{U}(T_i)$. This holds trivially if $\mathcal{U}(T_{i+1}) = \emptyset$. Otherwise, τ_{i+1} reveals new openings, which reduces the number of $\perp$ entries in $\widetilde{\pi}(\cdot)$ and hence the number of challenges that lead to rejection, so fewer challenges remain useful.
– **Depth.** We have $\mathsf{depth}(\mathcal{U}) \leq N - q + 2$. The first transcript opens q new indices, and each subsequent one opens at least one new index or reveals a position binding violation. Thus, after $N - q + 1$ transcripts all indices are open, and any further accepting transcript must violate position binding.
– **(Knowledge) Soundness Error.** Let σ be the (knowledge) soundness error of the PCP, then $\kappa(\mathcal{U}) \leq \sigma$. To see this, consider the two scenarios:
  • **Ordinary Soundness.** The adversary attempts to convince the verifier of a false statement. By the soundness of the PCP, any (partial) PCP string passes verification with probability at most σ, so at least a $(1 - \sigma)$-fraction of challenges must be useful at every stage of extraction.
  • **Knowledge Soundness.** The extractor applies the PCP witness recovery algorithm each time the partial PCP string is extended. By definition, if the partial PCP string passes verification with probability at least σ, then a witness is recovered. Thus, if the PCP witness recovery algorithm fails, at least a $(1 - \sigma)$-fraction of challenges must remain useful.[17]

We conclude that $\mathsf{Kilian}[\mathsf{PCP}, \mathsf{VC}]$ is computationally (knowledge) sound with soundness error at most σ, where σ is the (knowledge) soundness error of the PCP. More precisely, by Theorems 1.1 and 4.5, there exists a knowledge extractor that, given an adversary with success probability ε, succeeds in extracting either a witness or a commitment equivocation with probability at least $\frac{\varepsilon - \sigma}{1 - \sigma}$. Furthermore, the extractor runs in expected polynomial time, proportional to the length of the PCP string. The knowledge soundness result is summarised in the following corollary of Theorem 4.5.

Corollary 5.3 (Knowledge Soundness of Kilian). *Let* PCP *be a PCP for relation* $\mathcal{R}$ *with length* N*, query complexity* q *and knowledge error* $\sigma(\cdot)$*, i.e.* PCP *is* $\sigma(\cdot)$*-extractable. Let* $\mathcal{A}$ *be an adversary that succeeds in attacking* $\mathsf{Kilian}[\mathsf{PCP}, \mathsf{VC}]$ *on statement* stmt *with probability* $\varepsilon(\mathcal{A}, \mathsf{stmt})$*. Then, there exists an extractor that, on input a statement* stmt *and with rewindable black-box access to* $\mathcal{A}$*, making an expected number of at most* $N - q + 2$ *calls to* $\mathcal{A}$*, succeeds in outputting a witness for* stmt *or a commitment equivocation with probability at least*

$$\frac{\varepsilon(\mathcal{A}, \mathsf{stmt}) - \sigma(|\mathsf{stmt}|)}{1 - \sigma(|\mathsf{stmt}|)}\,.$$

Acknowledgments. Thomas Attema has been supported by TNO's Early Research Program - Next Generation Cryptography. The work of Russell Lai is supported by the Research Council of Finland grant 358951 and by the Future Makers Program project SPICED of the Technology Industries of Finland Centennial Foundation. Pavlo Yatsyna was supported by Charles University programme PRIMUS/24/SCI/010. This work was supported by KASTEL Security Research Labs.

[17] Formally, in the knowledge soundness case, the UCS $\mathcal{U}$ is defined slightly differently. In this case, $\mathcal{U}(T) = \emptyset$ if the PCP knowledge extractor successfully recovers a witness from the partial PCP string defined by the list of transcripts T. Otherwise, $\mathcal{U}(T)$ is defined as in the soundness case.

References

[AABKT24] Marius A. Aardal, Diego F. Aranha, Katharina Boudgoust, Sebastian Kolby, and Akira Takahashi. "Aggregating Falcon Signatures with LaBRADOR". In: *CRYPTO 2024, Part I*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14920. LNCS. Springer, Cham, Aug. 2024, pp. 71–106. DOI: 10.1007/978-3-031-68376-3_3 (cit. on pp. 4, 6, 7, 77–79).

[AC20] Thomas Attema and Ronald Cramer. "Compressed Σ-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics". In: *CRYPTO 2020, Part III*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12172. LNCS. Springer, Cham, Aug. 2020, pp. 513–543. DOI: 10.1007/978-3-030-56877-1_18 (cit. on pp. 12, 45).

[ACK21] Thomas Attema, Ronald Cramer, and Lisa Kohl. "A Compressed Σ-Protocol Theory for Lattices". In: *CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Cham, Aug. 2021, pp. 549–579. DOI: 10.1007/978-3-030-84245-1_19 (cit. on pp. 6–8, 10, 12, 45, 50, 55, 78).

[AF22] Thomas Attema and Serge Fehr. "Parallel Repetition of $(k_1, \ldots, k_\mu)$-Special-Sound Multi-round Interactive Proofs". In: *CRYPTO 2022, Part I*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13507. LNCS. Springer, Cham, Aug. 2022, pp. 415–443. DOI: 10.1007/978-3-031-15802-5_15 (cit. on pp. 7, 10, 12, 28, 30, 32).

[AFK22] Thomas Attema, Serge Fehr, and Michael Klooß. "Fiat-Shamir Transformation of Multi-round Interactive Proofs". In: *TCC 2022, Part I*. Ed. by Eike Kiltz and Vinod Vaikuntanathan. Vol. 13747. LNCS. Springer, Cham, Nov. 2022, pp. 113–142. DOI: 10.1007/978-3-031-22318-1_5 (cit. on pp. 12, 30, 77–80).

[AFK23] Thomas Attema, Serge Fehr, and Michael Klooß. "Fiat-Shamir Transformation of Multi-Round Interactive Proofs (Extended Version)". In: *Journal of Cryptology* 36.4 (Oct. 2023), p. 36. DOI: 10.1007/s00145-023-09478-y (cit. on pp. 12, 30, 80).

[AFKR26] Thomas Attema, Serge Fehr, Michael Klooß, and Nicolas Resch. "The Fiat - Shamir Transformation of $(\varGamma _1,\dots ,\varGamma _\mu)$-Special-Sound Interactive Proofs". In: *J. Cryptol.* 39.2 (2026), p. 19. DOI: 10.1007/S00145-025-09569-Y. URL: https://doi.org/10.1007/s00145-025-09569-y (cit. on pp. 7, 10, 13, 29, 30, 32, 33, 80).

[AFR23] Thomas Attema, Serge Fehr, and Nicolas Resch. "Generalized Special-Sound Interactive Proofs and Their Knowledge Soundness". In: *TCC 2023, Part III*. Ed. by Guy N. Rothblum and Hoeteck Wee. Vol. 14371. LNCS. Springer, Cham, 2023, pp. 424–454. DOI: 10.1007/978-3-031-48621-0_15 (cit. on pp. 4–8, 10, 12, 13, 16, 27–30, 32, 45, 80).

[AL21] Martin R. Albrecht and Russell W. F. Lai. "Subtractive Sets over Cyclotomic Rings - Limits of Schnorr-Like Arguments over Lattices". In: *CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Cham, Aug. 2021, pp. 519–548. DOI: 10.1007/978-3-030-84245-1_18 (cit. on pp. 6, 8, 50, 55).

[ALS20] Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. "Practical Product Proofs for Lattice Commitments". In: *CRYPTO 2020, Part II*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Cham, Aug. 2020, pp. 470–499. DOI: 10.1007/978-3-030-56880-1_17 (cit. on p. 8).

[Att23] Thomas Attema. "Compressed Σ-Protocol Theory". PhD thesis. Leiden University, 2023 (cit. on p. 7).

[BBBP+18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. "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 pp. 6, 8, 50, 55).

[BBCP+18] Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. "Sub-linear Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits". In: *CRYPTO 2018, Part II*. Ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10992. LNCS. Springer, Cham, Aug. 2018, pp. 669–699. DOI: 10.1007/978-3-319-96881-0_23 (cit. on p. 7).

[BBR12] D. Berkane, O. Bordellès, and O. Ramaré. "Explicit upper bounds for the remainder term in the divisor problem". In: *Math. Comp.* 81.278 (2012), pp. 1025–1051. ISSN: 0025-

5718,1088-6842. DOI: 10.1090/S0025-5718-2011-02535-4. URL: https://doi.org/10.1090/S0025-5718-2011-02535-4 (cit. on p. 71).

[BCCGP16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. "Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting". In: *EUROCRYPT 2016, Part II*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9666. LNCS. Springer, Berlin, Heidelberg, May 2016, pp. 327–357. DOI: 10.1007/978-3-662-49896-5_12 (cit. on pp. 4, 6, 8, 48, 50, 77).

[BCGG+17] Jonathan Bootle, Andrea Cerulli, Essam Ghadafi, Jens Groth, Mohammad Hajiabadi, and Sune K. Jakobsen. "Linear-Time Zero-Knowledge Proofs for Arithmetic Circuit Satisfiability". In: *ASIACRYPT 2017, Part III*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10626. LNCS. Springer, Cham, Dec. 2017, pp. 336–365. DOI: 10.1007/978-3-319-70700-6_12 (cit. on p. 55).

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. "Interactive Oracle Proofs". In: *TCC 2016-B, Part II*. Ed. by Martin Hirt and Adam D. Smith. Vol. 9986. LNCS. Springer, Berlin, Heidelberg, 2016, pp. 31–60. DOI: 10.1007/978-3-662-53644-5_2 (cit. on p. 7).

[BDLOP18] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. "More Efficient Commitments from Structured Lattice Assumptions". In: *SCN 18*. Ed. by Dario Catalano and Roberto De Prisco. Vol. 11035. LNCS. Springer, Cham, Sept. 2018, pp. 368–385. DOI: 10.1007/978-3-319-98113-0_20 (cit. on p. 8).

[BF06] Eva Bayer Fluckiger. "Upper bounds for Euclidean minima of algebraic number fields". In: *J. Number Theory* 121.2 (2006), pp. 305–323. ISSN: 0022-314X,1096-1658. DOI: 10.1016/j.jnt.2006.03.002. URL: https://doi.org/10.1016/j.jnt.2006.03.002 (cit. on p. 72).

[BF23] Benedikt Bünz and Ben Fisch. "Multilinear Schwartz-Zippel Mod N and Lattice-Based Succinct Arguments". In: *TCC 2023, Part III*. Ed. by Guy N. Rothblum and Hoeteck Wee. Vol. 14371. LNCS. Springer, Cham, 2023, pp. 394–423. DOI: 10.1007/978-3-031-48621-0_14 (cit. on pp. 4–8, 30, 50, 51, 53, 66, 67, 73, 75, 77, 78).

[BLNS20] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. "A Non-PCP Approach to Succinct Quantum-Safe Zero-Knowledge". In: *CRYPTO 2020, Part II*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Cham, Aug. 2020, pp. 441–469. DOI: 10.1007/978-3-030-56880-1_16 (cit. on pp. 6, 8, 50, 55).

[BLS19] Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. "Algebraic Techniques for Short(er) Exact Lattice-Based Zero-Knowledge Proofs". In: *CRYPTO 2019, Part I*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11692. LNCS. Springer, Cham, Aug. 2019, pp. 176–202. DOI: 10.1007/978-3-030-26948-7_7 (cit. on p. 8).

[BS23] Ward Beullens and Gregor Seiler. "LaBRADOR: Compact Proofs for R1CS from Module-SIS". In: *CRYPTO 2023, Part V*. Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14085. LNCS. Springer, Cham, Aug. 2023, pp. 518–548. DOI: 10.1007/978-3-031-38554-4_17 (cit. on pp. 4, 5, 7, 29, 30, 77–79).

[BS66] A. I. Borevich and I. R. Shafarevich. *Number theory*. Vol. Vol. 20. Pure and Applied Mathematics. Translated from the Russian by Newcomb Greenleaf. Academic Press, New York-London, 1966, pp. x+435 (cit. on p. 71).

[CDGSY24] Alessandro Chiesa, Marcel Dall'Agnol, Ziyi Guan, Nicholas Spooner, and Eylon Yogev. "Untangling the Security of Kilian's Protocol: Upper and Lower Bounds". In: *TCC 2024, Part I*. Ed. by Elette Boyle and Mohammad Mahmoody. Vol. 15364. LNCS. Springer, Cham, Dec. 2024, pp. 158–188. DOI: 10.1007/978-3-031-78011-0_6 (cit. on pp. 6, 8).

[CDPR16] Ronald Cramer, Léo Ducas, Chris Peikert, and Oded Regev. "Recovering Short Generators of Principal Ideals in Cyclotomic Rings". In: *EUROCRYPT 2016, Part II*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9666. LNCS. Springer, Berlin, Heidelberg, May 2016, pp. 559–585. DOI: 10.1007/978-3-662-49896-5_20 (cit. on p. 76).

[CGKR22] Geoffroy Couteau, Dahmun Goudarzi, Michael Klooß, and Michael Reichle. "Sharp: Short Relaxed Range Proofs". In: *ACM CCS 2022*. Ed. by Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi. ACM Press, Nov. 2022, pp. 609–622. DOI: 10.1145/3548606.3560628 (cit. on pp. 4, 29, 30).

[CLM23] Valerio Cini, Russell W. F. Lai, and Giulio Malavolta. "Lattice-Based Succinct Arguments from Vanishing Polynomials - (Extended Abstract)". In: *CRYPTO 2023, Part II*. Ed. by

Helena Handschuh and Anna Lysyanskaya. Vol. 14082. LNCS. Springer, Cham, Aug. 2023, pp. 72–105. DOI: 10.1007/978-3-031-38545-2_3 (cit. on p. 8).

[Dav51] H. Davenport. "On a principle of Lipschitz". In: *J. London Math. Soc.* 26 (1951), pp. 179–183. ISSN: 0024-6107. DOI: 10.1112/jlms/s1-26.3.179. URL: https://doi.org/10.1112/jlms/s1-26.3.179 (cit. on pp. 70, 77).

[EL93] G.R. Everest and J.H. Loxton. "Counting Algebraic Units with Bounded Height". In: *Journal of Number Theory* 44.2 (1993), pp. 222–227. ISSN: 0022-314X. DOI: https://doi.org/10.1006/jnth.1993.1047. URL: https://www.sciencedirect.com/science/article/pii/S0022314X83710474 (cit. on p. 76).

[ENS20] Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. "Practical Exact Proofs from Lattices: New Techniques to Exploit Fully-Splitting Rings". In: *ASIACRYPT 2020, Part II*. Ed. by Shiho Moriai and Huaxiong Wang. Vol. 12492. LNCS. Springer, Cham, Dec. 2020, pp. 259–288. DOI: 10.1007/978-3-030-64834-3_9 (cit. on p. 8).

[ESLL19] Muhammed F. Esgin, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. "Lattice-Based Zero-Knowledge Proofs: New Techniques for Shorter and Faster Constructions and Applications". In: *CRYPTO 2019, Part I*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11692. LNCS. Springer, Cham, Aug. 2019, pp. 115–146. DOI: 10.1007/978-3-030-26948-7_5 (cit. on p. 8).

[FHM20] Mikołaj Fraczyk, Gergely Harcos, and Péter Maga. "Counting Bounded Elements of a Number Field". In: *International Mathematics Research Notices* 2022.1 (June 2020), pp. 373–390. ISSN: 1073-7928. DOI: 10.1093/imrn/rnaa126. eprint: https://academic.oup.com/imrn/article-pdf/2022/1/373/42136518/rnaa126.pdf. URL: https://doi.org/10.1093/imrn/rnaa126 (cit. on pp. 70–72).

[FMN24] Giacomo Fenzi, Hossein Moghaddas, and Ngoc Khanh Nguyen. "Lattice-Based Polynomial Commitments: Towards Asymptotic and Concrete Efficiency". In: *Journal of Cryptology* 37.3 (July 2024), p. 31. DOI: 10.1007/s00145-024-09511-8 (cit. on pp. 6, 7, 43, 49, 50, 80).

[GHL22] Craig Gentry, Shai Halevi, and Vadim Lyubashevsky. "Practical Non-interactive Publicly Verifiable Secret Sharing with Thousands of Parties". In: *EUROCRYPT 2022, Part I*. Ed. by Orr Dunkelman and Stefan Dziembowski. Vol. 13275. LNCS. Springer, Cham, 2022, pp. 458–487. DOI: 10.1007/978-3-031-06944-4_16 (cit. on p. 4).

[HKR19] Max Hoffmann, Michael Klooß, and Andy Rupp. "Efficient Zero-Knowledge Arguments in the Discrete Log Setting, Revisited". In: *ACM CCS 2019*. Ed. by Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz. ACM Press, Nov. 2019, pp. 2093–2110. DOI: 10.1145/3319535.3354251 (cit. on p. 80).

[Kil92] Joe Kilian. "A Note on Efficient Zero-Knowledge Proofs and Arguments (Extended Abstract)". In: *24th ACM STOC*. ACM Press, May 1992, pp. 723–732. DOI: 10.1145/129712.129782 (cit. on pp. 5–7, 12, 20, 21).

[KLNO24] Michael Klooß, Russell W. F. Lai, Ngoc Khanh Nguyen, and Michal Osadnik. "RoK, Paper, SISsors Toolkit for Lattice-Based Succinct Arguments - (Extended Abstract)". In: *ASIACRYPT 2024, Part V*. Ed. by Kai-Min Chung and Yu Sasaki. Vol. 15488. LNCS. Springer, Singapore, Dec. 2024, pp. 203–235. DOI: 10.1007/978-981-96-0935-2_7 (cit. on pp. 8, 50).

[Klo21] Michael Klooß. "On Expected Polynomial Runtime in Cryptography". In: *TCC 2021, Part I*. Ed. by Kobbi Nissim and Brent Waters. Vol. 13042. LNCS. Springer, Cham, Nov. 2021, pp. 558–590. DOI: 10.1007/978-3-030-90459-3_19 (cit. on p. 15).

[Klo23] Michael Klooß. "On Efficient Zero-Knowledge Arguments". PhD thesis. Karlsruhe Institute of Technology, Germany, 2023. URL: https://d-nb.info/1282148052 (cit. on pp. 15, 80).

[KP23] Abhiram Kothapalli and Bryan Parno. "Algebraic Reductions of Knowledge". In: *CRYPTO 2023, Part IV*. Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14084. LNCS. Springer, Cham, Aug. 2023, pp. 669–701. DOI: 10.1007/978-3-031-38551-3_21 (cit. on p. 16).

[Lee23] Ethan S. Lee. "On the number of integral ideals in a number field". In: *Journal of Mathematical Analysis and Applications* 517.1 (2023), p. 126585. ISSN: 0022-247X. DOI: https://doi.org/10.1016/j.jmaa.2022.126585. URL: https://www.sciencedirect.com/science/article/pii/S0022247X22005996 (cit. on p. 76).

$\underline{\mathsf{AdSS}^{\mathtt{AND},n}_{\Pi,\mathcal{U},\mathcal{W},\mathcal{E}}(\mathsf{stmt}) \text{ where } \mathcal{U} = (\mathcal{U}_1,\ldots,\mathcal{U}_n)}$

$(T_1,\ldots,T_n) \leftarrow \mathcal{E}^{\mathcal{U}}(\mathsf{stmt})$

foreach $i \in [n]$

  parse $T_i = (\tau_{i,1},\ldots)$

assert $\tau_{1,1} = \tau_{2,1} = \ldots = \tau_{n,1}$

foreach $i \in [n]$

  assert $\mathsf{Complete}_{\Pi,\mathcal{U}_i}(\mathsf{stmt}, T_i) = 1$

return $\mathcal{W}(\mathsf{stmt}, T_1, \ldots, T_n) \notin \mathcal{R}(\mathsf{stmt})$

Fig. 4. Conjunctive adaptive special soundness experiment and subroutines.

[LMS22] Alex Lombardi, Fermi Ma, and Nicholas Spooner. "Post-Quantum Zero Knowledge, Revisited or: How to Do Quantum Rewinding Undetectably". In: *63rd FOCS*. IEEE Computer Society Press, 2022, pp. 851–859. DOI: 10.1109/FOCS54457.2022.00086 (cit. on p. 80).

[LNP22] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. "Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General". In: *CRYPTO 2022, Part II*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. LNCS. Springer, Cham, Aug. 2022, pp. 71–101. DOI: 10.1007/978-3-031-15979-4_3 (cit. on p. 8).

[LNS21] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. "SMILE: Set Membership from Ideal Lattices with Applications to Ring Signatures and Confidential Transactions". In: *CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Cham, Aug. 2021, pp. 611–640. DOI: 10.1007/978-3-030-84245-1_21 (cit. on p. 8).

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "A Toolkit for Ring-LWE Cryptography". In: *EUROCRYPT 2013*. Ed. by Thomas Johansson and Phong Q. Nguyen. Vol. 7881. LNCS. Springer, Berlin, Heidelberg, May 2013, pp. 35–54. DOI: 10.1007/978-3-642-38348-9_3 (cit. on pp. 71, 72).

[RS62] J. Barkley Rosser and Lowell Schoenfeld. "Approximate formulas for some functions of prime numbers". In: *Illinois Journal of Mathematics* 6.1 (1962), pp. 64–94 (cit. on p. 74).

[Val08] Paul Valiant. "Incrementally Verifiable Computation or Proofs of Knowledge Imply Time/Space Efficiency". In: *TCC 2008*. Ed. by Ran Canetti. Vol. 4948. LNCS. Springer, Berlin, Heidelberg, Mar. 2008, pp. 1–18. DOI: 10.1007/978-3-540-78524-8_1 (cit. on p. 21).

[Wik05] Douglas Wikström. "On the l-Ary GCD-Algorithm in Rings of Integers". In: *Automata, Languages and Programming*. Ed. by Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 1189–1201. ISBN: 978-3-540-31691-6 (cit. on p. 57).

[YAZX+19] Rupeng Yang, Man Ho Au, Zhenfei Zhang, Qiuliang Xu, Zuoxia Yu, and William Whyte. "Efficient Lattice-Based Zero-Knowledge Arguments with Standard Soundness: Construction and Applications". In: *CRYPTO 2019, Part I*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11692. LNCS. Springer, Cham, Aug. 2019, pp. 147–175. DOI: 10.1007/978-3-030-26948-7_6 (cit. on p. 8).

A Conjunctive Composition of UCS

As explained in Section 4.2, the UCS sequence $\mathcal{U}$ of a $(\mathtt{NEST}[\mathcal{U}], \mathcal{W})$-adaptive special sound proof system can be interpreted as an instruction to the transcript extractor to extract via a nested composition of extractors. In the following, we consider another mode of composition — conjunction — denoted by $(\mathtt{AND}[\mathcal{U}], \mathcal{W})$-adaptive special soundness which is a generalisation of coordinate-wise special soundness (CWSS).

Definition A.1 (Conjunctive Adaptive Special Soundness). *Let $\mathcal{R} : \mathbb{X} \times \mathbb{W} \to \{0,1\}$ be a relation. Let Π be a public-coin interactive proof system with a challenge set $\mathfrak{C}$. Let $\mathcal{U} = (\mathcal{U}_i)_{i=1}^n$ where*

$\mathcal{U}_i : \mathbb{X} \times \mathcal{T}_\Pi \to 2^{\mathfrak{C}}$ *and* $\mathcal{W} : \mathbb{X} \times \mathcal{T}_\Pi \to \mathbb{W}$ *be deterministic polynomial time algorithms. We say that* Π *is* $(\mathtt{AND}[\mathcal{U}], \mathcal{W})$*-adaptive special sound for* $\mathcal{R}$ *if for all statement* $\mathsf{stmt} \in \mathbb{X}$ *and for all probabilistic extractors* $\mathcal{E}$ *it holds that* $\Pr[\mathsf{AdSS}^{\mathtt{AND},n}_{\Pi,\mathcal{U},\mathcal{W},\mathcal{E}}(\mathsf{stmt}) = 1] = 0$ *with probability taken over the randomness of* $\mathcal{E}$*, where the experiment* $\mathsf{AdSS}^{\mathtt{AND},n}_{\Pi,\mathcal{U},\mathcal{W},\mathcal{E}}$ *is defined in Fig. 4. In this case, we also say that* $\mathcal{U}$ *is the tuple of useful challenge structures associated with the witness extractor* $\mathcal{W}$*. We say that* Π *is* $\mathtt{AND}[\mathcal{U}]$*-adaptive special sound for* $\mathcal{R}$ *if it is* $(\mathtt{AND}[\mathcal{U}], \mathcal{W})$*-adaptive special sound for some witness extractor* $\mathcal{W}$*.*

Theorem A.2 (Conjunctive AdSS $\implies$ KS). *Let* $\mathcal{R} : \mathbb{X} \times \mathbb{W} \to \{0,1\}$ *be a relation. Let* Π *be a public-coin interactive proof system with a challenge set* $\mathfrak{C}$*. Let* $\mathcal{U} = (\mathcal{U}_i)_{i=1}^n$ *where* $\mathcal{U}_i : \mathbb{X} \times \mathcal{T}_\Pi \to 2^{\mathfrak{C}}$ *and* $\mathcal{W} : \mathbb{X} \times \mathcal{T}_\Pi \to \mathbb{W}$ *be deterministic polynomial time algorithms. Suppose that each* $\mathcal{U}_i$ *is natural (Definition 4.4) and* Π *is* $(\mathtt{AND}[\mathcal{U}], \mathcal{W})$*-adaptive special sound for* $\mathcal{R}$ *(Definition A.1). Then* Π *is knowledge sound for* $\mathcal{R}$ *with knowledge error* $\kappa = \sum_{i=1}^n \kappa(\mathcal{U}_i)$*. Specifically, there exists a knowledge extractor* $\mathcal{E}$ *with runtime overhead* $\sum_{i=1}^n \frac{\mathsf{depth}(\mathcal{U}_i)}{1-\kappa(\mathcal{U}_i)}$ *such that for all statement* $\mathsf{stmt} \in \mathbb{X}$ *and all prover* $\mathcal{P}^*$ *it succeeds with probability*

$$\Pr\big((\mathsf{stmt}; \mathcal{E}^{\mathcal{P}^*}(\mathsf{stmt})) \in \mathcal{R}\big) \geq \frac{\epsilon(\mathcal{P}^*, \mathsf{stmt}) - \kappa}{1 - \kappa}.$$

Proof. This follows from Theorem 4.5 and Corollary D.3 in a straightforward manner.

B Examples of UCS

We provide some examples of UCSs. For readability, throughout this section we leave the statement stmt implicit and write $\mathcal{U}(\tau_1, \ldots, \tau_\ell)$ in place of $\mathcal{U}(\mathsf{stmt}, (\tau_1, \ldots, \tau_\ell))$; each UCS below is understood to be (implicitly) parametrised by the statement stmt.

B.1 Transcript-Agnostic Examples

We begin with the example of k-special soundness which we already mentioned in Section 2.1.

Example B.1 (k-special soundness). Consider a challenges space $\mathfrak{C} = \{1, \ldots, N\}$ of size N (and $k \leq N$). The corresponding UCS is

$$\mathcal{U}(\tau_1, \ldots, \tau_\ell) = \begin{cases} \mathfrak{C} \setminus \{\mathsf{challof}(\tau_i)\}_{i=1}^\ell & |\{\mathsf{challof}(\tau_i)\}_{i=1}^\ell| < k, \\ \emptyset & |\{\mathsf{challof}(\tau_i)\}_{i=1}^\ell| \geq k. \end{cases}$$

That is, a challenge is useful precisely when it differs from all previously collected challenges and fewer than k distinct challenges have been collected so far. Clearly, $\kappa(\mathcal{U}) = \frac{k-1}{N}$ and $\mathsf{depth}(\mathcal{U}) = k$.

Our definitions of and properties for useful challenge structures are closely related to the monotonic structures $\Gamma \subseteq 2^{\mathfrak{C}}$, and the associated useful element function $\mathcal{U}_\Gamma \colon 2^{\mathfrak{C}} \to 2^{\mathfrak{C}}$, introduced in [AFR23] to define the notion Γ-out-of-$\mathfrak{C}$ special soundness.

Example B.2 (Access-structure special soundness). Special soundness w.r.t. an access structure as in [AFR23] is a special case of our definition. Recall the useful element function $\mathcal{U}_\Gamma \colon 2^{\mathfrak{C}} \to 2^{\mathfrak{C}}$ of [AFR23], which maps a set $S \subseteq \mathfrak{C}$ of challenges to the set of challenges that are useful given S according to the access structure Γ (see [AFR23] for the definition of useful). We define

$$\mathcal{U}(\tau_1, \ldots, \tau_\ell) = \mathcal{U}_\Gamma\big(\{\mathsf{challof}(\tau_i)\}_{i=1}^\ell\big),$$

i.e. a challenge c is useful for $(\tau_1, \ldots, \tau_\ell)$ if and only if $c \in \mathcal{U}_\Gamma(S)$ for the set $S = \{\mathsf{challof}(\tau_i)\}_{i=1}^\ell$ of challenges collected so far. With this definition of $\mathcal{U}$, $\kappa(\mathcal{U})$ coincides with the knowledge error κ_Γ in [AFR23] and the depth of $\mathcal{U}$ then coincides with the quantity "t_Γ" of the respective access structure Γ.

The crucial difference between access structure special soundness and AdSS is that a UCS $\mathcal{U}\colon \mathcal{T}^* \to 2^{\mathfrak{C}}$ takes as sequence of *transcripts*, whereas $\mathcal{U}_\Gamma$ takes as input only a set of challenges. Allowing the useful challenges $\mathcal{U}(T)$ to depend on the complete transcripts, rather than only the challenges, is the crucial refinement which captures probabilistic tests. A second and more subtle difference is that we consider sequences, rather than sets, of transcripts, i.e. we keep track of the order in which the transcripts have been found. This difference is not essential to our results, but merely convenient in the analysis.

The next (impractical) example explains the necessity of nested composition, despite the apparent ability of handling tree structures "directly" via a UCS.

Example B.3 (Tree structures, directly). Since a (k_1, k_2)-tree (in challenge space $\mathfrak{C}_1 \times \mathfrak{C}_2$) can be modelled as an access structure, we also obtain a direct extraction strategy for trees. However, as observed in [AFR23], while the knowledge error is optimal, the depth of the UCS is too big to yield an *efficient* extractor. The problem is that any challenge which could eventually be used in the extracted tree is considered useful, thus for $\mathfrak{C}_1 \times \mathfrak{C}_2$, a chain can be longer than $|\mathfrak{C}_1|$ elements. We can exploit the flexibility of UCS to instead grow the tree by considering a challenge useful only if it extends an incomplete (sub)tree; challenges which start a (k_1+1)-th subtree are now deemed useless. Formally, write π_1, π_2 for the projections of $\mathfrak{C}_1 \times \mathfrak{C}_2$ onto its two components. With an abuse of notation, for a sequence $T = (\tau_1, \dots, \tau_\ell)$, let

$$\pi_1(T) \coloneqq \{\pi_1(\mathsf{challof}(\tau)) \mid \tau \in T\} \subseteq \mathfrak{C}_1$$

be the set of distinct first-round challenges occurring in T (the roots of the subtrees grown so far), and for each $g \in \mathfrak{C}_1$ let

$$\pi_{2,g}(T) \coloneqq \{\pi_2(\mathsf{challof}(\tau)) \mid \tau \in T \text{ with } \pi_1(\mathsf{challof}(\tau)) = g\} \subseteq \mathfrak{C}_2$$

be the set of distinct second-round challenges that appear in T paired with first-round challenge g (the leaves of the subtree rooted at g). Here τ ranges over the transcripts of T, and only those whose first-round challenge equals g contribute to $\pi_{2,g}(T)$; in particular $\pi_{2,g}(T) = \emptyset$ whenever no transcript in T has first-round challenge g. We then define

$$\mathcal{U}(\tau_1, \dots, \tau_\ell) = \left\{ (c_1, c_2) \in \mathfrak{C}_1 \times \mathfrak{C}_2 \;\middle|\; \begin{array}{l} (c_1 \in \pi_1(T) \;\wedge\; |\pi_{2,c_1}(T)| < k_2 \;\wedge\; c_2 \notin \pi_{2,c_1}(T)) \\ \vee\; (c_1 \notin \pi_1(T) \;\wedge\; |\pi_1(T)| < k_1) \end{array} \right\},$$

i.e. (c_1, c_2) is useful if it either adds a new leaf to an existing, not-yet-complete subtree, or starts a new subtree as long as fewer than k_1 subtrees have been started. This ensures a depth of $k_1 \cdot k_2$. However, the knowledge error suffers severely, becoming $\kappa(\mathcal{U}) \geq 1 - \frac{|\mathfrak{C}_2|}{|\mathfrak{C}_1 \times \mathfrak{C}_2|} = 1 - \frac{1}{|\mathfrak{C}_1|} \approx 1$, since for the final subtree and the last challenge to be found, at most $|\mathfrak{C}_2| - (k_2 - 1)$ challenges are useful.

Next, we show how the language of UCS can capture t-fold parallel repetition of a protocol.

Example B.4 (t-parallel repetition). Let $\mathcal{U}$ be a UCS over $\mathfrak{C}$. The t-fold parallel repetition of the underlying protocol has challenge set $\mathfrak{C}^t$, and each of its transcripts is a tuple of t base transcripts, one per thread. Given a sequence $T' = (\tau^1, \dots, \tau^i)$ of such transcripts, where $\tau^m = (\tau_1^m, \dots, \tau_t^m)$ has challenge $\mathsf{challof}(\tau^m) = (\mathsf{challof}(\tau_1^m), \dots, \mathsf{challof}(\tau_t^m)) \in \mathfrak{C}^t$, and a thread $j \in [t]$, we define the projection $T'|_j$ to thread j as follows: $T'|_j$ consists of the subsequence $(\tau_j^{i_1}, \dots, \tau_j^{i_k})$ of $(\tau_j^1, \dots, \tau_j^m)$ of thread-j transcripts, such that $i_1 = 1$ and for all k, i_k is the smallest choice with $\mathsf{challof}(\tau_j^{i_k}) \in \mathcal{U}(\tau_j^{i_1, \dots, i_{k-1}})$; in other words, we construct $T'|_j$ iteratively from $(\tau_j^1, \dots, \tau_j^m)$ by starting from the beginning and appending τ_j^i to $T'|_j$ if its challenge is $\mathcal{U}$-useful w.r.t. the transcripts already appended. We define the t-fold product useful challenge structure $\mathcal{U}'$ over $\mathfrak{C}^t$ by

$$\mathcal{U}'(T') = \begin{cases} \emptyset & \text{if} \quad \exists j\colon \mathcal{U}(T'|_j) = \emptyset \\ \bigcup_{j=1}^{t} \big(\mathfrak{C}^{j-1} \times \mathcal{U}(T'|_j) \times \mathfrak{C}^{t-j}\big), & \text{otherwise} \end{cases}$$

i.e. a parallel challenge $(c_1, \dots, c_t)$ is useful for $\mathcal{U}'$ if and only if at least one of its components c_j is useful for $\mathcal{U}$ given thread j's accumulated chain $T'|_j$. Observe that every parallel transcript added to a $\mathcal{U}'$-chain advances at least one thread (it extends some $T'|_j$), and that the first one advances all t threads since $\mathcal{U}(\emptyset) = \mathfrak{C}$. Moreover, $\mathcal{U}'(T') = \emptyset$ if $\mathcal{U}(T'|_j) = \emptyset$ for some thread j. Since each thread's chain has length at most $\mathsf{depth}(\mathcal{U})$, the depth of $\mathcal{U}'$ is at most $\mathsf{depth}(\mathcal{U}') \leq t \cdot (\mathsf{depth}(\mathcal{U}) - 1) + 1$. Moreover, it is easy to see that $\kappa(\mathcal{U}') = \kappa(\mathcal{U})^t$. Namely, this is the same combinatorial analysis as in [AF22; AFR23].

Unfortunately, similar to tree extraction, Example B.4 does not seem to generalise to multi-round protocols.

B.2 A Basic Lattice Example

In all previous examples, the UCS only depended on the *challenges* and not on the actual transcripts, i.e. they just recovered results of [AFR23; AFKR26]. Indeed, those examples are simple and based on generic concepts, and are decoupled from specific extraction requirements. The power of allowing transcript-dependent useful challenges is most clearly seen in sequential compositions of protocols. The reason for this is that a UCS allows reacting to *failures of probabilistic tests*, from which extraction may either be possible, or not desired (because it would blow up the runtime complexity). We give a toy example of a 2-move protocol.

Example B.5 (Short SIS preimage, toy example). Let $\alpha, \beta, \gamma > 0$, where $\alpha \leq \gamma$. Let $(\mathbf{A}, \mathbf{y})$ be the statement, where $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{y} = \mathbf{A}\mathbf{x} \bmod q \in \mathbb{Z}_q^n$, and $\mathbf{x} \in \mathbb{Z}_q^m$ the witness. Consider following template 2-move protocol (which is a 3-move protocol where the prover sends an empty first message):

- $\mathcal{V}(\mathbf{A}, \mathbf{y})$ sends a random $\mathbf{B} \leftarrow_\$ \{0,1\}^{\lambda \times m}$ where $\lambda \in \mathbb{N}$ is a parameter.
- $\mathcal{P}((\mathbf{A}, \mathbf{y}), \mathbf{x})$ responds with $\mathbf{x}$.
- $\mathcal{V}$ accepts if $\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q$ and $\|\mathbf{B}\mathbf{x}\| \leq \beta$ and $\|\mathbf{x}\| \leq \gamma$.

The purpose of the toy protocol is to verify if $\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q$ and $\|\mathbf{x}\| \leq \alpha$.

Let $\varepsilon = \varepsilon(\lambda)$ be such that $\max_{\mathbf{B}} \Pr[\|\mathbf{x}\| > \alpha \wedge \|\mathbf{B}\mathbf{x}\| \leq \beta] \leq \varepsilon$, where the probability is over $\mathbf{B}$. Intuitively, the check $\|\mathbf{B}\mathbf{x}\| \leq \beta$ is a probabilistic test (with error probability ε) which asserts that $\|\mathbf{x}\| \leq \alpha$.

Note that $\mathcal{V}$ does not check $\|\mathbf{x}\| \leq \alpha$ directly. In this example, this is an artificial choice, but it arises naturally in security proofs of real protocols: In a real protocol, $\mathcal{P}$ would not send $\mathbf{x}$ but instead prove knowledge of some $\mathbf{x}$ which satisfies the verifier's checks, so the verifier *could not* check $\|\mathbf{x}\| \leq \alpha$ directly in this setting, and the toy protocol would constitute a probabilistic check which the verifier can and does execute.[18] Similar approaches are used in [CGKR22; BS23].

The extraction goal in this example is to either recover $\mathbf{x}$ with $\|\mathbf{x}\| \leq \alpha$, or find two different preimages $\mathbf{x} \neq \mathbf{x}'$ of $\mathbf{y}$, i.e. $\mathbf{x} - \mathbf{x}'$ is a SIS solution to $\mathbf{A}$ of norm at most 2γ. Towards this goal, we define the following UCS (implicitly parametrised by $(\mathbf{A}, \mathbf{y})$). Recall that $\mathcal{T}_{\Pi}$ contains only accepting transcripts, so any transcript $(\mathbf{B}, \mathbf{x})$ in the input satisfies $\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q$, $\|\mathbf{B}\mathbf{x}\| \leq \beta$, and $\|\mathbf{x}\| \leq \gamma$. We set

$$\mathcal{U}(\tau_1, \ldots, \tau_\ell) = \begin{cases} \mathfrak{C} & \ell = 0, \\ \emptyset & \ell = 1,\ \tau_1 = (\mathbf{B}, \mathbf{x}),\ \|\mathbf{x}\| \leq \alpha, \\ \{\mathbf{B}' \mid \|\mathbf{B}'\mathbf{x}\| > \beta\} & \ell = 1,\ \tau_1 = (\mathbf{B}, \mathbf{x}),\ \|\mathbf{x}\| > \alpha, \\ \emptyset & \ell \geq 2. \end{cases}$$

It is straightforward to see that the knowledge error of this UCS satisfies $\kappa(\mathcal{U}) \leq \varepsilon$. Clearly, a witness $\mathbf{x}$ can be extracted from any complete $\mathcal{U}$-chain T, that is, a $\mathcal{U}$-chain with $\mathcal{U}(T) = \emptyset$. To see this, note that a complete $\mathcal{U}$-chain falls into one of two cases.

1. In the first case, we have $T = (\mathbf{B}, \mathbf{x})$ is accepting (so $\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q$) and $\|\mathbf{x}\| \leq \alpha$. Therefore $\mathbf{x}$ is an ISIS solution to $(\mathbf{A}, \mathbf{y})$.
2. In the second case, we have $T = ((\mathbf{B}, \mathbf{x}), (\mathbf{B}', \mathbf{x}'))$ where both transcripts are accepting (so $\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}' = \mathbf{y} \bmod q$, $\|\mathbf{B}'\mathbf{x}'\| \leq \beta$, $\|\mathbf{x}\| \leq \gamma$, and $\|\mathbf{x}'\| \leq \gamma$), $\|\mathbf{x}\| > \alpha$, and $\|\mathbf{B}'\mathbf{x}\| > \beta$. Note that $\mathbf{x} \neq \mathbf{x}'$, for otherwise $\|\mathbf{B}'\mathbf{x}'\| > \beta$ which is a contradiction. We therefore have $\mathbf{A}(\mathbf{x} - \mathbf{x}') = \mathbf{0} \bmod q$ and $0 < \|\mathbf{x} - \mathbf{x}'\| \leq 2\gamma$, i.e. $\mathbf{x} - \mathbf{x}'$ is a SIS solution to $\mathbf{A}$.

We make a few observations for the example above.

- The ability of the useful challenges to depend on the transcript, e.g. $(\mathbf{B}, \mathbf{x})$, was crucial to define $\mathcal{U}$. Unlike the examples in Section B.1, the challenge $\mathbf{B}$ says nothing about which $\mathbf{B}'$ enforce $\mathbf{x}' \neq \mathbf{x}$.

[18] We note that this is a common design principle in proof systems: Large communication (here $\mathbf{x}$) and certain checks can be replaced by shorter proofs of knowledge and potentially cheaper checks. For example, the folding protocols use this to shrink $\mathbf{x}$.

– The example easily extends to any protocol following the template where the verifier sends a probabilistic test.
– Recent works which use this approach, e.g. [CGKR22; BS23] and surely many more, rely on ad-hoc extractor constructions and analysis. Our framework can capture such settings, yielding simpler and tighter security proofs.[19]

While this simple example gives a straightforward application of our approach, in real protocols, the probabilistic tests may be intricately connected with the rest of the protocol. Indeed, in Section G, we see how to handle a lattice-based folding protocol (building on the ideas of [BF23]) with an implicit probabilistic test. Moreover, in Section I.2 we implicitly show that adaptive special soundness also applies to Labrador [BS23].

C Knowledge Extractors for AdSS Protocols

In this section, we define extend the knowledge extractor from [AFKR26] to handle adaptive special-sound Σ-protocols, and analyse its properties, thereby proving knowledge soundness. To simplify the exposition, we first introduce abstract adversaries and extractors, followed by some notation and conventions.

C.1 Abstract Adversaries and Extractors

Suppose $(\mathcal{P}, \mathcal{V})$ is a $\mathcal{U}$-adaptive special sound Σ-protocol for relation $\mathcal{R}$. Our goal is to construct a knowledge extractor that, given oracle access to a (dishonest) prover $\mathcal{P}^*$ attacking $(\mathcal{P}, \mathcal{V})$ on statement stmt, aims to output a witness $w \in \mathcal{R}(\mathsf{stmt})$. Without loss of generality, we may assume that $\mathcal{P}^*$ is deterministic (see [AF22]), i.e. $\mathcal{P}^*$ is effectively a deterministic function that takes as input a challenge c and outputs a protocol transcript (a, c, z).

Remark C.1 (Fixed statements). Our analysis will be done for any fixed statement stmt. For conciseness, we also consider stmt as an implicit input hard-coded into $\mathcal{P}^*$ and other relevant algorithms. As our extractors analysis eventually depends only on $\kappa(\mathcal{U}_{\mathsf{stmt}})$ and $\mathsf{depth}(\mathcal{U}_{\mathsf{stmt}})$ (or knowledge error $\kappa(\mathsf{stmt})$ and runtime overhead $\rho(\mathsf{stmt})$ for composition of extractors), fixing stmt for the analysis simplifies the presentation.

As in prior works [AF22; AFK22; AFK23; AFR23; AFKR26], we present our core results in a more abstract language, allowing us to later handle more complicated scenarios such as multi-round interactive proofs. In the following, $\mathfrak{C}$ is an arbitrary challenge set whose uniform distribution is efficiently sampleable. We write $\mathcal{T} = \{0,1\}^*$ to denote the space of transcripts. To simplify notation and handling of extractors, we make following definition.

Definition C.2 (Abstract adversary). *An **(abstract) adversary** is an algorithm (or a probabilistic function) $\mathcal{A}\colon \mathfrak{C} \to \mathcal{T} \cup \{\bot\}$ which maps a challenge $c \in \mathfrak{C}$ to a transcript. We require that there is an efficient function $\mathsf{challof}_{\mathcal{A}}\colon \mathcal{T} \to \mathfrak{C}$ which recovers the challenge of transcript, i.e. for all $c \in \mathfrak{C}$ and $\tau \leftarrow \mathcal{A}(c)$, if $\tau \neq \bot$ then $\mathsf{challof}_{\mathcal{A}}(\tau) = c$. Furthermore, for a verification predicate $\mathsf{V}\colon \mathcal{T} \to \{0,1\}$, we call $\mathcal{A}$ V**-respecting** if for all $c \in \mathfrak{C}$ and all $\tau \leftarrow \mathcal{A}(c)$, $\tau \neq \bot$ implies $\mathsf{V}(\tau) = 1$.*

We will **always assume V-respecting** $\mathcal{A}$ for an implicit V. Only for defining nested composition will this become of interest. We note that our approach is similar to prior works, except that we integrate the verifier into $\mathcal{A}$ itself. In other words, any non-$\bot$ output of $\mathcal{A}$ is accepting. This reduces visual noise in formulas.

Example C.3 (Σ-protocols). A Σ-protocol $\Pi = (\mathcal{P}, \mathcal{V})$ is a 3-move public-coin interactive proof, i.e. the verifier's message is sampled uniformly at random from some challenge set $\mathfrak{C}$. Moreover, the verifier's final output is $\mathsf{V}(\tau)$, where τ is the transcript. Given any (w.l.o.g.) deterministic malicious prover $\mathcal{P}^*(\mathsf{stmt}, \mathsf{aux})$, an abstract adversary $\mathcal{A}$ is obtained as follows: For a challenge c, run $\mathcal{P}^*(\mathsf{stmt}, \mathsf{aux})$ with challenge c to obtain a transcript $\tau = (a, c, z) \in \{0,1\}^* \times \mathfrak{C} \times \{0,1\}^*$. If $\mathsf{V}(\tau) = 1$, output τ, else $\bot$. Clearly, $\mathcal{A}$ is V-respecting. Also note that, since $\mathcal{A}$ is deterministic, all $(a, c, z) \leftarrow \mathcal{A}(c)$ have the same a.

[19] A simple ad-hoc solution (in our example) is to retry using uniformly random challenge $\mathbf{B}'$. However, it can happen that $\mathbf{B}'\mathbf{x} \leq \beta$ holds, making the transcript useless.

Our formalism can also capture multi-round protocols, by setting $\mathfrak{C} = \mathfrak{C}_1 \times \ldots \times \mathfrak{C}_\mu$. This is discussed in detail in Section E. Now, we turn to *abstract extractors.*

Definition C.4 ((Ordinary) Abstract extractor). *An **(abstract) extractor** $\mathcal{E}$ over challenge set $\mathfrak{C}$ is a (probabilistic) oracle algorithm $\mathcal{E}^{\mathcal{A}} \colon \mathfrak{C} \to \{0,1\}^* \cup \{\bot\}$, which takes any oracle $\mathcal{A} \colon \mathfrak{C} \to \{0,1\}^* \cup \{\bot\}$. We require that an abstract extractor is **ordinary**: If $\mathcal{E}^{\mathcal{A}}(c)$ queries $\mathcal{A}$, then the first query must be on its input $c \in \mathfrak{C}$, and if that response $\mathcal{A}(c)$ is $\bot$, then $\mathcal{E}^{\mathcal{A}}(c)$ immediately outputs $\bot$.*

*We call $\mathcal{E}$ $(\mathsf{V}^{\mathsf{pre}}, \mathsf{V}^{\mathsf{post}})$-**compatible**, if for any $\mathsf{V}^{\mathsf{pre}}$-respecting oracle $\mathcal{A}$, $\mathcal{E}^{\mathcal{A}}$ is $\mathsf{V}^{\mathsf{post}}$-respecting.*

Remark C.5 ($\mathsf{challof}$ *for extractors).* Note that an extractor $\mathcal{E}^{\mathcal{A}}$ is itself an adversary over $\mathfrak{C}$, i.e. it has the same interface. By the requirements of the $\mathsf{challof}$ function, for any initial challenge $c \in \mathfrak{C}$, if $T \leftarrow \mathcal{E}^{\mathcal{A}}(c)$ is not $\bot$, then $\mathsf{challof}_{\mathcal{E}^{\mathcal{A}}}(T) = c$.

As with adversaries, we **leave** $(\mathsf{V}^{\mathsf{pre}}, \mathsf{V}^{\mathsf{post}})$ **implicit** unless they are of relevance. Their relevance is limited to straightforward correctness claims. The explicit pre- and post-conditions are (only) used for correctness claims of extractors and nested composition, and thus usually left implicit.

As a simple example, for an extractor of a Σ-protocol, we set $\mathsf{V}^{\mathsf{pre}}(\tau)$ as the verifier's verification algorithm $\mathsf{V}(\mathsf{stmt}, \tau)$ and $\mathsf{V}^{\mathsf{post}}(\mathsf{wit}) = [(\mathsf{stmt}, \mathsf{wit}) \in \mathcal{R}]$. (Recall that we hard-code stmt in extractors and hence $(\mathsf{V}^{\mathsf{pre}}, \mathsf{V}^{\mathsf{post}})$.)

C.2 Notation and Conventions

In the following, we define shorthand notations for recurring expressions involving the abstract adversary $\mathcal{A}$ (Section C.1).

Definition C.6 (Success and knowledge-error quantities). *Let $U \subseteq C \subseteq \mathfrak{C}$. With respect to the adversary $\mathcal{A}$, we define the following quantities associated to U relative to C:*

Useful-success probability. *The probability that $c \leftarrow_\$ C$ lands inside U and $\mathcal{A}$ succeeds on it:*

$$\varepsilon(U \mid C) \coloneqq \Pr[\mathcal{A}(c) \neq \bot \wedge c \in U \mid c \leftarrow_\$ C]\,. \tag{2}$$

Useless-success probability. *The probability that $c \leftarrow_\$ C$ lands outside U and $\mathcal{A}$ succeeds on it:*

$$\Delta(U \mid C) \coloneqq \Pr[\mathcal{A}(c) \neq \bot \wedge c \notin U \mid c \leftarrow_\$ C]\,. \tag{3}$$

Relative knowledge error. *The relative size of the useless set $C \setminus U$ (independent of $\mathcal{A}$):*

$$\kappa(U \mid C) \coloneqq \Pr[c \notin U \mid c \leftarrow_\$ C]\,. \tag{4}$$

Relative extraction probability.

$$\delta(U \mid C) \coloneqq \frac{\varepsilon(C \mid C) - \Delta(U \mid C)}{1 - \Delta(U \mid C)} = \varepsilon(U \mid U) \cdot \frac{1 - \kappa(U \mid C)}{1 - \Delta(U \mid C)}\,. \tag{5}$$

The following (in)equalities are immediate from the definitions and basic probability theory.

Proposition C.7. *The following (in)equalities hold:*

$$\varepsilon(C \mid C) = \Pr[\mathcal{A}(c) \neq \bot \mid c \leftarrow_\$ C], \tag{6}$$

$$\Delta(U \mid C) = \varepsilon(C \mid C) - \varepsilon(U \mid C), \quad /\!/\ \Delta(U \mid C) \text{ is the exact (absolute) difference between } \varepsilon(U \mid C) \text{ and } \varepsilon(C \mid C) \tag{7}$$

$$\Delta(U \mid C) \leq \kappa(U \mid C), \tag{8}$$

$$\varepsilon(U \mid U) = \frac{\varepsilon(C \mid C) - \Delta(U \mid C)}{1 - \kappa(U \mid C)}, \tag{9}$$

$$\varepsilon(U \mid U) \geq \delta(U \mid C). \tag{10}$$

The last inequality will be crucial in our extractor analysis. In Section C.5, we further show that $\Delta(U \mid C)$ obeys useful chain rules. Using the notation in Definition C.6, we can write the **knowledge error** (defined in Definition 4.3) of a UCS $\mathcal{U}$ as follows.

Lemma C.8. *For a useful challenge structure $\mathcal{U}$ on $\mathfrak{C}$, the **knowledge error** and **extraction probability** of $\mathcal{U}$ are, respectively,*

$$\kappa(\mathcal{U}) \coloneqq \max_{S \in \mathsf{IncompChains}(\mathcal{U})} \kappa(\mathcal{U}(S) \mid \mathfrak{C}) \qquad \text{and} \qquad \delta(\mathcal{U}) \coloneqq \min_{S \in \mathsf{IncompChains}(\mathcal{U})} \delta(\mathcal{U}(S) \mid \mathfrak{C})\,. \tag{11}$$

Intuitively, $\kappa(\mathcal{U})$ is simply the relative size of the *largest useless subset* of $\mathfrak{C}$ which can be encountered during an extraction.

C.3 Extractor Construction

We are now ready to define and analyse our knowledge extractor $\mathcal{E}$, adapting the work [AFKR26]. The extractor $\mathcal{E}$ is given rewindable black-box access to an adversary $\mathcal{A}\colon \mathfrak{C} \to \mathcal{T} \cup \{\bot\}$, denoted as $\mathcal{E}^{\mathcal{A}}$, and aims to output a sequence of transcripts $T = (\tau_1, \ldots, \tau_k) \in \mathcal{T}^*$ such that $\mathcal{U}(T) = \emptyset$ for the useful challenge structure $\mathcal{U}$. In case of Σ-protocols, the transcripts $\tau_i = (a, c_i, z_i)$ have common first message a (as $\mathcal{P}^*$ is deterministic). To ease notation, we define the following shorthand.

Definition C.9 (Last Challenge Set). *We write $\mathcal{C}(T) := \mathcal{U}\big(\mathsf{pred}(T)\big)$ for $T \in \mathcal{T}^*$ to denote the challenge set from which the challenge of the last transcript in T is (supposed to be) sampled.*

The extractor is formally defined in Figure 5. In its first step, $\mathcal{E}^{\mathcal{A}}$ evaluates $\mathcal{A}$ on a challenge $c \leftarrow_{\$} \mathcal{U}(\emptyset) = \mathfrak{C}$ sampled uniformly at random. If $\bot \leftarrow \mathcal{A}(c)$, i.e. if $\mathcal{A}$ is unsuccessful on this challenge, the extractor aborts and returns $\bot$. Otherwise, i.e. if $\tau \leftarrow \mathcal{A}(c)$ with $\tau \neq \bot$, it continues to run an iterative sub-extractor $\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T)$ on input the sequence $T = (\tau)$ of transcripts.

The iterative sub-extractor $\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T)$ is an algorithm that, given rewindable black-box access to $\mathcal{A}$ and on input a sequence of transcripts $T \in \mathcal{T}^*$, proceeds as follows. First, it checks if $\mathcal{U}(T) = \emptyset$, i.e. if there are no more useful challenges. If this is the case, the extractor has already succeeded and it simply returns T. Otherwise, it enters a (while-)loop. In each iteration of this loop, a useful challenge $c \leftarrow_{\$} \mathcal{U}(T)$ is sampled and passed to $\mathcal{A}$, which returns $\tau \in \mathcal{T} \cup \{\bot\}$. If $\tau \neq \bot$, i.e. $\mathcal{A}(c)$ has successfully returned an accepting transcript, we say that this iteration is a success, and the iterative extractor returns whatever $\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T\|\tau)$ returns. If instead $\tau = \bot$, the extractor samples a coin *coin* with the following procedure:

- Sample $c \leftarrow \mathcal{C}(T)$;
- If $c \notin \mathcal{U}(T) (\subseteq \mathcal{C}(T))$, evaluate $\tau \leftarrow \mathcal{A}(c)$, else set $\tau = \bot$;
- If $\tau \neq \bot$, then set $coin = 1$, otherwise set $coin = 0$.

If $coin = 1$, then the iterative extractor aborts and returns $\bot$. The above steps are repeated until a successful loop is reached, or until $coin = 1$ is flipped. We call $\mathcal{E}_{\mathsf{iter}}$ *iterative*, because the recursive calls are all tail-recursive, and could trivially be unrolled into a loop.

For convenience, we keep track of a bit $\mathsf{Succ}(T)$ which indicates whether extraction of this call was successful in finding an accepting transcript. That is $\mathsf{Succ}(T) = 1$, if either $\mathcal{U}(T) = \emptyset$, i.e. there was nothing to do, or if the loop found $\tau \neq \bot$ and will return the result of the tail-recursive call $\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T\|\tau)$.

For completeness, we recall the differences of this knowledge extraction strategy from [AFKR26] (and our adaptation) compared to the previous knowledge extractors of [AF22; AFR23]. Intuitively, these approaches recursively or iteratively try to find useful challenges/transcripts until the extraction of a witness is possible. The coin tossing procedure allows the extractor to abort, without being able to extract a witness, so that the expected runtime is polynomial.

However, there are a number of crucial differences between (our adaptation of) the approach of [AFKR26] and the prior works [AF22; AFR23]. First, as discussed in Section 4, our useful challenge structure $\mathcal{U}\colon \mathcal{T}^* \to 2^{\mathfrak{C}}$ takes as input sequences of transcripts, whereas in prior works the useful challenges only depend on the challenges of the transcripts found so far. This generalises access structures of [AFKR26], and will for instance allow us to more efficiently handle the soundness slack in lattice-based bulletproofs. Second, we adapt the coin tossing procedure from [AFKR26] to handle UCSs. The careful coin tossing from [AFKR26] improves the (bound on the) success probability of the extractor by a factor $\mathsf{depth}(\mathcal{U})$ compared to [AF22; AFR23]. Finally, the extractor of [AFKR26] aims to find new transcripts one by one, i.e. it is tail-recursive. As a consequence, and in contrast to the extractors of [AF22; AFR23], our extractor never forgets or throws away accepting transcripts that have been found.

Let us remark that in all cases of interest, it is efficiently possible to sample from $\mathcal{U}(T)$ since membership and emptiness are efficiently decidable (cf. Definition 4.4)

Remark C.10 (Generically sampling from $\mathcal{U}(T)$). Recall that, by assumption, sampling uniformly from $\mathfrak{C}$ is efficient and membership in $\mathcal{U}(T) \subseteq \mathfrak{C}$ is efficiently testable for any $T \in \mathcal{T}^*$. Suppose now $\mathcal{U}(T) \neq \emptyset$, which is also efficiently testable. Then we can sample uniformly from $\mathcal{U}(T) \neq \emptyset$ by rejection sampling: Sample $C \leftarrow_{\$} \mathfrak{C}$ and return C if $C \in \mathcal{U}(T)$, else retry. It is easy to see that we require $\Pr[C \in \mathcal{U}(T) \mid C \leftarrow_{\$} \mathfrak{C}]^{-1} \leq \frac{1}{1-\kappa(\mathcal{U})}$ retries in expectation. Thus, as long as $\kappa(\mathcal{U}) \leq 1 - 1/\mathsf{poly}(|\mathsf{stmt}|)$, sampling is expected polynomial time in $|\mathsf{stmt}|$.

Parameters: a challenge set $\mathfrak{C}$, useful challenge structure $\mathcal{U}$ on $\mathfrak{C}$ and a sequence of transcripts $T \in \mathcal{T}^*$.
Probabilistic black-box access to: $\mathcal{A}\colon \mathfrak{C} \to (\{0,1\}^* \times \mathfrak{C} \times \{0,1\}^*) \cup \{\perp\}$.

$\mathcal{E}^{\mathcal{A}}_{\mathcal{U}}(c)$:

1. If $c = \perp$, sample $c \leftarrow_{\$} \mathcal{U}(\emptyset) = \mathfrak{C}$ uniformly.
2. Evaluate $\tau \leftarrow \mathcal{A}(c)$.
3. If $\tau \neq \perp$, let $T = (\tau)$ and return $\mathcal{E}_{\mathsf{iter}}(T)$. // Set $\mathsf{Succ}(\emptyset) = 1$.
4. Else return $\perp$.

$\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T)$:

1. If $\mathcal{U}(T) = \emptyset$:
   - Return T // Set $\mathsf{Succ}(T) = 1$
2. Else, repeat until return:
   (a) Sample a new transcripts τ as follows:
      - Sample $c \leftarrow_{\$} \mathcal{U}(T)$ uniformly at random;
      - Evaluate $\tau \leftarrow \mathcal{A}(c)$.
   (b) If $\tau \neq \perp$:
      - Return $\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T\|\tau)$. // Set $\mathsf{Succ}(T) = 1$
   (c) Else, sample *coin* as follows:
      - Sample $c \leftarrow_{\$} \mathcal{C}(T) = \mathcal{U}\big(\mathsf{pred}(T)\big)$ uniformly at random;
      - If $c \notin \mathcal{U}(T)$, then evaluate $\tau \leftarrow \mathcal{A}(c)$, else set $\tau = \perp$;
      - If $\tau \neq \perp$, then $coin = 1$, else $coin = 0$.
   (d) If $coin = 1$:
      - Return $\perp$.

Fig. 5. Transcript extractor $\mathcal{E}^{\mathcal{A}}_{\mathcal{U}}\colon \mathfrak{C} \to \mathcal{T} \cup \{\perp\}$ for $\mathcal{U}$-AdSS proof systems. The comments track when $\mathsf{Succ}(T)$ transitions to 1.

C.4 Auxiliary Lemmata

In our analysis, we use the following facts about geometric distributions and the monotonicity of certain functions. The proofs are straightforward and therefore omitted.

Lemma C.11. *Let $X \sim \mathsf{Geo}(p)$ and $Y \sim \mathsf{Geo}(q)$ be two geometrically distributed random variables. Then*

$$\min(X,Y) \sim \mathsf{Geo}(1-(1-p)(1-q)) \quad \textit{and} \quad \Pr[X \leq Y] = \frac{p}{1-(1-p)(1-q)}\,.$$

Lemma C.12. *We gather following simple facts for $x, y \in [0,1]$:*

- $f(x,y) = \frac{x-y}{1-y}$ *is monotonically decreasing in y;*
- $f(x,y) = \frac{x}{1-(1-x)(1-y)}$ *is monotonically increasing in x;*
- $f(x,y) = \frac{1}{1-(1-x)(1-y)}$ *is monotonically decreasing in x and y.*

The next lemma helps simplify runtime analysis in Sections C and E. It is a generalisation of a similar lemma in [AFKR26].

Lemma C.13 (Stopped summation). *Let $X_i \geq 0$ and $\tau \in \mathbb{N}$ be random variables; we make no independency requirements. Suppose there is some $\theta \in \mathbb{R}$ such that for all $i \in \mathbb{N}$ we have*

$$\mathbb{E}[X_i \mid \tau \geq i] \leq \theta.$$

Then, we get

$$\mathbb{E}\Big[\sum_{i=1}^{\tau} X_i\Big] \leq \mathbb{E}[\tau] \cdot \theta.$$

Proof. The claim follows from basic probability theory:

$$\begin{aligned}\mathbb{E}\Big[\sum_{i=1}^{\tau} X_i\Big] &= \mathbb{E}\Big[\sum_{i=1}^{\infty} X_i \cdot \mathbb{1}\{\tau \geq i\}\Big] = \sum_{i=1}^{\infty} \mathbb{E}[X_i \cdot \mathbb{1}\{\tau \geq i\}] \\ &= \sum_{i=1}^{\infty} \mathbb{E}[X_i \mid \tau \geq i] \Pr[\tau \geq i] \leq \theta \cdot \sum_{i=1}^{\infty} \Pr[\tau \geq i] = \theta \cdot \mathbb{E}[\tau]\end{aligned}$$

Note that we are allowed to exchange expectation and summation since all summands are non-negative. □

C.5 Two Chain Rules

Here, we state and prove two chain rules that will be useful later.

Lemma C.14 (Chain Rule 1). *Let $\mathfrak{C}$ be a challenge set, let $C_3 \subseteq C_2 \subseteq C_1 \subseteq \mathfrak{C}$ and let $\mathcal{A}\colon \mathfrak{C} \to \mathcal{T}^* \cup \{\bot\}$. Then it holds that*

$$(1 - \kappa(C_3 \mid C_2)) \cdot (1 - \kappa(C_2 \mid C_1)) = 1 - \kappa(C_3 \mid C_1)\,.$$

Proof. Abusing notation, for $c \leftarrow_\$ \mathfrak{C}$ sampled uniformly at random, we write C for the event $c \in C$ (for any $C \subseteq \mathfrak{C}$). Moreover, without loss of generality, we may assume that $C_1 = \mathfrak{C}$.

Then,

$$\begin{aligned}1 - \kappa(C_3 \mid C_2) &= \Pr(C_3 \mid C_2)\,, \\ 1 - \kappa(C_2 \mid C_1) &= \Pr(C_2)\,, \\ 1 - \kappa(C_3 \mid C_1) &= \Pr(C_3)\,,\end{aligned}$$

and thus

$$\begin{aligned}(1 - \kappa(C_3 \mid C_2)) \cdot (1 - \kappa(C_2 \mid C_1)) &= \Pr(C_3 \mid C_2) \cdot \Pr(C_2) \\ &= \Pr(C_3 \wedge C_2) = \Pr(C_3) = 1 - \kappa(C_3 \mid C_1)\,,\end{aligned}$$

which completes the proof. □

Lemma C.15 (Chain Rule 2). *Let $\mathfrak{C}$ be a challenge set, let $C_3 \subseteq C_2 \subseteq C_1 \subseteq \mathfrak{C}$ and let $\mathcal{A}\colon \mathfrak{C} \to \mathcal{T}^* \cup \{\bot\}$. Then it holds that*

$$\big(1 - \Delta(C_3 \mid C_2)\big) \cdot \big(1 - \Delta(C_2 \mid C_1)\big) \leq 1 - \Delta(C_3 \mid C_1)\,.$$

Proof. Let us use the same notation as in the proof of Lemma C.14, i.e. for $c \leftarrow_\$ \mathfrak{C}$ sampled uniformly at random, we write C for the event $c \in C$ (for any $C \subseteq \mathfrak{C}$) and, without loss of generality, we assume that $C_1 = \mathfrak{C}$. Further, abusing notation, we write A for the event $\mathcal{A}(c) \neq \bot$.

Then,

$$\begin{aligned}\Delta(C_3 \mid C_2) &= \Pr(A \wedge \neg C_3 \mid C_2)\,, \\ \Delta(C_2 \mid C_1) &= \Pr(A \wedge \neg C_2)\,, \\ \Delta(C_3 \mid C_1) &= \Pr(A \wedge \neg C_3)\,,\end{aligned}$$

and thus

$$\begin{aligned}&\big(1 - \Delta(C_3 \mid C_2)\big) \cdot \big(1 - \Delta(C_2 \mid C_1)\big) \\ &\quad= (1 - \Pr[A \wedge \neg C_3 \mid C_2]) \cdot (1 - \Pr[A \wedge \neg C_2]) \\ &\quad= 1 - \Pr[A \wedge \neg C_3 \mid C_2] - \Pr[A \wedge \neg C_2] \\ &\qquad\quad + \Pr[A \wedge \neg C_3 \mid C_2] \cdot \Pr[A \wedge \neg C_2] \\ &\quad= 1 - \Pr[A \wedge \neg C_2] - \Pr[A \wedge C_2 \wedge \neg C_3] + \Pr[A \wedge C_2 \wedge \neg C_3] \\ &\qquad\quad + \Pr[A \wedge \neg C_3 \mid C_2] \cdot \Pr[A \wedge \neg C_2] - \Pr[A \wedge \neg C_3 \mid C_2] \\ &\quad= 1 - \Pr[A \wedge \neg C_3] + \Pr[A \wedge \neg C_3 \mid C_2] \cdot \Pr(C_2) \\ &\qquad\quad + \Pr[A \wedge \neg C_3 \mid C_2] \cdot \Pr[A \wedge \neg C_2] - \Pr[A \wedge \neg C_3 \mid C_2] \\ &\quad= 1 - \Delta(C_3 \mid C_1) + \Pr[A \wedge \neg C_3 \mid C_2] \cdot \big(\Pr[C_2] + \Pr[A \wedge \neg C_2] - 1\big) \\ &\quad\leq 1 - \Delta(C_3 \mid C_1) + \Pr[A \wedge \neg C_3 \mid C_2] \cdot \big(\Pr[C_2] + \Pr[\neg C_2] - 1\big) \\ &\quad= 1 - \Delta(C_3 \mid C_1)\,,\end{aligned}$$

which completes the proof of the lemma. □

C.6 Analysis of the Extractor

Our analysis of $\mathcal{E}_{\mathcal{U}}$ proceeds in three steps. We first show *correctness* (Lemma C.16): a non-$\perp$ output is always a complete $\mathcal{U}$-chain of accepting transcripts, leaving only the success probability and runtime to bound. We then analyse a *single iteration* of the loop $\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T)$ (Lemmas C.17 to C.19), and finally lift these per-iteration bounds to the *whole extractor* in Lemma C.20 by induction over the depth of $\mathcal{U}$ via the chain rules of Lemmas C.14 and C.15. Corollary C.21 provides less tight but simpler bounds than Lemma C.20 which are useful in practice. Finally, Theorem 4.5 combines $\mathcal{E}_{\mathcal{U}}$ and the witness extractor $\mathcal{W}$ associated to $\mathcal{U}$ to obtain a knowledge extractor.

Lemma C.16 (Correctness of $\mathcal{E}_{\mathcal{U}}$). *Let $\mathcal{U}$ be a useful challenge structure on a challenge set $\mathfrak{C}$, and let $\mathcal{A}$ be some adversary. Then, the extractor $\mathcal{E}$ in Figure 5 outputs either $\perp$, or a $\mathcal{U}$-chain of transcripts $T = (\tau_1, \ldots, \tau_k)$ with $\mathcal{U}(T) = \emptyset$.*

Proof. The claim is immediate by construction of $\mathcal{E}_{\mathcal{U}}$. □

Let us now analyse the main properties of our knowledge extractor. To this end, we first bound the success probability $\Pr[\mathsf{Succ}(T) = 1]$ of the first iteration of $\mathcal{E}_{\mathsf{iter}}(T)$. In Lemma C.17 we provide a lower bound for this success probability and in Lemma C.18 we provide an upper bound. The lower bound will be used to bound the success probability of our knowledge extractor, and the upper bound will be used to bound its expected runtime.

Lemma C.17 (One-Iteration Success Probability, Lower Bound). *Fix an arbitrary non-empty sequence of transcripts. Further, recall that we write $\mathcal{C}(T) = \mathcal{U}\big(\mathsf{pred}(T)\big)$. Then*

$$\Pr[\mathsf{Succ}(T) = 1] \geq \frac{\delta(\mathcal{U}(T) \mid \mathcal{C}(T))}{\varepsilon(\mathcal{C}(T) \mid \mathcal{C}(T))} = \frac{\varepsilon(\mathcal{U}(T) \mid \mathcal{U}(T))}{\varepsilon(\mathcal{C}(T) \mid \mathcal{C}(T))} \cdot \frac{1 - \kappa(\mathcal{U}(T) \mid \mathcal{C}(T))}{1 - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T))}$$

Proof. For $\mathcal{U}(T) = \emptyset$ the claim is trivial. So suppose $\mathcal{U}(T) \neq \emptyset$. Let $\mathit{coin}(T)$ be the distribution of *coin* sampled as in Step 2(c) of $\mathcal{E}_{\mathsf{iter}}(T)$. We have

$$\Pr[\mathit{coin}(T) = 1] = \Pr[\mathcal{A}(c) \neq \perp \wedge c \notin \mathcal{U}(T) \mid c \leftarrow_{\$} \mathcal{C}(T)] = \Delta(\mathcal{U}(T) \mid \mathcal{C}(T))\,. \tag{12}$$

Furthermore, let us define the random variables

$$X = X(T) \sim \mathsf{Geo}(\varepsilon(\mathcal{U}(T) \mid \mathcal{U}(T)))\,,$$
$$Y = Y(T) \sim \mathsf{Geo}(\Delta(\mathcal{U}(T) \mid \mathcal{C}(T)))\,,$$

which are the number of trials required — without ever aborting — until the loop finds $\tau \neq \perp$ (and sets $\mathsf{Succ}(T) = 1$) for the first time (X), respectively the *coin* would be 1 for the first time (Y). It follows that $\Pr[\mathsf{Succ}(T) = 1] = \Pr[X \leq Y]$, as success occurs if and only if the extractor succeeds before the coin yields 1. Thus, by Lemma C.11, we obtain that

$$\Pr[\mathsf{Succ}(T) = 1] = \frac{\varepsilon(\mathcal{U}(T) \mid \mathcal{U}(T))}{1 - (1 - \varepsilon(\mathcal{U}(T) \mid \mathcal{U}(T)))(1 - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T)))}\,.$$

Now note that, by Equations (5) and (9),

$$\varepsilon(\mathcal{U}(T) \mid \mathcal{U}(T)) \geq \frac{\varepsilon(\mathcal{C}(T) \mid \mathcal{C}(T)) - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T))}{1 - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T))} = \delta(\mathcal{U}(T) \mid \mathcal{C}(T))\,.$$

Therefore, since $x \mapsto \frac{x}{1-(1-x)(1-y)}$ in monotonically increasing for $y \in [0, 1]$ (Lemma C.12), it follows that

$$\Pr[\mathsf{Succ}(T) = 1] \geq \frac{\delta(\mathcal{U}(T) \mid \mathcal{C}(T))}{1 - (1 - \delta(\mathcal{U}(T) \mid \mathcal{C}(T)))(1 - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T)))}\,.$$

Finally, by definition of δ (Equation (5)), we observe that

$$\begin{aligned}
&(1-\delta(\mathcal{U}(T) \mid \mathcal{C}(T)))(1 - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T))) \\
&= (1 - \frac{\varepsilon(\mathcal{C}(T) \mid \mathcal{C}(T)) - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T))}{1 - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T))})(1 - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T))) \\
&= 1 - \varepsilon(\mathcal{C}(T) \mid \mathcal{C}(T))\,.
\end{aligned} \tag{13}$$

Hence,

$$\Pr[\mathsf{Succ}(T) = 1] \geq \frac{\delta(\mathcal{U}(T) \mid \mathcal{C}(T))}{\varepsilon(\mathcal{C}(T) \mid \mathcal{C}(T))}\,,$$

which completes the proof. □

Lemma C.18 (One-Iteration Success Probability, Upper Bound). *In the same setting as Lemma C.17, we have*

$$\Pr[\mathsf{Succ}(T) = 1] \leq \frac{\varepsilon(\mathcal{U}(T) \mid \mathcal{U}(T))}{\varepsilon(\mathcal{C}(T) \mid \mathcal{C}(T))}\,.$$

Proof. Following the proof of Lemma C.17, but now using that $x \mapsto \frac{1}{1-(1-x)(1-y)}$ is monotonically decreasing for all $y \in [0,1]$ (Lemma C.12) and that

$$\delta(\mathcal{U}(T) \mid \mathcal{C}(T)) \leq \varepsilon(\mathcal{U}(T) \mid \mathcal{U}(T))\,,$$

it follows that

$$\begin{aligned}
\Pr[\mathsf{Succ}(T) = 1] &= \frac{\varepsilon(\mathcal{U}(T) \mid \mathcal{U}(T))}{1 - (1 - \varepsilon(\mathcal{U}(T) \mid \mathcal{U}(T)))(1 - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T)))} \\
&\leq \frac{\varepsilon(\mathcal{U}(T) \mid \mathcal{U}(T))}{1 - (1 - \delta(\mathcal{U}(T) \mid \mathcal{C}(T)))(1 - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T)))} \\
&= \frac{\varepsilon(\mathcal{U}(T) \mid \mathcal{U}(T))}{\varepsilon(\mathcal{C}(T) \mid \mathcal{C}(T))}\,,
\end{aligned}$$

which completes the proof. □

Lemma C.19 ($\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}$ Run-Time). *Let $T \in \mathcal{T}^*$ be any sequence of transcripts, and let the random variable $\mathsf{time}(T) \coloneqq \mathsf{time}_{\mathcal{A}}(\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T))$ denote the time spent in the blackbox $\mathcal{A}$ while running $\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T)$ until either $\mathsf{Succ}(T) = 1$ or $\mathit{coin} = 1$ occurs.*

$$\begin{aligned}
\mathbb{E}[\mathsf{time}(T)] \leq \frac{1}{\varepsilon(\mathcal{C}(T) \mid \mathcal{C}(T))} \cdot \Big(&\mathbb{E}[\mathsf{time}(\mathcal{A}(C)) \mid C \in \mathcal{U}(T)] \\
&+ \kappa(\mathcal{U}(T) \mid \mathcal{C}(T)) \cdot \mathbb{E}[\mathsf{time}(\mathcal{A}(C)) \mid C \in \mathcal{C}(T) \setminus \mathcal{U}(T)]\Big)
\end{aligned}$$

where $C \leftarrow_{\$} \mathfrak{C}(T)$ is uniform.

Proof. Define $X(T)$ and $Y(T)$ as in Lemma C.17. The number of iterations of the repeat-until loop in $\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T)$ is now given as $\min(X, Y)$. By Lemma C.11, we have

$$\begin{aligned}
\mathbb{E}[\min(X,Y)] &= \frac{1}{\big(1 - (1 - \varepsilon(\mathcal{U}(T) \mid \mathcal{U}(T)))(1 - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T)))\big)} \\
&\leq \frac{1}{\big(1 - (1 - \delta(\mathcal{U}(T) \mid \mathcal{U}(T)))(1 - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T)))\big)} \\
&= \frac{1}{\varepsilon(\mathcal{C}(T) \mid \mathcal{C}(T))}\,,
\end{aligned}$$

where we first use that $x \mapsto \frac{1}{1-(1-x)(1-y)}$ is monotonically decreasing for all $y \in [0,1]$ (Lemma C.12) together with $\varepsilon(\mathcal{U}(T) \mid \mathcal{U}(T)) \geq \delta(\mathcal{U}(T) \mid \mathcal{U}(T))$, and then simplify the denominator as in Equation (13) above.

In each iteration, one query $\mathcal{A}(C)$ is made with $C \leftarrow_{\$} \mathcal{C}(T)$, and then, if it was is not successful, the coin is sampled. When sampling the coin, the algorithm $\mathcal{A}$ is only invoked if the random challenges c does not lie in $\mathcal{C}(T) = \mathcal{U}\big(\mathsf{pred}(T)\big)$. This happens with probability $\kappa(\mathcal{U}(T) \mid \mathcal{C}(T))$, and in this case $\mathcal{A}(C)$ is invoked for a uniform $C \in \mathcal{C}(T) \setminus \mathcal{U}(T)$. Thus, the expected time spent in $\mathcal{A}(C)$ per iteration of the while-loop is at most

$$t = \mathbb{E}[\mathsf{time}(\mathcal{A}(C)) \mid C \in \mathcal{U}(T)] + \kappa(\mathcal{U}(T) \mid \mathcal{C}(T)) \cdot \mathbb{E}[\mathsf{time}(\mathcal{A}(C)) \mid C \in \overline{\mathcal{U}}(T)]\,,$$

where $\overline{\mathcal{U}}(T) = \mathcal{C}(T) \setminus \mathcal{U}(T)$. By Lemma C.13, it follows that

$$\mathbb{E}[\mathsf{time}(T)] \leq \mathbb{E}[\min(X,Y)] \cdot t\,,$$

which completes the proof of the lemma. □

We are now ready to state and prove the main lemma analysing the extractor $\mathcal{E}$.

Lemma C.20. *Let $\mathcal{U}$ be a useful challenge structure on a challenge set $\mathfrak{C}$, and recall that we write $\mathcal{C}(T) := \mathcal{U}(\mathsf{pred}(T))$ for all transcript sequences $T \in \mathcal{T}^*$. Let $\mathcal{A}$ be some adversary. Then, for the extractor $\mathcal{E}$ in Figure 5 we obtain*

$$\Pr[\mathcal{E}^{\mathcal{A}} \neq \bot] \geq \delta(\mathcal{U}) = \min_{S \in \mathsf{IncompChains}(\mathcal{U})} \delta(\mathcal{U}(S) \mid \mathfrak{C})\,, \tag{14}$$

$$\begin{aligned}\mathbb{E}[\mathsf{time}_{\mathcal{A}}(\mathcal{E}^{\mathcal{A}})] = &\max_{T \in \mathsf{IncompChains}(\mathcal{U})} \sum_{R \preceq T} \mathbb{E}[\mathsf{time}(\mathcal{A}(C)) \mid C \in \mathcal{U}(R)] \\ &+ \kappa(\mathcal{U}(R) \mid \mathcal{C}(R)) \cdot \mathbb{E}[\mathsf{time}(\mathcal{A}(C)) \mid C \in \mathcal{C}(R) \setminus \mathcal{U}(R)]\end{aligned} \tag{15}$$

Proof. We argue separately for success probability and runtime. In both cases, we argue by induction over the depth k of $\mathcal{U}$. Concretely, we establish claims for $\mathcal{U}^{\leq k}$, where

$$\mathcal{U}^{\leq k}(T) := \begin{cases} \emptyset\,, & \text{if } T \in \mathsf{Chains}(\mathcal{U}) \text{ with length } |T| \text{ at least } k\,, \\ \mathcal{U}(T)\,, & \text{otherwise}\,. \end{cases}$$

We let $\mathcal{E}_{\leq k}$ denote the extractor applied to $\mathcal{U}^{\leq k}$ instead of $\mathcal{U}$. More precisely, the extractor $\mathcal{E}_{\leq k}$ proceeds exactly as the extractor $\mathcal{E}$, except that it immediately succeeds when it has found a chain T of length at least k (even if $\mathcal{U}(T) \neq \emptyset$). In particular, note that $\mathcal{E}_{\leq k} = \mathcal{E}$ for all $k \geq \mathsf{depth}(\mathcal{U})$.

Success Probability. For all $k \in \mathbb{N}$, we will prove that

$$\Pr[\mathcal{E}^{\mathcal{A}}_{\leq k} \neq \bot] \geq \delta(\mathcal{U}) = \min_{S \in \mathsf{IncompChains}(\mathcal{U}^{\leq k})} \delta(\mathcal{U}(S) \mid \mathcal{C}(\emptyset))\,.$$

From this the claimed success probability immediately follows.

First, we observe that

$$\Pr[\mathcal{E}^{\mathcal{A}}_{\leq k} \neq \bot] = \sum_{T \in \mathsf{Chains}(\mathcal{U}^{\leq k})} \Pr[\mathcal{E}^{\mathcal{A}}_{\leq k} = T]\,.$$

Let us now fix $T = (\tau_1, \ldots, \tau_\ell) \in \mathsf{Chains}(\mathcal{U}^{\leq k})$, i.e. $\ell \leq k$, and let us write $T_0 = \emptyset$ and $T_i = (\tau_1, \ldots, \tau_i)$ for all $1 \leq i \leq \ell$. Then, $\mathcal{E}^{\mathcal{A}}_{\leq k}$ can only output the sequence T if either $\mathcal{U}(T) = \emptyset$, in which case $\Pr[\mathcal{E}^{\mathcal{A}}_{\leq k} = T] = \Pr[\mathcal{E}^{\mathcal{A}}_{\leq \ell} = T]$, or if $\ell = k$, in which case it also holds that $\Pr[\mathcal{E}^{\mathcal{A}}_{\leq k} = T] = \Pr[\mathcal{E}^{\mathcal{A}}_{\leq \ell} = T]$.

Further, recall that the binary random variable $\mathsf{Succ}(T_i)$ denotes whether the extractor has successfully found an additional transcript to append the sequence T_i. In particular, $\mathsf{Succ}(T_0) = \mathsf{Succ}(\emptyset)$ denotes the probability that the first $\mathcal{A}$-invocation of the extractor is successful. Additionally, we let $\Upsilon(T_i)$ be the random variable that denotes the transcript that has been found, when trying to append the sequence T_i. Then, it holds that

$$\begin{aligned}\Pr[\mathcal{E}^{\mathcal{A}}_{\leq k} = T] &= \Pr[\mathcal{E}^{\mathcal{A}}_{\leq \ell} = T] \\ &= \Pr[\mathcal{E}^{\mathcal{A}}_{\leq \ell-1} = T_{\ell-1}] \cdot \Pr[\Upsilon(T_{\ell-1}) = \tau_\ell] \\ &= \Pr[\mathcal{E}^{\mathcal{A}}_{\leq \ell-1} = T_{\ell-1}] \\ &\qquad \cdot \Pr[\mathsf{Succ}(T_{\ell-1}) = 1] \cdot \Pr[\Upsilon(T_{\ell-1}) = \tau_\ell \mid \mathsf{Succ}(T_{\ell-1}) = 1] \\ &= \Pr[\mathcal{E}^{\mathcal{A}}_{\leq \ell-2} = T_{\ell-2}] \cdot \Pr[\Upsilon(T_{\ell-2}) = \tau_{\ell-1}] \\ &\qquad \cdot \Pr[\mathsf{Succ}(T_{\ell-1}) = 1] \cdot \Pr[\Upsilon(T_{\ell-1}) = \tau_\ell \mid \mathsf{Succ}(T_{\ell-1}) = 1] \\ &\qquad \vdots \\ &= \prod_{i=1}^{\ell} \Big(\Pr[\mathsf{Succ}(T_{i-1}) = 1] \cdot \Pr[\Upsilon(T_{i-1}) = \tau_i \mid \mathsf{Succ}(T_{i-1}) = 1] \Big)\,.\end{aligned} \tag{16}$$

By Lemma C.17, it now follows that for all $2 \leq i \leq \ell$

$$\Pr[\mathsf{Succ}(T_{i-1}) = 1] \geq \frac{\varepsilon(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-1}))}{\varepsilon(\mathcal{C}(T_{i-1}) \mid \mathcal{C}(T_{i-1}))} \cdot \frac{1 - \kappa(\mathcal{U}(T_{i-1}) \mid \mathcal{C}(T_{i-1}))}{1 - \Delta(\mathcal{U}(T_{i-1}) \mid \mathcal{C}(T_{i-1}))}$$

$$= \frac{\varepsilon(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-1}))}{\varepsilon(\mathcal{U}(T_{i-2}) \mid \mathcal{U}(T_{i-2}))} \cdot \frac{1-\kappa(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-2}))}{1-\Delta(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-2}))}\,,$$

where we use that $\mathcal{C}(T_{i-1}) = \mathcal{U}(T_{i-2})$. Hence,

$$\begin{aligned}\prod_{i=2}^{\ell} \Pr[\mathsf{Succ}(T_{i-1}) = 1] &\geq \prod_{i=2}^{\ell} \frac{\varepsilon(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-1}))}{\varepsilon(\mathcal{U}(T_{i-2}) \mid \mathcal{U}(T_{i-2}))} \cdot \frac{1-\kappa(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-2}))}{1-\Delta(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-2}))} \\ &= \frac{\varepsilon(\mathcal{U}(T_{\ell-1}) \mid \mathcal{U}(T_{\ell-1}))}{\varepsilon(\mathcal{U}(\emptyset) \mid \mathcal{U}(\emptyset))} \cdot \prod_{i=2}^{\ell} \frac{1-\kappa(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-2}))}{1-\Delta(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-2}))}\,.\end{aligned}$$

Additionally using that $\Pr[\mathsf{Succ}(\emptyset) = 1] = \varepsilon(\mathcal{U}(\emptyset) \mid \mathcal{U}(\emptyset)) = \varepsilon(\mathfrak{C} \mid \mathfrak{C})$ shows that

$$\prod_{i=1}^{\ell} \Pr[\mathsf{Succ}(T_{i-1}) = 1] \geq \varepsilon(\mathcal{U}(T_{\ell-1}) \mid \mathcal{U}(T_{\ell-1})) \cdot \prod_{i=2}^{\ell} \frac{1-\kappa(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-2}))}{1-\Delta(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-2}))}\,.$$

From the chains rules of Lemma C.14 and Lemma C.15, it now follows that

$$\begin{aligned}\prod_{i=1}^{\ell} \Pr[\mathsf{Succ}(T_{i-1}) = 1] &\geq \varepsilon(\mathcal{U}(T_{\ell-1}) \mid \mathcal{U}(T_{\ell-1})) \cdot \frac{1-\kappa(\mathcal{U}(T_{\ell-1}) \mid \mathfrak{C})}{1-\Delta(\mathcal{U}(T_{\ell-1}) \mid \mathfrak{C})} \\ &= \delta(\mathcal{U}(T_{\ell-1}) \mid \mathfrak{C}) \geq \delta(\mathcal{U})\,.\end{aligned}$$

Plugging this inequality into Equation 16 shows that

$$\Pr[\mathcal{E}^{\mathcal{A}}_{\leq k} = T] \geq \delta(\mathcal{U}) \cdot \prod_{i=1}^{\ell} \Pr[\Upsilon(T_{i-1}) = \tau_i \mid \mathsf{Succ}(T_{i-1}) = 1]\,,$$

and thus

$$\begin{aligned}\Pr[\mathcal{E}^{\mathcal{A}}_{\leq k} \neq \bot] &= \sum_{T \in \mathsf{Chains}(\mathcal{U}^{\leq k})} \Pr[\mathcal{E}^{\mathcal{A}}_{\leq k} = T] \\ &\geq \delta(\mathcal{U}) \cdot \sum_{\ell=1}^{k} \sum_{\substack{T=(\tau_1,\ldots,\tau_\ell) \\ \in \mathsf{Chains}(\mathcal{U}^{\leq k})}} \prod_{i=1}^{\ell} \Pr[\Upsilon(T_{i-1}) = \tau_i \mid \mathsf{Succ}(T_{i-1}) = 1] \\ &= \delta(\mathcal{U})\,.\end{aligned}$$

For the final equality, recall that $\Upsilon(T_{i-1})$ is the random variable which denotes the i-th transcript found during the extraction (which exists since $\mathsf{Succ}(T_{i-1}) = 1$ implies it). Thus, the product is simply the probability that extraction finishes with sequence T, conditioned on success. Hence, the summation over all transcript T that the extractor can successfully output yields 1.

Expected Run-Time Spent in $\mathcal{A}$-Queries. We first bound the expected time $\mathcal{E}_{\mathsf{iter}}$ spends running $\mathcal{A}$, and then use this to bound the expected time of $\mathcal{E}$. For $\mathcal{E}_{\mathsf{iter}}$, we argue by induction over the depth. Unlike success probability, where we truncated executions at depth k, this time, we make the inductive argument from the end of the execution. For this, let $\mathsf{depth}_T(\mathcal{U})$ be the maximal remaining depth of sequences of transcripts appending the sequence T, i.e.

$$\mathsf{depth}_T(\mathcal{U}) = \max_{\substack{S \in \mathsf{Chains}(\mathcal{U}) \\ \text{s.t. } T \preceq S}} \mathsf{length}(S) - \mathsf{length}(T)\,.$$

Our induction will be over prefixes $T \in \mathsf{IncompChains}(\mathcal{U})$ of relative depth $\mathsf{depth}_T(\mathcal{U}) \leq k$. (Note that $\mathsf{depth}(\mathcal{U}) = \mathsf{depth}_\emptyset(\mathcal{U})$.) That is, induction over the remaining depth of $\mathcal{U}$ for chains starting at T. Let $\mathsf{T}(T)$ denote the expected time the extractor $\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T)$ spends in $\mathcal{A}$ until it either aborts or it finds the next transcript appending the sequence of transcripts T (see Lemma C.19). Let $\mathsf{T}_{\mathsf{tot}}(T)$ be the random variable denoting the total expected time $\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T)$ spends in $\mathcal{A}$, that is, the time until $\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T)$ finishes (including recursive calls). It thus holds that

$$\mathbb{E}[\mathsf{T}_{\mathsf{tot}}(T)] = \mathop{\mathbb{E}}_{S=T\|\tau}[\mathsf{T}(T) + \mathsf{T}_{\mathsf{tot}}(S)] = \mathbb{E}[\mathsf{T}(T)] + \mathop{\mathbb{E}}_{S=T\|\tau}[\mathsf{T}_{\mathsf{tot}}(S)]\,, \tag{17}$$

where the expectation is over all sequences S that are exactly one transcript longer than the sequence T, or $S = T\|\bot$, if no transcript was found, in which case $\mathsf{T}_{\mathsf{tot}}(S) = 0$.

Let us introduce the following short-hand notation:

- $\overline{\mathcal{U}}(R) = \mathcal{C}(R) \setminus \mathcal{U}(S)$ for any $R \in \mathcal{T}^*$. That is $\overline{\mathcal{U}}$ denotes useless challenges.
- $\kappa(R) = \kappa(\mathcal{U}(R)|\mathcal{C}(R))$ for any $R \in \mathcal{T}^*$. That is κ denotes the knowledge error at "step R".
- $\mathsf{T}_{\mathcal{A}} = \mathsf{time}(\mathcal{A}(C))$, the time spent in one call of $\mathcal{A}$ for uniform $C \in \mathfrak{C}$.
- $\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{M}] = \mathbb{E}[\mathsf{time}(\mathcal{A}(C)) \mid C \in \mathcal{M}]$ for $\mathcal{M} \subseteq \mathfrak{C}$ and uniform $C \in \mathfrak{C}$.

Our inductive claim is now that if $\mathsf{depth}_T(\mathcal{U}) \le k$ then

$$\begin{aligned}&\mathbb{E}[\mathsf{T}_{\mathsf{tot}}(T)]\\ &\le \tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))} \max_{\substack{T'' \in \mathsf{IncompChains}(\mathcal{U})\\ \text{s.t. } T \preceq T''}} \sum_{\substack{T' \text{ s.t.}\\ T \preceq T' \preceq T''}} \big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(T')] + \kappa(T') \cdot \mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(T')]\big)\end{aligned}$$

where the summation is over all chains T' in between T and T''. In the following, we always assume $T, T', T'' \in \mathsf{IncompChains}(\mathcal{U})$.

The base case is $\mathsf{depth}_T(\mathcal{U}) = 1$, which means $\mathcal{U}(T'') = \emptyset$ for any $T'' \succ T$. Consequently $T = T''$ is the only chain satisfying $T \preceq T'' \in \mathsf{IncompChains}(\mathcal{U})$, and the sum is just $\kappa(\mathcal{U}(T) \mid \mathcal{C}(T))$. Moreover, if $\tau \neq \bot$ is found, the extraction exits successfully (since the tail call immediately returns successfully). Hence, the time $\mathsf{T}_{\mathsf{tot}}(T)$ is exactly the time $\mathsf{T}_{\mathcal{A}}(T)$ spent in $\mathcal{A}$ during in a single iteration of $\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T)$ until $\mathit{coin}(T) = 1$ or $\mathsf{Succ}(T) = 1$. By Lemma C.19 we have

$$\mathbb{E}[\mathsf{T}_{\mathsf{tot}}(T)] = \mathbb{E}[\mathsf{T}_{\mathcal{A}}(T)] \le \tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))}\big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(T)] + \kappa(R) \cdot \mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(T)]\big)\,,$$

which yields the induction base case.

For the induction step, assume for $\mathsf{depth}_T(\mathcal{U}) \le k$ the claim holds. Let T be such that $\mathsf{depth}_T(\mathcal{U}) \le k+1$. The execution of $\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(T)$ first runs the loop where it either finds $\tau \neq \bot$, defines S and runs $\mathcal{E}^{\mathcal{A}}_{\mathsf{iter}}(S)$ (and sets $\mathsf{Succ}(T) = 1$), or it aborts (due to $\mathit{coin} = 1$). Now, by Lemma C.19, we find

$$\mathbb{E}[\mathsf{T}(T)] = \tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))}\big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(T)] + \kappa(R) \cdot \mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(T)]\big)\,. \tag{18}$$

Write $S = \bot$ if $\tau = \bot$. Then $S = \bot$ means that $\mathcal{E}_{\mathsf{iter}}(S)$ is never run, hence we have $\Pr[\mathsf{T}_{\mathsf{tot}}(S) \mid S = \bot] = 0$. Now, we bound $\mathbb{E}_S[\mathsf{T}_{\mathsf{tot}}(S)]$ as follows:

$$\begin{aligned}&\mathop{\mathbb{E}}_{S}[\mathsf{T}_{\mathsf{tot}}(S)] = \Pr[S \neq \bot] \cdot \mathop{\mathbb{E}}_{S}[\mathsf{T}_{\mathsf{tot}}(S) \mid S \neq \bot]\\
&\le \tfrac{\varepsilon(\mathcal{U}(T)|\mathcal{U}(T))}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))} \cdot \mathop{\mathbb{E}}_{S}[\mathsf{T}_{\mathsf{tot}}(S) \mid S \neq \bot]\\
&\le \tfrac{\varepsilon(\mathcal{U}(T)|\mathcal{U}(T))}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))} \mathop{\mathbb{E}}_{S}\Big[\tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))} \max_{S \preceq S''} \sum_{S \preceq S' \preceq S''} \big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(S')] + \kappa(S') \cdot \mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(S')]\big)\Big]\\
&\le \tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))} \cdot \mathop{\mathbb{E}}_{S}\Big[\max_{S \preceq S''} \sum_{S \preceq S' \preceq S''} \big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(S')] + \kappa(S') \cdot \mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(S')]\big)\Big]\\
&= \tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))} \cdot \max_{T \prec S} \max_{S \preceq S''} \sum_{S \preceq S' \preceq S''} \big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(S')] + \kappa(S') \cdot \mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(S')]\big)\\
&= \tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))} \cdot \max_{T \prec S''} \sum_{S \preceq S' \preceq S''} \big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(S')] + \kappa(S') \cdot \mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(S')]\big)\,,\end{aligned} \tag{19}$$

where we first use that $\Pr[\mathsf{T}_{\mathsf{tot}}(S) \mid S = \bot] = 0$; then $\Pr[S \neq \bot] \le \frac{\varepsilon(\mathcal{U}(S)|\mathcal{U}(S))}{\varepsilon(\mathcal{C}(S)|\mathcal{C}(S))}$ by Lemma C.18; the next step uses that $\mathsf{T}_{\mathsf{tot}}(S)$ is distributed as the number of queries $\mathcal{E}_{\mathsf{iter}}(S)$ (or 0 if $S = \bot$), and apply the induction hypothesis; then we use that $\mathcal{C}(S) = \mathcal{U}(\mathsf{pred}(S)) = \mathcal{U}(T)$, as $T = \mathsf{pred}(S)$, to cancel the ε-terms. Next, we replace expectation over $S = S(T)$ with the maximum, and finally, we use that

$$\max_{T \prec S} \max_{S \preceq S''} f(S') = \max_{T \prec S''} f(S')\,,$$

for any function f, in particular the summation.

The induction claim now follows by plugging the bounds from (18) and (19) back into (17), which yields

$$\begin{aligned}
&\mathbb{E}[\mathsf{T}_{\mathsf{tot}}(T)] \\
&= \mathbb{E}[\mathsf{T}(T)] + \mathop{\mathbb{E}}_{S=T\|\tau}[\mathsf{T}_{\mathsf{tot}}(S)]\,, \\
&\leq \tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))}\big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(T)] + \kappa(R)\cdot\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(T)]\big) \\
&+ \tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))}\cdot \max_{T\preceq S''} \sum_{S\preceq S'\preceq S''} \big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(S')] + \kappa(S')\cdot\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(S')]\big) \\
&= \tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))}\cdot \max_{T\preceq S''} \sum_{T\preceq S'\preceq S''} \big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(S')] + \kappa(S')\cdot\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(S')]\big)
\end{aligned}$$

where the equality uses that the summation *always* starts with $\kappa(\mathcal{U}(T) \mid \mathcal{C}(T))$.

Lastly, to obtain the statement of the theorem, observe that $\mathcal{E}$ makes exactly 1 query to $\mathcal{A}$ and then runs $\mathcal{E}_{\mathsf{iter}}$ with probability $\varepsilon(\mathcal{U}(\emptyset) \mid \mathcal{U}(\emptyset))$. Let $\mathsf{T}_{\mathsf{tot}} = \mathsf{T}_{\mathsf{tot}}(\emptyset)$ denote the total expected time $\mathcal{E}$ spends in $\mathcal{A}$ (including all recursive calls). Since $\mathcal{C}(\tau) = \mathcal{U}(\mathsf{pred}(\tau)) = \mathcal{U}(\emptyset)$, we obtain

$$\mathbb{E}[\mathsf{T}_{\mathsf{tot}}] = \max_{T\preceq T''} \sum_{T\preceq T'\preceq T''} \big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(T')] + \kappa(T')\cdot\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(T')]\big)$$

by the same argument as above, where we formally set $\varepsilon(\mathfrak{C}(\emptyset) \mid \mathfrak{C}(\emptyset)) = 1$. Lastly, since $T = \emptyset$ and $T'' \in \mathsf{IncompChains}(\mathcal{U})$, the claim of the theorem follows (with notation R, T instead of T', T''). □

The bounds in Lemma C.20 are precise but unwieldy. Below, we provide simplified bounds which are easy to use and sufficient for most applications.

Corollary C.21 ($\mathcal{E}^{\mathcal{A}}$ Success Probability and Run-Time). *Let $\mathfrak{C}$ be a challenge set and let $\mathcal{U}$ be a useful challenge structure on $\mathfrak{C}$. Let $\mathcal{A}$ be some adversary. Then, for the extractor $\mathcal{E}$ in Fig. 5 we obtain*

$$\Pr[\mathcal{E}^{\mathcal{A}}(\emptyset) \neq \bot] \geq \frac{\Pr[\mathcal{A}(C) \neq \bot \mid C \leftarrow_{\$} \mathfrak{C}] - \kappa(\mathcal{U})}{1-\kappa(\mathcal{U})}$$

$$\mathbb{E}[\mathsf{time}_{\mathcal{A}}(\mathcal{E}^{\mathcal{A}}(\emptyset))] \leq \frac{\mathsf{depth}(\mathcal{U})}{1-\kappa(\mathcal{U})}\cdot\mathbb{E}[\mathsf{time}(\mathcal{A}(C))]\,.$$

Proof. All claims are immediate consequences of Lemma C.20 and our requirement on $\mathcal{U}$. For the first claim, substitute the definition $\delta(U \mid C) = \frac{\varepsilon(C|C)-\Delta(U|C)}{1-\Delta(U|C)}$ from (5) into $\delta(\mathcal{U})$ from (11). Next, from $\Delta(U|C) \leq \kappa(U|C)$ and monotonicity of $f(x,y) = \frac{x-y}{1-y}$ (see Lemma C.12), the claim on the success probability follows.

To prove the bound on expected runtime, first observe that

$$\begin{aligned}
&\mathbb{E}[\mathsf{time}(\mathcal{A}(C)) \mid C \in \mathcal{U}(R)] + \kappa(\mathcal{U}(R) \mid \mathcal{C}(R))\cdot\mathbb{E}[\mathsf{time}(\mathcal{A}(C)) \mid C \in \mathcal{C}(R)\setminus\mathcal{U}(R)] \\
&= \frac{\mathbb{E}[\mathsf{time}(\mathcal{A}(C))\cdot\mathbb{1}\{C\in\mathcal{U}(R)\}]}{\Pr[C\in\mathcal{U}(R)|C\in\mathcal{C}(R)]} + \mathbb{E}[\mathsf{time}(\mathcal{A}(C))\cdot\mathbb{1}\{C\notin\mathcal{U}(R)\}] \\
&= (\tfrac{1}{\Pr[C\in\mathcal{U}(R)|C\in\mathcal{C}(R)]} - 1)\,\mathbb{E}[\mathsf{time}(\mathcal{A}(C))\cdot\mathbb{1}\{C\in\mathcal{U}(R)\}] + \mathbb{E}[\mathsf{time}(\mathcal{A}(C))] \\
&\leq (\tfrac{1}{\Pr[C\in\mathcal{U}(R)|C\in\mathcal{C}(R)]} - 1)\,\mathbb{E}[\mathsf{time}(\mathcal{A}(C))] + \mathbb{E}[\mathsf{time}(\mathcal{A}(C))] \\
&= \tfrac{1}{1-\kappa(\mathcal{U}(R)|\mathcal{C}(R))}\,\mathbb{E}[\mathsf{time}(\mathcal{A}(C))]\,,
\end{aligned}$$

where we use that $\kappa(\mathcal{C}(R) \mid \mathcal{U}(R)) = \Pr[C \in \mathcal{C}(R)\setminus\mathcal{U}(R) \mid C \in \mathcal{C}(R)]$ in the first equality, and we use $\mathsf{time}(\mathcal{A}(c)) \geq 0$ for the inequality. Now, recall that $\kappa(\mathcal{U}) = \max_{T\in\mathsf{IncompChains}(\mathcal{U})}\kappa(\mathcal{U}(T) \mid \mathcal{C}(T))$, and therefore $\frac{1}{1-\kappa(\mathcal{U})} \geq \frac{1}{1-\kappa(\mathcal{U}(R)|\mathcal{C}(R))}$. Finally, the sum over all prefixes $R \preceq T$, contains at most $\mathsf{depth}(\mathcal{U})$ terms, each bounded by $\frac{1}{1-\kappa(\mathcal{U})}\cdot\mathbb{E}[\mathsf{time}(\mathcal{A}(C))]$. This proves the claim. □

The following theorem shows that $\mathcal{U}$-adaptive special soundness implies knowledge soundness. It is a direct consequence of Lemma C.20 and Corollary C.21.

Theorem 4.5 (AdSS $\implies$ KS). *Let $\mathcal{R} : \mathbb{X} \times \mathbb{W} \to \{0,1\}$ be a relation. Let Π be a public-coin interactive proof system with a challenge set $\mathfrak{C}$. Let $\mathcal{U} : \mathbb{X} \times \mathcal{T}_\Pi \to 2^{\mathfrak{C}}$ and $\mathcal{W} : \mathbb{X} \times \mathcal{T}_\Pi \to \mathbb{W}$ be deterministic polynomial time algorithms. Suppose that $\mathcal{U}$ is natural (Definition 4.4) and Π is $(\mathcal{U}, \mathcal{W})$-adaptive special sound for $\mathcal{R}$ (Definition 4.2). Then Π is knowledge sound for $\mathcal{R}$ with knowledge error $\kappa = \kappa(\mathcal{U})$. Specifically, there exists a knowledge extractor $\mathcal{E}$ with runtime overhead $\frac{\mathsf{depth}(\mathcal{U})}{1-\kappa(\mathcal{U})}$ (Definition 4.3) such that for all statement $\mathsf{stmt} \in \mathbb{X}$ and all prover $\mathcal{P}^*$ it succeeds with probability*

$$\Pr\big((\mathsf{stmt}; \mathcal{E}^{\mathcal{P}^*}(\mathsf{stmt})) \in \mathcal{R}\big) \geq \frac{\epsilon(\mathcal{P}^*, \mathsf{stmt}) - \kappa(\mathcal{U})}{1 - \kappa(\mathcal{U})}.$$

Proof. By linearity of expectation, we can w.l.o.g. assume a deterministic malicious prover $\mathcal{P}^*$. Such a prover induces an abstract adversaries $\mathcal{A}$ as follows: For $c \in \mathfrak{C}$, $\mathcal{A}(c)$ outputs (a, c, z), where $\mathcal{A}(c)$ is the fixed first message a and z is $\mathcal{P}^*$'s response to c. If $\mathsf{V}(a, c, z) = 0$, i.e., if $\mathcal{V}$ would not accept the transcript, then $\mathcal{A}(c)$ outputs $\perp$ instead. Thus, $\mathcal{A}$ is V-respecting which means it only outputs accepting transcripts. By $\mathcal{U}$-adaptive special soundness, there exists an efficient witness extractor $\mathcal{W}$ which outputs a witness whenever its input is a complete $\mathcal{U}$-chain T (i.e., $\mathcal{U}(T) = \emptyset$) of accepting transcripts. By Corollary C.21, we obtain such a $\mathcal{U}$-chain T in expected polynomial time and with knowledge error $\kappa(\mathcal{U})$. This completes the proof. □

Remark C.22 (Large knowledge error). For large knowledge error $\kappa(\mathcal{U}) \approx 1$, the runtime bound of Lemma C.20 may be dominated by a few summands. From this, slightly more precise bounds than those of Corollary C.21 can be derived . E.g., for N-out-of-N special soundness, one gets the bound $N(\ln(N) + 1)$ instead of N^2.

D AND-Composition of Abstract Extractors

In this section, we specify and analyse the AND-composition of extractors. That is, given extractors $\mathcal{E}_1, \ldots, \mathcal{E}_n$ (for relations $\mathcal{R}_1, \ldots, \mathcal{R}_n$), we want to successfully extract for all of them. Moreover, we want *all* of them to use the *same initial challenge* c.[20] Hence, a natural approach is to define the AND-composition extractor $\mathcal{E}_{\mathsf{AND}} = \mathsf{AND}[\mathcal{E}_1, \ldots, \mathcal{E}_n]$ is by running each of them (on same initial c and same adversary $\mathcal{A}$) and returning the tuple of witnesses, or $\perp$ in case of failure. This is defined in pseudo-code in Fig. 6.

Parameters: a (multi-round) challenge set $\mathfrak{C}$, and a (uniform) family of extractors $\mathcal{E}_1, \ldots, \mathcal{E}_n$.
$\mathcal{E}^{\mathcal{A}}_{\mathsf{AND}}(\mathsf{stmt}, c) \coloneqq \mathsf{AND}[\mathcal{E}_1, \ldots, \mathcal{E}_n]$:

- If c is not given as input, sample $c \leftarrow_{\$} \mathfrak{C}$. (Else use given c.)
- For $i = 1, \ldots, n$
  - $\mathsf{wit}_i \leftarrow \mathcal{E}^{\mathcal{A}}_i(\mathsf{stmt}, c)$.
  - If $\mathsf{wit}_i = \perp$ return $\perp$
- Return $\mathsf{wit} = (\mathsf{wit}_1, \ldots, \mathsf{wit}_n)$.

Fig. 6. AND-Composition of knowledge extractors. By convention, the extractors $\mathcal{E}_1, \ldots, \mathcal{E}_n$ output $\perp$ if $(\mathsf{stmt}, \mathsf{wit}) \notin \mathcal{R}_i$, that is, $\mathsf{V}^{\mathsf{post}}_i(\mathsf{stmt}, \mathsf{wit}_i) \coloneqq [(\mathsf{stmt}, \mathsf{wit}) \in \mathcal{R}_i]$.

Recall that, we formally capture the validity of outputs of adversaries and extractors via predicates. E.g., for $\tau \leftarrow \mathcal{A}(c)$ the usual $\mathsf{V}^{\mathsf{pre}}$ ensures that the Σ-protocol verifier would accept (stmt, τ). As we handle arbitrary extractors, we consider their outputs to be witnesses wit_i in Fig. 6. Of course, wit_i may be a list (or a tree) of transcript, e.g., given k-special soundness extractors.

Lemma D.1 (Correctness). *Let $\mathcal{E}_i$ be a knowledge extractor that is $(\mathsf{V}^{\mathsf{pre}}_i, \mathsf{V}^{\mathsf{post}}_i)$-compatible, for $i = 1, \ldots, n$, respectively. Let $\mathcal{E}_{\mathsf{AND}} = \mathsf{AND}[\mathcal{E}_1, \ldots, \mathcal{E}_n]$ be defined as in Fig. 6. Then $\mathcal{E}_{\mathsf{AND}}$ is $(\mathsf{V}^{\mathsf{pre}}, \mathsf{V}^{\mathsf{post}})$-compatible, where $\mathsf{V}^{\mathsf{pre}}(\mathsf{stmt}, \tau) = \bigwedge_{i=1}^n \mathsf{V}^{\mathsf{pre}}_i(\mathsf{stmt}, \tau)$ and $\mathsf{V}^{\mathsf{post}}(\mathsf{stmt}, (\mathsf{wit}_1, \ldots, \mathsf{wit}_n)) = \bigwedge_{i=1}^n \mathsf{V}^{\mathsf{post}}_i(\mathsf{stmt}, \tau_i)$.*

[20] Fixing the initial challenge c is not necessary in all applications of AND-composition, but it is often convenient and only strengthens the result. For example, it is needed for CWSS, see also Remark D.4.

The proof of Lemma D.1 is straightforward and omitted. We note that $\mathsf{V}^{\mathsf{post}}$ in Lemma D.1 does not assert that all wit_i were derived from extraction with the same initial challenge. However, if the wit_i consist of transcripts, one may assert that all contain a (first) transcript with the same (initial) challenge within $\mathsf{V}^{\mathsf{post}}$. This is clearly ensured by the construction in Fig. 6.

Next, we state our main lemma for AND-composition.

Lemma D.2. *Let $\mathcal{E}_1, \mathcal{E}_2$ be knowledge extractors that are $(\mathsf{V}_i^{\mathsf{pre}}, \mathsf{V}_i^{\mathsf{post}})$-compatible, for $i = 1, 2$, respectively. Let $\mathcal{E}_{\mathsf{AND}} = \mathsf{AND}[\mathcal{E}_1, \mathcal{E}_2]$ be defined as in Fig. 6. Suppose that $\mathcal{E}_i$ has knowledge function f_i and runtime overhead ρ_i. Then $\mathcal{E}_{\mathsf{AND}}$ has knowledge function $f(\varepsilon) \geq f_1(\varepsilon) + f_2(\varepsilon) - \varepsilon$ and runtime overhead $\rho_1 + \rho_2$. In particular:*

- *If each $\mathcal{E}_i$ has additive-type knowledge function $f_i(\varepsilon) = \varepsilon - \kappa_i$, then $f(\varepsilon) = \varepsilon - \kappa$ with $\kappa = \kappa_1 + \kappa_2$.*
- *If each $\mathcal{E}_i$ has optimal-type knowledge function $f_i(\varepsilon) = \frac{\varepsilon - \kappa_i}{1 - \kappa_i}$, then $f(\varepsilon) = \frac{\varepsilon - \kappa}{1 - \kappa}$ with $\kappa = \frac{1 - (1 - \kappa_1)(1 - \kappa_2)}{1 - \kappa_1 \kappa_2}$. Moreover, $\kappa \leq \kappa_1 + \kappa_2$ if $\kappa_1 + \kappa_2 \leq 1$.*

Proof. The runtime overhead follows immediately by linearity of expectation. Let us now turn towards knowledge soundness. By an averaging argument, we can w.l.o.g. assume that $\mathcal{A}$ is deterministic. For $\mathsf{AND}[\mathcal{E}_1, \mathcal{E}_2]$ and a $\mathsf{V}^{\mathsf{pre}}$-respecting $\mathcal{A}$, we find for $C \leftarrow_{\$} \mathfrak{C}$

$$\Pr[\mathsf{AND}[\mathcal{E}_1, \mathcal{E}_2](C) \neq \bot] = \Pr[\mathcal{E}_1(C) \neq \bot \wedge \mathcal{E}_2(C) \neq \bot]$$

Define as a shorthand the events

$$E_1 = \{\mathcal{E}_1(C) \neq \bot\} \qquad E_2 = \{\mathcal{E}_2(C) \neq \bot\} \qquad A = \{\mathcal{A}(C) \neq \bot\}$$

and note that $\neg E_i = \{\mathcal{E}_i(C) = \bot\}$ by definition. We find that

$$\begin{aligned} \Pr[\mathsf{AND}[\mathcal{E}_1, \mathcal{E}_2](C) \neq \bot] &= \Pr[\mathcal{E}_1(C) \neq \bot \wedge \mathcal{E}_2(C) \neq \bot] \\ &= \Pr[E_1 \wedge E_2] \\ &= \Pr[E_1] + \Pr[E_2] - \Pr[E_1 \vee E_2] \end{aligned}$$

Since $\mathcal{E}_i$ is ordinary and $\mathcal{A}$ is deterministic, we know that $\mathcal{E}_i(c) \neq \bot \implies \mathcal{A}(c) \neq \bot$, and thus $E_1 \vee E_2 \subseteq A$. We find

$$\Pr[E_1] + \Pr[E_2] - \Pr[E_1 \vee E_2] \leq \Pr[E_1] + \Pr[E_2] - \Pr[A] \leq f_1(\varepsilon) + f_2(\varepsilon) - \varepsilon$$

by definition of the knowledge functions f_1, f_2. Plugging in $f_i(\varepsilon) = \frac{\varepsilon - \kappa_i}{\alpha_i}$ yields

$$f(\varepsilon) = f_1(\varepsilon) + f_2(\varepsilon) - \varepsilon = \frac{1}{\alpha_1 \alpha_2} \cdot \big((\alpha_1 + \alpha_2 - \alpha_1 \alpha_2)\varepsilon - (\alpha_2 \kappa_1 + \alpha_1 \kappa_2)\big)$$

Hence, for additive-type f_i, where $\alpha_1 = \alpha_2 = 1$, we get

$$f_1(\varepsilon) + f_2(\varepsilon) - \varepsilon = \varepsilon - (\kappa_1 + \kappa_2).$$

For optimal-type, where $\alpha_i = 1 - \kappa_i$, we get

$$f(\varepsilon) = \frac{(1 - \kappa_1 \kappa_2)\varepsilon - \alpha_1 \alpha_2}{\alpha_1 \alpha_2} = \frac{\varepsilon - \kappa}{\kappa}$$

where

$$\kappa = \frac{1 - (1 - \kappa_1)(1 - \kappa_2)}{(1 - \kappa_1 \kappa_2)} = \frac{\kappa_1 + \kappa_2 - \kappa_1 \kappa_2}{(1 - \kappa_1 \kappa_2)} \leq \kappa_1 + \kappa_2$$

and final inequality holds if $\kappa_1 + \kappa_2 \leq 1$. □

From Lemma D.2, we obtain the following result.

Corollary D.3. *Let $\mathcal{E}_i$ be a knowledge extractor that is $(\mathsf{V}_i^{\mathsf{pre}}, \mathsf{V}_i^{\mathsf{post}})$-compatible, for $i = 1, \ldots, n$, respectively. Let $\mathcal{E}_{\mathsf{AND}} = \mathsf{AND}[\mathcal{E}_1, \ldots, \mathcal{E}_n]$ be defined as in Fig. 6. Suppose $\mathcal{E}_i$ has runtime overhead ρ_i. Then $\mathcal{E}_{\mathsf{AND}}$ has runtime overhead $\sum_{i=1}^{n} \rho_i$ and moreover:*

– *If each $\mathcal{E}_i$ has additive-type knowledge function $f_i(\varepsilon) = \varepsilon - \kappa_i$, then $\mathcal{E}_{\mathsf{AND}}$ has additive-type knowledge function $f(\varepsilon) = \varepsilon - \kappa$ with $\kappa = \sum_{i=1}^{n} \kappa_i$.*
– *If each $\mathcal{E}_i$ has optimal-type knowledge function $f_i(\varepsilon) = \frac{\varepsilon - \kappa_i}{1-\kappa_i}$, then $\mathcal{E}_{\mathsf{AND}}$ has optimal-type knowledge function $f(\varepsilon) = \frac{\varepsilon-\kappa}{1-\kappa}$ with $\kappa \leq \sum_{i=1}^{n} \kappa_i$, if $\kappa \leq 1$.*

Proof. By definition, $\mathsf{AND}[\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3] = \mathsf{AND}[\mathsf{AND}[\mathcal{E}_1, \mathcal{E}_2], \mathcal{E}_3]$, and analogously for $n > 3$. Hence, the claim follows easily by an inductive application of Lemma D.2. □

Remark D.4 (AND $\implies$ CWSS). Coordinate-wise special soundness [FMN24] can be captured by AND-composition of k-special soundness extractors $\mathcal{E}_i$ which only resample the i-th component of a challenge vector $(c_1, \ldots, c_n)$. Corollary D.3 recovers the guarantees from [FMN24, Lemma 7.1], and improves upon it by (1) preserving the optimal-type knowledge function of extractors and (2) not being limited to k-special soundness.

E Nested Composition of Abstract Extractors

In this section, we define and analyse nested composition. First, in Sections E.1 and E.2, we provide the necessary definitions for our nested composition. Then, in Section E.1 we present our abstract analysis of nested composition of knowledge extractors. While the abstraction leads to highly general results and simple proofs, it requires "boilerplate" transformations to capture the well-known "traditional" (round-based) composition within this framework. We discuss this in Section E.4. Finally, in Section E.5, we present specifically the setting of nested UCS and their knowledge soundness. For completeness, Section F provides examples which illustrate the "boilerplate" transformations and the curiosity that our composition does not per se need to correspond to some protocol round structure.

E.1 Nested Composition

In this section, we define nested composition of extractors which generalises traditional composition. The latter implicitly considers extractors $\mathcal{E}_i$ which expect oracle access to adversaries $\mathcal{A}_i \colon \mathfrak{C}_i \to \{0,1\}^* \cup \{\bot\}$. However, as motivated in the introduction (Section 2.5), our parent extractors must be able to choose the initial challenge of their nested child-extractors. For this, we should consider $\mathcal{A}_i \colon \mathfrak{C}_{[i:\mu]} \to \{0,1\}^* \cup \{\bot\}$. Instead, it is convenient to have all oracles defined over the whole challenge set $\mathfrak{C} = \mathfrak{C}_{[1:\mu]}$, instead of $\mathfrak{C}_{[i:\mu]}$. Now, applying an extractor $\mathcal{E}$ to an adversary

$$\mathcal{A} \colon \mathfrak{C} \to \{0,1\}^* \cup \{\bot\}$$

yields a new adversary

$$\mathcal{E}^{\mathcal{A}} \colon \mathfrak{C} \to \{0,1\}^* \cup \{\bot\}$$

over the same challenge space.[21] Thus, in the following, we consider extractors $\mathcal{E}_i$ with oracle access to $\mathcal{A} \colon \mathfrak{C} \to \{0,1\}^* \cup \{\bot\}$.

Note that extractors and adversaries have the same interface (except that the former is an oracle algorithm). This formal unification of adversary and extractor is very convenient. Recall that for abstract extractors (Definition C.4), we specify via $(\mathsf{V}^{\mathsf{pre}}, \mathsf{V}^{\mathsf{post}})$-compatibility, the pre-condition $\mathcal{E}$ requires on oracle outputs and the post-condition it asserts on its own outputs.

Definition E.1 (Nested composition). *Let $(\mathcal{E}_i)_{i=1}^{\mu}$ be abstract extractors over $\mathfrak{C}$ which are $(\mathsf{V}_i^{\mathsf{pre}}, \mathsf{V}_i^{\mathsf{post}})$-compatible. We call $(\mathcal{E}_i)_{i=1}^{\mu}$ **composable (w.r.t.** $(\mathsf{V}_i^{\mathsf{pre}}, \mathsf{V}_i^{\mathsf{post}})_{i=1,2}$**)** if $\mathsf{V}_{i+1}^{\mathsf{post}} \implies \mathsf{V}_i^{\mathsf{pre}}$ for $i = 1, \ldots, \mu - 1$. For composable $(\mathcal{E}_i)_{i=1}^{\mu}$, we define their **nested composition** $\mathcal{E}$ as follows:*

For some adversary $\mathcal{A}$, let $\mathcal{A}_\mu(c) = \mathcal{E}_\mu^{\mathcal{A}}(c)$ and $\mathcal{A}_i(c) = \mathcal{E}_i^{\mathcal{A}_{i+1}}(c)$, each defined over $\mathfrak{C}$. The composed extractor $\mathcal{E}^{\mathcal{A}}$ defined as $\mathcal{E}_1^{\mathcal{A}}$. Note that $\mathcal{E}_1$ is $(\mathsf{V}_\mu^{\mathsf{pre}}, \mathsf{V}_1^{\mathsf{post}})$-compatible by construction.

In the rest of this section, unless stated otherwise, we **always assume** that adversaries and extractors which are composed are **compatible and composable** and omit often $\mathsf{V}^{\mathsf{pre}}$ and $\mathsf{V}^{\mathsf{post}}$.

[21] Interestingly, we can now consider nested compositions which do not correspond to protocol rounds, and thus get a more general composition result. However, most applications of this generality we are aware of rely on a product structure on the (i-th) challenge set, say $\mathfrak{C}_i = \mathfrak{C}_i' \times \mathfrak{C}_i''$, and can be covered by introducing artificial dummy rounds (which split sampling and sending c_i into (c_i', c_i'')).

E.2 Additional Preliminaries

In the following, we consider a challenge set with structure $\mathfrak{C} \cong \mathfrak{C}' \times \mathfrak{C}''$, which means that up to an efficient bijection we have $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$. For simplicity, we will just write $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$ throughout.

Definition E.2 (Extractor families and challenge set extension). *Let $(\mathcal{E}_{c'})_{c' \in \mathfrak{C}'}$ be a family of extractors over $\mathfrak{C}''$. We extend $(\mathcal{E}_{c'})_{c' \in \mathfrak{C}'}$ to $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$ by having $\mathcal{E}$ pick $c' \leftarrow_{\$} \mathfrak{C}'$ uniformly and run $\mathcal{E}_{c_1}$.*

A $\mathfrak{C}'$-indexed family of extractors gives rise to a $\mathfrak{C}'$-fixing extractor.

Definition E.3 ($\mathfrak{C}'$-fixing extractor). *Let $\mathcal{E}$ be an extractor over $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$. We say $\mathcal{E}$ is $\mathfrak{C}'$-fixing if for any execution of $\mathcal{E}^{\mathcal{A}}$, the queries which $\mathcal{E}$ makes to $\mathcal{A}$ lie in a subset $\{c'\} \times \mathfrak{C}''$ for some $c' \in \mathfrak{C}'$. In other words, if the first query to $\mathcal{A}$ is $c = (c', c'')$, then all subsequent queries are of the form $(c', \cdot)$.*

Remark E.4. Let $(\mathcal{E}_{c'})_{c' \in \mathfrak{C}'}$ and $\mathcal{E}$ be as in Definition E.2. Then $\mathcal{E}$ is $\mathfrak{C}'$-fixing. Conversely, any $\mathfrak{C}'$-fixing extractor $\mathcal{E}$ defines a family $(\mathcal{E}_{c'})_{c' \in \mathfrak{C}'}$ of extractors over $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$ in the obvious way.

To argue about success of nested compositions, we require the very natural property of *uniform initial challenges*.

Definition E.5. *Let $\mathcal{E}$ be an extractor over challenge space $\mathfrak{C}$. We say $\mathcal{E}$ has **uniform initial challenge** if the first query $\mathcal{A}(c)$ of $\mathcal{E}$ always uses a uniformly random $c \leftarrow_{\$} \mathfrak{C}$ and if $\mathcal{A}(c) = \perp$ for this first query, then $\mathcal{E}$ returns $\perp$.*[22]

More generally, let $(\mathcal{E}_{c'})_{c' \in \mathfrak{C}'}$ be a family of extractors over $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$. We say $(\mathcal{E}_{c'})_{c' \in \mathfrak{C}'}$ has runtime overhead ρ f if every $\mathcal{E}_{c'}$ has (for $c' \in \mathfrak{C}'$).

Remark E.6 (Runtime overhead for families). Suppose $(\mathcal{E}_{c'})_{c' \in \mathfrak{C}'}$ is a family of extractors over $\mathfrak{C}''$ with runtime overhead ρ. Consider its extension $\mathcal{E}$ to $\mathfrak{C}$ as in Definition E.2 that is, $\mathcal{E}$ picks $c' \leftarrow_{\$} \mathfrak{C}'$ and fixes it. Then $\mathcal{E}$ has runtime overhead ρ. (This follows, since the expectation over c' can be replaced by maximum, and each $\mathcal{E}_{c'}$ has runtime overhead ρ.)

E.3 Generic Success Bounds for Nested Composition

In this section, we prove completely generic bounds for our nested composition. We will later see how to recover the bounds for traditional composition from these. We extend the notion of knowledge functions from Definition 3.4 to abstract extractors

Definition E.7 (Knowledge function). *Let $\mathcal{E}$ be an extractor over challenge space $\mathfrak{C}$, and let $f\colon [0,1] \to \mathbb{R}$ be monotonically increasing. We say $\mathcal{E}$ has **knowledge function** f if, for every adversary $\mathcal{A}$, we have*

$$\Pr[\mathcal{E}^{\mathcal{A}} \neq \perp] \geq f(\Pr[\mathcal{A}(C) \neq \perp \mid C \leftarrow_{\$} \mathfrak{C}])\,.$$

For a family of extractors $(\mathcal{E}_{c'})_{c' \in \mathfrak{C}'}$ over $\mathfrak{C}''$, we let f be a knowledge function *of $(\mathcal{E}_{c'})_{c' \in \mathfrak{C}'}$ if f is a knowledge function of $\mathcal{E}_{c'}$ for every $c' \in \mathfrak{C}'$*

For convex f, the extension $\mathcal{E}$ (over $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$) of $(\mathcal{E}_{c'})_{c' \in \mathfrak{C}'}$ inherits f.

Lemma E.8 (Convex knowledge functions are preserved by extensions). *Suppose $(\mathcal{E}_{c'})_{c' \in \mathfrak{C}'}$ is a family of extractors over $\mathfrak{C}''$ with knowledge function f. Let $\mathcal{E}$ denote the extension to $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$ as in Definition E.2, that is, $\mathcal{E}$ picks $c' \leftarrow_{\$} \mathfrak{C}'$ and fixes it. Suppose f in* convex. *Then $\mathcal{E}$ has knowledge function f (as an extractor over $\mathfrak{C}$). Namely, we have*

$$\begin{aligned}\Pr[\mathcal{E}^{\mathcal{A}} \neq \perp] &= \mathop{\mathbb{E}}_{C'}[\Pr[\mathcal{E}_{C'}^{\mathcal{A}(C',\cdot)} \neq \perp]] \geq \mathop{\mathbb{E}}_{C'}[f(\Pr_{C''}[\mathcal{A}(C', C'') \neq \perp])] \\ &\geq f(\mathop{\mathbb{E}}_{C'}[\Pr_{C''}[\mathcal{A}(C', C'') \neq \perp]]) = f(\Pr_{C}[\mathcal{A}(C) \neq \perp])\end{aligned}$$

where C', C'', C are uniformly random in $\mathfrak{C}'$, $\mathfrak{C}''$, and $\mathfrak{C}$, respectively.

[22] Since we only consider *ordinary* extractors (Definition C.4), the second property always holds.

The proof of Lemma E.8 applies Jensen's inequality to swap expectation and f. Next is our main result for nesting two extractors. By induction, any number of extractors can be composed.

Theorem E.9. *Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be extractors over challenge space $\mathfrak{C}$ with knowledge functions f_1, f_2 and runtime overhead ρ_1, ρ_2. Suppose $\mathcal{E}_2$ has uniform initial challenge. Let $\mathcal{E}$ be the nested composition of $\mathcal{E}_1$ and $\mathcal{E}_2$, that is $\mathcal{E}^{\mathcal{A}} \coloneqq \mathcal{E}_1^{\mathcal{E}_2^{\mathcal{A}}}$. Then $\mathcal{E}$ has knowledge error $(f_1 \circ f_2)(x) = f_1(f_2(x))$ and runtime overhead $\rho_1 \cdot \rho_2$.*

Proof. By definition of $\mathcal{E}$ as $\mathcal{E}_1^{\mathcal{A}_2}$ with $\mathcal{A}_2(c) = \mathcal{E}_2(c)$, and the knowledge function of $\mathcal{E}_i$, we obtain

$$\Pr[\mathcal{E}^{\mathcal{A}} \neq \bot] = \Pr[\mathcal{E}_1^{\mathcal{A}_2} \neq \bot] \geq f_1(\Pr[\mathcal{A}_2(C) \neq \bot]) \geq f_1(f_2(\Pr[\mathcal{A}(C) \neq \bot]))$$

where C denotes a uniformly random challenge in $\mathfrak{C}$.

For the runtime overhead, let $\mathcal{A}_1 = \mathcal{E}_2^{\mathcal{A}}$. By definition, we have

$$\mathop{\mathbb{E}}_{c \leftarrow_{\$} \mathfrak{C}}[\mathsf{time}_{\mathcal{A}}(\mathcal{E}_1^{\mathcal{A}_1})] \leq \rho_1 \cdot \mathop{\mathbb{E}}_{c \leftarrow_{\$} \mathfrak{C}}[\mathsf{time}_{\mathcal{A}}(\mathcal{A}_1)] \leq \rho_1 \cdot \rho_2 \cdot \mathop{\mathbb{E}}_{c \leftarrow_{\$} \mathfrak{C}}[\mathsf{time}(\mathcal{A})] \qquad \square$$

Finally, let us take a look at some examples of knowledge functions.

Example E.10 (Optimal-type). In Corollary C.21 we show that our UCS extractors have knowledge function $f(x) = \frac{x-\kappa(\mathcal{U})}{1-\kappa(\mathcal{U})}$, which is also the knowledge function of [ACK21] and optimal for k-special soundness. The composition law for $f_\kappa(x) = \frac{x-\kappa}{1-\kappa}$ is $f_{\kappa_1} \circ \ldots \circ f_{\kappa_n} = f_\kappa$ where $\kappa = 1 - \prod_{i=1}^{\mu}(1-\kappa_i)$.

Example E.11 (Additive-type). A natural choice of knowledge function is $g(x) = x - \kappa(\mathcal{U})$. The composition law for $g_\kappa(x) = x - \kappa$ is just addition, i.e. $\kappa = \sum_i \kappa_i$

In both of the above examples, we recover the known success bounds for traditional composition. Next is an example where our generic approach is insufficient.

Example E.12. The knowledge function from [AFR23] is $h(x) = \frac{x-\kappa_\Gamma}{t_\Gamma \cdot (1-\kappa_\Gamma)}$ where t_Γ is a value which depends on the access structure Γ. In this case, the generic composition would yield a bound which is worse than the bound for composition proved in [AFR23]. Indeed, Attema, Fehr, and Resch [AFR23] require a different, non-generic analysis to prove their bounds.

E.4 Multi-round protocols, Traditional and Nested Composition

Let us now apply our results to recover and generalise the "traditional" extractor composition of $(2\mu+1)$-move protocol obtained through sequential application of μ Σ-protocols (e.g., as defined in [AC20]). Let $\mathfrak{C}_i$ denote the challenge space of the i-th challenge. For traditional composition, one considers the extractor $\widetilde{\mathcal{E}}_i$ which is defined over $\mathfrak{C}_i$, the challenge set of the i-th round.[23] That is, $\widetilde{\mathcal{E}}_i$ expects an adversary $\widetilde{\mathcal{A}}_i$ defined over $\mathfrak{C}_i$ which outputs a (partial) witness, or in the case of tree-special soundness, a tree of transcripts.

For nested composition, recall that our extractors are always defined over the whole challenge space $\mathfrak{C} = \mathfrak{C}_1 \times \ldots \times \mathfrak{C}_\mu$. We now describe how to "extend" an extractor $\widetilde{\mathcal{E}}_i$ appropriately.

Step 1: Extension to $\mathfrak{C}_{[i:\mu]} = \mathfrak{C}_i \times \ldots \times \mathfrak{C}_\mu$. As a first step, we extend $\widetilde{\mathcal{E}}_i$ and $\widetilde{\mathcal{A}}_i$ from $\mathfrak{C}_i$ to $\mathfrak{C}_{[i:\mu]}$ and call the resulting extractor $\widehat{\mathcal{E}}_i$. To extend the adversary, we simply set $\widehat{\mathcal{A}}_i(c_i, \ldots, c_\mu) = \widetilde{\mathcal{A}}_i(c_i)$ for all $(c_i, \ldots, c_\mu) \in \mathfrak{C}_{[i:\mu]}$. That is, $\widehat{\mathcal{A}}_i$ simply ignores all later challenges. We define $\widehat{\mathcal{E}}_i$, given access to some $\widehat{\mathcal{A}}\colon \mathfrak{C}_{[i:\mu]} \to \mathcal{T} \cup \{\bot\}$, as follows:

- Sample $(c_{i+1}, \ldots, c_\mu) \leftarrow_{\$} \mathfrak{C}_{[i+1:\mu]}$.
- Let $\widetilde{\mathcal{A}}_i(C_i) \coloneqq \widehat{\mathcal{A}}_i(C_i, c_{i+1}, \ldots, c_\mu)$ for $C_i \in \mathfrak{C}_i$.
- Run and output $\widetilde{\mathcal{E}}_i^{\widetilde{\mathcal{A}}_i}$.

[23] As a consequence, in traditional composition, the parent cannot does not choose the child's first challenge, which is why we needed nested composition to being with.

We observe that this extension of $\widehat{\mathcal{A}}_i$ and $\widehat{\mathcal{E}}_i$ from $\widetilde{\mathcal{A}}_i$ and $\widetilde{\mathcal{E}}_i$ satisfies the property that the machines $\widetilde{\mathcal{E}}_i^{\widetilde{\mathcal{A}}_i}$ and $\widehat{\mathcal{E}}_i^{\widehat{\mathcal{A}}_i}$ behave identically (except for sampling $c_{i+1}, \ldots, c_\mu$ in $\widehat{\mathcal{E}}_i$ which are ignored by $\widehat{\mathcal{A}}_i$, but this has only a minor effect on runtime and randomness complexity).

Therefore, we can w.l.o.g. assume we have $\widetilde{\mathcal{E}}_i$ defined over $\mathfrak{C}_{[i:\mu]}$ by working with $\widehat{\mathcal{E}}_i$ instead. Note that, being defined over $\mathfrak{C}_{[i:\mu]}$, this extractor already allows the parent to choose all later challenges of the initial invocation of it child. In fact, for our application to lattice-based Bulletproof (Section G) and most non-trivial applications of (nested) adaptive special soundness, this is the type of extractor one defines.

Step 2: Extension to $\mathfrak{C} = \mathfrak{C}_{[1:\mu]}$. Starting with $\widehat{\mathcal{E}}_i$ defined over $\mathfrak{C}_{[i:\mu]}$, we can treat them as (constant) families over $\mathfrak{C}_{[1:i-1]}$. From Lemma E.8, we know that considering the extension $\mathcal{E}_i$ of $\widehat{\mathcal{E}}_i$ to all of $\mathfrak{C}$, defined by sampling and fixing $(c_1, \ldots, c_{i-1})$, preserves the knowledge error. Similarly, we can extend an adversary $\widehat{\mathcal{A}}_i$ from $\mathfrak{C}_{[i:\mu]}$ to $\mathfrak{C} = \mathfrak{C}_{[1:\mu]}$ by setting $\mathcal{A}_i(c_1, \ldots, c_\mu) = \widehat{\mathcal{A}}_i(c_i, \ldots, c_\mu)$. Again, except for a minor effect on runtime and randomness complexity, the machines $\widehat{\mathcal{E}}_i^{\widehat{\mathcal{A}}_i}$ and $\mathcal{E}_i^{\mathcal{A}_i}$ behave essentially identically.[24]

Summary. Taken together, we see that extractor obtained from traditional composition can be "lifted" to our abstract extractors (defined over $\mathfrak{C}$) in such a way, that their efficiency and quality is preserved. Hence, nested composition generalises traditional composition. In Section F.1, we compare and illustrate the setting of (2,2,2)-special soundness in more detail.

E.5 Nested Composition of UCS extractors

In the following, we apply the generic composition results directly to nested UCSs. For this, let $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$ as before. Let $(\mathcal{U}_{c'})_{c' \in \mathfrak{C}'}$ be a family of UCSs over $\mathfrak{C}''$ and let $(\mathcal{E}_{\mathcal{U}_{c'}})_{c' \in \mathfrak{C}'}$ be the family of UCS-extractors (Lemma C.20). By abuse of notation, we write $\mathcal{U} = (\mathcal{U}_{c'})_{c' \in \mathfrak{C}'}$ and treat $\mathcal{U}$ as a UCS over $\mathfrak{C}$ when extraction is concerned. We generalise κ and depth to $\mathfrak{C}'$-indexed families in the obvious way.

Definition E.13. *Let $\mathcal{U} = (\mathcal{U}_{c'})_{c' \in \mathfrak{C}'}$ be a family of UCSs over $\mathfrak{C}''$. Then*

$$\kappa_{\mathfrak{C}'}(\mathcal{U}) = \max_{c' \in \mathfrak{C}'} \kappa(\mathcal{U}_{c'})$$
$$\mathsf{depth}_{\mathfrak{C}'}(\mathcal{U}) = \max_{c' \in \mathfrak{C}'} \mathsf{depth}(\mathcal{U}_{c'})$$

Lemma E.14. *Let $\mathcal{U} = (\mathcal{U}_{c'})_{c' \in \mathfrak{C}'}$ be a family of UCS over $\mathfrak{C}''$. Let $\mathcal{E}_{\mathcal{U}}$ be the abstract UCS extractor extended to $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$. Then $\mathcal{E}_{\mathcal{U}}$ has knowledge function $f(x) = \frac{x - \kappa_{\mathfrak{C}'}(\mathcal{U})}{1 - \kappa_{\mathfrak{C}'}(\mathcal{U})}$ and runtime overhead $\frac{1}{1-\kappa_{\mathfrak{C}'}(\mathcal{U})}\mathsf{depth}_{\mathfrak{C}'}(\mathcal{U})$.*

Proof. This is an immediate consequence of Corollary C.21 and Theorem E.9.

Remark E.15 (Aside: $\mathfrak{C}'$-fixing UCS). We justify our abuse of notation. Any family $(\mathcal{U}_{c'})_{c' \in \mathfrak{C}'}$ of UCS over $\mathfrak{C}'$ induces a UCS $\mathcal{U}$ over $\mathfrak{C}$ as follows: $\mathcal{U}(\emptyset) = \mathfrak{C}$ and $\mathcal{U}(\tau_1, \ldots, \tau_i) = \{c_1'\} \times \mathcal{U}_{c_1'}(\tau_1, \ldots, \tau_i)$ where $(c_1', c_1'') = \mathsf{challof}(\tau_1) \in \mathfrak{C}' \times \mathfrak{C}''$. In other words, $\mathcal{U}$ fixes the $\mathfrak{C}'$-component throughout all transcripts, and we call a UCS with that property $\mathfrak{C}'$-fixing. Conversely, a $\mathfrak{C}'$-fixing UCS induces a family $(\mathcal{U}_{c'})_{c' \in \mathfrak{C}'}$ via $\mathcal{U}_{c'}(\tau_1, \ldots, \tau_i) = \{c'' \mid (c', c'') \in \mathcal{U}(\tau_1, \ldots, \tau_i)\}$. This duality of $\mathfrak{C}'$-indexed families and $\mathfrak{C}'$-fixing of UCS is completely analogous to the extractor setting, cf. Remark E.4. Thus, our abuse of notation is justified.

It is easy to verify that $\kappa(\mathcal{U}) \geq 1 - \frac{1}{|\mathfrak{C}'|}$, hence typically $\kappa(\mathcal{U}) \gg \kappa_{\mathfrak{C}'}(\mathcal{U})$ except if $|\mathfrak{C}'| = 1$. Thus, the distinction is between κ for UCS and $\kappa_{\mathfrak{C}'}$ for $\mathfrak{C}'$-indexed family of UCSs is important. For depth, it does not matter since $\mathsf{depth}(\mathcal{U}) = \mathsf{depth}_{\mathfrak{C}'}(\mathcal{U})$ holds by construction.

[24] The astute reader might have noticed that $\mathfrak{C} = \mathfrak{C}_{[1:\mu]} \cong \mathfrak{C}' \times \mathfrak{C}_i$, where $\mathfrak{C}' = \mathfrak{C}_1 \times \ldots \times \mathfrak{C}_{i-1} \times \mathfrak{C}_{i+1} \times \ldots \times \mathfrak{C}_\mu$. Instead of taking steps 1 and 2 sequentially, one could take a big step 2. The resulting behaviour remains essentially equivalent.

$(\mathcal{U}_1,\dots,\mathcal{U}_\mu)$*-Nested Composition.* Now, we turn to the correctness requirements and their meaning for the UCS definitions. Let $\mathcal{A}$ be an abstract adversary over $\mathfrak{C}$ which is V-respecting. Let $(\mathcal{U}_i)_{i=1}^{\mu}$ be families of UCSs (where $(\mathcal{U}_{i,c'})_{c'\in\mathfrak{C}'_i}$ and $\mathfrak{C} = \mathfrak{C}'_i \times \mathfrak{C}''_i$).

Definition E.16 (Composition of UCS extractors). *Let $(\mathcal{U}_i)_{i=1}^{\mu}$ be (families of) UCSs as above. For $\mathcal{A}$, define $(\mathcal{E}_i)_{i=1}^{\mu+1}$ as follows: Let $\mathcal{E}_\mu = \mathcal{E}_{\mathcal{U}_\mu}$ and let $\mathcal{E}_i(c) = \mathcal{E}_{\mathcal{U}_i}^{\mathcal{E}_{i+1}^{\mathcal{A}}}(c)$, with each $\mathcal{E}_i$ defined over $\mathfrak{C}$. We call $\mathcal{E}_1$ the (final) composed extractor.*

Let $\mathcal{A}_i = \mathcal{E}_{i+1}^{\mathcal{A}}$ and $\mathcal{A}_{\mu+1} = \mathcal{A}$. Observe that, by construction, $\mathcal{E}_i$ outputs a nested sequence of transcripts of height $\mu - i + 1$. Hence, $\mathcal{U}_i$ expects nested $(\mathcal{U}_{i+1},\dots,\mathcal{U}_\mu)$-chains as "$\mathcal{A}_i$-transcripts". In other words, an "$\mathcal{A}_i$-transcript" for $\mathcal{U}_i$ is a subtree (i.e. $(\mathcal{U}_{i+1},\dots,\mathcal{U}_\mu)$-chain), not a basic $\mathcal{A}$-transcript. Furthermore, the challenge $\mathsf{challof}_{\mathcal{A}_{i+1}}(T)$ of such a nested chain is necessarily the challenge of the first transcript of the chain, because $\mathsf{challof}_{\mathcal{A}_i}(\mathcal{A}_i(c)) = c$ must be satisfied (unless $\mathcal{A}_i(c) = \bot$), cf. Remark C.5. We thus find:

Lemma E.17 (Correctness). *Let $(\mathcal{U}_1,\dots,\mathcal{U}_\mu)$ be UCSs over $\mathfrak{C}$. Let $\mathcal{A}$ be a V-respecting adversary. Let $\mathcal{E}_1$ denote the composed extractor of the sequential composition of $(\mathcal{E}_{\mathcal{U}_i})_{i=1}^{\mu}$. Then $\mathcal{E}_1$ outputs complete V-respecting nested $(\mathcal{U}_1,\dots,\mathcal{U}_\mu)$-chains. That is, $\mathcal{E}_1$ is $(\mathsf{V},\mathsf{V}_1^{\mathsf{post}})$-compatible, where $\mathsf{V}_1^{\mathsf{post}}(T) = 1$ if T is a complete V-respecting nested $(\mathcal{U}_1,\dots,\mathcal{U}_\mu)$-chain.*

The proof follows from a simple induction.

Proof. The case of $\mu = 1$ is immediate: By Lemma C.16, $\mathcal{E}_{\mathcal{U}}$ outputs a list $T = (\tau^1,\dots,\tau^k)$ of accepting transcripts with $\mathcal{U}(T) = \emptyset$, i.e. a complete $\mathcal{U}$-chain of height 1. Moreover, if $\mathcal{A}$ is V-respecting, then all τ^i satisfy $\mathsf{V}(\tau^i) = 1$, hence T is a complete V-respecting $\mathcal{U}$-chain.

For the induction, define $\mathsf{V}_i^{\mathsf{post}}(T)$ for a nested UCS-chain of height $\mu - i + 1$ by checking if T is a complete V-respecting nested $(\mathcal{U}_i,\dots,\mathcal{U}_\mu)$-chain. Define $\mathsf{V}_i^{\mathsf{pre}}(S) = \mathsf{V}_{i+1}^{\mathsf{post}}(S)$, with $\mathsf{V}_\mu^{\mathsf{pre}} = \mathsf{V}$. Then for all i, $\mathcal{E}_{\mathcal{U}_i}$ is $(\mathsf{V}_i^{\mathsf{pre}},\mathsf{V}_i^{\mathsf{post}})$-compatible; the proof is analogous to $\mu = 1$: By Lemma C.20, $\mathcal{E}_{\mathcal{U}_i}$ outputs $T = (S^1,\dots,S^k) \in \mathsf{Chains}(\mathcal{U}_i)$ with $\mathcal{U}_i(S^1,\dots,S^k) = \emptyset$. By $\mathsf{V}_i^{\mathsf{pre}}$ each S^j is a complete V-respecting nested $(\mathcal{U}_{i+1},\dots,\mathcal{U}_\mu)$-chain. Hence, T is a complete V-respecting nested $(\mathcal{U}_i,\dots,\mathcal{U}_\mu)$-chain. □

The following theorem is stated in generic terms. The typical round-based composition considers $\mathfrak{C} \coloneqq \mathfrak{C}_1 \times \dots \times \mathfrak{C}_\mu$ and families $\mathcal{U}_i$ indexed over $\mathfrak{C}'_i \coloneqq \mathfrak{C}_1 \times \dots \times \mathfrak{C}_{i-1}$ with adversaries over $\mathfrak{C}''_i \coloneqq \mathfrak{C}_i \times \dots \times \mathfrak{C}_\mu$. In the most basic case of composition, $\mathcal{U}_i$ depends only on $\mathfrak{C}_i$. That is, it is a constant family (i.e., essentially independent of $(c_1,\dots,c_{i-1})$), and it also "ignores" $c_{i+1},\dots,c_\mu$, see also Section E.4.[25]

Theorem E.18 (Knowledge Error and Run-Time Bounds for $(\mathcal{U}_1,\dots,\mathcal{U}_\mu)$-chains). *Let $\mathcal{U}_1,\dots,\mathcal{U}_\mu$ be families of UCS, where $\mathcal{U}_i$ is $\mathfrak{C}'_i$-indexed over $\mathfrak{C}''_i$ and $\mathfrak{C} = \mathfrak{C}'_i \times \mathfrak{C}''_i$. Let $\mathcal{E}_1$ be the final composed extractor (as in Lemma E.17). The success of $\mathcal{E}_1$ satisfies*

$$\Pr[\mathcal{E}_1^{\mathcal{A}} \neq \bot] \geq \frac{\varepsilon(\mathfrak{C}|\mathfrak{C}) - \kappa}{1-\kappa} \qquad \textit{where} \qquad \kappa = 1 - \prod_{i=1}^{\mu}(1 - \kappa_{\mathfrak{C}'_i}(\mathcal{U}_i))\,.$$

The expected time $\mathcal{E}_1$ spends in $\mathcal{A}$ is bounded by

$$\mathbb{E}\big[\mathsf{time}_{\mathcal{A}}(\mathcal{E}_1^{\mathcal{A}})\big] \leq \frac{1}{1-\kappa} \cdot \prod_{i=1}^{\mu} \mathsf{depth}_{\mathfrak{C}'_i}(\mathcal{U}_i) \cdot \mathop{\mathbb{E}}_{C \leftarrow\!\!\$\, \mathfrak{C}} \big[\mathsf{time}(\mathcal{A}(C))\big]\,.$$

Proof. This is an immediate consequence of Lemma E.17 and Theorem E.9. Indeed, by Lemma E.17 the knowledge function of $\mathcal{E}_{\mathcal{U}_i}$ is $f_{\kappa_i}(x) = \frac{x-\kappa_i}{1-\kappa_i}$, and the runtime overhead is $\rho_i = \frac{1}{1-\kappa_i}\mathsf{depth}_{\mathfrak{C}'_i}(\mathcal{U}_i)$ where $\kappa_i = \kappa_{\mathfrak{C}'_i}(\mathcal{U}_i)$. By Theorem E.9, the composition of the $\mathcal{E}_i$ then yields the claimed parameters. □

It is now immediate that adaptive special sound protocols are proofs of knowledge with extractors whose runtime and success is bounded as in Theorem E.18. For completeness, we state it below.

[25] $\mathcal{U}_i(\tau_1,\dots,\tau_\ell) = \mathcal{U}'_i(\tau_1,\dots,\tau_\ell) \times \mathfrak{C}_{i+1} \times \dots \times \mathfrak{C}_\mu$, where $\mathcal{U}'_i$ is a UCS defined over $\mathfrak{C}_i$, i.e. for adversaries $\mathcal{A}' \colon \mathfrak{C}_i \to \mathcal{T} \cup \{\bot\}$.

Theorem 4.12 (μ-Nested AdSS $\implies$ KS). *Let $\mathcal{R} : \mathbb{X} \times \mathbb{W} \to \{0,1\}$ be a relation. Let Π be a public-coin interactive proof system with a challenge set $\mathfrak{C} \cong \mathfrak{C}_1 \times \ldots \times \mathfrak{C}_\mu$. Let $\mathcal{U} = (\mathcal{U}_i)_{i=1}^{\mu}$ where $\mathcal{U}_i : \mathbb{X} \times \mathcal{T}_\Pi^{(\mu-i+1)} \to 2^{\prod_{j=i}^{\mu} \mathfrak{C}_j}$ and $\mathcal{W} : \mathbb{X} \times \mathcal{T}_\Pi^{(\mu)} \to \mathbb{W}$ be deterministic polynomial time algorithms. Suppose that $\mathcal{U}$ is natural (Definition 4.11) and Π is $(\mathtt{NEST}[\mathcal{U}], \mathcal{W})$-adaptive special sound for $\mathcal{R}$ (Definition 4.9). Then Π is knowledge sound for $\mathcal{R}$ with knowledge error $\kappa = \kappa(\mathcal{U})$. Specifically, there exists a knowledge extractor $\mathcal{E}$ with runtime overhead $\frac{\mathsf{depth}(\mathcal{U})}{1-\kappa(\mathcal{U})}$ (Definition 4.10) such that for all statement $\mathsf{stmt} \in \mathbb{X}$ and all prover $\mathcal{P}^*$ it succeeds with probability*

$$\Pr\big((\mathsf{stmt}; \mathcal{E}^{\mathcal{P}^*}(\mathsf{stmt})) \in \mathcal{R}\big) \geq \frac{\epsilon(\mathcal{P}^*, \mathsf{stmt}) - \kappa(\mathcal{U})}{1 - \kappa(\mathcal{U})}.$$

F Examples of Nested Composition of Extractors

We give examples of nested composition of extractors, in particular demonstrating that it is not less expressive than traditional sequential composition.

F.1 (Tree-)Special Soundness

We want to mimic, in our generalised setting, the "traditional" composition of three extractors $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$ for 2-special soundness in 3 challenge rounds. Here, $\mathcal{E}_i$ is an extractor which finds 2 correlated transcripts in challenge round i. First note that usually, $\mathcal{E}_i$ is only defined for $\mathfrak{C}_i$. We need to define it for $\mathfrak{C} = \mathfrak{C}_1 \times \mathfrak{C}_2 \times \mathfrak{C}_3$.

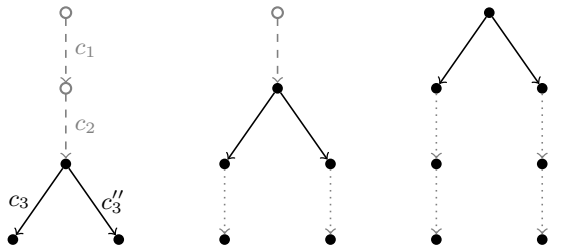


Fig. 7. 2-special sound tree extraction in 3 challenge rounds. From left to right, the trees represent the output tree structure of $\mathcal{E}_3$, $\mathcal{E}_2$, $\mathcal{E}_1$. Dashed edges illustrate an extension. Dotted edges illustrate a difference to extractors in traditional (nested) composition, where the extractor does not choose (or know of) this challenge.

It is instructive to consider tree-special soundness [BCCGP16] here: Indeed, we can view extractor $\mathcal{E}_3$ as outputting a $(1,1,2)$-tree, $\mathcal{E}_2$ a $(1,2,1)$-tree, and $\mathcal{E}_1$ a $(2,1,1)$-tree. This is illustrated in Fig. 7. Intuitively, there are two steps of "extensions" made to the typical extractors for (tree-)special soundness. Concretely, for $\mathcal{E}_2$:

- In the traditional composition, the adversary that $\mathcal{E}_2$ would operate on is $\mathcal{E}_3^{\mathcal{A}} : \mathfrak{C}_1 \times \mathfrak{C}_2 \to \mathcal{T}^{(1)}$. Since $\mathcal{E}_3$ has consumed $\mathfrak{C}_3$, this adversary is defined over $\mathfrak{C}_1 \times \mathfrak{C}_2$ (i.e. $\mathfrak{C}_3$ is not exposed anymore). However, with our nested composition, $\mathcal{E}_3^{\mathcal{A}}$ is again defined over all of $\mathfrak{C}$. In particular, $\mathcal{E}_2$ must choose c_3 (and c_1, which we handle later). By definition, we let $\mathcal{E}_2$ pick c_3 uniformly from $\mathfrak{C}_3$ in every query. This extends $\mathcal{E}_2$ to $\mathfrak{C}_2 \times \mathfrak{C}_3$.
- By our convention, we need to further extend $\mathcal{E}_2$ to all of $\mathfrak{C}$, which is done as in Section E.4, namely, $\mathcal{E}_2(c_1, c_2, c_3)$ fixes c_1 in all of its calls to $\mathcal{A}$.

We see that in this very simple case, extension to the whole challenge space is straightforward. In Section G, we encounter the slightly more complicated case, where $\mathcal{E}_2$ actually picks challenges (c_2, c_3) which satisfy certain predicates, and thus must be aware of $\mathfrak{C}_3$ already.

Overall, we see that our notion and conventions essentially make us view an extractor $\mathcal{E}_i$ for k-special soundness in the i-th round as an extractor for $(1, \ldots, 1, k, 1, \ldots, 1)$-special soundness over the complete challenge space $\mathfrak{C}$.

F.2 Self-Composition with 2-Special Soundness

Let $\mathcal{E}_{\text{2-ss}}$ be an extractor for two-special soundness over $\mathfrak{C}$ (i.e., no round structure) and consider its nested self-composition. That is, consider the extractor $\mathcal{E}^{\mathcal{A}}(c) = \mathcal{E}_{\text{2-ss}}^{\mathcal{E}_{\text{2-ss}}^{\mathcal{A}}}(c)$. While this construction is not useful, it helps illustrate nested composition.

By Remark C.5, the definition of $\mathsf{challof}_{\mathcal{A}_2}(\,\cdot\,)$ for $\mathcal{A}_2(c) \coloneqq \mathcal{E}_{\text{2-ss}}^{\mathcal{A}}(c)$ is given as $\mathsf{challof}_{\mathcal{A}_2}((\tau, \tau')) = \mathsf{challof}_{\mathcal{A}}(\tau)$. With this in mind, we take a closer look at the output of $\mathcal{E}$, which is a $(2,2)$-regular nested sequence of transcripts (i.e., a $(2,2)$-regular tree) that we denote by $T = ((\tau^1, \tau^2), (\tau^3, \tau^4))$, see also Fig. 8. Let (c^1, c^3) denote the accepting challenges found by the parent extractor (which are input to the child extractor), and (c^1, c^2, c^3, c^4) be the accepting challenges of the child extractor runs (in order). Perhaps surprisingly, there is no guarantee that all c^i are distinct. That is, by definition of $\mathcal{E}_{\text{2-ss}}$ applied to sub-extractor $\mathcal{A}_2$, we know that (c^1, c^3) are chosen to be distinct, and similarly, when applied to $\mathcal{A}$ (within subextractor calls $\mathcal{A}_2(c^1)$ and $\mathcal{A}_2(c^3)$) also (c^1, c^2) and (c^3, c^4), respectively. However, there is no assertion that (c^2, c^4) are distinct, since there is no immediate connection between them.

The illustrated behaviour differs from traditional (nested) composition (cf. Section F.1), where the round/product structure of $\mathfrak{C}$ and choice of extractors make each child extractor "consume" a challenge from a different round. In that case, it is ensured that all challenges $c^i = (c^i_1, \ldots, c^i_\mu) \in \mathfrak{C}_1 \times \ldots \times \mathfrak{C}_\mu$ are distinct, because for all $i \neq j$ there is one round ℓ where an extractor ensured $c^i_\ell \neq c^j_\ell$.

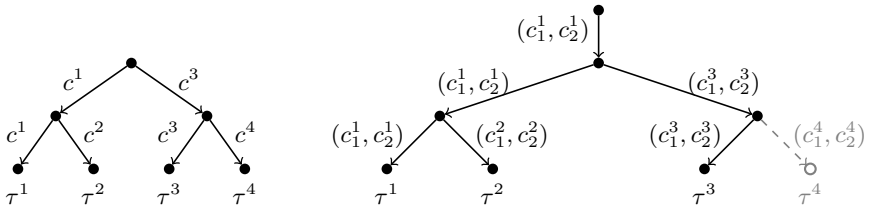


Fig. 8. Self-composition (left) and coordinate-wise special soundness (right): The graph illustrates the tree structure in which the transcripts are arranged. *Left:* The lower depth-2 edges are labelled with $\mathsf{challof}_{\mathcal{A}}(\tau^i)$ for the i-th leaf. The upper depth-1 edges are labelled with $\mathsf{challof}_{\mathcal{A}_2}((\tau^{2i-1}, \tau^{2i}))$ for the i-th subtree. *Right:* Analogous conventions are used, and additionally $c^2_2 = c^1_2$ and $c^3_1 = c^1_1$.

F.3 Coordinate-wise k-special soundness (CWSS) from nested composition

In this example, we show how coordinate-wise special soundness [FMN24] of a Σ-protocol can be modelled via nested composition (instead of AND-composition as in Section D). This demonstrates the additional flexibility provided by controlling the initial challenge of the subextractor in the nested setting.

We consider $(2,2)$-coordinate-wise special soundness, which means the challenge space is $\mathfrak{C} = \mathfrak{C}_1 \times \mathfrak{C}_2$, and we need to find accepting transcripts τ^1, τ^2, τ^3 whose challenges $c^i = (c^i_1, c^i_2)$ satisfy $c^2_2 = c^1_2$ and $c^3_1 = c^1_1$. That is, c^1 is the "root challenge" and c^2 (resp. c^3) differs in exactly one coordinate. Let us produce such transcripts via nested composition of 3 extractors:

- $\mathcal{E}_{\text{1-ss}}(c)$: This is an extractor for "1-special soundness", i.e., it just finds a single accepting transcript. It is used to determine the root challenge $c = c^1 \in \mathfrak{C}$.
- $\mathcal{E}_{\text{2-ss},\mathfrak{C}_1}((c_1, c_2))$, is an extractor for 2-special soundness, which extracts in the $\mathfrak{C}_1$-component but leaves the $\mathfrak{C}_2$-component c_2 fixed. Hence, it is $\mathfrak{C}_2$-fixing (Definition E.3)
- $\mathcal{E}_{\text{2-ss},\mathfrak{C}_2}$ is analogous to $\mathcal{E}_{\text{2-ss},\mathfrak{C}_1}((c_1, c_2))$, but is $\mathfrak{C}_1$-fixing.

Consider the composition $\mathcal{B}_3(c) = \mathcal{E}^{\mathcal{A}}_{\text{2-ss},\mathfrak{C}_1}(c)$, then $\mathcal{B}_2(c) = \mathcal{E}^{\mathcal{B}_3}_{\text{2-ss},\mathfrak{C}_2}(c)$, then $\mathcal{B}_1(c) = \mathcal{E}^{\mathcal{B}_2}_{\text{1-ss}}(c)$. The composed extractor is $\mathcal{B}_1$. It outputs a nested sequence of transcripts $(((\tau^1, \tau^2), (\tau^3, \tau^4)))$, where (τ^1, τ^2, τ^3) satisfies the requirements of $(2,2)$-CWSS. We note that the approach clearly generalises to $(k_1, \ldots, k_\ell)$-CWSS.

Remark F.1. Compared to tailor-made CWSS extractors [FMN24] or Section D, we observe a few differences: Firstly, an additional transcript τ^4 is generated, which is unneeded. Secondly, a negative consequence of the previous point is that, for $(2, \ldots, 2)$-CWSS, the runtime of this composition becomes

$\approx 2^\ell$ for ℓ coordinates, instead of $\ell + 1$ as in [FMN24]. Thirdly, as a curious positive consequence, the knowledge error is $1 - \prod_{i=1}^{\ell}(1 - \frac{1}{|\mathfrak{C}_i|})$ which is better than $\sum_{i=1}^{\ell} \frac{1}{|\mathfrak{C}_i|}$ as in [FMN24] or in Lemma D.2, especially for small $\mathfrak{C}$.

G Lattice-based Bulletproofs

The goal of this section is to prove that the (lattice-based) Bulletproofs protocol, when instantiated over certain choices of rings and with the challenge set instantiated with a set of all bounded-norm elements, has negligible knowledge error. The overall argument also serves as an example of how to utilise the notion of adaptive special soundness to prove that a protocol is knowledge sound.

G.1 Overview

The lattice-based Bulletproofs protocol [BLNS20; AL21; ACK21; BF23] is the lattice-analogue of the Bulletproofs protocol [BCCGP16; BBBP+18] in the discrete logarithm setting. It allows a prover to convince a verifier, with poly-logarithmic communication, about its knowledge of a vector $\mathbf{x}$ satisfying the Inhomogeneous Short Integer Solution (ISIS) instance $(\mathbf{A}, \mathbf{y})$, i.e.

$$\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q \quad \text{and} \quad \|\mathbf{x}\| \leq \alpha$$

where $\mathbf{A}, \mathbf{x}, \mathbf{y}$ are a matrix and vectors over some (typically cyclotomic) ring $\mathcal{O}$.

In a nutshell, the protocol proceeds in $2\mu+1$ moves (or $\mu+1$ rounds), where $m = 2^\mu m_\mu$ is the number of columns (resp. rows) of $\mathbf{A}$ (resp. $\mathbf{x}$) and $m_\mu \geq 1$ is the base dimension at which the folding stops, as follows:

1. At each round $i \in [\mu]$:
   (a) The prover computes two univariate degree-1 polynomials in variable C,
   $$f_{\mathbf{A}}(C) \coloneqq \mathbf{A}_0 \cdot C + \mathbf{A}_1 \bmod q \qquad \text{and} \qquad g_{\mathbf{x}}(C) \coloneqq \mathbf{x}_0 + \mathbf{x}_1 \cdot C$$
   where $(\mathbf{A}_0, \mathbf{A}_1)$ and $(\mathbf{x}_0, \mathbf{x}_1)$ are bisections of $\mathbf{A}$ and $\mathbf{x}$ respectively satisfying $\mathbf{A}\mathbf{x} = \mathbf{A}_0\mathbf{x}_0 + \mathbf{A}_1\mathbf{x}_1 = \mathbf{y} \bmod q$. It sends the product polynomial $h(C) \coloneqq f_{\mathbf{A}}(C) \cdot g_{\mathbf{x}}(C) \bmod q$ to the verifier.
   (b) The verifier checks that $h(C)$ is indeed a quadratic polynomial, and that the coefficient of C is $\mathbf{y}$. If so, it sends a random challenge $c \leftarrow_\$ \mathfrak{C} \subseteq \mathcal{O}$ to the prover, where the challenge set $\mathfrak{C}$ is a parameter of the protocol.
   (c) Prover and verifier update the statement to $(\mathbf{A}', \mathbf{y}') \coloneqq (f_{\mathbf{A}}(c), h(c))$. The prover also updates its witness to $\mathbf{x}' \coloneqq g_{\mathbf{x}}(c)$.
   (d) Continue to the next round with new statement $(\mathbf{A}', \mathbf{y}')$ and witness $\mathbf{x}'$.
2. After μ rounds, the witness $\mathbf{x}$ has dimension m_μ, and the matrix $\mathbf{A}$ has m_μ columns. The prover simply sends $\mathbf{x}$.
3. The verifier checks that $\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q$, and that $\|\mathbf{x}\|$ does not exceed a certain norm bound given as a parameter of the protocol.

Existing Results. Earlier works [BLNS20; AL21; ACK21] analysed the tree-special soundness of the lattice-based Bulletproofs protocol when instantiated with challenge sets $\mathfrak{C}$ which admit a certain algebraic structure, i.e. that the Vandermonde determinant defined by any 3 distinct challenges divides some small slack element s. In such settings, the knowledge extractor is able to extract a vector $\mathbf{x}^\dagger$ satisfying

$$\mathbf{A}\mathbf{x}^\dagger = \mathbf{y}s \bmod q \quad \text{and} \quad \|\mathbf{x}\| \leq \alpha'$$

for some stretched $\alpha' > \alpha$. Note that the relation $\mathbf{A}\mathbf{x}^\dagger = \mathbf{y}s \bmod q$ is satisfied only with a multiplicative error s.[26] The soundness errors achieved by such instantiations and analyses are unfortunately lower bounded by $\Omega(1/\varphi)$, where φ is the degree of the cyclotomic ring, due to upper bounds on the cardinality of sets satisfying the required algebraic structures (e.g. see [AL21]), at least for cyclotomic rings with prime-power conductors.

[26] A slack of $s = 1$ is achievable over cyclotomic rings with both prime-power [AL21] and non-prime-power [KLNO24] conductors.

Bünz and Fisch [BF23] circumvented the inverse-polynomial lower bound by choosing an exponential-size unstructured challenge set $\mathfrak{C}$ consisting of short elements. To reason about the soundness of such an instantiation, they proposed the notion of *almost special soundness* (AMSS), and showed that the instantiation is AMSS and consequently also knowledge sound with negligible soundness error (without repetition). There are, however, important questions still left open:

1. First, AMSS is a rather complex notion that is intimately connected to a deterministic commitment scheme in its very definition. It is therefore not generally applicable for analysing other protocols. Also, the knowledge error achieved by the AMSS knowledge extractor given in [BF23] has a complex expression and is suboptimal.
2. Second, the AMSS analysis of the lattice-based Bulletproofs protocol was only done for the instantiation over $\mathcal{O} = \mathbb{Z}$, while essentially all practical lattice-based schemes are instantiated over some (typically cyclotomic) number rings $\mathcal{O}$. Generalising the analysis to the ring setting is highly non-trivial due to the distinction between the algebraic and geometric norm over a number field $\mathcal{K}$. Over $\mathbb{Q}$, these norms coincide.

Our Goal. We follow Bünz and Fisch [BF23] to instantiate the challenge set by an exponential-size unstructured set of short elements. Different from their approach, however, is that we will be analysing the instantiated lattice Bulletproofs protocol via the adaptive special soundness framework. This serves as an example of how the framework can be used to analyse the soundness of protocols with complicated witness extraction strategies. Putting the framework aside, our analysis also covers more general choices of $\mathcal{O}$. Concretely, we cover $\mathcal{O} = \mathbb{Z}$ as a special case, as well as the rings of integers of imaginary quadratic fields with class number 1.

To establish some context, we will be working over a number field $\mathcal{K}$ of degree φ, its ring of integers $\mathcal{O}$, and the quotient ring $\mathcal{O}_q \coloneqq \mathcal{O}/q\mathcal{O}$. Statements $(\mathbf{A}, \mathbf{y})$ are defined over $\mathcal{O}_q$, and witnesses $\mathbf{x}$ are defined over $\mathcal{O}$. Our choice of (geometric) norm of an element $x \in \mathcal{K}$ is the ℓ_∞-norm over the canonical embedding of x, denoted simply by $\|x\|$. Our goal is to analyse the soundness of the lattice Bulletproofs protocol instantiated over $\mathcal{O}$ and with the challenge set $\mathfrak{C}$ being the set of all $\mathcal{O}$ elements of norm at most some β, i.e. $\mathfrak{C} \coloneqq \{x \in \mathcal{O} : \|x\| \leq \beta\}$. With this instantiation, it is straightforward to see that the protocol is complete for the ISIS relation defined above if the final norm bound that the verifier checks is set to $\alpha\beta^\mu$.

In the following, we will overview the analysis of the protocol via the adaptive special soundness framework. The roadmap is as follows. We first establish some convenient notation for parsing prover and verifier messages from a tree of transcripts. Then, we describe the high-level structure of a UCS $\mathcal{U}$ and briefly discuss why it has negligible knowledge soundness. Finally, we discuss why the protocol is $\mathcal{U}$-adaptive special sound.

Tree Representation of Transcripts. To prepare for the following discussion, it will be convenient to establish some notation concerning trees of transcripts.

Let $\mathfrak{T}$ be a (rooted) tree. We count the depth of $\mathfrak{T}$ by its edge-depth, i.e. the number of edges in the longest root-to-leaf path. For a node $v \in \mathfrak{T}$, the depth of v in $\mathfrak{T}$, denoted $\mathsf{depth}_{\mathfrak{T}}(v)$, is defined as the number of edges in the path from the root to v. Consequently, the root has depth 0. For a depth-μ tree $\mathfrak{T}$, and a node $v \in \mathfrak{T}$ of depth $i \in \{0, \ldots, \mu\}$, we write $\mathfrak{T}_v$ for the depth-$(\mu - i)$ subtree rooted at v.

Observe that each execution of the protocol produces a transcript τ consisting of a tuple of $2\mu + 1$ elements $(h_1, c_1, h_2, c_2, \ldots, h_\mu, c_\mu, \mathbf{x})$, where h_i are quadratic univariate polynomials sent by the prover, $c_i \in \mathfrak{C}$ are challenges sent by the verifier, and $\mathbf{x} \in \mathcal{O}^{m_\mu}$ is the final prover message. Transcripts with agreeing prefixes can be arranged into a tree of transcripts $\mathfrak{T}$ of (edge-)depth μ, where nodes are labelled with prover messages, and edges are labelled with verifier challenges. Each individual transcript can be parsed from the tree by collecting the labels from the corresponding root-to-leaf path.

We index prover and verifier messages by the identifiers of nodes and edges respectively, i.e. the prover message at a non-leaf node v is denoted h_v, the challenge at an edge connecting a parent v with a child w is denoted $c_{v,w}$, and the prover message at a leaf node v is denoted $\mathbf{x}_v$. The subtree of $\mathfrak{T}$ rooted at v is denoted $\mathfrak{T}_v$, and the partial transcript obtained by parsing the labels from the root to the node v is denoted τ_v.

For any node v in a tree of transcripts $\mathfrak{T}$ for some statement $\mathsf{stmt} = (\mathbf{A}, \mathbf{y})$, we can assign a statement $\mathsf{stmt}_v = (\mathbf{A}_v, \mathbf{y}_v)$ to the node v as follows:

- The root node is assigned $\mathsf{stmt} = (\mathbf{A}, \mathbf{y})$.

– For any non-root node w with parent v, we can assign $\mathbf{A}_w \coloneqq f_{\mathbf{A}_v}(c_{v,w})$ and $\mathbf{y}_w \coloneqq h_v(c_{v,w})$, and set $\mathsf{stmt}_w \coloneqq (\mathbf{A}_w, \mathbf{y}_w)$.

Finally, we extend the notation of the folding polynomial $f_{\mathbf{A}}$ and $g_{\mathbf{x}}$ so that they also take as input a sequence of challenges $\mathbf{c}$ instead of single challenges. On input $\mathbf{c} = (c_{i+1}, \ldots, c_\mu)$, $g_{\mathbf{x}}(\mathbf{c})$ is defined by recursively folding $\mathbf{x}$ by c_{i+1}, followed by c_{i+2}, all the way until c_μ. The value of $f_{\mathbf{A}}(\mathbf{c})$ is defined analogously.

Constructing the UCS. We overview our construction of a (family of) UCS(s) $\mathcal{U} = (\mathcal{U}_i)_{i=0}^{\mu-1}$ designed for the above instantiation of the lattice Bulletproofs protocol. The goal of $\mathcal{U}$ is to guide an extractor to produce, when given oracle access to a prover, a tree of transcripts $\mathfrak{T}$ for a statement $\mathsf{stmt} = (\mathbf{A}, \mathbf{y})$. Given such a tree of transcripts, a witness extractor can produce a witness $(\mathbf{x}, s)$ with the guarantee that one of the following is satisfied:

$$\begin{array}{lllll} (1) & \mathbf{A}\mathbf{x} = \mathbf{y}s \bmod q, & \|\mathbf{x}\| \leq \alpha_0, & \|s\| \leq \delta_0, \\ (2) & \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q, & \|\mathbf{x}\| \leq \alpha_0', & \end{array}$$

for some $\alpha_0, \alpha_0', \delta_0$. In fact, the witness extractor that we are about to construct will consist of a family of extractors $(\mathcal{W}_i)_{i=0}^{\mu}$, where $\mathcal{W}_i$ is responsible for extracting a witness $(\mathbf{x}_v, s_v)$ for the statement $(\mathbf{A}_v, \mathbf{y}_v)$ for any depth-i node v by repeatedly calling $\mathcal{W}_{i+1}$. That is, it is guaranteed that one of the following is satisfied:

$$\begin{array}{lllll} (1) & \mathbf{A}_v\mathbf{x}_v = \mathbf{y}_v s_v \bmod q, & \|\mathbf{x}_v\| \leq \alpha_i, & \|s_v\| \leq \delta_i, \\ (2) & \mathbf{A}_v\mathbf{x}_v = \mathbf{0} \bmod q, & \|\mathbf{x}_v\| \leq \alpha_i', & \end{array}$$

for some $\alpha_i, \alpha_i', \delta_i$. The witness to stmt is then simply the output of $\mathcal{W}_0$. While the formal construction of $\mathcal{U}$ depends on that of $(\mathcal{W}_i)_{i=0}^{\mu}$, the inner-workings of the latter are not necessary for understanding the structure of $\mathcal{U}$.

Recall that a UCS $\mathcal{U}$ expects as input a sequence of transcripts. To succinctly represent transcript sequences which satisfy a tree structure, we introduce the notation $\hat{\mathfrak{T}}_v \coloneqq (\tau_v, \mathfrak{T}_v)$. It is convenient to view $\hat{\mathfrak{T}}_v$ as a subtree of some bigger tree $\mathfrak{T}$, where τ_v is the partial transcript parsed from the labels along the path from the root node to the node v, and $\mathfrak{T}_v$ is the subtree rooted at node v. The labels along each root-to-leaf path forms a transcript.

With the above notation, the behaviour of $\mathcal{U}_i(\hat{\mathfrak{T}}_v)$ is defined as follows:

1. If the candidate witness $(\mathbf{x}_w, s_w)$ extracted at any child w of v is a SIS solution, i.e. $\mathbf{A}_w\mathbf{x}_w = \mathbf{0} \bmod q$, then no further transcripts are needed, i.e. $\mathcal{U}_i(\hat{\mathfrak{T}}_v)$ returns the empty set. Otherwise, continue as described below.
2. If the node v has fewer than 3 children, then every challenge sequence $\mathbf{c} = (c_{i+1}, \ldots, c_\mu)$ is useful except for those with $c_{i+1} \in \{c_{v,w} : w \in \mathsf{children}(v)\}$. We call these the "uncharted" challenge sequences.
3. If the node v has more than 3 children, then no further transcripts are needed.
4. If v has exactly 3 children, then let $(\mathbf{x}_v^*, s_v^*)$ be the candidate witness extracted from those of the first 3 children, i.e. the candidate witness that one would extract in the traditional tree-special soundness setting. There are 3 sub-cases:
   (a) If the candidate witness $(\mathbf{x}_v^*, s_v^*)$ is inconsistent with the prover message h_v, the extractor continues to look for an uncharted challenge sequence.
   (b) If the slack s_v^* of the candidate witness is too large, then the extractor aims for an uncharted challenge sequence with a special property: If we use $\mathbf{x}_v^*/s_v^*$ (which may not be integral) as the witness to honestly respond to such a challenge sequence, the final prover message should be non-integral. Concisely, the property states that $g_{\mathbf{x}_v^*/s_v^*}(\mathbf{c}) \notin \mathcal{O}$.
   (c) If none of the above cases is triggered, no further transcripts are needed.

Note that Steps 1, 4(a) and 4(b) are where probabilistic tests take place, and they depend on not only challenges but also entire transcripts. These steps are therefore not captured by the traditional tree-special soundness setting.

To see why $\mathcal{U}$ has a negligible knowledge error, we first observe that the cardinality of $|\mathfrak{C}|$ is exponential in the degree φ of the ring. We also see that when $\mathcal{U}$ restricts the useful challenges to an uncharted set, only a negligible portion of $\mathfrak{C}$ is ruled out. Finally, the largest non-empty set output by $\mathcal{U}$ is the intersection between an uncharted set and a set of challenges $\mathbf{c}$ such that $g_{\mathbf{x}_v^*/s_v^*}(\mathbf{c}) \notin \mathcal{O}$, or equivalently

$g_{\mathbf{x}_v^*}(\mathbf{c}) \neq 0 \bmod s_v^*$, where $g_{\mathbf{x}_v^*}$ is a multilinear polynomial over $\mathcal{K}$ with coefficients given by $\mathbf{x}_v^*$. The challenges $\mathbf{c}$ that are ruled out are either charted, which contributes a negligible portion of $\mathfrak{C}$, and those which satisfy $g_{\mathbf{x}_v^*}(\mathbf{c}) = 0 \bmod s_v^*$. By a generalisation of the so called "multilinear composite Schwartz-Zippel" lemma of [BF23] to the ring setting, we can upper bound the latter in terms of the inverse of the norm of the denominator ideal of $\mathbf{x}_v^*/s_v^*$. Intuitively, if s_v^* is large in (geometric) norm, then chances are that it is also large in field norm, and therefore the event $g_{\mathbf{x}_v^*}(\mathbf{c}) = 0 \bmod s_v^*$ happens with low probability.

Obviously, an element s with large (geometric) norm does not necessarily have large field norm. Indeed, for rings $\mathcal{O}$ which contain infinitely many units, there exist units with unbounded (geometric) but field norm ± 1. To deal with this issue, heuristically, we observe that the slack s_v^* is not fully maliciously chosen, but is rather largely influenced by the random challenges picked by the extractor. We can therefore expect it to behave like a "random" element, for which having large geometric norm implies having large field norm. Formally, we focus our attention to the rings of integers of imaginary quadratic fields, where the number of units is finite (namely, 2), and the field norm is simply the square of the geometric norm.

$\mathcal{U}$-Adaptive Special Soundness Analysis. We next explain why the lattice Bulletproofs protocol instantiated as above is $\mathcal{U}$-adaptive special sound. For this, it suffices to argue that given a $\mathcal{U}$-tree of transcripts $\mathfrak{T}$, the root witness extractor $\mathcal{W}_0$ is able to output a witness $(\mathbf{x}^*, s^*)$ which either satisfies the ISIS relation $(\mathbf{A}, \mathbf{y})$ or the SIS relation $\mathbf{A}$. We sketch the argument for this below.

First, we observe that the witness $(\mathbf{x}_\ell, 1)$ extracted at any leaf node ℓ must satisfy the ISIS relation $(\mathbf{A}_\ell, \mathbf{y}_\ell)$. We want to show by induction that, at every node v, the extracted witness $(\mathbf{x}_v, s_v)$ either satisfies the ISIS relation $(\mathbf{A}_v, \mathbf{y}_v)$ or the SIS relation $\mathbf{A}_v$.

Second, we argue that if there exists a node w of depth $i+1$ at which the extracted witness satisfies the SIS relation given by $\mathbf{A}_w$, then $\mathcal{W}_i$ executed at the parent node v can output a witness satisfying the SIS relation given by $\mathbf{A}_v$. This allows to propagate a SIS solution anywhere in the tree $\mathfrak{T}$ all the way to the root. In the remaining analysis of $(\mathbf{x}_v, s_v)$, we therefore assume that, for any node $w \neq v$ under the subtree $\mathfrak{T}_v$, $(\mathbf{x}_w, s_w)$ satisfies the ISIS relation $(\mathbf{A}_w, \mathbf{y}_w)$.

Next, we argue that if the candidate witness $(\mathbf{x}_v^*, s_v^*)$, extracted based on the first 3 children of v, is inconsistent with the prover message h_v, or is such that the slack s_v^* is too large, then $\mathcal{W}_i$ must be able to produce a witness $\mathbf{x}_v$ which satisfy the SIS relation given by $\mathbf{A}_v$.

Finally, we argue that if none of the above exceptions occur, then $(\mathbf{x}_v^*, s_v^*)$ must be a valid witness to the ISIS relation $(\mathbf{A}_v, \mathbf{y}_v)$.

In the final step above, we glossed over a detail that $\mathcal{W}_i$ should actually output $(\mathbf{x}_v^\dagger, s_v^\dagger)$, where $\mathbf{x}_v^\dagger/s_v^\dagger$ is a reduced representation of $\mathbf{x}_v^*/s_v^*$ so that $\mathbf{x}_v^\dagger$ is not much longer (in geometric norm) than $\mathbf{x}_v^*$, but $s_v^\dagger$ is much shorter (in geometric norm) than s_v^*. For rings where unique factorisation fails, i.e. the class number is larger than 1, it is unclear how this can be done efficiently. For this reason, our provable claims focus on fields with class number 1.

G.2 Algebraic Number Theory Background

Let $\mathcal{K}$ be a Galois number field and $\mathcal{O}$ its ring of integers. The literature in lattice-based cryptography primarily focuses on the field of rational numbers $\mathbb{Q}$ or cyclotomic fields $\mathcal{K} = \mathbb{Q}(\zeta)$, where $\zeta \in \mathbb{C}$ denote any fixed $\mathfrak{f}$-th primitive root of unity. In this case, we call $\mathcal{O}$ a cyclotomic ring. For our purpose, we also consider imaginary quadratic fields $\mathcal{K} = \mathbb{Q}(\sqrt{\Delta})$, where $\Delta < 0$ is a square-free integer. In this case, we call $\mathcal{O}$ an imaginary quadratic ring. Let φ denote the degree of $\mathcal{K}/\mathbb{Q}$. When an element $a \in \mathcal{O}$, we say that a is integral.

Canonical Embedding, Algebraic and Geometric Norms. We write $\sigma = (\sigma_i)_{\sigma_i \in \mathrm{Gal}(\mathcal{K}/\mathbb{Q})}$ for the tuple of all automorphisms fixing $\mathbb{Q}$. For $a \in \mathcal{K}$, the vector $\sigma(a) \in \mathbb{C}^\varphi$ is called the canonical embedding of a. The (algebraic or field) norm of an element $a \in \mathcal{K}$ is defined as $\mathcal{N}(a) \coloneqq \prod_{\sigma_i \in \mathrm{Gal}(\mathcal{K}/\mathbb{Q})} \sigma_i(a)$. The (geometric) norm[27] of a is taken as the ℓ_∞-norm of the canonical embedding of a, i.e. $\|a\| \coloneqq \|\sigma(a)\|_\infty$. The definition extends naturally to vectors. For $\mathbf{a} \in \mathcal{K}^m$, we write $\|\mathbf{a}\| \coloneqq \|(\sigma(a_1), \ldots, \sigma(a_m))\|$.

[27] We suggest calling this the "house" of a to avoid saying geometric norm all the time.

Ideals. For elements $a_1, \ldots, a_n \in \mathcal{K}$, we write $\langle a_1, \ldots, a_n \rangle$ for the (fractional) ideal generated by them. For $a, b \in \mathcal{O}$, we define the greatest common divisor (GCD) of a and b to be the GCD of the ideals generated by them, denoted by $\gcd(a, b) \coloneqq \gcd(\langle a \rangle, \langle b \rangle) = \langle a, b \rangle$. We say that a and b are co-prime if the ideals generated by them are co-prime, i.e. $\langle a, b \rangle = \langle 1 \rangle = \mathcal{O}$. We say that a vector $\mathbf{z} \in \mathcal{O}^m$ is co-prime with an element $d \in \mathcal{O}$ if $\gcd(z_1, \ldots, z_m, d) = \mathcal{O}$. By a slight abuse of notation, we also write $\gcd(\mathbf{z}, d) = \mathcal{O}$.

The (algebraic or field) norm of a ideal $\mathcal{I} \subseteq \mathcal{O}$ is defined as $\mathcal{N}(\mathcal{I}) \coloneqq |\mathcal{O}/\mathcal{I}|$. The definition extends naturally to fractional ideals due to the multiplicativity of $\mathcal{N}(\cdot)$. It is known that $|\mathcal{N}(a)| = \mathcal{N}(\langle a \rangle)$.

G.3 Protocol Description

Shorthand notation:

- For $0 \le i \le \mu$, $\mathbf{C}_i = (C_1, \ldots, C_i)$ is a sequence of formal variables.
- For a matrix (of polynomials) $\mathbf{M}$, $\mathbf{M}[i : j]$ denotes column i to column j of $\mathbf{M}$.

Folding: The same operation folds the matrix $\mathbf{A}$ and the witness $\mathbf{x}$, and is applied both to them and to their folded versions at internal nodes. For an operand $\mathbf{M} \in \mathcal{O}_q^{n \times d}$ (resp. $\mathbf{v} \in \mathcal{K}^d$) and a tuple $\mathbf{C}_t = (C_1, \ldots, C_t)$ of formal variables with $2^t \mid d$, define $f_{\mathbf{M}}(\mathbf{C}_t) \coloneqq f_{\mathbf{M},t}(\mathbf{C}_t)$ and $g_{\mathbf{v}}(\mathbf{C}_t) \coloneqq g_{\mathbf{v},t}(\mathbf{C}_t)$ by the recursions, for $1 \le j \le t$,

$$
\begin{aligned}
f_{\mathbf{M},0} &\coloneqq \mathbf{M} \in \mathcal{O}_q^{n \times d}, \\
f_{\mathbf{M},j}(\mathbf{C}_j) &\coloneqq f_{\mathbf{M},j-1}[1 : d/2^j] \cdot C_j + f_{\mathbf{M},j-1}[d/2^j + 1 : d/2^{j-1}] \in \mathcal{O}_q^{n \times d/2^j}[\mathbf{C}_j], \\
g_{\mathbf{v},0} &\coloneqq \mathbf{v} \in \mathcal{K}^d, \\
g_{\mathbf{v},j}(\mathbf{C}_j) &\coloneqq g_{\mathbf{v},j-1}[1 : d/2^j] + g_{\mathbf{v},j-1}[d/2^j + 1 : d/2^{j-1}] \cdot C_j \in \mathcal{K}^{d/2^j}[\mathbf{C}_j],
\end{aligned}
$$

so that $f_{\mathbf{M}}(\mathbf{C}_t) \in \mathcal{O}_q^{n \times d/2^t}[\mathbf{C}_t]$ and $g_{\mathbf{v}}(\mathbf{C}_t) \in \mathcal{K}^{d/2^t}[\mathbf{C}_t]$. The slice indices are taken relative to the column count d of the operand (resp. the length of $\mathbf{v}$), and since the index j is determined by the input as $j = |\mathbf{C}_j|$ and d by the operand, we drop the subscript and write $f_{\mathbf{M}}$ and $g_{\mathbf{v}}$. On the statement $\mathbf{A} \in \mathcal{O}_q^{n \times m}$ and witness $\mathbf{x} \in \mathcal{K}^m$ with $t = \mu$, this gives the full fold $f_{\mathbf{A}}(\mathbf{C}_\mu) \in \mathcal{O}_q^{n \times m_\mu}[\mathbf{C}_\mu]$ and $g_{\mathbf{x}}(\mathbf{C}_\mu) \in \mathcal{K}^{m_\mu}[\mathbf{C}_\mu]$. On a depth-$i$ folded statement $\mathbf{A}_v$ or coprime witness $\mathbf{x}_v^\dagger / s_v^\dagger$ of dimension $m_i = 2^{\mu - i} m_\mu$ with $t = \mu - i$, it folds to dimension m_μ, and a single fold $t = 1$ halves the dimension.

Fig. 9. Folding subroutines.

Parameters: $\mathcal{O}, n, m = 2^\mu m_\mu, q, \alpha, \beta$ are lattice parameters which defines a challenge set $\mathfrak{C} \coloneqq \{c \in \mathcal{O} : \|c\| \le \beta - 1\} \subseteq \mathcal{O}$.

$\langle P((\mathbf{A}, \mathbf{y}), \mathbf{x}), V((\mathbf{A}, \mathbf{y})) \rangle$, where $(\mathbf{A}, \mathbf{y}) \in \mathcal{O}_q^{n \times m} \times \mathcal{O}_q^n$ and $\mathbf{x} \in \mathcal{O}^m$:

1. P, V: Set $\mathbf{A}_0 \coloneqq \mathbf{A}$ and $\mathbf{y}_0 \coloneqq \mathbf{y}$.
2. For $i \in [\mu]$:
   (a) P: Send the polynomial $h_i(C) \coloneqq f_{\mathbf{A}_{i-1}}(C) \cdot g_{\mathbf{x}_{i-1}}(C) \in \mathcal{O}_q^n[C]$ to V.
   (b) V: Assert that $h_i \in \mathcal{O}_q^n[C]$ is a quadratic polynomial.
   (c) V: Assert that the coefficient of C in h_i is $\mathbf{y}_{i-1}$.
   (d) V: Send $c_i \leftarrow_\$ \mathfrak{C}$ to P.
   (e) P, V: Compute $\mathbf{A}_i \coloneqq f_{\mathbf{A}_{i-1}}(c_i)$ and $\mathbf{y}_i \coloneqq h_i(c_i)$.
   (f) P: Compute $\mathbf{x}_i \coloneqq g_{\mathbf{x}_{i-1}}(c_i)$.
3. P: Send $\mathbf{x}_\mu$ to V.
4. V: Assert that $((\mathbf{A}_\mu, \mathbf{y}_\mu), \mathbf{x}_\mu) \in \mathcal{R}^{\mathsf{ISIS}}_{\mathcal{O}, n, m_\mu, q, \alpha\beta\mu}$

Fig. 10. The lattice Bulletproofs protocol.

In Fig. 10 we recall the lattice analogue [BLNS20; AL21; ACK21] of the Bulletproofs protocol [BCGG+17; BBBP+18] for proving ISIS relations

$$\mathcal{R}^{\mathsf{ISIS}}_{\mathcal{O},n,m,q,\alpha} \coloneqq \left\{ ((\mathbf{A},\mathbf{y}),\mathbf{x}) \in (\mathcal{O}_q^{n\times m} \times \mathcal{O}_q^n) \times \mathcal{O}^m \;\middle|\; \begin{array}{l} \mathbf{A}\mathbf{x} = \mathbf{y} \bmod q \\ \wedge\ \|\mathbf{x}\| \leq \alpha \end{array} \right\},$$

described in a way that will be convenient for the subsequent soundness analysis. The protocol uses the "folding" subroutines defined in Fig. 9 and is summarised as follows. On input an ISIS statement $(\mathbf{A},\mathbf{y}) \in \mathcal{O}_q^{n\times m} \times \mathcal{O}_q^n$ and a witness $\mathbf{x} \in \mathcal{O}^m$, where $m = 2^\mu m_\mu$ for a base dimension $m_\mu \geq 1$ and, for simplicity, we assume that $m/m_\mu = 2^\mu$ is a power of 2, the prover and the verifier recursively "fold" $(\mathbf{A},\mathbf{x},\mathbf{y})$ by iterating between 1) defining multilinear polynomials $(f_{\mathbf{A},i}, g_{\mathbf{x},i}, h_{\mathbf{y},i})$ with coefficients given by (folded version of) $(\mathbf{A},\mathbf{x},\mathbf{y})$, and 2) evaluating them at a random point c, i.e. a challenge, specified by the verifier.

Theorem G.1 (Completeness). *The lattice Bulletproofs protocol with parameters $(\mathcal{O}, n, m = 2^\mu m_\mu, q, \alpha, \beta)$ described in Fig. 10 is complete for the relation $\mathcal{R}^{\mathsf{ISIS}}_{\mathcal{O},n,m,q,\alpha}$.*

The proof of completeness is trivial and thus omitted.

G.4 UCS and Witness Extractor for Lattice Bulletproofs

Goal. We would like to construct a UCS $\mathcal{U}$ and a witness extractor $\mathcal{W}$ and show that the lattice Bulletproofs protocol is $(\mathcal{U},\mathcal{W})$-adaptive special sound for the relation

$$\mathcal{R}^{\sim\mathsf{ISIS}}_{\mathcal{O},n,m,q,\alpha,\delta} \cup \mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m,q,\alpha}$$

defined below:

$$\mathcal{R}^{\sim\mathsf{ISIS}}_{\mathcal{O},n,m,q,\alpha,\delta} \coloneqq \left\{ ((\mathbf{A},\mathbf{y}),(\mathbf{x},s)) \;\middle|\; \begin{array}{l} \mathbf{A}\mathbf{x} = \mathbf{y}s \bmod q \\ \wedge\ \langle x_1,\ldots,x_m,s\rangle = \mathcal{O} \\ \wedge\ \|\mathbf{x}\| \leq \alpha \\ \wedge\ \|s\| \leq \delta \end{array} \right\},$$

$$\mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m,q,\alpha} \coloneqq \left\{ ((\mathbf{A},\mathbf{y}),(\mathbf{x},s)) \;\middle|\; \begin{array}{l} \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q \\ \wedge\ \|\mathbf{x}\| \leq \alpha \end{array} \right\}$$

where $((\mathbf{A},\mathbf{y}),(\mathbf{x},s)) \in (\mathcal{O}_q^{n\times m} \times \mathcal{O}_q^n) \times (\mathcal{O}^m \times \mathcal{O})$. Note that $\mathbf{y}$ and s are not used in the second relation, i.e. $\mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m,q,\alpha}$. We include them so that the statement and witness spaces of the two relations are compatible.

Intuitively, being knowledge sound for the relation $\mathcal{R}$ means that if a prover succeeds in convincing the verifier in the Bulletproofs protocol often enough, then a knowledge extractor can extract one of the following:

- (Knowledge of ISIS.) A short solution $\mathbf{x} \in \mathcal{O}^m$, and a small slack $s \in \mathcal{O}$ coprime with $\mathbf{x}$, such that $\mathbf{A}\mathbf{x} = \mathbf{y}s \bmod q$.
- (Knowledge of SIS.) A short solution $\mathbf{x} \in \mathcal{O}^m$ such that $\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$.

Remark G.2. The requirement that $\langle x_1,\ldots,x_m,s\rangle = \mathcal{O}$ in $\mathcal{R}^{\sim\mathsf{ISIS}}_{\mathcal{O},n,m,q,\alpha,\delta}$ can be lifted by proving a more general multilinear composite Schwartz-Zippel lemma Theorem H.1 which drops the $\mathsf{cont}(f) + \langle d\rangle = \langle 1\rangle$ requirement.

In the following, we discuss two algebraic (non-cryptographic) sources of difficulties which force us to relax the soundness claim for certain choices of the ring $\mathcal{O}$, which in particular include cyclotomic rings.

Issue 1: Infinitude of units. As we will show shortly, we can indeed prove that the lattice Bulletproofs protocol is adaptive special sound for rings $\mathcal{O}$ which contain only finitely many units, which are precisely the rational integers $\mathbb{Z}$ and the imaginary quadratic rings due to Dirichlet's unit theorem. To handle rings which contain infinitely many units, e.g. cyclotomic rings, we unfortunately need to introduce a "failure relation" which captures cases where witness extraction fails

$$\mathcal{R}^{\perp}_{\mathcal{O},\delta,\nu} \coloneqq \left\{ ((\mathbf{A},\mathbf{y}),(\mathbf{x},s)) \;\middle|\; \mathcal{N}(s) \leq \nu \ \wedge\ \delta < \|s\| \leq \delta^4 \right\},$$

and we require the extractor $\mathcal{W}$ to output a witness for the relation

$$\mathcal{R} \coloneqq \mathcal{R}^{\sim\mathsf{ISIS}}_{\mathcal{O},n,m,q,\alpha,\delta} \cup \mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m,q,\alpha} \cup \mathcal{R}^{\perp}_{\mathcal{O},\delta,\nu}.$$

In the case where $\mathcal{O}$ has finitely many units, i.e. $\mathcal{O} = \mathbb{Z}$ or $\mathcal{O}$ is an imaginary quadratic ring, there exist choices of (δ, ν) so that $\mathcal{R}^{\perp}_{\mathcal{O},\delta,\nu} = \emptyset$, i.e. extraction cannot fail.

Lemma G.3. *For $\mathcal{O} = \mathbb{Z}$ and $\delta \geq \nu$, $\mathcal{R}^{\perp}_{\mathcal{O},\delta,\nu} = \emptyset$. For $\mathcal{O}$ being the ring of integers of $\mathbb{Q}(\sqrt{\Delta})$, where $\Delta < 0$ is a square-free integer, and $\delta^2 \geq \nu$, $\mathcal{R}^{\perp}_{\mathcal{O},\delta,\nu} = \emptyset$.*

Proof. The case $\mathcal{O} = \mathbb{Z}$ is trivial. For the case where $\mathcal{O}$ is the ring of integers of $\mathbb{Q}(\sqrt{\Delta})$, the claim follows by observing that $\|s\| = |s| = \sqrt{\bar{s}s}$ and $\mathcal{N}(s) = \bar{s}s = \|s\|^2$. □

In other cases, e.g. $\mathcal{O}$ is a cyclotomic ring, the inclusion of $\mathcal{R}^{\perp}_{\mathcal{O},\delta,\nu}$ might make soundness for the relation $\mathcal{R}$ formally trivial, since a witness extractor $\mathcal{W}$ might always output a witness of the failure relation $\mathcal{R}^{\perp}_{\mathcal{O},\delta,\nu}$. At present, we are not aware of a way to formally upper bound the failure probability, since the adversary can maliciously influence the distribution of witnesses extracted by $\mathcal{W}$, making the latter difficult to model. Instead, we provide heuristic evidence for why we believe that the failure probability should be negligible for the case of cyclotomic rings.

Issue 2: Lack of Euclidean algorithm. Another source of difficulty is the lack of efficient Euclidean algorithm over the ring $\mathcal{O}$ in general. In more detail, during witness extraction, the extractor $\mathcal{W}$ needs to reduce a vector of fractions $\mathbf{x}/s$ where $\mathbf{x} \in \mathcal{O}^m$ and $s \in \mathcal{O}$ to $\mathbf{x}'/s' = \mathbf{x}/s$, such that $\gcd(\mathbf{x}', s') = \mathcal{O}$, $\|\mathbf{x}'\|$ is not much larger than $\|\mathbf{x}\|$, and $\|s'\| \leq \|s\|$. A natural way to achieve this is to find an element s' small in both algebraic and geometric norms such that $s' \cdot \frac{\langle \mathbf{x} \rangle}{\langle s \rangle} \subset \mathcal{O}$, and define $\mathbf{x}' \coloneqq s' \cdot \frac{\mathbf{x}}{s} \in \mathcal{O}^m$. Except for very special rings, e.g. $\mathbb{Z}$ or imaginary quadratic rings, it is unclear if this can be done efficiently in general.

To abstract away the above issue, we construct $\mathcal{W}$ conditioned on the existence of an oracle O which performs coprime fraction operations as described above. A precise description of the requirements of O is given in Section G.4. For the special case where $\mathcal{O} = \mathbb{Z}$ or an imaginary quadratic ring, we show that O can be efficiently instantiated.

UCS and Witness Extractor. For any fixed statement $(\mathbf{A}, \mathbf{y})$, we construct a depth-μ UCS $\mathcal{U}^{\mathsf{O}}_{\mathbf{A},\mathbf{y}} = (\mathcal{U}^{\mathsf{O}}_{\mathbf{A},\mathbf{y},i})_{0 \leq i < \mu}$ and a witness extractor $\mathcal{W}^{\mathsf{O}}_{\mathbf{A},\mathbf{y}} = \mathcal{W}^{\mathsf{O}}_{\mathbf{A},\mathbf{y},0}$ for the lattice Bulletproofs protocol, both using a sequence of sub-extractors $(\mathcal{W}^{\mathsf{O}}_{\mathbf{A},\mathbf{y},i})_{1 \leq i < \mu}$ as subroutines. For the sake of readability, we suppress the subscript $(\mathbf{A}, \mathbf{y})$ and write $\mathcal{U}^{\mathsf{O}}_i$ and $\mathcal{W}^{\mathsf{O}}_i$ instead of $\mathcal{U}^{\mathsf{O}}_{\mathbf{A},\mathbf{y},i}$ and $\mathcal{W}^{\mathsf{O}}_{\mathbf{A},\mathbf{y},i}$.

In the following, we first establish some notation recurring in this section and then proceed to define the UCS $\mathcal{U}^{\mathsf{O}}$ and witness extractor $\mathcal{W}^{\mathsf{O}}$. The analysis of the $(\mathcal{U}^{\mathsf{O}}, \mathcal{W}^{\mathsf{O}})$-adaptive special soundness and the knowledge error of $\mathcal{U}^{\mathsf{O}}$ will be given in Section G.5.

Terminologies and Notation. Fix a statement $(\mathbf{A}, \mathbf{y}) \in \mathcal{O}_q^{n \times m} \times \mathcal{O}_q^n$, where $m = m_0 = 2^\mu m_\mu$ for a base dimension $m_\mu \geq 1$. Throughout this section, we make use of the following terminologies and notation. A tree of transcripts $\mathfrak{T}$ is a tree of (edge-)depth μ, where each root-to-leaf path can be parsed as a transcript with the i-th prover message given by the i-th node label (counting from 0 from the root) and the i-th verifier message given by the i-th edge label (counting from 1 from the root).

Given a tree of transcripts $\mathfrak{T}$, we define the following:

- (Node labels.) For any non-leaf node $v \in \mathfrak{T}$, let h_v be the label of node v, i.e. the prover's message at node v. Similarly, for a leaf node $\ell \in \mathfrak{T}$, let $\mathbf{x}_\ell$ denote the node label at ℓ.
- (Edge labels.) For any non-leaf node $v \in \mathfrak{T}$ and any child $w \in \mathsf{children}(v)$, let $c_{v,w}$ denote the label of the edge (v, w), i.e. the verifier's challenge at edge (v, w). We write $\mathbf{c}_v$ for the vector formed by concatenating all edge labels along the path from the root to the node v.
- (Path.) For any node $v \in \mathfrak{T}$, let τ_v be the concatenation of labels along path from the root to the node v.
- (Subtrees.) For any node $v \in \mathfrak{T}$, write $\mathfrak{T}_v$ for the subtree of $\mathfrak{T}$ rooted at v. We write $\hat{\mathfrak{T}}_v \coloneqq (\tau_v, \mathfrak{T}_v)$ for the pair consisting of the path τ_v and the subtree $\mathfrak{T}_v$. It is a compact representation of the sequence of transcripts obtained from all root-to-leaf paths for all leaves under v.

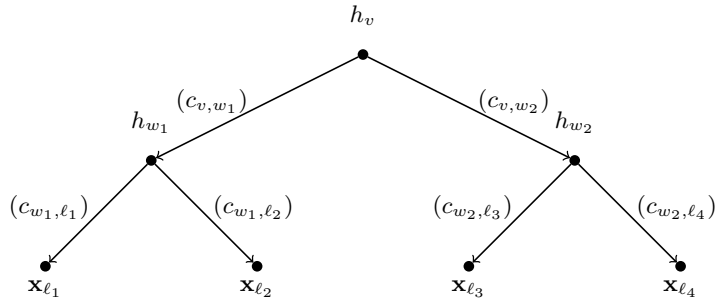


Fig. 11. Depiction of a (sub)tree of transcripts $\mathfrak{T}_v$.

– (Folded statements.) Each node $v \in \mathfrak{T}$ is associated with a "folded" statement $\mathsf{stmt}_v := (\mathbf{A}_v, \mathbf{y}_v)$ defined recursively as follows. The root node is associated with $(\mathbf{A}, \mathbf{y})$. For any non-leaf node $v \in \mathfrak{T}$ and any child $w \in \mathsf{children}(v)$,

$$\mathbf{A}_w := f_{\mathbf{A}_v}(c_{v,w}) \qquad \text{and} \qquad \mathbf{y}_w := h_v(c_{v,w}).$$

Note that stmt_v can be efficiently computed given stmt and $\hat{\mathfrak{T}}_v$.

– (Uncharted set.) For any $v \in \mathfrak{T}$, the "uncharted set" of $\mathfrak{T}_v$ is the set containing every challenge sequence where the first challenge differs from $c_{v,w}$ for any child w of v given in $\mathfrak{T}_v$. Formally, assuming that $\mathsf{depth}(\mathfrak{T}_v) = j$,

$$\mathsf{Uncharted}(\mathfrak{T}_v) := (\mathfrak{C} \setminus \{c_{v,w} : w \in \mathsf{children}(v)\}) \times \mathfrak{C}^{\mu-j-1}.$$

Parameters. Let $\gamma \geq 1$ to be a looseness parameter of an oracle O to be specified shortly. Define the following quantities:

$$\delta := 2^3\beta^3, \qquad \alpha_\mu := \alpha\beta^\mu.$$

For $0 \leq i < \mu$, define

$$\begin{aligned}
m_i &:= m/2^i \\
&= 2^{\mu-i} m_\mu, \\
\alpha_i &:= \alpha_{i+1}\gamma\delta^3 = \alpha\beta^\mu\gamma^{\mu-i}\delta^{3(\mu-i)} \\
&= (m_i/m_\mu)^9 \cdot \alpha \cdot \beta^{10\mu-9i} \cdot \gamma^{\mu-i}, \\
\alpha'_i &:= \alpha_{i+1}\delta^5 = \alpha\beta^\mu\gamma^{\mu-i-1}\delta^{3(\mu-i-1)+5} \\
&= 2^6 \cdot (m_i/m_\mu)^9 \cdot \alpha \cdot \beta^{10\mu-9i+6} \cdot \gamma^{\mu-i-1}.
\end{aligned}$$

Coprime Fraction Oracle. We assume the existence of a γ-coprime fraction oracle O which computes coprime fractions, where $\gamma \geq 1$ is some looseness parameter. More concretely, given a tuple $(\mathbf{x}, s) \in \mathcal{O}^k \times \mathcal{O}$, the tuple $(\hat{\mathbf{x}}, \hat{s}) \leftarrow \mathsf{O}(\mathbf{x}, s)$ satisfies the following:

$$\hat{\mathbf{x}} \cdot s = \mathbf{x} \cdot \hat{s}, \qquad \langle \hat{x}_1, \ldots, \hat{x}_k, \hat{s} \rangle = \mathcal{O}, \qquad \|\hat{\mathbf{x}}\| \leq \gamma \cdot \|\mathbf{x}\|, \qquad \text{and} \qquad \|\hat{s}\| \leq \|s\|.$$

In words, $\hat{\mathbf{x}}/\hat{s} = \mathbf{x}/s$ assuming that $s \neq 0$, the output $(\hat{\mathbf{x}}, \hat{s})$ is guaranteed to be coprime, $\hat{\mathbf{x}}$ is at most γ times longer than $\mathbf{x}$, and $\hat{s}$ is no longer than s.

Lemma G.4. *If $\mathcal{O} = \mathbb{Z}$ or $\mathcal{O}$ is the ring of integers of $\mathbb{Q}(\sqrt{\Delta})$ for some square-free integer $\Delta < 1$ with class number 1, then there exists efficient 1-coprime fraction oracle O.*

Proof. If $\mathcal{O} = \mathbb{Z}$ (i.e. $\mathcal{K} = \mathbb{Q}$), then there exists a trivial 1-coprime fraction oracle O which computes $d = \gcd(\mathbf{x}, s)$ and outputs $\hat{\mathbf{x}} := \mathbf{x}/d$ and $\hat{s} := s/d$. Clearly, we have $\|\hat{\mathbf{x}}\| \leq \|\mathbf{x}\|$ and $\|\hat{s}\| \leq \|s\|$.

In general, for a number field $\mathcal{K}$ with class number 1, GCDs can be computed in polynomial time due to [Wik05]. Consider the case where $\mathcal{K}$ is an imaginary quadratic field with class number 1, and $\mathcal{O}$ being its ring of integers. It is easy to check that, for any integral element $d \in \mathcal{O}$, it holds that $\delta := \min_{\sigma_i \in \mathrm{Gal}(\mathcal{K}/\mathbb{Q})}(|\sigma_i(d)|) \geq 1$. Therefore, $\|\hat{\mathbf{x}}\| \leq \|\mathbf{x}\|/\delta \leq \|\mathbf{x}\|$ and $\|\hat{s}\| \leq \|s\|/\delta \leq \|s\|$. □

Extraction Algorithms. In Fig. 12 we define, for each $0 \le i < \mu$, a UCS $\mathcal{U}_i^{\mathsf{O}}$ and a witness sub-extractor $\mathcal{W}_i^{\mathsf{O}}$. Both take as input a sequence of transcripts represented by $\hat{\mathfrak{T}}_v$ for some node v in $\mathfrak{T}$ of depth i. As the two algorithms share most of their computation, we describe them jointly by a single algorithm $(\mathcal{U}_i^{\mathsf{O}}, \mathcal{W}_i^{\mathsf{O}})$ that returns a tuple: its first element is the output of the UCS $\mathcal{U}_i^{\mathsf{O}}(\hat{\mathfrak{T}}_v)$, a set of useful challenge sequences, and its second element is the output of the witness extractor $\mathcal{W}_i^{\mathsf{O}}(\hat{\mathfrak{T}}_v)$, a candidate witness $(\mathbf{x}_v, s_v)$ for stmt_v.

The extractors make use of a sequence of subroutines $(\mathsf{Lift}_i)_{i=0}^{\mu-1}$. In a nutshell, invoking Lift_i on the 3 witnesses $(\mathbf{x}_1, s_1), (\mathbf{x}_2, s_2), (\mathbf{x}_3, s_3)$ extracted from the first 3 children of the depth-i node v lifts the depth-i witnesses to a (extended) candidate depth-$(i-1)$ witness $(\mathbf{X}^*, s^*)$ which "explains" the prover message h_v at node v in the sense of Lemma G.5 below, where $\mathbf{X}^*$ is a 3-column matrix. This candidate depth-$(i-1)$ witness is "extended" in the sense that its middle column is an (ordinary) candidate depth-$(i-1)$ witness while the first and last columns contain extra information.

Formally, we write

$$(\mathbf{X}^*, s^*) \leftarrow \mathsf{Lift}_i((c_j, \mathbf{x}_j, s_j)_{j=1}^3).$$

The subroutine Lift_i satisfies the following property. The proof follows from a direct calculation and is omitted.

Lemma G.5. *Let $\mathbf{A}_i \in \mathcal{O}_q^{n \times m_i}$ and $h_v \in \mathcal{O}_q^n[C]$ be a quadratic polynomial. If $(c_j, \mathbf{x}_j, s_j)_{j=1}^3 \in (\mathcal{O} \times \mathcal{O}^{m_{i+1}} \times \mathcal{O})^3$ satisfies*

$$f_{\mathbf{A}_i}(c_j) \cdot \mathbf{x}_j = h_v(c_j) \cdot s_j \bmod q$$

then for all $C \in \mathcal{O}$ the output $(\mathbf{X}^*, s^*) \leftarrow \mathsf{Lift}_i((c_j, \mathbf{x}_j, s_j)_{j=1}^3)$ satisfies

$$\mathbf{A}_i \mathbf{X}^* \begin{pmatrix} 1 \\ C \\ C^2 \end{pmatrix} = h_v(C) \cdot s^* \bmod q.$$

G.5 Knowledge Soundness Analysis

We prove that the lattice-based Bulletproofs protocol is $(\mathcal{U}, \mathcal{W})$-adaptive special sound for the $(\mathcal{U}, \mathcal{W})$ constructed in Section G.4. Furthermore, we show that the knowledge error of $\mathcal{U}$ is at most $3\mu/|\mathfrak{C}| + \varepsilon_{SZ}$ for appropriate choices of parameters, where ε_{SZ} is an upper-bound of some Schwartz-Zippel-like probability. Then, by invoking the results in Sections C and E, we conclude that the lattice-based Bulletproofs protocol is knowledge-sound with the same knowledge error.

Theorem G.6 (Adaptive Special Soundness). *Let $\mathcal{O}, n, m, q, \alpha_0, \alpha_0', \gamma, \delta, \nu$ be parameters as defined in Section G.4. Assuming the existence of a γ-coprime fraction oracle O, for $(\mathcal{U}^{\mathsf{O}}, \mathcal{W}^{\mathsf{O}})$ constructed in Section G.4, the lattice Bulletproofs protocol is $(\mathcal{U}^{\mathsf{O}}, \mathcal{W}^{\mathsf{O}})$-adaptive special sound for the relation*

$$\mathcal{R} \coloneqq \mathcal{R}^{\sim\mathsf{ISIS}}_{\mathcal{O},n,m,q,\alpha_0,\delta} \cup \mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m,q,\alpha_0'} \cup \mathcal{R}^{\perp}_{\mathcal{O},n,m,q,\alpha_0,\delta,\nu}.$$

Furthermore, the oracle O is efficiently instantiable and the lattice Bulletproofs protocol is $(\mathcal{U}^{\mathsf{O}}, \mathcal{W}^{\mathsf{O}})$-adaptive special sound for the relation

$$\mathcal{R} \coloneqq \mathcal{R}^{\sim\mathsf{ISIS}}_{\mathcal{O},n,m,q,\alpha_0,\delta} \cup \mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m,q,\alpha_0'}$$

in any of the following settings:

- *$\mathcal{O} = \mathbb{Z}$ and $\delta \ge \nu$, or*
- *$\mathcal{O} = \mathbb{Q}(\sqrt{\Delta})$, where $\Delta < 0$ is a square-free integer, and $\delta^2 \ge \nu$.*

Proof. Given the first part of the theorem, the second part follows immediately from Lemmas G.3 and G.4. It remains to prove the first part.

Fix $\mathsf{stmt} = (\mathbf{A}, \mathbf{y})$. It suffices to show that, on input a complete depth-μ tree $\mathfrak{T}$ of accepting transcripts with $\mathfrak{T} \in \mathsf{Chains}(\mathcal{U}^{\mathsf{O}})$, $\mathcal{W}^{\mathsf{O}}(\mathsf{stmt}, \mathfrak{T})$ outputs a witness $\mathsf{wit} \in \mathcal{R}(\mathsf{stmt})$.

$(\mathcal{U}_i^{\mathsf{O}}, \mathcal{W}_i^{\mathsf{O}})(\mathsf{stmt}, \hat{\mathfrak{T}}_v)$, $0 \le i < \mu$:

The algorithm returns a tuple; its first element is the output of the UCS $\mathcal{U}_i^{\mathsf{O}}(\mathsf{stmt}, \hat{\mathfrak{T}}_v)$ and its second element is the output of the witness extractor $\mathcal{W}_i^{\mathsf{O}}(\mathsf{stmt}, \hat{\mathfrak{T}}_v)$.

1. If $\hat{\mathfrak{T}}_v$ contains a rejecting transcript, return $(\emptyset, \bot)$.
2. If $\hat{\mathfrak{T}}_v$ is an empty tree, return $(\mathfrak{C}^{\mu-i}, \bot)$.
3. Assert that $\mathsf{depth}(\mathfrak{T}_v) = \mu - i$.
4. Compute $\mathsf{stmt}_v = (\mathbf{A}_v, \mathbf{y}_v)$ from stmt and $\hat{\mathfrak{T}}_v$.
5. For $w \in \mathsf{children}(v)$:
   (a) If $i = \mu - 1$, let $\mathbf{x}_w$ be the label of node w and $s_w := 1$.
   (b) If $i < \mu - 1$, let $(\mathbf{x}_w, s_w) := \mathcal{W}_{i+1}^{\mathsf{O}}(\mathsf{stmt}, \tau_w, \mathfrak{T}_w) \in \mathcal{O}^{m_{i+1}} \times \mathcal{O}$.
   (c) If $((\mathbf{A}_w, \mathbf{y}_w), (\mathbf{x}_w, s_w)) \in \mathcal{R}^{\bot}_{\mathcal{O},\delta,\nu}$ then return $(\emptyset, (\bot, s_w))$.
   (d) If $((\mathbf{A}_w, \mathbf{y}_w), (\mathbf{x}_w, s_w)) \in \mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m_{i+1},q,\alpha_{i+1}}$ then return $(\emptyset, (\mathbf{x}_v, 1))$ where

$$\mathbf{x}_v := \begin{pmatrix} \mathbf{x}_w c_{(v,w)} \\ \mathbf{x}_w \end{pmatrix}.$$

6. Let $k := |\mathsf{children}(v)|$ and write $(w_1, \ldots, w_k) := \mathsf{children}(v)$.
7. If $k < 3$, then return $(\mathsf{Uncharted}(\mathfrak{T}_v), \bot)$.
8. Let $(\mathbf{X}_v^*, s_v^*) := \mathsf{Lift}_i^{\mathsf{O}}((c_{v,w_j}, \mathbf{x}_{w_j}, s_{w_j})_{j=1}^3)$ and $\mathbf{x}_v^* := \mathbf{X}_v^* \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$.
9. Compute the coprime fraction $(\mathbf{x}_v^\dagger, s_v^\dagger) := \mathsf{O}(\mathbf{x}_v^*, s_v^*)$.
10. If $k = 3$:
    – Define the following predicates:

$$b_0 := (s_v^* \cdot h_v(C) \stackrel{?}{=} f_{\mathbf{A}_v}(C) \cdot g_{\mathbf{x}_v^*}(C)),$$
$$b_1 := (\mathcal{N}(s_v^\dagger) \stackrel{?}{\le} \nu).$$

    – If $b_0 = 0$, return $(\mathsf{Uncharted}(\mathfrak{T}_v), \bot)$.
    – If $b_1 = 0$, return $(\mathsf{Uncharted}(\mathfrak{T}_v) \cap \mathfrak{C}_{\mathsf{SZ},\mu-i}(\mathbf{x}_v^\dagger, s_v^\dagger), \bot)$ where

$$\mathfrak{C}_{\mathsf{SZ},\mu-i}(\mathbf{x}_v^\dagger, s_v^\dagger) := \left\{ \mathbf{c} \in \mathfrak{C}^{\mu-i} : g_{\mathbf{x}_v^\dagger / s_v^\dagger}(\mathbf{c}) \notin \mathcal{O}^{m_\mu} \right\}.$$

    – If $b_0 = 1$ and $b_1 = 1$, return $(\emptyset, (\mathbf{x}_v^\dagger, s_v^\dagger))$.
11. If $k = 4$ and there exists $w \in \mathsf{children}(v)$ such that

$$\mathbf{x}_v^{(w)} := \mathbf{X}_v^* \begin{pmatrix} 1 \\ c_{(v,w)} \\ c_{(v,w)}^2 \end{pmatrix} s_w - \begin{pmatrix} \mathbf{x}_w c_{(v,w)} \\ \mathbf{x}_w \end{pmatrix} s_w^2 \neq \mathbf{0}$$

    then return $(\emptyset, (\mathbf{x}_v^{(w)}, 1))$.
12. Return $(\emptyset, \bot)$.

$\mathsf{Lift}_i^{\mathsf{O}}((c_i, \mathbf{x}_i, s_i)_{i=1}^3 \in (\mathcal{O} \times \mathcal{O}^{m_{i+1}} \times \mathcal{O})^3)$, $0 \le i < \mu$:

1. Let $\mathbf{V} := \begin{pmatrix} 1 & 1 & 1 \\ c_1 & c_2 & c_3 \\ c_1^2 & c_2^2 & c_3^2 \end{pmatrix} \in \mathcal{O}^{3\times 3}$ and $s^* := \det(\mathbf{V}) \cdot \prod_{j=1}^3 s_j \in \mathcal{O}$.
2. For $i \in [3]$, let $\mathbf{x}_i^* := \mathbf{x}_i \cdot \prod_{j \in [3]\setminus\{i\}} s_j \in \mathcal{O}^{m_{i+1}}$.
3. Let $\mathbf{X}^* := \begin{pmatrix} \mathbf{x}_1^* c_1 & \mathbf{x}_2^* c_2 & \mathbf{x}_3^* c_3 \\ \mathbf{x}_1^* & \mathbf{x}_2^* & \mathbf{x}_3^* \end{pmatrix} \mathbf{V}^{-1} \det(\mathbf{V}) \in \mathcal{O}^{m_i \times 3}$.
4. Return $(\mathbf{X}^*, s^*)$.

Fig. 12. The UCS $\mathcal{U}_i^{\mathsf{O}}$ and witness extractor $\mathcal{W}_i^{\mathsf{O}}$ for lattice Bulletproofs.

Invariant. To show the above claim, it suffices to show that the following invariant holds for each node $v \in \mathfrak{T}$: For each leaf node $v \in \mathfrak{T}$, the prover's final message $\mathbf{x}_v$ satisfies

$$((\mathbf{A}_v, \mathbf{y}_v), (\mathbf{x}_v, 1)) \in \mathcal{R}^{\sim\mathsf{ISIS}}_{\mathcal{O},n,m_\mu,q,\alpha_\mu,\delta_\mu}.$$

For each non-lead node $v \in \mathfrak{T}$, let $0 \le i < \mu$ denote the depth of v. The extracted witness $(\mathbf{x}_v, s_v) \coloneqq \mathcal{W}^{\mathsf{O}}_i(\hat{\mathfrak{T}}_v)$ satisfies

$$((\mathbf{A}_v, \mathbf{y}_v), (\mathbf{x}_v, s_v)) \in \mathcal{R}^{\sim\mathsf{ISIS}}_{\mathcal{O},n,m_i,q,\alpha_i,\delta} \cup \mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m_i,q,\alpha'_i} \cup \mathcal{R}^{\perp}_{\mathcal{O},n,m_i,q,\alpha_i,\delta,\nu}.$$

The main claim is captured by the special case where v is the root node.

We start with the observation that the invariant trivially holds for all leaf nodes by the construction of the lattice Bulletproofs protocol and by the assumption that $\mathfrak{T}$ is accepting. Below, we argue for the invariant by induction from the bottom to the top of the tree $\mathfrak{T}$.

Fix any non-leaf node $v \in \mathfrak{T}$ of depth-i, with $0 \le i < \mu$, and assume that the invariant holds for all $w \in \mathsf{children}(v)$. One of the following must be the case:

1. There exists $w \in \mathsf{children}(v)$ such that
$$((\mathbf{A}_w, \mathbf{y}_w), (\mathbf{x}_w, s_w)) \in \mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m_{i+1},q,\alpha'_{i+1}} \cup \mathcal{R}^{\perp}_{\mathcal{O},\delta,\nu}.$$
2. For all $w \in \mathsf{children}(v)$, $((\mathbf{A}_w, \mathbf{y}_w), (\mathbf{x}_w, s_w)) \in \mathcal{R}^{\sim\mathsf{ISIS}}_{\mathcal{O},n,m_{i+1},q,\alpha_{i+1},\delta}$.

Since $\mathfrak{T}$ is complete, by the definition of $\mathcal{W}^{\mathsf{O}}_i$, we must have $|\mathsf{children}(v)| \le 4$.

We analyse the behaviour of $\mathcal{W}^{\mathsf{O}}_i(\hat{\mathfrak{T}}_v)$ in the above cases. For clarity, we will use the symbol $\lozenge$ to signify the end of ease case.

Case 1. In this case, there exists $w \in \mathsf{children}(v)$ such that

$$((\mathbf{A}_w, \mathbf{y}_w), (\mathbf{x}_w, s_w)) \in \mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m_{i+1},q,\alpha'_{i+1}} \cup \mathcal{R}^{\perp}_{\mathcal{O},\delta,\nu}.$$

If $((\mathbf{A}_w, \mathbf{y}_w), (\mathbf{x}_w, s_w)) \in \cup\mathcal{R}^{\perp}_{\mathcal{O},\delta,\nu}$ then $\mathcal{W}^{\mathsf{O}}_i(\hat{\mathfrak{T}}_v)$ would output $(\perp, s_w)$. Clearly we have $((\mathbf{A}_v, \mathbf{y}_v), (\perp, s_w)) \in \cup\mathcal{R}^{\perp}_{\mathcal{O},\delta,\nu}$.

If $((\mathbf{A}_w, \mathbf{y}_w), (\mathbf{x}_w, s_w)) \in \mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m_{i+1},q,\alpha'_{i+1}}$, then $\mathcal{W}^{\mathsf{O}}_i(\hat{\mathfrak{T}}_v)$ would output $(\mathbf{x}_v, 1)$ where $\mathbf{x}_v = \begin{pmatrix} \mathbf{x}_w c_{(v,w)} \\ \mathbf{x}_w \end{pmatrix}$. Note that

$$\mathbf{A}_v \mathbf{x}_v = \mathbf{A}_v \begin{pmatrix} \mathbf{x}_w c_{(v,w)} \\ \mathbf{x}_w \end{pmatrix} = \mathbf{A}_w \mathbf{x}_w = \mathbf{0} \bmod q, \text{ and}$$
$$\|\mathbf{x}_v\| \le \beta \alpha'_{i+1} \le \alpha'_i.$$

In other words,

$$((\mathbf{A}_v, \mathbf{y}_v), (\mathbf{x}_v, 1)) \in \mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m_i,q,\alpha'_i}$$

and thus the invariant holds at node v. $\lozenge$

Case 2. In this case, for all $w \in \mathsf{children}(v)$, we have

$$((\mathbf{A}_w, \mathbf{y}_w), (\mathbf{x}_w, s_w)) \in \mathcal{R}^{\sim\mathsf{ISIS}}_{\mathcal{O},n,m_{i+1},q,\alpha_{i+1},\delta}.$$

In fact, we have a stronger guarantee that, for all $u \in \mathfrak{T}_v$ of depth-j with $u \ne v$, it holds that

$$((\mathbf{A}_u, \mathbf{y}_u), (\mathbf{x}_u, s_u)) \in \mathcal{R}^{\sim\mathsf{ISIS}}_{\mathcal{O},n,m_j,q,\alpha_j,\delta}.$$

Indeed, observe that if any node $u \ne v$ in the subtree $\mathfrak{T}_v$ satisfies

$$((\mathbf{A}_u, \mathbf{y}_u), (\mathbf{x}_u, s_u)) \in \mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m_j,q,\alpha'_j} \cup \mathcal{R}^{\perp}_{\mathcal{O},\delta,\nu}$$

then Case 1 would have been triggered at v.

Define the following predicates:

$$b_0 \coloneqq (s_v^* \cdot h_v(C) \stackrel{?}{=} f_{\mathbf{A}_v}(C) \cdot g_{\mathbf{x}_v^*}(C)),$$
$$b_1 \coloneqq (\mathcal{N}(s_v^\dagger) \stackrel{?}{\leq} \nu),$$
$$b_2 \coloneqq \left(\exists w \in \mathsf{children}(v) : \ \mathbf{X}_v^* \begin{pmatrix} 1 \\ c_{(v,w)} \\ c_{(v,w)}^2 \end{pmatrix} s_w - \begin{pmatrix} \mathbf{x}_w c_{(v,w)} \\ \mathbf{x}_w \end{pmatrix} s_w^2 \stackrel{?}{\neq} \mathbf{0} \right).$$

We further consider 4 sub-cases, one of which must occur by the construction of $\mathcal{W}_i$ and by the completeness of $\mathfrak{T}$:

a) $|\mathsf{children}(v)| = 3$, $b_0 = 1$, and $b_1 = 1$. (If $b_0 = 0$ or $b_1 = 0$, then the construction of $\mathcal{W}_i$ forces $|\mathsf{children}(v)| \geq 4$.)
b) $|\mathsf{children}(v)| = 4$ and $b_2 = 1$.
c) $|\mathsf{children}(v)| \geq 4$, $b_0 = 0$, and $b_2 = 0$.
d) $|\mathsf{children}(v)| \geq 4$, $b_1 = 0$, and $b_2 = 0$.

Below, we show that Cases 2a and 2b both lead to the conclusion that the invariant holds at node v. We also show that Cases 2c and 2d never happen. This completes the proof. ◊

Case 2a. In this case, $|\mathsf{children}(v)| = 3$, and it holds that

$$\mathcal{N}(s_v^\dagger) \leq \nu \qquad \text{and} \qquad s_v^* \cdot h_v(C) = f_{\mathbf{A}_v}(C) \cdot g_{\mathbf{x}_v^*}(C).$$

The witness extractor $\mathcal{W}_i^{\mathsf{O}}(\hat{\mathfrak{T}}_v)$ would return $(\mathbf{x}_v, s_v) \coloneqq (\mathbf{x}_v^\dagger, s_v^\dagger)$.

Recall that

$$\mathbf{x}_v^* = \begin{pmatrix} \mathbf{x}_{w_1}^* c_{v,w_1} & \mathbf{x}_{w_2}^* c_{v,w_2} & \mathbf{x}_{w_3}^* c_{v,w_3} \\ \mathbf{x}_{w_1}^* & \mathbf{x}_{w_2}^* & \mathbf{x}_{w_3}^* \end{pmatrix} \mathbf{V}^{-1} \det(\mathbf{V}) \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \qquad \text{and}$$
$$s_v^* = \det(\mathbf{V}) \prod_{j=1}^{3} s_{w_j}$$

where $\mathbf{x}_{w_i}^* = \mathbf{x}_{w_i} \prod_{j \in [3] \setminus \{i\}} s_{w_j}$ and $\mathbf{V}$ is the Vandermonde matrix defined by $(c_{v,w_1}, c_{v,w_2}, c_{v,w_3})$. Since the invariant holds at w_1, w_2, w_3, we have $\|\mathbf{x}_{w_j}\| \leq \alpha_{i+1}$ and $\|s_{w_j}\| \leq \delta$ for all $j \in [3]$. We therefore have

$$\mathbf{A}_v \mathbf{x}_v^* = \mathbf{y}_v s_v^* \bmod q,$$
$$\|\mathbf{x}_v^*\| \leq 6\alpha_{i+1}\beta^3\delta^2 < \alpha_{i+1}\delta^3.$$

By the guarantee of O, we have

$$\mathbf{x}_v^* \cdot s_v^\dagger = \mathbf{x}_v^\dagger \cdot s_v^*,$$
$$\|\mathbf{x}_v^\dagger\| \leq \gamma \cdot \|\mathbf{x}_v^*\| \leq \alpha_{i+1}\gamma\delta^3 = \alpha_i,$$
$$\|s_v^\dagger\| \leq \|s_v^*\| \leq \delta^4.$$

If it happens that $\|s_v^\dagger\| \leq \delta$, then we have

$$((\mathbf{A}_v, \mathbf{y}_v), (\mathbf{x}_v^\dagger, s_v^\dagger)) \in \mathcal{R}_{\mathcal{O},n,m_i,q,\alpha_i,\delta}^{\sim\mathsf{ISIS}}.$$

Otherwise, $\|s_v^\dagger\| > \delta$, and since $\mathcal{N}(s_v^\dagger) \leq \nu$, we have

$$((\mathbf{A}_v, \mathbf{y}_v), (\perp, s_v^\dagger)) \in \mathcal{R}_{\mathcal{O},\delta,\nu}^{\perp}.$$ ◊

Case 2b. In this case, $|\mathsf{children}(v)| = 4$ and

$$\exists w \in \mathsf{children}(v) : \ \mathbf{X}_v^* \begin{pmatrix} 1 \\ c_{(v,w)} \\ c_{(v,w)}^2 \end{pmatrix} s_w - \begin{pmatrix} \mathbf{x}_w c_{(v,w)} \\ \mathbf{x}_w \end{pmatrix} s_w^2 \neq \mathbf{0}.$$

We argue that $\mathcal{W}_i^{\mathsf{O}}(\hat{\mathfrak{T}}_v)$ produces $(\mathbf{x}_v, s_v)$ satisfying

$$((\mathbf{A}_v, \mathbf{y}_v), (\mathbf{x}_v, s_v)) \in \mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m_i,q,\alpha_i'}.$$

Since the invariant holds for all $w \in \mathsf{children}(v)$, we have

$$\mathbf{A}_v \begin{pmatrix} \mathbf{x}_w c_{v,w} \\ \mathbf{x}_w \end{pmatrix} = f_{\mathbf{A}_v}(c_{v,w}) \cdot \mathbf{x}_w = h_v(c_{v,w}) \cdot s_w \bmod q.$$

Therefore, by Lemma G.5, $(\mathbf{X}_v^*, s_v^*) \leftarrow \mathsf{Lift}_i^{\mathsf{O}}((c_{v,w_j}, \mathbf{x}_{w_j}, s_{w_j})_{j=1}^3)$ satisfies

$$\mathbf{A}_v \mathbf{X}_v^* \begin{pmatrix} 1 \\ C \\ C^2 \end{pmatrix} = h_v(C) s_v^* \bmod q$$

for all $C \in \mathcal{O}$, and thus for all c_{v,w_j} for all $w \in \mathsf{children}(v)$. Combining the two equations yields

$$\mathbf{A}_v \left(\mathbf{X}_v^* \begin{pmatrix} 1 \\ c_{v,w_j} \\ c_{v,w_j}^2 \end{pmatrix} s_{w_j} - \begin{pmatrix} \mathbf{x}_{w_j} c_{v,w_j} \\ \mathbf{x}_{w_j} \end{pmatrix} s_v^* \right) = \mathbf{0} \bmod q.$$

Since there exists $w \in \mathsf{children}(v)$ such that

$$\mathbf{x}_v^{(w)} \coloneqq \mathbf{X}_v^* \begin{pmatrix} 1 \\ c_{v,w} \\ c_{v,w}^2 \end{pmatrix} s_w - \begin{pmatrix} \mathbf{x}_w c_{v,w} \\ \mathbf{x}_w \end{pmatrix} s_v^* \neq \mathbf{0}$$

$\mathcal{W}_i^{\mathsf{O}}(\hat{\mathfrak{T}}_v)$ would output $(\mathbf{x}_v^{(w)}, 1)$. Note that $\mathbf{A}_v \mathbf{x}_v^{(w)} = \mathbf{0} \bmod q$. We next analyse the norm of $\mathbf{x}_v^{(w)}$.

Recall that $\mathbf{X}_v^*$ is computed as

$$\mathbf{X}_v^* = \begin{pmatrix} \mathbf{x}_1^* c_1 & \mathbf{x}_2^* c_2 & \mathbf{x}_3^* c_3 \\ \mathbf{x}_1^* & \mathbf{x}_2^* & \mathbf{x}_3^* \end{pmatrix} \mathbf{V}^{-1} \det(\mathbf{V})$$

where we abbreviated $c_i = c_{v,w_i}$ for $i \in [3]$, and

$$\mathbf{V}^{-1} \det(\mathbf{V}) = \begin{pmatrix} c_2 c_3 (c_2 - c_3) & -(c_2^2 - c_3^2) & c_2 - c_3 \\ c_3 c_1 (c_3 - c_1) & -(c_3^2 - c_1^2) & c_3 - c_1 \\ c_1 c_2 (c_1 - c_2) & -(c_1^2 - c_2^2) & c_1 - c_2 \end{pmatrix}.$$

It follows that each entry of

$$\mathbf{V}^{-1} \det(\mathbf{V}) \begin{pmatrix} 1 \\ c_{v,w} \\ c_{v,w}^2 \end{pmatrix}$$

is of norm at most $6\beta^4$. Also recall that $s_v^* = \det(\mathbf{V}) s_{w_1} s_{w_2} s_{w_3}$ where $\det(\mathbf{V}) = (c_1 - c_2)(c_2 - c_3)(c_3 - c_1)$ and thus

$$\|\mathbf{x}_v^{(w)}\| \leq 3 \cdot \alpha_{i+1} \cdot \beta \cdot 6\beta^4 \cdot \delta + \alpha_{i+1} \cdot \beta \cdot (2\beta)^3 \cdot \delta^3 \leq \alpha_{i+1} \delta^5 = \alpha_i'.$$

In other words,

$$((\mathbf{A}_v, \mathbf{y}_v), (\mathbf{x}_v^{(w)}, 1)) \in \mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m_i,q,\alpha_i'}$$

and thus the invariant holds at node v. $\lozenge$

Case 2c. In this case, $|\mathsf{children}(v)| = 4$,

$$s_v^* \cdot h_v(C) \neq f_{\mathbf{A}_v}(C) \cdot g_{\mathbf{x}_v^*}(C),$$

and, for all $w \in \mathsf{children}(v)$, it holds that

$$\mathbf{X}_v^* \begin{pmatrix} 1 \\ c_{(v,w)} \\ c_{(v,w)}^2 \end{pmatrix} s_w = \begin{pmatrix} \mathbf{x}_w c_{(v,w)} \\ \mathbf{x}_w \end{pmatrix} s_w^2. \tag{20}$$

We argue that the above constraints are impossible.

Note that Eq. (20) gives two expressions of $\mathbf{x}_w \cdot s_v^*$. Writing

$$\mathbf{X}_v^* = \begin{pmatrix} \mathbf{x}_{v,00}^* & \mathbf{x}_{v,01}^* & \mathbf{x}_{v,02}^* \\ \mathbf{x}_{v,10}^* & \mathbf{x}_{v,11}^* & \mathbf{x}_{v,12}^* \end{pmatrix}$$

and equating the two expressions of $\mathbf{x}_w \cdot s_v^*$ yields

$$\mathbf{x}_w \cdot s_v^* = \begin{pmatrix} \mathbf{x}_{v,00}^* & \mathbf{x}_{v,01}^* & \mathbf{x}_{v,02}^* \end{pmatrix} \begin{pmatrix} 1 \\ c_{v,w} \\ c_{v,w}^2 \end{pmatrix} s_w = \begin{pmatrix} \mathbf{x}_{v,10}^* & \mathbf{x}_{v,11}^* & \mathbf{x}_{v,12}^* \end{pmatrix} \begin{pmatrix} c_{v,w} \\ c_{v,w}^2 \\ c_{v,w}^3 \end{pmatrix} s_w,$$

$$\mathbf{x}_{v,00}^* + (\mathbf{x}_{v,01}^* - \mathbf{x}_{v,10}^*)c_{v,w} + (\mathbf{x}_{v,02}^* - \mathbf{x}_{v,11}^*)c_{v,w}^2 - \mathbf{x}_{v,12}^* c_{v,w}^3 = 0, \tag{21}$$

where the second equality holds by cancelling s_w on both sides, which is permitted since there are no zero-divisors in $\mathcal{O}$. Note that Eq. (21) is cubic polynomial over $\mathcal{O}$ with 4 distinct roots $(c_{v,w_j})_{j=1}^4$. Since there are no zero-divisors in $\mathcal{O}$, the only possibility is that the cubic polynomial is the zero polynomial. Therefore we have

$$\begin{aligned} \mathbf{x}_{v,00}^* = \mathbf{x}_{v,12}^* = \mathbf{0}, \\ \mathbf{x}_{v,01}^* = \mathbf{x}_{v,10}^* = \mathbf{x}_{v,0}^*, \\ \mathbf{x}_{v,02}^* = \mathbf{x}_{v,11}^* = \mathbf{x}_{v,1}^* \end{aligned}$$

for some $\mathbf{x}_{v,0}^*, \mathbf{x}_{v,1}^* \in \mathcal{O}^{m_{i+1}}$. Substituting these back into $\mathbf{X}_v^*$, we have

$$\mathbf{X}_v^* = \begin{pmatrix} \mathbf{0} & \mathbf{x}_{v,0}^* & \mathbf{x}_{v,1}^* \\ \mathbf{x}_{v,0}^* & \mathbf{x}_{v,1}^* & \mathbf{0} \end{pmatrix}.$$

Denote the middle column by

$$\mathbf{x}_v^* := \begin{pmatrix} \mathbf{x}_{v,0}^* \\ \mathbf{x}_{v,1}^* \end{pmatrix}.$$

It follows that

$$\begin{aligned} f_{\mathbf{A}_v}(C) \cdot g_{\mathbf{x}_v^*}(C) &= \mathbf{A}_v \begin{pmatrix} \mathbf{0} \\ \mathbf{x}_{v,0}^* \\ \mathbf{x}_{v,1}^* \end{pmatrix} + \mathbf{A}_v \begin{pmatrix} \mathbf{x}_{v,0}^* \\ \mathbf{x}_{v,1}^* \end{pmatrix} C + \mathbf{A}_v \begin{pmatrix} \mathbf{x}_{v,1}^* \\ \mathbf{0} \end{pmatrix} C^2 \bmod q \\ &= \mathbf{A}_v \mathbf{X}_v^* \begin{pmatrix} 1 \\ C \\ C^2 \end{pmatrix} \bmod q \\ &= h_v(C) s_v^* \bmod q \end{aligned}$$

which contradicts with the assumption that

$$s_v^* \cdot h_v(C) \neq f_{\mathbf{A}_v}(C) \cdot g_{\mathbf{x}_v^*}(C). \qquad \Diamond$$

Case 2d. In this case, $|\mathsf{children}(v)| = 4$,

$$\mathcal{N}(s_v^\dagger) > \nu,$$

and Eq. (20) holds for all $w \in \mathsf{children}(v)$. We argue that the above constraints are impossible.

Let ℓ_4 denote the leftmost leaf of the fourth child w_4 of v. Since $\mathfrak{T} \in \mathsf{Chains}(\mathcal{U}^{\mathsf{O}})$ and $\mathcal{N}(s_v^\dagger) > \nu$, the sequence of challenges $\mathbf{c}_{v,\ell_4}$ must have been chosen from

$$\mathfrak{C}_{\mathsf{SZ},\mu-i}(\mathbf{x}_v^\dagger, s_v^\dagger).$$

It thus holds that

$$g_{\mathbf{x}_v^\dagger/s_v^\dagger}(\mathbf{c}_{v,\ell_4}) \notin \mathcal{O}^{m_\mu}.$$

Applying the argument in Case 2c, we can deduce from Eq. (20) that $\mathbf{X}_v^*$ can be expressed as $\mathbf{X}_v^* = \begin{pmatrix} \mathbf{0} & \mathbf{x}_{v,0}^* & \mathbf{x}_{v,1}^* \\ \mathbf{x}_{v,0}^* & \mathbf{x}_{v,1}^* & \mathbf{0} \end{pmatrix}$. Denoting the middle column of $\mathbf{X}_v^*$ by

$$\mathbf{x}_v^* \coloneqq \begin{pmatrix} \mathbf{x}_{v,0}^* \\ \mathbf{x}_{v,1}^* \end{pmatrix},$$

we can substitute $\mathbf{X}_v^*$ back into Eq. (20) and obtain, for each $w \in \mathsf{children}(v)$,

$$\begin{aligned} g_{\mathbf{x}_v^*}(c_{v,w}) s_w &= \mathbf{x}_{v,0}^* + \mathbf{x}_{v,1}^* c_{v,w} = \mathbf{x}_w s_v^*, \\ g_{\mathbf{x}_v^\dagger/s_v^\dagger}(c_{v,w}) &= g_{\mathbf{x}_v^*/s_v^*}(c_{v,w}) = \mathbf{x}_w / s_w \end{aligned}$$

where for the first equality we again relied on the lack of zero-divisor in $\mathcal{O}$ to divide out $c_{v,w}$.

Since $((\mathbf{A}_u, \mathbf{y}_u), (\mathbf{x}_u, s_u)) \in \mathcal{R}_{\mathcal{O},n,m_j,q,\alpha_j,\delta}^{\sim\mathsf{ISIS}}$ for any node $u \in \mathfrak{T}_v \setminus \{v\}$, Eq. (20) must hold when substituting v with u, i.e. for all $t \in \mathsf{children}(u)$, it holds that

$$\mathbf{X}_u^* \begin{pmatrix} 1 \\ c_{(u,t)} \\ c_{(u,t)}^2 \end{pmatrix} s_t = \begin{pmatrix} \mathbf{x}_t c_{(u,t)} \\ \mathbf{x}_w \end{pmatrix} s_t^2.$$

We can thus apply the above argument recursively for all nodes along the path from v to ℓ_4. This yields

$$g_{\mathbf{x}_v^\dagger/s_v^\dagger}(\mathbf{c}_{v,\ell_4}) = \mathbf{x}_{\ell_4}.$$

In words, this means that if we honestly continue the execution of the protocol at node v using $\mathbf{x}_v^\dagger/s_v^\dagger$ as the intermediate witness, then on challenges $\mathbf{c}_{v,\ell_4}$ we would have derived the prover's message at node ℓ_4.

Recall that at every leaf node, in particular ℓ_4,

$$((\mathbf{A}_{\ell_4}, \mathbf{y}_{\ell_4}), (\mathbf{x}_{\ell_4}, 1)) \in \mathcal{R}_{\mathcal{O},n,m_\mu,q,\alpha_\mu,\delta_\mu}^{\sim\mathsf{ISIS}}.$$

which in particular implies $g_{\mathbf{x}_v^\dagger/s_v^\dagger}(\mathbf{c}_{v,\ell_4}) = \mathbf{x}_{\ell_4} \in \mathcal{O}^{m_\mu}$, a contradiction. ◊ □

Remark G.7. In the above analysis, for $\mathcal{O}$ other than $\mathbb{Z}$ or imaginary quadratic rings, such as cyclotomic rings, the extractor $\mathcal{W}$ may output a witness of the failure relation $\mathcal{R}_{\mathcal{O},\delta,\nu}^{\perp}$. This happens if and only if $\mathcal{W}$ encounters an element $s_v^\dagger$ with $\mathcal{N}(s_v^\dagger) \le \nu$ but $\|s_v^\dagger\| > \delta$. Note that it is always guaranteed that $\|s_v^\dagger\| \le \delta$. Consider the set

$$\frac{|\{s_v^\dagger \in \mathcal{O} : \mathcal{N}(s) \le \nu \wedge \delta < \|s\| \le \delta^4\}|}{|\{c \in \mathcal{O} : \|c\| \le \beta - 1\}|}$$

which compares the number of "bad" denominators $s_v^\dagger$, i.e. those having too low field norm but too high geometric norm, and the challenge set size. Since the prover has only limited control over the value of $s_v^\dagger$ even if they know the value of the first two challenges c_{w_1} and c_{w_2}, where w_1, w_2 are the first two children of v, the above ratio approximates the probability of obtaining a bad denominator with probability over the random choice of the first challenge c_{w_3}. It is an approximation because the space where c_{w_3} is sampled from is not the entire set $\mathfrak{C}$ but an overwhelming fraction of it, and because there might be collisions, i.e. two challenges giving the same denominator. Nevertheless, by Lemma H.13, we have that the ratio is negligible in φ for cyclotomic rings of degree φ. We therefore heuristically conjecture that the probability of $\mathcal{W}_i$ finding a witness of $\mathcal{R}_{\mathcal{O},\delta,\nu}^{\perp}$ is negligible.

We next analyse the knowledge error of $\mathcal{U}$.

Lemma G.8. *Let $\mathcal{O}, \beta, \varphi, \mu, \nu, \mathcal{C}$ satisfy the properties in Corollary H.11, namely:*

- *$\mathcal{O}$ is $\mathbb{Z}$, or the ring of integers of a cyclotomic field $\mathcal{K}$ of degree φ, or the ring of integers of an imaginary quadratic field $\mathcal{K} = \mathbb{Q}(\sqrt{\Delta})$ for a square-free integer $\Delta < 0$, with (η_1, η_2) the corresponding ring-dependent constants of Lemma H.7.*
- *$\mathcal{C} \coloneqq \mathcal{C}_\beta \coloneqq \{x \in \mathcal{O} : \|x\| \leq \beta\}$.*
- *$|\mathcal{C}_\beta| \geq 2\mu\,\eta_2\,\eta_1^{\mu-1}\,2^\lambda$.*
- *$\log_2 \nu \geq 8\varphi\mu^2 + \log_2(2\mu)\big(\lambda + 1 + \mu \log_2 \eta_1\big)$.*

The UCS $\mathcal{U}$ has knowledge error

$$\kappa(\mathcal{U}) = 1 - (1 - \tfrac{3}{|\mathfrak{C}|} - \varepsilon_{\mathsf{sz}})^\mu \leq \mu \cdot (\tfrac{3}{|\mathfrak{C}|} + \varepsilon_{\mathsf{sz}})$$

where $\varepsilon_{\mathsf{sz}} \leq 2^{-\lambda}$ by Corollary H.11, and depth $\mathsf{depth}(\mathcal{U}) \leq 4^\mu = (m/m_\mu)^2 \leq m^2$.

Proof. To bound $\kappa(\mathcal{U}_i \mid \mathfrak{C}^{\mu-i})$, i.e. the knowledge error of $\mathcal{U}_i$ over $\mathfrak{C}^{\mu-i}$, we need to maximise $\kappa(\mathcal{U}_i(\hat{\mathfrak{T}}_v) \mid \mathfrak{C}^{\mu-i})$, where $\mathfrak{T}_v$ is a tree rooted at v f depth $0 \leq i < \mu$, with subtrees $\mathfrak{T}_{w_1}, \ldots, \mathfrak{T}_{w_k}$, where $(w_1, \ldots, w_k) = \mathsf{children}(v)$. That is, we need to upper-bound

$$\Pr[\mathbf{c} \notin \mathcal{U}_i(\hat{\mathfrak{T}}_v)],$$

with probability taken over $\mathbf{c} \leftarrow_{\$} \mathfrak{C}^{\mu-i}$ sampled uniformly at random, over all choices of accepting $\hat{\mathfrak{T}}_v$ which is incomplete but consistent with $\mathcal{U}_i$ in the following sense:

- (Incompleteness.) $\mathcal{U}_i(\hat{\mathfrak{T}}_v) \neq \emptyset$.
- (Consistency.) For each $i \in [k]$, let ℓ_i be the left-most leaf in the subtree $\mathfrak{T}_{w_i}$, and let τ_{ℓ_i} be the transcript parsed from the root-to-ℓ_i path in $\mathfrak{T}$. It holds that $(\tau_{\ell_1}, \ldots, \tau_{\ell_k}) \in \mathsf{Chains}(\mathcal{U}_i)$.

By the construction of $\mathcal{U}_i$, it is immediate that we only need to consider $\hat{\mathfrak{T}}_v$ where v has $0 \leq k \leq 3$ children, since $\mathcal{U}_i(\hat{\mathfrak{T}}_v) = \emptyset$ for $k \geq 4$.

By the construction of $\mathcal{U}_i$, it is clear that if v has $k \in \{0, 1, 2\}$ children, then

$$\Pr[\mathbf{c} \notin \mathcal{U}_i(\hat{\mathfrak{T}}_v)] \leq 2/|\mathfrak{C}|$$

as at most $k \cdot |\mathfrak{C}^{\mu-(i+1)}|$ challenges lie in $\mathfrak{C}^{\mu-i} \setminus \mathsf{Uncharted}(\mathfrak{T}_v)$ and $k \leq 2$.

We next consider $k = 3$. Clearly, the set $\mathcal{U}_i(\hat{\mathfrak{T}}_v)$ in Case 4.1 is a subset of that in Case 4.2, and in Case 4.3 we have $\mathcal{U}_i(\hat{\mathfrak{T}}_v) = \emptyset$. It thus suffices to focus on Case 4.2, where

$$\mathcal{U}_i(\hat{\mathfrak{T}}_v) = \mathsf{Uncharted}(\mathfrak{T}_v) \cap \mathfrak{C}_{\mathsf{sz},\mu-i}(\mathbf{x}_v^\dagger, s_v^\dagger), \text{ where}$$
$$\mathfrak{C}_{\mathsf{sz},\mu-i}(\mathbf{x}_v^\dagger, s_v^\dagger) = \Big\{\mathbf{c} \in \mathfrak{C}^{\mu-i} : g_{\mathbf{x}_v^\dagger/s_v^\dagger}(\mathbf{c}) \notin \mathcal{O}^{m_\mu}\Big\}.$$

The m_μ coordinates of $g_{\mathbf{x}_v^\dagger}$ are multilinear polynomials in the $\mu - i$ challenges whose coefficients are the entries of $\mathbf{x}_v^\dagger$, and $g_{\mathbf{x}_v^\dagger/s_v^\dagger}(\mathbf{c}) \in \mathcal{O}^{m_\mu}$ if and only if $s_v^\dagger$ divides every coordinate of $g_{\mathbf{x}_v^\dagger}(\mathbf{c})$. Since $(\mathbf{x}_v^\dagger, s_v^\dagger)$ is a coprime fraction, the content of $g_{\mathbf{x}_v^\dagger}$, namely the ideal generated by the entries of $\mathbf{x}_v^\dagger$, is coprime to $s_v^\dagger$. Hence, by the multilinear composite Schwartz-Zippel lemma (Theorem H.1), applied with $w = m_\mu$ coordinates, $\mu - i$ variables and modulus $d = s_v^\dagger$, we have

$$\Pr[g_{\mathbf{x}_v^\dagger/s_v^\dagger}(\mathbf{c}) \in \mathcal{O}^{m_\mu}] \leq \varepsilon_{\mathsf{sz}}.$$

Here $\varepsilon_{\mathsf{sz}} \leq 2^{-\lambda}$ by Corollary H.11: the set $\mathfrak{C}_{\mathsf{sz},\mu-i}(\mathbf{x}_v^\dagger, s_v^\dagger)$ is output only when $b_1 = 0$ (Fig. 12), i.e. when $\mathcal{N}(s_v^\dagger) > \nu$, and the first condition of Corollary H.11 is monotone in the number of variables while the second is monotone in the number of variables and in the field norm of the modulus, so the conditions assumed in the statement for μ variables and the parameter ν imply those for $\mu - i$ variables and the modulus $s_v^\dagger$. Then, by a union bound, we have

$$\begin{aligned}\Pr[\mathbf{c} \notin \mathcal{U}_i(\hat{\mathfrak{T}}_v)] &\leq \Pr[\mathbf{c} \notin \mathsf{Uncharted}(\mathfrak{T}_v)] + \Pr[g_{\mathbf{x}_v^\dagger/s_v^\dagger}(\mathbf{c}) \in \mathcal{O}^{m_\mu}] \\ &\leq \tfrac{3}{|\mathfrak{C}|} + \varepsilon_{\mathsf{sz}}.\end{aligned}$$

This shows that $\kappa(\mathcal{U}_i \mid \mathfrak{C}^{\mu-i}) \le 3/|\mathfrak{C}| + \varepsilon_{\mathsf{SZ}}$, as claimed.

Furthermore, since $\mathcal{U}_i(\hat{\mathfrak{T}}_v) = \emptyset$ whenever v has at least four children, as noted above, each level contributes depth at most 4, so $\mathsf{depth}(\mathcal{U}) = \prod_{i=1}^{\mu} \mathsf{depth}^{(i)}(\mathcal{U}) \le 4^\mu = (m/m_\mu)^2 \le m^2$. □

Corollary G.9 (Knowledge Soundness of Lattice Bulletproofs). *Let $\mathcal{O}, n, m, q, \alpha_0, \alpha_0', \gamma, \delta, \nu$ be parameters as defined in Section G.4 where*

- *$\mathcal{O} = \mathbb{Z}$ and $\delta \ge \nu$, or*
- *$\mathcal{O}$ is the ring of integers of $\mathbb{Q}(\sqrt{\Delta})$, where $\Delta < 0$ is a square-free integer, and $\delta^2 \ge \nu$.*

In addition, let β and ν satisfy the conditions of Lemma G.8. The lattice Bulletproofs protocol is knowledge sound for the relation

$$\mathcal{R} \coloneqq \mathcal{R}^{\sim\mathsf{ISIS}}_{\mathcal{O},n,m,q,\alpha_0,\delta} \cup \mathcal{R}^{\mathsf{SIS}}_{\mathcal{O},n,m,q,\alpha_0'}$$

with knowledge error

$$\kappa \le \mu \cdot \left(\tfrac{3}{|\mathfrak{C}|} + \varepsilon_{\mathsf{SZ}}\right).$$

More precisely, there is a knowledge extractor with runtime overhead $\frac{4^\mu}{1-\kappa}$ which, for every statement stmt and every prover $\mathcal{P}^*$ with success probability $\varepsilon(\mathcal{P}^*, \mathsf{stmt})$, outputs a witness $\mathsf{wit} \in \mathcal{R}(\mathsf{stmt})$ with probability at least $\frac{\varepsilon(\mathcal{P}^*,\mathsf{stmt})-\kappa}{1-\kappa}$.

Proof. By Theorem G.6 the protocol is $(\mathcal{U}^{\mathsf{O}}, \mathcal{W}^{\mathsf{O}})$-adaptive special sound. By Lemma G.8 we have $\kappa(\mathcal{U}^{\mathsf{O}}) \le \mu \cdot \left(\frac{3}{|\mathfrak{C}|} + \varepsilon_{\mathsf{SZ}}\right)$ and $\mathsf{depth}(\mathcal{U}^{\mathsf{O}}) \le 4^\mu = (m/m_\mu)^2 \le m^2$. The family $\mathcal{U}^{\mathsf{O}}$ is natural (Definition 4.11): its depth $4^\mu = (m/m_\mu)^2 \le m^2$ is polynomial in $|\mathsf{stmt}|$; each $\mathcal{U}_i$ is monotonic by inspection; and $\mathcal{U}_i(\hat{\mathfrak{T}}_v)$ is efficiently decidable given the efficiently instantiable oracle O and hence, by rejection sampling, efficiently sampleable. The claim then follows from Theorem 4.12. □

Remark G.10. The conditions of Lemma G.8 and Corollary G.9 are met by parameters of bit-length polynomial in $\mu = \log_2(m/m_\mu)$, λ and φ. The condition on ν is the lower bound $\log_2 \nu \ge 8\varphi\mu^2 + \log_2(2\mu)(\lambda + 1 + \mu \log_2 \eta_1)$ with $\log_2 \eta_1 \le 2\varphi - 1$, and $\delta \ge \nu$ (resp. $\delta^2 \ge \nu$) with $\delta = 2^3\beta^3$ then holds once $\log_2 \beta \ge (\log_2 \nu - 3)/3$ (resp. $(\log_2 \nu - 6)/6$). The condition on $|\mathcal{C}_\beta|$ holds for $\mathcal{O} = \mathbb{Z}$ once $\beta \ge \mu\, 2^\lambda + \frac{1}{2}$, as $|\mathcal{C}_\beta| = 2\lfloor\beta\rfloor + 1 \ge 2\beta - 1$, and for imaginary quadratic $\mathcal{K}$ once $\beta \ge \max\left(e, 6\sqrt{\mu|\Delta|8^{\mu-1}2^\lambda}\right)$, as then $|\mathcal{C}_\beta| \ge \pi\beta^2/(4\sqrt{|\Delta|})$ by Lemma H.5; the cyclotomic case is analogous via the cyclotomic lower bound of Lemma H.5. Every other parameter of Section G.4 is a fixed polynomial in α, β, γ and m, e.g. $\log_2 \alpha_0 = \log_2 \alpha + 10\mu \log_2 \beta + \mu \log_2 \gamma + 9\mu$, so all of them have polynomial bit-length as well.

Remark G.11 (Comparison with [BF23]). For $\mathcal{O} = \mathbb{Z}$, Corollary G.9 analyses the lattice instantiation of the setting of [BF23], with the challenge set $\mathfrak{C} = \{x \in \mathbb{Z} : |x| \le \beta\}$ in place of their λ-bit challenges. Their Theorem 5 (ePrint) shows that Bulletproofs is $(4^{(\mu)}, 3\mu/2^\lambda, \Gamma^*, \phi)$-almost-special-sound, and their Theorem 4 turns this into the knowledge error $2\lambda(k+1)^\mu(\mu + t_V) \cdot \max(3\mu/2^\lambda, k2^{-\lambda}) + 2^{-\lambda}$ with $k = 4$ for a strict-time extractor, where t_V is the running time of the verifier. Corollary G.9 gives $\kappa \le \mu \cdot (3/|\mathfrak{C}| + \varepsilon_{\mathsf{SZ}})$ with no factor $(k+1)^\mu$. Corollary G.9 covers imaginary quadratic rings in addition to $\mathcal{O} = \mathbb{Z}$.

H Multilinear Composite Schwartz-Zippel over Rings

We prove a natural generalisation of the so-called multilinear composite Schwartz-Zippel lemmas [BF23, Theorems 1 and 2]. Throughout this section, for vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^\ell$, we write $\mathbf{a} \le \mathbf{b}$ if $a_j \le b_j$ for all $j \in [\ell]$.

H.1 Main Theorem

Theorem H.1 (Generalisation of [BF23, Theorem 1 (ePrint)]). *Let $\mathcal{O}$ be the ring of integers of a number field $\mathcal{K}$ of degree φ. Let $\mu, \ell, w \in \mathbb{N}$, $\mathbf{f} = (f^{(1)}, \ldots, f^{(w)}) \in \mathcal{O}[\mathbf{X}]^w$ be a tuple of μ-variate multilinear polynomials, $d \in \mathcal{O}$ with ideal factorisation $\langle d\rangle = \prod_{j=1}^{\ell} \mathfrak{p}_j^{r_j}$ satisfying $\mathsf{cont}(\mathbf{f}) + \langle d\rangle = \langle 1\rangle$, where $\mathsf{cont}(\mathbf{f}) \coloneqq \sum_{c=1}^{w} \mathsf{cont}(f^{(c)})$ denotes the content of $\mathbf{f}$, i.e. the ideal generated by the coefficients of all the $f^{(c)}$, and $\mathcal{C} \coloneqq \mathcal{C}_\beta \coloneqq \{x \in \mathcal{O} : \|x\| \le \beta\}$ for some $\beta > 0$. For $j \in [\ell]$, $k \in [\mu]$, let $Z_{j,k} \sim \mathsf{Geo}(1 - 1/\mathcal{N}(\mathfrak{p}_j))$*

be independent random variables, each distributed geometrically with parameter $1 - 1/\mathcal{N}(\mathfrak{p}_j)$*. Let* $\eta_1 \geq 1$ *and* $\eta_2 \geq 0$ *be such that, for every non-zero ideal* $\mathcal{I} \subseteq \mathcal{O}$ *and every* $a \in \mathcal{O}$*,*

$$|(a + \mathcal{I}) \cap \mathcal{C}_\beta| \leq \eta_1 \cdot \frac{|\mathcal{C}_\beta|}{\mathcal{N}(\mathcal{I})} + \eta_2. \tag{22}$$

It holds that

$$\begin{aligned}\varepsilon_{\mathtt{SZ}} &\coloneqq \Pr\Big[\forall c \in [w],\ f^{(c)}(\mathbf{x}) = 0 \bmod d \ \Big|\ \mathbf{x} \leftarrow_{\$} \mathcal{C}^\mu\Big] \\ &\leq \eta_1^\mu \cdot \Pr\left[\sum_{k=1}^{\mu} \mathbf{Z}_k \geq \mathbf{r}\right] + \frac{\mu\, \eta_2\, \eta_1^{\mu-1}}{|\mathcal{C}_\beta|}\end{aligned}$$

where $\mathbf{Z}_k \coloneqq (Z_{1,k}, \ldots, Z_{\ell,k})$ *and* $\mathbf{r} \coloneqq (r_1, \ldots, r_\ell)$*. Admissible pairs* (η_1, η_2) *for* $\mathcal{O} = \mathbb{Z}$*, for cyclotomic fields and for imaginary quadratic fields are given in Lemma H.7.*

Remark H.2. The term $\Pr[\sum_{k=1}^{\mu} \mathbf{Z}_k \geq \mathbf{r}]$ appearing in the above bound can be written as

$$\Pr\left[\sum_{k=1}^{\mu} \mathbf{Z}_k \geq \mathbf{r}\right] = \prod_{j=1}^{\ell} I_{1/\mathcal{N}(\mathfrak{p}_j)}(r_j, \mu)$$

where $I_{1/p}(r, \mu) = (1-1/p)^\mu \sum_{j=r}^{\infty} \binom{\mu+j-1}{j} (1/p)^j$ is the regularised (incomplete) beta function. Convenient facts about $I_{1/p}(r, \mu)$ were given in [BF23, Section 3].

Below, we prove Theorem H.1.

Proof. For a given μ-linear coordinate $f^{(c)}$ and $\mathbf{x}_i = (x_1, \ldots, x_i) \in \mathcal{O}^i$ with $1 \leq i \leq \mu$, write

$$f_i^{(c)}[\mathbf{x}_i](X_{i+1}, \ldots, X_\mu) \coloneqq f^{(c)}(x_1, \ldots, x_i, X_{i+1}, \ldots, X_\mu)$$

for the $(\mu - i)$-linear polynomial in $(X_{i+1}, \ldots, X_\mu)$, and write $\mathbf{f}_i[\mathbf{x}_i] \coloneqq (f_i^{(1)}[\mathbf{x}_i], \ldots, f_i^{(w)}[\mathbf{x}_i])$ for the tuple of partial evaluations.

For $k \in [\mu]$, let $\mathbf{Y}_k \coloneqq (Y_{1,k}, \ldots, Y_{\ell,k})$ be a sequence of random variables representing the multiplicities of $\mathfrak{p}_j$ in $\mathsf{cont}(\mathbf{f}_k[\mathbf{x}_k])$ where $\mathbf{x}_k \leftarrow_{\$} \mathcal{C}^k$ for $j \in [\ell]$. In other words, if $\mathsf{cont}(\mathbf{f}_k[\mathbf{x}_k]) = \prod_{j \in J} \mathfrak{p}_j^{y_j}$, then $Y_{j,k} = y_j$. We naturally extend the definition to $k = 0$ by setting $\mathbf{Y}_0 = \mathbf{0}$. This is justified because $\mathbf{f}_0[] = \mathbf{f}$ and $\mathsf{cont}(\mathbf{f}) + \langle d \rangle = \langle 1 \rangle$. We see that, for any $k \in [\mu]$, the following events are equivalent:

$$\forall c \in [w],\ f_k^{(c)}[\mathbf{x}_k] = 0 \bmod d \qquad \text{and} \qquad \mathbf{Y}_k \geq \mathbf{r},$$

since all coordinates of $\mathbf{f}_k[\mathbf{x}_k]$ vanish modulo $\mathfrak{p}_j^{r_j}$ if and only if $\mathfrak{p}_j^{r_j}$ divides the content $\mathsf{cont}(\mathbf{f}_k[\mathbf{x}_k])$.

Consider the following induction hypothesis for $i \in [\mu]$:

$$\forall\ \mathbf{y} \in \mathbb{Z}^\ell,\ \Pr[\mathbf{Y}_i \geq \mathbf{y}] \leq \eta_1^i \cdot \Pr\left[\sum_{k=1}^{i} \mathbf{Z}_k \geq \mathbf{y}\right] + \frac{i\, \eta_2\, \eta_1^{i-1}}{|\mathcal{C}_\beta|}. \tag{23}$$

It is clear that if the induction hypothesis holds for $i = \mu$ and $\mathbf{y} = \mathbf{r}$, then the theorem statement holds.

To proceed with the induction, we will repeatedly use a core lemma, Lemma H.3, which states that, for any $i \in [\mu]$,

$$\forall\ \mathbf{y}, \mathbf{y}' \in \mathbb{Z}^\ell \text{ with } \mathbf{y}' \geq \mathbf{0},\ \Pr[\mathbf{Y}_i \geq \mathbf{y} + \mathbf{Y}_{i-1} \mid \mathbf{Y}_{i-1} = \mathbf{y}'] \leq \eta_1 \cdot \Pr[\mathbf{Z}_i \geq \mathbf{y}] + \frac{\eta_2}{|\mathcal{C}_\beta|}.$$

Setting $i = 1$ in Lemma H.3 and realising that $\mathbf{Y}_0 \equiv \mathbf{0}$ yield the base case $i = 1$ of the induction hypothesis (Eq. (23)).

Next, suppose that the induction hypothesis holds for some $i \in [\mu - 1]$. We will prove the hypothesis for $i+1$. Fix any $\mathbf{y}_{i+1} \in \mathbb{Z}^\ell$. We observe that

$$\begin{aligned}
&\Pr[\mathbf{Y}_{i+1} \geq \mathbf{y}_{i+1}] \\
&= \sum_{\mathbf{y}_i \in \mathbb{Z}^\ell : \mathbf{y}_i \geq \mathbf{0}} \Pr[\mathbf{Y}_{i+1} \geq \mathbf{y}_{i+1} - \mathbf{y}_i + \mathbf{Y}_i \mid \mathbf{Y}_i = \mathbf{y}_i] \cdot \Pr[\mathbf{Y}_i = \mathbf{y}_i] \\
&\leq \sum_{\mathbf{y}_i \in \mathbb{Z}^\ell : \mathbf{y}_i \geq \mathbf{0}} \left(\eta_1 \cdot \Pr[\mathbf{Z}_{i+1} \geq \mathbf{y}_{i+1} - \mathbf{y}_i] + \frac{\eta_2}{|\mathcal{C}_\beta|} \right) \cdot \Pr[\mathbf{Y}_i = \mathbf{y}_i] \qquad (24) \\
&= \eta_1 \cdot \sum_{\mathbf{y}_i \in \mathbb{Z}^\ell} \Pr[\mathbf{Z}_{i+1} \geq \mathbf{y}_{i+1} - \mathbf{y}_i] \cdot \Pr[\mathbf{Y}_i = \mathbf{y}_i] + \frac{\eta_2}{|\mathcal{C}_\beta|}
\end{aligned}$$

where the restriction to $\mathbf{y}_i \geq \mathbf{0}$ in the sum is due to $\Pr[\mathbf{Y}_i = \mathbf{y}_i] = 0$ for $\mathbf{y}_i \not\geq \mathbf{0}$, the inequality is obtained by applying Lemma H.3 with $\mathbf{y} = \mathbf{y}_{i+1} - \mathbf{y}_i$ and $\mathbf{y}' = \mathbf{y}_i$, and the last equality uses $\sum_{\mathbf{y}_i} \Pr[\mathbf{Y}_i = \mathbf{y}_i] = 1$. Then, by some routine calculation (shown below), we observe that

$$\begin{aligned}
&\sum_{\mathbf{y}_i \in \mathbb{Z}^\ell} \Pr[\mathbf{Z}_{i+1} \geq \mathbf{y}_{i+1} - \mathbf{y}_i] \cdot \Pr[\mathbf{Y}_i = \mathbf{y}_i] \\
&= \sum_{\mathbf{y}' \in \mathbb{Z}^\ell} \Pr[\mathbf{Z}_{i+1} \geq \mathbf{y}'] \Pr[\mathbf{Y}_i = \mathbf{y}_{i+1} - \mathbf{y}'] \quad /\!/\ \mathbf{y}' \coloneqq \mathbf{y}_{i+1} - \mathbf{y}_i \\
&= \sum_{\mathbf{y}' \in \mathbb{Z}^\ell} \sum_{\mathbf{t} \geq \mathbf{0}} \Pr[\mathbf{Z}_{i+1} = \mathbf{y}' + \mathbf{t}] \Pr[\mathbf{Y}_i = \mathbf{y}_{i+1} - \mathbf{y}'] \\
&= \sum_{\mathbf{y}'' \in \mathbb{Z}^\ell} \sum_{\mathbf{t} \geq \mathbf{0}} \Pr[\mathbf{Z}_{i+1} = \mathbf{y}''] \Pr[\mathbf{Y}_i = \mathbf{y}_{i+1} - \mathbf{y}'' + \mathbf{t}] \quad /\!/\ \mathbf{y}'' = \mathbf{y}' + \mathbf{t} \\
&= \sum_{\mathbf{y}'' \in \mathbb{Z}^\ell} \Pr[\mathbf{Z}_{i+1} = \mathbf{y}''] \Pr[\mathbf{Y}_i \geq \mathbf{y}_{i+1} - \mathbf{y}''].
\end{aligned}$$

Finally, applying the induction hypothesis to each term, we see that the above quantity satisfies

$$\begin{aligned}
&\sum_{\mathbf{y}'' \in \mathbb{Z}^\ell} \Pr[\mathbf{Z}_{i+1} = \mathbf{y}''] \Pr[\mathbf{Y}_i \geq \mathbf{y}_{i+1} - \mathbf{y}''] \\
&\leq \sum_{\mathbf{y}'' \in \mathbb{Z}^\ell} \Pr[\mathbf{Z}_{i+1} = \mathbf{y}''] \left(\eta_1^i \cdot \Pr\left[\sum_{k=1}^{i} \mathbf{Z}_k \geq \mathbf{y}_{i+1} - \mathbf{y}'' \right] + \frac{i\,\eta_2\,\eta_1^{i-1}}{|\mathcal{C}_\beta|} \right) \\
&= \eta_1^i \cdot \Pr\left[\sum_{k=1}^{i+1} \mathbf{Z}_k \geq \mathbf{y}_{i+1} \right] + \frac{i\,\eta_2\,\eta_1^{i-1}}{|\mathcal{C}_\beta|},
\end{aligned}$$

where the equality uses $\sum_{\mathbf{y}''} \Pr[\mathbf{Z}_{i+1} = \mathbf{y}''] = 1$ and the independence of $\mathbf{Z}_{i+1}$ from $\mathbf{Z}_1, \ldots, \mathbf{Z}_i$. Substituting into Eq. (24) gives

$$\Pr[\mathbf{Y}_{i+1} \geq \mathbf{y}_{i+1}] \leq \eta_1^{i+1} \cdot \Pr\left[\sum_{k=1}^{i+1} \mathbf{Z}_k \geq \mathbf{y}_{i+1} \right] + \frac{(i\,\eta_1^i + 1)\,\eta_2}{|\mathcal{C}_\beta|} \leq \eta_1^{i+1} \cdot \Pr\left[\sum_{k=1}^{i+1} \mathbf{Z}_k \geq \mathbf{y}_{i+1} \right] + \frac{(i+1)\,\eta_2\,\eta_1^i}{|\mathcal{C}_\beta|},$$

where the last inequality uses $\eta_1 \geq 1$. This is the induction hypothesis for $i+1$. □

H.2 Proof of Inductive Step

Below, we prove Lemma H.3 which was used in the inductive step in the proof of Theorem H.1.

Lemma H.3. *Let the notation be as in Theorem H.1, to be recalled below: Let $\mathcal{O}$ be the ring of integers of number field $\mathcal{K}$ of degree φ. Let $\mu, \ell, w \in \mathbb{N}$, $\mathbf{f} = (f^{(1)}, \ldots, f^{(w)}) \in \mathcal{O}[\mathbf{X}]^w$ be a tuple of μ-variate multilinear polynomials, $d \in \mathcal{O}$ with ideal factorisation $\langle d \rangle = \prod_{j=1}^{\ell} \mathfrak{p}_j^{r_j}$, and $\mathcal{C} \coloneqq \mathcal{C}_\beta \coloneqq \{x \in \mathcal{O} : \|x\| \leq \beta\}$ for some $\beta > 0$. For $j \in [\ell]$, $k \in [\mu]$, let $Z_{j,k} \sim \mathsf{Geo}(1 - 1/\mathcal{N}(\mathfrak{p}_j))$ be independent random variables,*

each distributed geometrically with parameter $1 - 1/\mathcal{N}(\mathfrak{p}_j)$*, and* $Y_{j,k}$ *be a random variable representing the multiplicity of* $\mathfrak{p}_j$ *in* $\mathsf{cont}(\mathbf{f}_k[\mathbf{x}_k])$ *where* $\mathbf{x}_k \leftarrow_\$ \mathcal{C}^k$*. Let* $\eta_1 \geq 1$ *and* $\eta_2 \geq 0$ *satisfy Eq.* (22) *for every non-zero ideal* $\mathcal{I} \subseteq \mathcal{O}$ *and every* $a \in \mathcal{O}$*. For any* $k \in [\mu]$ *and* $\mathbf{y}, \mathbf{y}' \in \mathbb{Z}^\ell$ *with* $\mathbf{y}' \geq \mathbf{0}$*, it holds that*

$$\Pr[\mathbf{Y}_k \geq \mathbf{y} + \mathbf{Y}_{k-1} \mid \mathbf{Y}_{k-1} = \mathbf{y}'] \leq \eta_1 \cdot \Pr[\mathbf{Z}_k \geq \mathbf{y}] + \frac{\eta_2}{|\mathcal{C}_\beta|}$$

where $\mathbf{Z}_k \coloneqq (Z_{1,k}, \ldots, Z_{\ell,k})$*.*

Proof (Proof of Lemma H.3). It suffices to consider the case that $y_j > 0$ for some $j \in [\ell]$ (i.e. $\mathbf{y} \not\leq \mathbf{0}$); otherwise $\mathbf{Z}_k \geq \mathbf{0} \geq \mathbf{y}$, so $\Pr[\mathbf{Z}_k \geq \mathbf{y}] = 1$ and the statement is trivial since $\eta_1 \geq 1$. Let $J \subseteq [\ell]$ denote the set

$$J \coloneqq \{j \in [\ell] : y_j > 0\}.$$

We note that multiplicities are non-decreasing in k: since each $f_k^{(c)}[\mathbf{x}_k]$ arises from $f_{k-1}^{(c)}[\mathbf{x}_{k-1}]$ by substituting $X_k = x_k$, we have $\mathsf{cont}(\mathbf{f}_{k-1}[\mathbf{x}_{k-1}]) \mid \mathsf{cont}(\mathbf{f}_k[\mathbf{x}_k])$, and hence $Y_{j,k} \geq Y_{j,k-1}$ for every j. For $j \notin J$ we have $y_j \leq 0$, so, conditioned on $\mathbf{Y}_{k-1} = \mathbf{y}'$, the inequality $Y_{j,k} \geq y'_j + y_j$ holds automatically. It therefore suffices to control the coordinates $j \in J$. In what follows, we treat $\mathfrak{p}_j^{y'_j+y_j} = \mathcal{O}$ if $y'_j + y_j \leq 0$ (which happens only when $j \notin J$).

Fix $k \in [\mu]$. For each coordinate $c \in [w]$, write

$$f_{k-1}^{(c)}[\mathbf{x}_{k-1}](X_k, X_{k+1}, \ldots, X_\mu) = h_k^{(c)}(X_{k+1}, \ldots, X_\mu) + X_k \cdot g_k^{(c)}(X_{k+1}, \ldots, X_\mu)$$

where $h_k^{(c)}$ and $g_k^{(c)}$ are $(\mu - k)$-linear polynomials over $\mathcal{O}$. We make two observations.

First, conditioned on the event $\mathbf{Y}_{k-1} = \mathbf{y}'$, for every $j \in [\ell]$ we have

$$\forall c \in [w] :\ h_k^{(c)} = g_k^{(c)} = 0 \bmod \mathfrak{p}_j^{y'_j}, \quad \text{and} \quad \exists c \in [w] :\ h_k^{(c)} \neq 0 \bmod \mathfrak{p}_j^{y'_j+1} \vee g_k^{(c)} \neq 0 \bmod \mathfrak{p}_j^{y'_j+1},$$

for otherwise the multiplicity of $\mathfrak{p}_j$ in the content $\mathsf{cont}(\mathbf{f}_{k-1}[\mathbf{x}_{k-1}]) = \sum_{c=1}^{w} \mathsf{cont}(f_{k-1}^{(c)}[\mathbf{x}_{k-1}])$ would not equal y'_j. This means that, for $j \in J$, since $y_j > 0$, there exist a coordinate $c_j \in [w]$ and a monomial (of $(X_{k+1}, \ldots, X_\mu)$) whose coefficient $h_{k,j} \in \mathcal{O}$ in $h_k^{(c_j)}$ and coefficient $g_{k,j} \in \mathcal{O}$ in $g_k^{(c_j)}$ form a univariate affine polynomial with

$$h_{k,j} + X_k \cdot g_{k,j} \begin{cases} = 0 \bmod \mathfrak{p}_j^{y'_j} \\ \neq 0 \bmod \mathfrak{p}_j^{y'_j+1} \end{cases}.$$

Crucially, the coordinate c_j may depend on j: the joint coprimality $\mathsf{cont}(\mathbf{f}) + \langle d \rangle = \langle 1 \rangle$ guarantees that for each prime $\mathfrak{p}_j$ some coordinate attains the boundary multiplicity, but not necessarily the same coordinate across primes.

Second, conditioned on the event $\mathbf{Y}_{k-1} = \mathbf{y}'$, the following events are equivalent:

$$\mathbf{Y}_k \geq \mathbf{y} + \mathbf{Y}_{k-1} \qquad \text{and} \qquad \forall j \in [\ell],\ \forall c \in [w],\ h_k^{(c)} + x_k \cdot g_k^{(c)} = 0 \bmod \mathfrak{p}_j^{y'_j+y_j}.$$

Putting the above two observations together, we have

$$\begin{aligned} & \Pr[\mathbf{Y}_k \geq \mathbf{y} + \mathbf{Y}_{k-1} \mid \mathbf{Y}_{k-1} = \mathbf{y}'] \\ = \ & \Pr\Big[\forall j \in [\ell],\ \forall c \in [w],\ h_k^{(c)} + x_k \cdot g_k^{(c)} = 0 \bmod \mathfrak{p}_j^{y'_j+y_j} \ \Big|\ \mathbf{Y}_{k-1} = \mathbf{y}'\Big] \\ \leq \ & \Pr\Big[\forall j \in J,\ h_{k,j} + x_k \cdot g_{k,j} = 0 \bmod \mathfrak{p}_j^{y'_j+y_j} \ \Big|\ \mathbf{Y}_{k-1} = \mathbf{y}'\Big], \end{aligned}$$

where the inequality holds because, for each $j \in J$, the polynomials $h_{k,j}$ and $g_{k,j}$ are coefficients of a monomial in coordinate c_j, so the event in the second line (all coordinates vanishing modulo $\mathfrak{p}_j^{y'_j+y_j}$, in particular coordinate c_j) forces $h_{k,j} + x_k \cdot g_{k,j} = 0 \bmod \mathfrak{p}_j^{y'_j+y_j}$.

Conditioned on the event $\mathbf{Y}_{k-1} = \mathbf{y}'$, for each $j \in J$, since

$$h_{k,j} + X_k \cdot g_{k,j} \begin{cases} = 0 \bmod \mathfrak{p}_j^{y'_j} \\ \neq 0 \bmod \mathfrak{p}_j^{y'_j+1} \end{cases},$$

there is at most one solution $x_k^* \in \mathcal{O}/\mathfrak{p}_j^{y_j}$ satisfying

$$h_{k,j} + X_k \cdot g_{k,j} = 0 \bmod \mathfrak{p}_j^{y'_j+y_j}.$$

By the Chinese remainder theorem, the system of equations

$$\forall j \in J,\ h_{k,j} + X_k \cdot g_{k,j} = 0 \bmod \mathfrak{p}_j^{y'_j+y_j}$$

admits at most one solution $x_k^* \in \mathcal{O}/\prod_{j\in J} \mathfrak{p}_j^{y_j}$.

Consider the ideal $\mathcal{I} := \prod_{j\in J} \mathfrak{p}_j^{y_j}$. Denote $N := \mathcal{N}(\mathcal{I}) = \prod_{j\in J} \mathcal{N}(\mathfrak{p}_j)^{y_j}$. Conditioned on the event $\mathbf{Y}_{k-1} = \mathbf{y}'$, and further on any value of $\mathbf{x}_{k-1}$ consistent with it, the coset $x_k^* + \mathcal{I}$ is fixed while x_k remains uniformly distributed over $\mathcal{C}$, so by Eq. (22) the probability of x_k falling into the coset $x_k^* + \mathcal{I}$ is

$$\Pr[x_k = x_k^* \bmod \mathcal{I}] \le \frac{|(x_k^* + \mathcal{I}) \cap \mathcal{C}_\beta|}{|\mathcal{C}_\beta|} \le \frac{\eta_1}{N} + \frac{\eta_2}{|\mathcal{C}_\beta|}.$$

To complete the proof, it remains to compute $\Pr[\mathbf{Z}_k \ge \mathbf{y}]$. For each $j \in J$, since $Z_{j,k} \sim \mathsf{Geo}(1 - 1/\mathcal{N}(\mathfrak{p}_j))$ and $y_j > 0$,

$$\Pr[Z_{j,k} \ge y_j] = \left(\frac{1}{\mathcal{N}(\mathfrak{p}_j)}\right)^{y_j},$$

while for $j \notin J$ we have $y_j \le 0$ and hence $\Pr[Z_{j,k} \ge y_j] = 1$, as $Z_{j,k} \ge 0$. Therefore

$$\Pr[\mathbf{Z}_k \ge \mathbf{y}] = \prod_{j\in J} \left(\frac{1}{\mathcal{N}(\mathfrak{p}_j)}\right)^{y_j} = \frac{1}{N}.$$ □

H.3 Coset-Counting Constants

The proof of Lemma H.3 rests on the coset-counting condition Eq. (22). Below, we establish admissible constants (η_1, η_2) for $\mathcal{O} = \mathbb{Z}$ and for the rings of integers of cyclotomic fields and imaginary quadratic fields. For this, we recall some known lower and upper bounds of the number of lattice points in a convex body.

Lemma H.4 ([FHM20, Theorems 1 and 7]). *Let $\mathcal{D} \subset \mathbb{R}^n$ be a convex body containing the origin, and let $\Lambda \subseteq \mathbb{R}^n$ be a full-rank lattice. If $\mathcal{D}$ is symmetric around the origin then:*

$$2^{-n} \cdot \frac{\mathrm{vol}(\mathcal{D})}{\det(\Lambda)} \le |\mathcal{D} \cap \Lambda|.$$

If $\mathcal{D}$ contains n linearly independent vectors of Λ, then

$$|\mathcal{D} \cap \Lambda| \le (n+1)! \cdot \frac{\mathrm{vol}(\mathcal{D})}{\det(\Lambda)}.$$

Alternatively, for $\Lambda = \mathbb{Z}^n$, applying the theorem of Davenport [Dav51], we get

$$||\mathcal{D} \cap \mathbb{Z}^n| - \mathrm{vol}(\mathcal{D})| \le \sum_{j=0}^{n-1} \mathrm{vol}_j(\mathcal{D}),$$

where $\mathrm{vol}_j(\mathcal{D})$ is the sum of the volumes of j-dimensional projections of $\mathcal{D}$, and $\mathrm{vol}_0(\mathcal{D}) = 1$.

We use the above to lower and upper bound $|\mathcal{C}_\beta|$.

Lemma H.5. *Let $\mathcal{O}$ be the ring of integers of a number field $\mathcal{K}$ of degree $\varphi = \varphi_1 + 2\varphi_2$, where φ_1 and φ_2 are the number of real embeddings and pairs of complex embeddings of $\mathcal{K}$. For sufficiently large $\beta > 0$, let $\mathcal{C}_\beta := \{x \in \mathcal{O} : \|x\| \le \beta\}$. It holds that*

$$2^{-\varphi_2} \cdot \frac{\pi^{\varphi_2}\beta^\varphi}{|\Delta_\mathcal{K}|^{1/2}} \le |\mathcal{C}_\beta| \le (\varphi+1)! \cdot \frac{2^{\varphi_1}(2\pi)^{\varphi_2}\beta^\varphi}{|\Delta_\mathcal{K}|^{1/2}}$$

where $\Delta_{\mathcal{K}}$ is the discriminant of $\mathcal{K}$. Furthermore, if $\mathcal{K} = \mathbb{Q}(\sqrt{\Delta})$ for a square-free integer $\Delta < 0$ and $\beta \geq e$, then

$$\frac{\pi\beta^2}{4\sqrt{|\Delta|}} \leq |\mathcal{C}_\beta| \leq 2w_{\mathcal{K}}\beta^2(\ln\beta + 1),$$

where $w_{\mathcal{K}} = |\mathcal{O}^\times| \in \{2, 4, 6\}$ is the number of units (roots of unity). If $\mathcal{K}$ is cyclotomic, then

$$\left(\frac{\pi\beta^2}{2\varphi}\right)^{\varphi/2} \leq |\mathcal{C}_\beta| \leq (\varphi+1)! \cdot \left(\frac{2\pi\beta^2}{\sqrt{\varphi}}\right)^{\varphi/2},$$

assuming the conductor is greater than three.

Proof. We work in the canonical embedding: $\mathcal{O}$ is a full-rank lattice in the φ-dimensional real inner product space $H \subseteq \mathbb{C}^\varphi$ carrying the inner product of $\mathbb{C}^\varphi$, its covolume is $\det(\mathcal{O}) = |\Delta_{\mathcal{K}}|^{1/2}$ ([LPR13, Section 2.5.3]; for a general number field see [FHM20, Theorem 1]), and $\mathcal{C}_\beta = B_\beta \cap \mathcal{O}$ for the box $B_\beta \coloneqq \{\mathbf{z} \in H : \|\mathbf{z}\| \leq \beta\}$. In the coordinates of H, the box B_β is a product of φ_1 intervals $[-\beta, \beta]$, one per real embedding, and φ_2 discs of radius $\sqrt{2}\beta$, one per pair of complex embeddings, so

$$\mathrm{vol}(B_\beta) = 2^{\varphi_1}(2\pi)^{\varphi_2}\beta^\varphi.$$

The general version of the bounds follows by applying Lemma H.4 to the symmetric convex body B_β and the lattice $\mathcal{O}$, using $2^{-\varphi} \cdot 2^{\varphi_1}(2\pi)^{\varphi_2} = 2^{-\varphi_2}\pi^{\varphi_2}$ for the lower bound, provided β is at least the φth successive minimum of $\mathcal{O}$, so that B_β contains φ linearly independent vectors of $\mathcal{O}$ as the upper bound there requires.

We now specialise to the imaginary quadratic case $\mathcal{K} = \mathbb{Q}(\sqrt{\Delta})$, where $\varphi_1 = 0$, $\varphi_2 = 1$ and $\varphi = 2$. For the lower bound, the general lower bound specialises to

$$|\mathcal{C}_\beta| \geq 2^{-1} \cdot \frac{\pi\beta^2}{|\Delta_{\mathcal{K}}|^{1/2}} = \frac{\pi\beta^2}{2\sqrt{|\Delta_{\mathcal{K}}|}}.$$

Since $|\Delta_{\mathcal{K}}| = |\Delta|$ or $4|\Delta|$ according as $\Delta \equiv 1$ or $\Delta \equiv 2, 3 \bmod 4$, we have $|\Delta_{\mathcal{K}}| \leq 4|\Delta|$, and therefore $|\mathcal{C}_\beta| \geq \frac{\pi\beta^2}{4\sqrt{|\Delta|}}$. For the upper bound, the canonical embedding gives $\|x\| = \sqrt{|\mathcal{N}(x)|}$, so $\|x\| \leq \beta$ if and only if $|\mathcal{N}(x)| \leq \beta^2$. Counting the elements of $\mathcal{C}_\beta$ by their norm value,

$$|\mathcal{C}_\beta| = 1 + \sum_{a=1}^{\lfloor\beta^2\rfloor} |\{x \in \mathcal{O} : |\mathcal{N}(x)| = a\}|,$$

where the leading 1 counts $x = 0$. Each nonzero x with $|\mathcal{N}(x)| = a$ generates a principal ideal (x) of norm a, and a principal ideal has exactly $w_{\mathcal{K}} = |\mathcal{O}^\times|$ generators (its unit multiples), so $|\{x \in \mathcal{O} : |\mathcal{N}(x)| = a\}| = w_{\mathcal{K}} \cdot$ (number of principal ideals of norm a) $\leq w_{\mathcal{K}} \cdot$ (number of ideals of norm a). By [BS66, p. 231], the number of ideals of norm a is at most the number of divisors $d(a)$ of a, so

$$|\mathcal{C}_\beta| \leq 1 + w_{\mathcal{K}} \sum_{a=1}^{\lfloor\beta^2\rfloor} d(a).$$

By [BBR12, Theorem 1.1], the divisor summatory function satisfies $\sum_{a \leq x} d(a) \leq x\ln x + (2\gamma_{EM} - 1)x + 0.961\sqrt{x}$, where $\gamma_{EM} \approx 0.57721$ is the Euler-Mascheroni constant. Evaluating at $x = \beta^2$ gives $\sum_{a \leq \beta^2} d(a) \leq \beta^2(2\ln\beta + 2\gamma_{EM} - 1) + 0.961\beta$, and hence

$$|\mathcal{C}_\beta| \leq 2w_{\mathcal{K}}\beta^2\ln\beta + (2\gamma_{EM} - 1)w_{\mathcal{K}}\beta^2 + 0.961\, w_{\mathcal{K}}\beta + 1.$$

Finally, for $\beta \geq e$ the trailing terms are at most $2w_{\mathcal{K}}\beta^2$: dividing by $w_{\mathcal{K}}\beta^2$ and using $\beta \geq e$ and $w_{\mathcal{K}} \geq 2$,

$$(2\gamma_{EM} - 1) + \frac{0.961}{\beta} + \frac{1}{w_{\mathcal{K}}\beta^2} \leq 0.155 + \frac{0.961}{e} + \frac{1}{2e^2} < 0.58 < 2,$$

so $(2\gamma_{EM} - 1)w_{\mathcal{K}}\beta^2 + 0.961\, w_{\mathcal{K}}\beta + 1 \leq 2w_{\mathcal{K}}\beta^2$, giving

$$|\mathcal{C}_\beta| \leq 2w_{\mathcal{K}}\beta^2(\ln\beta + 1).$$

If $\mathcal{K}$ is cyclotomic with conductor $\mathfrak{f}$, its discriminant $\Delta_{\mathcal{K}}$ satisfies (see e.g. [LPR13, Section 2.5.3] for the equality and the upper bound)

$$\varphi^{\varphi/2} \leq |\Delta_{\mathcal{K}}| = \left(\frac{\mathfrak{f}}{\prod_{\text{prime } p|\mathfrak{f}} p^{1/(p-1)}}\right)^{\varphi} \leq \varphi^{\varphi},$$

where for power-of-2 $\mathfrak{f}$ the upper bound is tight. The lower bound is the inequality $\mathfrak{f}/P \geq \sqrt{\varphi}$ for $P \coloneqq \prod_{\text{prime } p|\mathfrak{f}} p^{1/(p-1)}$, raised to the φ-th power. For an odd prime p we have $p^{1/(p-1)} \leq \sqrt{p}$, and the product of the distinct primes dividing $\mathfrak{f}$ is at most $\mathfrak{f}$. If $\mathfrak{f}$ is odd, then $P \leq \sqrt{\mathfrak{f}}$, so $\mathfrak{f}/P \geq \sqrt{\mathfrak{f}} \geq \sqrt{\varphi}$. If $\mathfrak{f}$ is even, then the prime 2 contributes the factor 2 to P and the odd primes dividing $\mathfrak{f}$ contribute at most $\sqrt{\mathfrak{f}/2}$, so $P \leq \sqrt{2\mathfrak{f}}$ and $\mathfrak{f}/P \geq \sqrt{\mathfrak{f}/2} \geq \sqrt{\varphi}$, since φ counts the units of $\mathbb{Z}/\mathfrak{f}\mathbb{Z}$, none of which is even. The concrete version of the bounds thus follows. □

We also need the following bounds on the covering radius of an ideal lattice.

Remark H.6. Let $\mathcal{I} \subseteq \mathcal{O}$ be a non-zero ideal. The covering radius of $\mathcal{I}$ is bounded above by $\frac{\sqrt{\varphi}}{2}|\Delta_{\mathcal{K}}\mathcal{N}(\mathcal{I})|^{1/\varphi}$ [BF06, Proposition 4.2]: the proof of that proposition bounds the squared covering radius of $\mathcal{I}$ by $\frac{\varphi}{4}|\Delta_{\mathcal{K}}|^{1/\varphi}\text{covol}(\mathcal{I})^{2/\varphi}$ under every inner product $(x, y) \mapsto \text{Tr}(\alpha x\bar{y})$ on $\mathcal{K} \otimes_{\mathbb{Q}} \mathbb{R}$ with α totally positive, in particular under the inner product of $\mathbb{C}^{\varphi}$ ($\alpha = 1$), for which $\text{covol}(\mathcal{I}) = \mathcal{N}(\mathcal{I})|\Delta_{\mathcal{K}}|^{1/2}$. So in the case of the imaginary quadratic field $\mathcal{K} = \mathbb{Q}(\sqrt{\Delta})$, where $\Delta < 0$ is a square-free integer, $|\Delta_{\mathcal{K}}| = 4|\Delta|$ or $|\Delta|$, depending on whether $\Delta \equiv 2, 3 \pmod 4$ or not, and hence, the covering radius will be bounded above by $\sqrt{2|\Delta\mathcal{N}(\mathcal{I})|}$ or $\sqrt{|\Delta\mathcal{N}(\mathcal{I})|/2}$, respectively. In case of a cyclotomic field, the first and last minima of $\mathcal{I}$ as a lattice coincide, i.e. $\lambda_1(\mathcal{I}) = \lambda_{\varphi}(\mathcal{I})$. It then holds (see e.g. [LPR13, Section 2.5.3]) that the first and last minima satisfy

$$\lambda_{\varphi}(\mathcal{I}) = \lambda_1(\mathcal{I}) \leq \sqrt{\varphi}\mathcal{N}(\mathcal{I})^{1/\varphi}\sqrt{\Delta_{\mathcal{K}}^{1/\varphi}} \leq \varphi\mathcal{N}(\mathcal{I})^{1/\varphi}.$$

Let $r_{\mathcal{I}}$ denote the covering radius of $\mathcal{I}$. It holds that

$$2r_{\mathcal{I}} \leq \sqrt{\varphi}\lambda_{\varphi}(\mathcal{I}) \leq \varphi^{3/2}\mathcal{N}(\mathcal{I})^{1/\varphi},$$

i.e. $r_{\mathcal{I}} \leq \frac{\varphi^{3/2}}{2}\mathcal{N}(\mathcal{I})^{1/\varphi}$. Note that all geometric quantities in this remark (covering radius, minima) are measured in the canonical ℓ_2 norm, i.e. the norm of $\mathbb{C}^{\varphi}$ restricted to the image of the canonical embedding.

Lemma H.7 (Coset-counting constants). *Let $\mathcal{O}$ be the ring of integers of a number field $\mathcal{K}$ of degree φ, let $\beta > 0$, and let $\mathcal{C}_{\beta} \coloneqq \{x \in \mathcal{O} : \|x\| \leq \beta\}$. For every non-zero ideal $\mathcal{I} \subseteq \mathcal{O}$ and every $a \in \mathcal{O}$, the coset-counting condition Eq. (22) holds with*

1. $(\eta_1, \eta_2) = (1, 1)$ *if* $\mathcal{O} = \mathbb{Z}$;
2. $(\eta_1, \eta_2) = \left(2^{2\varphi-1}, (3\varphi^{5/4})^{\varphi}\right)$ *if $\mathcal{K}$ is cyclotomic;*
3. $(\eta_1, \eta_2) = \left(8, 13\sqrt{|\Delta|}\right)$ *if* $\mathcal{K} = \mathbb{Q}(\sqrt{\Delta})$ *for a square-free integer* $\Delta < 0$.

Proof. Write $N \coloneqq \mathcal{N}(\mathcal{I})$. For $\mathcal{O} = \mathbb{Z}$, the set $\mathcal{C}_{\beta} = \{-\lfloor\beta\rfloor, \ldots, \lfloor\beta\rfloor\}$ consists of $|\mathcal{C}_{\beta}| = 2\lfloor\beta\rfloor + 1$ consecutive integers, and any M consecutive integers contain at most $\lceil M/N\rceil \leq M/N + 1$ elements of a given residue class modulo N. This proves the first case.

For the remaining cases, we work in the canonical embedding with the conventions of Remark H.6: $\mathcal{O}$ is a full-rank lattice in the φ-dimensional real inner product space $H \subseteq \mathbb{C}^{\varphi}$ carrying the inner product of $\mathbb{C}^{\varphi}$, its covolume is $\text{covol}(\mathcal{O}) = |\Delta_{\mathcal{K}}|^{1/2}$ ([LPR13, Section 2.5.3]; for a general number field see [FHM20, Theorem 1]), and $\|\cdot\|$ is the ℓ_{∞}-norm on H, which is bounded by the ℓ_2-norm. For $t > 0$, write $B_t \coloneqq \{\mathbf{z} \in H : \|\mathbf{z}\| \leq t\}$, so that $\mathcal{C}_t = B_t \cap \mathcal{O}$ and $\text{vol}(B_t) = t^{\varphi} \cdot \text{vol}(B_1)$. Let $\rho \coloneqq r_{\mathcal{I}}$ be the covering radius of $\mathcal{I}$ in the ℓ_2-norm, and let $V \subseteq H$ be the Voronoi cell of $\mathcal{I}$ around the origin, i.e. the set of points of H at least as close to $\mathbf{0}$ as to any other point of $\mathcal{I}$. The cell V is a fundamental domain of $\mathcal{I}$, so $\text{vol}(V) = \text{covol}(\mathcal{I}) = N \cdot \text{covol}(\mathcal{O})$, and every point of V has ℓ_2-norm at most ρ. For distinct $x, x' \in (a + \mathcal{I}) \cap \mathcal{C}_{\beta}$, the translates $x + V$ and $x' + V$ have disjoint interiors, since $x - x' \in \mathcal{I} \setminus \{0\}$, and each translate $x + V$ is contained in $B_{\beta+\rho}$, since $\|x + \mathbf{v}\| \leq \|x\| + \|\mathbf{v}\| \leq \beta + \rho$ for every $\mathbf{v} \in V$. Comparing volumes,

$$|(a + \mathcal{I}) \cap \mathcal{C}_{\beta}| \cdot N \cdot \text{covol}(\mathcal{O}) \leq \text{vol}(B_{\beta+\rho}) = (\beta + \rho)^{\varphi} \cdot \text{vol}(B_1) \leq 2^{\varphi-1}\left(\beta^{\varphi} + \rho^{\varphi}\right) \cdot \text{vol}(B_1),$$

where the last step is the convexity of $t \mapsto t^\varphi$. By the lower bound of Lemma H.4 applied to the symmetric convex body B_β and the lattice $\mathcal{O}$, we have $\beta^\varphi \cdot \mathrm{vol}(B_1) = \mathrm{vol}(B_\beta) \le 2^\varphi \cdot \mathrm{covol}(\mathcal{O}) \cdot |\mathcal{C}_\beta|$. Substituting and dividing by $N \cdot \mathrm{covol}(\mathcal{O})$ gives

$$|(a+\mathcal{I}) \cap \mathcal{C}_\beta| \le 2^{2\varphi-1} \cdot \frac{|\mathcal{C}_\beta|}{N} + 2^{\varphi-1} \cdot \frac{\mathrm{vol}(B_1)}{\mathrm{covol}(\mathcal{O})} \cdot \frac{\rho^\varphi}{N}. \tag{25}$$

In the coordinates of H, the region B_1 is a product of φ_1 intervals $[-1,1]$, one per real embedding, and φ_2 discs of radius $\sqrt{2}$, one per pair of complex embeddings, so $\mathrm{vol}(B_1) = 2^{\varphi_1}(2\pi)^{\varphi_2}$.

If $\mathcal{K}$ is cyclotomic with $\mathcal{K} \neq \mathbb{Q}$ (the case $\mathcal{K} = \mathbb{Q}$ is the first case), then $\varphi_1 = 0$ and $\varphi_2 = \varphi/2$, $|\Delta_\mathcal{K}|^{1/2} \ge \varphi^{\varphi/4}$ (see the proof of Lemma H.5), and $\rho \le \frac{\varphi^{3/2}}{2} N^{1/\varphi}$ by Remark H.6. Hence the second summand of Eq. (25) is at most

$$2^{\varphi-1} \cdot \frac{(2\pi)^{\varphi/2}}{\varphi^{\varphi/4}} \cdot \frac{\varphi^{3\varphi/2}}{2^\varphi} = \frac{1}{2}\big(\sqrt{2\pi}\,\varphi^{5/4}\big)^\varphi \le (3\varphi^{5/4})^\varphi.$$

If $\mathcal{K} = \mathbb{Q}(\sqrt{\Delta})$, then $\varphi = 2$, $\varphi_1 = 0$, $\varphi_2 = 1$, $|\Delta_\mathcal{K}| \le 4|\Delta|$, and $\rho \le \sqrt{|\Delta_\mathcal{K}| N/2}$ by Remark H.6. Hence the second summand of Eq. (25) is at most

$$2 \cdot \frac{2\pi}{|\Delta_\mathcal{K}|^{1/2}} \cdot \frac{|\Delta_\mathcal{K}|}{2} = 2\pi |\Delta_\mathcal{K}|^{1/2} \le 4\pi\sqrt{|\Delta|} \le 13\sqrt{|\Delta|}.$$

In both cases the first summand of Eq. (25) is $2^{2\varphi-1} \cdot |\mathcal{C}_\beta|/N$, which for $\varphi = 2$ is $8 \cdot |\mathcal{C}_\beta|/N$. □

H.4 Inverse Bound

Our next goal is to upper bound

$$\Pr\left[\sum_{k=1}^{\mu} \mathbf{Z}_k \ge \mathbf{r}\right]$$

in terms of $\nu \coloneqq |\mathcal{N}(d)|$. Before proving the main result, we need the following lemma:

Lemma H.8 ([BF23, Lemma 10, Lemma 11 (ePrint)]). *For the regularised incomplete Beta function $I_\epsilon(k,\mu)$ with $\epsilon \in (0,1)$ and $k, \mu \in \mathbb{N}$:*

1. *if $\epsilon\mu \le \frac{1}{2}$, then $I_\epsilon(k,\mu) \le (\epsilon\mu)^k$; and*
2. *if $k \ge 2\mu$ and $\epsilon \le \frac{1}{2}$, then $I_\epsilon(k,\mu) \le \epsilon^k\, k^\mu$.*

Theorem H.9 (Inverse LCSZ, generalisation of [BF23, Theorem 2 (ePrint)]). *Let μ, d, $\mathbf{Z}_k$, and $\mathbf{r}$ be as in Theorem H.1, write $\nu \coloneqq |\mathcal{N}(d)|$, let φ be the degree of the number field $\mathcal{K}$, and let $\lambda > 0$.*

1. *If $\mu = 1$ and $\log_2 \nu \ge \lambda$, then $\Pr[\sum_{k=1}^{\mu} \mathbf{Z}_k \ge \mathbf{r}] \le 2^{-\lambda}$.*
2. *If $\mu \ge 2$, $\theta \ge \log_\mu 2$, and*

$$\log_2 \nu \;\ge\; 4\,\varphi\,\mu^{2+\theta} \;+\; \Big(1 + \tfrac{1}{\theta}\Big)\lambda,$$

then $\Pr[\sum_{k=1}^{\mu} \mathbf{Z}_k \ge \mathbf{r}] \le 2^{-\lambda}$.

In particular, for every $\mu \ge 1$,

$$\log_2 \nu \;\ge\; 8\,\varphi\,\mu^2 \;+\; \log_2(2\mu)\,\lambda \qquad \Longrightarrow \qquad \Pr\left[\sum_{k=1}^{\mu} \mathbf{Z}_k \ge \mathbf{r}\right] \le 2^{-\lambda}.$$

Proof. Write $P_j \coloneqq \mathcal{N}(\mathfrak{p}_j) \ge 2$ for $j \in [\ell]$, so that $\log_2 \nu = \sum_{j=1}^{\ell} r_j \log_2 P_j$, and let $w_j \coloneqq -\log_2 I_{1/P_j}(r_j, \mu) \ge 0$. By the remark following Theorem H.1, $\Pr[\sum_{k=1}^{\mu} \mathbf{Z}_k \ge \mathbf{r}] = \prod_{j=1}^{\ell} I_{1/P_j}(r_j,\mu) = 2^{-\sum_{j=1}^{\ell} w_j}$, so in both cases it suffices to show $\sum_{j=1}^{\ell} w_j \ge \lambda$.

The case $\mu = 1$. Here $I_{1/P}(r,1) = P^{-r}$, so $w_j = r_j \log_2 P_j$ and $\sum_{j=1}^{\ell} w_j = \log_2 \nu \geq \lambda$.

The case $\mu \geq 2$. Fix $\theta \geq \log_\mu 2$, so that $\mu^\theta \geq 2$, and set

$$F \coloneqq 2(1+\theta)\,\mu \log_2 \mu.$$

The heart of the proof is the following inequality, which holds for every prime ideal $\mathfrak{p}$ of norm $P \coloneqq \mathcal{N}(\mathfrak{p}) \geq 2$ and every integer $r \geq 1$:

$$r \log_2 P \;\leq\; \left(1+\tfrac{1}{\theta}\right)\left(-\log_2 I_{1/P}(r,\mu)\right) \;+\; F \cdot (P \overset{?}{<} \mu^{1+\theta}). \tag{26}$$

We first derive the theorem from Eq. (26) and then prove the inequality.

Summing Eq. (26) over the factorisation of $\langle d \rangle$ gives

$$\log_2 \nu = \sum_{j=1}^{\ell} r_j \log_2 P_j \;\leq\; \left(1+\tfrac{1}{\theta}\right) \sum_{j=1}^{\ell} w_j \;+\; F \cdot |\{j \in [\ell] : P_j < \mu^{1+\theta}\}|.$$

A prime ideal of norm $P < \mu^{1+\theta}$ lies above a rational prime q with $P = q^f$ for its residue degree $f \geq 1$, so $q \leq P < \mu^{1+\theta}$, and at most $[\mathcal{K} : \mathbb{Q}] = \varphi$ prime ideals lie above any rational prime, so the number of prime ideals of norm below $\mu^{1+\theta}$ is at most $\varphi \cdot \pi_{\mathrm{pr}}(\mu^{1+\theta})$, where π_{pr} denotes the prime-counting function. Using $\pi_{\mathrm{pr}}(x) \leq 1.3\,x/\ln x$ for $x > 1$ [RS62],

$$F \cdot \varphi\, \pi_{\mathrm{pr}}(\mu^{1+\theta}) \;\leq\; \varphi \cdot \frac{1.3\,\mu^{1+\theta}}{(1+\theta)\ln\mu} \cdot \frac{2(1+\theta)\,\mu \ln \mu}{\ln 2} \;=\; \frac{2.6}{\ln 2}\,\varphi\,\mu^{2+\theta} \;\leq\; 4\,\varphi\,\mu^{2+\theta}.$$

Hence $\log_2 \nu \leq 4\varphi\mu^{2+\theta} + (1+\frac{1}{\theta})\sum_{j=1}^{\ell} w_j$ for every d, and the hypothesis $\log_2 \nu \geq 4\varphi\mu^{2+\theta} + (1+\frac{1}{\theta})\lambda$ yields $\sum_{j=1}^{\ell} w_j \geq \lambda$.

Proof of Eq. (26). Fix $\mathfrak{p}$ and r as above and write $w \coloneqq -\log_2 I_{1/P}(r,\mu) \geq 0$. We distinguish three cases.

Case 1: $P \geq \mu^{1+\theta}$. Then $P \geq \mu \cdot \mu^\theta \geq 2\mu$, so $\mu/P \leq \frac{1}{2}$ and Lemma H.8(1) with $\epsilon = 1/P$ gives $I_{1/P}(r,\mu) \leq (\mu/P)^r$, i.e. $w \geq r(\log_2 P - \log_2 \mu)$. Since $\log_2 \mu \leq \frac{1}{1+\theta}\log_2 P$, this gives $w \geq \frac{\theta}{1+\theta}\, r \log_2 P$, which is Eq. (26).

For the remaining cases assume $P < \mu^{1+\theta}$, i.e. $\ln P < (1+\theta)\ln\mu$, and define the threshold

$$r_c \coloneqq \frac{2(1+\theta)\,\mu \ln \mu}{\ln P} > 2\mu, \qquad \text{so that } r_c \log_2 P = F.$$

Case 2: $P < \mu^{1+\theta}$ and $r \leq r_c$. Then $r \log_2 P \leq r_c \log_2 P = F \leq (1+\frac{1}{\theta})w + F$.

Case 3: $P < \mu^{1+\theta}$ and $r > r_c$. Then $r > 2\mu$ and $1/P \leq \frac{1}{2}$, so Lemma H.8(2) with $\epsilon = 1/P$ gives $I_{1/P}(r,\mu) \leq P^{-r} r^\mu$, i.e. $w \geq g(r)$ for

$$g(x) \coloneqq x \log_2 P - \mu \log_2 x \qquad (x > 0).$$

Since $r \log_2 P = (r - r_c)\log_2 P + F$, it suffices to show $h(x) \geq 0$ for all real $x \geq r_c$, where

$$h(x) \coloneqq \left(1+\tfrac{1}{\theta}\right) g(x) - (x - r_c)\log_2 P,$$

because then $r \log_2 P \leq (1+\frac{1}{\theta})g(r) + F \leq (1+\frac{1}{\theta})w + F$. Set $A \coloneqq 2(1+\theta)\ln\mu$, so that $r_c = \mu A/\ln P$. Two facts about A are used below: $A = 2\ln\mu + 2\ln\mu^\theta \geq 4\ln 2$, since $\mu \geq 2$ and $\mu^\theta \geq 2$; and $A \geq 1+\theta$, since $2\ln\mu \geq 2\ln 2 > 1$.

h is non-decreasing on $[r_c, \infty)$. For $x \geq r_c$,

$$g'(x) = \log_2 P - \frac{\mu}{x \ln 2} \;\geq\; \log_2 P - \frac{\mu}{r_c \ln 2} \;=\; \log_2 P - \frac{\log_2 P}{A} \;=\; \left(1 - \frac{1}{A}\right)\log_2 P,$$

hence

$$h'(x) = \left(1+\tfrac{1}{\theta}\right) g'(x) - \log_2 P \;\geq\; \log_2 P \cdot \left(\left(1+\tfrac{1}{\theta}\right)\left(1-\tfrac{1}{A}\right) - 1\right) \;=\; \frac{\log_2 P}{\theta}\left(1 - \frac{1+\theta}{A}\right) \;\geq\; 0$$

by $A \geq 1 + \theta$.

$h(r_c) \geq 0$. We have $h(r_c) = (1 + \frac{1}{\theta})\, g(r_c)$ and $g(r_c) \ln 2 = r_c \ln P - \mu \ln r_c = \mu(A - \ln r_c)$, so it suffices to show $\ln r_c \leq A$. Since $P \geq 2$, $r_c \leq \mu A / \ln 2$, and since $\ln \mu \leq (1+\theta) \ln \mu = A/2$,

$$\ln r_c \;\leq\; \ln \mu + \ln A - \ln \ln 2 \;\leq\; \tfrac{A}{2} + \ln A - \ln \ln 2.$$

The function $f(A) \coloneqq \frac{A}{2} - \ln A + \ln \ln 2$ satisfies $f'(A) = \frac{1}{2} - \frac{1}{A} > 0$ for $A > 2$ and $f(4 \ln 2) = 2 \ln 2 - \ln 4 - \ln \ln 2 + \ln \ln 2 = 0$, so $f(A) \geq 0$ for all $A \geq 4 \ln 2$, which gives $\ln r_c \leq A$.

Together, $h(x) \geq h(r_c) \geq 0$ for all $x \geq r_c$, which completes Case 3 and the proof of Eq. (26).

The "in particular" clause. For $\mu \geq 2$, take $\theta \coloneqq \log_\mu 2$: then $\mu^\theta = 2$, so $4\varphi\mu^{2+\theta} = 8\varphi\mu^2$ and $1 + \frac{1}{\theta} = 1 + \log_2 \mu = \log_2(2\mu)$. For $\mu = 1$, the displayed condition reads $\log_2 \nu \geq 8\varphi + \lambda$, which implies $\log_2 \nu \geq \lambda$. □

Remark H.10 (Comparison with [BF23]). Setting $\mathcal{K} = \mathbb{Q}$ in Theorem H.9, so that $\varphi = 1$ and $\nu = |d|$, recovers the inverse bound of [BF23, Theorem 2 (ePrint)].

Combining Theorems H.1 and H.9 and Lemma H.7 we obtain the following Corollary H.11.

Corollary H.11. *Let $\mathcal{O}, \beta, \varphi, \mu, \nu, \mathbf{f}, d, \mathcal{C}$ satisfy the following properties:*

- *$\mathcal{O}$ is $\mathbb{Z}$, or the ring of integers of a cyclotomic field $\mathcal{K}$ of degree φ, or the ring of integers of an imaginary quadratic field $\mathcal{K} = \mathbb{Q}(\sqrt{\Delta})$ for a square-free integer $\Delta < 0$, and (η_1, η_2) are the corresponding constants of Lemma H.7, namely $(1,1)$, $(2^{2\varphi-1}, (3\varphi^{5/4})^\varphi)$ and $(8, 13\sqrt{|\Delta|})$, respectively.*
- *$\mathbf{f} = (f^{(1)}, \ldots, f^{(w)}) \in \mathcal{O}[\mathbf{X}]^w$ is a tuple of μ-variate multilinear polynomials.*
- *$d \in \mathcal{O}$ with $\mathsf{cont}(\mathbf{f}) + \langle d \rangle = \langle 1 \rangle$, where $\mathsf{cont}(\mathbf{f}) \coloneqq \sum_{c=1}^{w} \mathsf{cont}(f^{(c)})$ denotes the content of $\mathbf{f}$, i.e. the ideal generated by the coefficients of all the $f^{(c)}$, and $\nu \coloneqq |\mathcal{N}(d)|$.*
- *$\mathcal{C} \coloneqq \mathcal{C}_\beta \coloneqq \{x \in \mathcal{O} : \|x\| \leq \beta\}$ for some $\beta > 0$.*

For any target security parameter $\lambda > 0$, the Schwartz-Zippel soundness error is bounded by $\varepsilon_{\mathsf{sz}} \leq 2^{-\lambda}$ provided that

$$|\mathcal{C}_\beta| \;\geq\; 2\mu\, \eta_2\, \eta_1^{\mu-1}\, 2^\lambda \qquad \text{and} \qquad \log_2 \nu \;\geq\; 8\varphi\mu^2 + \log_2(2\mu)\big(\lambda + 1 + \mu \log_2 \eta_1\big).$$

Concretely, the second condition reads $\log_2 \nu \geq 8\mu^2 + \log_2(2\mu)(\lambda + 1)$ for $\mathcal{O} = \mathbb{Z}$, $\log_2 \nu \geq 8\varphi\mu^2 + \log_2(2\mu)\big(\lambda + 1 + (2\varphi - 1)\mu\big)$ for cyclotomic $\mathcal{K}$, and $\log_2 \nu \geq 16\mu^2 + \log_2(2\mu)(\lambda + 1 + 3\mu)$ for imaginary quadratic $\mathcal{K}$.

Proof. By Lemma H.7, the pair (η_1, η_2) satisfies Eq. (22) for every non-zero ideal of $\mathcal{O}$, so Theorem H.1 bounds $\varepsilon_{\mathsf{sz}}$ by $\eta_1^\mu \cdot \Pr[\sum_{k=1}^{\mu} \mathbf{Z}_k \geq \mathbf{r}] + \mu\, \eta_2\, \eta_1^{\mu-1} / |\mathcal{C}_\beta|$. The first condition bounds the second summand by $2^{-\lambda-1}$. For the first summand, we invoke the "in particular" clause of Theorem H.9 with $\lambda' \coloneqq \lambda + 1 + \mu \log_2 \eta_1$: the second condition reads $\log_2 \nu \geq 8\varphi\mu^2 + \log_2(2\mu)\lambda'$, so $\Pr[\sum_{k=1}^{\mu} \mathbf{Z}_k \geq \mathbf{r}] \leq 2^{-\lambda'}$ and the first summand is at most $\eta_1^\mu \cdot 2^{-\lambda'} = 2^{-\lambda-1}$. Summing the two bounds gives $\varepsilon_{\mathsf{sz}} \leq 2^{-\lambda}$. □

Remark H.12 (Comparison with [BF23]). Setting $\mathcal{O} = \mathbb{Z}$ in Theorem H.1, where $(\eta_1, \eta_2) = (1, 1)$ and $|\mathcal{C}_\beta| = 2\lfloor \beta \rfloor + 1$, gives the bound $\mu / |\mathcal{C}_\beta| + \prod_{j=1}^{\ell} I_{1/p_j}(r_j, \mu)$, which is the bound of [BF23, Theorem 1 (ePrint)] with $m = |\mathcal{C}_\beta|$; combined with Theorem H.9, this gives their Corollary 1, $\varepsilon_{\mathsf{sz}} \leq 2^{-\lambda} + \mu / |\mathcal{C}_\beta|$ for $\log_2 |d| \geq 8\mu^2 + \log_2(2\mu)\lambda$.

H.5 On the Ratio of Disproportionately Long Elements

In the above, we obtain an upper bound of the probability of a multivariate polynomial vanishing modulo d when evaluated at a uniformly random point of geometric norm at most β, where the upper bound shrinks as the field norm $\mathcal{N}(d)$ grows. The upper bound is not immediately useful in situations, e.g. for lattice Bulletproofs (Section G), where we can only upper bound the geometric norm $\|d\|$ instead of the field norm $\mathcal{N}(d)$ of the modulus, e.g. when $\mathcal{O}$ is a cyclotomic ring. In what follows, we assume that $\mathcal{O}$ is a cyclotomic ring of degree φ.

For any $e \in \mathcal{O}$, we call any $d \in \mathcal{O}$ satisfying $d = e \cdot u$ for some unit $u \in \mathcal{O}^\times$ an associate of e. Note that due to the multiplicativity of $\mathcal{N}(\cdot)$, all associates of e has the same field norm, i.e. $\mathcal{N}(d) = \mathcal{N}(e)$. Let

$e \in \mathcal{O}$ be such that its geometric norm is the smallest among its associates.[28] Intuitively, an associate d of e has disproportionately large geometric norm if $d = e \cdot u$ for some unit u with disproportionately large geometric norm (note that the field norm is $\mathcal{N}(u) = 1$).

In the following, we show that the ratio of elements with disproportionately large geometric norms is small. More specifically, we study the number of elements in the set

$$\{x \in \mathcal{O} : \mathcal{N}(x) \le \nu \ \wedge \ \gamma < \|x\| \le \delta\}$$

in relation to the number of elements in the set

$$\{x \in \mathcal{O} : \|x\| \le \beta\}$$

where $\beta \le \gamma \le \delta$ and ν is suitably small compared to β^φ, and show that the ratio shrinks rapidly as β grows.

Lemma H.13. *Let $\mathcal{O}$ be a cyclotomic ring of degree $\varphi \ge 4$. Let $0 \le \beta \le \gamma \le \delta$ and $\nu \in \mathbb{N}$. Define the sets:*

$$\begin{aligned} N_\nu &\coloneqq \{x \in \mathcal{O} : \mathcal{N}(x) \le \nu\}, \\ M_\delta &\coloneqq \{x \in \mathcal{O} : \|x\| \le \delta\}, \\ M_{\gamma,\delta} &\coloneqq M_\delta \setminus M_\gamma = \{x \in \mathcal{O} : \gamma < \|x\| \le \delta\}. \end{aligned}$$

Assume in addition that $\ln(\delta/\gamma) \ge 1$. Then

$$\frac{|N_\nu \cap M_{\gamma,\delta}|}{|M_\beta|} \le O\left(\frac{\nu\,(\ln\delta)^{\varphi/2-2}\ln(\delta/\gamma)}{\beta^\varphi}\right).$$

Proof. Let $x \in N_\nu$ with $x \ne 0$, and write $\mathfrak{a} = (x)$. Then $\mathcal{N}(\mathfrak{a}) = |\mathcal{N}(x)| \le \nu$, so $\mathfrak{a}$ is one of at most N_ν^* principal ideals of norm at most ν. Fixing a generator $e_\mathfrak{a}$ of $\mathfrak{a}$, every x with $(x) = \mathfrak{a}$ equals $e_\mathfrak{a} u$ for a unique unit $u \in \mathcal{O}^\times$, so

$$|N_\nu \cap M_{\gamma,\delta}| \le \sum_{\substack{\mathfrak{a} \text{ principal} \\ \mathcal{N}(\mathfrak{a}) \le \nu}} |\{u \in \mathcal{O}^\times : \gamma < \|e_\mathfrak{a} u\| \le \delta\}|.$$

Choose $e_\mathfrak{a}$ to be a near-shortest generator (via the log-unit lattice geometry of [CDPR16]), so that $\|e_\mathfrak{a}\| \le c_\varphi \mathcal{N}(\mathfrak{a})^{1/\varphi}$ for some $c_\varphi = e^{\widetilde{O}(\sqrt{\varphi})}$. As the geometric norm is submultiplicative up to a φ-dependent factor ($\|eu\| \le \|e\|\|u\|$ under the ℓ_∞-norm, and $\|eu\|_2 \le \sqrt{\varphi}\,\|e\|_2\|u\|_2$ under the ℓ_2-norm), there is a constant $C = C(\varphi)$ with

$$\gamma < \|e_\mathfrak{a} u\| \le \delta \quad \Longrightarrow \quad \frac{C^{-1}\gamma}{\|e_\mathfrak{a}\|} < \|u\| \le \frac{C\delta}{\|e_\mathfrak{a}\|}.$$

Write $U(t) \coloneqq |\{u \in \mathcal{O}^\times : \|u\| \le t\}|$. The number of admissible units is then at most $U(t_2) - U(t_1)$, with $t_2 \coloneqq C\delta/\|e_\mathfrak{a}\|$ and $t_1 \coloneqq C^{-1}\gamma/\|e_\mathfrak{a}\|$.

For cyclotomic fields of degree $\varphi \ge 4$ the unit group has rank $\varphi/2 - 1$, and [EL93] gives, as $t \to \infty$,

$$U(t) = C'''_\mathcal{K}(\ln t)^{\varphi/2-1} + O\Big((\ln t)^{\varphi/2-2}\Big), \qquad C'''_\mathcal{K} = \frac{\omega(\varphi/2)^{\varphi/2-1}}{\mathrm{R}_\mathcal{K}(\varphi/2-1)!},$$

where ω is the number of roots of unity and $\mathrm{R}_\mathcal{K}$ the regulator. Now $t_2/t_1 = C^2\delta/\gamma$ is independent of $\mathfrak{a}$, and $\ln t_2 = \ln(C\delta) - \ln\|e_\mathfrak{a}\| = O(\ln\delta)$ uniformly in $\mathfrak{a}$, since $1 \le \|e_\mathfrak{a}\| \le c_\varphi \hat{\nu}$. By the mean value theorem applied to $s \mapsto s^{\varphi/2-1}$,

$$(\ln t_2)^{\varphi/2-1} - (\ln t_1)^{\varphi/2-1} \le \left(\tfrac{\varphi}{2} - 1\right)(\ln t_2)^{\varphi/2-2}\ln(t_2/t_1) = O\Big((\ln\delta)^{\varphi/2-2}\ln(\delta/\gamma)\Big).$$

Using $\ln(\delta/\gamma) \ge 1$ to absorb the additive $O((\ln\delta)^{\varphi/2-2})$ error of U, we get $U(t_2) - U(t_1) = O((\ln\delta)^{\varphi/2-2}\ln(\delta/\gamma))$ uniformly in $\mathfrak{a}$. Summing over the at most N_ν^* principal ideals of norm at most ν, and using $N_\nu^* \le C'_\mathcal{K}\nu + C''_\mathcal{K}\nu^{1-2/\varphi} = O(\nu)$ from [Lee23, Theorem 1.1],

$$|N_\nu \cap M_{\gamma,\delta}| \le O\Big(\nu(\ln\delta)^{\varphi/2-2}\ln(\delta/\gamma)\Big).$$

Finally, by Lemma H.5, $|M_\beta| \ge \left(\frac{\pi\beta^2}{2\varphi}\right)^{\varphi/2} = \varphi^{-\Theta(\varphi)}\beta^\varphi$, and dividing gives the claim. □

[28] For concreteness, in [CDPR16] it is shown that $\|e\| \le \exp(\tilde{O}(\sqrt{\mathfrak{f}})) \cdot |\mathcal{N}(e)|^{1/\varphi}$.

Remark H.14. It is important to note that the above can be made far more explicit with the use of Davenport's [Dav51] result and would not amount to more than an exercise in multivariable calculus. This also avoids the determination of number field constants such as regulators, etc. That being said, for the exposition of the result and to avoid computing "unpleasant" volumes of $i-$th dimensional projections, which would not add much to readers' understanding, and are case- (number field) sensitive, we employed the slicker approach of hiding all the errors with the big O notations.

I Notions Related to Adaptive Special Soundness

In this section, we discuss the connections between adaptive special soundness and other related notions. We first look at almost special soundness [BF23], which was introduced to study certain folding protocols, one of which we also analyse using adaptive special soundness in Section G. A strict formal comparison would be tedious and inconclusive, as differences in syntax and settings of these works make the essentially incomparable. However, we can explain and relate the high-level concepts, and show which aspects adaptive special soundness covers. We then compare AdSS with predicate special soundness [AABKT24], the latter of which was introduced to study the proof system LaBRADOR [BS23]. We argue that AdSS provides the same expressiveness as predicate special soundness under mild conditions, but with a simpler interface. Finally, we also discuss other less directly related notions.

I.1 Almost Special Soundness

As noted in the introduction, almost special soundness [BF23] is formulated in a specific setting, with specific goals in mind. Namely, it is defined for certain types of protocols built from certain deterministic commitment schemes. Hence, both syntax and semantics are incomparable to adaptive special soundness and other special soundness notions. Setting that aside, the idea of almost special soundness is one way to formalise the intuition, that in certain protocols (in particular, when probabilistic tests are used), one first extracts a candidate witness (and some additional transcripts), such that either the candidate witness satisfies the relation or with sufficient probability can extract a witness to another "hard relation", e.g. a binding break. Almost special soundness considers this idea within specific constraints, such as the deterministic commitments, which are also designed such that the Fiat–Shamir transformation of such protocols can be proven secure based on a simple analysis (without using the Fiat–Shamir extractor of [AFK22]). A downside of this special-purpose approach is that typical proofs of knowledge do not naturally fit into the deterministic folding framework of [BF23]. (For example, [AABKT24] argues that it is not applicable to Labrador [BS23].) And that it is not inapplicable to protocols with non-deterministic provers.

On slightly more formal level, almost special soundness is parameterised by two predicatese ϕ_a, ϕ_b, where ϕ_a implies ϕ_b, and there are multiple extraction algorithms:

- An *extract witness* algorithm extracts a witness from a (labelled) tree (of commitment openings). It is guaranteed to succeed if all tree nodes satisfy ϕ_a.
- An *extract internal* algorithm deals with the case where the above extraction fails (since not all nodes satisfy ϕ_a). Given a subtree, this algorithm either outputs a binding break or satisfies a certain property w.r.t. ϕ_b.
- An *extend algorithm* essentially predicts an (honest) prover's message from commitments openings. This captures the idea that once a candidate witness is obtained, a malicious prover must either comply with it (and if the witness is bad, fail in probabilistic tests with high probability), or it deviates from it, but then a binding break will be extracted (by the *extract internal* algorithm). This algorithm is used in tandem with the promises of the extract internal algorithm, to derive the binding breaks.

We leave the comparison at this high level, and only add a few further remarks. Firstly, it seems plausible that in typical protocols, adaptive special soundness can replace almost special soundness, because it captures the idea behind almost special soundness; we consider Section G a clear indicator for this plausibility. Secondly, we note that the extractor construction of [BF23] builds on [BCCGP16], which uses a Markov bound for runtime truncation, which leads to suboptimal guarantees for the knowledge error. Thirdly, partly due to the specific setting, almost special soundness must be applied to a protocol en block and analysed as such; there is no (nested) composition result provided in [BF23].

I.2 Predicate Special Soundness

The notion of predicate special soundness [AABKT24] is more abstract than almost special soundness [BF23]. In the Σ-protocol (3-move) setting, it considers k-special soundness with three additional predicates ϕ^{chall}, ϕ^{prop}, ϕ^{bind}, where:

- $\phi^{\mathsf{chall}}_{\ell}(c_1, \ldots, c_\ell) \in \{0,1\}$ indicates if a challenge c_ℓ is useful, given previous challenges $c_1, \ldots, c_{\ell-1}$, for $\ell = 1, \ldots, k$. For example, $\phi^{\mathsf{chall}}_{\ell}(c_1, \ldots, c_\ell)$ is 1 if all challenges distinct (capturing k-special soundness), or linearly independent, or any other predicate.
- $\phi^{\mathsf{prop}}(\tau_1, \ldots, \tau_{k-1}) \in \{0,1\}$ indicates whether a property holds true for the $k-1$ extracted transcripts. For example, that an extracted SIS solution is sufficiently short, or that a probabilistic test was not fooled.
- $\phi^{\mathsf{bind}}((\tau_1, \ldots, \tau_{k-1}), \tau_k) \in \{0,1\}$ indicates whether a "binding" holds for the k extracted transcripts.

Notice that $\phi^{\mathsf{chall}}_{\ell}$ is defined for every $\ell = 1, \ldots, k$, but the predicates ϕ^{prop} (resp. ϕ^{bind}) are only defined for ($k-1$ resp. k) transcripts resp. as input. The reason for the latter is, that the k-th transcript is used to derive a binding break.

Again, predicate special soundness captures the idea that a candidate witness, derived from $k-1$ transcripts $\tau_1, \ldots, \tau_{k-1}$ may fail to satisfy a property, i.e. $\phi^{\mathsf{prop}}(\tau_1, \ldots, \tau_{k-1}) = 0$, but that in this case, we expect that another response τ_k of the malicious prover has to break the binding property with "high" probability, i.e. $\phi^{\mathsf{bind}}((\tau_1, \ldots, \tau_{k-1}), \tau_k) = 1$.

This idea is quantified through the so-called *failure densities* [AABKT24, Definition 5.2/6] of the predicate ϕ^{chall} and the pair $(\phi^{\mathsf{prop}}, \phi^{\mathsf{bind}})$, where essentially:

- The failure density $p^{\mathsf{chall}}_{\ell}$ of $\phi^{\mathsf{chall}}_{\ell}$ is

$$\max_{c_1, \ldots, c_{\ell-1}} \Pr_{c_\ell}[c_\ell \notin \{c_1, \ldots, c_{\ell-1}\} \mid \phi^{\mathsf{chall}}(c_1, \ldots, c_{\ell-1}, c_\ell) = 0]$$

  where $c_1, \ldots, c_{\ell-1}$ are distinct and $c_\ell \leftarrow_{\$} \mathfrak{C}$ is uniformly random. That is, this is the worst-case probability for a challenge being "bad" in round ℓ.
- The failure density p^{com} of ϕ^{chall} is

$$\max_{\tau_1, \ldots, \tau_{k-1}} \Pr_{c_k} \left[\begin{array}{c} \exists \tau_k \colon c_k = \mathsf{challof}(\tau_k) \\ \wedge \phi^{\mathsf{bind}}((\tau_1, \ldots, \tau_{k-1}), \tau_k) = 1 \\ \wedge \phi^{\mathsf{chall}}_{k}(c_1, \ldots, c_{k-1}, c_k) = 1 \end{array} \middle| \begin{array}{c} \forall i \in [0, k-1] \colon \phi^{\mathsf{chall}}_{i}(c_1, \ldots, c_i) = 1 \\ \wedge \phi^{\mathsf{prop}}((\tau_1, \ldots, \tau_{k-1}), \tau_k) = 0 \end{array} \right]$$

  where $c_k \leftarrow_{\$} \mathfrak{C}$ and $c_i = \mathsf{challof}(\tau_i)$ and τ_i are accepting transcripts with a shared first message. That is, this is the worst-case probability for the property predicate failing, while the binding predicate holds.

Lastly, predicate special soundness requires that:

- If all predicates hold, then the extractor must succeed to find a witness.
- If the binding predicate fails, then the extractor must succeed to find a "binding break" (i.e. witness to a different relation, called the *binding relation*).
- If the above cases do not occur, then one of the failure cases occurs. The core result [AABKT24, Theorem 5.1] asserts that the probability that an extracted tree falls into this case is bounded (roughly) by $p^{\mathsf{com}} + \sum_{i=1}^{k} p^{\mathsf{chall}}_{i}$.

We note here that, similar to almost special soundness, predicate special soundness was introduced and analysed for the Fiat–Shamir transformation of protocols and extended to a coordinate-wise special soundness notion as well, since that is required for analysing Labrador [BS23]. Compared to almost special soundness, here the extractors from [ACK21; AFK22] is used, which offer better parameters.

Let us now compare predicate special soundness with adaptive special soundness. Firstly, we have simplified the presentation of predicate special soundness significantly in one aspect: In [AABKT24], all predicates are defined w.r.t. to a (sub)tree of transcripts. Modularity of the notion is not considered in [AABKT24], and no explicit composition results are discussed. Hence, the predicates must be specified for the full protocol in one go, and analysed as such. In that aspect, our composition results simplify understanding and treatment.

Secondly, we provide an extractor which works well even for larger knowledge errors. The analysis for predicate special soundness does not handle this as well as adaptive special soundness. Indeed, it is not clear if predicate special soundness extractors can handle cut-and-choose protocols, such as Kilian's protocol (see Section 5), without increasing either runtime or knowledge error significantly (compared to adaptive special soundness). The problem is, that the extractor in [AABKT24] uses [AFK22] essentially black-box (which is optimised only for k-special soundness), and fails to extract if the k-tree of transcripts fails to satisfy all predicates, e.g. if too many transcripts have useless challenges. With larger knowledge error it becomes crucial to steer extraction towards useful challenges, because too many challenges are useless towards the end of the extraction process. Hence, predicate special soundness, or at least the extractor, seems limited to negligible knowledge error regimes.

Thirdly, and finally, let us consider a special case of predicate special soundness, which is effectively captured by special soundness. For this, we assume:

- *Monotonicity:* The challenge predicate family $\phi_\ell^{\mathsf{chall}}$ is monotonic.[29]
- *Efficiency:* The maps $\phi_i^{\mathsf{chall}}(c_1, \dots, c_i)$ and

$$\psi^{\mathsf{bind}}((\tau_1, \dots, \tau_{k-1}), c_k) = [\exists \tau_k \colon c_k = \mathsf{challof}(\tau_k) \wedge \phi^{\mathsf{bind}}((\tau_1, \dots, \tau_{k-1}), \tau_k)]$$

  are efficiently computable.

Note that both properties are very natural (and indeed satisfied in the application in [AABKT24]). In this case, there is a natural translation of the predicate system into a useful challenges structure. For this, let

- $\mathcal{U}^{\mathsf{chall}}(\tau_1, \dots, \tau_\ell) = \{c \in \mathfrak{C} \mid \phi_\ell^{\mathsf{chall}}(c_1, \dots, c_\ell) = 1\}$, where we write $c_i = \mathsf{challof}(\tau_i)$. (More precisely, we have a sequence $\mathcal{U}_j^{\mathsf{chall}}$, with one UCS per protocol round, and c_i is only the i-th round challenge.)
- $\mathcal{U}^{\mathsf{com}}(\tau_1, \dots, \tau_i) = \mathfrak{C}$ for $i < k - 1$ and

$$\mathcal{U}^{\mathsf{chall}}(\tau_1, \dots, \tau_{k-1}) = \{c \in \mathfrak{C} \mid \phi^{\mathsf{prop}}(\tau_1, \dots, \tau_{k-1}) = 0 \wedge \psi^{\mathsf{bind}}((\tau_1, \dots, \tau_{k-1}), c)\}$$

- $\mathcal{U}(\tau_1, \dots, \tau_i) = \mathcal{U}^{\mathsf{chall}}(\tau_1, \dots, \tau_i) \cup \mathcal{U}^{\mathsf{com}}(\tau_1, \dots, \tau_i)$ is the union of the UCSs.

By our monotonicity assumptions $\mathcal{U}^{\mathsf{chall}}$ and $\mathcal{U}^{\mathsf{com}}$ are UCSs, and it is straightforward to observe that the union of two UCSs is again a UCS, and the knowledge error satisfies

$$\kappa(\mathcal{U}) = \kappa(\mathcal{U}^{\mathsf{chall}} \cup \mathcal{U}^{\mathsf{com}}) \leq \kappa(\mathcal{U}^{\mathsf{chall}}) + \kappa(\mathcal{U}^{\mathsf{com}}).$$

By construction and assumptions, it holds that

$$\kappa(\mathcal{U}^{\mathsf{chall}}) \leq p_k^{\mathsf{chall}} \leq \sum_{i=1}^{k} p_k^{\mathsf{chall}}$$

and that

$$\kappa(\mathcal{U}^{\mathsf{com}}) \leq p^{\mathsf{com}}$$

Hence, under these mild restrictions, adaptive special soundness essentially encompasses predicate special soundness for 3-move protocols. By nested composition and the tree-structure of the predicates in [AABKT24], this extends beyond 3-move protocols through our nested composition of extractors. As a consequence, adaptive special soundness applies to Labrador [BS23].

Remark I.1 (Combining adaptive and predicate special soundness). Predicate special soundness and adaptive special soundness are in general incomparable in their expressive power, because predicate special soundness is a way to *keep track of failure events*, whereas adaptive special soundness is a way to *steer extraction*. Above, we showed that sometimes, we can convert "tracking" into "steering". In general, these two notions may be combined to get the best of both worlds: Steer extraction to improve knowledge error and runtime as much as possible, and track failure events to handle non-monotonic or inefficient predicates.

[29] By monotonic, we mean that if the predicate is satisfied for some input, then it will be satisfied given additional challenges (or transcripts) as input.

I.3 Other Notions Related to AdSS

We discuss other notions that are connected or related to AdSS but not necessarily generalised by the latter.

Fiat-Sharmir transform of special sound protocols. The knowledge soundness of protocols obtained via the Fiat–Shamir transformation for $(k_1, \ldots, k_n)$-special sound protocols was analysed in [AFK22; AFK23]. The result was later generalised to access-structure special sound protocols in [AFR23; AFKR26] and coordinate-wise special sound protocols in [FMN24]. While we expect that the approach in [AFKR26] carries over to AdSS, given the already expansive scope of this work, we leave analysing the knowledge soundness of Fiat–Shamir transformed AdSS protocols as an important future work.

Short-circuit extraction [HKR19; Klo23]. The notion of *short-circuit extraction* of $(k_1, \ldots, k_\mu)$-special soundness was defined with the aim of significantly reducing the extractor's runtime by observing that certain events lead to early success. While adaptive special soundness captures the idea of short-circuit extraction in a general setting, a runtime analysis which justifies the conjectural claims of [HKR19; Klo23] is left open.

(Probabilistic) (k, g)-special soundness [LMS22]. The work [LMS22] provides a knowledge extractor against quantum adversaries. For this, it introduces (probabilistic) (k, g)-special soundness, where g is a *consistency predicate* on a set of transcripts. Further, it considers the case where instead of the complete response z, only partial information $f(z)$ is available to the witness extractor. Then (k, g)-special soundness asserts that, given a g-consistent set of transcripts $\tau_i = (a, c_i, f(z_i))$, the witness extractor outputs a valid witness. While receiving $f(z)$ instead of z makes extraction harder, the choice of g can ensure nice properties on the set of transcripts, e.g. in [LMS22, Claim 5.14] it is used to ensure that commitment openings are consistent through all transcripts. Overall, the relation to AdSS is not quite clear and only asymptotic results are stated in [LMS22].