Adaptive Special Soundness: Improved Knowledge Extraction by Adaptive Useful Challenge Sampling

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

[1] CWI, Cryptology Group, Amsterdam, The Netherlands
[2] TNO, Applied Cryptography & Quantum Applications, The Hague, The Netherlands
thomas.attema@tno.nl
[3] Aalto University, Espoo, Finland
{michael.klooss,russell.lai}@aalto.fi
[4] Charles University, Faculty of Mathematics and Physics, Czech Republic
p.yatsyna@matfyz.cuni.cz

Abstract. Proving knowledge soundness of an interactive proof from scratch is often a challenging task. This has motivated the development of various special soundness frameworks which, in a nutshell, separate knowledge extractors into two parts: (1) an extractor to produce a set of accepting transcripts conforming to some structure; (2) a witness recovery algorithm to recover a witness from a set of transcripts with said structure. These frameworks take care of (1), so it suffices for a protocol designer to specify (2) which is often simple(r).

Recently, works by Bünz–Fisch (TCC'23) and Aardal et al. (CRYPTO'24) provide new frameworks, called almost special soundness and predicate special soundness, respectively. To handle insufficiencies of special soundness, they deviate from its spirit and augment it in different ways. The necessity for their changes is that special soundness does not allow the challenges for useful sets of transcripts to depend on the transcripts themselves, but only on the *challenges* in the transcripts. As a consequence, (generalised) special soundness cannot express extraction strategies which reduce a computational problem to finding "inconsistent" accepting transcripts, for example in PCP/IOP-based or lattice-based proof systems, and thus provide (very) sub-optimal extractors.

In this work, we introduce *adaptive special soundness* which captures extraction strategies exploiting inconsistencies between transcripts, e.g. transcripts containing different openings of the same commitment. Unlike (generalised) special soundness (Attema, Fehr, and Resch (TCC'23)), which specifies a target transcript structure, our framework allows specifying an extraction *strategy* which guides the extractor to sample challenges *adaptively* based on the history of prior *transcripts*. We extend the recent (almost optional) extractor of Attema, Fehr, Klooß and Resch (EPRINT 2023/1945) to our notion, and argue that it encompasses almost special soundness and predicate special soundness in many cases of interest. As a challenging application, we modularise and generalise the lattice Bulletproofs analysis by Bünz–Fisch (TCC'23) using the adaptive special soundness framework. Moreover, we extend their analysis to the ring setting for a slightly wider selection of rings than rational integers.

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 small probability. A stronger variant, known as knowledge soundness, postulates the existence of a knowledge extractor, which can *efficiently obtain a witness* from any convincing prover. Interactive proofs that are knowledge sound are called *proofs of knowledge* (PoKs). Such PoKs are important in cases where a witness trivially exists, e.g. for proving knowledge of a discrete logarithm or a short integer solution (SIS). Designing and analysing knowledge extractors is often challenging, even for simple and natural proof systems. A general research direction is therefore to design more expressive frameworks for analysing knowledge extractors.

As highly efficient proofs of knowledge are at the core of recent advances in succinct proof systems (or arguments), a recent line of works analyses, extends, and systematises knowledge soundness in this situation. These results include abstract frameworks, such as reductions of knowledge [KP23], and improved analyses, namely (conjecturally) faster extraction [HKR19; Klo23], optimally tight extraction [ACK21], parallel repetition [AF22], and security of the Fiat–Shamir transformation [AFK22; AFK23] for $(k_1, \dots, k_n)$-special soundness, which was generalised to access structure special soundness in [AFR23; AFKR23], and generalised to component-wise special soundness in [FMN23]. Moreover, the notion of *almost special soundness* has been introduced in [BF23] to obtain a meaningful security analysis for lattice-based folding protocols [BLNS20; AL21; ACK21] with exponential challenge space à la Bulletproofs [BCC+16; BBB+18]. A concurrent work [AAB+24b] introduces *predicate special soundness* and applies it to modularise (the Fiat–Shamir transformation) a generalisation of [BS23], another lattice-based proof system with exponential challenge space.

Besides special soundness, another general design principle is based on PCP and IOP-based protocols [BCS16]. These are first proven (unconditionally) secure, and then compiled into interactive proof systems. Often, the compilation uses idealised (random oracle) assumptions which allow so-called *straight-line* extraction of the PCP/IOP strings that were committed. Thus, this framework splits knowledge soundness into extractability of commitments and security of the IOP. Recently, in [AFR23] first efforts have been made to bridge the gap between the IOP and special soundness frameworks, and remove the need for idealised assumptions. Namely, they obtain an analysis of a specific IOP protocol in terms of generalised special soundness. The difficulty of analysing (compiled) IOP protocols through special soundness is related to a major limitation of special soundness notions: Their inability to (efficiently) capture probabilistic tests. In this work, we continue to bridge the gap between IOPs and special soundness by providing novel techniques.

1.1 Special Soundness and its Limitations

Special Soundness. The starting point of our work is knowledge soundness based on (tree-)special soundness. That is, the knowledge extractor consists of two parts: An algorithm Ext, called extractor, which finds a tree of accepting transcripts, given black-box access to a malicious prover. And an algorithm $\mathcal{W}$, which given such a tree outputs a witness. In the simplest case, namely k-special soundness for public-coin 3-move protocols (a.k.a. Σ-protocols), the *tree* is just a list of k transcripts with the property that all transcripts have the same first prover message, and *all challenges are distinct.* Generalisations to multiple rounds lift this to trees [BCC+16], whereas generalisations to access structures relax the distinctness requirement on challenges [AFR23].

Limitations of Special Soundness: Capturing Probabilistic Tests. The main motivation of this work is to obtain a notion of special soundness that can capture the natural setting, where properties of knowledge soundness and ordinary soundness are mixed. The goal is to obviate the need for ad-hoc, tailor-made approaches, which are currently used to handle this setting, while achieving best possible soundness guarantees. In particular, we aim to generalise and improve upon [BF23], which is a first formal treatment of this setting (in a limited special case). Later, in Sections 1.3 and 6.3, we compare to [BF23] and another concurrent work [AAB+24b] in detail.

An important feature of probabilistic tests is that they need not be extractable, or, that their soundness parameters are much better than their knowledge parameters. This leverage is useful to design better proof systems. Abstractly, the situation is the following:

A proof of knowledge is combined with a probabilistic test (or proof system) which is *only sound* but not knowledge sound. For concreteness, consider a proof of knowledge of a committed vector $\mathbf{x}$, combined with a probabilistic test/proof of shortness which simply consists of (one or more) random linear combination (RLC) $z = \sum_i r_i x_i$ with random short coefficients r_i, so that $|z|$ is big with sufficiently high probability if any $|x_i|$ is big, and short otherwise. Indeed, this is what happens in [CGKR22; BS23].

With this setup, the extractor runs into a difficulty: What should it do if the extracted $\mathbf{x}$ does not satisfy the required shortness bound? As the probabilistic test only provides *soundness for fixed* $\mathbf{x}$, but $\mathbf{x}$ depends on the adversary and extraction process, the soundness of the test does not immediately apply.

Fortunately, one property is typically self-evident: Let σ be the soundness error of the probabilistic test. Then, for a given bad $\mathbf{x}$, the adversary can fool the verifier with probability at most σ, so *unless the adversary equivocates commitment* to a different $\mathbf{x}$, the knowledge error should merely increase by σ. To show this, the extractor is typically modified as follows: If the bad case happens, i.e. if the probabilistic test was fooled for the extracted $\mathbf{x}$, try again, and abort if no valid $\mathbf{x}$ or binding break was found. This approach cannot be properly captured by (access-structure) special soundness as it can only impose a certain (access) structure on the *challenges* of accepting transcripts which are necessary for extraction, e.g. k distinct challenges. However, in our case, the *bad event depends on the candidate witness* $\mathbf{x}$ which was computed from transcripts. Thus, we must consider a generalisation of special soundness which allows dependencies on the accepting *transcripts*, not just their challenges.

1.2 Our Contributions

Our main technical contributions are two-fold.

Adaptive Special Soundness and Useful Challenge Structures We define *useful challenge structures (UCS)*, and a corresponding special soundness notion. This notion generalises prior special soundness notions for 3-move Σ-protocols. Intuitively, at any stage the extractor determines which challenges are currently useful, and then tries to find an accepting transcript for a useful challenge. In prior works, the set of useful challenges does not depend on the entire transcript, rather, it only depends on the verifier's *challenges* in the transcripts that have been found so far, e.g. requiring distinct challenges. We generalise the notion of "usefulness" by allowing the set of useful challenges to adaptively depend on the actual *transcripts* that have been found so far. We arrive at the following.

- We define the notion of *adaptive special soundness* and an *(almost) optimal extractor* for any (monotonic) useful challenge structure $\mathcal{U}$.
- We give an *almost optimal knowledge extractor* by extending the improved extractor analysis of generalised special soundness from [AFKR23] to adaptive special soundness (for interactive proof systems). In many cases, this yields an almost optimal extractor, i.e., measured in terms of the "knowledge error" $\kappa(\mathcal{U})$ and "depth" $\mathsf{depth}(\mathcal{U})$ of a UCS $\mathcal{U}$, we almost achieve the best-possible success and runtime bounds.[5]
- To demonstrate the usefulness of our framework, we sketch how to handle Kilian's succinct argument [Kil92] and lattice-based folding from [BF23] in our framework. We observe that our analysis of Kilian's argument is very natural and on par with the recent concurrent work [CDG+24a].
- We provide a *framework for generalised sequential composition* which is crucial for UCS-based extraction as needed in [BF23]. We expect our framework to be useful to capture other types of composition as well, e.g. component-wise special soundness.

Motivating Application: Soundness of Lattice-based Bulletproofs A motivating application for our notion of adaptive special soundness is to analyse the soundness of lattice-based Bulletproof protocols [BLNS20; AL21; ACK21;

[5] In particular, modelling generalised special soundness via adaptive special soundness recovers the exact same parameters. Moreover in many cases (such as generalised special soundness), one can exhibit a attacks which enforce a knowledge error of $\kappa(\mathcal{U})$ and worst-case $\mathsf{depth}(\mathcal{U})$ transcripts for extraction, thus providing an intuitive lower-bound on (expected) runtime for black-box extraction.

CLM23; BF23] for the Inhomogeneous Short Integer Solution (ISIS) relation. Bünz and Fisch [BF23] were the first to analyse this setting with (unstructured) exponential size challenge sets using their notion of "almost special soundness", while all other prior works used structured polynomial size challenge sets. However, their analysis was limited to integer lattices, and an analysis of ring lattice instantiations was left as an important open problem. We contribute the following:

- Building on our new framework, we modularise (and generalise) the soundness analysis of [BF23] by using $\mathcal{U}$-adaptive special soundness.
- We make a first step towards generalising [BF23] to cover the number ring $\mathcal{R}$ setting. In particular, we generalise their "multilinear composite Schwartz–Zippel lemma" to rings and make an initial effort in generalising the so-called "inverse bound" [BF23] of the Schwartz–Zippel lemma in the ring setting.

Handling general number rings $\mathcal{R}$ turns out to be significantly more complex than $\mathcal{R} = \mathbb{Z}$. A major obstacle is that *geometric* and *algebraic* norms of elements do not coincide (unlike in $\mathbb{Z}$) or even scale proportionally, due to the existence of possibly infinitely many units. Another barrier is that fractions of ring elements do not necessarily admit coprime representations, since factorisation is generally not unique, and even reduced representations may be hard to compute. As such, our analysis is unconditional only for the ring of rational integers and the rings of integers of imaginary quadratic fields with class number 1, i.e. where unique factorisation holds.

1.3 Related Work

Our work is in the plain/CRS models, so we focus on sufficiently closely related works. We omit most ROM-based works, as their techniques (e.g., straightline extraction or Fiat–Shamir transformation) are mostly orthogonal.

Special Soundness Notions As we define a new notion for special soundness, our work is strongly related to such prior and concurrent generalizations of k-special soundness. Some special soundness notions are discussed as examples in more detail in the main body, see Sections 6 and 8.

(Γ-)Special Soundness Extractors [ACK21; AF22; Att23; AFR23; AFKR23]. Our extractor generalises the extractor from [AFKR23] from Γ-special soundness [AFR23] to adaptive special soundness. In terms of knowledge error and expressiveness, our results encompass those for Γ-special soundness [AFKR23] for interactive proof systems.

Almost Special Soundness [BF23]. Our notion of adaptive special soundness is inspired by *almost special soundness* of [BF23] and as an abstraction of the many ad-hoc techniques for analysing probabilistic tests within proof systems. Unfortunately, since [BF23] is tailored to certain folding-like protocols in the random oracle model, a proper comparison with adaptive special soundness seems essentially impossible.[6] Nevertheless, our motivating application, lattice-based

[6] The setting of [BF23] is very specific. In Section 6.3, we discuss this in more detail.

folding, clearly follows the same outline as [BF23]. In particular, our proof involves generalising a Schwartz–Zippel-type lemma given in [BF23] to the ring setting.

Predicate Special Soundness [AAB+24a]. The concurrent work [AAB+24a] 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 analyzing probabilistic tests, and [AAB+24a] uses it specifically to analyze (Fiat–Shamir transformed) Labrador [BS23]. Under mild restrictions on the predicates (roughly, efficiency and monotonicity), predicate special soundness is a special case of adaptive special soundness (see Section 6.3) and thus benefits from our improved knowledge extractors and composition results.

Component-wise special soundness (CWSS) [FMN23]. The notion of *component-wise* special soundness provides a framework for extraction techniques that were used implicitly in prior works, e.g. [BBC+18]. Our generalised sequential composition result captures this partially: It yields minimally better knowledge error, but the run-time becomes exponential in the number of coordinates (see Section 8.2).

Short-circuit extraction [HKR19; Klo23]. The notion of *short-circuit extraction* of $(k_1, \dots, k_\mu)$-special soundness was defined with the aim of significantly reducing the extractor's run-time 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 run-time 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. The witness extractor only needs to succeed for g-consistent sets of k transcripts and is given *only challenges and responses*, but not the full transcripts. To the best of our knowledge, this notion is tailored to the quantum setting. Classically, there is no evident benefit over standard k-special soundness.

1.4 Other work

We briefly discuss other work which is closely related our knowledge extractor, setting, or application.

PCPs and IOPs. While adaptive special soundness is not tailored to PCP or IOP analysis, we show in Section 2.3 how Kilian's succinct argument $\mathsf{Kilian}[\mathsf{PCP}, \mathsf{VC}]$ is handled in our setting in a straightforward manner, something prior works were incapable of.[7] Concurrently, Chiesa, Dall'Agnol, Guan, and Spooner [CDGS23]

[7] [BF23; AAB+24a] can potentially be applied, but are not good fits. They are tailored to k-special soundness which leads to large run-time or knowledge error.

provide an analysis of $\mathsf{Kilian}[\mathsf{PCP}, \mathsf{VC}]$, and IOPs in general, in a concrete security setting (which simplifies run-time analysis). This is was published in a follow-up [CDG+24a] focusing on Kilian's argument. Our notion is well-suited to analyse Kilian's argument (Section 2.3), but an analysis of multi-round IOPs remains open, because, similar to CWSS, the bounds on run-time growth of compositions of extractors become superpolynomial (as in [AFR23]).

Lattice-based Proof Systems. A large body of works (e.g. [BDL+18; YAZ+19; BLS19; ESLL19; ALS20; BLNS20; ENS20; LNS21; AL21; ACK21; LNP22; CLM23; BF23]) has been devoted to studying the task of proving knowledge of a (possibly committed) vector $\mathbf{x}$ satisfying relations of the form

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

or more generally bounded-norm satisfiability of polynomial equations over $\mathcal{R}_q$. In particular, the Bulletproofs protocol [BCC+16; BBB+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.

An annoying issue unique to the lattice setting is the presence a soundness gap – often one can only construct knowledge extractors which extract an approximate witness $\mathbf{x}^*$ which is an approximation in two ways: (1) The extracted witness only satisfies $\mathbf{A}\mathbf{x} = \mathbf{y}s \bmod q$ for some short element s, and (2) the norm of $\mathbf{x}$ is only bounded by some $\alpha' \gg \alpha$. Using the terminology in [AL21], the element s is called the *slack* and the ratio α'/α is called the *stretch*. Although it has been demonstrated [AL21] that the slack s can be eliminated, i.e. $s = 1$, in certain choices of the ring $\mathcal{R}$, it is achieved at the cost of using a challenge set which is of size at most the smallest ideal norm in $\mathcal{R}$, which is at most the degree of $\mathcal{R}$ for cyclotomic rings of prime-power conductor. As discussed earlier, [BF23] was the first to circumvent this limitation in the $\mathbb{Z}$ setting by analysing the use of large challenge sets via the lens of almost special soundness. Part of this work aims to generalise this strategy to the $\mathcal{R}$ setting via the more general notion of adaptive special soundness.

1.5 Roadmap

In Section 1, we provide a general introduction and discuss our contribution. A more technical overview is given in Section 2. In Section 3 we provide general preliminaries and auxiliary lemmata. Section 4 gives definitions around useful challenge structures. Eventually, Section 5 introduces our extractor together with its analysis. In Section 6, we provide examples for useful challenge structure and extended comparisons to some prior special soundness notions. We recommend he reader to skip ahead and read parts of Section 6 after the definition of UCS.

Our framework for extraction is completed by Sections 7 and 9, where the former introduces sequential composition in the special case of UCS extractors, and the latter generalizes it to arbitrary extractors. In between is Section 8 which contains examples to illustrate how our more general sequential composition notion for extractors can be applied.

In Section 10, we introduce and analyze the lattice-based folding analogue of Bulletproofs in the ring setting. The generalised multilinear composite Schwartz–Zippel lemma modulo ideals is presented in Section 11.

2 Technical Overview

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 (a, c, z) or $\perp$. We write $\mathsf{challof}(\tau)$ for the challenge of a transcript τ.

Useful Challenge Structures (UCS) Roughly, a *useful challenge structure (UCS)* is a function $\mathcal{U}$, which inputs a sequence of transcripts $(\tau_1, \dots \tau_i)$ and outputs a subset $\mathcal{U}(\tau_1, \dots \tau_i)$ of $\mathfrak{C}$; we call challenges $c \in \mathcal{U}(\tau_1, \dots \tau_i)$ *useful* (according to $\mathcal{U}$). That is, $\mathcal{U}$ abstracts which challenges narrow in on a set of transcripts we can extract from. We require that $\mathcal{U}(\tau_1, \dots, \tau_i) \subseteq \mathcal{U}(\tau_1, \dots, \tau_{i-1})$, i.e. the set of useful challenges only shrinks when more history/information is available. Moreover, we call a sequence $(\tau_1, \dots, \tau_\ell)$ a $\mathcal{U}$-chain, if $\mathsf{challof}(\tau_i) \in \mathcal{U}(\tau_1, \dots, \tau_{i-1})$ holds for all $i = 2, \dots, \ell$. Transcripts sequences generated by our rewinding-based extractor will always be $\mathcal{U}$-chains. Intuitively, once $\mathcal{U}(\tau_1, \dots, \tau_i) = \emptyset$, “extraction” succeeded, hence we call such a $\mathcal{U}$-chain *complete*.

We associate two values with a UCS $\mathcal{U}$: The *depth* $\mathsf{depth}(\mathcal{U})$ is the length of the longest complete $\mathcal{U}$-chain. The *knowledge error* $\kappa(\mathcal{U})$ of $\mathcal{U}$ is $1 - \min_T \frac{|\mathcal{U}(T)|}{|\mathfrak{C}|} = \min_T \frac{|\mathfrak{C} \setminus \mathcal{U}(T)|}{|\mathfrak{C}|}$, where T is an incomplete $\mathcal{U}$-chain, i.e. $\mathcal{U}(T) \neq \emptyset$. In other words, $\kappa(\mathcal{U})$ is the relative size of the *largest useless challenge set* $\mathfrak{C} \setminus \mathcal{U}(T)$.

Adaptive Special Soundness. Let $\mathcal{R}$ be an NP-relation and $\mathcal{U} = (\mathcal{U}_{\mathsf{stmt}})_{\mathsf{stmt}}$ be a uniform family of UCSs, parametrised by the statement stmt, We define $\mathcal{U}$-*adaptive special soundness* of a 3-move[8] public-coin protocol Π as follows: There exists a uniform family of witness extraction algorithms $(\mathcal{W}_{\mathsf{stmt}})_{\mathsf{stmt}}$, which on input stmt and a complete $\mathcal{U}_{\mathsf{stmt}}$-chain $T = (\tau_1, \dots, \tau_\ell)$ of accepting Π-transcripts outputs witness wit for stmt.

Simple Examples. We can capture k-special soundness over challenge set $\mathfrak{C}$ with a UCS $\mathcal{U}$, by setting $\mathcal{U}(\tau_1, \dots, \tau_\ell) = \mathfrak{C} \setminus \{\mathsf{challof}(\tau_i)\}_{i=1}^{\ell}$. 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 6 for more examples.

[8] Adaptive special soundness can be applied to multi-round protocols. For simplicity, we restrict to the most common case in the introduction, namely 3-move protocols.

The (Almost Optimal) Extractor We generalise the extractor from [AFKR23], which in turn builds on [AF22; AFR23], so that it handles useful challenge structures instead of access structures.

All technical details are in Section 5.3. Here, we only sketch our final result:

Theorem 2.1 (Informal). *A $\mathcal{U}$-adaptive special sound 3-move interactive protocol Π is knowledge sound with knowledge error $\max_{\mathsf{stmt}} \kappa(\mathcal{U}_{\mathsf{stmt}})$. More precisely, there exists a knowledge extractor* Ext *that, given oracle access to an adversary $\mathcal{A}$ attacking Π on input* stmt*,*

1. *spends at most $\frac{\mathsf{depth}(\mathcal{U}_{\mathsf{stmt}})}{1-\kappa(\mathcal{U}_{\mathsf{stmt}})} \cdot \mathbb{E}[\mathsf{time}(\mathcal{A}(C)) \mid C \leftarrow_R \mathfrak{C}]$ steps executing $\mathcal{A}$,*
2. *succeeds in extracting a witness with probability at least $\frac{\varepsilon-\kappa(\mathcal{U}_{\mathsf{stmt}})}{1-\kappa(\mathcal{U}_{\mathsf{stmt}})}$, where ε denotes the success probability of $\mathcal{A}$.*

We call our extractor *almost optimal*, since for the running time, it roughly $\frac{\mathsf{depth}(\mathcal{U}_{\mathsf{stmt}})}{1-\kappa(\mathcal{U}_{\mathsf{stmt}})}$ times the expected runtime of the adversary, and must recover $\mathsf{depth}(\mathcal{U}_{\mathsf{stmt}})$ transcript in the worst case, hence run the adversary that often. Hence, it is only a factor $\frac{1}{1-\kappa(\mathcal{U}_{\mathsf{stmt}})}$ larger than optimal. For the knowledge error, our extractor recovers the optimal error as [ACK21; AFR23] for k-special soundness and Γ-special soundness [AFR23], which is known to be optimal by (trivial) attack strategies. The optimality for a useful challenge structure $\mathcal{U}$ is less clear, because the associated knowledge error $\kappa(\mathcal{U})$ can be *larger* than the actual knowledge error of the extractor, see Section 9. Hence $\kappa(\mathcal{U})$ is not always the "correct" measure for optimality.

2.2 Template for Using UCS in Applications

A typical setting where adaptive special soundness applies is the following template, for which we sketch the ad-hoc extraction approach and how a UCS could be used:

- A protocol uses a PoK for some witness wit, and proves for probabilistic test φ that $\varphi_{\mathsf{wit}}(c) = 1$, asserting property $\mathsf{pred}(\mathsf{wit}) = 1$ on the witness (w.h.p.).
- Ad-hoc: Extract a candidate witness wit' from the PoK. If $\mathsf{pred}(\mathsf{wit}') = 0$, extract again, finding wit''. If φ is a σ-sound test, then only with probability $\varepsilon - \sigma$ can $\mathsf{wit}' = \mathsf{wit}''$ occur. And if $\mathsf{wit}' \neq \mathsf{wit}''$, binding is broken.
- UCS: Restricting useful challenges to $\{c \mid \varphi_{\mathsf{wit}'}(c) = 0\}$ *ensures* a break.

Recent examples of the above are [CGKR22; BS23; BC24], which mix a PoK with a proof of (rational) shortness. The prior work [BF23] and concurrent work [AAB+24a] provide frameworks based on the ad-hoc approach. Our UCS framework deviates from the ad-hoc approach, but still captures these settings in a systematic and simple way. Next, we consider a concrete example.

2.3 Kilian's Succinct Argument

A surprisingly simple application of adaptive special soundness is Kilian's succinct argument [Kil92].

The Protocol In the following, we apply analyse Kilian's succinct interactive argument [Kil92] when used with computationally binding succinct vector commitments, such as Merkle trees. We treat Kilian's succinct argument as a family of 3-move protocols[9] for some NP-relation Ξ; we write $(\mathsf{stmt}, \mathsf{wit}) \in \Xi$ for statement and witness, respectively. Let PCP be a q-query, length N PCP with randomness complexity r over some alphabet Σ, where q, N, r may depend on stmt. Let VC be a (succinct) vector commitment, e.g. a Merkle tree commitment. Then the protocol $(\mathcal{P}, \mathcal{V}) = \mathsf{Kilian}[\mathsf{PCP}, \mathsf{VC}]$ 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 $\rho \leftarrow_R \{0,1\}^r$ 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, write $I(\rho) = \mathsf{PCP.Query}(\mathsf{stmt}; \rho)$. Moreover, write $\pi_I = \pi_{i \in I}$ and let $\mathsf{PCP.Vfy}(\pi_I, I)$ reject if π_I contains $\perp$.

Computational Soundness As a warm-up, let us prove *computational soundness* of $\mathsf{Kilian}[\mathsf{PCP}, \mathsf{VC}]$. For this, we assume the PCP has soundness error σ, that is, if stmt is not in the NP-language, then *any* PCP string will make the verifier accept with probability at most σ. We repurpose our knowledge extractor as a (expected) polynomial time reduction: Suppose an adversary $\mathcal{A}$ convinces with probability ε. If $\varepsilon \leq \sigma$, there is nothing to prove, so we may assume $\varepsilon > \sigma$. The only thing we can do is to reduce to the binding property of the vector commitment scheme. The only way to do this is by searching for *inconsistent openings of vc*. By soundness, we know that to succeed with probability $\varepsilon > \sigma$, the adversary *must* equivocate commitments for *some* challenges. (Because PCP soundness is unconditional, and thus, if no equivocation ever occurs, success of $\mathcal{A}$ is bounded by σ, a contradiction.) We now define a UCS $\mathcal{U}$.

First, we note that Theorem 2.1 requires that challenges are chosen *uniformly at random* in $\mathfrak{C}$. This forces us to formally define $\mathcal{U}$ over $\mathfrak{C} = \{0,1\}^r$, since $I(\rho)$ may not be uniform.[10] The idea and definition of $\mathcal{U}(\tau_1, \dots, \tau_\ell)$ is now as follows: Let $T = (\tau_1, \dots \tau_\ell)$ be a sequence of transcripts. Write $\widetilde{\pi}(T) \in (\Sigma \cup \{\perp\})^N$ for the partial PCP string obtained from T, where $\widetilde{\pi}_i(T) = \perp$ if and only if index i was not opened in any transcript in T. We observe two cases:

[9] In the original protocol, the sender has to provide a commitment key. Our analysis formally extends to this setting by considering it as a part of the statement.

[10] This "transformation" (namely, using ρ instead of $I(\rho)$ as "challenge") shows how our results apply to any efficiently sampleable challenge distribution.

- Case 1: Some index i in was opened to *differing* values in transcripts of T. In this case, $\mathcal{U}(T) = \emptyset$: We already broke binding of VC. No more challenges are useful.
- Case 2: All openings are consistent. Thus, $\widetilde{\pi}(T)$ is uniquely defined. What are useful challenges now? Clearly, if $\mathsf{PCP.Vfy}(\widetilde{\pi}_{I(\rho)}(T), I(\rho)) = 1$ for challenge $I(\rho)$, then it is not useful — we already know (how to construct) an accepting transcript for $I(\rho)$. However, if $\mathsf{PCP.Vfy}(\widetilde{\pi}_{I(\rho)}(T), I(\rho)) = 0$, then we learn new information. Thus, we define ρ to be useful, i.e. we define $\mathcal{U}(T) = \{\rho \mid \mathsf{PCP.Vfy}(\widetilde{\pi}_{I(\rho)}(T), I(\rho)) = 0\}$.

Observe that any useful challenge brings us closer to our goal: Either $\widetilde{\pi}_{I(\rho)}(T)$ is rejected in $\mathsf{PCP.Vfy}$ due to a $\perp$ entry. In that case, a transcript for $I(\rho)$ opens a new index. Or all indices of $\widetilde{\pi}_{I(\rho)}(T)$ are known, but PCP verification reject the substring. In that case, an *accepting* transcript *must* open *vc* to something different than $\widetilde{\pi}_{I(\rho)}(T)$, i.e. break the binding property.

The UCS $\mathcal{U}$ is monotonic: Let $T_i = (\tau_1, \dots, \tau_i)$ and $T_{i+1} = (T_i, \tau_i)$ be $\mathcal{U}$-chains. We must show $\mathcal{U}(T_{i+1}) \subseteq \mathcal{U}(T_i)$. Fortunately, this is a trivial property: If $\mathcal{U}(T_{i+1}) = \emptyset$, there is nothing to do. And otherwise, τ_{i+1} reveals one or more new indices, which reduces the number of $\perp$s in $\widetilde{\pi}(\,\cdot\,)$, and thus, only reduces the choices $\rho \in \mathfrak{C}$ for which PCP verification rejects, i.e. fewer challenges are useful. This shows that $\mathcal{U}$ is monotonic. Moreover, we have

- $\mathsf{depth}(\mathcal{U}) \leq N - q + 2$: Clearly, if we reach $N - q + 1$ transcripts, all indices are known. Thus, after at most $N - q + 2$ transcripts, we must break binding of VC, and hence $\mathcal{U}(T) = \emptyset$.
- $\kappa(\mathcal{U}) \leq \sigma$: A challenge $I(\rho)$ is useful if an index of $I(\rho)$ was unopened or if $\widetilde{\pi}_{I(\rho)}(T)$ would be rejected by PCP verification. Note that the latter condition includes the former (since a $\perp$ entry causes rejection). By the soundness property, we find that at least a $(1 - \sigma)$-fraction of all $\rho \in \mathfrak{C}$ yield $I(\rho)$ are rejected. In other words, at least a $(1 - \sigma)$-fraction of $\mathfrak{C} = \{0, 1\}^r$ is useful. By definition, the knowledge error is thus at most σ.

Now, Theorem 2.1 concludes the proof: We have an *expected* (polynomial) time reduction to VC-binding, which succeeds with probability $\frac{\varepsilon - \sigma}{1 - \sigma}$. It can be made strict (polynomial) time through standard truncation arguments.

Knowledge Soundness To prove knowledge soundness, we require that PCP is *strongly γ-extractable*, a notion inspired by [Val08].[11] Namely, there exists an efficient deterministic witness recovery algorithm W, such that if $\mathsf{W}(\widetilde{\pi}) \notin \Xi_{\mathsf{stmt}}$, i.e. $\mathsf{W}(\widetilde{\pi}) = \perp$, then $\Pr_\rho[\mathsf{PCP.Vfy}(\widetilde{\pi}_{I(\rho)}, I(\rho)) = 1] \leq \gamma$. Perhaps (un)surprisingly, our definition of $\mathcal{U}$ remains almost unchanged, except that $\mathcal{U}(T)$ also checks if $\mathsf{W}(\widetilde{\pi}) \neq \perp$, in which case $\mathsf{W}(\widetilde{\pi}) \in \Xi_{\mathsf{stmt}}$, and hence $\mathcal{U}(T) = \emptyset$. Clearly, $\mathcal{U}$ remains monotonic, as we only provided short-cuts which result in $\mathcal{U}(T) = \emptyset$; unless this happens, the useful challenges are exactly the same as before, and

[11] [Val08] defines γ-extractable as: $\Pr[\mathsf{PCP.Vfy}(\pi, I) = 1] \geq 1 - \gamma$ implies $\mathsf{W}(\pi) \in \Xi_{\mathsf{stmt}}$.

we already showed that these only shrink. Consequently $\mathcal{U}$ is monotonic and $\mathsf{depth}(\mathcal{U}) \leq N - q + 2$ holds as before.

It remains to analyse the knowledge error. So suppose T is a $\mathcal{U}$-chain and $\mathcal{U}(T) \neq \bot$. Certainly $\mathsf{W}(\widetilde{\pi}(T)) = \bot$, as otherwise $\mathcal{U}(T) = \emptyset$. Since PCP is strongly γ-extractable, this implies that $\Pr_\rho[\mathsf{PCP}.\mathsf{Vfy}(\widetilde{\pi}_{I(\rho)}(T), I(\rho)) = 1] \leq \gamma$, for uniform $\rho \in \mathfrak{C}$, hence $\Pr_\rho[\mathsf{PCP}.\mathsf{Vfy}(\widetilde{\pi}_{I(\rho)}(T), I(\rho)) = 0] \geq 1 - \gamma$. In other words, at least a $(1-\gamma)$-fraction of $\rho \in \mathfrak{C}$ is useful. Thus, we have $\kappa(\mathcal{U}) \leq \gamma$ as claimed.

We sketch the our proven security claims below, but omit a full formalisation of the properties, in particular of VC, as this is not the focus of this work.

Corollary 2.2 (Informal). *Let* PCP *be a PCP for language* $\mathcal{L}$ *(resp. NP-relation* $\mathcal{R}$*) which is* σ*-sound (resp.* γ*-extractable), then* $(\mathcal{P}, \mathcal{V}) = \mathsf{Kilian}[\mathsf{PCP}, \mathsf{VC}]$ *for every adversary* $\mathcal{P}^*$ *against (knowledge) soundness, there is an adversary* $\mathcal{A}$ *against* VC*, such that for any* $\mathsf{stmt} \notin \mathcal{L}$

$$\Pr[\langle \mathcal{P}^*(1^\lambda, \mathsf{stmt}), \mathcal{V}(1^\lambda, \mathsf{stmt})\rangle] \leq \sigma + \mathsf{AdvBind}^{\mathsf{VC}}_{\mathcal{A}}(\lambda)$$

resp. there is an extractor for relaxed relation $\mathcal{R}'$*, where* $(\mathsf{stmt}, \mathsf{wit}) \in \mathcal{R}'$ *if either* $(\mathsf{stmt}, \mathsf{wit}) \in \mathcal{R}$ *or* wit *contains a binding break for* VC*, such that*[12]

$$\Pr[\langle \mathcal{P}^*(1^\lambda, \mathsf{stmt}), \mathcal{V}(\mathsf{stmt})\rangle] \leq \gamma.$$

Remark 2.3. Our notion of *strongly* γ*-extractable* PCP is related to a concurrent work by Chiesa, Dall'Agnol, Guan, and Spooner [CDGS23]. They formulate knowledge soundness of PCP in a natural game-based way, which generalises easily to IOPs. Fortunately, their notion is essentially equivalent to ours, except that they allow a *probabilistic* witness recovery algorithm W. To work with UCS, we want deterministic success, i.e. $\widetilde{\pi} \mapsto [\mathsf{W}(\widetilde{\pi}) \neq \bot]$ should implement a function.

Remark 2.4. A very recent concurrent work [CDG+24b; CDG+24a], studies the security of Kilian's protocol using a concrete security framework. Our above analysis recovers their bounds for soundness and knowledge error, if we truncate the expected time extractors suitably. In fact, our analysis slightly improves upon the expected time analysis of [CDG+24a] as we do not need an "error parameter" ϵ for success and also have no factor $\log(\mathsf{q}/\epsilon)$ in the runtime, where $\mathsf{q} = |I(\rho)|$ is the (constant) query complexity.

2.4 Generalised Sequential Composition of Extractor

Up 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 something stronger. This is needed to fully exploit the power of UCS to choose challenges which *ensure* a certain

[12] There is no advantage against VC, since binding breaks are witnesses for $\mathcal{R}'$.

property, e.g., $\phi_{\mathsf{wit}'}(c) = 0$ as used in the template for UCS and with Kilian's argument. In a multi-round setting, $\phi_{\mathsf{wit}'}$ may depend not only on challenge c_i of round i, but on *all* challenges. So for μ-fold composition, we must ensure that $\varphi_{\mathsf{wit}'}(c_1, \ldots, c_\mu) = 0$. This requires a generalisation of the standard sequential composition theorem (where extractors are completely unaware of other the existence of other challenges and rounds). Indeed, for our Bulletproofs analysis, this is a crucial ingredient. In Section 9, we provide such a generalisation and analyse its knowledge error and run-time. Under mild conditions, these are (almost) identical to standard sequential composition.

At a very high level, our generalisation considers an abstract adversary $\mathcal{A}\colon \mathfrak{C} \to \{0,1\}^*$, and requires that extractors are abstract adversaries (over $\mathfrak{C}$) as well. Hence, $\mathsf{Ext}^{\mathcal{A}}(c)$ takes as input a challenge $c \in \mathfrak{C}$ (and satisfies some mild properties).[13] This unification of adversary and extractor "trivializes" composition, and is already sufficient for our purposes. As not all extractors are naturally defined over $\mathfrak{C}$, we introduce simple tools to mimic standard sequential composition. We refer to Sections 7 and 9 for definitions and Section 8 for examples.

2.5 Revisiting Lattice-based Bulletproofs

The lattice-based Bulletproofs protocol [BLNS20; AL21; ACK21; BF23] is the lattice-analogue of the Bulletproofs protocol [BCC+16; BBB+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 $\mathcal{R}$.

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

1. At each round $i \in [\mu]$:
   (a) The prover computes two univariate degree-1 polynomials in $\mathcal{R}[C]$,

   $$f_{\mathbf{A}}(C) \coloneqq \mathbf{A}_0 \cdot C + \mathbf{A}_1 \bmod q \qquad \text{and} \qquad g_{\mathbf{x}}(C) \coloneqq \mathbf{x}_0 + \mathbf{x}_1 \cdot C$$

   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 $c \leftarrow_R \mathfrak{C} \subseteq \mathcal{R}$ to the prover, where the challenge set $\mathfrak{C}$ is a parameter of the protocol.

[13] Recall that standard sequential composition assumes $\mathfrak{C} = \mathfrak{C}_1 \times \ldots \times \mathfrak{C}_\mu$, and extractor $\mathsf{Ext}_i^{\mathcal{A}_i}$ expects $\mathcal{A}_i\colon \mathfrak{C}_i \to \{0,1\}^*$. Moreover, Ext_i takes no input (or only $(c_1, \ldots, c_{i-1})$). Generalised sequential composition assumes no structure on $\mathfrak{C}$, as each Ext_i is over $\mathfrak{C}$.

(c) Prover and verifier update the statement to $(\mathbf{A}', \mathbf{y}') := (f_{\mathbf{A}}(c), h(c))$. The prover also updates its witness to $\mathbf{x}' := g_{\mathbf{x}}(c)$.
(d) Continue to the next round with new statement $(\mathbf{A}', \mathbf{y}')$ and witness $\mathbf{x}'$.

2. After μ rounds, the witness $\mathbf{x}$ becomes a single value, and the matrix $\mathbf{A}$ has only one column. 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.[14] 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.

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.
2. Second, the AMSS analysis of the lattice-based Bulletproofs protocol was only done for the instantiation over $\mathcal{R} = \mathbb{Z}$, while essentially all practical lattice-based schemes are instantiated over some (typically cyclotomic) number rings $\mathcal{R}$. 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 and UCS framework. This serves as an example

[14] Over a cyclotomic ring with prime-power conductor, $s = 1$ is achievable [AL21].

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 $\mathcal{R}$. Concretely, we cover $\mathcal{R} = \mathbb{Z}$ 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 $\mathcal{R}$, and the quotient ring $\mathcal{R}_q \coloneqq \mathcal{R}/q\mathcal{R}$. Statements $(\mathbf{A}, \mathbf{y})$ are defined over $\mathcal{R}_q$, and witnesses $\mathbf{x}$ are defined over $\mathcal{R}$. 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 $\mathcal{R}$ and with the challenge set $\mathfrak{C}$ being the set of all $\mathcal{R}$ elements of norm at most some β, i.e. $\mathfrak{C} \coloneqq \{x \in \mathcal{R} : \|x\| \leq \beta\}$. 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. 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 $\mathbf{x} \in \mathcal{R}$ 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:

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

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:

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

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 $(\tau_v, \mathfrak{T}_v)$. It is convenient to view $(\tau_v, \mathfrak{T}_v)$ as a subtree of 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 $\mathcal{U}_i(\tau_v, \mathfrak{T}_v)$ 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. $\mathcal{U}_i(\tau_v, \mathfrak{T}_v)$ 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 $g_{\mathbf{x}_v^*/s_v^*}(\mathbf{c}) \notin \mathcal{R}$.

(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 $g_{\mathbf{x}_v^*/s_v^*}(\mathbf{c}) \notin \mathcal{R}$, 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 $\mathcal{R}$ 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.

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 $\mathbf{x} \in \mathbb{Z}^n$, we denote its infinity norm by $\|\mathbf{x}\| = \max_{i \in [n]} |x_i|$.

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

We often use "efficient algorithms" to refer to algorithms which run in probabilistic polynomial time in its input length. A relation $\varXi : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$ is a deterministic function which maps a statement-witness tuple $(\mathsf{stmt}, \mathsf{wit})$ to a bit. We write $\mathsf{wit} \in \varXi(\mathsf{stmt})$ if $\varXi(\mathsf{stmt}, \mathsf{wit}) = 1$.

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. The w-prefix subtree of $\mathfrak{T}$, denoted $\mathsf{prefix}(\mathfrak{T}, w)$, is the subtree obtained by removing node w and all nodes which come after w in pre-order tree traversal. Equivalently, $\mathsf{prefix}(\mathfrak{T}, w)$ is the subtree of $\mathfrak{T}$ formed by removing all subtrees $\mathfrak{T}_{w'}$ for all w' ranging from w and nodes on the right-hand-side of w of depth $\mathsf{depth}_\mathfrak{T}(w)$.

3.1 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 3.1. *Let $X \sim \mathsf{Geo}(p)$ and $Y \sim \mathsf{Geo}(q)$ be two geometrically distributed random variables. Then*

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

Lemma 3.2. *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 run-time analysis in Sections 5 and 9. It is a generalisation of a similar lemma in [AFKR23].

Lemma 3.3 (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. □

4 Adaptive Special Soundness

In this section, we define the notion of *adaptive special soundness.* This notion is even more general than the generalised special soundness notion introduced in [AFR23]. In the subsequent sections, we will show that adaptive special soundness implies knowledge soundness. First we introduce our notion of abstract adversaries and extractor, and show how it maps to standard settings such as Σ-protocols. Then we define adaptive special soundness for "single-stage" extraction, which handles Σ-protocol as detailed in Section 5. Multi-round protocols requires composition of extractors, or "multi-stage" extraction, and is defined and handled in Sections 7 and 9.

The following convention suppresses the inputs of extractors and adversaries to only those which are essential to our analysis.

Remark 4.1 (Convention). We will generally assume that inputs other than transcripts, such as **statement** and **auxiliary input** are ***hard-coded*** into adversaries, extractors, etc. When required, we consider *uniform* families parametrised by these inputs. For example, $(\mathcal{U}_{\mathsf{stmt}})_{\mathsf{stmt}}$ can be seen as a single Turing machine $\mathcal{U}$ taking stmt as part of its input.

4.1 Abstract Adversaries and Extractors

In the following, $\mathfrak{C}$ is an arbitrary challenge set with uniform distribution, and we write $\mathcal{T} = \{0,1\}^*$ to denote the space of transcripts. To simplify notation and handling of extractors, we make following definition.

Definition 4.2 (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 sequential 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 4.3 (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 $\tau = (a, c, z)$. 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.

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 9. Now, we turn to *abstract extractors.*

Definition 4.4 ((Ordinary) Abstract extractor). *An **(abstract) extractor** Ext over challenge set $\mathfrak{C}$ is a (probabilistic) oracle algorithm $\mathsf{Ext}^{\mathcal{A}} \colon \mathfrak{C} \to$*

$\{0,1\}^* \cup \{\bot\}$, *which takes any oracle* $\mathcal{A}\colon \mathfrak{C} \to \{0,1\}^* \cup \{\bot\}$. *We require that an abstract extractor is* ***ordinary****: If* $\mathsf{Ext}^{\mathcal{A}}(c)$ *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* $\mathsf{Ext}^{\mathcal{A}}(c)$ *immediately outputs* $\bot$.

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

*Remark 4.5 (*challof *for extractors).* Note that an extractor $\mathsf{Ext}^{\mathcal{A}}$ is itself an adversary over $\mathfrak{C}$, i.e. it has the same interface. By the requirements of the challof function, for any initial challenge $c \in \mathfrak{C}$, if $T \leftarrow \mathsf{Ext}^{\mathcal{A}}(c)$ is not $\bot$, then $\mathsf{challof}_{\mathsf{Ext}^{\mathcal{A}}}(T) = c$.

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 sequential 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 $\mathsf{V}^{\mathsf{post}}(\mathsf{wit}) = [(\mathsf{stmt}, \mathsf{wit}) \in \Xi]$. (Recall that we hard-code stmt in extractors and hence $(\mathsf{V}^{\mathsf{pre}}, \mathsf{V}^{\mathsf{post}})$.)

4.2 Useful Challenge Structures (UCS) and Adaptive Special Soundness for Σ-Protocols

We formalise some technical concepts and crucially the abstract notion of "useful" challenge structures (UCS).

Definition 4.6 (Sequence of Transcripts). *Denote by* $\mathcal{T} = \{0,1\}^*$ *the set of transcripts, and by* $\mathcal{T}^*$ *the set of finite sequences of transcripts, i.e. elements* $T \in \mathcal{T}^*$ *are sequences* $T = (\tau_1, \dots, \tau_k)$*, for some* k*, of transcripts* $\tau_i \in \mathcal{T}$*. Further, we write* $\mathit{pred}(T)$ *for the predecessor of* T*, i.e.,*

$$\mathit{pred}((\tau_1, \dots, \tau_k)) \coloneqq (\tau_1, \dots, \tau_{k-1}) \quad \text{and} \quad \mathit{pred}(\emptyset) = \emptyset.$$

Definition 4.7 (Challenge Set Function). *Let* $\mathfrak{C}$ *be a challenge set. Let* $\mathcal{F}\colon \mathcal{T}^* \to 2^{\mathfrak{C}}$ *be a function which takes as input a* sequence T of transcripts*, i.e.* $T = (\tau_1, \dots, \tau_\ell)$ *(for some* ℓ*), and outputs a subset* $\mathcal{F}(T) \subseteq \mathfrak{C}$ *of challenges. We call* $\mathcal{F}$ *a* ***challenge set function*** *over* $\mathfrak{C}$ *if it is*

- ***efficient****, i.e. the membership relation* $c \in \mathcal{F}(T)$ *can be efficiently decided;*
- ***efficiently sampleable****, i.e. sampling uniformly at random from* $\mathcal{F}(T)$*, for any given* T*, takes expected polynomial time (in the size* $|T|$ *of the input* T*).*

Definition 4.8 (Chains of Transcripts). *Let* $\mathcal{F}$ *be a challenge set function. A sequence of transcripts* $(\tau_1, \dots, \tau_\ell) \in \mathcal{T}^*$ *is a* $\mathcal{F}$***-chain*** *if* $\mathsf{challof}(\tau_i) \in \mathcal{F}(\tau_1, \dots, \tau_{i-1})$ *for all* $i \in \{1, \dots, \ell\}$*. We write*

- $\mathsf{Chains}(\mathcal{F})$ *for the set of all chains of* $\mathcal{F}$*;*
- $\mathsf{Chains}^{\circ}(\mathcal{F})$ *for the subset of chains* $((\tau_1, \dots, \tau_i))$ *with* $\mathcal{F}(\tau_1, \dots, \tau_i) \neq \emptyset$*.*

In our setting, $\mathcal{F}(T) = \emptyset$ indicates that a witness can be extracted from the chain of transcripts T. Hence, $\mathsf{Chains}^{\circ}(\mathcal{F}) \subseteq \mathsf{Chains}(\mathcal{F})$ contains only the chains that are incomplete, i.e. from these chains a witness cannot be extracted. It is often convenient to consider only $\mathsf{Chains}^{\circ}(\mathcal{F})$, instead of $\mathsf{Chains}(\mathcal{F})$, e.g. when considering the *next* transcript that is to be considered by the extractor.

Definition 4.9 (Depth of a Challenge Set Function). *Let $\mathcal{F}$ be a challenge set function. The **depth** of $\mathcal{F}$ is $\mathsf{depth}(\mathcal{F}) = \max\{i \mid (\tau_1, \dots, \tau_i) \in \mathsf{Chains}(\mathcal{F})\}$, i.e. the length of the longest chain in $\mathsf{Chains}(\mathcal{F})$.*

Definition 4.10 (Useful Challenge Structure (UCS)). *A challenge set function $\mathcal{U}$ over a challenge set $\mathfrak{C}$ is called a **useful challenge structure (UCS)** if it is **monotonic**, i.e. if $\mathcal{U}(T) \subseteq \mathcal{U}\big(\mathit{pred}(T)\big)$ for all $T \in \mathsf{Chains}(\mathcal{U})$, and $\mathcal{U}(\emptyset) = \mathfrak{C}$. Further, we define $\mathcal{C}(T) := \mathcal{U}(\mathit{pred}(T))$ for all $T \in \mathcal{T}^*$.*

We note that the "usefulness" of challenges is *defined* by the UCS, i.e. we use the adjective "useful" in an abstract definitional sense. Of course, for all UCSs of interest, the usefulness of challenges is indeed motivated by the needs of some extraction procedure. Indeed, looking ahead, the definition of UCS specifies "useful" challenges and hence extraction strategies abstractly. A UCS effectively encapsulates an extraction strategy: Our knowledge extractor will iteratively try to find new accepting transcripts until it can extract a witness. Given a sequence T of transcripts that have already been found, the set $\mathcal{U}(T)$ defines the challenges that are useful to the extractor, i.e. these challenges will bring the extractor closer to its goals of extracting a witness. For concreteness, we suggest the reader to skip ahead to Section 6 for examples of UCSs for common extraction scenarios.

We note that both depth and monotonicity are defined over the $\mathcal{U}$-chains only, i.e. how $\mathcal{U}$ is defined outside of $\mathsf{Chains}(\mathcal{U})$ is irrelevant. This is sufficient, since during extraction, only $\mathcal{U}$-chains will occur.[15]

We are now ready to define adaptive special soundness for Σ-protocols.

Definition 4.11 ($\mathcal{U}$-Adaptive Special Soundness – Σ-Protocols). *Let $\mathfrak{C}$ be a challenge set and $\mathcal{U} = (\mathcal{U}_{\mathsf{stmt}})_{\mathsf{stmt}}$ be a UCS over $\mathfrak{C}$. Let $(\mathcal{P}, \mathcal{V})$ be a Σ-protocol with challenge set $\mathfrak{C}$. We say that $(\mathcal{P}, \mathcal{V})$ is **$\mathcal{U}$-adaptive special sound** for a relation Ξ, if there exists an efficient witness extractor $\mathcal{W} = (\mathcal{W}_{\mathsf{stmt}})_{\mathsf{stmt}}$, such that the following holds: For every* stmt *and every $\mathcal{U}_{\mathsf{stmt}}$-chain of* accepting *transcripts $T = \big((a, c_1, z_1), \dots, (a, c_k, z_k)\big) \in \mathsf{Chains}(\mathcal{U}_{\mathsf{stmt}})$ with $\mathcal{U}_{\mathsf{stmt}}(T) = \emptyset$, the witness extractor $\mathcal{W}_{\mathsf{stmt}}(T)$ outputs a witness $\mathsf{wit} \in \Xi(\mathsf{stmt})$.*

4.3 Nested UCS-Chains and Adaptive Special Soundness for Multi-Round Protocols

In what follows, we generalise the above notions, such as UCS and $\mathcal{U}$-chains, to the setting of multi-round setting. The notions defined here are quite abstract, and the reader may skip to this section, and come back to it later. This section

[15] A good way to handle "invalid input" T is by setting $\mathcal{U}(T) = \emptyset$.

is first required for sequential composition of extractors, which is sketched in Section 7, and treated more precisely in Section 9 with several examples provided in Section 8.2. We recommend to cross-reference definitions with the latter examples, which illustrate how the notions behave and how they can be used.

Definition 4.12 (Nested Sequence of Transcripts (cf. Definition 4.6)). *Let $\mu \in \mathbb{N}$. A depth-μ nested sequence of transcripts $T = (T_1, T_2, \ldots, T_h)$ (for some h) is a sequence where each term T_i is a depth-$(\mu - 1)$ nested sequence of transcripts. A depth-1 nested sequence of transcripts $T = (\tau_1, \ldots, \tau_k) \in \mathcal{T}$ (for some k) is a sequence of transcripts.*

Remark 4.13 (Interpreting Nested Sequences as Trees). A nested sequence of transcripts can be parsed to have a tree representation. For example, consider a depth-2 nested sequence $T = (T_1, T_2) = ((\tau_1, \tau_2), (\tau_3, \tau_4))$. This corresponds to a tree of (edge-)depth 2 where the four leaves are labelled with $\tau_1, \ldots, \tau_4$ respectively. We define the **height** of T as the height viewed as a tree.

Nested sequences of transcripts arise naturally when composing *extractors.*[16] We emphasise that the notion of nested sequence of transcripts *does not necessarily reflect* protocol executions, i.e. the tree representation of a nested sequence of transcripts may or may not have the structure of a typical “tree of transcripts” constructed in the sense of protocol execution. To distinguish between the two notions, we call the former *extraction trees* and the latter *protocol (execution) trees.* More concretely, a protocol tree of transcripts of a $(2\mu+1)$-move interactive argument is a tree of (edge-)depth μ, where each root-to-leaf path represents a transcript, with the i-th prover message given by the i-th node label (counting from 0), and the i-th verifier message given by the i edge label (counting from 1). In contrast, the messages in a transcript stored at a leaf in a depth-μ extraction tree may or may not be assigned to the intermediate nodes and edges consistently in the same fashion as in a protocol execution tree. Indeed, a tree parsed from a nested sequence of transcripts has less structure: Only the leaves are labelled (by complete transcripts). This allows us to capture more extraction procedure as tree-special soundness. See Section 8 for clarifying examples. We remark, however, that there are many (if not more) examples where a protocol tree can be parsed as an extraction tree and vice versa, e.g. in Section 10.

Definition 4.14 ($(\mathcal{F}_1, \ldots, \mathcal{F}_\mu)$-Chains of Transcripts). *Let $\mu \in \mathbb{N}$ and $\mathcal{F} = (\mathcal{F}_1, \ldots, \mathcal{F}_\mu)$ be a sequence of challenge set functions. For $\mu = 1$, a (nested)* $\mathcal{F}_1$-chain of transcripts *is an $\mathcal{F}_1$-chain of transcripts (Definition 4.8). Let $T = (T_1, \ldots, T_\ell) \in \mathcal{T}^*$ be a depth-μ nested sequence of transcripts with T_i being the i-th depth-$(\mu - 1)$ subtree. The tree T is said to be a (nested)* ***$\mathcal{F}$-chain*** *if*

- *$(T_1, \ldots, T_\ell) \in \mathsf{Chains}(\mathcal{F}_1)$, and*
- *each T_i is a (nested) $(\mathcal{F}_2, \ldots, \mathcal{F}_\mu)$-chain of transcripts.*

[16] They also arise naturally when composing protocols, but here we are taking an extractor-centric perspective.

We write $\mathsf{Chains}(\mathcal{F})$ *for the set of all (nested)* $\mathcal{F}$*-chains. We say* T *is* ***complete*** *if* $\mathcal{F}(T_1, \ldots, T_\ell) = \emptyset$ *and each* T_i *is complete (w.r.t.* $(\mathcal{F}_2, \ldots, \mathcal{F}_\mu)$*.) Finally, let* $\mathsf{V}\colon \mathcal{T} \to \{0,1\}$ *be a predicate. We call a* $(\mathcal{F}_1, \ldots, \mathcal{F}_\mu)$*-chain* T V***-respecting****, if all transcripts in* T *satisfy* V*.*

We note that in the above definition, when instantiated with useful challenges sets for extractor composition, then each $\mathcal{F}_i$ expects as input depth-$(\mu - i)$ nested sequences of transcripts. While formally $\mathcal{F}_i$ must be defined for any input, we will make use of V_i-respecting (trees of) transcripts (for suitable V_i), to ensure a proper tree structure.

We are now ready to extend adaptive special soundness to nested UCS (and thus multi-round protocols and more). Note again, that we do not assume a round-structure on the protocol, and thus do not require any (product) structure on the challenge set $\mathfrak{C}$ in the definition below.

Definition 4.15 ($\mathcal{U}$-Adaptive Special Soundness). *Let* $\mu \in \mathbb{N}$*,* $\mathfrak{C}$ *be a challenge set, and* $\mathcal{U} = (\mathcal{U}_{\mathsf{stmt}})_{\mathsf{stmt}} = ((\mathcal{U}_{\mathsf{stmt},i})_{i=1}^{\mu})_{\mathsf{stmt}}$ *be sequence of* μ *useful challenge structures over* $\mathfrak{C}$*. A public-coin interactive argument* $(\mathcal{P}, \mathcal{V})$ *with challenge set* $\mathfrak{C}$ *is* $\mathcal{U}$*-adaptive special sound for a relation* Ξ *if the following are satisfied: There exists an efficient witness extractor* $\mathcal{W} = (\mathcal{W}_{\mathsf{stmt}})_{\mathsf{stmt}}$*, such that for all* stmt*, on input a complete depth-*μ *nested sequence* T *of* accepting *transcripts with* $T \in \mathsf{Chains}(\mathcal{U}_{\mathsf{stmt}})$*, the witness extractor* $\mathcal{W}_{\mathsf{stmt}}(T)$ *outputs a witness* $\mathsf{wit} \in \Xi(\mathsf{stmt})$*.*

4.4 Adaptive Witness Extractors and Adaptive Special Soundness

We conclude the section by introducing the notion of adaptive witness extractors (AWE) and that of adaptive special soundness with respect to an AWE. These concrete notions endow the abstract notions considered above (adversary, extractor, UCS) with semantic meanings. Moreover, they provide a more convenient template for specification and analysis of the extraction process.

First, we define adaptive witness extractors which take as input (non-nested) sequences of transcripts. These extractors are natural, for example, for Σ-protocols.

Definition 4.16 (Adaptive Witness Extractors). *A (uniform family of) efficient algorithm(s)* $\mathcal{W} = (\mathcal{W}_{\mathsf{stmt}})_{\mathsf{stmt}}$ *is said to be an adaptive witness extractor for a relation* Ξ *and challenge set* $\mathfrak{C}$ *if it satisfies the following requirements:*

Syntax. *For any* $\mathsf{stmt} \in \{0,1\}^*$ *and on input any sequence of transcripts* $T \in \mathcal{T}^*$*,* $\mathcal{W}_{\mathsf{stmt}}(T)$ *outputs a tuple* $(\mathfrak{C}', \mathsf{wit})$ *conforming to one of the following:*

- *(Extraction Incomplete.)* $\mathfrak{C}' \neq \emptyset$ *(and* $\mathsf{wit} = \bot$ *without loss of generality).*
- *(Extraction Success.)* $\mathfrak{C}' = \emptyset$ *and* $\mathsf{wit} \in \Xi(\mathsf{stmt})$*.*
- *(Extraction Failure.)* $\mathfrak{C}' = \emptyset$ *and* $\mathsf{wit} = \bot$*.*

By abuse of notation, we write $\mathcal{W}_{\mathsf{stmt}}(T) \in \Xi(\mathsf{stmt})$ *and* $c \leftarrow \mathcal{W}_{\mathsf{stmt}}(T)$ *where it is clear from the context whether the output tuple corresponds to a witness or a challenge subset.*

Induced UCS. *We require that $\mathcal{W}_{\mathsf{stmt}}$ restricted to its first output, denoted by $\mathcal{U}_{\mathcal{W},\mathsf{stmt}}$, is a challenge set function (Definition 4.7), i.e. $\mathfrak{C}'$ is an efficiently decidable and efficiently sampleable subset of $\mathfrak{C}$. Furthermore, we require that $\mathcal{U} = \mathcal{U}_{\mathcal{W},\mathsf{stmt}}$ is useful (Definition 4.10), i.e. $\mathcal{U}(T) \subseteq \mathcal{U}(\mathsf{pred}(T))$ for all $T \in \mathsf{Chains}(\mathcal{U})$ and $\mathcal{U}(\emptyset) = \mathfrak{C}$. We call $\mathcal{U}_{\mathcal{W},\mathsf{stmt}}$ the UCS induced by $\mathcal{W}_{\mathsf{stmt}}$.*

Analogous to the notion of nested UCS, we introduce the notion of nested AWE. These extractors are natural, for example, for multi-round protocols, e.g. Bulletproofs.

Definition 4.17 (Nested Adaptive Witness Extractors). *Let $\mu \in \mathbb{N}$. A sequence of efficient algorithms $\mathcal{W} = (\mathcal{W}_{\mathsf{stmt}})_{\mathsf{stmt}} = ((\mathcal{W}_{\mathsf{stmt},i})_{i=1}^{\mu})_{\mathsf{stmt}}$ is said to be a depth-μ adaptive witness extractor for a relation Ξ and challenge set $\mathfrak{C}$ if there exists a sequence of relations $(\Xi_i)_{i=1}^{\mu}$ with $\Xi = \Xi_1$ such that, for each $i \in [\mu]$, $(\mathcal{W}_{\mathsf{stmt},i})_{\mathsf{stmt}}$ is an adaptive witness extractor for relation Ξ_i and challenge set $\mathfrak{C}$. We write $\mathcal{U}_{\mathcal{W},\mathsf{stmt}} \coloneqq (\mathcal{U}_{\mathcal{W}_{\mathsf{stmt},i}})_{i=1}^{\mu}$ for the sequence of UCSs induced by $\mathcal{W}$ and stmt and $\mathcal{U}_{\mathcal{W},\mathsf{stmt}} \coloneqq (\mathcal{U}_{\mathcal{W},\mathsf{stmt}})_{\mathsf{stmt}}$.*

Finally, we define the notion of adaptive special soundness with respect to an AWE instead of a UCS. This notion allows us to discuss the adaptive special soundness of a protocol in a more simplistic language, since we can avoid constructing a (sequence of) UCS(s) which is based on a (sequence of) witness extractor(s), and then argue that the witness extractor(s) must successfully produce a witness given a sequence of transcripts conforming to the UCS.

Definition 4.18 ($\mathcal{W}$-Adaptive Special Soundness). *Let $\mu \in \mathbb{N}$, $\mathfrak{C}$ be a challenge set, and $\mathcal{W} = (\mathcal{W}_{\mathsf{stmt}})_{\mathsf{stmt}} = ((\mathcal{W}_{\mathsf{stmt},i})_{i=1}^{\mu})_{\mathsf{stmt}}$ be a family of (sequences of) adaptive witness extractors over $\mathfrak{C}$. A public-coin interactive argument $(\mathcal{P}, \mathcal{V})$ is $\mathcal{W}$-adaptive special sound, if $(\mathcal{P}, \mathcal{V})$ is $\mathcal{U}_{\mathcal{W},\mathsf{stmt}}$-adaptive special sound with witness extraction algorithm $\mathcal{W}_{\mathsf{stmt},1}$.*

According to Definitions 4.15 and 4.18, to prove that public-coin interactive argument $(\mathcal{P}, \mathcal{V})$ is adaptive special sound, it suffices to construct a (sequence of) adaptive witness extractor $\mathcal{W} = (\mathcal{W}_{\mathsf{stmt}})_{\mathsf{stmt}}$ which, on input a complete depth-μ nested sequence T of *accepting* transcripts with $T \in \mathsf{Chains}(\mathcal{U}_{\mathcal{W}_{\mathsf{stmt}}})$, outputs a witness $\mathsf{wit} \in \Xi(\mathsf{stmt})$. Note that $T \in \mathsf{Chains}(\mathcal{U}_{\mathcal{W}_{\mathsf{stmt}}})$ means that the transcripts in T are produced according to the specifications of $\mathcal{W}_{\mathsf{stmt}}$.

5 Knowledge Extractors for Useful Challenge Structures

In this section, we define extend the knowledge extractor from [AFKR23] to useful challenge structure, and hence adaptive special-sound Σ-protocols, and analyse its properties, thereby proving knowledge soundness. To simplify the exposition, we first introduce some notation and conventions.

5.1 Notation and Conventions

Suppose $(\mathcal{P}, \mathcal{V})$ is a $\mathcal{U}$-adaptive special sound Σ-protocol for relation Ξ. 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 $w \in \Xi(\mathsf{stmt})$. 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).

As in prior works [AF22; AFK22; AFK23; AFR23; AFKR23], we present our core results in a more abstract language, allowing us to later handle more complicated scenarios such as multi-round interactive proofs. Namely, we considers an abstract adversary $\mathcal{A} \colon \mathfrak{C} \to \mathcal{T}^* \cup \{\bot\}$, whose natural instantiation is given by a prover $\mathcal{P}^*$ attacking the Σ-protocol $(\mathcal{P}, \mathcal{V})$ on a statement stmt. In contrast to prior works, verification is implicitly handled by the adversary $\mathcal{A}$, i.e. for any challenge $c \in \mathfrak{C}$, $\mathcal{A}(c)$ either outputs an *accepting* transcript $\tau = (a, c, z) \in \mathcal{T}^* = \{0,1\}^* \times \mathcal{C} \times \{0,1\}^*$ or it fails and outputs $\bot$. For this reason, we do not need to specify a separate verification predicate. In the formal terms of Section 4.1, we assume throughout that $\mathcal{A}$ is V-respecting, where V is the verification function of $\mathcal{V}$.

In the following, we define shorthand notations for recurring expressions. To this end, let $U \subseteq C \subseteq \mathfrak{C}$. Then, we define

$$\varepsilon(U \mid C) \coloneqq \Pr[\mathcal{A}(c) \neq \bot \land c \in U \mid c \leftarrow_R C]\,. \tag{1}$$

This is the probability that a challenge c, sampled uniformly at random from C, lies in $U \subseteq C$ and the algorithm $\mathcal{A}$ succeeds on c. In particular, it holds that $\varepsilon(C \mid C) = \Pr[\mathcal{A}(c) \neq \bot \mid c \leftarrow_R C]$. Further, we define

$$\Delta(U \mid C) \coloneqq \Pr[\mathcal{A}(c) \neq \bot \land c \notin U \mid c \leftarrow_R C]\,. \tag{2}$$

This is the probability that c is sampled *outside* the subset U and the adversary succeeds for c. By basic probability theory, it holds that

$$\varepsilon(U \mid C) = \varepsilon(C \mid C) - \Delta(U \mid C)\,,$$

i.e. $\Delta(U \mid C)$ is the exact (absolute) difference between $\varepsilon(U \mid C)$ and $\varepsilon(C \mid C)$.

Next, we define

$$\kappa(U \mid C) \coloneqq \Pr[c \notin U \mid c \leftarrow_R C]\,. \tag{3}$$

It is easily seen that $\kappa(U \mid C)$ is an upper bound for the (absolute) difference $\Delta(U \mid C)$ between $\varepsilon(U \mid C)$ and $\varepsilon(C \mid C)$. Moreover, unlike ϵ and Δ, κ is independent of the adversary $\mathcal{A}$. For these reasons, $\kappa(U \mid C)$ can also be viewed as a relative knowledge error between the sets U and C. We define the **knowledge error** of a UCS $\mathcal{U}$ as

$$\kappa(\mathcal{U}) \coloneqq \max_{S \in \mathsf{Chains}^\circ(\mathcal{U})} \kappa(\mathcal{U}(S) \mid \mathfrak{C})\,. \tag{4}$$

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

The equality

$$\varepsilon(U \mid U) = \frac{\varepsilon(C \mid C) - \Delta(U \mid C)}{1 - \kappa(U \mid C)} \tag{5}$$

holds by definition. Finally, we define the following quantity

$$\delta(U \mid C) := \frac{\varepsilon(C \mid C) - \Delta(U \mid C)}{1 - \Delta(U \mid C)} = \varepsilon(U \mid U) \cdot \frac{1 - \kappa(U \mid C)}{1 - \Delta(U \mid C)}\,. \tag{6}$$

The inequality $\varepsilon(U \mid U) \geq \delta(U \mid C)$ will be crucial in our extractor analysis. Similar to κ, we define

$$\delta(\mathcal{U}) := \min_{S \in \mathsf{Chains}^{\circ}(\mathcal{U})} \delta(\mathcal{U}(S) \mid \mathfrak{C})\,. \tag{7}$$

5.2 Two Chain Rules

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

Lemma 5.1 (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 $c \leftarrow_R \mathfrak{C}$ 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 5.2 (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 5.1, i.e. for $c \leftarrow_R \mathfrak{C}$ 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,

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

and thus

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

which completes the proof of the lemma. □

5.3 Knowledge Extractor Construction and Analysis

We are now ready to define and analyse our knowledge extractor Ext, adapting the work [AFKR23]. The extractor Ext is given rewindable black-box access to an adversary $\mathcal{A}\colon \mathfrak{C} \to \mathcal{T} \cup \{\bot\}$, denoted as $\mathsf{Ext}^{\mathcal{A}}$, 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 $\varSigma$-protocols, the transcripts $\tau_i = (a, c_i, z_i)$ have common first message a (as $\mathcal{P}^*$ is deterministic).

The extractor is formally defined in Figure 1. In its first step, $\mathsf{Ext}^{\mathcal{A}}$ evaluates $\mathcal{A}$ on a challenge $c \leftarrow_R \mathcal{U}(\emptyset) = \mathfrak{C}$ 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 $\mathsf{Ext}^{\mathcal{A}}_{\mathsf{iter}}(T)$ on input the sequence $T = (\tau)$ of transcripts.

The iterative sub-extractor $\mathsf{Ext}^{\mathcal{A}}_{\mathsf{iter}}(T)$ 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 $c \leftarrow_R \mathcal{U}(T)$ 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 $\mathsf{Ext}^{\mathcal{A}}_{\mathsf{iter}}\big((T, \tau)\big)$ returns. Otherwise, the extractor samples a coin *coin* with the following procedure:

– Sample $c \leftarrow \mathcal{U}\big(\mathsf{pred}(T)\big)$;
– If $c \notin \mathcal{U}(T) \subseteq \mathcal{U}\big(\mathsf{pred}(T)\big)$, evaluate $\tau \leftarrow \mathcal{A}(c)$;
– If $\tau \neq \bot$, then set $\mathit{coin} = 1$, otherwise set $\mathit{coin} = 0$.

If $\mathit{coin} = 1$, then the iterative extractor aborts and returns $\bot$. The above steps are repeated until a successful loop is reached, or until $\mathit{coin} = 1$ is flipped. We call $\mathsf{Ext}_{\mathsf{iter}}$ *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 $\mathsf{Ext}_{\mathsf{iter}}^{\mathcal{A}}\big((T, \tau)\big)$.

For completeness, we recall the differences of this knowledge extraction strategy from [AFKR23] (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 run-time is polynomial.

However, there are a number of crucial differences between (our adaptation of) the approach of [AFKR23] 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 [AFKR23], 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 [AFKR23] to handle UCSs. The careful coin tossing from [AFKR23] 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 [AFKR23] 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 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 $\mathsf{Ext}_{\mathsf{iter}}(T)$. In Lemma 5.3 we provide a lower bound for this success probability and in Lemma 5.4 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 run-time.

Lemma 5.3 (Success Probability of One Iteration of $\mathsf{Ext}_{\mathsf{iter}}^{\mathcal{A}}$). *Fix an arbitrary non-empty sequence of transcripts. Further, let us 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 $\mathsf{Ext}_{\mathsf{iter}}(T)$. We have

$$\Pr[\mathit{coin}(T) = 1] = \Pr[\mathcal{A}(c) \neq \bot \wedge c \notin \mathcal{U}(T) \mid c \leftarrow_R \mathcal{C}(T)] = \Delta(\mathcal{U}(T) \mid \mathcal{C}(T))\,. \quad (8)$$

```
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: $\mathcal{A}\colon \mathfrak{C} \to (\{0,1\}^* \times \mathfrak{C} \times \{0,1\}^*) \cup \{\bot\}$.

$\mathsf{Ext}_{\mathcal{U}}^{\mathcal{A}}$:

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

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

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

Fig. 1. Basic Knowledge Extractor. The comments track when $\mathsf{Succ}(T)$ transitions to 1.

Furthermore, let us define the random variables

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

which are the number of trials required — without ever aborting — until the loop finds $\tau \neq \bot$ (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 $\Pr[\mathsf{Succ}(T) = 1] = \Pr[X \le Y]$, as success occurs if and only if the extractor succeeds before the coin yields 1. Thus, by Lemma 3.1, we obtain that

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

Now note that, by Equations (5) and (6),

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

Therefore, since $x \mapsto \frac{x}{1-(1-x)(1-y)}$ in monotonically increasing for $y \in [0,1]$ (Lemma 3.2), 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 (6)), we observe that

$$\begin{aligned}
&(1-\delta(\mathcal{U}(T) \mid \mathcal{C}(T)))(1 - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T))) \\
&= (1 - \frac{\varepsilon(\mathcal{C}(T) \mid \mathcal{C}(T)) - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T))}{1 - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T))})(1 - \Delta(\mathcal{U}(T) \mid \mathcal{C}(T))) \\
&= 1 - \varepsilon(\mathcal{C}(T) \mid \mathcal{C}(T))\,.
\end{aligned} \tag{9}$$

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 5.4 (Success Probability of One Iteration of $\mathsf{Ext}^{\mathcal{A}}_{\mathsf{iter}}$). *In the same setting as Lemma 5.3, 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 5.3, but now using that $x \mapsto \frac{1}{1-(1-x)(1-y)}$ is monotonically decreasing for all $y \in [0,1]$ (Lemma 3.2) 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 5.5 ($\mathsf{Ext}^{\mathcal{A}}_{\mathsf{iter}}$ Run-Time). *Let $T \in \mathcal{T}^*$ be any sequence of transcripts, and let the random variable $\mathsf{time}(T) := \mathsf{time}_{\mathcal{A}}(\mathsf{Ext}^{\mathcal{A}}_{\mathsf{iter}}(T))$ denote the time spent in the blackbox $\mathcal{A}$ while running $\mathsf{Ext}^{\mathcal{A}}_{\mathsf{iter}}(T)$ until either $\mathsf{Succ}(T) = 1$ or $\mathsf{coin} = 1$ 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 $C \leftarrow_R \mathfrak{C}(T)$ is uniform.

Proof. Define $X(T)$ and $Y(T)$ as in Lemma 5.3. The number of iterations of the repeat-until loop in $\mathsf{Ext}_{\mathsf{iter}}^{\mathcal{A}}(T)$ is now given as $\min(X, Y)$. By Lemma 3.1, we have

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

where we first use that $x \mapsto \frac{1}{1-(1-x)(1-y)}$ is monotonically decreasing for all $y \in [0,1]$ (Lemma 3.2) 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 (9) above.

In each iteration, one query $\mathcal{A}(C)$ is made with $C \leftarrow_R \mathcal{C}(T)$, 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 3.3, it follows that

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

which completes the proof of the lemma. □

Next, we prove correctness of $\mathsf{Ext}_{\mathcal{U}}$ and bound its success probability and expected run-time.

Lemma 5.6 (Correctness of $\mathsf{Ext}_{\mathcal{U}}$). *Let $\mathcal{U}$ be a useful challenge structure on a challenge set $\mathfrak{C}$, and let $\mathcal{A}$ be some adversary. Then, the extractor* Ext *in Figure 1 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 $\mathsf{Ext}_{\mathcal{U}}$. □

Lemma 5.7. *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* Ext *in Figure 1 we obtain*

$$\Pr[\mathsf{Ext}^{\mathcal{A}} \neq \perp] \geq \delta(\mathcal{U}) = \min_{S\in\mathsf{Chains}^\circ(\mathcal{U})} \delta(\mathcal{U}(S)\mid\mathfrak{C})\,, \tag{10}$$

$$\begin{aligned}\mathbb{E}[\mathsf{time}_{\mathcal{A}}(\mathsf{Ext}^{\mathcal{A}})] = \max_{T\in\mathsf{Chains}^\circ(\mathcal{U})} \sum_{R\preceq T} &\mathbb{E}[\mathsf{time}(\mathcal{A}(C))\mid C\in\mathcal{U}(R)]\\ &+ \kappa(\mathcal{U}(R)\mid\mathcal{C}(R))\cdot\mathbb{E}[\mathsf{time}(\mathcal{A}(C))\mid C\in\mathcal{C}(R)\setminus\mathcal{U}(R)]\end{aligned} \tag{11}$$

Proof. We argue separately for success probability and run-time. 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 $\mathsf{Ext}_{\leq k}$ denote the extractor applied to $\mathcal{U}^{\leq k}$ instead of $\mathcal{U}$. More precisely, the extractor $\mathsf{Ext}_{\leq k}$ proceeds exactly as the extractor Ext, 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 $\mathsf{Ext}_{\leq k} = \mathsf{Ext}$ for all $k \geq \mathsf{depth}(\mathcal{U})$.

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

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

From this the claimed success probability immediately follows.

First, we observe that

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

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

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

$$\begin{aligned}
\Pr[\mathsf{Ext}^{\mathcal{A}}_{\leq k} = T] &= \Pr[\mathsf{Ext}^{\mathcal{A}}_{\leq \ell} = T] \\
&= \Pr[\mathsf{Ext}^{\mathcal{A}}_{\leq \ell-1} = T_{\ell-1}] \cdot \Pr[\Upsilon(T_{\ell-1}) = \tau_\ell] \\
&= \Pr[\mathsf{Ext}^{\mathcal{A}}_{\leq \ell-1} = T_{\ell-1}] \\
&\qquad \cdot \Pr[\mathsf{Succ}(T_{\ell-1}) = 1] \cdot \Pr[\Upsilon(T_{\ell-1}) = \tau_\ell \mid \mathsf{Succ}(T_{\ell-1}) = 1] \\
&= \Pr[\mathsf{Ext}^{\mathcal{A}}_{\leq \ell-2} = T_{\ell-2}] \cdot \Pr[\Upsilon(T_{\ell-2}) = \tau_{\ell-1}] \\
&\qquad \cdot \Pr[\mathsf{Succ}(T_{\ell-1}) = 1] \cdot \Pr[\Upsilon(T_{\ell-1}) = \tau_\ell \mid \mathsf{Succ}(T_{\ell-1}) = 1] \\
&\qquad \vdots \\
&= \prod_{i=1}^{\ell} \Big(\Pr[\mathsf{Succ}(T_{i-1}) = 1] \cdot \Pr[\Upsilon(T_{i-1}) = \tau_i \mid \mathsf{Succ}(T_{i-1}) = 1] \Big)\,. \qquad (12)
\end{aligned}$$

By Lemma 5.3, it now follows that for all $2 \le i \le \ell$

$$\begin{aligned}\Pr[\mathsf{Succ}(T_{i-1}) = 1] &\ge \frac{\varepsilon(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-1}))}{\varepsilon(\mathcal{C}(T_{i-1}) \mid \mathcal{C}(T_{i-1}))} \cdot \frac{1 - \kappa(\mathcal{U}(T_{i-1}) \mid \mathcal{C}(T_{i-1}))}{1 - \Delta(\mathcal{U}(T_{i-1}) \mid \mathcal{C}(T_{i-1}))} \\ &= \frac{\varepsilon(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-1}))}{\varepsilon(\mathcal{U}(T_{i-2}) \mid \mathcal{U}(T_{i-2}))} \cdot \frac{1 - \kappa(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-2}))}{1 - \Delta(\mathcal{U}(T_{i-1}) \mid \mathcal{U}(T_{i-2}))} ,\end{aligned}$$

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

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

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

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

From the chains rules of Lemma 5.1 and Lemma 5.2, it now follows that

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

Plugging this inequality into Equation 12 shows that

$$\Pr[\mathsf{Ext}^{\mathcal{A}}_{\le k} = T] \ge \delta(\mathcal{U}) \cdot \prod_{i=1}^{\ell} \Pr[\varUpsilon(T_{i-1}) = \tau_i \mid \mathsf{Succ}(T_{i-1}) = 1] ,$$

and thus

$$\begin{aligned}\Pr[\mathsf{Ext}^{\mathcal{A}}_{\le k} \ne \bot] &= \sum_{T \in \mathsf{Chains}(\mathcal{U}^{\le k})} \Pr[\mathsf{Ext}^{\mathcal{A}}_{\le k} = T] \\ &\ge \delta(\mathcal{U}) \cdot \sum_{\ell=1}^{k} \sum_{\substack{T=(\tau_1,\dots,\tau_\ell) \\ \in \mathsf{Chains}(\mathcal{U}^{\le k})}} \prod_{i=1}^{\ell} \Pr[\varUpsilon(T_{i-1}) = \tau_i \mid \mathsf{Succ}(T_{i-1}) = 1] \\ &= \delta(\mathcal{U}) .\end{aligned}$$

For the final equality, recall that $\varUpsilon(T_{i-1})$ 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 $\mathsf{Ext}_{\mathsf{iter}}$ spends running $\mathcal{A}$, and then use this to bound the expected time of Ext. For $\mathsf{Ext}_{\mathsf{iter}}$, 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 $T \in \mathsf{Chains}^\circ(\mathcal{U})$ 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 $\mathsf{Ext}^{\mathcal{A}}_{\mathsf{iter}}(T)$ spends in $\mathcal{A}$ until it either aborts or it finds the next transcript appending the sequence of transcripts T (see Lemma 5.5). Let $\mathsf{T}_{\mathsf{tot}}(T)$ be the random variable denoting the total expected time $\mathsf{Ext}^{\mathcal{A}}_{\mathsf{iter}}(T)$ spends in $\mathcal{A}$, that is, the time until $\mathsf{Ext}^{\mathcal{A}}_{\mathsf{iter}}(T)$ finishes (including recursive calls). It thus holds that

$$\mathbb{E}[\mathsf{T}_{\mathsf{tot}}(T)] = \mathop{\mathbb{E}}_{S=(T,\tau)}[\mathsf{T}(T) + \mathsf{T}_{\mathsf{tot}}(S)] = \mathbb{E}[\mathsf{T}(T)] + \mathop{\mathbb{E}}_{S=(T,\tau)}[\mathsf{T}_{\mathsf{tot}}(S)] \,, \tag{13}$$

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 $\mathsf{depth}_T(\mathcal{U}) \leq k$ then

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

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

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 $T \preceq T'' \in \mathsf{Chains}^\circ(\mathcal{U})$, 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 $\mathsf{Ext}^{\mathcal{A}}_{\mathsf{iter}}(T)$ until $coin(T) = 1$ or $\mathsf{Succ}(T) = 1$. By Lemma 5.5 we have

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

which yields the induction base case.

For the induction step, assume for $\mathsf{depth}_T(\mathcal{U}) \leq k$ the claim holds. Let T be such that $\mathsf{depth}_T(\mathcal{U}) \leq k+1$. The execution of $\mathsf{Ext}^{\mathcal{A}}_{\mathsf{iter}}(T)$ first runs the loop where it either finds $\tau \neq \bot$, defines S and runs $\mathsf{Ext}^{\mathcal{A}}_{\mathsf{iter}}(S)$ (and sets $\mathsf{Succ}(T) = 1$), or it aborts (due to $coin = 1$). Now, by Lemma 5.5, we find

$$\mathbb{E}[\mathsf{T}(T)] = \tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))}\big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(T)] + \kappa(R) \cdot \mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(T)]\big)\,. \tag{14}$$

Write $S = \bot$ if $\tau = \bot$. Then $S = \bot$ means that $\mathsf{Ext}_{\mathsf{iter}}(S)$ 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:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{S}[\mathsf{T}_{\mathsf{tot}}(S)] = \Pr[S \neq \bot] \cdot \mathop{\mathbb{E}}_{S}[\mathsf{T}_{\mathsf{tot}}(S) \mid S \neq \bot] \\
&\leq \tfrac{\varepsilon(\mathcal{U}(T)|\mathcal{U}(T))}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))} \cdot \mathop{\mathbb{E}}_{S}[\mathsf{T}_{\mathsf{tot}}(S) \mid S \neq \bot] \\
&\leq \tfrac{\varepsilon(\mathcal{U}(T)|\mathcal{U}(T))}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))} \mathop{\mathbb{E}}_{S}\Big[\tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))} \max_{S \preceq S''} \sum_{S \preceq S' \preceq S''} \big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(S')] + \kappa(S') \cdot \mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(S')]\big)\Big] \\
&\leq \tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))} \cdot \mathop{\mathbb{E}}_{S}\Big[\max_{S \preceq S''} \sum_{S \preceq S' \preceq S''} \big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(S')] + \kappa(S') \cdot \mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(S')]\big)\Big] \\
&= \tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))} \cdot \max_{T \prec S} \max_{S \preceq S''} \sum_{S \preceq S' \preceq S''} \big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(S')] + \kappa(S') \cdot \mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(S')]\big) \\
&= \tfrac{1}{\varepsilon(\mathcal{C}(T)|\mathcal{C}(T))} \cdot \max_{T \prec S''} \sum_{S \preceq S' \preceq S''} \big(\mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \mathcal{U}(S')] + \kappa(S') \cdot \mathbb{E}[\mathsf{T}_{\mathcal{A}} \mid \overline{\mathcal{U}}(S')]\big)\,, \qquad (15)
\end{aligned}$$

where we first use that $\Pr[\mathsf{T}_{\mathsf{tot}}(S) \mid S = \bot] = 0$; then $\Pr[S \neq \bot] \leq \frac{\varepsilon(\mathcal{U}(S)|\mathcal{U}(S))}{\varepsilon(\mathcal{C}(S)|\mathcal{C}(S))}$ by Lemma 5.4; the next step uses that $\mathsf{T}_{\mathsf{tot}}(S)$ is distributed as the number of queries $\mathsf{Ext}_{\mathsf{iter}}(S)$ (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 (14) and (15) back into (13), which yields

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

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 Ext makes exactly 1 query to $\mathcal{A}$ and then runs $\mathsf{Ext}_{\mathsf{iter}}$ 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 Ext 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 $T'' \in \mathsf{Chains}^{\circ}(\mathcal{U})$, the claim of the theorem follows (with notation R, T instead of T', T''). □

The bounds in Lemma 5.7 are precise but unwieldy. Below, we provide simplified bounds which are easy to use and sufficient for most applications.

Corollary 5.8 ($\mathsf{Ext}^{\mathcal{A}}$ Success Probability and Run-Time). *Let $\mathfrak{C}$ be a challenge set and let $(\mathcal{U}, \mathcal{C})$ be a useful challenge structure on $\mathfrak{C}$ which satisfies $\mathcal{C}(T) = \mathcal{U}(\mathsf{pred}(T))$ for all $T \in \mathsf{Chains}(\mathcal{U})$. Let $\mathcal{A}$ be some adversary. Then, for the extractor* Ext *in Fig. 1 we obtain*

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

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

Proof. All claims are immediate consequences of Lemma 5.7 and our requirement on $(\mathcal{U}, \mathcal{C})$. For the first claim, substitute the definition $\delta(U \mid C) = \frac{\varepsilon(C|C) - \Delta(U|C)}{1 - \Delta(U|C)}$ from (6) into $\delta(\mathcal{U})$ from (7). Next, from $\Delta(U|C) \leq \kappa(U|C)$ and monotonicity of $f(x,y) = \frac{x-y}{1-y}$ (see Lemma 3.2), the claim on the success probability follows.

To prove the bound on expected run-time, 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 $\kappa(\mathcal{U}) = \max_{T \in \mathsf{Chains}^{\circ}(\mathcal{U})} \kappa(\mathcal{U}(T) \mid \mathcal{C}(T))$, 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 5.7 and Corollary 5.8.

Theorem 5.9 (Adaptive Special Soundness $\implies$ Knowledge Soundness). *For each statement* $\mathsf{stmt} \in \{0,1\}^*$*, let* $\mathcal{U}_{\mathsf{stmt}}$ *be a UCS over the challenge set* $\mathfrak{C}_{\mathsf{stmt}}$ *such that* $\mathsf{depth}(\mathcal{U}_{\mathsf{stmt}})$ *is polynomial in* stmt*, and let* $\mathcal{U} = (\mathcal{U}_{\mathsf{stmt}})_{\mathsf{stmt}}$*. Let* $\Pi = (\mathcal{P}, \mathcal{V})$ *be a* $\mathcal{U}$*-adaptive special sound* 3*-move public-coin interactive proof for relation* Ξ*. Then* Π *is knowledge sound with knowledge error* $\kappa(\mathcal{U}_{\mathsf{stmt}})$*.*

The proof is straightforward.

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 5.8, we obtain such a $\mathcal{U}$-chain T in expected polynomial time and with knowledge error $\kappa(\mathcal{U}_{\mathsf{stmt}})$. This completes the proof.

6 Examples and Comparisons with Related Work

We provide some examples of UCSs.

6.1 Transcript-Agnostic Examples and Access Structures [AFR23]

Example 6.1 (k-special soundness). Consider a challenges space $\mathfrak{C} = \{1, \dots, N\}$ of size N (and $k \leq N$). The useful challenge structure corresponding to k-out-of-N-special soundness is $\mathcal{U}(\tau_1, \dots, \tau_\ell) = \mathfrak{C} \setminus \{\mathsf{challof}(\tau_i)\}_{i=1}^{\ell}$, for $\ell < k$, and $\mathcal{U}(\tau_1, \dots, \tau_\ell) = \emptyset$ for $\ell \geq k$. Clearly, $\kappa(\mathcal{U}) = \frac{k-1}{N}$ and $\mathsf{depth}(\mathcal{U}) = k$.

Remark 6.2 (Large knowledge error). For very large knowledge error $\kappa(\mathcal{U}) \approx 1$, the run-time bound of Lemma 5.7 may be dominated by a few summands. From this, slightly more precise bounds can be derived than those of Corollary 5.8. E.g., for N-out-of-N special soundness, one gets the bound $N(\ln(N) + 1)$ instead of N^2.

Example 6.3 (Access-structure special soundness). Special soundness w.r.t. an access structure as in [AFR23] is a special case of our definition. We define $c \in \mathcal{U}(S)$ if and only if c is a useful challenge given $S \subseteq \mathfrak{C}$ according to the access structure Γ (see [AFR23] for the definition of useful). 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.

Remark 6.4 (Continued comparison with [AFR23]). Our definitions of and properties for useful challenge structures are also 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. The crucial difference (and generalisation) of our notion is that $\mathcal{U} \colon \mathcal{T}^* \to 2^{\mathfrak{C}}$ takes as input a number 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 example explains the necessity of sequential composition, despite the apparent ability of handling tree structures "directly" via a UCS.

Example 6.5 (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. 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 $|\mathfrak{C}_2| - (k-1)$ challenges are useful.

The previous examples did not benefit from the possibility to adaptively specify useful challenges much. For our next examples, this possibility is crucial.

Example 6.6 (t-parallel repetition). Let $(\mathcal{U}, \mathcal{C})$ be a challenge structure over $\mathfrak{C}$. Define the useful challenge structure $(\mathcal{U}', \mathcal{C}')$ over $\mathfrak{C}^t$ as the t-fold product of $(\mathcal{U}, \mathcal{C})$. More precisely,

$$\mathcal{U}'((\tau_j^1)_{j=1}^t, \ldots, (\tau_j^i)_{j=1}^t) = \bigcup_{j=1}^t \mathfrak{C}^{j-1} \times \mathcal{U}(\tau_j^1, \ldots, \tau_j^i) \times \mathfrak{C}^{t-j}$$

where by abuse of notation, we write $\mathcal{U}(\tau_j^1, \ldots, \tau_j^i)$, but mean $\mathcal{U}(\tau_1', \ldots, \tau_\ell')$ for the subsequence $(\tau_1', \ldots, \tau_\ell')$ of $(\tau_j^1, \ldots, \tau_j^i)$ which is a $\mathcal{U}$-chain. Observe that, by definition of $\mathcal{U}'$, if $(\tau_j^1)_{j=1}^t, \ldots, (\tau_j^k)_{j=1}^t$ is a chain w.r.t. $\mathcal{U}'$, then for the i-th τ^i, there is at least one thread, say j, where $\tau_j^i \in \mathcal{U}(\tau_1^j, \ldots, \tau_i^j)$. Also note that if $\mathcal{U}(\tau_1^j, \ldots, \tau_i^j) = \emptyset$, then $\mathcal{U}'(\tau^1 \ldots, \tau^i) = \emptyset$. Thus, 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 6.6 does not seem to generalise to multi-round protocols.

6.2 A Basic Lattice Example

In all of the previous examples, the UCS only depended on the *challenges*, and not on the actual transcripts. This is unsurprising, as all were simple and generic concepts, 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 to react to *failures of probabilistic tests*, from which extraction may either be possible, or not desired (because it would blow up the run-time complexity). We give a toy example of a 2-move protocol.

Example 6.7 (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 $\mathbf{B} \leftarrow_R \{0,1\}^{\lambda \times m}$ 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.[17] 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 $(\mathbf{A}, \mathbf{y})$):

- $\mathcal{U}(\emptyset) := \mathfrak{C}$ as always,
- $\mathcal{U}((\mathbf{B}, \mathbf{x})) := \emptyset$ if $\|\mathbf{x}\| \leq \alpha$ and $(\mathbf{B}, \mathbf{x})$ is accepting (so $\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q$).
- $\mathcal{U}((\mathbf{B}, \mathbf{x})) := \{\mathbf{B}' \mid \|\mathbf{B}'\mathbf{x}\| > \beta\}$ if $\|\mathbf{x}\| > \alpha$ and $(\mathbf{B}, \mathbf{x})$ is accepting.
- $\mathcal{U}((\mathbf{B}, \mathbf{x}), (\mathbf{B}', \mathbf{x}')) = \emptyset$ if both $(\mathbf{B}, \mathbf{x})$ and $(\mathbf{B}', \mathbf{x}')$ are accepting.

[17] 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 $\mathbf{x}$.

- $\mathcal{U}(T) \coloneqq \bot$ for all other cases of T.

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 when given a sequence of transcripts T satisfying $\mathcal{U}(T) = \emptyset$. To see this, note that $\mathcal{U}(T) = \emptyset$ only happens in 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 6.1, the challenge $\mathbf{B}$ says nothing about which $\mathbf{B}'$ enforce $\mathbf{x}' \neq \mathbf{x}$.
- 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.[18]

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 10, we see how to handle a lattice-based folding protocol (building on the ideas of [BF23]) with an implicit probabilistic test. Moreover, in Section 6.3 we implicitly show that adaptive special soundness also applies to Labrador [BS23].

6.3 Predicate and Almost Special Soundness

In this section, we take a closer look at the concurrent work [AAB+24b], which introduces predicate special soundness to study the proof system of [BS23]. We also extend our comparison with almost special soundness [BF23], which was introduced to study certain folding protocols, one of which we also analyze using adaptive special soundness in Section 10. 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.

[18] 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.

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, [AAB+24a] 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 parameterized 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 10 a clear indicator for this plausibility. Secondly, we note that the extractor construction of [BF23] builds on [BCC+16], which uses a Markov bound for run-time 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 (sequential) composition result provided in [BF23].

Predicate Special Soundness The notion of predicate special soundness [AAB+24a] 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:

- $\phi_\ell^{\mathsf{chall}}(c_1, \dots, c_\ell) \in \{0, 1\}$ indicates if a challenge c_ℓ is useful, given previous challenges $c_1, \dots, c_{\ell-1}$, for $\ell = 1, \dots, k$. For example, $\phi_\ell^{\mathsf{chall}}(c_1, \dots, c_\ell)$ is 1 if all challenges distinct (capturing k-special soundness), or linearly independent, or any other predicate.
- $\phi^{\mathsf{prop}}(\tau_1, \dots, \tau_{k-1}) \in \{0, 1\}$ 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.
- $\phi^{\mathsf{bind}}((\tau_1, \dots, \tau_{k-1}), \tau_k) \in \{0, 1\}$ indicates whether a "binding" holds for the k extracted transcripts.

Notice that $\phi_\ell^{\mathsf{chall}}$ is defined for every $\ell = 1, \dots, k$, 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 $\tau_1, \dots, \tau_{k-1}$ may fail to satisfy a property, i.e. $\phi^{\mathsf{prop}}(\tau_1, \dots, \tau_{k-1}) = 0$, 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. $\phi^{\mathsf{bind}}((\tau_1, \dots, \tau_{k-1}), \tau_k) = 1$.

This idea is quantified through the so-called *failure densities* [AAB+24b, Definition 5.2/6] of the predicate ϕ^{chall} and the pair $(\phi^{\mathsf{prop}}, \phi^{\mathsf{bind}})$, where essentially:

- The failure density p_ℓ^{chall} of $\phi_\ell^{\mathsf{chall}}$ is

$$\max_{c_1,\dots,c_{\ell-1}} \Pr_{c_\ell}[\phi_\ell^{\mathsf{chall}}(c_1, \dots, c_\ell) = 0 \mid \phi^{\mathsf{chall}}(c_1, \dots, c_{\ell-1}) = 0]$$

  where $c_\ell \leftarrow_R \mathfrak{C} \setminus \{c_1, \dots, c_{\ell-1}\}$ is uniformly random. That is, this is the worst-case probability for a challenge being "bad" in round ℓ.
- The failure density p^{prop} of ϕ^{chall} is

$$\max_{\tau_1,\dots,\tau_{k-1}} \Pr_{c_k}\left[\mathcal{A}(c_k) \neq \bot \;\middle|\; \begin{array}{c} \phi_k^{\mathsf{chall}}(c_1, \dots, c_k) = 1 \\ \wedge \phi^{\mathsf{prop}}((\tau_1, \dots, \tau_{k-1}), \tau_k) = 0 \\ \wedge \exists \tau_k \colon c_k = \mathsf{challof}(\tau_k) \wedge \phi^{\mathsf{bind}}((\tau_1, \dots, \tau_{k-1}), \tau_k) = 1 \end{array}\right]$$

  where $c_k \leftarrow_R \mathfrak{C}$ and $c_i = \mathsf{challof}(\tau_i)$. 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 [AAB+24b, Theorem 5.1] asserts that the probability that an extracted tree falls into this case is bounded (roughly) by $p^{\mathsf{prop}} + \sum_{i=1}^{k} p_i^{\mathsf{chall}}$.

We note here that, similar to almost special soundness, predicate special soundness was introduced and analyzed for the Fiat–Shamir transformation of protocols and extended to a component-wise special soundness notion as well, since that is required for analyzing 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 [AAB+24b], all predicates are defined w.r.t. to a (sub)tree of transcripts. Modularity of the notion is not considered in [AAB+24b], and no (sequential) composition results are discussed. Hence, the predicates must be specified for the full protocol in one go, and analyzed 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 2.3), without increasing either run-time or knowledge error significantly (compared to adaptive special soundness). The problem is, that the extractor in [AAB+24b] uses [AFK22] essentially black-box (which is optimized 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 $\phi^{\mathsf{chall}}_{\ell}$ is monotone (and, if we consider trees as inputs, also that families for ϕ^{prop} and ϕ^{bind} are monotone).[19]
- *Efficiency:* The maps $\phi^{\mathsf{chall}}_{i}(c_1, \dots, c_i)$ 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 [AAB+24a]). In this case, there is a natural translation of the predicate system into a useful challenges structure. For this, let

- $\mathcal{U}^{\mathsf{chall}}(\tau_1, \dots, \tau_\ell) = \{c \in \mathfrak{C} \mid \phi^{\mathsf{chall}}_{\ell}(c_1, \dots, c_\ell) = 1\}$, where we write $c_i = \mathsf{challof}(\tau_i)$. (More precisely, we have a sequence $\mathcal{U}^{\mathsf{chall}}_{j}$, with one UCS per protocol round, and c_i is only the i-th round challenge.)
- $\mathcal{U}^{\mathsf{prop}}(\tau_1, \dots, \tau_i) = \mathfrak{C}$ for $i < k-1$ and
$$\mathcal{U}^{\mathsf{chall}}(\tau_1, \dots, \tau_{k-1}) = \{c \in \mathfrak{C} \mid \phi^{\mathsf{prop}}(\tau_1, \dots, \tau_{k-1}) = 0 \wedge \psi^{\mathsf{bind}}((\tau_1, \dots, \tau_{k-1}), c)\}$$

[19] By monotone, we mean that if the predicate is satisfied for some input, then it will be satisfied given additional challenges (or transcripts) as input.

– $\mathcal{U}(\tau_1, \dots, \tau_i) = \mathcal{U}^{\mathsf{chall}}(\tau_1, \dots, \tau_i) \cup \mathcal{U}^{\mathsf{prop}}(\tau_1, \dots, \tau_i)$ is the union of the UCSs.

By our monotonicity assumptions $\mathcal{U}^{\mathsf{chall}}$ and $\mathcal{U}^{\mathsf{prop}}$ are UCSs, and it is straightforward to observe that the union of two UCSs is again a UCS, and the knowledge error satisfies

$$\kappa(\mathcal{U}) = \kappa(\mathcal{U}^{\mathsf{chall}} \cup \mathcal{U}^{\mathsf{prop}}) \leq \kappa(\mathcal{U}^{\mathsf{chall}}) + \kappa(\mathcal{U}^{\mathsf{prop}}).$$

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

$$\kappa(\mathcal{U}^{\mathsf{prop}}) \leq p^{\mathsf{prop}}$$

Hence, under these mild restrictions, adaptive special soundness essentially encompasses predicate special soundness for 3-move protocols. By sequential composition and the tree-structure of the predicates in [AAB+24b], this extends beyond 3-move protocols through our generalized sequential composition of extractors. As a consequence, adaptive special soundness applies to Labrador [BS23].

Remark 6.8 (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 run-time as much as possible, and track failure events to handle non-monotone or inefficient predicates.

7 Sequential Composition of UCS Extractors

This section outlines our generalised sequential composition of extractors, specialised to extractors for useful challenge structures. In Section 9, we provide the full treatment of sequential composition of arbitrary extractors. Unless mentioned otherwise, the challenge set is assumed $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$ below.

Standard Composition: Extending the Challenge Space. As noted in Section 2.4, standard sequential composition is insufficient to make full use of UCS-extraction. For generalised composition, the parent extractor chooses the initial challenge for its child extractor. We sketch how standard composition fits into that. More detailed explanations are given in Section 8.1.

Let us consider the case where $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$ consists of two challenges (e.g. in a 5-move protocol). For standard sequential composition, it is natural to define a UCS $\mathcal{U}$ only over one challenge, i.e. over $\mathfrak{C}'$ or $\mathfrak{C}''$, and handle more via composition. For example, (k', k'')-special soundness is handled like this. However, for our generalised sequential composition, the extractor $\mathsf{Ext}_{\mathcal{U}}$ must be defined over $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$. In case $\mathcal{U}$ is only defined over $\mathfrak{C}''$, the straightforward way to extend extraction to $\mathfrak{C}$ is by making $\mathsf{Ext}_{\mathcal{U}}$ pick the first challenge $C_1 = (C_1', C_1'')$ randomly and then fix $C_i' = C_1'$ for all future challenges. This preserves run-time and knowledge error.

Towards Generalised Composition: Families of UCSs. With the above, defining a UCS $\mathcal{U}$ only over $\mathfrak{C}''$ and "extending" the extractor captures standard sequential composition. However, it is insufficient for our applications of adaptive special soundness, because there, the useful challenges in $\mathcal{U}$ can *depend on* c'. To capture this, we consider a *family* of UCSs $(\mathcal{U}_{c'})_{c' \in \mathfrak{C}'}$ over $\mathfrak{C}''$ instead. Alternatively, we can look at a UCS $\mathcal{U}$ over $\mathfrak{C}$ which is $\mathfrak{C}'$-fixing:

Definition 7.1 ($\mathfrak{C}'$-fixing). *Let $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$. A UCS $\mathcal{U}$ is* ***$\mathfrak{C}'$-fixing*** *if for every $(\tau_1, \dots, \tau_i) \in \mathsf{Chains}(\mathcal{U})$, there is a $c' \in \mathfrak{C}'$ such that $\mathsf{challof}_{\mathcal{A}}(\tau_i) \in \{c'\} \times \mathfrak{C}''$.*

For convenience, any UCS $\mathcal{U}$ is $\mathfrak{C}$-fixing, which puts no restriction[20] *on $\mathcal{U}$.*

Clearly, a family of UCSs $(\mathcal{U}_{c'})_{c' \in \mathfrak{C}'}$ over $\mathfrak{C}''$ is equivalent to a UCS $\mathcal{U}$ over $\mathfrak{C}$ which is **$\mathfrak{C}'$-fixing**. From family to UCS, define $\mathcal{U}(\tau_1, \dots, \tau_i) = \mathcal{U}_{c'}(\tau_1, \dots, \tau_i)$, where $c' \in \mathfrak{C}$ is the unique c' from Definition 7.1 (and $\emptyset$ outside of chains).[21] From UCS to family, use the same equality in reverse.

The natural definition of the knowledge error of a family $\mathcal{U} = (\mathcal{U}_{c'})_{c' \in \mathfrak{C}'}$ is to take the maximum over all all c'.

Definition 7.2. *For a $\mathfrak{C}'$-fixing UCS $\mathcal{U}$, denote by*

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

a more precise knowledge error of $\mathcal{U}$. For convenience, we define $\kappa_{\mathfrak{C}}(\mathcal{U}) = \kappa(\mathcal{U})$ for any UCS $\mathcal{U}$.

Note that we write $\kappa_{\mathfrak{C}'}(\mathcal{U})$ to clarify that we consider the knowledge error of $\mathcal{U}$ as a family of UCS which is $\mathfrak{C}'$-fixing. The knowledge error $\kappa(\mathcal{U})$ of $\mathcal{U}$ over the ambient space $\mathfrak{C}$ is much worse.[22]

The depth of a $\mathfrak{C}'$-fixing UCS $\mathcal{U}$ is the same as the maximum depth over the respective UCS family $(\mathcal{U}_{c'})_{c' \in \mathfrak{C}'}$, i.e. $\mathsf{depth}(\mathcal{U}) = \max_{c' \in \mathcal{C}'} \mathsf{depth}(\mathcal{U}_{c'})$ holds (essentially by definition). Hence, we make no explicit disambiguation there. Finally, note that the picking a family which is constant in $\mathfrak{C}'$ specialises the setting to standard sequential composition.

The following result asserts that properties of families behave as expected.

[20] Fomally, $\mathfrak{C} = \{c'\} \times \mathfrak{C}'$ is impossible. But we can extend $\mathfrak{C}$ to the naturally isomorphic set $\{0\} \times \mathfrak{C}$, where $\{0\}$ is an arbitrary set of one element.

[21] Recall that only properties of $\mathcal{U}$-chains are of interest to extraction (and knowledge error or depth of $\mathcal{U}$). So for an equivalence of UCSs, we do not care how they are defined outside of chains.

[22] If $\mathcal{U}$ is $\mathfrak{C}'$-fixing, then the knowledge error satisfies $\kappa(\mathcal{U}) \geq 1 - 1/|\mathfrak{C}'| \approx 1$.

Lemma 7.3. *Let $\mathcal{U} = (\mathcal{U}_{c'})_{c' \in \mathfrak{C}'}$ be a family of UCS over $\mathfrak{C}''$ (i.e., a $\mathfrak{C}'$-fixing UCS). Let $\mathsf{Ext}_{\mathcal{U}}$ be the UCS extractor for $\mathcal{U}$ (over $\mathfrak{C}$). Then $\mathsf{Ext}_{\mathcal{U}}$ has knowledge error $\kappa_{\mathfrak{C}'}(\mathcal{U})$ and expected run-time of $\mathsf{depth}(\mathcal{U})$.*

Proof. The run-time claim follows immediately from Corollary 5.8 (as the depth is unchanged). The knowledge error follows by a simple application of Jensen's inequality, see Lemma 9.9 and Example 9.10.

Sequential Composition of UCS Extractors. Now, we turn back to sequential composition and define our generalised notion. Let $\mathcal{A}$ be an abstract adversary over $\mathfrak{C}$ which is V-respecting. Following definition is for the general case; the reader may want to think the specific challenge space $\mathfrak{C} = \mathfrak{C}_1 \times \ldots \times \mathfrak{C}_\mu$, where $\mathfrak{C}_i$ can be viewed as the i-th challenge set of a $(2\mu + 1)$-move public coin protocol $\Pi = (\mathcal{P}, \mathcal{V})$ and V is the transcript verification of $\mathcal{V}$.

Definition 7.4 (Composition of UCS extractors). *Let $(\mathcal{U}_i)_{i=1}^{\mu}$ be UCSs over $\mathfrak{C}$. For $\mathcal{A}$, define $(\mathsf{Ext}_i)_{i=1}^{\mu+1}$ as follows: Let $\mathsf{Ext}_\mu = \mathsf{Ext}_{\mathcal{U}_\mu}$ and let $\mathsf{Ext}_i(c) = \mathsf{Ext}_{\mathcal{U}_i}^{\mathsf{Ext}_{i+1}^{\mathcal{A}}}(c)$, with each Ext_i defined over $\mathfrak{C}$. We call Ext_1 the (final) composed extractor.*

Observe that, by construction, Ext_i outputs a nested sequence of transcripts of height $\mu - i + 1$. Hence, $\mathcal{U}_i$ expects nested $(\mathcal{U}_{i+1}, \ldots, \mathcal{U}_\mu)$-chains as "transcripts". 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 $\mathsf{challof}_{\mathcal{A}_{i+1}}(\mathcal{A}_{i+1}(c)) = c$ must be satisfied (unless $\mathcal{A}_{i+1}(c) = \bot$), cf. Remark 4.5. We thus find:

Lemma 7.5 (Correctness). *Let $(\mathcal{U}_1, \ldots, \mathcal{U}_\mu)$ be UCSs over $\mathfrak{C}$. Let $\mathcal{A}$ be a V-respecting adversary. Let Ext_1 denote the composed extractor of the sequential composition of $(\mathsf{Ext}_{\mathcal{U}_i})_{i=1}^{\mu}$. Then Ext_1 outputs complete V-respecting nested $(\mathcal{U}_1, \ldots, \mathcal{U}_\mu)$-chains. That is, Ext_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 $(\mathcal{U}_1, \ldots, \mathcal{U}_\mu)$-chain.*

The proof follows from a simple induction.

Proof. The case of $\mu = 1$ is immediate: By Lemma 5.6, $\mathsf{Ext}_{\mathcal{U}}$ outputs a list $T = (\tau^1, \ldots, \tau^k)$ 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 $(\mathcal{U}_i, \ldots, \mathcal{U}_\mu)$-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, $\mathsf{Ext}_{\mathcal{U}_i}$ is $(\mathsf{V}_i^{\mathsf{pre}}, \mathsf{V}_i^{\mathsf{post}})$-compatible; the proof is analogous to $\mu = 1$: By Lemma 5.7, $\mathsf{Ext}_{\mathcal{U}_i}$ outputs $T = (S^1, \ldots, S^k) \in \mathsf{Chains}(\mathcal{U}_i)$ with $\mathcal{U}_i(S^1, \ldots, S^k) = \emptyset$. By $\mathsf{V}_i^{\mathsf{pre}}$ each S^j is a complete V-respecting nested $(\mathcal{U}_{i+1}, \ldots, \mathcal{U}_\mu)$-chain. Hence, T is a complete V-respecting nested $(\mathcal{U}_i, \ldots, \mathcal{U}_\mu)$-chain. □

Theorem 7.6 (Knowledge Error and Run-Time Bounds for $(\mathcal{U}_1, \ldots, \mathcal{U}_\mu)$-chains). *Let $(\mathcal{U}_1, \ldots, \mathcal{U}_\mu)$ be UCS over $\mathfrak{C}$ and suppose $\mathcal{U}_i$ is $\mathfrak{C}_i$-fixing.*[23] *Let Ext_1 be the final composed extractor (as in Lemma 7.5). The success of Ext_1 satisfies*

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

The expected time Ext_1 spends in $\mathcal{A}$ is bounded by

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

Proof. The run-time bound follows by a straightforward induction, repeatedly applying Corollary 5.8. Namely, let $\mathcal{A} = \mathcal{A}_{\mu+1}$ and $\mathcal{A}_i(c) = \mathsf{Ext}_{\mathcal{U}_i}^{\mathcal{A}_{i+1}}(c)$ for $i = 1, \ldots, \mu$ we have that

$$\mathop{\mathbb{E}}_{c \leftarrow_R \mathfrak{C}}[\mathsf{time}_{\mathcal{A}}(\mathcal{A}_i)] \leq \frac{\mathsf{depth}(\mathcal{U}_i)}{1 - \kappa(\mathcal{U}_i)} \cdot \mathop{\mathbb{E}}_{c \leftarrow_R \mathfrak{C}}[\mathsf{time}_{\mathcal{A}}(\mathcal{A}_{i+1})]$$

For the knowledge error, nothing interesting happens and the proof is a straightforward induction, repeatedly applying the generalized sequential composition lemma for knowledge errors (Lemma 9.9). For the refined bounds, one just observes that the extractor $\mathsf{Ext}_{\mathcal{U}_i}$ has knowledge error $\kappa_{\mathfrak{C}_i}(\mathcal{U}_i)$ which may be much better than $\kappa(\mathcal{U}_i)$. □

It is now immediate that adaptive special sound protocols are proofs of knowledge with extractors whose run-time and success is bounded as in Theorem 7.6. For completeness, we state it below.

Theorem 7.7 ($(\mathcal{U}_1, \ldots, \mathcal{U}_\mu)$-adaptive special soundness implies knowledge soundness). *Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a public-coin proof system for Ξ with challenge space $\mathfrak{C}$. Suppose Π is $(\mathcal{U}_{\mathsf{stmt},1}, \ldots, \mathcal{U}_{\mathsf{stmt},\mu})_{\mathsf{stmt}}$-adaptive special sound, where $\mathcal{U}_i$ is $\mathfrak{C}_i$-fixing. Let stmt be some statement and*

$$\kappa(\mathsf{stmt}) = 1 - \prod_{i=1}^{\mu}(1 - \kappa_{\mathfrak{C}_i}(\mathcal{U}_{\mathsf{stmt},i})) ,$$

$$\rho(\mathsf{stmt}) = \frac{1}{1 - \kappa(\mathsf{stmt})} \cdot \prod_{i=1}^{\mu} \mathsf{depth}(\mathcal{U}_{\mathsf{stmt},i}) .$$

There exists a universal extractor Ext, which given a statement stmt and black-box access to any malicious prover $\mathcal{P}^*$, has run-time tightness $\rho(\mathsf{stmt})$, i.e. spends at

[23] Recall that if $\mathfrak{C}_i = \mathfrak{C}$, then $\mathcal{U}_i$ is just any UCS. Also consider the typical extraction of protocol rounds, where with $\mathfrak{C} = \mathfrak{C}'_1 \times \ldots \mathfrak{C}'_\mu$ and $\mathcal{U}_i$ is a typical UCS for i-th challenge extraction. Then $\mathfrak{C}_i = \mathfrak{C}'_1 \times \ldots \times \mathfrak{C}'_{i-1}$, because $\mathcal{U}_i$ only cares about the i-th challenge and later ones. Challenges before the i-th round are kept fixed by the i-th round extractor (and hence $\mathcal{U}_i$).

most an expected time of $\rho(\mathsf{stmt}) \cdot \mathbb{E}[\mathsf{time}(\mathcal{A}(C)) \mid C \leftarrow_R \mathfrak{C}]$ *in running* $\mathcal{A}$*, and outputs a witness* $\mathsf{wit} \in \Xi(\mathsf{stmt})$ *with probability at least*

$$\Pr_{C \leftarrow_R \mathfrak{C}}\big[\mathsf{Ext}^{\mathcal{A}}(C) = \mathsf{wit} \in \Xi(\mathsf{stmt})\big] \geq \frac{\Pr[\langle \mathcal{P}^*(\mathsf{stmt}), \mathcal{V}(\mathsf{stmt})\rangle = 1] - \kappa(\mathsf{stmt})}{1 - \kappa(\mathsf{stmt})} .$$

We note here that in the family $(\mathcal{U}_{\mathsf{stmt},1}, \ldots, \mathcal{U}_{\mathsf{stmt},\mu})_{\mathsf{stmt}}$, all UCSs $\mathcal{U}_{\mathsf{stmt},i}$ have the *same* parameter stmt, i.e. this parameter is fixed throughout extraction. An adaptive choice of stmt on, say the i-th level, is not covered by our analysis.

8 Examples for Sequential Composition

8.1 Translation between Extractor Settings

Recall that our extractors are always defined over the whole challenge space $\mathfrak{C}$. For "usual" sequential composition, this is not used and extractors consider only one protocol round instead. We explain how to "extend" (sub-)extractors to the entire challenge space $\mathfrak{C}$.

Remark 8.1 (Challenge-Set Extension of Ext_i*).* Let Ext_i be an extractor defined for $\mathcal{A}_i \colon \mathfrak{C}_i \times \ldots \times \mathfrak{C}_\mu \to \{0,1\}^* \cup \{\bot\}$. We extend Ext_i to any adversary $\mathcal{A} \colon \mathfrak{C} \to \{0,1\}^* \cup \{\bot\}$ as follows: $\mathsf{Ext}_i^{\mathcal{A}}(c_1, \ldots, c_\mu)$ runs $\mathsf{Ext}_i^{\mathcal{A}(c_1,\ldots,c_{i-1},\,\cdot\,)}(c_i, \ldots, c_\mu)$, i.e.:

- Pick $(c_1, \ldots, c_{i-1})$ uniformly at random (or take the input) and fix it.
- With $(c_1, \ldots, c_{i-1})$ fixed, let $\mathcal{A}_i \colon \mathfrak{C}_i \times \ldots \times \mathfrak{C}_\mu \to \{0,1\}^* \cup \{\bot\}$ where $\mathcal{A}_i(\cdot) := \mathcal{A}(c_1, \ldots, c_{i-1}, \,\cdot\,)$. Then run $\mathsf{Ext}_i^{\mathcal{A}_i}(c_i, \ldots, c_\mu)$.

If $\Pr[\mathsf{Ext}_i^{\mathcal{A}_i} \neq \bot] \geq \frac{\Pr[\mathcal{A}_i(C_i,\ldots,C_\mu) \neq \bot] - \kappa_i}{1-\kappa_i}$ holds for all $\mathcal{A}_i$, then $\Pr[\mathsf{Ext}_i^{\mathcal{A}} \neq \bot] \geq \frac{\Pr[\mathcal{A}(C_1,\ldots,C_\mu) \neq \bot] - \kappa_i}{1-\kappa_i}$, where $C_j \leftarrow_R \mathfrak{C}_j$. This follows by linearity of the expectation.

8.2 Worked Examples of Sequential Compositions

(Tree-)Special Soundness We want to mimic, in our generalised setting, the standard sequential composition of three extractors Ext_1, Ext_2, Ext_3 for 2-special soundness in 3 challenge rounds. Here, Ext_i is an extractor which finds 2 correlated transcripts in challenge round i. First note that usually, Ext_i is only defined for $\mathfrak{C}_i$. We need to define it for $\mathfrak{C} = \mathfrak{C}_1 \times \mathfrak{C}_2 \times \mathfrak{C}_3$.

It is instructive to consider tree-special soundness [BCC+16] here: Indeed, we can view extractor Ext_3 as outputting a $(1,1,2)$-tree, Ext_2 a $(1,2,1)$-tree, and Ext_1 a $(2,1,1)$-tree. This is illustrated Fig. 2. Intuitively, there are two steps of "extensions" made to the typical extractors for (tree-)special soundness. Concretely, for Ext_2:

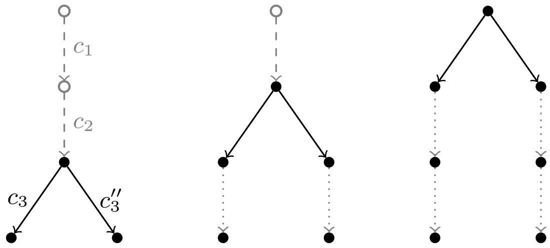

Fig. 2. 2-special sound tree extraction in 3 challenge rounds. From left to right, the trees represent the output tree structure of Ext_3, Ext_2, Ext_1. Dashed edges illustrate an extension. Dotted edges illustrate a difference to extractors in standard sequential composition, where the extractor does not choose (or know of) this challenge.

- In the standard sequential composition, Ext_2 would operate on adversaries $\mathcal{A}_3 \colon \mathfrak{C}_2 \to \{0,1\}^* \cup \{\bot\}$, because $\mathsf{Ext}_3^{\mathcal{A}}$ would be defined over $\mathfrak{C}_1 \times \mathfrak{C}_2$ (i.e. $\mathfrak{C}_3$ is not exposed anymore). However, with generalised sequential composition, $\mathsf{Ext}_3^{\mathcal{A}}$ is again defined over all of $\mathfrak{C}$. In particular, Ext_2 must choose c_3 (and c_1, which we handle later). By definition, we let Ext_2 pick c_3 uniformly from $\mathfrak{C}_3$ in every query. This extends Ext_2 to $\mathfrak{C}_2 \times \mathfrak{C}_3$.
- By our convention, we need to further extend Ext_2 to all of $\mathfrak{C}$, which is done as in Remark 8.1, namely, $\mathsf{Ext}_2(c_1, c_2, c_3)$ 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 10, we encounter the slightly more complicated case, where Ext_2 actually pick 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 Ext_i k-special soundness in the the i-th round as an extractor for $(1, \ldots, 1, k, 1, \ldots, 1)$-special soundness over the complete challenge space $\mathfrak{C}$.

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

By Remark 4.5, the definition of $\mathsf{challof}_{\mathcal{A}_2}(\,\cdot\,)$ for $\mathcal{A}_2(c) \coloneqq \mathsf{Ext}_{\text{2-ss}}^{\mathcal{A}}(c)$ 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 Ext, 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. 3. 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 $\mathsf{Ext}_{\text{2-ss}}$ 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 standard sequential composition (cf. Section 8.2), 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, is 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 $c^i_\ell \neq c^j_\ell$.

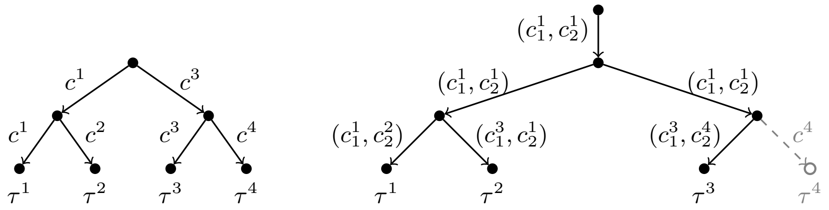

Fig. 3. Self-composition (left) and component-wise special soundness (right): The graph illustrates the tree structure in which the transcripts are arranged. *Left:* The lower depth-2 edges are labelled with $\mathsf{challof}_{\mathcal{A}}(\tau^i)$ for the i-th leaf. The upper depth-1 edges are labelled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

Component-wise k-special soundness (CWSS) In this example, we show how component-wise special soundness [FMN23] of a Σ-protocol can be modelled via generalized sequential composition. We consider $(2, 2)$-component-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 $c^2_1 = c^1_1$ and $c^2_2 = c^1_2$. That is, c^1 is the "root challenge" and c^2 (resp. c^3) differs in exactly one coordinate. Let produce such transcripts via generalized sequential composition of 3 extractors:

- $\mathsf{Ext}_{\text{1-ss}}(c)$: 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 $c = c^1 \in \mathfrak{C}$.
- $\mathsf{Ext}_{\text{2-ss},\mathfrak{C}_1}((c_1, c_2))$, 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 9.2)
- $\mathsf{Ext}_{\text{2-ss},\mathfrak{C}_2}$ is analogous to $\mathsf{Ext}_{\text{2-ss},\mathfrak{C}_1}((c_1, c_2))$, but is $\mathfrak{C}_1$-fixing.

Consider the composition $\mathcal{B}_3(c) = \mathsf{Ext}_{\text{2-ss},\mathfrak{C}_1}\mathcal{A}(c)$, then $\mathcal{B}_2(c) = \mathsf{Ext}^{\mathcal{B}_3}_{\text{2-ss},\mathfrak{C}_1}(c)$, then $\mathcal{B}_1(c) = \mathsf{Ext}^{\mathcal{B}_2}_{\text{1-ss}}(c)$. 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) satisfy the requirements of CWSS.

Remark 8.2. Compared to tailor-made CWSS extractors [FMN23], we observe a few differences: Firstly, an additional transcript τ^4 is generated, which is unneeded. Secondly, a negative consequence of the previous point, the run-time of this composition $\approx 2^\ell$ for ℓ coordinates, instead of $2\ell - 1$ as in [FMN23]. Thirdly, as a positive consequence, the knowledge error is $1 - \prod_{i=1}^{\ell}(1 - \frac{1}{|\mathfrak{C}_i|})$ which better than $\sum_{i=1}^{\ell} \frac{1}{|\mathfrak{C}_i|}$ as in [FMN23], especially for small $\mathfrak{C}$.

9 Sequential Composition of Abstract Extractors

In this section, we take an abstract approach to knowledge extraction and the composition of extractors. We stress that we *compose extractors*, not protocols.

9.1 Introduction to Sequential Composition

The usual sequential composition operation for extractors is strongly tied to protocol composition. It considers a challenge space $\mathfrak{C} = \mathfrak{C}_1 \times \ldots \times \mathfrak{C}_\mu$ which is the product of challenge sets, where $\mathfrak{C}_i$ is the i-th round challenge set of a protocol. Given extractors $\mathsf{Ext}_1, \ldots, \mathsf{Ext}_\mu$, where Ext_i handles an abstract adversary $\mathcal{A}_i \colon \mathfrak{C}_i \to \{0,1\}^* \cup \{\bot\}$, they are composed sequentially as follows:

- $\mathcal{B}_\mu = \mathcal{A}$ is the μ-round adversary.
- $\mathcal{B}_i(c_1, \ldots, c_i) = \mathsf{Ext}_{i+1}^{\mathcal{B}_{i+1}(c_1,\ldots,c_i,\,\cdot\,)}$ defines a i-round adversary where (recursively) the rounds after i are extracted.[24]
- $\mathcal{B}_0 =: \mathsf{Ext}^{\mathcal{A}}$ is the final composed extractor.

This is the natural choice of sequential composition, as it reflects the independence of the sequential rounds in a protocol. However, to properly exploit the possibilities of *adaptive* special soundness, namely to exclude challenges depending on the accepting transcripts which were found, we require something more flexible. For handling probabilistic testing during extraction, we saw in the examples of Section 6 that it is important to zoom in on challenges which contradict the soundness of said probabilistic tests. However, in a multi-round setting, the test's challenge may be *spread over multiple rounds*. For example, the test may be a multi-variate Schwartz–Zippel variant, where in each of the μ rounds, one variable is chosen randomly, and only in the final round a check occurs. Indeed, this is what happens in our lattice-based folding analysis, cf. Section 10. So, for this kind argument, we essentially need to treat $\mathfrak{C} = \mathfrak{C}_1 \times \ldots \times \mathfrak{C}_\mu$ more like a single challenge set, instead of a sequence of challenge sets. More specifically, for the i-th extraction, not only $\mathfrak{C}_i$ but $\mathfrak{C}_i \times \ldots \times \mathfrak{C}_\mu$ is needed to define useful challenges. Unfortunately, the standard sequential composition is insufficient to capture the sketched example. Consider the following composition template of Ext_1, Ext_2 over $\mathfrak{C}_1 \times \mathfrak{C}_2$:

[24] For (adaptive) tree special soundness extractors, the output of $\mathcal{B}_i$ is w.l.o.g. a set of transcripts or $\bot$. We omit the formal details of stitching the input/output behaviour in this overview, see Section 9.2.

- Ext_1 learns enough transcripts to extract a *candidate* witness wit. If $\mathsf{wit} \in \varXi_{\mathsf{stmt}}$, then Ext_1 outputs wit. Otherwise, Ext_1 identifies a predicate $\varphi_{\mathsf{wit}} \colon \mathfrak{C}_1 \times \mathfrak{C}_2 \to \{0,1\}$ which depends on wit. Roughly, $\varphi_{\mathsf{wit}}(c_1, c_2) = 1$ if "honestly" executing the prover using wit makes the verifier accept. Intuitively, if $\varphi_{\mathsf{wit}}(c_1, c_2) = 1$, then the extractor may not learn anything useful from a transcript for (c_1, c_2), as it could produce an accepting transcript itself. Typically, finding a transcript with $\varphi_{\mathsf{wit}}(c_1, c_2) = 1$ will (eventually) ensure that Ext_1 breaks a hard relation. In the setting of Section 10, this means that a SIS break is encountered.
- The useful challenges set is now $\{(c_1, c_2) \mid \varphi_{\mathsf{wit}}(c_1, c_2) \neq 0\} \subseteq \mathfrak{C}_1 \times \mathfrak{C}_2$, i.e. those challenges where the extractor is guaranteed to learn something.

With standard sequential composition, Ext_1 only provides $\mathcal{A}(c_1, \cdot\,)$ for Ext_2, and cannot provide c_2, let alone φ_{wit} or wit This causes a problem: The probability that $\varphi_{\mathsf{wit}}(c_1, c_2) \neq 0$ is queried by Ext_2 is not 1, and can be quite small. Thus, Ext_2 may return an *accepting but useless* tree of transcripts. This behaviour interacts badly with the model of UCS-extractors: We must modify the UCS to search until $\varphi_{\mathsf{wit}}(c_1, c_2) = 1$ eventually holds (else, Ext_1 cannot extract from the transcripts), making the depth of the UCS infinite in the worst case.

Our approach is a judicious strengthening of the standard approach: We allow Ext_1 to specify the first challenge of Ext_2. That is, Ext_1 calls $\mathsf{Ext}_2^{\mathcal{A}(c_1, \cdot\,)}(c_2)$ where $\varphi(c_1, c_2) \neq 0$. This suffices for our application in Section 10. Moreover, we generalise sequential composition further, removing the requirement for a product structure on $\mathfrak{C}$ (corresponding to round structure in a protocol) entirely.

9.2 Generalised Sequential Composition

In this section, we define generalised sequential composition. One natural adaptation in our setting would consider extractors Ext_i which expect oracle access to adversaries $\mathcal{A} \colon \mathfrak{C}_i \times \ldots \times \mathfrak{C}_\mu \to \{0,1\}^* \cup \{\bot\}$. However, it will be convenient to have all oracles defined over the whole challenge set $\mathfrak{C} = \mathfrak{C}_1 \times \ldots \times \mathfrak{C}_\mu$, instead of $\mathfrak{C}_i \times \ldots \times \mathfrak{C}_\mu$. In other words, applying an extractor Ext to an adversary

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

yields a new adversary

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

over the same challenge space. This gives a more general composition result, which is also simpler to state. Thus, in the following, we consider extractors Ext_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). Recall that for abstract extractors (Definition 4.4), we specify via $(\mathsf{V}^{\mathsf{pre}}, \mathsf{V}^{\mathsf{post}})$-compatibility, the pre-condition Ext requires on oracle outputs and the post-condition it asserts on its own outputs.

Definition 9.1 (Generalised sequential composition). *Let* $(\mathsf{Ext}_i)_{i=1}^{\mu}$ *be abstract extractors over* $\mathfrak{C}$ *which are* $(\mathsf{V}_i^{\mathsf{pre}}, \mathsf{V}_i^{\mathsf{post}})$*-compatible. We call* $(\mathsf{Ext}_i)_{i=1}^{\mu}$

***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 $i = 1, \dots, \mu - 1$. For composable $(\mathsf{Ext}_i)_{i=1}^{\mu}$, we define their **(generalised) sequential composition** Ext as follows:*

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

9.3 Additional Preliminaries

We define $\mathfrak{C}'$-fixing extractors, which extends the notion of $\mathfrak{C}'$-fixing useful challenge structure to any extractor, and also serves to generalize challenge set extension from Remark 8.1.

Definition 9.2 ($\mathfrak{C}'$-fixing extractor). *Let Ext be an extractor over $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$. We say Ext is $\mathfrak{C}'$-fixing if for any execution of $\mathsf{Ext}^{\mathcal{A}}$, the queries which Ext makes to $\mathcal{A}$ lie in a subset $\{c'\} \times \mathfrak{C}''$ for some $c' \in \mathfrak{C}'$. In other words, the first query $c = (c', c'')$ to $\mathcal{A}$ fixes the $\mathfrak{C}'$-component and all other queries use this fixed c'.*

Remark 9.3 (Extractor families and challenge set extension). Let $(\mathsf{Ext}_{c'})_{c' \in \mathfrak{C}'}$ be a family of extractors over $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}'$. We extend $(\mathsf{Ext}_{c'})_{c' \in \mathfrak{C}'}$ to $\mathfrak{C}$ by having Ext pick $c' \leftarrow_R \mathfrak{C}'$ uniformly and run Ext_{c_1}. Then Ext is a $\mathfrak{C}'$-fixing extractor. Conversely, any $\mathfrak{C}'$-fixing extractor defines a family $(\mathsf{Ext}_{c'})_{c' \in \mathfrak{C}'}$ of extractors over $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$ in the obvious way. Extending families of extractors to $\mathfrak{C}$ generalises Remark 8.1, where a "constant family" was was considered, that is, where $\mathsf{Ext}_{c'} = \mathsf{Ext}'$ for all $c' \in \mathfrak{C}'$.

To exemplify how standard composition and definitions of extractors need to be adapted, we consider examples. For simplicity, we do not consider families of extractors.

For easy handling of expected run-time, we use the following notion, which is roughly adapted from [Klo21] but is well-known and used in various forms, especially when phrased as Rényi divergence.

Definition 9.4. *Let Ext be an extractor. We say that Ext has **run-time tightness** ρ, if for every adversary $\mathcal{A}$, we have*

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

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

Definition 9.5. *Let* Ext *be an extractor over challenge space* $\mathfrak{C}$*. We say* Ext *has* ***uniform initial challenge*** *if the first query* $\mathcal{A}(c)$ *of* Ext *always uses a uniformly random* $c \leftarrow_R \mathfrak{C}$ *and if* $\mathcal{A}(c) = \bot$ *for this first query, then* Ext *returns* $\bot$*.*[25]

More generally, let $(\mathsf{Ext}_{c'})_{c' \in \mathfrak{C}'}$ *be a family of extractors over* $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$*. We say* $(\mathsf{Ext}_{c'})_{c' \in \mathfrak{C}'}$ *has run-time tightness* ρ f *if every* $\mathsf{Ext}_{c'}$ *has (for* $c' \in \mathfrak{C}'$*).*

Remark 9.6 (Run-time tightness is preserved by extensions). Suppose $(\mathsf{Ext}_{c'})_{c' \in \mathfrak{C}'}$ is a family of extractors over $\mathfrak{C}''$ for $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}'$ with run-time tightness ρ. Consider its extension Ext to $\mathfrak{C}$ as in Remark 9.3m that is, Ext picks $c' \leftarrow_R \mathfrak{C}'$ and fixes it. Then Ext has run-time tightness ρ. (This follows, since the expectation over c' can be replaced by maximum, and each $\mathsf{Ext}_{c'}$ has run-time tightness ρ.)

9.4 Generic Success Bounds for Sequential Composition

In this section, we prove completely generic bounds for our generalised sequential composition. Despite our generalisation, the bounds coincide with the known bounds for standard sequential composition. For additional generality, we consider following definition.

Definition 9.7 (Knowledge error function). *Let* Ext *be an extractor over challenge space* $\mathfrak{C}$*, and let* $f\colon [0,1] \to \mathbb{R}$ *be monotonically increasing. We say* Ext *has* ***knowledge error function*** f *if, for every adversary* $\mathcal{A}$*, we have*

$$\Pr[\mathsf{Ext}^{\mathcal{A}} \neq \bot] \geq f(\Pr[\mathcal{A}(C) \neq \bot \mid C \leftarrow_R \mathfrak{C}])\,.$$

More generally, let $(\mathsf{Ext}_{c'})_{c' \in \mathfrak{C}'}$ *be a family of extractors over* $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$*. We say* f *is a* knowledge error function *of* $(\mathsf{Ext}_{c'})_{c' \in \mathfrak{C}'}$ *if* f *is a knowledge error function of* $\mathsf{Ext}_{c'}$ *for every* $c' \in \mathfrak{C}'$

Following remark asserts that we can, without loss of generality, assume that an extractor is extended to all of $\mathfrak{C}$.

Lemma 9.8 (Convex knowledge error functions is preserved by extensions). *Suppose* $(\mathsf{Ext}_{c'})_{c' \in \mathfrak{C}'}$ *is a family of extractors over* $\mathfrak{C}''$ *for* $\mathfrak{C} = \mathfrak{C}' \times \mathfrak{C}''$ *with knowledge error function* f*. Extend* Ext *to* $\mathfrak{C}$ *as in Remark 9.3, that is,* Ext *picks* $c' \leftarrow_R \mathfrak{C}'$ *and fixes it. Then this extended* Ext *has convex knowledge error function* f *(as an extractor over* $\mathfrak{C}$*). Namely, we have*

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

where C'*,* C''*,* C *are uniformly random in* $\mathfrak{C}'$*,* $\mathfrak{C}''$*, and* $\mathfrak{C}$*, respectively. In the chain of (in)equalities, we use that* f *is convex so we can use Jensen's inequality to swap expectation and* f*.*

[25] Since we consider *ordinary* extractors (Definition 4.4), the second property always holds.

Lemma 9.9. *Let* Ext_1 *and* Ext_2 *be extractors over challenge space* $\mathfrak{C}$ *with knowledge error functions* f_1*,* f_2*. Suppose* Ext_2 *has uniform initial challenge. Let* Ext *be the (generalised) sequential composition of* Ext_1 *and* Ext_2*, that is* $\mathsf{Ext}^{\mathcal{A}} := \mathsf{Ext}_1^{\mathsf{Ext}_2^{\mathcal{A}}}$*. Then* Ext *has knowledge error* $(f_1 \circ f_2)(x) = f_1(f_2(x))$*.*

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

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

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

By induction, any number of extractors can be sequentially composed.

Example 9.10. In Corollary 5.8 we show that our UCS extractors have knowledge error function $f(x) = \frac{x-\kappa(\mathcal{U})}{1-\kappa(\mathcal{U})}$, which is also the knowledge error function of [ACK21] and optimal for k-special soundness. Another natural choice is $g(x) = x - \kappa(\mathcal{U})$, i.e. the denominator is one. In both cases, Lemma 9.9 recovers the known success bounds for the sequential compositions.

Finally, we note that the knowledge error 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.

10 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.

The outline of the section goes as follows. First, we recall some preliminaries of algebraic number theory which will be used in Sections 10 and 11. Then, we construct a (nested) adaptive witness extractor $\mathcal{W}$ for the Bulletproofs protocol. Next, we prove that the Bulletproofs protocol is $\mathcal{W}$-adaptive special sound, and that the knowledge error $\kappa(\mathcal{U}_{\mathcal{W}})$ of the UCS $\mathcal{U}_{\mathcal{W}}$ induced by $\mathcal{W}$ is negligible. Finally, calling the results from the previous sections, we arrive at the conclusion that the Bulletproofs protocol is $\kappa(\mathcal{U}_{\mathcal{W}})$-knowledge sound.

10.1 Algebraic Number Theory Background

Let $\mathcal{K}$ be a Galois number field and $\mathcal{R}$ 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 $\mathcal{R}$ 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 $\mathcal{R}$ an imaginary quadratic ring. Let φ denote the degree of $\mathcal{K}/\mathbb{Q}$. When an element $a \in \mathcal{R}$, 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[26] 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 $\|\mathbf{a}\| \coloneqq \|(\sigma(a_1), \ldots, \sigma(a_m))\|$.

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 $a, b \in \mathcal{R}$, 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. $\langle a, b \rangle = \langle 1 \rangle = \mathcal{R}$. We say that a vector $\mathbf{z} \in \mathcal{R}^m$ is co-prime with an element $d \in \mathcal{R}$ if $\gcd(z_1, \ldots, z_m, d) = \mathcal{R}$. By a slight abuse of notation, we also write $\gcd(\mathbf{z}, d) = \mathcal{R}$.

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

10.2 Protocol Description

In Fig. 5 we recall the lattice analogue [BLNS20; AL21; ACK21] of the Bulletproofs protocol [BCG+17; BBB+18] for proving ISIS relations

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

described in a way that will be convenient for the subsequent soundness analysis. The protocol uses the "folding" subroutines defined in Fig. 4 and is summarised as follows. On input an ISIS statement $(\mathbf{A}, \mathbf{y}) \in \mathcal{R}_q^{n \times m} \times \mathcal{R}_q^n$ and a witness $\mathbf{x} \in \mathcal{R}^m$, where for simplicity we assume that $m = 2^\mu$ 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 10.1 (Completeness). *The lattice Bulletproofs protocol with parameters $(\mathcal{R}, n, m = 2^\mu, q, \alpha, \beta)$ described in Fig. 5 is complete for the relation $\Xi^{\mathsf{ISIS}}_{\mathcal{R},n,m,q,\alpha}$.*

The proof of completeness is trivial and thus omitted.

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

Shorthand notation:

- For $0 \leq i \leq \mu$, $\mathbf{C}_i = (C_1, \ldots, C_i)$ 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 A: For $\mathbf{A} \in \mathcal{R}_q^{n \times m}$, define $f_{\mathbf{A}}(\mathbf{C}_\mu) \in \mathcal{R}_q^n[\mathbf{C}_\mu]$ recursively as follows:

$$\begin{aligned} f_{\mathbf{A},0} &\coloneqq \mathbf{A} \in \mathcal{R}_q^{n \times m} \\ f_{\mathbf{A},i}(\mathbf{C}_i) &\coloneqq f_{\mathbf{A},i-1}[1 : m/2^i] \cdot C_i + f_{\mathbf{A},i-1}[m/2^i + 1 : m] \in \mathcal{R}_q^{n \times m/2^i}[\mathbf{C}_i] \\ f_{\mathbf{A}}(\mathbf{C}_\mu) &\coloneqq f_{\mathbf{A},\mu}(\mathbf{C}_\mu) \in \mathcal{R}_q^n[\mathbf{C}_\mu] \end{aligned}$$

Since the subscript i of $f_{\mathbf{A},i}$ is fully determined by its input as $i = |\mathbf{C}_i|$, we often drop the subscript and just write $f_{\mathbf{A}}$.

Folding x: For $\mathbf{x} \in \mathcal{K}^m$, define $g_{\mathbf{x}}(\mathbf{C}_\mu) \in \mathcal{K}[\mathbf{C}_\mu]$ as follows:

$$\begin{aligned} g_{\mathbf{x},0} &\coloneqq \mathbf{x} \in \mathcal{K}^m \\ g_{\mathbf{x},i}(\mathbf{C}_i) &\coloneqq g_{\mathbf{x},i-1}[1 : m/2^i] + g_{\mathbf{x},i-1}[m/2^i + 1 : m] \cdot C_i \in \mathcal{K}^{m/2^i}[\mathbf{C}_i] \\ g_{\mathbf{x}}(\mathbf{C}_\mu) &\coloneqq g_{\mathbf{x},\mu}(\mathbf{C}_\mu) \in \mathcal{K}[\mathbf{C}_\mu] \end{aligned}$$

Since the subscript i of $g_{\mathbf{x},i}$ is fully determined by its input as $i = |\mathbf{C}_i|$, we often drop the subscript and just write $g_{\mathbf{x}}$.

Fig. 4. Folding subroutines.

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

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

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 $h_i(C) \coloneqq f_{\mathbf{A}_{i-1}}(C) \cdot g_{\mathbf{x}_{i-1}}(C) \in \mathcal{R}_q^n[C]$ to V.
   (b) V: Assert that $h_i \in \mathcal{R}_q^n[C]$ is a quadratic polynomial.
   (c) V: Assert that the coefficient of C in h_i is $\mathbf{y}_{i-1}$.
   (d) V: Send $c_i \leftarrow_R \mathfrak{C}$ 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 $((\mathbf{A}_\mu, \mathbf{y}_\mu), \mathbf{x}_\mu) \in \Xi^{\mathsf{ISIS}}_{\mathcal{R},n,1,q,\alpha\beta^\mu}$

Fig. 5. The lattice Bulletproofs protocol.

10.3 Soundness Claim

We would like to construct an adaptive witness extractor $\mathcal{W}$ and show that the lattice Bulletproofs protocol is $\mathcal{W}$-adaptive special sound for the relation

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

defined below:

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

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

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

Intuitively, being knowledge sound for the relation $\varXi$ 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 $\mathbf{x} \in \mathcal{R}^m$, and a small slack $s \in \mathcal{R}$ coprime with $\mathbf{x}$, such that $\mathbf{A}\mathbf{x} = \mathbf{y}s \bmod q$.
- (Knowledge of SIS.) A short solution $\mathbf{x} \in \mathcal{R}^m$ such that $\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$.

Remark 10.2. The requirement that $\langle x_1,\ldots,x_m,s\rangle = \mathcal{R}$ in $\varXi^{\sim\mathsf{ISIS}}_{\mathcal{R},n,m,q,\alpha,\delta}$ can be lifted by proving a more general multilinear composite Schwartz-Zippel lemma Theorem 11.1 which drops a similar 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 $\mathcal{R}$, 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 $\mathcal{R}$ 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

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

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

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

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

Lemma 10.3. *For $\mathcal{R} = \mathbb{Z}$ and $\delta \geq \nu$, $\Xi^{\perp}_{\mathcal{R},\delta,\nu} = \emptyset$. For $\mathcal{R}$ being the ring of integers of $\mathbb{Q}(\sqrt{\Delta})$, where $\Delta < 0$ is a square-free integer, and $\delta^2 \geq \nu$, $\Xi^{\perp}_{\mathcal{R},\delta,\nu} = \emptyset$.*

Proof. The case $\mathcal{R} = \mathbb{Z}$ is trivial. For the case where $\mathcal{R}$ 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. $\mathcal{R}$ is a cyclotomic ring, the inclusion of $\Xi^{\perp}_{\mathcal{R},\delta,\nu}$ might make soundness for the relation Ξ formally trivial, since a witness extractor $\mathcal{W}$ might always output a witness of the failure relation $\Xi^{\perp}_{\mathcal{R},\delta,\nu}$. 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 $\mathcal{R}$ in general. In more detail, during witness extraction, the extractor $\mathcal{W}$ needs to reduce a vector of fractions $\mathbf{x}/s$ where $\mathbf{x} \in \mathcal{R}^m$ and $s \in \mathcal{R}$ to $\mathbf{x}'/s' = \mathbf{x}/s$, such that $\gcd(\mathbf{x}', s') = \mathcal{R}$, $\|\mathbf{x}'\|$ 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 $s' \cdot \frac{\langle \mathbf{x} \rangle}{\langle s \rangle} \subset \mathcal{R}$, and define $\mathbf{x}' := s' \cdot \frac{\mathbf{x}}{s} \in \mathcal{R}^m$. 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 10.4. For the special case where $\mathcal{R} = \mathbb{Z}$ or an imaginary quadratic ring, we show that O can be efficiently instantiated.

10.4 Adaptive Witness Extractor for Lattice Bulletproofs

For any fixed statement $(\mathbf{A}, \mathbf{y})$, we construct a depth-μ adaptive witness extractor $\mathcal{W}^{\mathsf{O}}_{\mathbf{A},\mathbf{y}} = (\mathcal{W}^{\mathsf{O}}_{\mathbf{A},\mathbf{y},i})_{0 \leq i < \mu}$ for the lattice Bulletproofs protocol. For the sake of readability, we suppress the subscript $(\mathbf{A}, \mathbf{y})$ and write $\mathcal{W}^{\mathsf{O}}_i$ instead of $\mathcal{W}^{\mathsf{O}}_{\mathbf{A},\mathbf{y},i}$.

In the following, we first establish some notation recurring in this section and then proceed to define the adaptive witness extractor $\mathcal{W}^{\mathsf{O}}$. The analysis of the $\mathcal{W}^{\mathsf{O}}$-adaptive special soundness extractor and the knowledge error of $\mathcal{W}^{\mathsf{O}}$ will be given in Section 10.5.

Terminologies and Notation. Fix a statement $(\mathbf{A}, \mathbf{y}) \in \mathcal{R}_q^{n\times m} \times \mathcal{R}_q^n$, where $m = 2^\mu$. 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).

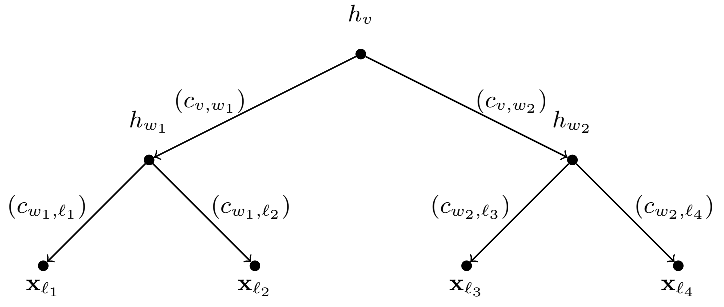

Fig. 6. Depiction of a (sub)tree of transcripts $\mathfrak{T}_v$.

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. The tuple $(\tau_v, \mathfrak{T}_v)$ is a compact representation of a sequence of transcripts obtained from all root-to-leaf paths for all leaves under 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 $(\tau_v, \mathfrak{T}_v)$.
- (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:

$$\delta \coloneqq 2^3\beta^3, \qquad \alpha_\mu \coloneqq \alpha\beta^\mu, \qquad m_\mu \coloneqq m.$$

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

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

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 $(\mathbf{x}, s) \in \mathcal{R}^k \times \mathcal{R}$ for some $k = \mathsf{poly}(\lambda)$, the tuple $(\hat{\mathbf{x}}, \hat{s}) \leftarrow \mathsf{O}(\mathbf{x}, s)$ satisfies the following:

$$\hat{\mathbf{x}} \cdot s = \mathbf{x} \cdot \hat{s}, \quad \langle \hat{x}_1, \dots, \hat{x}_k, \hat{s} \rangle = \mathcal{R}, \quad \|\hat{\mathbf{x}}\| \leq \gamma \cdot \|\mathbf{x}\|, \quad \text{and} \quad \|\hat{s}\| \leq \|s\|.$$

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 10.4. *If $\mathcal{R} = \mathbb{Z}$ or $\mathcal{R}$ 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 $\mathcal{R} = \mathbb{Z}$ (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 $\hat{\mathbf{x}} \coloneqq \mathbf{x}/d$ and $\hat{s} \coloneqq s/d$. 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 $\mathcal{R}$ being its ring of integers. It is easy to check that, for any integral element $d \in \mathcal{R}$, it holds that $\delta \coloneqq \min_{\sigma_i \in \mathrm{Gal}(\mathcal{K}/\mathbb{Q})}(|\sigma_i(d)|) \geq 1$. Therefore, $\|\hat{\mathbf{x}}\| \leq \|\mathbf{x}\|/\delta \leq \|\mathbf{x}\|$ and $\|\hat{s}\| \leq \|s\|/\delta \leq \|s\|$. □

Extraction Algorithms. In Fig. 7 we define a sequence of algorithms $(\mathcal{W}_i^{\mathsf{O}})_{i=0}^{\mu-1}$ which we will later show to be a (nested) adaptive witness extractor. The extractor $\mathcal{W}_i^{\mathsf{O}}$ expects as input a sequence of transcripts represented by $(\tau_v, \mathfrak{T}_v)$ for some node v in $\mathfrak{T}$ of depth i, and produces a candidate witness $(\mathbf{x}_v, s_v)$ for stmt_v. Formally, we write

$$(\mathbf{x}_v, s_v) \leftarrow \mathcal{W}_i^{\mathsf{O}}(\tau_v, \mathfrak{T}_v).$$

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 10.5 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 10.5. *Let $\mathbf{A}_i \in \mathcal{R}_q^{n\times m_i}$ and $h_v \in \mathcal{R}_q^n[C]$ be a quadratic polynomial. If $(c_j, \mathbf{x}_j, s_j)_{j=1}^3 \in (\mathcal{R} \times \mathcal{R}^{m_{i+1}} \times \mathcal{R})^3$ satisfies*

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

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

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

10.5 Knowledge Soundness Analysis

We prove two main results regarding the adaptive witness extractor $\mathcal{W}$ constructed in Section 10.4. The first is that the lattice-based Bulletproofs protocol is $\mathcal{W}$-adaptive special sound. The second is that the knowledge error of $\mathcal{U}_{\mathcal{W}}$ 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 5 and 9, we conclude that the lattice-based Bulletproofs protocol is knowledge-sound with the same knowledge error.

Theorem 10.6 (Adaptive Special Soundness). *Let $\mathcal{R}, n, m, q, \alpha_0, \alpha_0', \gamma, \delta, \nu$ be parameters as defined in Section 10.4. Assuming the existence of a γ-coprime fraction oracle O, for the adaptive witness extractor $\mathcal{W}^{\mathsf{O}}$ constructed in Section 10.4, the lattice Bulletproofs protocol is $\mathcal{W}^{\mathsf{O}}$-adaptive special sound for the relation*

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

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

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

in any of the following settings:

```
$\mathcal{W}_i^{\mathsf{O}}(\tau_v, \mathfrak{T}_v)$, $0 \leq i < \mu$:

 1. If $(\tau_v, \mathfrak{T}_v)$ contains a rejecting transcript, return $(\emptyset, \perp)$.
 2. If $(\tau_v, \mathfrak{T}_v)$ is an empty tree, return $(\mathfrak{C}^{\mu-i}, \perp)$.
 3. Assert that $\mathsf{depth}(\mathfrak{T}_v) = \mu - i$.
 4. Compute $\mathsf{stmt}_v = (\mathbf{A}_v, \mathbf{y}_v)$ from $\mathsf{stmt}$ and $(\tau_v, \mathfrak{T}_v)$.
 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 $(\mathbf{x}_w, s_w) := \mathcal{W}_{i+1}^{\mathsf{O}}(\tau_w, \mathfrak{T}_w) \in \mathcal{R}^{m_{i+1}} \times \mathcal{R}$.
    (c) If $((\mathbf{A}_w, \mathbf{y}_w), (\mathbf{x}_w, s_w)) \in \Xi^{\perp}_{\mathcal{R},\delta,\nu}$ then return $(\emptyset, (\perp, s_w))$.
    (d) If $((\mathbf{A}_w, \mathbf{y}_w), (\mathbf{x}_w, s_w)) \in \Xi^{\mathsf{SIS}}_{\mathcal{R},n,m_{i+1},q,\alpha_{i+1}}$ 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 $(\mathsf{Uncharted}(\mathfrak{T}_v), \perp)$.
 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:
            $$b_0 := (s_v^* \cdot h_v(C) \overset{?}{=} f_{\mathbf{A}_v}(C) \cdot g_{\mathbf{x}_v^*}(C)),$$
            $$b_1 := (\mathcal{N}(s_v^\dagger) \overset{?}{\leq} \nu).$$
    – If $b_0 = 0$, return $(\mathsf{Uncharted}(\mathfrak{T}_v), \perp)$.
    – If $b_1 = 0$, return $(\mathsf{Uncharted}(\mathfrak{T}_v) \cap \mathfrak{C}_{\mathsf{SZ},\mu-i}(\mathbf{x}_v^\dagger, s_v^\dagger), \perp)$ where
            $$\mathfrak{C}_{\mathsf{SZ},\mu-i}(\mathbf{x}_v^\dagger, s_v^\dagger) := \left\{ \mathbf{c} \in \mathfrak{C}^{\mu-i} : g_{\mathbf{x}_v^\dagger / s_v^\dagger}(\mathbf{c}) \notin \mathcal{R} \right\}.$$
    – 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 $(\emptyset, \perp)$.


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

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

Fig. 7. Adaptive witness extractor for lattice Bulletproofs.

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

Proof. Given the first part of the theorem, the second part follows immediately from Lemmas 10.3 and 10.4. 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 sequence of accepting $\mathfrak{T}$ of accepting transcripts with $\mathfrak{T} \in \mathsf{Chains}(\mathcal{U}_{\mathcal{W},\mathsf{stmt}})$, $\mathcal{W}^{\mathsf{O}}_{\mathsf{stmt}}(\mathfrak{T})$ outputs a witness $\mathsf{wit} \in \Xi(\mathsf{stmt})$.

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

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

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

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

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 $0 \leq i < \mu$, 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
$$((\mathbf{A}_w, \mathbf{y}_w), (\mathbf{x}_w, s_w)) \in \Xi^{\mathsf{SIS}}_{\mathcal{R},n,m_{i+1},q,\alpha'_{i+1}} \cup \Xi^{\perp}_{\mathcal{R},\delta,\nu}.$$
2. For all $w \in \mathsf{children}(v)$, $((\mathbf{A}_w, \mathbf{y}_w), (\mathbf{x}_w, s_w)) \in \Xi^{\sim\mathsf{ISIS}}_{\mathcal{R},n,m_{i+1},q,\alpha_{i+1},\delta}$.

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

We analyse the behaviour of $\mathcal{W}^{\mathsf{O}}_i(\tau_v, \mathfrak{T}_v)$ 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

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

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

If $((\mathbf{A}_w, \mathbf{y}_w), (\mathbf{x}_w, s_w)) \in \Xi^{\mathsf{SIS}}_{\mathcal{R},n,m_{i+1},q,\alpha'_{i+1}}$, then $\mathcal{W}^{\mathsf{O}}_i(\tau_v, \mathfrak{T}_v)$ 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,

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

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

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

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

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

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

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

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

then Case 1 would have been triggered at v.

Define the following predicates:

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

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 $|\mathsf{children}(v)| \ge 4$.)
b) $|\mathsf{children}(v)| = 4$ and $b_2 = 1$.
c) $|\mathsf{children}(v)| \ge 4$, $b_0 = 0$, and $b_2 = 0$.
d) $|\mathsf{children}(v)| \ge 4$, $b_1 = 0$, 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. $\Diamond$

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

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

The witness extractor $\mathcal{W}^{\mathsf{O}}_i(\tau_v, \mathfrak{T}_v)$ would return $(\mathbf{x}_v, s_v) := (\mathbf{x}^\dagger_v, s^\dagger_v)$.

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

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

By the guarantee of O, we have

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

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

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

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

$$((\mathbf{A}_v, \mathbf{y}_v), (\bot, s_v^\dagger)) \in \Xi^{\bot}_{\mathcal{R},\delta,\nu}. \qquad \lozenge$$

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

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

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

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

for all $C \in \mathcal{R}$, 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

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

$\mathcal{W}_i^{\mathsf{O}}(\tau_v, \mathfrak{T}_v)$ 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,

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

and thus the invariant holds at node v. ◊

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

$$\mathbf{X}_v^* \begin{pmatrix} 1 \\ c_{(v,w)} \\ c_{(v,w)}^2 \end{pmatrix} s_w = \begin{pmatrix} \mathbf{x}_w c_{(v,w)} \\ \mathbf{x}_w \end{pmatrix} s_w^2. \tag{16}$$

We argue that the above constraints are impossible.

Note that Eq. (16) 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

$$\mathbf{x}_w \cdot s_v^* = \begin{pmatrix} \mathbf{x}_{v,00}^* & \mathbf{x}_{v,01}^* & \mathbf{x}_{v,02}^* \end{pmatrix} \begin{pmatrix} 1 \\ c_{v,w} \\ c_{v,w}^2 \end{pmatrix} s_w = \begin{pmatrix} \mathbf{x}_{v,10}^* & \mathbf{x}_{v,11}^* & \mathbf{x}_{v,12}^* \end{pmatrix} \begin{pmatrix} c_{v,w} \\ c_{v,w}^2 \\ c_{v,w}^3 \end{pmatrix} s_w,$$

$$\mathbf{x}_{v,00}^* + (\mathbf{x}_{v,01}^* - \mathbf{x}_{v,10}^*) c_{v,w} + (\mathbf{x}_{v,02}^* - \mathbf{x}_{v,11}^*) c_{v,w}^2 - \mathbf{x}_{v,12}^* c_{v,w}^3 = 0, \qquad (17)$$

where the second equality holds by cancelling s_w on both sides, which is permitted since there are no zero-divisors in $\mathcal{R}$. Note that Eq. (17) is cubic polynomial over $\mathcal{R}$ with 4 distinct roots $(c_{v,w_j})_{j=1}^4$. Since there are no zero-divisors in $\mathcal{R}$, 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 $\mathbf{x}_{v,0}^*, \mathbf{x}_{v,1}^* \in \mathcal{R}^{m_{i+1}}$. 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. (16) 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 $\mathfrak{T} \in \mathsf{Chains}(\mathcal{U}_{\mathcal{W}_{\mathsf{stmt}}})$ 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

$$g_{\mathbf{x}_v^\dagger/s_v^\dagger}(\mathbf{c}_{v,\ell_4}) \notin \mathcal{R}.$$

Applying the argument in Case 2c, we can deduce from Eq. (16) 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

$$\mathbf{x}_v^* := \begin{pmatrix} \mathbf{x}_{v,0}^* \\ \mathbf{x}_{v,1}^* \end{pmatrix},$$

we can substitute $\mathbf{X}_v^*$ back into Eq. (16) and obtain, for each $w \in \mathsf{children}(v)$,

$$g_{\mathbf{x}_v^*}(c_{v,w}) s_w = \mathbf{x}_{v,0}^* + \mathbf{x}_{v,1}^* c_{v,w} = \mathbf{x}_w s_v^*,$$
$$g_{\mathbf{x}_v^\dagger/s_v^\dagger}(c_{v,w}) = g_{\mathbf{x}_v^*/s_v^*}(c_{v,w}) = \mathbf{x}_w/s_w$$

where for the first equality we again relied on the lack of zero-divisor in $\mathcal{R}$ to divide out $c_{v,w}$.

Since $((\mathbf{A}_u, \mathbf{y}_u), (\mathbf{x}_u, s_u)) \in \varXi^{\sim\mathsf{ISIS}}_{\mathcal{R},n,m_j,q,\alpha_j,\delta}$ for any node $u \in \mathfrak{T}_v \setminus \{v\}$, Eq. (16) 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,

$$((\mathbf{A}_{\ell_4}, \mathbf{y}_{\ell_4}), (\mathbf{x}_{\ell_4}, 1)) \in \varXi^{\sim\mathsf{ISIS}}_{\mathcal{R},n,1,q,\alpha_\mu,\delta_\mu}.$$

which in particular implies $g_{\mathbf{x}_v^\dagger/s_v^\dagger}(\mathbf{c}_{v,\ell_4}) = \mathbf{x}_{\ell_4} \in \mathcal{R}$, a contradiction. $\diamondsuit$ $\square$

Remark 10.7. In the above analysis, for $\mathcal{R}$ other than $\mathbb{Z}$ or imaginary quadratic rings, such as cyclotomic rings, the extractor $\mathcal{W}$ may output a witness of the failure relation $\Xi^{\perp}_{\mathcal{R},\delta,\nu}$. This happens if and only if $\mathcal{W}$ encounters an element $s^{\dagger}_{v}$ with $\mathcal{N}(s^{\dagger}_{v}) \leq \nu$ but $\|s^{\dagger}_{v}\| > \delta$. Note that it is always guaranteed that $\|s^{\dagger}_{v}\| \leq \delta$. Consider the set

$$\frac{|\{s^{\dagger}_{v} \in \mathcal{R} : \mathcal{N}(s) \leq \nu \wedge \delta < \|s\| \leq \delta^4\}|}{|\{c \in \mathcal{R} : \|c\| \leq \beta - 1\}|}$$

which compares the number of "bad" denominators $s^{\dagger}_{v}$, 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 $s^{\dagger}_{v}$ 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 11.11, we have that the ratio is at most $\mathsf{negl}(\lambda)$ for cyclotomic rings. We therefore heuristically conjecture that the probability of $\mathcal{W}_i$ finding a witness of $\Xi^{\perp}_{\mathcal{R},\delta,\nu}$ is negligible.

We next analyse the knowledge error of $\mathcal{U}_{\mathcal{W}}$.

Lemma 10.8. *The induced UCS for $\mathcal{U} = \mathcal{U}_{\mathcal{W}}$ 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 $\varepsilon_{\mathsf{SZ}}$ is given in Corollary 11.9 or Corollary 11.10 depending on the choice of $\mathcal{R}$, and expected run-time bounded by $\frac{1}{1-\kappa(\mathcal{U})} \cdot 4^{\mu} = \frac{1}{1-\kappa(\mathcal{U})} \cdot m^2$.

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 $\kappa(\mathcal{U}_i(\tau_v, \mathfrak{T}_v) \mid \mathfrak{C}^{\mu-i})$, 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

$$\Pr[\mathbf{c} \notin \mathcal{U}_i(\tau_v, \mathfrak{T}_v)],$$

with probability taken over $\mathbf{c} \leftarrow_R \mathfrak{C}^{\mu-i}$ sampled uniformly at random, over all choices of accepting $(\tau_v, \mathfrak{T}_v)$ which is incomplete but consistent with $\mathcal{U}_i$ in the following sense:

- (Incompleteness.) $\mathcal{U}_i(\tau_v, \mathfrak{T}_v) \neq \emptyset$.
- (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 $(\tau_v, \mathfrak{T}_v)$ where v has $0 \leq k \leq 3$ children, since $\mathcal{U}_i(\tau_v, \mathfrak{T}_v) = \emptyset$ 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

$$\Pr[\mathbf{c} \notin \mathcal{U}_i(\tau_v, \mathfrak{T}_v)] \leq 2/|\mathfrak{C}|$$

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 $\mathcal{U}_i(\tau_v, \mathfrak{T}_v)$ in Case 4.1 is a subset of that in Case 4.2, and in Case 4.3 we have $\mathcal{U}_i(\tau_v, \mathfrak{T}_v) = \emptyset$. It thus suffices to focus on Case 4.2, where

$$\begin{aligned} \mathcal{U}_i(\tau_v, \mathfrak{T}_v) &= \mathsf{Uncharted}(\mathfrak{T}_v) \cap \mathfrak{C}_{\mathsf{SZ},\mu-i}(\mathbf{x}_v^\dagger, s_v^\dagger), \text{ where} \\ \mathfrak{C}_{\mathsf{SZ},\mu-i}(\mathbf{x}_v^\dagger, s_v^\dagger) &= \left\{ \mathbf{c} \in \mathfrak{C}^{\mu-i} : g_{\mathbf{x}_v^\dagger/s_v^\dagger}(\mathbf{c}) \notin \mathcal{R} \right\}. \end{aligned}$$

By the soundness of multilinear composite Schwartz-Zippel lemma (Theorem 11.1), we have

$$\Pr[g_{\mathbf{x}_v^\dagger/s_v^\dagger}(\mathbf{c}) \in \mathcal{R}] \leq \varepsilon_{\mathsf{SZ}}.$$

Then, by a union bound, we have

$$\begin{aligned} \Pr[\mathbf{c} \notin \mathcal{U}_i(\tau_v, \mathfrak{T}_v)] &\leq \Pr[\mathbf{c} \notin \mathsf{Uncharted}(\mathfrak{T}_v)] + \Pr[g_{\mathbf{x}_v^\dagger/s_v^\dagger}(\mathbf{c}) \notin \mathcal{R}] \\ &\leq \tfrac{3}{|\mathfrak{C}|} + \varepsilon_{\mathsf{SZ}}. \end{aligned}$$

This shows that $\kappa(\mathcal{U}_i \mid \mathfrak{C}^{\mu-i}) \leq 3/|\mathfrak{C}| + \varepsilon_{\mathsf{SZ}}$, as claimed.

By Theorem 7.6, we obtain that the overall knowledge error of the composition, i.e. the knowledge error for extracting a $(\mathcal{U}_1, \ldots, \mathcal{U}_\mu)$-tree is

$$\kappa(\mathcal{U}) = 1 - (1 - \tfrac{3}{|\mathfrak{C}|} + \varepsilon_{\mathsf{SZ}})^\mu \leq \mu \cdot (\tfrac{3}{|\mathfrak{C}|} + \varepsilon_{\mathsf{SZ}})$$

and the expected run-time of the extractor is at most $\frac{1}{1-\kappa(\mathcal{U})} \cdot 4^\mu$. This proves the claim. □

11 Multilinear Composite Schwartz-Zippel over Rings

We prove a natural generalisation of the so-called multilinear composite Schwartz-Zippel lemmas [BF23, Theorem 1 and 2].

11.1 Main Theorem

Theorem 11.1 (Generalisation of [BF23, Theorem 1]). *Let $\mathcal{R}$ be the ring of integers of a number field $\mathcal{K}$ of degree φ. Let $\mu, \ell \in \mathbb{N}$, $f \in \mathcal{R}[\mathbf{X}]$ be a μ-variate multilinear polynomial, $d \in \mathcal{R}$ with ideal factorisation $\langle d \rangle = \prod_{j=1}^{\ell} \mathfrak{p}_j^{r_j}$ satisfying $\mathsf{cont}(f) + \langle d \rangle = \langle 1 \rangle$, where $\mathsf{cont}(f)$ denotes the content of f, i.e. the ideal generated by the coefficients of f, and $\mathcal{C}_\beta \coloneqq \{x \in \mathcal{R} : \|x\| \leq \beta\}$. For $j \in [\ell]$, $k \in [\mu]$, let $Z_{j,k} \sim \mathsf{Geo}(1 - 1/\mathcal{N}(\mathfrak{p}_\mathsf{j}))$ be a random variable distributed geometrically with parameter $1 - 1/\mathcal{N}(\mathfrak{p}_\mathsf{j})$. Let $r \geq \rho(\langle d \rangle)$, the covering radius of the ideal lattice $\langle d \rangle$. Let $\gamma_{\beta,r}$ denote the ratio $\gamma_{\beta,r} \coloneqq \left(\frac{|\mathcal{C}_{\beta+2r}|}{|\mathcal{C}_\beta|} \right)$. It holds that*

$$\varepsilon_{\mathsf{SZ}} \coloneqq \Pr[f(\mathbf{x}) = 0 \bmod d \mid \mathbf{x} \leftarrow_R \mathcal{C}^\mu\,]$$

$$\leq \gamma_{\beta,r}^{\mu} \cdot \Pr\left[\sum_{k=1}^{\mu} \mathbf{Z}_k \geq \mathbf{r}\right]$$

where $\mathbf{Z}_k \coloneqq (Z_{1,k}, \ldots, Z_{\ell,k})$ *and* $\mathbf{r} \coloneqq (r_1, \ldots, r_\ell)$.

Remark 11.2. 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 11.1.

Proof. For given μ-linear f and $\mathbf{x}_i = (x_1, \ldots, x_i) \in \mathcal{R}^i$ with $1 \leq i \leq \mu$, write

$$f_i[\mathbf{x}_i](X_{i+1}, \ldots, X_\mu) \coloneqq f(x_1, \ldots, x_i, X_{i+1}, \ldots, X_\mu)$$

for the $(\mu - i)$-linear polynomial in $(X_{i+1}, \ldots, X_\mu)$.

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 $\mathfrak{p}_{\mathfrak{j}}$ in $\mathsf{cont}(f_k[\mathbf{x}_k])$ where $\mathbf{x}_k \leftarrow_R \mathcal{C}^k$ for $j \in [\ell]$. In other words, if $\mathsf{cont}(f_k[\mathbf{x}_k]) = \prod_{j=1}^{\ell} \mathfrak{p}_{\mathfrak{j}}{}^{y_j}$, 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 $f_0[] = f$ and $\mathsf{cont}(f) + \langle d \rangle = \langle 1 \rangle$. We see that, for any $k \in [\mu]$, the following events are equivalent:

$$f_k[\mathbf{x}_k] = 0 \bmod d \qquad\qquad \text{and} \qquad\qquad \mathbf{Y}_k \geq \mathbf{r}.$$

Define the shorthand $\gamma_{\beta,r} \coloneqq \frac{|\mathcal{C}_{\beta+2r}|}{|\mathcal{C}_\beta|}$. Consider the following induction hypothesis for $i \in [\mu]$:

$$\forall\, \mathbf{y} \in \mathbb{Z}^\ell,\ \Pr[\mathbf{Y}_i \geq \mathbf{y}] \leq \gamma_{\beta,r}^{i} \cdot \Pr\left[\sum_{k=1}^{i} \mathbf{Z}_k \geq \mathbf{y}\right]. \tag{18}$$

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 11.3, which states that, for any $i \in [\mu]$,

$$\forall\, \mathbf{y}, \mathbf{y}' \in \mathbb{Z}^\ell,\ \Pr[\mathbf{Y}_i \geq \mathbf{y} + \mathbf{Y}_{i-1} \mid \mathbf{Y}_{i-1} = \mathbf{y}'] \leq \gamma_{\beta,r} \cdot \Pr[\mathbf{Z}_i \geq \mathbf{y}].$$

Setting $i = 1$ in Lemma 11.3 and realising that $\mathbf{Y}_0 \equiv \mathbf{0}$ yield the base case $i = 1$ of the induction hypothesis (Eq. (18)).

Next, suppose that the induction hypothesis holds for some $i \in [\mu - 1]$, i.e.

$$\forall\, \mathbf{y} \in \mathbb{Z}^\ell,\ \Pr[\mathbf{Y}_i \geq \mathbf{y}] \leq \gamma_{\beta,r}^{i} \cdot \Pr\left[\sum_{k=1}^{i} \mathbf{Z}_k \geq \mathbf{y}\right].$$

We will prove the hypothesis for $i+1$, i.e.

$$\forall\, \mathbf{y} \in \mathbb{Z}^\ell,\ \Pr[\mathbf{Y}_{i+1} \geq \mathbf{y}] \leq \gamma_{\beta,r}^{i+1} \cdot \Pr\left[\sum_{k=1}^{i+1} \mathbf{Z}_k \geq \mathbf{y}\right].$$

Fix any $\mathbf{y}_{i+1} \in \mathbb{Z}^\ell$. We observe that

$$\begin{aligned}
&\Pr[\mathbf{Y}_{i+1} \geq \mathbf{y}_{i+1}] \\
=\ &\sum_{\mathbf{y}_i \in \mathbb{Z}^\ell} \Pr[\mathbf{Y}_{i+1} \geq \mathbf{y}_{i+1} - \mathbf{y}_i + \mathbf{Y}_i \mid \mathbf{Y}_i = \mathbf{y}_i] \cdot \Pr[\mathbf{Y}_i = \mathbf{y}_i] \\
\leq\ &\sum_{\mathbf{y}_i \in \mathbb{Z}^\ell} \gamma_{\beta,r} \cdot \Pr[\mathbf{Z}_{i+1} \geq \mathbf{y}_{i+1} - \mathbf{y}_i] \cdot \Pr[\mathbf{Y}_i = \mathbf{y}_i]
\end{aligned} \tag{19}$$

where the inequality is obtained by applying the proof of Lemma 11.3 with $\mathbf{y} = \mathbf{y}_{i+1} - \mathbf{y}_i$ and $\mathbf{y}' = \mathbf{y}_i$. Then, by some routine calculation (shown below), we observe that the upper bound in Eq. (19) satisfies

$$\begin{aligned}
&\sum_{\mathbf{y}_i \in \mathbb{Z}^\ell} \gamma_{\beta,r} \cdot \Pr[\mathbf{Z}_{i+1} \geq \mathbf{y}_{i+1} - \mathbf{y}_i] \cdot \Pr[\mathbf{Y}_i = \mathbf{y}_i] \\
&= \gamma_{\beta,r} \cdot \sum_{\mathbf{y}' \in \mathbb{Z}^\ell} \Pr[\mathbf{Z}_{i+1} \geq \mathbf{y}'] \Pr[\mathbf{Y}_i = \mathbf{y}_{i+1} - \mathbf{y}'] \quad /\!/\ \mathbf{y}' := \mathbf{y}_{i+1} - \mathbf{y}_i \\
&= \gamma_{\beta,r} \cdot \sum_{\mathbf{y}' \in \mathbb{Z}^\ell} \sum_{\mathbf{d} \geq \mathbf{0}} \Pr[\mathbf{Z}_{i+1} = \mathbf{y}' + \mathbf{d}] \Pr[\mathbf{Y}_i = \mathbf{y}_{i+1} - \mathbf{y}'] \\
&= \gamma_{\beta,r} \cdot \sum_{\mathbf{y}'' \in \mathbb{Z}^\ell} \sum_{\mathbf{d} \geq \mathbf{0}} \Pr[\mathbf{Z}_{i+1} = \mathbf{y}''] \Pr[\mathbf{Y}_i = \mathbf{y}_{i+1} - \mathbf{y}'' + \mathbf{d}] \quad /\!/\ \mathbf{y}'' = \mathbf{y}' + \mathbf{d} \\
&= \gamma_{\beta,r} \cdot \sum_{\mathbf{y}'' \in \mathbb{Z}^\ell} \Pr[\mathbf{Z}_{i+1} = \mathbf{y}''] \Pr[\mathbf{Y}_i \geq \mathbf{y}_{i+1} - \mathbf{y}''].
\end{aligned}$$

Finally, we see that the above quantity satisfies

$$\begin{aligned}
&\gamma_{\beta,r} \cdot \sum_{\mathbf{y}'' \in \mathbb{Z}^\ell} \Pr[\mathbf{Z}_{i+1} = \mathbf{y}''] \Pr[\mathbf{Y}_i \geq \mathbf{y}_{i+1} - \mathbf{y}''] \\
&\leq \gamma_{\beta,r} \cdot \sum_{\mathbf{y}'' \in \mathbb{Z}^\ell} \Pr[\mathbf{Z}_{i+1} = \mathbf{y}''] \left(\gamma_{\beta,r}^i \cdot \Pr\left[\sum_{k=1}^{i} \mathbf{Z}_k \geq \mathbf{y}_{i+1} - \mathbf{y}''\right]\right) \\
&= \gamma_{\beta,r}^{i+1} \cdot \Pr\left[\sum_{k=1}^{i+1} \mathbf{Z}_k \geq \mathbf{y}_{i+1}\right]
\end{aligned}$$

where the inequality is due to the induction hypothesis. □

11.2 Proof of Inductive Step

Below, we prove Lemma 11.3 which was used in the inductive step in the proof of Theorem 11.1.

Lemma 11.3. *Let the notation be as in Theorem 11.1, to be recalled below: Let $\mathcal{R}$ be the ring of integers of number field $\mathcal{K}$ of degree φ. Let $\mu, \ell \in \mathbb{N}$, $f \in \mathcal{R}[\mathbf{X}]$ be a μ-variate multilinear polynomial, $d \in \mathcal{R}$ with ideal factorisation $\langle d \rangle = \prod_{j=1}^{\ell} \mathfrak{p}_j^{r_j}$ satisfying $\mathsf{cont}(f) + \langle d \rangle = \langle 1 \rangle$, where $\mathsf{cont}(f)$ denotes the content of f, i.e. the ideal generated by the coefficients of f, and $\mathcal{C} := \mathcal{C}_\beta := \{x \in \mathcal{R} : \|x\| \leq \beta\}$. For $j \in [\ell]$, $k \in [\mu]$, let $Z_{j,k} \sim \mathsf{Geo}(1 - 1/\mathcal{N}(\mathfrak{p}_\mathrm{j}))$ be a random variable distributed geometrically with parameter $1 - 1/\mathcal{N}(\mathfrak{p}_\mathrm{j})$, and $Y_{j,k}$ be a random variable representing the multiplicity of $\mathfrak{p}_\mathrm{j}$ in $\mathsf{cont}(f_k[\mathbf{x}_k])$ where $\mathbf{x}_k \leftarrow_R \mathcal{C}^k$. Let $r \geq \rho(\langle d \rangle)$, the covering radius of the ideal lattice $\langle d \rangle$. For any $k \in [\mu]$, $\mathbf{y}, \mathbf{y}' \in \mathbb{Z}^\ell$, $\mathbf{y}' \geq \mathbf{0}$, it holds that*

$$\Pr[\mathbf{Y}_k \geq \mathbf{y} + \mathbf{Y}_{k-1} \mid \mathbf{Y}_{k-1} = \mathbf{y}'] \leq \frac{|\mathcal{C}_{\beta+2r}|}{|\mathcal{C}_\beta|} \cdot \Pr[\mathbf{Z}_k \geq \mathbf{y}'].$$

where $\mathbf{Z}_k := (Z_{1,k}, \ldots, Z_{\ell,k})$ and $\mathbf{r} := (r_1, \ldots, r_\ell)$.

Proof. (Lemma 11.3) It suffices to consider $\mathbf{y} > \mathbf{0}$, i.e. $y_j > 0$ for some $j \in [\ell]$, else $\Pr[\mathbf{Z}_k \geq \mathbf{y}] = 1$ and the statement is trivial. Let $J \subseteq [\ell]$ denote the set

$$J := \{j \in [\ell] : y_j > 0\}.$$

Fix $k \in [\mu]$. Write

$$f_{k-1}[\mathbf{x}_{k-1}](X_k, X_{k+1}, \ldots, X_\mu) = h_k(X_{k+1}, \ldots, X_\mu) + X_k \cdot g_k(X_{k+1}, \ldots, X_\mu)$$

where h_k and g_k are $(\mu - k)$-linear polynomials over $\mathcal{R}$. We make two observations.

First, conditioned on the event $\mathbf{Y}_{k-1} = \mathbf{y}'$, we have that

$$\forall j \in [\ell], \begin{cases} h_k = g_k = 0 \bmod {\mathfrak{p}_\mathrm{j}}^{y'_j} \\ h_k \neq 0 \bmod {\mathfrak{p}_\mathrm{j}}^{y'_j+1} \vee g_k \neq 0 \bmod {\mathfrak{p}_\mathrm{j}}^{y'_j+1} \end{cases}$$

for otherwise we would have $Y_{j,k-1} \geq y'_j + 1$ for some j. Consequently, for all $j \in [\ell]$,

$$h_k(X_{k+1}, \ldots, X_\mu) + X_k \cdot g_k(X_{k+1}, \ldots, X_\mu) \begin{cases} \equiv 0 \bmod {\mathfrak{p}_\mathrm{j}}^{y'_j} \\ \not\equiv 0 \bmod {\mathfrak{p}_\mathrm{j}}^{y'_j+1}. \end{cases}$$

This means that, for $j \in J$, since $y_j > 0$, there exists a coefficient $h_{k,j} \in \mathcal{R}$ (resp. $g_{k,j} \in \mathcal{R}$) of a monomial (of $(X_{k+1}, \ldots, X_\mu)$) in h_k (resp. g_k) such that the univariate affine polynomial

$$h_{k,j} + X_k \cdot g_{k,j} \begin{cases} = 0 \bmod {\mathfrak{p}_\mathrm{j}}^{y'_j} \\ \neq 0 \bmod {\mathfrak{p}_\mathrm{j}}^{y'_j+1}. \end{cases}$$

Second, conditioned on the event $\mathbf{Y}_{k-1} = \mathbf{y}'$, the following events are equivalent:

$$\mathbf{Y}_k \geq \mathbf{y} + \mathbf{Y}_{k-1} \qquad \text{and} \qquad \forall j \in [\ell],\ h_k + x_k \cdot g_k = 0 \bmod {\mathfrak{p}_\mathrm{j}}^{y'_j + y_j}.$$

Putting the above two observations together, we have

$$\begin{aligned}
&\Pr[\mathbf{Y}_k \geq \mathbf{y} + \mathbf{Y}_{k-1} \mid \mathbf{Y}_{k-1} = \mathbf{y}'] \\
=\ & \Pr\left[\forall j \in [\ell],\ h_k + x_k \cdot g_k = 0 \bmod \mathfrak{p}_\mathfrak{j}{}^{y'_j + y_j} \;\middle|\; \mathbf{Y}_{k-1} = \mathbf{y}'\right] \\
\leq\ & \Pr\left[\forall j \in J,\ h_{k,j} + x_k \cdot g_{k,j} = 0 \bmod \mathfrak{p}_\mathfrak{j}{}^{y'_j + y_j} \;\middle|\; \mathbf{Y}_{k-1} = \mathbf{y}'\right].
\end{aligned}$$

Conditioned on the event $\mathbf{Y}_{k-1} = \mathbf{y}'$, for each $j \in J$, since

$$h_{k,j} + X_k \cdot g_{k,j} \begin{cases} = 0 \bmod \mathfrak{p}_\mathfrak{j}{}^{y'_j} \\ \neq 0 \bmod \mathfrak{p}_\mathfrak{j}{}^{y'_j + 1}, \end{cases}$$

there is at most one solution $x_k^* \in \mathcal{R}/\mathfrak{p}_\mathfrak{j}{}^{y_j}$ satisfying

$$h_{k,j} + X_k \cdot g_{k,j} = 0 \bmod \mathfrak{p}_\mathfrak{j}{}^{y'_j + y_j}.$$

By the Chinese remainder theorem, the system of equations

$$\forall j \in J,\ h_{k,j} + X_k \cdot g_{k,j} = 0 \bmod \mathfrak{p}_\mathfrak{j}{}^{y'_j + y_j}$$

admits at most one solution $x_k^* \in \mathcal{R}/\prod_{j \in J} \mathfrak{p}_\mathfrak{j}{}^{y_j}$.

For each coset of $\mathcal{R}/\prod_{j \in J} \mathfrak{p}_\mathfrak{j}{}^{y_j}$, we will show that the number of elements in $\mathcal{C}$ belonging to this coset is at most

$$\frac{|\mathcal{C}_{\beta+2r}|}{\prod_{j \in J} \mathcal{N}(\mathfrak{p}_\mathfrak{j})^{y_j}}.$$

Consider the ideal $\mathcal{I} := \prod_{j \in J} \mathfrak{p}_\mathfrak{j}{}^{y_j}$. Denote $N := \mathcal{N}(I) = \prod_{j \in J} \mathcal{N}(\mathfrak{p}_\mathfrak{j})^{y_j}$ and let $a_1, \ldots, a_N$ be the shortest representatives of the cosets of $\mathcal{I}$. Recall that r is greater than the covering radius of $\langle d \rangle$, hence, of $\mathcal{I}$, so that $\|a_i\| \leq r$ for all $i \in [N]$.

For each $i \in [N]$, define $\mathcal{C}_{\beta,i} := \{\alpha \in a_i + \mathcal{I} : \|\alpha\| \leq \beta\}$ and similarly define $\mathcal{C}_{\beta+2r,i}$. We have the following injection for all $i, j \in [N]$:

$$\begin{aligned}
\mathcal{C}_{\beta,i} &\hookrightarrow \mathcal{C}_{\beta+2r,j} \\
\alpha &\mapsto \alpha - a_i + a_j.
\end{aligned}$$

This yields

$$N \cdot |\mathcal{C}_{\beta,i}| \leq \sum_{i=1}^{N} |\mathcal{C}_{\beta+2r,i}| = |\mathcal{C}_{\beta+2r}|$$

where the equality is due to $\{\mathcal{C}_{\beta+2r,i}\}_{i=1}^{N}$ being a partition of $\mathcal{C}_{\beta+r}$, and the inequality is due to the injection constructed above.

The probability of x_k sampled uniformly from $\mathcal{C}$ falling into the coset $\mathcal{R}/\prod_{j=1}^{\ell} \mathfrak{p}_\mathfrak{j}{}^{y_j} + x_k^*$ is therefore

$$\Pr\left[\forall j \in [\ell],\ x_k = x_k^* \bmod \prod_{j \in J} \mathfrak{p}_\mathfrak{j}{}^{y_j}\right] \leq \frac{|\mathcal{C}_{\beta+2r}|}{|\mathcal{C}_\beta|} \cdot \frac{1}{N}.$$

where the inequality is due to the above derivation. To complete the proof, it remains to see that, for each $j \in [\ell]$,

$$\Pr[Z_{j,k} \geq y_j] = \left(\frac{1}{\mathcal{N}(\mathfrak{p}_\mathsf{j})}\right)^{y_j}$$

since $Z_{j,k} \sim \mathsf{Geo}(1 - 1/\mathcal{N}(\mathfrak{p}_\mathsf{j}))$. Therefore $\Pr[\mathbf{Z}_k \geq \mathbf{y}] = \frac{1}{N}$. □

Remark 11.4. The covering radius is bounded above by $\dfrac{\varphi}{4}|\Delta_\mathcal{K}\mathcal{N}(I)|^{1/\varphi}$ [BF06, Proposition 4.2]. So in the case of the imaginary quadratic field $\mathcal{K} = \mathbb{Q}(\sqrt{\Delta})$, where $\Delta < 0$ is square-free integer, $|\Delta_\mathcal{K}| = 4\Delta$ or Δ, depending on whether $\Delta \equiv 2, 3 \pmod 4$ or not. And hence, the covering radius will be bounded above by $\sqrt{|\Delta\mathcal{N}(I)|}$ or $\dfrac{1}{2}\sqrt{|\Delta\mathcal{N}(I)|}$, respectively, which we can use for the r above.

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

$$2r_\mathcal{I} \leq \sqrt{\varphi}\lambda_\varphi(\mathcal{I}) \leq \varphi^{3/2}\mathcal{N}(\mathcal{I})^{1/\varphi},$$

so we can pick $r \geq \frac{\varphi^{3/2}}{2}\mathcal{N}(\mathcal{I})^{1/\varphi}$ in the lemma above.

11.3 Bounding Ratio of Number of Bounded Norm Elements

In Theorem 11.1, we derived an upper bound in terms of the ratio

$$\gamma_{\beta,r} \coloneqq \frac{|\mathcal{C}_{\beta+2r}|}{|\mathcal{C}_\beta|}.$$

Below, we upper bound this quantity 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 11.5 ([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 lattice. If $\mathcal{D}$ is symmetric around the origin then:*

$$2^{-n} \cdot \frac{\mathrm{vol}(\mathcal{D})}{\det(\Lambda)} \leq |\mathcal{D} \cap \Lambda|.$$

If $\mathcal{D}$ contains m linear independent vectors of Λ, then

$$|\mathcal{D} \cap \Lambda| \leq (m+1)! \cdot \frac{\mathrm{vol}(\mathcal{D})}{\det(\Lambda)}.$$

Alternatively, for $\Lambda = \mathbb{Z}^n$, applying the theorem of Davenport [Dav51], we get

$$||\mathcal{D} \cap \mathbb{Z}^n| - \mathrm{vol}(\mathcal{D})| \leq \sum_{j=0}^{n-1} \mathrm{vol}_j(\mathcal{D}),$$

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 11.6. *Let $\mathcal{R}$ 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 $\mathcal{C}_\beta := \{x \in \mathcal{R} : \|x\| \leq \beta\}$. It holds that*

$$2^{-\varphi_2} \cdot \frac{\pi^{\varphi_2} \beta^{\varphi_1 + \varphi_2}}{\Delta_{\mathcal{K}}^{1/2}} \leq |\mathcal{C}_\beta| \leq (\varphi_1 + \varphi_2 + 1)! \cdot \frac{2^{\varphi_1} \pi^{\varphi_2} \beta^{\varphi_1 + \varphi_2}}{\Delta_{\mathcal{K}}^{1/2}}$$

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

$$\frac{\pi\beta}{4\sqrt{|\Delta|}} \leq |\mathcal{C}_\beta| \leq 2\beta \ln \beta.$$

If $\mathcal{K}$ is cyclotomic, then

$$\left(\frac{\pi \cdot \beta}{\varphi}\right)^{\varphi/2} \leq |\mathcal{C}_\beta| \leq (\varphi/2 + 1)! \cdot \left(\frac{\pi \cdot \beta}{\varphi^{1/2}} \cdot\right)^{\varphi/2}$$

Proof. The volume $\mathrm{vol}(\mathcal{C}_\beta)$, corresponding to the volume of $\{\mathbf{x} \in \mathbb{R}^{\varphi_1 + \varphi_2} : \|\mathbf{x}\| \leq \beta\}$, is given by

$$\mathrm{vol}(\mathcal{C}_\beta) = 2^{\varphi_1} \pi^{\varphi_2} \beta^{\varphi_1 + \varphi_2}.$$

Considering $\mathcal{R}$ in the Minkowski space, we have $\mathcal{C} \subset \mathbb{R}^{\varphi_1 + \varphi_2}$. The general version of the bounds follows immediately by applying Lemma 11.5, here the sufficiently large value for β is at least the $\varphi_1 + \varphi_2$th successive minima. In the case of imaginary quadratic fields, for the lower bound, we use the general bound. Note that $\|x\| = \sqrt{|\mathcal{N}(x)|}$, and for upper bound, we count integers of bounded norm (or ideals of bounded norm) up to β. We use the result from [BS66, p. 231] that the number of ideals of norm a is bounded by the number of divisors of a. The upper bound for the divisor summation function up to β is less than $\beta(2\ln(\beta) + 2\gamma_{EM} - 1) + 0.961\beta$ [BBR12, Theorem 1.1], where $\gamma_{EM} \approx 0.57721\ldots$ is the Euler-Mascheroni constant.

If $\mathcal{K}$ is cyclotomic with conductor $\mathfrak{f}$, its discriminant $\Delta_{\mathcal{K}}$ satisfies (see e.g. [LPR13, Section 2.5.3])

$$\varphi^{\varphi/2} \leq \Delta_{\mathcal{K}} = \left(\frac{\mathfrak{f}}{\prod_{\text{prime } p|\mathfrak{f}} p^{1/(p-1)}}\right)^{\varphi} \leq \varphi^{\varphi}.$$

where for power-of-2 $\mathfrak{f}$ the upper bound is tight. The concrete version of the bounds thus follows. □

As an immediate corollary, we obtain the following upper bounds for $\gamma_{\beta,r}$.

Corollary 11.7. *If $\mathcal{K} = \mathbb{Q}(\sqrt{\Delta})$ for a square-free integer $\Delta < 0$ and $\beta \geq e$, then*

$$\gamma_{\beta,r} = \frac{|\mathcal{C}_{\beta+2r}|}{|\mathcal{C}_\beta|} \leq \frac{8|\Delta|}{\pi}(1 + 2r/\beta)\ln(\beta + 2r).$$

If $\mathcal{K}$ is cyclotomic of degree φ, then

$$\gamma_{\beta,r} = \frac{|\mathcal{C}_{\beta+2r}|}{|\mathcal{C}_\beta|} \leq (\varphi/2 + 1)! \cdot \varphi^{\varphi/4} \cdot (1 + 2r/\beta)^{\varphi/2}.$$

11.4 Inverse Bound

Our next goal is to upper bound

$$\Pr\left[\sum_{k=1}^{i} \mathbf{Z}_k \geq \mathbf{r}\right].$$

in terms of $\frac{1}{|\mathcal{N}(d)|}$.

Theorem 11.8. *Let d, $\mathbf{Z}_k$, and $\mathbf{r}$ be defined as in Theorem 11.1. Write $\nu = |\mathcal{N}(d)|$. It holds that*

$$\Pr\left[\sum_{k=1}^{i} \mathbf{Z}_k \geq \mathbf{r}\right] \leq \nu^{-\left(\frac{1}{2} - \frac{2\mu\varphi \ln\ln\ln\nu}{\ln\ln\nu}\right)}.$$

Proof. Recall that $\epsilon = \Pr[\sum_{k=1}^{\mu} \mathbf{Z}_k \geq \mathbf{r}]$ can be written as

$$\Pr\left[\sum_{k=1}^{\mu} \mathbf{Z}_k \geq \mathbf{r}\right] = \prod_{j=1}^{\ell} I_{1/\mathcal{N}(\mathfrak{p}_j)}(r_j, \mu).$$

Consider the prime ideal factorisation $\langle d \rangle = \prod_{j=1}^{\ell} \mathfrak{p}_j^{r_j}$. Given that we can have multiple primes laying above p, we cannot take the result [BF23, Theorem 2] verbatim. This however does not affect the case when $\mu = 1$, as then $I_{1/\mathcal{N}(\mathfrak{p}_i)}(r_i, 1) = \frac{1}{\mathcal{N}(\mathfrak{p}_i)^{r_i}}$. Thus, here $\mathcal{N}(d) \geq 2^\lambda$ suffices to yield $\epsilon = \Pr[\mathbf{Z}_1 \geq \mathbf{r}] \leq 2^{-\lambda}$.

We next consider the case $\mu \geq 2$. Unlike in the proof of [BF23, Theorem 2], we take an elementary approach. We use the following identity

$$I_{1/\mathcal{N}(\mathfrak{p}_j)}(r_j, \mu + 1) = I_{1/\mathcal{N}(\mathfrak{p}_j)}(r_j, \mu) + \frac{1}{\mathcal{N}(\mathfrak{p}_j)^{r_j}} \frac{(1 - 1/\mathcal{N}(\mathfrak{p}_j))^\mu}{\mu B(r_j, \mu)},$$

and

$$B(r_j, \mu) = \frac{r_j + \mu}{r_j \mu} / \binom{r_j + \mu}{\mu},$$

to get

$$I_{1/\mathcal{N}(\mathfrak{p}_j)}(r_j, \mu) = \frac{1}{\mathcal{N}(\mathfrak{p}_j)^{r_j}} \sum_{j=0}^{\mu-1} (1 - 1/\mathcal{N}(\mathfrak{p}_j))^i \binom{r_j + i - 1}{i}.$$

We want to upper-bound the following:

$$\begin{aligned} I_{1/\mathcal{N}(\mathfrak{p}_j)}(r_j, \mu) &\leq \frac{1}{\mathcal{N}(\mathfrak{p}_j)^{r_j}} \sqrt{\sum_{i=0}^{\mu-1} (1 - 1/\mathcal{N}(\mathfrak{p}_j))^i} \sqrt{\sum_{i=0}^{\mu-1} \binom{r_j + i - 1}{i}} \\ &\leq \frac{1}{\mathcal{N}(\mathfrak{p}_j)^{r_j}} \sqrt{\frac{1 - (1 - 1/\mathcal{N}(\mathfrak{p}_j))^\mu}{1 - 1/\mathcal{N}(\mathfrak{p}_j)}} \sqrt{\binom{r_j + \mu - 1}{\mu - 1}} \end{aligned}$$

where the first inequality is due to Cauchy-Schwartz, and the second is elementary. Taking products over all primes and then taking logarithm gives:

$$\begin{aligned} \ln \epsilon &= \ln \left(\prod_{j=1}^{\ell} I_{1/\mathcal{N}(\mathfrak{p}_j)}(r_j, \mu) \right) \\ &\leq -\ln |\mathcal{N}(d)| + \frac{1}{2} \underbrace{\ln \left(\prod_{j=1}^{\ell} \frac{1 - (1 - 1/\mathcal{N}(\mathfrak{p}_j))^\mu}{1 - 1/\mathcal{N}(\mathfrak{p}_j)} \right)}_{=:A} \\ &\quad + \frac{1}{2} \underbrace{\ln \left(\prod_{j=1}^{\ell} \binom{r_j + \mu - 1}{\mu - 1} \right)}_{=:B} \\ &= -\ln \nu + \frac{1}{2}(A + B), \end{aligned} \tag{20}$$

where in the last equality we used the shorthand $\nu \coloneqq |\mathcal{N}(d)|$.

Next, we upper bound the terms A and B. For that, we need to introduce the prime omega functions $\omega, \Omega : \mathbb{N} \longrightarrow \mathbb{N}$. The first one, ω, counts the number of distinct primes, while the other, Ω, counts them with multiplicity. We have $\omega(n) \leq \Omega(n)$ for all $n \in \mathbb{N}$, and the obvious bound for $\Omega(n) \leq \ln(n)/\ln(2)$, with equality when n is a power of 2. For A, we observe

$$\begin{aligned} A &= \ln \left(\prod_{j=1}^{\ell} \mathcal{N}(\mathfrak{p}_j)^{-\mu+1} (\mathcal{N}(\mathfrak{p}_j) - 1)^{-1} (\mathcal{N}(\mathfrak{p}_j)^\mu - (\mathcal{N}(\mathfrak{p}_j) - 1)^\mu) \right) \\ &< \ln \left(\mu^\ell\right) \prod_{j=1}^{\ell} (\mathcal{N}(\mathfrak{p}_j) - 1)^{-1} \quad /\!/ \ \textstyle\sum_{i=0}^{\mu-1} (-1)^{i+1} \binom{\mu}{i} / \mathcal{N}(d)^i < \mu. \end{aligned}$$

$$\leq \ell \ln \mu \quad /\!\!/ \text{ with equality if and only if } \mathcal{N}(\mathfrak{p}_j) = 2 \text{ for all } j.$$

For B, we have

$$\begin{aligned}
B &\leq \ln \left(\prod_{j=1}^{\ell} \left(\frac{(r_j + \mu - 1)e}{\mu - 1} \right)^{\mu-1} \right) \quad /\!\!/ \text{ using } \binom{n}{k} < \left(\frac{ne}{k}\right)^k. \\
&\leq (\mu - 1) \ln \left(\prod_{j=1}^{\ell} r_j e \right) \quad /\!\!/ \text{ assuming } \mu > 2. \\
&\leq (\mu - 1) \left(\ell + \ell \ln \left(\frac{1}{\ell} \sum_{j=1}^{\ell} r_j \right) \right) \quad /\!\!/ \text{ applying AM-GM inequality.} \\
&\leq (\mu - 1)\ell \left(1 + \ln(\Omega(\nu)/\ell)\right) \quad /\!\!/ \text{ rewriting in terms of the prime } \Omega \text{ function.}
\end{aligned}$$

Putting the above together, and using the inequality $\omega(\nu) \leq \ell \leq \varphi\omega(\nu)$, i.e. bounding ℓ with number of distinct prime divisors of ν, we see that

$$A + B < 2\mu\varphi\omega(\nu)(1 + \ln(\Omega(\nu)/\omega(\nu))).$$

This ultimately brings us to finding an upper bound for

$$\omega(\nu)(1 + \ln(\Omega(\nu)/\omega(\nu))), \tag{21}$$

which we will show is at most $\dfrac{\ln \nu \ln \ln \ln \nu}{\ln \ln \nu}$. We first note that $\omega(\nu) < \frac{3}{2} \dfrac{\ln \nu}{\ln \ln \nu}$ [Rob83, Théorème 11]. For the special case of square-free ν, we have that $\Omega(\nu) = \omega(\nu)$, i.e. the upper bound for the equation (21) is just $\frac{3}{2} \dfrac{\ln \nu}{\ln \ln \nu}$. In what follows, we handle the case where ν is not square-free.

The strategy is, starting with any initial value of $\omega(\nu)$, to adjust ν (viewing it as a variable in what follows) so that equation (21) is maximised. Let $X > 0$, which will be later defined, and let $\nu = (\prod_{p<X} p)(2^{(\ln(\nu)-X)/\ln(X)})$ vary with X, where the product is over all the primes up to X (here we use Chebyshev's theorem that $\prod_{p<X} p = e^{(1+o(1))X}$). Our goal is to search for the maximum of equation (21) in some range around ν. From the prime number theorem, we have that $\omega(\nu) \approx X/\ln(X)$ and $\Omega(\nu) \approx X/\ln(X) + (\ln(\nu) - X)/\ln(2)$. Substituting these values into the equation (21), specifically, for $\ln\left(\frac{\Omega(\nu)}{\omega(\nu)}\right)$:

$$\begin{aligned}
\ln \left(1 + \frac{\ln(\nu)\ln(X)}{X \ln(2)} - \frac{\ln(X)}{\ln(2)} \right) &\leq \ln \left(\frac{\ln(\nu)\ln(X)}{X \ln(2)} \right) \\
&\leq \ln \ln(\nu) + \ln \ln(X) - \ln(X) - \ln \ln(2).
\end{aligned}$$

Putting everything together in (21) gives us:

$$\frac{X}{\ln(X)}(\ln \ln(\nu) + \ln \ln(X) - \ln(X) - \ln \ln(2)) \leq X \left(\frac{\ln \ln(\nu)}{\ln(X)} + \frac{\ln \ln(X)}{\ln(X)} - 1 \right).$$

Analysing now, the cases when X is sufficiently larger or smaller than $\dfrac{\ln\nu}{\ln\ln\nu}$. For the larger case, let $Y = \ln\ln(\nu)$, we get:

$$\frac{\ln(\nu)}{\ln\ln(\nu)}\left(\frac{Y}{Y-\ln(Y)}+\frac{\ln(Y)}{Y-\ln(Y)}-1\right)=\frac{\ln(\nu)}{\ln\ln(\nu)}\left(\frac{2\ln(Y)}{Y-\ln(Y)}\right),$$

and the required bound follows. The other case is much simpler.

Going back now to the sum in the equation (20), we see that $-\ln(\nu)+\frac{1}{2}(A+B)$ is at most

$$-\left(\frac{1}{2}-\frac{2\mu\varphi\ln\ln\ln\nu}{\ln\ln\nu}\right)\cdot\ln\nu.$$

□

Combining Theorems 11.1 and 11.8 and Corollary 11.7 we obtain the following Corollaries 11.9 and 11.10.

Corollary 11.9. *Let $\mathcal{R},\beta,\varphi,\mu,\nu,f,d,\mathcal{C},r$ satisfy the following properties:*

- *$\mathcal{R}$ is the ring of integers of a cyclotomic field $\mathcal{K}$ of degree φ.*
- *$\beta > 0$.*
- *$f \in \mathcal{R}[\mathbf{X}]$ is a μ-variate multilinear polynomial.*
- *$d \in \mathcal{R}$ with $\mathsf{cont}(f) + \langle d\rangle = \langle 1\rangle$, where $\mathsf{cont}(f)$ denotes the content of f, i.e. the ideal generated by the coefficients of f.*
- *$\mathcal{C} := \mathcal{C}_\beta := \{x \in \mathcal{R} : \|x\| \le \beta\}$, $\gamma_{\beta,r} := \frac{|\mathcal{C}_{\beta+2r}|}{|\mathcal{C}_\beta|}$.*
- *r is a number larger than the covering radius of $\langle d\rangle$.*
- *$\nu = |\mathcal{N}(d)|$.*

It holds that

$$\varepsilon_{\mathsf{SZ}} \le \frac{((\varphi/2+1)!)^\mu \cdot \left(\varphi^{1/2}\cdot(1+2r/\beta)\right)^{\mu\varphi/2}}{\nu^{\left(\frac{1}{2}-\frac{2\mu\varphi\ln\ln\ln\nu}{\ln\ln\nu}\right)}}.$$

We note that the term φ appearing in the exponent of $\nu^{\left(\frac{1}{2}-\frac{2\mu\varphi\ln\ln\ln\nu}{\ln\ln\nu}\right)}$ makes the bound very weak, since ν would need to be doubly exponential to make $\varepsilon_{\mathsf{SZ}}$ negligible. We expect this to be a limitation of proof techniques, and it should suffice to have ν being (singly) exponential for $\varepsilon_{\mathsf{SZ}}$ to be negligible.

Next, we turn to the case of imaginary quadratic fields.

Corollary 11.10. *Let $\mathcal{R},\beta,\mu,\nu,f,d,\mathcal{C},r$ satisfy the following properties:*

- *$\mathcal{R}$ is the ring of integers of an imaginary quadratic field $\mathcal{K} = \mathbb{Q}(\sqrt{\varDelta})$.*
- *$\beta \ge e$.*
- *$f \in \mathcal{R}[\mathbf{X}]$ is a μ-variate multilinear polynomial.*
- *$d \in \mathcal{R}$ with $\mathsf{cont}(f) + \langle d\rangle = \langle 1\rangle$, where $\mathsf{cont}(f)$ denotes the content of f, i.e. the ideal generated by the coefficients of f.*
- *r is a number larger than the covering radius of $\langle d\rangle$.*
- *$\nu = |\mathcal{N}(d)|$.*

It holds that

$$\varepsilon_{\mathsf{SZ}} \leq \frac{\left(\frac{12|\Delta|}{\pi}\right)^{\mu} (1 + 2r/\beta)^{2\mu} \ln^{\mu}(\beta + 2r)}{\nu^{\left(\frac{1}{2} - \frac{4\mu \ln\ln\ln\nu}{\ln\ln\nu}\right)}}.$$

Note that it suffices for $\nu \approx 2^{O(\lambda\mu)}$ for $\varepsilon_{\mathsf{SZ}}$ to be negligible.

11.5 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 10), where we can only upper bound the geometric norm $\|d\|$ instead of the field norm $\mathcal{N}(d)$ of the modulus, e.g. when $\mathcal{R}$ is a cyclotomic ring. In what follows, we assume that $\mathcal{R}$ is a cyclotomic ring of degree φ.

For any $e \in \mathcal{R}$, we call any $d \in \mathcal{R}$ satisfying $d = e \cdot u$ for some unit $u \in \mathcal{R}^{\times}$ 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 $e \in \mathcal{R}$ be such that its geometric norm is the smallest among its associates.[27] 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

$$\{x \in \mathcal{R} : \mathcal{N}(x) \leq \nu \ \wedge \ \gamma < \|x\| \leq \delta\}$$

in relation to the number of elements in the set

$$\{x \in \mathcal{R} : \|x\| \leq \beta\}$$

where $\beta \leq \gamma \leq \delta$ and ν is suitably small compared to β^{φ}, and show that the ratio shrinks rapidly as β grows.

Lemma 11.11. *Let $\mathcal{R}$ be a cyclotomic ring of degree $\varphi \geq 4$. Let $0 \leq \beta \leq \gamma \leq \delta$ and $\nu \in \mathbb{N}$. Define the sets:*

$$\begin{aligned} N_{\nu} &\coloneqq \{x \in \mathcal{R} : \mathcal{N}(x) \leq \nu\}, \\ M_{\delta} &\coloneqq \{x \in \mathcal{R} : \|x\| \leq \delta\}, \\ M_{\gamma,\delta} &\coloneqq M_{\delta} \setminus M_{\gamma} = \{x \in \mathcal{R} : \gamma < \|x\| \leq \delta\}. \end{aligned}$$

It holds that as $\delta, \gamma \to \infty$

$$\frac{|N_{\nu} \cap M_{\gamma,\delta}|}{|M_{\beta}|} \leq O\left(\frac{\nu \ln(\delta/\gamma)^{\varphi/2-1}}{\beta^{\varphi}}\right).$$

[27] For concreteness, in [CDPR16] it is shown that $\|e\| \leq \exp(\tilde{O}(\sqrt{\mathfrak{f}})) \cdot |\mathcal{N}(e)|^{1/\varphi}$.

Proof. To prove the above asymptotic, we rely on the following two results: First, up to multiplication by units, there are only finitely many algebraic integers of bounded norm (and more generally, ideals). We bound the number of ideals of norm at most ν, denote this value by N_ν^*. Then, by Theorem 1.1 (and a better and more elaborate bound in Theorem 1.2 (and as explicit, depending only on the φ and the discriminant)) in [Lee23] it follows

$$|N_\nu^*| \leq C'_{\mathcal{K}}\nu + C''_{\mathcal{K}}\nu^{1-2/\varphi},$$

where $C'_{\mathcal{K}} = \dfrac{(2\pi)^{\varphi/2}\mathrm{h}_{\mathcal{K}}\mathrm{R}_{\mathcal{K}}}{\omega|\mathit{\Delta}_{\mathcal{K}}|^{1/2}}$ and

$$\begin{aligned} C''_{\mathcal{K}} &= \exp(28.2\varphi+5)(\varphi+1)^{5(\varphi+1)/2}|\mathit{\Delta}_{\mathcal{K}}|^{1/(\varphi+1)}\ln(|\mathit{\Delta}_{\mathcal{K}}|)^{\varphi} \\ &\leq \exp(28.2\varphi+5)(\varphi+1)^{5(\varphi+1)/2}\varphi^{(\varphi^2+2\varphi)/(\varphi+1)}\ln(\varphi)^{\varphi}, \end{aligned}$$

above we used the inequality $|\mathit{\Delta}_{\mathcal{K}}| \leq \varphi^{\varphi}$. In the first constant, the absolute constant ω is the number of roots of unity, $\mathrm{R}_{\mathcal{K}}$ and $\mathrm{h}_{\mathcal{K}}$ is the regulator and the class number respectively. The second result we need is an estimate of the number of units of the bounded norm. We have that for cyclotomic number fields of degree ≥ 4, as $\gamma \to \infty$

$$|N_1 \cap M_\gamma| = C'''_{\mathcal{K}}\ln(\gamma)^{\phi/2-1} + O(\ln(\gamma)^{\varphi/2-2}),$$

again, here, $C'''_{\mathcal{K}}$ depends only on the number field (see Theorem in [EL93]). Specifically,

$$C'''_{\mathcal{K}} = \frac{\omega(\varphi/2)^{\varphi/2-1}}{\mathrm{R}_{\mathcal{K}}(\varphi/2-1)!}.$$

The big O term can be improved, and the improvement will depend only on φ.

Now it follows that:

$$\begin{aligned} |N_\nu \cap M_{\delta,\gamma}| &\leq |N_\nu^*||N_1 \cap M_\gamma| - |N_\nu^*||N_1 \cap M_\delta| \\ &\leq (C'_{\mathcal{K}}\nu + C''_{\mathcal{K}})C'''_{\mathcal{K}}(\ln(\gamma)^{\varphi/2-1} - \ln(\delta)^{\varphi/2-1} + O(\ln(\gamma)^{\varphi/2-2})) \\ &\leq C_{\mathcal{K}}\nu\ln(\delta/\gamma)^{\varphi/2-1} + O(\nu^{1-2/\varphi}\ln(\gamma)^{\varphi/2-2}), \end{aligned}$$

where we collect all the field's constants, using that $\mathit{\Delta}_{\mathcal{K}} \geq \varphi^{\varphi/2}$ and Theorem 6.5 in [Len92] for the class number bound to get:

$$C_{\mathcal{K}} = \frac{2^{\varphi/2+1}\varphi^{3/4\varphi-1}(\varphi-1+\varphi/2\ln(2\varphi/\pi))^{\varphi-1}}{(\varphi/2-1)!(\varphi-1)!} = \varphi^{\Theta(\varphi)}.$$

Finally, from Lemma 11.6, we know that $M_\beta \geq \left(\dfrac{\pi}{2\varphi}\right)^{\varphi/2}\beta^{\varphi}$, therefore, the bound above follows. □

Remark 11.12. 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.

References

[AAB+24a] Marius A. Aardal, Diego F. Aranha, Katharina Boudgoust, Sebastian Kolby, and Akira Takahashi. *Aggregating Falcon Signatures with LaBRADOR*. Cryptology ePrint Archive, Report 2024/311. 2024. URL: `https://eprint.iacr.org/2024/311` (cit. on pp. 6, 9, 42–44).

[AAB+24b] 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. 2, 3, 41, 43–45).

[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 p. 12).

[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. 2, 4, 5, 7, 9, 12–14, 44, 56, 57).

[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. 2, 5, 9, 12, 26, 29, 39).

[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. 2, 12, 26, 42, 44).

[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. 2, 12, 26).

[AFKR23] Thomas Attema, Serge Fehr, Michael Klooß, and Nicolas Resch. *The Fiat–Shamir Transformation of $(\Gamma_1, \dots, \Gamma_\mu)$-Special-Sound Interactive Proofs.* Cryptology ePrint Archive, Report 2023/1945. 2023. URL: `https://eprint.iacr.org/2023/1945` (cit. on pp. 2, 4, 5, 9, 19, 25, 26, 28, 29).

[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. 2, 3, 5, 7–9, 12, 19, 26, 29, 38, 39, 56).

[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. 2, 4, 7, 13, 14, 57).

[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. 7).

[Att23] Thomas Attema. "Compressed Σ-Protocol Theory". PhD thesis. Leiden University, 2023 (cit. on p. 5).

[BBB+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. 2, 7, 13, 57).

[BBC+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. 6).

[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. 78).

[BC24] Dan Boneh and Binyi Chen. *LatticeFold: A Lattice-based Folding Scheme and its Applications to Succinct Proof Systems.* Cryptology ePrint Archive, Report 2024/257. 2024. URL: `https://eprint.iacr.org/2024/257` (cit. on p. 9).

[BCC+16] 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. 2, 3, 7, 13, 42, 49).

[BCG+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. 57).

[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. 2).

[BDL+18] 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. 7).

[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. 77).

[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. 2–7, 9, 13, 14, 17, 41–43, 72, 73, 79).

[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. 2, 4, 7, 13, 14, 57).

[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. 7).

[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. 2, 3, 6, 9, 40–42, 44, 45).

[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. 78).

[CDG+24a] Alessandro Chiesa, Marcel Dall'Agnol, Ziyi Guan, Nicholas Spooner, and Eylon Yogev. "Untangling the Security of Kilian's Protocol: Upper and Lower Bounds". In: *To appear at TCC* (2024) (cit. on pp. 4, 7, 12).

[CDG+24b] Alessandro Chiesa, Marcel Dall'Agnol, Ziyi Guan, Nicholas Spooner, and Eylon Yogev. *Untangling the Security of Kilian's Protocol: Upper and Lower Bounds*. Cryptology ePrint Archive, Paper 2024/1434. 2024. URL: `https://eprint.iacr.org/2024/1434` (cit. on p. 12).

[CDGS23] Alessandro Chiesa, Marcel Dall'Agnol, Ziyi Guan, and Nicholas Spooner. *On the Security of Succinct Interactive Arguments from Vector Commitments*. Cryptology ePrint Archive, Report 2023/1737. 2023. URL: `https://eprint.iacr.org/2023/1737` (cit. on pp. 6, 12).

[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. 83).

[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. 3, 9, 40, 41).

[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 pp. 5, 7).

[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. 78, 84).

[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. 84).

[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. 7).

[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. 7).

[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 p. 77).

[FMN23] Giacomo Fenzi, Hossein Moghaddas, and Ngoc Khanh Nguyen. *Lattice-Based Polynomial Commitments: Towards Asymptotic and Concrete Efficiency*. Cryptology ePrint Archive, Report 2023/846. 2023. URL: https://eprint.iacr.org/2023/846 (cit. on pp. 2, 6, 51, 52).

[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 pp. 2, 6).

[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. 4, 9, 10).

[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. 54).

[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. 2, 6).

[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. 2).

[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. 84).

[Len92] H. W. Lenstra Jr. “Algorithms in algebraic number theory”. In: *Bull. Amer. Math. Soc. (N.S.)* 26.2 (1992), pp. 211–244. ISSN: 0273-0979. DOI: 10.1090/S0273-0979-1992-00284-7. URL: https://doi.org/10.1090/S0273-0979-1992-00284-7 (cit. on p. 84).

[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. 6).

[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. 7).

[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. 7).

[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. 77, 78).

[Rob83] Guy Robin. “Estimation de la fonction de Tchebychef θ sur le k-ième nombre premier et grandes valeurs de la fonction $\omega(n)$ nombre de diviseurs premiers de n”. In: *Acta Arith.* 42.4 (1983), pp. 367–389. ISSN: 0065-1036. DOI: 10.4064/aa-42-4-367-389. URL: https://doi.org/10.4064/aa-42-4-367-389 (cit. on p. 81).

[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. 11).

[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. 62).

[YAZ+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. 7).

Table of Contents

Adaptive Special Soundness:
Improved Knowledge Extraction by Adaptive Useful Challenge Sampling . 1
Thomas Attema, Michael Klooß, Russell W. F. Lai, and Pavlo Yatsyna
1 Introduction . . . 2
1.1 Special Soundness and its Limitations . . . 3
1.2 Our Contributions . . . 4
Adaptive Special Soundness and Useful Challenge Structures . . . 4
Motivating Application: Soundness of Lattice-based Bulletproofs 4
1.3 Related Work . . . 5
Special Soundness Notions . . . 5
1.4 Other work . . . 6
1.5 Roadmap . . . 7
2 Technical Overview . . . 8
2.1 Adaptive Special Soundness . . . 8
Useful Challenge Structures (UCS) . . . 8
The (Almost Optimal) Extractor . . . 9
2.2 Template for Using UCS in Applications . . . 9
2.3 Kilian's Succinct Argument . . . 9
The Protocol . . . 10
Computational Soundness . . . 10
Knowledge Soundness . . . 11
2.4 Generalised Sequential Composition of Extractor . . . 12
2.5 Revisiting Lattice-based Bulletproofs . . . 13
Existing Results . . . 14
Our Goal . . . 14
Tree Representation of Transcripts . . . 15
Constructing the UCS . . . 16
$\mathcal{U}$-Adaptive Special Soundness Analysis . . . 17
3 Preliminaries . . . 18
3.1 Auxiliary Lemmata . . . 19
4 Adaptive Special Soundness . . . 19
4.1 Abstract Adversaries and Extractors . . . 20
4.2 Useful Challenge Structures (UCS) and Adaptive Special Soundness for Σ-Protocols . . . 21
4.3 Nested UCS-Chains and Adaptive Special Soundness for Multi-Round Protocols . . . 22
4.4 Adaptive Witness Extractors and Adaptive Special Soundness . . . 24
5 Knowledge Extractors for Useful Challenge Structures . . . 25
5.1 Notation and Conventions . . . 26
5.2 Two Chain Rules . . . 27

5.3 Knowledge Extractor Construction and Analysis 28
6 Examples and Comparisons with Related Work 38
6.1 Transcript-Agnostic Examples and Access Structures [AFR23] . . . 38
6.2 A Basic Lattice Example 40
6.3 Predicate and Almost Special Soundness 41
Almost Special Soundness 42
Predicate Special Soundness 43
7 Sequential Composition of UCS Extractors 45
8 Examples for Sequential Composition 49
8.1 Translation between Extractor Settings 49
8.2 Worked Examples of Sequential Compositions 49
(Tree-)Special Soundness 49
Self-Composition with 2-Special Soundness 50
Component-wise k-special soundness (CWSS) 51
9 Sequential Composition of Abstract Extractors 52
9.1 Introduction to Sequential Composition 52
9.2 Generalised Sequential Composition 53
9.3 Additional Preliminaries 54
9.4 Generic Success Bounds for Sequential Composition 55
10 Lattice-based Bulletproofs 56
10.1 Algebraic Number Theory Background 56
10.2 Protocol Description 57
10.3 Soundness Claim 59
10.4 Adaptive Witness Extractor for Lattice Bulletproofs 60
10.5 Knowledge Soundness Analysis 63
11 Multilinear Composite Schwartz-Zippel over Rings 72
11.1 Main Theorem 72
11.2 Proof of Inductive Step 74
11.3 Bounding Ratio of Number of Bounded Norm Elements 77
11.4 Inverse Bound 79
11.5 On the Ratio of Disproportionately Long Elements 83