The Fiat–Shamir Transformation of $(\varGamma_1, \ldots, \varGamma_\mu)$-Special-Sound Interactive Proofs

Thomas Attema[1,3], Serge Fehr[1,2], Michael Klooß[4], and Nicolas Resch[5]

[1] CWI, Cryptology Group, Amsterdam, The Netherlands
serge.fehr@cwi.nl
[2] Leiden University, Mathematical Institute, Leiden, The Netherlands
[3] TNO, Applied Cryptography & Quantum Applications, The Hague, The Netherlands
thomas.attema@tno.nl
[4] Aalto University, Espoo, Finland
michael.klooss@aalto.fi
[5] University of Amsterdam, Informatics Institute, Amsterdam, The Netherlands
n.a.resch@uva.nl

December 22, 2023

Abstract. The Fiat-Shamir transformation is a general principle to turn any public-coin interactive proof into non-interactive one (with security then typically analyzed in the random oracle model). While initially used for 3-round protocols, many recent constructions use it for *multi-round* protocols. However, in general the soundness error of the Fiat-Shamir transformed protocol degrades exponentially in the number of rounds. On the positive side, it was shown that for the special class of $(k_1, \ldots, k_\mu)$-special-sound protocols the loss is actually only *linear* in the number of random oracle queries, and *independent* of the number of rounds, which is optimal.
A natural next question is whether this positive result extends to the Fiat-Shamir transformation of so-called $(\varGamma_1, \ldots, \varGamma_\mu)$-special-sound protocols, a notion recently defined and analyzed in the interactive case, with the aim to capture the most general notion of special-soundness.
We show in this work that this is indeed the case. Concretely, we show that the Fiat–Shamir transformation of any $(\varGamma_1, \ldots, \varGamma_\mu)$-special-sound interactive proof is knowledge sound under the same condition under which the original interactive proof is knowledge sound. Furthermore, also here the loss is linear in the number of random-oracle queries and independent of the number of rounds.
In light of the above, one might suspect that our argument follows as a straightforward combination of the above mentioned prior works. However, this is not the case. The approach used for $(k_1, \ldots, k_\mu)$-special-sound protocols, which is based on an extractor that samples without replacement, does not (seem to) generalize; on the other hand, the other approach, which uses an extractor based on sampling with replacement, comes with an additional loss that would blow up in the recursive multi-round analysis.

1 Introduction

1.1 Interactive Proofs and Special Sound Protocols

Interactive proofs play an important role in modern cryptography (and well beyond). They allow a prover $\mathcal{P}$ to convince a verifier $\mathcal{V}$ of the truth of a statement, i.e., that a certain instance x is an element of a given language L, without revealing any additional information (*zero-knowledge*), or with a proof that is (much) smaller than the statement (*succinctness*).

A crucial property of an interactive proof is *soundness*, which means that no dishonest prover $\mathcal{P}^*$ can convince the verifier $\mathcal{V}$ of a false statement (except with small probability), i.e., make $\mathcal{V}$ accept if $x \notin L$. However, in many situations a stronger property is needed: *knowledge soundness*, which informally means that in order to make $\mathcal{V}$ accept, not only has x to be in L, the prover actually needs to know a witness w attesting that $x \in L$ (considering L to be an NP language). One then also speaks of a *proof of knowledge*.

More formally, knowledge soundness requires the existence of a *knowledge extractor*, i.e., an efficient algorithm that outputs a witness when given rewindable oracle access to a (possibly dishonest) prover $\mathcal{P}^*$ that convinces a verifier $\mathcal{V}$ to accept with sufficiently high probability. In certain cases, ordinary soundness could be meaningless; this happens, e.g., when the considered language L is trivial, i.e., when every instance admits a witness. In such cases, knowledge soundness is the only meaningful soundness notion. For example, consider a prover claiming to know a hash collision for a hash function $H : \{0,1\}^* \to \{0,1\}^m$: here, the statement that H has a hash collision is vacuously true, and the nontrivial requirement is that the prover *actually knows* inputs $x \neq x'$ for which $H(x) = H(x')$.

Proving knowledge soundness is tricky in general; typically much harder than proving ordinary soundness. The problem is that different interactive proofs may require different extraction strategies. Thus, for a given protocol it is oftentimes unclear how to design a successful knowledge extractor, and/or how to analyze it. Furthermore, the formal definition sets a rather stringent requirement on the success probability of the extractor, which needs to hold for any dishonest prover. It is thus desirable to identify classes of interactive proofs for which there exist generic knowledge extraction results. An important example are *special-sound Σ-protocols*: classic results guarantee they are knowledge sound (with a soundness error determined by the size of the challenge set).

While interactive proofs studied in the past tended to be special-sound Σ-protocols, this is not the case anymore for many of the more recently proposed protocols, like Bulletproofs [BCC$^+$16, BBB$^+$18]. Motivated by this, the original result on special-sound Σ-protocols was extended in [ACK21, AF22, Att23] to k-special-sound Σ-protocols and to their multi-round variants, $(k_1, \ldots, k_\mu)$-special-sound protocols. However, many modern interative proofs [BBHR18, RR22, ACY23] – especially those employing Merkle-tree commitments – do not satisfy $(k_1, \ldots, k_\mu)$-special-soundness for reasonable parameters (in particular, the implied knowledge soundness would be far from optimal). Motivated by this, in the recent work [AFR23], the generalization of special-soundness is pushed further, and in some sense to the extreme, by introducing the notion of $(\Gamma_1, \ldots, \Gamma_\mu)$-special-sound interactive proofs. In spirit, while for k-special-sound Σ-protocols the special-soundness is specified by a *threshold* k, it is specified by a *general access structure* Γ in the case of Γ-special-sound Σ-protocols, and correspondingly for the multi-round variants. Besides introducing the definition, [AFR23] proves knowledge soundness and strong parallel repetition of (certain) $(\Gamma_1, \ldots, \Gamma_\mu)$-special-sound interactive proofs. One cannot expect efficient extractability to hold for all choices of the access structures $\mathbf{\Gamma} = (\Gamma_1, \ldots, \Gamma_\mu)$; thus, in more detail, [AFR23] identifies two relevant parameters $\kappa_{\mathbf{\Gamma}}$ and $t_{\mathbf{\Gamma}}$, determined solely by $\mathbf{\Gamma}$, and prove knowledge soundness with knowledge error $\kappa_{\mathbf{\Gamma}}$ for any $\mathbf{\Gamma}$-special-sound interactive proof for which $t_{\mathbf{\Gamma}}$ is polynomial.

1.2 The Fiat–Shamir Transformation

The *Fiat–Shamir transformation* is a powerful technique for turning (certain) *interactive* proofs, as discussed above, into *non-interactive* ones, or for designing signature schemes. Although originally suggested for Σ-protocols, it has become popular to apply the Fiat–Shamir transformation also to *multi-round* public-coin interactive proofs. Indeed, there has been a recent focus on interactive proofs with succinct communication, and those tend to have a non-constant number of rounds; the Fiat–Shamir transformation is then often used to avoid the increased round complexity (which could then actually form the bottle neck) by making the proof entirely non-interactive. Furthermore, for certain applications, it is crucial that a proof be non-interactive, making the use of the Fiat–Shamir transformation necessary.

However, the Fiat–Shamir transformation is not free. First of all, the security of the Fiat–Shamir transformed scheme is typically "only" proven in the random oracle model. Furthermore, there is often a non-trivial security loss involved. Indeed, until recently, the best reduction had a security loss on the knowledge error that is exponential in the number of rounds. More precisely, if the interactive proof Π has a knowledge error κ then the knowledge error of the Fiat–Shamir transformed non-interactive proof $\mathsf{FS}[\Pi]$ can be as bad as (roughly) $Q^\mu \cdot \kappa$, where Q is the number of oracle queries performed by the attacker, and μ the number of challenge rounds of the interactive proof Π. Only in the recent work [AFK22], it was shown that the class of $(k_1, \ldots, k_\mu)$-special-sound interactive proofs avoid this exponential security loss under the Fiat–Shamir transformation. As a matter of fact, the knowledge error of the Fiat–Shamir transformation of such a $(k_1, \ldots, k_\mu)$-special-sound protocol was shown to be $(Q + 1) \cdot \kappa$, independent of the number of rounds, where κ denotes the knowledge error of the interactive proof.

Unfortunately, while this notion of special-soundness covers certain interactive proofs with succinct communication, e.g., Bulletproofs, there are many interesting examples that fall outside this class. A notable example is a standard amortization technique, where a prover proves knowledge of n homomorphism preimages by proving knowledge of a random linear combination of these preimages. This amortization technique is used in many (lattice-based) interactive proofs to reduce the communication complexity, e.g., [BBC$^+$18, ALS20, ENS20, LNP22]. Unfortunately, it renders the corresponding threshold special-soundness parameter k too large to provide reasonable security guarantees. However, the amortization technique is $\varGamma$-special-sound for a $\varGamma$ with small parameters. Second, a common design principle is to first design an *interactive oracle proof* (IOP) with succinct communication, and to then instantiate the oracle with a hash-based Merkle-tree commitment. By construction, an interactive proof obtained via this recipe is not $(k_1, \dots, k_\mu)$-special-sound, at least not for reasonable parameters k_i, whereas it *can* be cast as a $(\varGamma_1, \dots, \varGamma_\mu)$-special-sound protocol with reasonable parameters.[6] As a final example, we note that the t-fold parallel repetition $\varPi^t$ of a k-special-sound $\varSigma$-protocol $\varPi$ is $((k-1)^t + 1)$-special-sound, i.e., the threshold special-soundness parameter grows exponentially in t, and thus this property does not imply knowledge soundness. However, $\varPi^t$ is also $\varGamma$-special-sound with parameter $t_\varGamma$ growing only linearly in t, i.e., the $\varGamma$-special-soundness property immediately implies knowledge soundness.

Given the current situation on the Fiat–Shamir front, the natural question that we address in this work is whether the techniques and the results from [AFK22] on the Fiat–Shamir transformation of $(k_1, \dots, k_\mu)$-special-sound protocols extend to this general notion of $(\varGamma_1, \dots, \varGamma_\mu)$-special-soundness introduced in [AFR23].

The short answer is: the results generalize, but not the techniques; the long answer follows below and in the rest of the paper.

1.3 Our Results

Indeed, in this paper, we show that the results of [AFK22], which show the security of the Fiat–Shamir transform of multi-round $(k_1, \dots, k_\mu)$-special-sound protocols with a security loss independent of the number of rounds, carry over to the generalization of special-sound (multi-round) protocols considered in [AFR23]. Thus, in detail, we show that the Fiat–Shamir transformation $\mathsf{FS}[\varPi]$ of any $(\varGamma_1, \dots, \varGamma_\mu)$-special-sound interactive proof $\varPi$, where $\mathbf{\Gamma} := (\varGamma_1, \dots, \varGamma_\mu)$ is such that the parameter $t_{\mathbf{\Gamma}}$ is polynomial, is knowledge sound with knowledge error

$$(Q+1) \cdot \kappa_{\mathbf{\Gamma}} \,, \tag{1}$$

where Q is the number of random oracle calls made by the attacker, and $\kappa_{\mathbf{\Gamma}}$ is the knowledge error of the interactive proof $\varPi$.[7] We stress the independence of the bound on μ, the number of challenge rounds of the protocol.

One might hope that this result can be obtained by generalizing the reasoning of [AFK22] to this generalized notion of $(\varGamma_1, \dots, \varGamma_\mu)$special-soundness. But this is not the case. The extractor considered in [AFK22] is based on sampling *without replacement*, and then properties of the *negative hypergeometric distribution* are used for the analysis; however, this use of the negative hypergeometric distribution is tailored to the threshold case and does not (seem to) generalize. Thus, a new extraction strategy is needed.

Fortunately, the extraction strategy from [AF22, AFR23] — which was so far considered in the *interactive* setting — comes to the rescue. These extractors work by sampling *with replacement*, together with an appropriately chosen, randomized stopping criteria, and using properties of the *geometric distribution* to analyze the success probability and the run time.

In more detail, as a first step, we show that for the 3-round case (i.e., for $\varSigma$-protocols), we can actually reduce the analysis of the considered extractor for the non-interactive Fiat–Shamir transformed protocol $\mathsf{FS}[\varPi]$ to that of (some variant of) the underlying interactive proof $\varPi$. Simply relying on the extraction properties shown in [AF22, AFR23] for the interactive proof $\varPi$, would then settle the 3-round case, showing that the Fiat–Shamir transformation of a $\varGamma$-special-sound $\varSigma$-protocol is a proof of knowledge with knowledge soundness $\kappa_\varGamma$ if the parameter $t_\varGamma$ is polynomial.

However, the multi-round case turns out to be more subtle. A naive approach would be to follow the same strategy as in the 3-round case, and reduce the non-interactive analysis to the analysis of a multi-round interactive proof. As we will discuss in Section 6.2, this approach is too restrictive and will result

[6] We note that modeling the hash function used in the Merkle tree commitment as a random oracle allows for so-called *straight-line extraction*. Hence, in this model rewinding can be avoided, which may result in a more efficient knowledge extractor. In fact, straight-line extraction is the standard approach for analyzing IOP-based proof systems. The notion $(\varGamma_1, \dots, \varGamma_\mu)$-special-soundness, together with its knowledge extractors, provides an alternative for analyzing their knowledge soundness.

[7] The definition of $\kappa_{\mathbf{\Gamma}}$ is provided in (14).

in an exponential security loss. For this reason, our approach is to follow a similar recursive strategy as in [AFK22], which uses the 3-round extractor and applies it recursively over the different rounds to extract the required tree of transcripts, using an "early-aborting" trick to achieve the round-independent expected runtime. Even though we can use a similar recursive approach with the same early-abort trick, we encountered multiple barriers in the analysis that we had to solve.

One problem of using the extraction strategy from [AF22, AFR23] for the 3-round case is that the success probability of the extractor suffers a factor t_Γ-loss, compared to the 3-round extractor used in [AFK22]. Since we anyway have to require this parameter to be polynomial for the extractor to be efficient, this is fine for the 3-round case; however, in the recursive analysis from [AFK22] of the extractor's success probability, the loss t_Γ negatively impacts the knowledge error. More precisely, this approach would introduce an additional factor $t_{\mathbf{\Gamma}} = \prod_{i=1}^{\mu} t_{\Gamma_i}$ loss in the knowledge error. Since $t_{\mathbf{\Gamma}}$ is required to be polynomial, this would actually be sufficient for proving knowledge soundness. However, in this work we aim to derive a *tight* knowledge error that is essentially equal to the trivial cheating probability of a dishonest prover.

Our solution is to apply the above recursive approach to a different base extractor for 3-round Γ-special-sound interactive proofs [KL23]. This base extractor is an improved variant of the ones presented in [AF22, AFR23], which does not suffer the factor t_Γ loss. Since the base extractor of [KL23] has not been published yet (it was communicated via personal communication), we provide a detailed description (Section 3.4) and complete analysis (Appendix B). This approach ensures the desired success probability of our knowledge extractor for the Fiat–Shamir transformation of $(\Gamma_1, \dots, \Gamma_\mu)$-special sound interactive proof Π.

To additionally achieve the required expected polynomial running time, we must however refine the running time analysis of this base extractor. This refinement is inspired by [AFK22], but requires new analysis techniques. As before, the difference is caused by the fact that our extractor uses sampling *with* replacement, whereas [AFK22] use sampling *without* replacement. The refined analysis allows us to apply the same kind of early-abort strategy as used in [AFK22], which significantly reduces the expected running time without compromising the extractor's success probability. Altogether, this proves our main result: the Fiat–Shamir transformation of $(\Gamma_1, \dots, \Gamma_\mu)$-special sound interactive proof Π is knowledge sound with knowledge error $(Q+1) \cdot \kappa_{\mathbf{\Gamma}}$, if the parameter $t_{\mathbf{\Gamma}}$ is polynomial. [8]

1.4 Related Work

We already discussed the related work [AF22, AFK22, AFR23], and their relevance and relation to this work, above.

Another closely related work is [Wik18], in which Wikström introduces and studies an abstraction of the problem of knowledge extraction when considering a generalized notion of special-soundness. His generalization resembles that of $(\Gamma_1, \dots, \Gamma_\mu)$-special-soundness studied in [AFR23], except that the access structure Γ_i is restricted to be the set of bases of a matroid $\mathbb{M}_i$. The main technical result [Wik18, Theorem 1] is the existence of a so-called *accepting basis extractor* for a *matroid tree*, with bounds on (1) the *extraction error* (which corresponds to the knowledge error), (2) the expected number of queries, and (3) the *tail bound.*

A central goal of [Wik18] is establishing good tail bounds (instead of only expected query bounds) with a precise analysis. The presented bounds are "convoluted expressions" [Wik18, p. 12] (and depend on tweakable parameters of the extractor). While an interpretation of the bounds is given, no simplifications are provided. This, together with the non-standard language and formalism used in [Wik18], makes a comparison of [Wik18] and [AFR23] that goes beyond the high-level similarities challenging.

In [Wik21], Wikström extends the language and results of [Wik18] to the Fiat–Shamir transformation. This amplifies the difficulties in comparing our work (which extends [AFR23]) with [Wik21] (which extends [Wik18]). In fact, we argue that the relative simplicity of our language, definitions, and theorem claims is a major differentiation. For completeness, we mention that [Wik21, Sect. 7] claims similar result as ours, i.e., security loss linear in the number of queries.

[8] Our main result (Theorem 4) is expressed in terms of $T_{\mathbf{\Gamma}} := \prod_{i=1}^{\mu}(t_{\Gamma_i} + 1)$ instead of $t_{\mathbf{\Gamma}}$. This is merely for notational convenience, since $t_{\mathbf{\Gamma}}$ is polynomial if and only if $T_{\mathbf{\Gamma}}$ (for nontrivial $t_{\Gamma_i} > 1$).

2 Preliminaries: Interactive and Non-Interactive Proofs

We recall here some standard definitions and concepts related to interactive and non-interactive proofs.

2.1 Interactive Proofs

We begin by introducing the necessary terminology for our discussion of interactive proofs.

Let $R \subseteq \{0,1\}^* \times \{0,1\}^*$ be a binary relation, which we view as a set of statement-witness pairs $(x; w)$. We let $R(x) = \{w : (x; w) \in R\}$ denote the set of valid witnesses for a statement x. Throughout, all relations are NP-relations, i.e., verifying $(x; w) \in R$ on input $(x; w)$ can be done in time polynomial in $|x|$ (thus, without loss of generality we also have $|w|$ polynomial in $|x|$). An *interactive proof* is an (interactive) protocol wherein a prover attempts to convince a verifier that a public statement x admits a (secret) witness $w \in R(x)$, or even that the prover *knows* such a witness.

If the verifier publishes all its random coins, the protocol is called *public-coin*. In such a case we may assume without loss of generality that all the messages from verifier to prover are uniformly random elements from some finite challenge set $\mathcal{C}$. The special case in which a public-coin interactive proof consists of 3 communication rounds, and in which the prover speaks in the first and third rounds, is termed a *Σ-protocol*.

An interactive proof is *complete* if, on public input x and private prover input $w \in R(x)$, the protocol execution will result in an accepting transcript (with high probability; in many protocols this probability is in fact 1). An interactive proof is *sound* if, on public input x that does not admit a witness w (i.e., $R(x) = \emptyset$), even for a potentially cheating prover $\mathcal{P}^*$ the probability that the protocol transcript is rejecting is large. The stronger notion of *knowledge soundness* informally requires that, on public input x, if a potentially cheating prover $\mathcal{P}^*$ manages to convince the verifier to accept with high enough probability, then it in fact must “know” a witness $w \in R(x)$. This is formalized in terms of a *knowledge extractor*, which is an expected polynomial-time algorithm that is able to extract a witness given blackbox access to such a prover $\mathcal{P}^*$. This is the main property of interactive proofs that we study in this work, so we provide a precise definition.

Definition 1 (Knowledge Soundness). *An interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ is* knowledge-sound *with* knowledege error $\kappa : \mathbb{N} \to [0,1]$ *if there exists a positive polynomial p and an algorithm $\mathcal{E}$, called a* knowledge extractor*, with the following properties. Given input x and blackbox access to $\mathcal{P}^*$, $\mathcal{E}$ runs in expected polynomial time (counting queries to $\mathcal{P}^*$ as one time-step) and outputs a witness $w \in R(x)$ with the following probability:*

$$\Pr\left(\mathcal{E}^{\mathcal{P}^*}(x) \in R(x)\right) \geq \frac{\epsilon(\mathcal{P}^*, x) - \kappa(|x|)}{p(|x|)} \ .$$

In the above, $\epsilon(\mathcal{P}^*, x) = \Pr\left((\mathcal{P}^*, \mathcal{V})(x) = \mathsf{accept}\right)$ is the success probability of $\mathcal{P}^*$ on input x.

Black-box access means in particular that $\mathcal{E}$ can run $\mathcal{P}^*$ multiple times. In case of a randomized $\mathcal{P}^*$, $\mathcal{E}$ can rerun $\mathcal{P}^*$ with the same randomness as in the previous run. This is referred to as *rewinding*.

Remark 1 (Interactive Arguments). In cryptography, one is often concerned with interactive *arguments*, i.e., interactive proofs where the soundness only holds against computationally bounded adversaries. In particular, computationally unbounded provers could convince the verifier with high probability. It thus may appear that our study of interactive proofs is not relevant for this setting. However, in practice most interactive arguments can in fact be cast as interactive proofs for the following “or” relation:

$$R' = \{(x; w) : (x; w) \in R \text{ or } w \text{ solves computational problem } X\} .$$

That is, knowledge soundness in this case guarantees that a successful prover either knows a witness for the given instance, or that it can solve some presumably hard computational problem (e.g., it outputs a discrete logarithm, a factor of an RSA modulus, a hash collision, etc.). Therefore, knowledge extractors for interactive proofs can typically be repurposed to prove knowledge soundness for interactive arguments as well.

2.2 Non-Interactive Random Oracle Proofs (NIROPs)

In certain applications, it is essential for proofs to be non-interactive. Certain interactive proofs can be made non-interactive via the Fiat–Shamir transformation (see below); the resulting non-interactive proof is then typically analyzed in the *random oracle model* (ROM). Proofs in the ROM are also referred to as non-interactive *random oracle proofs* (NIROPs). Below, we briefly recall the ROM, the formal definition of NIROPs and knowledge-soundness for NIROPs.

In the *random oracle model* (ROM), all algorithms have black-box access to an oracle $\mathsf{ro}\colon \{0,1\}^* \to \mathcal{Y}$, called the *random oracle*, which is instantiated as a uniformly random function. Typically $\mathcal{Y} = \{0,1\}^\eta$ for some $\eta \in \mathbb{N}$ related to the security parameter, but we find it more convenient to not make any restrictions on the codomain of the random oracle. For technical reasons, we limit the domain of the random oracle to be some arbitrary but finite message set $\mathcal{M}$.

One can naturally extend the ROM to allow $\mathcal{A}$ to access multiple independent random oracles $\mathsf{ro}_1, \ldots, \mathsf{ro}_\mu$, possibly with different codomains. This can be implemented from a single random oracle using standard techniques for domain separation and for any sampling algorithms, which admit reverse sampling or explainable sampling [LW22]. The following definitions and discussion may be extended to this multiple random oracle model in a natural way.

Definition 2 (Non-Interactive Random Oracle Proof (NIROP)). *A* non-interactive random oracle proof (NIROP) *for a relation R is a pair $(\mathcal{P}, \mathcal{V})$ of (probabilistic) random-oracle algorithms, a prover $\mathcal{P}$ and a polynomial-time verifier $\mathcal{V}$, such that the following holds. Given $(x; w) \in R$ and access to a random oracle* ro*, the prover $\mathcal{P}^{\mathsf{ro}}(x; w)$ outputs a proof π. Given $x \in \{0,1\}^*$, a purported proof π, and access to a random oracle* ro*, the verifier $\mathcal{V}^{\mathsf{ro}}(x, \pi)$ outputs* 0 *to reject or* 1 *to accept the proof.*

The definition of completeness is the natural one: honestly generated proofs indeed convince the verifier (with high probability). The basic stipulation of soundness is that it is infeasible for a prover to convince a verfier that a false statement is true, except with some small probability. In the non-interactive setting, the soundness error will depend on the number of queries that the cheating prover is permitted to make to the random oracle. The concept of knowledge soundness for NIROP's is the natural analogue of knowledge soundness for interactive proofs (Definition 1). We now formally define it.

Definition 3 (Knowledge Soundness - NIROP). *A non-interactive random oracle proof $(\mathcal{P}, \mathcal{V})$ for a relation R is* knowledge sound *with* knowledge error $\kappa : \mathbb{N} \times \mathbb{N} \to [0,1]$ *if there exists a positive polynomial p and an algorithm $\mathcal{E}$ – called a* knowledge extractor *– with the following properties. The extractor, given input x and oracle access to any (potentially dishonest) Q-query random oracle prover $\mathcal{P}^*$, runs in an expected number of steps that is polynomial in $|x|$ and Q and outputs a witness w such that, for all $x \in \{0,1\}^*$,*

$$\Pr\bigl(w \in R(x) : w \leftarrow \mathcal{E}^{\mathcal{P}^*}(x)\bigr) \geq \frac{\epsilon(\mathcal{P}^*, x) - \kappa(|x|, Q)}{p(|x|)}$$

where $\epsilon(\mathcal{P}^*, x) = \Pr\bigl(\mathcal{V}^{\mathsf{ro}}(x, \mathcal{P}^{*,\mathsf{ro}}) = 1\bigr)$. Here, $\mathcal{E}$ implements ro *for $\mathcal{P}^*$: in particular, it may arbitrarily program* ro*. Moreover, the randomness is over the randomness of $\mathcal{E}$, $\mathcal{V}$, $\mathcal{P}^*$ and* ro*.*

2.3 Fiat–Shamir Transformation

The Fiat–Shamir Transformation is a general-purpose operation that converts a given public-coin interactive proof into a non-interactive random oracle proof (NIROP). The idea is to replace the challenges from the verifier (which, recall, are without loss of generality uniformly random bit strings) by hashes of (some part of) the transcript up until that point. For concreteness, given a Σ-protocol with first message a and challenge c, the Fiat–Shamir transformed NIROP either sets $c = \mathsf{ro}(a)$ or $c = \mathsf{ro}(x, a)$, where x is the public input. The former definition is sufficient for *static* security where a malicious prover is given the input and then must attempt to convince the verifier, while the latter is required for *adaptive* security [AFK23, Definition 10] where the malicious prover may first choose the input x and subsequently attempt to forge a false proof [BPW12].

For multi-round protocols, there are multiple variants that one could consider. In this work, we will focus on the most conservative choice where all the prior prover messages are hashed along with the current round's message, i.e., the i-th challenge is computed as

$$c_i = \mathsf{ro}_i(a_1, \ldots, a_{i-1}, a_i)$$

where $a_1, \ldots, a_{i-1}, a_i$ are the i messages sent from the prover to the verifier so far, and ro_i is a random oracle with suitable codomain $\mathcal{C}_i$. Again in this multi-round setting the Fiat–Shamir transform comes with a statically secure and an adaptively secure variant: in the latter case, the statement x is included in the input for each hash function evaluation. As our results apply equally well to each variant, we will not explicitly spell out this distinction (static versus adaptive) in the rest of this work. A formal definition of the Fiat–Shamir transformation is included in Appendix A.

2.4 Geometric Distribution

Finally, as in prior works [AF22, AFR23], our extractor analysis relies on certain facts about the geometric distribution, which we now quickly recall.

A random variable B with two possible outcomes, denoted 0 (failure) and 1 (success), is said to follow a Bernoulli distribution with parameter p if $p = \Pr(B = 1)$. Sampling from a Bernoulli distribution is also referred to as running a Bernoulli trial. The probability distribution of the number X of independent and identical Bernoulli trials needed to obtain a success is called the geometric distribution with parameter $p = \Pr(B = 1)$. In this case, $\Pr(X = k) = (1-p)^{k-1}p$ for all $k \in \mathbb{N}$ and we write $X \sim \mathrm{Geo}(p)$. For two independent geometric distributions we have the following lemma.

Lemma 1 ([AFR23, Lemma 1]). *Let $X \sim \mathrm{Geo}(p)$ and $Y \sim \mathrm{Geo}(q)$ be independently distributed. Then,*

$$\Pr(X \leq Y) = \frac{p}{p+q-pq} \,.$$

The following simple argument shows that, if in a geometric experiment each trial is associated with a cost whose expected value has a constant upper bound, then the expected cost of the experiment is upper bounded by the expected number of trials times the upper bound of the cost for each trial.

Lemma 2. *Let $X \sim \mathrm{Geo}(p)$, let Z_i denote the cost of the i-th Bernoulli trial in the geometric experiment, and let $Z = Z_1 + Z_2 + \cdots + Z_X$ denote the total cost of running the geometric experiment. Then, if $\mathbb{E}[Z_i \mid X \geq i] \leq \theta$ for all i, it holds that*

$$\mathbb{E}[Z] \leq \frac{\theta}{p} \,.$$

Proof. Without loss of generality, we may assume that $\mathbb{E}[Z_i \mid X < i] = 0$. Then,

$$Z = \sum_{i=1}^{X} Z_i = \sum_{i=1}^{\infty} Z_i \,,$$

and

$$\begin{aligned}\mathbb{E}[Z_i] &= \Pr(X < i) \cdot \mathbb{E}[Z_i \mid X < i] + \Pr(X \geq i) \cdot \mathbb{E}[Z_i \mid X \geq i] \\ &\leq \Pr(X < i) \cdot 0 + \Pr(X \geq i) \cdot \theta = (1-p)^{i-1}\theta \,.\end{aligned}$$

Hence,

$$\mathbb{E}[Z] = \sum_{i=1}^{\infty} \mathbb{E}[Z_i] \leq \sum_{i=1}^{\infty} (1-p)^{i-1}\theta = \frac{\theta}{p} \,,$$

which completes the proof. □

3 Preliminaries: $\varGamma$-Special-Sound Protocols

The standard concept of *special-soundness* for $\varSigma$-protocols [Cra96, CD98] has recently seen many generalizations. Firstly, a $\varSigma$-protocol is *k-special-sound* for $k \in \mathbb{N}$ if one can efficiently construct a witness given k accepting transcripts with the same first message but pairwise distinct second messages (setting $k = 2$ recovers standard special-soundness). There are also generalizations to multi-round public-coin interactive proofs; in this case, an efficient procedure for constructing a witness given an appropriate *tree* of accepting transcripts. Very recently, a very general notion of $\varGamma$-special-soundness (and $(\varGamma_1, \ldots, \varGamma_\mu)$-special-soundness in the mutli-round case) was defined and shown to imply knowledge soundness [AFR23] under certain conditions. In our work, we will study the knowledge-soundness of non-interactive proofs in the random oracle model obtained by applying the Fiat–Shamir transform to such $(\varGamma_1, \ldots, \varGamma_\mu)$-special-sound protocols. We provide the precise definition below.

3.1 Γ-out-of-$\mathcal{C}$ Special-Soundness

Firstly, we must recall the definition of *monotone structures.*

Definition 4 (Monotone Structure). *Let $\mathcal{C}$ be a nonempty finite set and let $\Gamma \subseteq 2^{\mathcal{C}}$ be a family of subsets of $\mathcal{C}$. Then, $(\Gamma, \mathcal{C})$, or just Γ, is said to be a* monotone structure *if it is closed under taking supersets. That is, $S \in \Gamma$ and $S \subseteq T \subseteq \mathcal{C}$ implies $T \in \Gamma$.*

Note that $\emptyset \subseteq 2^{\mathcal{C}}$ and $2^{\mathcal{C}}$ *are* monotone structures according to our definition (which differs from some textbook definitions). We now provide the definition of Γ-out-of-$\mathcal{C}$ special-soundness.

Definition 5 (Γ-out-of-$\mathcal{C}$ Special-Soundness). *Let $(\Gamma, \mathcal{C})$ be a monotone structure. A 3-round public-coin interactive proof $(\mathcal{P}, \mathcal{V})$ for a relation R, with challenge set $\mathcal{C}$, is* Γ-out-of-$\mathcal{C}$ special sound *if there exists an algorithm that, on input a statement x and a set of accepting transcripts $(a, c_1, z_1), \dots, (a, c_k, z_k)$ with common first message a and such that $\{c_1, \dots, c_k\} \in \Gamma$, runs in polynomial time and outputs a witness $w \in R(x)$. We also say $(\mathcal{P}, \mathcal{V})$ is Γ-special-sound.*

Note that this definition recovers k-special soundness by taking Γ to be the family of subsets of $\mathcal{C}$ of size at least k.

Remark 2. Technically, the monotone structure $(\Gamma, \mathcal{C})$ of Definition 5 may depend on the input x. We should therefore refer to a family $(\Gamma_x, \mathcal{C}_x)_{x \in \{0,1\}^*}$ of monotone structures. For ease of notation, we will not make the dependency explicit and simply write $(\Gamma, \mathcal{C})$.

3.2 Some Concepts Related to Γ-Special-Sound Protocols

In this subsection we introduce some of the concepts and technical tools developed in [AFR23] which are used to analyze $(\Gamma, \mathcal{C})$-special sound protocols. We refer to this paper – particularly Section 4 – for additional context and motivation for these ideas.

Firstly, we require the concept of useful elements, which informally are elements that "bring us closer" to finding a set of challenges for which a (potentially cheating) prover $\mathcal{P}^*$ succeeds. That is, if the set of challenges the extractor has currently found is $S \subseteq \mathcal{C}$, $\mathcal{U}_\Gamma(S)$ is the set of challenges that could be useful in its quest to find a set of accepting transcripts $(a, c_1, z_1), \dots, (a, c_k, z_k)$ with $\{c_1, \dots, c_k\} \in \mathcal{C}$.

Definition 6 (Useful Elements). *For a monotone structure $(\Gamma, \mathcal{C})$ we define the following function:*

$$\mathcal{U}_\Gamma \colon 2^{\mathcal{C}} \to 2^{\mathcal{C}}, \quad S \mapsto \left\{ c \in \mathcal{C} \setminus S : \exists A \in \Gamma \text{ s.t. } S \subset A \wedge A \setminus \{c\} \notin \Gamma \right\}.$$

It is easily seen that

$$\mathcal{U}_\Gamma(B) \subseteq \mathcal{U}_\Gamma(A) \quad \text{for all } A \subseteq B\,. \tag{2}$$

Further, [AFR23, Lemma 3] shows that

$$\mathcal{C} \setminus \mathcal{U}_\Gamma(S) \notin \Gamma \quad \text{for all } S \notin \Gamma\,. \tag{3}$$

The efficiency of the knowledge extractor depends on how long it could take to find enough useful challenges. This is formalized by the t-value.

Definition 7 (t-value). *Let $(\Gamma, \mathcal{C})$ be a monotone structure and $S \subseteq \mathcal{C}$. Then*

$$t_\Gamma(S) := \max \left\{ t \in \mathbb{N}_0 : \begin{array}{c} \exists c_1, \dots, c_t \in \mathcal{C} \text{ s.t.} \\ c_i \in \mathcal{U}_\Gamma\big(S \cup \{c_1, \dots, c_{i-1}\}\big) \ \forall i \end{array} \right\}.$$

Further,

$$t_\Gamma := t_\Gamma(\emptyset)\,.$$

Observe that $t_\Gamma(S) = 0$ if and only if $S \in \Gamma$ or $\Gamma = \emptyset$. The basic fact that we require is that adding an element $c \in \mathcal{U}_\Gamma(S)$ to S decreases the t-value.

Lemma 3 ([AFR23, Lemma 4]). *Let $(\Gamma, \mathcal{C})$ be a nonempty monotone structure and let $S \subseteq \mathcal{C}$ such that $S \notin \Gamma$. Then, for all $c \in \mathcal{U}_\Gamma(S)$,*

$$t_\Gamma(S \cup \{c\}) < t_\Gamma(S)\,.$$

3.3 Knowledge Soundness of $\varGamma$-Special Sound $\varSigma$-Protocols

For any $\varGamma$-special sound $\varSigma$-protocol $\varPi$, Theorem 1 of [AFR23] proves the existence of a knowledge extractor that makes an expected number of at most $2t_\varGamma - 1$ queries to the considered prover and successfully extracts a witness with probability at least

$$\frac{1}{t_\varGamma} \cdot \frac{\epsilon - \kappa_\varGamma}{1 - \kappa_\varGamma}$$

where ϵ is the success probability of the considered prover, and

$$\kappa_\varGamma := \max_{S \notin \varGamma} \frac{|S|}{|\mathcal{C}|} \,. \tag{4}$$

Looking ahead, we will reduce the existence of a knowledge extractor for the Fiat–Shamir transformation of a $\varGamma$-special sound $\varSigma$-protocol to the existence of a knowledge extractor for the original protocol $\varPi$, which is provided by Theorem 1 of [AFR23] (if $t_\varGamma$ is polynomial). However, it turns out that when trying to extend our result to the multi-round case (which we do by applying the 3-round case recursively), the additional denominator $t_\varGamma$ in the above success probability is problematic; indeed, it results in an unwanted (and unnecessary) blowup of the knowledge error. For this reason, we will consider an improved version of the above extractor, proposed in [KL23], which avoids the $t_\varGamma$ in the denominator. We discuss this improved extractor in detail below.

3.4 An Improved Extractor for $\varGamma$-Special Sound $\varSigma$-Protocols

Here, we discuss the improved knowledge extractor from the unpublished manuscript [KL23], shared in personal communication, that avoids the factor $1/t_\varGamma$ loss in success probability.

Similar to the recent works on the topic, we now continue the discussion in a more abstract language. Consider an algorithm $\mathcal{A}\colon \mathcal{C} \to \{0,1\}^*$, as well as a verification predicate $\mathcal{V}\colon \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. Naturally, we would instantiate $\mathcal{A}$ with a dishonest prover $\mathcal{P}^*$, attacking a $\varGamma$-out-of-$\mathcal{C}$ special-sound $\varSigma$-protocol, with the understanding that the output (a, z) of $\mathcal{A}(c)$ consists of the two messages a and z produced by $\mathcal{P}^*$ on challenge c. By a standard averaging argument we may assume the dishonest prover $\mathcal{P}^*$, and thus the algorithm $\mathcal{A}$, to be deterministic. See for instance [AF22] for a formal proof of this claim. Note that, under this assumption, the first message a is fixed and independent of the challenge c. We call an output $y \leftarrow \mathcal{A}(c)$ *accepting* or *correct* if $\mathcal{V}(c, y) = 1$. For C uniformly random over $\mathcal{C}$, the *success probability* of $\mathcal{A}$ is denoted as

$$\epsilon^{\mathcal{V}}(\mathcal{A}) := \Pr\bigl(\mathcal{V}(C, \mathcal{A}(C)) = 1\bigr)\,;$$

this then obviously coincides with the success probability of $\mathcal{P}^*$. The goal is to design an extractor, with black-box access to $\mathcal{A}$, that finds accepting y's for challenges $c_1, \dots, c_k$ that form a set in $\varGamma$.

The extractor proposed in [KL23] is given in Figure 1 below. In spirit, it is quite similar to the extractor considered in [AFR23]: it first runs $\mathcal{A}$ on a random challenge c and aborts if $\mathcal{A}$ fails to produce an accepting transcript; otherwise, if $\mathcal{A}$ has produced an accepting transcript, the extractor runs a geometric experiment with the goal of finding enough additional accepting transcripts. These two phases of the extractor are respectively denoted $\mathcal{E}^{\mathcal{A}}_{\text{init},\varGamma}$ and $\mathcal{E}^{\mathcal{A}}_{\text{search},\varGamma}$.

A crucial difference is that, in contrast to the extractor of [AFR23], the coin toss in the geometric experiments of in [KL23] depends on the challenges collected so far. More precisely, in [AFR23] the probability that the coin returns 1 equals the success probability of the first $\mathcal{A}$-invocation of the extractor, and thus the probability of entering the second phase of the extraction. This yields a simple running time analysis, but incurs the factor $t_\varGamma$ loss in the success probability. In [KL23], the probability that the coin returns 1 is chosen more carefully, which leads to a (much) smaller probability for the coin returning 1, thus reducing the probability that the extractor aborts before succeeding, which in turn improves the success probability of the extractor (while still being able to control the expected running time sufficiently).

Further, the extractor of [AFR23] recursively invokes a (sub)extractor for a larger monotone structure. As a consequence, the invocation of a (sub)extractor might fail even after some accepting transcripts have been found, in which case these accepting transcripts are "forgotten", and either the extractor aborts or

Fig. 1. Extractor $\mathcal{E}_{\Gamma}^{\mathcal{A}}$ of [KL23].

Parameters: a nonempty monotone structure $(\Gamma, \mathcal{C})$.
Oracle access to: algorithm $\mathcal{A}\colon \mathcal{C} \to \{0,1\}^*$ and verification predicate $\mathcal{V}\colon \mathcal{C} \times \{0,1\}^* \to \{0,1\}$.
Output: $\perp$ or $(c_1, \dots, c_k, y_1, \dots, y_k)$ with $\{c_1, \dots, c_k\} \in \Gamma$ and $\mathcal{V}(c_j, y_j) = 1 \quad \forall j$.

```
                                                                          $\boxed{\mathcal{E}_{\text{init},\Gamma}^{\mathcal{A}}}$
1. Sample $c_1 \in \mathcal{C}$ uniformly at random and evaluate $y_1 \leftarrow \mathcal{A}(c_1)$.
2. If $\mathcal{V}(c_1, y_1) = 0$, abort and output $\perp$.
3. If $\mathcal{V}(c_1, y_1) = 1$ and $\{c_1\} \in \Gamma$, output $(c_1, y_1) \in \mathcal{C} \times \{0,1\}^*$.
                                                                          $\boxed{\mathcal{E}_{\text{search},\Gamma}^{\mathcal{A}}(c_1, y_1)}$
4. Else set $S_1 = \{c_1\} \subseteq \mathcal{C}$, $i = 1$ and COIN $= 0$.
5. Repeat:
     – sample $c_{i+1} \in \mathcal{U}_\Gamma(S_i)$ uniformly at random and evaluate $y_{i+1} \leftarrow \mathcal{A}(c_{i+1})$;
         • if $\mathcal{V}(c_{i+1}, y_{i+1}) = 1$, set $S_{i+1} := S_i \cup \{c_{i+1}\}$ and then set $i = i + 1$;
         • else sample $d \in \mathcal{U}_\Gamma(S_{i-1})$, respectively in $d \in \mathcal{C}$ in case $i = 1$, uniformly at random and set
           COIN $= 1$ if $d \notin \mathcal{U}_\Gamma(S_i)$ and $\mathcal{V}\big(d, \mathcal{A}(d)\big) = 1$;
   until $S_i \in \Gamma$ or until COIN $= 1$.
6. If COIN $= 1$, abort and output $\perp$.
7. If $S_i = \{c_1, \dots, c_i\} \in \Gamma$, output $(c_1, \dots, c_i, y_1, \dots, y_i)$ with $\mathcal{V}(c_j, y_j) = 1$ for all $1 \leq j \leq i$.
```

it invokes the subextractor again with fresh randomness. By contrast, the extractor of [KL23] collects the challenges iteratively, and accepting transcripts are never forgotten.

Additionally, when analyzing the geometric experiment, [KL23] expresses the relevant bounds in terms of the challenges collected so far, while [AFR23] uses a worst case bound over all possibilities.

Altogether, this allows [KL23] to argue the following extractability result, which, compared to [AFR23, Theorem 1], avoids the t_Γ in the denominator of the success probability. For completeness, we provide a proof of Theorem 1 in Appendix B.

Theorem 1 (Extraction Algorithm - Σ-protocols [KL23]). *Let $(\Gamma, \mathcal{C})$ be a nonempty monotone structure and let $\mathcal{V}\colon \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. Then there exists an oracle algorithm $\mathcal{E}_\Gamma$ with the following properties: The algorithm $\mathcal{E}_\Gamma^{\mathcal{A}}$, given oracle access to a (probabilistic) algorithm $\mathcal{A}\colon \mathcal{C} \to \{0,1\}^*$, requires an expected number of at most $1 + t_\Gamma \cdot (1 + \kappa_\Gamma)$ queries to $\mathcal{A}$ and, with probability at least*

$$\frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - \kappa_\Gamma}{1 - \kappa_\Gamma},$$

it outputs pairs $(c_1, y_1), (c_2, y_2), \dots, (c_k, y_k) \in \mathcal{C} \times \{0,1\}^*$ with $\mathcal{V}(c_i, y_i) = 1$ for all i and $\{c_1, \dots, c_k\} \in \Gamma$.

Remark 3. In contrast to the extractors of [AFR23] and [KL23], the knowledge extractor of Figure 1 samples the first challenge uniformly at random from $\mathcal{C}$ rather than from $\mathcal{U}_\Gamma(\emptyset)$. This minor adaptation may cause the first challenge c_1 found by the extractor to be useless, i.e., c_1 may be in $\mathcal{C} \setminus \mathcal{U}_\Gamma(\emptyset)$. As a consequence, the expected number of $\mathcal{A}$-queries is bounded by $1 + t_\Gamma \cdot (1 + \kappa_\Gamma)$ instead of the slightly smaller bound $1 + (t_\Gamma - 1) \cdot (1 + \kappa_\Gamma)$. The reason for this seemingly suboptimal design choice is to simplify the analysis of the Fiat–Shamir transformation. More precisely, we will see that, due to this design choice, Theorem 1 can be deployed in a black-box manner when analyzing the Fiat–Shamir transformation of Γ-special-sound Σ-protocols. Note that only for contrived examples of interactive proofs does it hold that $\mathcal{C} \neq \mathcal{U}_\Gamma(\emptyset)$. Hence, the above discussion can be avoided by the reasonable assumption that $\mathcal{C} = \mathcal{U}_\Gamma(\emptyset)$, as is done in [KL23].

4 Refined Running Time Analysis of the Extractor from Theorem 1

In this section, we will refine the running time analysis of the knowledge extractor of Figure 1. Instead of simply counting the (expected) number of $\mathcal{A}$-invocations, we associate $\mathcal{A}$ with a cost function $\theta\colon \mathcal{C} \to \mathbb{R}_{\geq 0}$, such that $\theta(c)$ denotes the cost of evaluating $\mathcal{A}(c)$. We can thereby give a tighter bound on the running time in scenarios where some $\mathcal{A}$-invocations are more costly than others. This refinement turns out to be essential when considering Fiat–Shamir transformations of multi-round interactive proofs.

Lemma 4 (Refined Running Time Analysis - IP Extractor). *Let $(\mathit{\Gamma}, \mathcal{C})$ be a nonempty monotone structure and $\mathcal{A}\colon \mathcal{C} \to \{0,1\}^*$ an algorithm accompanied by a verification predicate $\mathcal{V}\colon \mathcal{C}\times\{0,1\}^* \to \{0,1\}$. Further, let $\theta\colon \mathcal{C} \to \mathbb{R}_{\geq 0}$, such that $\theta(c)$ denotes the cost of evaluating $\mathcal{A}(c)$.*

Then the expected cost of the $\mathcal{A}$-invocations of the extractor $\mathcal{E}_{\mathit{\Gamma}}^{\mathcal{A}}$ of Figure 1 is at most

$$\mathbb{E}[\theta(C)] + t_{\mathit{\Gamma}} \cdot \frac{\mathbb{E}[\theta(C)]}{1-\kappa_{\mathit{\Gamma}}}$$

where C is distributed uniformly at random over $\mathcal{C}$.

Before we prove this lemma, we introduce some convenient notation. We also note that even though it is not necessary, it may be didactically helpful to first verify the non-refined analysis in Appendix B.

For any $0 < k \in \mathbb{Z}$, $\mathbf{c} = (c_1, \dots, c_k) \in \mathcal{C}^k$ and $1 \leq i \leq k$, we write $\mathbf{c}_i = (c_1, \dots, c_i)$. We will abuse notation by occasionally interpreting the vector $\mathbf{c}$ as the subset of challenges containing the coordinates of $\mathbf{c}$, i.e., the statement $\mathbf{c} \in \mathit{\Gamma} \subseteq 2^{\mathcal{C}}$ is interpreted as $\{c_1, \dots, c_k\} \in \mathit{\Gamma}$. Further, to simplify notation, we write V for the event $\mathcal{V}\big(C, \mathcal{A}(C)\big) = 1$ and U_i for the event $C \in \mathcal{U}_{\mathit{\Gamma}}(\mathbf{c}_i)$, where C is distributed uniformly at random over $\mathcal{C}$ and $\mathbf{c}_i$ is given by the context. Additionally, U_0 denotes the event $C \in \mathcal{C}$, i.e., $\Pr(U_0) = 1$.[9] Recall that $\mathcal{U}_{\mathit{\Gamma}}(\mathbf{c}) \subseteq \mathcal{U}_{\mathit{\Gamma}}(\mathbf{c}_{k-1}) \subseteq \cdots \subseteq \mathcal{U}_{\mathit{\Gamma}}(\mathbf{c}_1) \subseteq \mathcal{C}$ (Equation (2)), or, in terms of probability events,

$$U_k \implies U_{k-1} \implies \cdots \implies U_0 \,. \tag{5}$$

The following three quantities will play a crucial role in the analysis:

$$\begin{aligned}
\delta(\mathbf{c}_i) &= \Pr\big(V \mid U_i\big)\,, \\
\mathit{\Delta}(\mathbf{c}_i) &= \Pr\big(V \wedge \neg U_i \mid U_{i-1}\big)\,, \\
\tilde{\delta}(\mathbf{c}_i) &= \frac{\delta(\mathbf{c}_{i-1}) - \mathit{\Delta}(\mathbf{c}_i)}{1 - \mathit{\Delta}(\mathbf{c}_i)}\,.
\end{aligned} \tag{6}$$

The first two quantities represent certain parameters of the considered extractor from Figure 1. Namely, $\delta(\mathbf{c}_i)$ is the probability that $\mathcal{V}(c_{i+1}, y_{i+1}) = 1$ in the repeat loop in step 5, given that the extractor has already found some set (or vector) of accepting challenges $\mathbf{c}_i = (c_1, \dots, c_i)$. On the other hand, $\mathit{\Delta}(\mathbf{c}_i)$ is the probability for the considered coin to become 1 (which then means that the extractor stops unsuccessfully). Hence, $\delta(\mathbf{c}_i)$ and $\mathit{\Delta}(\mathbf{c}_i)$ are the parameters of the two geometric experiments that are run in parallel in step 5, considering the challenges collected so far. Additionally, $\tilde{\delta}(\mathbf{c}_i)$ is an auxiliary quantity that will be relevant in the analysis; for instance, the following shows that it lower bounds $\delta(\mathbf{c}_i)$:

$$\begin{aligned}
\delta(\mathbf{c}_i) = \Pr(V \mid U_i) &= \frac{\Pr(V \wedge U_i \mid U_{i-1})}{\Pr(U_i \mid U_{i-1})} = \frac{\Pr(V \wedge U_i \mid U_{i-1})}{1 - \Pr(\neg U_i \mid U_{i-1})} \geq \frac{\Pr(V \wedge U_i \mid U_{i-1})}{1 - \Pr(V \wedge \neg U_i \mid U_{i-1})} \\
&= \frac{\Pr(V \wedge U_i \mid U_{i-1})}{1 - \mathit{\Delta}(\mathbf{c}_i)} = \frac{\Pr(V \mid U_{i-1}) - \Pr(V \wedge \neg U_i \mid U_{i-1})}{1 - \mathit{\Delta}(\mathbf{c}_i)} = \frac{\delta(\mathbf{c}_{i-1}) - \mathit{\Delta}(\mathbf{c}_i)}{1 - \mathit{\Delta}(\mathbf{c}_i)} = \tilde{\delta}(\mathbf{c}_i)\,.
\end{aligned} \tag{7}$$

Furthermore, exploiting this inequality $\delta(\mathbf{c}_i) \geq \tilde{\delta}(\mathbf{c}_i)$, and using that $\mathit{\Delta}(\mathbf{c}_i) \leq 1$, we have

$$\delta(\mathbf{c}_i) + \mathit{\Delta}(\mathbf{c}_i) - \mathit{\Delta}(\mathbf{c}_i)\delta(\mathbf{c}_i) \geq \tilde{\delta}(\mathbf{c}_i) + \mathit{\Delta}(\mathbf{c}_i) - \mathit{\Delta}(\mathbf{c}_i)\tilde{\delta}(\mathbf{c}_i) = \delta(\mathbf{c}_{i-1})\,, \tag{8}$$

where the equality is obtained by solving the definition of $\tilde{\delta}(\mathbf{c}_i)$ for $\delta(\mathbf{c}_{i-1})$. Similarly,

$$\frac{\delta(\mathbf{c}_i) + \mathit{\Delta}(\mathbf{c}_i) - \mathit{\Delta}(\mathbf{c}_i)\delta(\mathbf{c}_i)}{\delta(\mathbf{c}_i)} \leq \frac{\tilde{\delta}(\mathbf{c}_i) + \mathit{\Delta}(\mathbf{c}_i) - \mathit{\Delta}(\mathbf{c}_i)\tilde{\delta}(\mathbf{c}_i)}{\tilde{\delta}(\mathbf{c}_i)} = \frac{\delta(\mathbf{c}_{i-1})}{\tilde{\delta}(\mathbf{c}_i)}\,. \tag{9}$$

Equation (8) is used in the running time analysis (below and in Appendix B), while Equation (9) is used for the analysis of the success probability (in Appendix B).

We are now ready for the proof of Lemma 4.

[9] This convention reflects that the extractor sampling the first challenge from $\mathcal{C}$, rather than from $\mathcal{U}_{\mathit{\Gamma}}(\emptyset)$.

Proof of Lemma 4. Let us write $t = t_\Gamma$. For $1 \le i \le t$, let the random variable C_i denote the i-th successful challenge found by the extractor, where we let $C_i = \bot$ if the extractor finishes before finding i challenges, i.e., C_i has support in $\mathcal{C} \cup \{\bot\}$. Note that if $\boldsymbol{C}_i := (C_1, \dots, C_i) \in \Gamma$ then $C_{i+1} = \dots = C_t = \bot$. Vice versa, $C_i = \bot$ implies that the extractor was either successful before the i-th iteration, or that it aborted and failed before the i-th iteration.

For $1 \le i < t$, let $\mathbf{c}_i \in \mathcal{C}^i$ with $\mathbf{c}_i \notin \Gamma$ and $\Pr[\boldsymbol{C}_i = \mathbf{c}_i] > 0$. Then, conditioning on $[\boldsymbol{C}_i = \mathbf{c}_i]$ means that we consider a case where the extractor has found i challenges, but it needs at least one more. In order to do so, it runs two geometric experiments in parallel with parameters $\delta(\mathbf{c}_i)$ and $\Delta(\mathbf{c}_i)$, trying to find the $(i+1)$-th challenge (step 5). The probability that at least one of the experiments finishes in a single trial equals

$$p := 1 - \big(1 - \delta(\mathbf{c}_i)\big)\big(1 - \Delta(\mathbf{c}_i)\big) = \delta(\mathbf{c}_i) + \Delta(\mathbf{c}_i) - \Delta(\mathbf{c}_i)\delta(\mathbf{c}_i) \ge \delta(\mathbf{c}_{i-1}) \,,$$

where the inequality follows from Equation (8). Thus, this run of two geometric experiments is again a geometric experiment, but now with parameter p.

Next, we evaluate the cost of this (combined) geometric experiment. For this purpose we note that to determine the value of the coin, the extractor only needs to invoke $\mathcal{A}$ if $d \notin \mathcal{U}_\Gamma(\mathbf{c}_i)$. Hence, recalling that the distribution of c_{i+1} is that of the random variable C conditioned on U_i, and that d is distributed as C conditioned on U_{i-1}, in each trial the expected cost of the $\mathcal{A}$-invocations is at most

$$\begin{aligned}
&\mathbb{E}[\theta(C) \mid U_i] + \Pr(\neg U_i \mid U_{i-1}) \cdot \mathbb{E}[\theta(C) \mid U_{i-1} \wedge \neg U_i] \\
&= \frac{\Pr\big(U_i\big)}{\Pr\big(U_i\big)} \cdot \mathbb{E}[\theta(C) \mid U_i] + \frac{\Pr(U_{i-1} \wedge \neg U_i)}{\Pr(U_{i-1})} \cdot \mathbb{E}[\theta(C) \mid U_{i-1} \wedge \neg U_i] \\
&\le \frac{1}{\Pr(U_i)} \Big(\Pr(U_i) \cdot \mathbb{E}[\theta(C) \mid U_i] + \Pr(U_{i-1} \wedge \neg U_i) \cdot \mathbb{E}[\theta(C) \mid U_{i-1} \wedge \neg U_i] \Big) \\
&\le \frac{\mathbb{E}[\theta(C)]}{1 - \Pr(\neg U_i)} \le \frac{\mathbb{E}[\theta(C)]}{1 - \kappa_\Gamma} \,,
\end{aligned}$$

where, in the first inequality, we use that $U_i \implies U_{i-1}$ and thus $\Pr(U_i) \le \Pr(U_{i-1})$. Moreover, the second inequality follows since $\theta(c) \ge 0$ for all $c \in \mathcal{C}$, and the last inequality follows from the definition of κ_Γ, exploiting that $\mathbf{c}_i \notin \Gamma$ and thus $\mathcal{C} \setminus \mathcal{U}_\Gamma(\mathbf{c}_i) \notin \Gamma$ (Equation (3)).

Hence, if we let Y_i denote the cost of the $\mathcal{A}$-queries that the extractor makes in its i-th iteration, i.e., when trying to find the i-th challenge, then

$$\mathbb{E}[Y_1] = \mathbb{E}[\theta(C)] \quad \text{and} \quad \mathbb{E}[Y_{i+1} \mid \boldsymbol{C}_i = \mathbf{c}_i] \le \frac{\mathbb{E}[\theta(C)]}{\delta(\mathbf{c}_{i-1}) \cdot (1 - \kappa_\Gamma)} \,,$$

where the upper-bound is the product of the expected number of trials and the expected cost per trial (by Lemma 2). On the other hand, if $\mathbf{c}_i$ is such that $\mathbf{c}_i \in \Gamma$ or $c_i = \bot$ (in either case, the extractor is done) then $\mathbb{E}[Y_{i+1} \mid \boldsymbol{C}_i = \mathbf{c}_i] = 0$.

Putting these observations together shows that for all $1 \le k \le t$

$$\mathbb{E}[Y_{k+1}] = \sum_{\mathbf{c}_k \in T_k} \Pr(\boldsymbol{C}_k = \mathbf{c}_k) \cdot \mathbb{E}[Y_{k+1} \mid \boldsymbol{C}_k = \mathbf{c}_k] \le \frac{\mathbb{E}[\theta(C)]}{1 - \kappa_\Gamma} \cdot \sum_{\mathbf{c}_k \in T_k} \frac{\Pr(\boldsymbol{C}_k = \mathbf{c}_k)}{\delta(\mathbf{c}_{k-1})} \,.$$

where $T_k := \{\mathbf{c}_k \in \mathcal{C}^k : \mathbf{c}_k \notin \Gamma \wedge \Pr(\boldsymbol{C}_k = \mathbf{c}_k) > 0\}$.

We will now expand the probability $\Pr(\boldsymbol{C}_k = \mathbf{c}_k)$, aiming for an upper bound on the above sum. For any $\mathbf{c}_k \in T_k$ and $1 \le i < k$, by Lemma 1 it holds that

$$\Pr\big(C_{i+1} \neq \bot \mid \boldsymbol{C}_i = \mathbf{c}_i\big) = \frac{\delta(\mathbf{c}_i)}{\delta(\mathbf{c}_i) + \Delta(\mathbf{c}_i) - \Delta(\mathbf{c}_i)\delta(\mathbf{c}_i)} \le \frac{\delta(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} \,,$$

where the last inequality applies Equation (8). Noting that $c_{i+1} \neq \bot$ by definition of T_k, it follows that

$$\begin{aligned}
\Pr\big(C_{i+1} = c_{i+1} \mid \boldsymbol{C}_i = \mathbf{c}_i\big) &= \Pr\big(C_{i+1} = c_{i+1} \wedge C_{i+1} \neq \bot \mid \boldsymbol{C}_i = \mathbf{c}_i\big) \\
&= \tau(c_{i+1} \mid \mathbf{c}_i) \cdot \Pr\big(C_{i+1} \neq \bot \mid \boldsymbol{C}_i = \mathbf{c}_i\big)
\end{aligned}$$

$$\leq \tau(c_{i+1} \mid \mathbf{c}_i) \cdot \frac{\delta(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})}\,.$$

where,

$$\tau(c_{i+1} \mid \mathbf{c}_i) := \Pr\big(C_{i+1} = c_{i+1} \mid C_{i+1} \neq \bot \wedge \boldsymbol{C}_i = \mathbf{c}_i\big)\,.$$

Hence, it follows that

$$\Pr(\boldsymbol{C}_k = \mathbf{c}_k) = \prod_{i=1}^{k} \Pr(C_i = c_i \mid \boldsymbol{C}_{i-1} = \mathbf{c}_{i-1}) = \delta(\mathbf{c}_{k-1}) \cdot \prod_{i=1}^{k} \tau(c_i \mid \mathbf{c}_{i-1})\,.$$

Hence, for all $k \geq 1$,

$$\begin{aligned}
\mathbb{E}[Y_{k+1}] &\leq \frac{\mathbb{E}[\theta(C)]}{1-\kappa_\Gamma} \cdot \sum_{\mathbf{c}_k \in T_k} \prod_{i=1}^{k} \tau(c_i \mid \mathbf{c}_{i-1}) \\
&\leq \frac{\mathbb{E}[\theta(C)]}{1-\kappa_\Gamma} \cdot \sum_{c_1 \in \mathcal{C}} \tau(c_1 \mid \mathbf{c}_0) \sum_{c_2 \in \mathcal{C}} \tau(c_2 \mid \mathbf{c}_1) \cdots \sum_{c_k \in \mathcal{C}} \tau(c_k \mid \mathbf{c}_{k-1}) \\
&\leq \frac{\mathbb{E}[\theta(C)]}{1-\kappa_\Gamma}\,.
\end{aligned}$$

Hence, the expected running time of the extractor is at most

$$\mathbb{E}[\theta(C)] + \sum_{k=1}^{t} \mathbb{E}[Y_{k+1}] \leq \mathbb{E}[\theta(C)] + t \cdot \frac{\mathbb{E}[\theta(C)]}{1-\kappa_\Gamma}\,,$$

which completes the proof of the lemma. □

5 The Fiat–Shamir Transformation of Γ-Special-Sound Σ-Protocols

In this section, we analyze the knowledge soundness of the Fiat–Shamir transformation of Γ-out-of-$\mathcal{C}$ special-sound Σ-protocols, i.e., we first restrict ourselves to 3-round interactive proofs. In Section 6, we will generalize the analysis to multi-round $(\Gamma_1, \ldots, \Gamma_\mu)$-out-of-$(\mathcal{C}_1, \ldots, \mathcal{C}_\mu)$ special-sound interactive proofs. In both cases, we show that the security loss of the Fiat–Shamir transformation is linear in the query complexity Q of a dishonest prover attacking the non-interactive random oracle proof; in particular, the security loss is independent of the number of rounds. In comparison, in general the security loss of the Fiat–Shamir transformation may be exponential in the number of rounds.

In Section 5.1, we will set the stage and introduce the required notation and auxiliary lemmas. In Section 5.2, we will present our knowledge extractor and analyze its properties. Subsequently, to prepare for our analysis of Fiat–Shamir transformations of multi-round interactive proofs, we will provide a refined running time analysis in Section 5.3.

5.1 Preliminary Discussion

Following the notation of prior works, and in line with the notation used in Section 3 for *interactive* proofs, we present our core results in an abstract language. More precisely, we define an abstract algorithm $\mathcal{A}$ by which we capture the behavior of a (dishonest) Q-query prover $\mathcal{P}^*$ attacking the non-interactive proof. A prover attacking the interactive proof receives a challenge c and aims to provide an accepting response. In the non-interactive setting, the prover $\mathcal{P}^*$ receives (access to) a random oracle $\mathsf{ro}\colon \mathcal{M} \to \mathcal{C}$, where $\mathcal{M}$ is some finite set containing all potential first messages a of the interactive proof.

On input a statement x, and after making at most Q queries to ro, the prover $\mathcal{P}^*$ outputs a proof $\pi = (a, z)$, which is accepting if and only if (a, c, z) is an accepting transcript, where $c = \mathsf{ro}(a)$.[10]

[10] In the adaptive setting, the prover does not receive an input and outputs the statement x together with a proof $\pi = (a, z)$. In this case, one should define the challenge as $c = \mathsf{ro}(x, a)$.

The algorithm $\mathcal{A}$, which captures the behavior of a prover attacking the non-interactive proof, is hence of the form

$$\mathcal{A}\colon \mathcal{C}^{\mathcal{M}} \to \mathcal{M} \times \{0,1\}^* , \quad \mathsf{ro} \mapsto (a, z) ,$$

where $\mathcal{C}^{\mathcal{M}}$ denotes the set of all functions with domain $\mathcal{M}$ and codomain $\mathcal{C}$. Where convenient, we also write $\mathcal{A} = (A, Z)$ such that $A(\mathsf{ro})$ and $Z(\mathsf{ro})$ denote the first and second output of $\mathcal{A}(\mathsf{ro})$, respectively. As before,

$$\mathcal{V}\colon \mathcal{M} \times \mathcal{C} \times \{0,1\}^* \to \{0,1\} , \quad (a, c, z) \mapsto v$$

denotes the verification predicate for the underlying *interactive* proof. This verification predicate naturally extends to a verification predicate

$$\mathcal{V}^{\mathsf{ro}}\colon \mathcal{M} \times \{0,1\}^* \to \{0,1\} , \quad (a, z) \mapsto \mathcal{V}(a, \mathsf{ro}(a), z) ,$$

for the non-interactive random oracle proof. By a slight abuse of nation, we sometimes reorder the inputs, so as to write $\mathcal{V}\big(c, \mathcal{A}(\mathsf{ro})\big)$ for $\mathcal{V}(a, c, z)$ with $(a, z) \leftarrow \mathcal{A}(\mathsf{ro})$. The algorithm $\mathcal{A}$ now has a naturally defined success probability

$$\epsilon^{\mathcal{V}}(\mathcal{A}) = \Pr\big(\mathcal{V}^{\mathsf{RO}}\big(\mathcal{A}(\mathsf{RO})\big) = 1\big) ,$$

where RO is distributed uniformly at random over the set of random oracles $\mathcal{C}^{\mathcal{M}}$.

We note that even though the above notation suggests that the entire function table ro is given as input to $\mathcal{A}$, it is understood that $\mathcal{A}$ represents a Q-query algorithm, and so accesses at most Q positions of ro. Similarly, $\mathcal{V}^{\mathsf{ro}}$ only needs to make a single query to the random oracle to verify the proof. This difference is only relevant when considering efficiency, but not when considering the success probability.

Remark 4 (Probabilistic Algorithms). As discussed in Section 3.4, we may assume $\mathcal{P}^*$, and thus $\mathcal{A}$, to be deterministic. However, in our (recursive) analysis of multi-round protocols, it will be essential to allow for probabilistic algorithm $\mathcal{A}$. For this reason, we do not restrict to deterministic algorithms $\mathcal{A}$. Further, given a probabilistic algorithm $\mathcal{A}$, we will write $\mathcal{A}[r]$ for the deterministic algorithm that evaluates $\mathcal{A}$ with fixed random coins r.

The goal of the knowledge extractor is to find accepting transcripts $(a, c_1, z_1), \dots, (a, c_k, z_k)$, with common first message a and such that $\{c_1, \dots, c_k\} \in \mathit{\Gamma}$. As we show below, this can be achieved by combining techniques from [AFR23, KL23] and [AFK22, AFK23]. Deriving its success probability and expected running time is more involved.

To aid in our analysis we observe that, for all $\alpha \in \mathcal{M}$ and $\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}$, the algorithm $\mathcal{A}$ defines the algorithm

$$\mathcal{A}_\alpha^{\mathsf{ro}}\colon \mathcal{C} \to \mathcal{M} \times \{0,1\}^* , \quad c \mapsto (a, z) := \mathcal{A}(\mathsf{ro}[\alpha \mapsto c]) , \tag{10}$$

which takes as input a challenge $c \in \mathcal{C}$, reprograms the random oracle ro so that it answers queries to α with c, and then runs $\mathcal{A}$ with the reprogrammed random oracle

$$\mathsf{ro}[\alpha \mapsto c]\colon \mathcal{M} \to \mathcal{C} , \quad m \mapsto \begin{cases} \mathsf{ro}(m) , & \text{if } m \neq \alpha , \\ c , & \text{if } m = \alpha . \end{cases}$$

The corresponding verification predicate is

$$\mathcal{V}_\alpha(a, c, z) = \begin{cases} 1 & \text{if } \mathcal{V}(a, c, z) = 1 \text{ and } a = \alpha; \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

which is as $\mathcal{V}$, but additionally insists on a being α. Again, we allow $\mathcal{V}_\alpha$ to reorder its input so that we can write $\mathcal{V}_\alpha\big(c, \mathcal{A}(\mathsf{ro})\big)$ for $\mathcal{V}_\alpha(a, c, z)$ with $(a, z) \leftarrow \mathcal{A}(\mathsf{ro})$. By definition $\mathcal{V}_\alpha^{\mathsf{ro}}(a, z) = \mathcal{V}_\alpha(a, \mathsf{ro}(a), z)$, and it is easily seen that $\mathcal{V}_\alpha(a, \mathsf{ro}(a), z) = \mathcal{V}_\alpha(a, \mathsf{ro}(\alpha), z)$. It thus follows that $\mathcal{V}_\alpha^{\mathsf{ro}}\big(\mathcal{A}(\mathsf{ro})\big) = \mathcal{V}_\alpha\big(\mathsf{ro}(\alpha), \mathcal{A}(\mathsf{ro})\big)$.

Remark 5 (Early-abort). For later purposes, we assume that the computation of $(a, z) \leftarrow \mathcal{A}(\mathsf{ro})$ is split into two steps: $\mathcal{A}(\mathsf{ro})$ first computes $a = A(\mathsf{ro})$, and then continues to compute $z = Z(\mathsf{ro})$. This allows for an early-abort strategy for computing $\mathcal{A}_\alpha^{\mathsf{ro}}(c) = \mathcal{A}(\mathsf{ro}[\alpha \mapsto c])$, in which the $\mathcal{A}$-invocation is aborted if $a \neq \alpha$ is output. This assumption is not well motivated for the 3-round case, where we cannot expect $\mathcal{A}$ to spend significantly more time in computing z once it has decided on a, but it will be crucial in the multi-round running time analysis, where the 3-round case is recursively applied to $\mathcal{A}$ being a (sub)extractor that indeed decides on a early on.

The algorithm $\mathcal{A}_\alpha^{\mathsf{ro}}$ can be understood as an attacker against the underlying *interactive* proof, with the additional requirement that the attacker must use a particular first message α. Indeed, this algorithm is precisely of the form required by the extraction algorithm $\mathcal{E}_\Gamma$ (for interactive proofs) of Section 3.4, i.e., $\mathcal{E}_\Gamma^{\mathcal{A}_\alpha^{\mathsf{ro}}}$ is well-defined. The following lemma now relates the success probability of $\mathcal{A}_\alpha^{\mathsf{ro}}$ to that of $\mathcal{A}$.

Lemma 5. *Let $\mathcal{A}\colon \mathcal{C}^{\mathcal{M}} \to \mathcal{M} \times \{0,1\}^*$ be an algorithm together with a verification predicate $\mathcal{V}\colon \mathcal{M} \times \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. Then*[11]

$$\frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro}\in\mathcal{C}^{\mathcal{M}}} \sum_{\alpha\in\mathcal{M}} \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) = \epsilon^{\mathcal{V}}(\mathcal{A})\,.$$

Intuitively, this is pretty clear. Averaged over a random choice of the random oracle ro, the reprogramming at α has no effect, and then the summing over all α removes the requirement on a being α. The formal proof is spelled out as follows.

Proof. Let RO be distributed uniformly at random over $\mathcal{C}^{\mathcal{M}}$, and let C be distributed uniformly at random over $\mathcal{C}$. Then

$$\begin{aligned}\frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro}\in\mathcal{C}^{\mathcal{M}}} \sum_{\alpha\in\mathcal{M}} \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) &= \sum_{\alpha\in\mathcal{M}} \Pr\big(\mathcal{V}_\alpha\big(C, \mathcal{A}_\alpha^{\mathsf{RO}}(C)\big) = 1\big)\\ &= \sum_{\alpha\in\mathcal{M}} \Pr\big(\mathcal{V}_\alpha\big(C, \mathcal{A}(\mathsf{RO}[\alpha\mapsto C])\big) = 1\big)\\ &= \sum_{\alpha\in\mathcal{M}} \Pr\big(\mathcal{V}_\alpha\big(\mathsf{RO}[\alpha\mapsto C](\alpha), \mathcal{A}(\mathsf{RO}[\alpha\mapsto C])\big) = 1\big)\\ &= \sum_{\alpha\in\mathcal{M}} \Pr\big(\mathcal{V}_\alpha\big(\mathsf{RO}(\alpha), \mathcal{A}(\mathsf{RO})\big) = 1\big)\\ &\overset{*}{=} \sum_{\alpha\in\mathcal{M}} \Pr\big(\mathcal{V}_\alpha^{\mathsf{RO}}\big(\mathcal{A}(\mathsf{RO})\big) = 1\big)\\ &= \Pr\big(\mathcal{V}^{\mathsf{RO}}\big(\mathcal{A}(\mathsf{RO})\big) = 1\big) = \epsilon^{\mathcal{V}}(\mathcal{A})\,,\end{aligned}$$

where, in the equality marked by $*$, we use that $\mathcal{V}_\alpha^{\mathsf{ro}}\big(\mathcal{A}(\mathsf{ro})\big) = \mathcal{V}_\alpha\big(\mathsf{ro}(\alpha), \mathcal{A}(\mathsf{ro})\big)$, as explained earlier. This completes the proof of the lemma. $\square$

Before we define our knowledge extractor, we introduce the following crucial quantity, denoted $q(\mathcal{A})$. For a fixed random oracle ro and a fixed randomness of $\mathcal{A}$, we count the number of prover messages α such that, after reprogramming ro in α to a random value C, $\mathcal{A}$'s first output equals α with positive probability. The quantity $q(\mathcal{A})$ is then defined as the expectation of this number, averaged over the choice of the the random oracle and of $\mathcal{A}$'s randomness. It turns out that both the expected running time and the success probability of our knowledge extractor depend on $q(\mathcal{A})$.

Definition 8. *Let $\mathcal{A} = (A, Z)\colon \mathcal{C}^{\mathcal{M}} \to \mathcal{M} \times \{0,1\}^*$, $\mathsf{ro} \mapsto \big(A(\mathsf{ro}), Z(\mathsf{ro})\big)$ be a (probabilistic) algorithm. Then*

$$q(\mathcal{A}) := \mathop{\mathbb{E}}_{r}\left[\frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro}\in\mathcal{C}^{\mathcal{M}}} \big|\{\alpha \in \mathcal{M} : \Pr\big(A[r](\mathsf{ro}[\alpha\mapsto C]) = \alpha\big) > 0\}\big|\right],$$

where C is distributed uniformly random over $\mathcal{C}$ and the expectation is over the random coins r of $\mathcal{A}$.

The following lemma shows that we can control the value $q(\mathcal{A})$ via the query complexity of $\mathcal{A}$.

Lemma 6. *If $\mathcal{A} = (A, Z)\colon \mathcal{C}^{\mathcal{M}} \to \mathcal{M} \times \{0,1\}^*$ is a Q-query algorithm, then $q(\mathcal{A}) \le Q + 1$.*

[11] We note that we use ϵ for the success probability in the interactive setting with a uniformly random challenge, as in $\epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}})$, and in the non-interactive setting, as in $\epsilon^{\mathcal{V}}(\mathcal{A})$.

Proof. Without loss of generality, we may assume $\mathcal{A}$ to be deterministic. Let us fix $\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}$ and $a = A(\mathsf{ro})$. Further, let $S(\mathsf{ro}) \subseteq \mathcal{C}$ be the set of messages queried by $\mathcal{A}(\mathsf{ro})$, then $|S(\mathsf{ro})| \leq Q$.

If $\mathcal{A}(\mathsf{ro})$ does not query message α, i.e., $\alpha \notin \mathcal{S}(\mathsf{ro})$, then $\mathcal{A}^{\mathsf{ro}}_{\alpha}(c) = \mathcal{A}(\mathsf{ro}[\alpha \mapsto c])$ is oblivious to the input $c \in \mathcal{C}$. Hence, since $\mathcal{A}$ is deterministic, $A(\mathsf{ro}[\alpha \mapsto c]) = a$ for all $\alpha \notin S(\mathsf{ro})$ and all $c \in \mathcal{C}$. It therefore follows that

$$p(\alpha) := \Pr\big(A(\mathsf{ro}[\alpha \mapsto C]) = \alpha\big) = 0$$

for all $\alpha \notin S(\mathsf{ro}) \cup \{a\}$, which implies that

$$|\{\alpha : p(\alpha) > 0\}| = |\{\alpha \in S(\mathsf{ro}) : p(\alpha) > 0\}| + |\{\alpha \notin S(\mathsf{ro}) : p(\alpha) > 0\}| \leq |S(\mathsf{ro})| + 1 \leq Q + 1\,.$$

The lemma now follows trivially. □

5.2 The Knowledge Extractor

With the above observations at hand, we can define a knowledge extractor $\mathcal{F}_\Gamma$ for Fiat–Shamir transformations of Γ-out-of-$\mathcal{C}$ special-sound interactive proofs.

In the first step, the extractor $\mathcal{F}_\Gamma$ evaluates $(a, z) \leftarrow \mathcal{A}(\mathsf{ro})$ for a random oracle $\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}$ sampled uniformly at random. If $\mathcal{V}^{\mathsf{ro}}(a, z) = 0$, the extractor aborts. Otherwise, the extractor proceeds by running $\mathcal{E}_{\text{search},\Gamma}(c, a, z)$ on $\mathcal{A}^{\mathsf{ro}}_a$, i.e., it proceeds as in the interactive setting, but now using the algorithm $\mathcal{A}^{\mathsf{ro}}_a$ (and using the same random coins for $\mathcal{A}$ as in the first step). Furthermore, in this second phase of the extraction algorithm, every $\mathcal{A}$-invocation is early-aborted if its first output is incorrect, i.e., $(a', z') \leftarrow \mathcal{A}(\mathsf{ro}[a \mapsto c']) = \mathcal{A}^{\mathsf{ro}}_a(c')$ is early-aborted if $a' \neq a$ (see also Remark 5). The extractor is formally described in Figure 2. Its properties are summarized in Theorem 2.

We note that the early-abort property is only exploited in the refined running time analysis of Section 5.3 and, subsequently, in the analysis of multi-round interactive proofs. For this reason, this property will not play a role in Theorem 2 and its proof.

Fig. 2. Expected polynomial time extractor $\mathcal{F}_\Gamma$ for the Fiat–Shamir transformation of Γ-out-of-$\mathcal{C}$ special-sound Σ-protocols.

Parameters: a nonempty monotone structure $(\Gamma, \mathcal{C})$.
Oracle access to: algorithm $\mathcal{A}\colon \mathcal{C}^{\mathcal{M}} \to \mathcal{M} \times \{0,1\}^*$ with verification predicate $\mathcal{V}\colon \mathcal{M} \times \mathcal{C} \times \{0,1\}^* \to \{0,1\}$, such that $\mathcal{A}^{\mathsf{ro}}_\alpha$ and $\mathcal{V}_\alpha$ are defined as above (Equations (10) and (11)) for all $\alpha \in \mathcal{M}$ and $\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}$.
Output: $\perp$ or $(a, c_1, z_1), \ldots, (a, c_k, z_k)$ with $\{c_1, \ldots, c_k\} \in \Gamma$ and $\mathcal{V}(a, c_j, y_j) = 1$ for all j.

```
                                                                        $\boxed{\mathcal{F}^{\mathcal{A}}_{\text{init},\Gamma}}$
1. Sample $\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}$ uniformly at random and evaluate $(a, z) \leftarrow \mathcal{A}(\mathsf{ro})$.
     – From here on, all subsequent $\mathcal{A}$-invocations use the same random coins, i.e., from here on the first
       $\mathcal{A}$-invocation is rewound when $\mathcal{A}$ is invoked.
2. If $\mathcal{V}^{\mathsf{ro}}(a, z) = 0$, abort and output $\perp$.
3. If $\mathcal{V}^{\mathsf{ro}}(a, z) = 1$ and $\{\mathsf{ro}(a)\} \in \Gamma$, output $(a, \mathsf{ro}(a), z)$.
--------------------------------------------------------------------------------------------
                                        $\boxed{\mathcal{F}^{\mathcal{A}}_{\text{search},\Gamma}(a, z, \mathsf{ro}) = \mathcal{E}^{\mathcal{A}^{\mathsf{ro}}_a}_{\text{search},\Gamma}(\mathsf{ro}(a), a, z)}$
4. Else, set $S_1 = \{\mathsf{ro}(a)\} \subseteq \mathcal{C}$, $i = 1$ and COIN $= 0$.
5. Repeat:
     – sample $c_{i+1} \in \mathcal{U}_\Gamma(S_i)$ uniformly at random and evaluate

           $(a', z_{i+1}) \leftarrow \mathcal{A}^{\mathsf{ro}}_a(c_{i+1}) = \mathcal{A}(\mathsf{ro}[a \mapsto c_{i+1}])\,,$

       early-aborting the $\mathcal{A}$-invocation if $a' \neq a$;
         • if $\mathcal{V}_a(a', c_{i+1}, z_{i+1}) = 1$, set $S_{i+1} := S_i \cup \{c_{i+1}\}$ and then set $i = i + 1$;
         • else sample $d \in \mathcal{U}_\Gamma(S_{i-1})$, respectively $d \in \mathcal{C}$ if $i = 1$, uniformly at random and set COIN $= 1$
           if $d \notin \mathcal{U}_\Gamma(S_i)$ and $\mathcal{V}_a\big(a'', d, z''\big) = 1$ for $(a'', z'') \leftarrow \mathcal{A}^{\mathsf{ro}}_a(d)$, early-aborting the $\mathcal{A}$-invocation if
           $a'' \neq a$;
   until $S_i \in \Gamma$ or until COIN $= 1$.
6. If COIN $= 1$, abort and output $\perp$.
7. If $S_i = \{\mathsf{ro}(a), c_2, \ldots, c_i\} \in \Gamma$, output $(a, \mathsf{ro}(a), z_1), (a, c_2, z_2), \ldots, (a, c_i, z_i)$ with $\mathcal{V}(a, \mathsf{ro}(a), z_1) = 1$ and
   $\mathcal{V}(a, c_j, z_j) = 1$ for all $2 \leq j \leq i$.
```

Theorem 2 (Extractor - Fiat–Shamir Transformation of Σ-Protocols). *Let $(\Gamma, \mathcal{C})$ be a nonempty monotone structure. Then there exists an oracle algorithm $\mathcal{F}_\Gamma$ with the following properties: The algorithm $\mathcal{F}_\Gamma^{\mathcal{A}}$, given oracle access to an algorithm $\mathcal{A} = (A, Z) \colon \mathcal{C}^{\mathcal{M}} \to \mathcal{M} \times \{0,1\}^*$ defined as above with verification predicate $\mathcal{V} \colon \mathcal{M} \times \mathcal{C} \times \{0,1\}^* \to \{0,1\}$, requires an expected number of at most $1 + q(\mathcal{A}) \cdot t_\Gamma \cdot (1 + \kappa_\Gamma)$ queries to $\mathcal{A}$ and, with probability at least*

$$\frac{\epsilon(\mathcal{A}) - q(\mathcal{A}) \cdot \kappa_\Gamma}{1 - \kappa_\Gamma}$$

it outputs tuples $(a, c_1, z_1), \dots, (a, c_k, z_k) \in \mathcal{M} \times \mathcal{C} \times \{0,1\}^*$, for some $k \in \mathbb{N}$, with $\mathcal{V}(a, c_j, z_j) = 1$ for all j and $\{c_1, \dots, c_k\} \in \Gamma$.

Proof. Note that, by linearity of expected value and the quantities we aim to bound, the running time and success bounds can be written as expressions $\mathbb{E}_r[\dots]$ over the random coins of r of $\mathcal{A}$. Thus, without loss of generality, we may assume that $\mathcal{A}$ is deterministic.

For simplicity, we drop the subscript Γ and write $\mathcal{E}$ and $\mathcal{F}$ for $\mathcal{E}_\Gamma$ and $\mathcal{F}_\Gamma$. Let the random variable RO denote the random oracle sampled by the extractor for its first $\mathcal{A}$-invocation, i.e., RO is distributed uniformly at random over $\mathcal{C}^{\mathcal{M}}$. By a small abuse of notation, let $(A, Z) = \mathcal{A}(\mathsf{RO})$, i.e., the random variables A and Z denote the output of $\mathcal{F}^{\mathcal{A}}$'s first invocation of $\mathcal{A}$. Finally, for all $\alpha \in \mathcal{M}$ and $\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}$, let $\Psi_\alpha(\mathsf{ro})$ denote the event

$$\mathsf{RO}(m) = \mathsf{ro}(m) \quad \forall\, m \in \mathcal{M} \setminus \{\alpha\}\,,$$

i.e., the condition $\Psi_\alpha(\mathsf{ro})$ fixes RO everywhere outside α. We are now ready to analyze our extractor.

Success Probability. Conditioned on $\Psi_\alpha(\mathsf{ro})$, the extractor $\mathcal{F}^{\mathcal{A}}$ first samples a challenge $\mathsf{RO}(\alpha) = c \in \mathcal{C}$ and evaluates $(a, z) = \mathcal{A}(\mathsf{ro}[\alpha \mapsto c]) = \mathcal{A}_\alpha^{\mathsf{ro}}(c)$. As such, conditioned on $\Psi_\alpha(\mathsf{ro})$, the first steps of the extractors $\mathcal{F}^{\mathcal{A}}$ and $\mathcal{E}^{\mathcal{A}_\alpha^{\mathsf{ro}}}$ are almost identical. The only difference between the two is that $\mathcal{E}^{\mathcal{A}_\alpha^{\mathsf{ro}}}$ aborts if its first $\mathcal{A}$-invocation outputs a message $a \neq \alpha$, whereas $\mathcal{F}^{\mathcal{A}}$ would then proceed with running $\mathcal{E}_{\text{search}}^{\mathcal{A}_\alpha^{\mathsf{ro}}}$.[12] It thus follows that

$$\Pr\bigl(\mathcal{F}^{\mathcal{A}} = 1 \wedge A = \alpha \mid \Psi_\alpha(\mathsf{ro})\bigr) = \Pr\bigl(\mathcal{E}^{\mathcal{A}_\alpha^{\mathsf{ro}}} = 1\bigr)\,,$$

where we write $\mathcal{F}^{\mathcal{A}} = 1$ and $\mathcal{E}^{\mathcal{A}_\alpha^{\mathsf{ro}}} = 1$ for the events that the extractors succeed.

By basic probability theory, we can now derive the following bound

$$\begin{aligned}
\Pr(\mathcal{F}^{\mathcal{A}} = 1) &= \sum_{\alpha \in \mathcal{M}} \Pr(\mathcal{F}^{\mathcal{A}} = 1 \wedge A = \alpha) \\
&= \frac{1}{|\mathcal{C}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha \in \mathcal{M}} \Pr\bigl(\mathcal{F}^{\mathcal{A}} = 1 \wedge A = \alpha \wedge \Psi_\alpha(\mathsf{ro})\bigr) \\
&= \frac{1}{|\mathcal{C}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha \in \mathcal{M}} \Pr\bigl(\Psi_\alpha(\mathsf{ro})\bigr) \cdot \Pr\bigl(\mathcal{F}^{\mathcal{A}} = 1 \wedge A = \alpha \mid \Psi_\alpha(\mathsf{ro})\bigr) \\
&= \frac{1}{|\mathcal{C}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha \in \mathcal{M}} \frac{|\mathcal{C}|}{|\mathcal{C}^{\mathcal{M}}|} \cdot \Pr\bigl(\mathcal{E}^{\mathcal{A}_\alpha^{\mathsf{ro}}} = 1\bigr) \\
&= \frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha \in \mathcal{M}} \Pr\bigl(\mathcal{E}^{\mathcal{A}_\alpha^{\mathsf{ro}}} = 1\bigr) \\
&\geq \frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha \in \mathcal{M}} \max\left(\frac{\epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) - \kappa_\Gamma}{1 - \kappa_\Gamma}, 0\right) \\
&\geq \frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha : \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) > 0} \frac{\epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) - \kappa_\Gamma}{1 - \kappa_\Gamma} \\
&\geq \frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - q(\mathcal{A}) \cdot \kappa_\Gamma}{1 - \kappa_\Gamma}\,,
\end{aligned}$$

[12] Here we are exploiting that $\mathcal{E}$ samples c_1 from $\mathcal{C}$, rather than from $\mathcal{U}_\Gamma(\emptyset)$, which would be more natural in the context of $\mathcal{E}$. See also Remark 3.

where the first inequality follows from Theorem 1 and the final inequality follows from Lemma 5 together with the observation that

$$\epsilon^{\mathcal{V}_\alpha}(\mathcal{A}^{\mathsf{ro}}_\alpha) \leq \Pr\big(A(\mathsf{ro}[\alpha \mapsto C]) = \alpha\big)\,,$$

and thus

$$\big|\{\alpha : \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}^{\mathsf{ro}}_\alpha) > 0\}\big| \leq \big|\{\alpha : \Pr\big(A(\mathsf{ro}[\alpha \mapsto C]) = \alpha\big) > 0\}\big|\,.$$

This proves the claimed bound on the success probability.

Expected Running Time. As in the success probability analysis, we will use the fact that after the first $\mathcal{A}$-invocation the extractor $\mathcal{F}^{\mathcal{A}}$ proceeds exactly as in the interactive case.

For $\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}$ and $\alpha \in \mathcal{M}$, let V^{ro}_α denote the event that the first $\mathcal{A}$-invocation of the extractor $\mathcal{E}^{\mathcal{A}^{\mathsf{ro}}_\alpha}$, when applied to $\mathcal{A}^{\mathsf{ro}}_\alpha$, is successful. Further, we write X^{ro}_α for the number of $\mathcal{A}$-queries made by $\mathcal{E}^{\mathcal{A}^{\mathsf{ro}}_\alpha}$, not counting the first $\mathcal{A}$-query. Then, exploiting $\mathbb{E}[X^{\mathsf{ro}}_\alpha \mid \neg V^{\mathsf{ro}}_\alpha] = 0$ and using Theorem 1, if follows that

$$\Pr(V^{\mathsf{ro}}_\alpha) \cdot \mathbb{E}[X^{\mathsf{ro}}_\alpha \mid V^{\mathsf{ro}}_\alpha] = \mathbb{E}[X^{\mathsf{ro}}_\alpha] \leq t_\Gamma \cdot (1 + \kappa_\Gamma)\,. \tag{12}$$

Let us now do something similar for the extractor $\mathcal{F}^{\mathcal{A}}$. More precisely, we let V denote the event that $\mathcal{F}^{\mathcal{A}}$'s first $\mathcal{A}$-invocation is successful, and we let Y denote the number $\mathcal{A}$-queries made by $\mathcal{F}^{\mathcal{A}}$ except for the first one. Then,

$$\Pr\big(V \wedge A = \alpha \mid \Psi_\alpha(\mathsf{ro})\big) = \Pr(V^{\mathsf{ro}}_\alpha)\,,$$

and $\mathbb{E}[Y \mid \neg V] = 0$. Further, by construction of the extractors,

$$\mathbb{E}[Y \mid V \wedge A = \alpha \wedge \Psi_\alpha(\mathsf{ro})] = \mathbb{E}[X^{\mathsf{ro}}_\alpha \mid V^{\mathsf{ro}}_\alpha]\,.$$

Hence,

$$\begin{aligned}
\mathbb{E}[Y] &= \Pr(V) \cdot \mathbb{E}[Y \mid V] \\
&= \frac{1}{|\mathcal{C}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha \in \mathcal{M}} \Pr\big(\Psi_\alpha(\mathsf{ro})\big) \cdot \Pr\big(V \wedge A = \alpha \mid \Psi_\alpha(\mathsf{ro})\big) \cdot \mathbb{E}[Y \mid V \wedge A = \alpha \wedge \Psi_\alpha(\mathsf{ro})] \\
&= \frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha \in \mathcal{M}} \Pr(V^{\mathsf{ro}}_\alpha) \cdot \mathbb{E}[X^{\mathsf{ro}}_\alpha \mid V^{\mathsf{ro}}_\alpha] \\
&\leq \frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha : \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}^{\mathsf{ro}}_\alpha) > 0} t_\Gamma \cdot (1 + \kappa_\Gamma) \\
&\leq q(\mathcal{A}) \cdot t_\Gamma \cdot (1 + \kappa_\Gamma)\,,
\end{aligned}$$

where we use that $\Pr(V^{\mathsf{ro}}_\alpha) = \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}^{\mathsf{ro}}_\alpha) \leq \Pr\big(A(\mathsf{ro}[\alpha \mapsto C]) = \alpha\big)$. Hence, the expected running time of $\mathcal{F}^{\mathcal{A}}$ is at most $1 + q(\mathcal{A}) \cdot t_\Gamma \cdot (1 + \kappa_\Gamma)$, which completes the proof. □

The following theorem now immediately follows from Lemma 6 and Theorem 2.

Theorem 3 (Fiat–Shamir Transformation of a Σ-Protocol). *Let Π be a Γ-out-of-$\mathcal{C}$ special-sound Σ-protocol Π, with t_Γ polynomial in the size $|x|$ of the statement x and such that sampling from $\mathcal{U}_\Gamma(S)$ takes polynomial time (in $|x|$) for all $S \subseteq \mathcal{C}$ with $|S| \leq t_\Gamma$. Then, the Fiat–Shamir transformation $\mathsf{FS}[\Pi]$ of a Π is knowledge sound with knowledge error*

$$(Q + 1) \cdot \kappa_\Gamma\,,$$

where $\kappa_\Gamma = \max_{S \notin \Gamma} \frac{|S|}{|\mathcal{C}|}$ is the knowledge error of the (interactive) Σ-protocol Π.

More precisely, $\mathsf{FS}[\Pi]$ admits a knowledge extractor that, on input a statement x and given oracle access to a Q-query prover $\mathcal{P}^*$, makes an expected number of at most $1 + (Q + 1) \cdot t_\Gamma \cdot (1 + \kappa_\Gamma)$ queries to $\mathcal{P}^*$ and succeeds to extract a witness for x with probability at least

$$\frac{\epsilon(\mathcal{P}^*, x) - (Q + 1) \cdot \kappa_\Gamma}{1 - \kappa_\Gamma}\,.$$

Remark 6 (Generic Applicability of our Fiat–Shamir Analysis). The knowledge extractor $\mathcal{F}_\Gamma$ only mildly depends on the actual construction of the underlying extractor $\mathcal{E}_\Gamma$ for Γ-special-sound *interactive* proofs. More precisely, $\mathcal{F}_\Gamma$ uses the fact that $\mathcal{E}_\Gamma$ first tries a random challenge and only continues if this challenges is accepted. However, to $\mathcal{F}_\Gamma$ it is irrelevant how $\mathcal{E}_\Gamma$ continues after this first phase; the second phase of $\mathcal{E}_\Gamma$ is deployed in a black-box manner. This shows that our design principle allows one to transform any knowledge extractor for Γ-special-sound interactive proofs, containing the aforementioned two phases, into a knowledge extractor for the Fiat–Shamir transformation of Γ-special-sound interactive proofs.

5.3 A Refined Running Time Analysis

For the multi-round version of our result, we need a refined running time analysis, which considers an $\mathcal{A} = (A, Z)$ for which it holds that computing the output $A(\mathsf{ro})$ is significantly cheaper than computing the full output $\mathcal{A}(\mathsf{ro}) = (A(\mathsf{ro}), Z(\mathsf{ro}))$. This allows us to improve the extractor $\mathcal{F}_\Gamma$ by early aborting, i.e., by only computing the output Z if it is really necessary. In this section, we make this early abort assumption, which was introduced in Remark 5, quantitative.

Looking ahead, in the multi-round extractor construction we will apply $\mathcal{F}_\Gamma$ to a particular subextractor, which we cast as an algorithm $\mathcal{A} = (A, Z)$, and for which computing A is indeed cheap while computing Z is expensive. In this case, the cost will be measured by the number of calls the subextractor makes to the dishonest prover. Below, we just consider a generic cost measure θ.

Lemma 7 (Refined Running Time Analysis - NIROP Extractor). *Let $(\Gamma, \mathcal{C})$ be a nonempty monotone structure and*

$$\mathcal{A}\colon \mathcal{C}^{\mathcal{M}} \to \mathcal{M} \times \{0,1\}^*\,, \quad \mathsf{ro} \mapsto \big(A(\mathsf{ro}), Z(\mathsf{ro})\big)$$

an algorithm with a verification predicate $\mathcal{V}\colon \mathcal{M} \times \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. Let $\theta\colon \mathcal{C}^{\mathcal{M}} \to \mathbb{R}_{\geq 1}$, such that $\theta(\mathsf{ro})$ denotes the expected cost of evaluating $\big(A(\mathsf{ro}), Z(\mathsf{ro})\big) = \mathcal{A}(\mathsf{ro})$. Additionally, let us assume that solely evaluating $A(\mathsf{ro})$ has constant cost 1.

Let $\mathcal{F}_\Gamma^{\mathcal{A}}$ denote the knowledge extractor of Figure 2 equipped with the early-abort strategy. Then the expected cost of the $\mathcal{A}$-invocations of $\mathcal{F}_\Gamma^{\mathcal{A}}$ is at most

$$\mathbb{E}[\theta(\mathsf{RO})] + t_\Gamma \cdot \frac{\mathbb{E}[\theta(\mathsf{RO})] - 1 + q(\mathcal{A})}{1 - \kappa_\Gamma}\,.$$

Proof. By linearity of the expected value (over the random coins of $\mathcal{A}$) and the quantities we aim to bound, we may, without loss of generality, assume that $\mathcal{A}$ is deterministic.

As in the proof of Theorem 2, we will reduce the problem to the analysis of the extractor $\mathcal{E}$ for the interactive case, but now using the refined running time analysis of Lemma 4.

We first fix $\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}$ and $\alpha \in \mathcal{M}$ and consider the execution of $\mathcal{E}_\Gamma^{\mathcal{A}_\alpha^{\mathsf{ro}}}$. Here, in line with the early abort strategy, $\mathcal{A}_\alpha^{\mathsf{ro}}(c) = \mathcal{A}(\mathsf{ro}[\alpha \mapsto c])$ first evaluates $\alpha' \leftarrow A_\alpha^{\mathsf{ro}}(c)$ and aborts if $\alpha \neq \alpha'$, otherwise it continues to additionally evaluate $Z_\alpha^{\mathsf{ro}}(c)$, where here and below $(A_\alpha^{\mathsf{ro}}(c), Z_\alpha^{\mathsf{ro}}(c))$ denotes the output of $\mathcal{A}_\alpha^{\mathsf{ro}}(c)$. In particular, it holds that $A_\alpha^{\mathsf{ro}}(c) = A(\mathsf{ro}[\alpha \mapsto c])$. Thus, $\mathcal{A}_\alpha^{\mathsf{ro}}$ has cost function

$$\theta_\alpha^{\mathsf{ro}}\colon \mathcal{C} \to \mathbb{R}_{\geq 0}\,, \quad c \mapsto \begin{cases} \theta(\mathsf{ro}[\alpha \mapsto c])\,, & \text{if } A_\alpha^{\mathsf{ro}}(c) = \alpha\,, \\ 1\,, & \text{otherwise}\,. \end{cases}$$

As in the proof of Theorem 2, let V_α^{ro} denote the event that the first $\mathcal{A}_\alpha^{\mathsf{ro}}$-invocation of the extractor $\mathcal{E}_\Gamma^{\mathcal{A}_\alpha^{\mathsf{ro}}}$ is successful, i.e., $\Pr(V_\alpha^{\mathsf{ro}}) = \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}})$, and let us write X_α^{ro} for the cost of the $\mathcal{A}_\alpha^{\mathsf{ro}}$-queries made by $\mathcal{E}^{\mathcal{A}_\alpha^{\mathsf{ro}}}$, not counting the cost of the first $\mathcal{A}_\alpha^{\mathsf{ro}}$-query. Then, recalling (12), we have

$$\Pr(V_\alpha^{\mathsf{ro}}) \cdot \mathbb{E}[X_\alpha^{\mathsf{ro}} \mid V_\alpha^{\mathsf{ro}}] = \mathbb{E}[X_\alpha^{\mathsf{ro}}]\,.$$

In the proof of Theorem 2, we used Theorem 1 to bound $\mathbb{E}[X_\alpha^{\mathsf{ro}}]$. Here we use the refined running time bound of Lemma 4, from which it follows that

$$\mathbb{E}[X_\alpha^{\mathsf{ro}}] \leq t_\Gamma \cdot \frac{\mathbb{E}[\theta_\alpha^{\mathsf{ro}}(C)]}{1 - \kappa_\Gamma} = \frac{t_\Gamma}{1 - \kappa_\Gamma}\Big(\Pr(A_\alpha^{\mathsf{ro}}(C) = \alpha)\, \mathbb{E}[\theta_\alpha^{\mathsf{ro}}(C) \mid A_\alpha^{\mathsf{ro}}(C) = \alpha] + \Pr(A_\alpha^{\mathsf{ro}}(C) \neq \alpha)\Big) \tag{13}$$

where C is uniformly random in $\mathcal{C}$. Note that we ignore the cost of the first $\mathcal{A}$ invocation, and thus, in Equation (13), omitted the first summand $\mathbb{E}[\theta_\alpha^{\mathsf{ro}}(C)]$ from the running time bound of Lemma 4.

Let us now proceed similarly for the extractor $\mathcal{F}_\Gamma^{\mathcal{A}}$. More precisely, we let V denote the event that the first $\mathcal{A}$-query is successful, and we let Y denote the cost of the $\mathcal{A}$-queries made by $\mathcal{F}_\Gamma^{\mathcal{A}}$ not counting the cost of the first $\mathcal{A}$-query. Clearly, $\mathbb{E}[Y \mid \neg V] = 0$. Further, we let the random variable RO denote the initial choice of the random oracle by $\mathcal{F}_\Gamma^{\mathcal{A}}$ and, as before, $\Psi_\alpha(\mathsf{ro})$ denotes the event

$$\mathsf{RO}(m) = \mathsf{ro}(m) \quad \forall m \in \mathcal{M} \setminus \{\alpha\} \,.$$

The crucial observation is that, by construction of $\mathcal{F}_\Gamma$,

$$\mathbb{E}[Y \mid V \wedge A(\mathsf{RO}) = \alpha \wedge \Psi_\alpha(\mathsf{ro})] = \mathbb{E}[X_\alpha^{\mathsf{ro}} \mid V_\alpha^{\mathsf{ro}}] \,,$$

and

$$\Pr\big(V \wedge A(\mathsf{RO}) = \alpha \mid \Psi_\alpha(\mathsf{ro})\big) = \Pr(V_\alpha^{\mathsf{ro}}) = \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) \,.$$

Therefore,

$$\begin{aligned}
\mathbb{E}[Y] &= \Pr(V) \cdot \mathbb{E}[Y \mid V] \\
&= \frac{1}{|\mathcal{C}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha \in \mathcal{M}} \Pr\big(\Psi_\alpha(\mathsf{ro})\big) \Pr\big(V \wedge A(\mathsf{RO}) = \alpha \mid \Psi_\alpha(\mathsf{ro})\big) \, \mathbb{E}[Y \mid V \wedge A(\mathsf{RO}) = \alpha \wedge \Psi_\alpha(\mathsf{ro})] \\
&= \frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha : \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) > 0} \Pr(V_\alpha^{\mathsf{ro}}) \, \mathbb{E}[X_\alpha^{\mathsf{ro}} \mid V_\alpha^{\mathsf{ro}}] \\
&= \frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha : \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) > 0} \mathbb{E}[X_\alpha^{\mathsf{ro}}] \,.
\end{aligned}$$

Hence, by Equation (13),

$$\mathbb{E}[Y] \le \frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha : \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) > 0} \frac{t_\Gamma}{1 - \kappa_\Gamma} \Big(\Pr(A_\alpha^{\mathsf{ro}}(C) = \alpha) \, \mathbb{E}[\theta_\alpha^{\mathsf{ro}}(C) \mid A_\alpha^{\mathsf{ro}}(C) = \alpha] + 1 - \Pr(A_\alpha^{\mathsf{ro}}(C) = \alpha) \Big).$$

As before, observe that $\epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) \le \Pr\big(A(\mathsf{ro}[\alpha \mapsto C]) = \alpha\big)$, and thus

$$\big|\{\alpha : \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) > 0\}\big| \le \big|\{\alpha : \Pr\big(A(\mathsf{ro}[\alpha \mapsto C]) = \alpha\big) > 0\}\big| \,,$$

which shows that

$$\frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha : \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) > 0} 1 \le q(\mathcal{A}) \,.$$

It therefore follows that

$$\begin{aligned}
\mathbb{E}[Y] \le \frac{t_\Gamma}{1 - \kappa_\Gamma} \Bigg(q(\mathcal{A}) + \frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha : \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) > 0} \\
\Big(\Pr(A_\alpha^{\mathsf{ro}}(C) = \alpha) \cdot \mathbb{E}[\theta_\alpha^{\mathsf{ro}}(C) \mid A_\alpha^{\mathsf{ro}}(C) = \alpha] - \Pr(A_\alpha^{\mathsf{ro}}(C) = \alpha) \Big) \Bigg) .
\end{aligned}$$

Further, since $\theta_\alpha^{\mathsf{ro}}(c) \ge 1$ for all $c \in \mathcal{C}$, it holds that all the terms in the above summation are nonnegative. Thus, we can extend the summation from $\{\alpha : \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) > 0\} \subseteq \mathcal{M}$ to all of $\mathcal{M}$. It therefore follows that

$$\begin{aligned}
&\frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha : \epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) > 0} \Big(\Pr(A_\alpha^{\mathsf{ro}}(C) = \alpha) \cdot \mathbb{E}[\theta_\alpha^{\mathsf{ro}}(C) \mid A_\alpha^{\mathsf{ro}}(C) = \alpha] - \Pr(A_\alpha^{\mathsf{ro}}(C) = \alpha) \Big) \\
&\quad \le \frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \sum_{\alpha \in \mathcal{M}} \Big(\Pr(A_\alpha^{\mathsf{ro}}(C) = \alpha) \cdot \mathbb{E}[\theta_\alpha^{\mathsf{ro}}(C) \mid A_\alpha^{\mathsf{ro}}(C) = \alpha] - \Pr(A_\alpha^{\mathsf{ro}}(C) = \alpha) \Big) \\
&\quad = \sum_{\alpha \in \mathcal{M}} \frac{1}{|\mathcal{C}^{\mathcal{M}}|} \sum_{\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}} \Big(\Pr(A_\alpha^{\mathsf{RO}}(C) = \alpha \mid \mathsf{RO} = \mathsf{ro}) \cdot \mathbb{E}[\theta_\alpha^{\mathsf{RO}}(C) \mid A_\alpha^{\mathsf{RO}}(C) = \alpha \wedge \mathsf{RO} = \mathsf{ro}] \\
&\qquad\qquad - \Pr(A_\alpha^{\mathsf{RO}}(C) = \alpha \mid \mathsf{RO} = \mathsf{ro}) \Big)
\end{aligned}$$

$$\begin{aligned}
&= \sum_{\alpha\in\mathcal{M}} \sum_{\mathsf{ro}\in\mathcal{C}^{\mathcal{M}}} \Big(\Pr(A_\alpha^{\mathsf{RO}}(C) = \alpha \wedge \mathsf{RO} = \mathsf{ro}) \cdot \mathbb{E}[\theta_\alpha^{\mathsf{RO}}(C) \mid A_\alpha^{\mathsf{RO}}(C) = \alpha \wedge \mathsf{RO} = \mathsf{ro}] \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad - \Pr(A_\alpha^{\mathsf{RO}}(C) = \alpha \wedge \mathsf{RO} = \mathsf{ro})\Big) \\
&= \sum_{\alpha\in\mathcal{M}} \Big(\Pr(A_\alpha^{\mathsf{RO}}(C) = \alpha) \cdot \mathbb{E}[\theta_\alpha^{\mathsf{RO}}(C) \mid A_\alpha^{\mathsf{RO}}(C) = \alpha] - \Pr(A_\alpha^{\mathsf{RO}}(C) = \alpha)\Big) \\
&= \sum_{\alpha\in\mathcal{M}} \Big(\Pr(A(\mathsf{RO}[\alpha\mapsto C]) = \alpha) \cdot \mathbb{E}[\theta(\mathsf{RO}[\alpha\mapsto C]) \mid A(\mathsf{RO}[\alpha\mapsto C])] - \Pr(A(\mathsf{RO}[\alpha\mapsto C]) = \alpha)\Big) \\
&\overset{*}{=} \sum_{\alpha\in\mathcal{M}} \Big(\Pr(A(\mathsf{RO}) = \alpha) \cdot \mathbb{E}[\theta(\mathsf{RO}) \mid A(\mathsf{RO}) = \alpha] - \Pr(A(\mathsf{RO}) = \alpha)\Big) \\
&= \mathbb{E}[\theta(\mathsf{RO})] - 1\,,
\end{aligned}$$

where equality $\overset{*}{=}$ uses that that RO and $\mathsf{RO}[\alpha\mapsto C]$ are identically distributed for all $\alpha\in\mathcal{M}$. Thus,

$$\mathbb{E}[Y] \le \frac{t_\Gamma}{1-\kappa_\Gamma}\big(q(\mathcal{A}) + \mathbb{E}[\theta(\mathsf{RO})] - 1\big)\,.$$

Hence, the expected cost of the $\mathcal{A}$-queries made by $\mathcal{F}_\Gamma^{\mathcal{A}}$ is at most

$$\mathbb{E}[\theta(\mathsf{RO})] + t_\Gamma \cdot \frac{\mathbb{E}[\theta(\mathsf{RO})] - 1 + q(\mathcal{A})}{1-\kappa_\Gamma}\,,$$

which completes the proof. □

6 The Fiat–Shamir Transformation of Multi-Round Interactive Proofs

In this section, we extend our result to the multi-round setting. First, following prior works, we formally introduce multi-round special-soundness, and then proceed to describe and analyze our extractor in the multi-round setting.

6.1 $(\Gamma_1,\ldots,\Gamma_\mu)$-out-of-$(\mathcal{C}_1,\ldots,\mathcal{C}_\mu)$ Special-Soundness

Let us write $\boldsymbol{\Gamma} := (\Gamma_1,\ldots,\Gamma_\mu)$ and $\boldsymbol{\mathcal{C}} := (\mathcal{C}_1,\ldots,\mathcal{C}_\mu)$. Then, towards defining multi-round special-soundness as defined in [AFR23], we recall the following.

Definition 9 (Tree of Transcripts). *Given a monotone structure* $(\Gamma,\mathcal{C})$*, a* Γ-tree of transcripts with trunk a *is a set* T *of triples* $(a,c,z)\in\{a\}\times\mathcal{C}\times\{0,1\}^*$ *such that*

$$\{c\in C \mid \exists\, z\in\{0,1\}^* : (a,c,z)\in T\}\in\Gamma\,.$$

We write $\textsc{Tree}_\Gamma(a)$ *and* $\textsc{Tree}_\Gamma$ *for the set of* Γ*-trees with trunk* a *and with arbitrary trunk, respectively.*

Given $(\Gamma_1,\mathcal{C}_1),\ldots,(\Gamma_\mu,\mathcal{C}_\mu)$*, a* $\boldsymbol{\Gamma}$-tree of transcripts with *trunk* a_1 *is recursively defined to be a set* $T\in\textsc{Tree}_{\boldsymbol{\Gamma}}$ *of triples* $(a_1,c_1,T')\in\{a_1\}\times\mathcal{C}_1\times\textsc{Tree}_{\Gamma_2,\ldots,\Gamma_\mu}$ *such that*

$$\{c_1\in C \mid \exists\, T'\in\textsc{Tree}_{\Gamma_2,\ldots,\Gamma_\mu} : (a_1,c_1,T')\in T\}\in\Gamma_1\,,$$

where $\textsc{Tree}_{\Gamma_2,\ldots,\Gamma_\mu}(a_2)$ *and* $\textsc{Tree}_{\Gamma_2,\ldots,\Gamma_\mu}$ *denote the set of all* $(\Gamma_2,\ldots,\Gamma_\mu)$*-trees with trunk* a_2 *and with arbitrary trunk, respectively.*

Remark 7. In the above recursive definition of a $\boldsymbol{\Gamma}$-tree, we naturally identify the set T of triples (a_1,c_1,T'), where each T' is a set again (namely a set of triples (a_2,c_2,T'') etc.), with the union of $\{(a_1,c_1)\}\times T'$ over the triples in T. This way, a $\boldsymbol{\Gamma}$-tree of transcripts with trunk a_1 becomes a set of tuples $(a_1,c_1,a_2,\ldots,a_\mu,c_\mu,a_{\mu+1})$.

Definition 10 (Tree of Accepting Transcripts). *Let*

$$\mathcal{V} \colon \mathcal{M} \times \mathcal{C}_1 \times \mathcal{M} \times \cdots \times \mathcal{M} \times \mathcal{C}_\mu \times \{0,1\}^* \to \{0,1\} \,,$$

be a verification predicate. A $\mathbf{\Gamma}$*-tree* T *of transcripts (with arbitrary trunk* a_1*) is called a* $\mathbf{\Gamma}$*-tree of* $\mathcal{V}$-accepting *transcripts if*

$$\mathcal{V}(a_1, c_1, \ldots, a_\mu, c_\mu, a_{\mu+1}) = 1$$

for every $(a_1, c_1, \ldots, a_\mu, c_\mu, a_{\mu+1}) \in T$*. As above, we write* $\textsc{Tree}_{\mathbf{\Gamma}}^{\mathcal{V}}$ *and* $\textsc{Tree}_{\mathbf{\Gamma}}^{\mathcal{V}}(a_1)$ *for the set of all* $(\mathit{\Gamma}_1, \ldots, \mathit{\Gamma}_\mu)$*-trees of* $\mathcal{V}$-accepting transcripts (with trunk a_1).

The definition of the generalized special-soundness notion for multi-round protocols can now be provided.

Definition 11 ($(\mathit{\Gamma}_1, \ldots, \mathit{\Gamma}_\mu)$-out-of-$(\mathcal{C}_1, \ldots, \mathcal{C}_\mu)$ Special-Soundness). *Let* $(\mathit{\Gamma}_i, \mathcal{C}_i)$ *be monotone structures for* $1 \leq i \leq \mu$*, and let* $\mathbf{\Gamma} := (\mathit{\Gamma}_1, \ldots, \mathit{\Gamma}_\mu)$ *and* $\boldsymbol{\mathcal{C}} := (\mathcal{C}_1, \ldots, \mathcal{C}_\mu)$*. A* $2\mu{+}1$*-round public-coin interactive proof* $(\mathcal{P}, \mathcal{V})$ *for relation* R*, with challenge sets* $\mathcal{C}_1, \ldots, \mathcal{C}_\mu$*, is* $\mathbf{\Gamma}$-out-of-$\boldsymbol{\mathcal{C}}$ special-sound *if there exists a polynomial time algorithm that, on input a statement* x *and a* $\mathbf{\Gamma}$*-tree of accepting transcripts, outputs a witness* $w \in R(x)$*. We also say that* $(\mathcal{P}, \mathcal{V})$ *is* $\mathbf{\Gamma}$*-special-sound.*

Our goal is to prove that, under certain mild assumptions, the Fiat–Shamir transformation of a $\mathbf{\Gamma}$-special-sound interactive proof is knowledge sound. For context, we note that if the product of the t-values $t_{\mathbf{\Gamma}} := \prod_{i=1}^{\mu} t_{\mathit{\Gamma}_i}$ is polynomially bounded, then [AFR23] establishes that $\mathbf{\Gamma}$-out-of-$\boldsymbol{\mathcal{C}}$ special-sound interactive proofs are knowledge sound with knowledge error

$$\kappa_{\mathbf{\Gamma}} = 1 - \prod_{i=1}^{\mu} (1 - \kappa_{\mathit{\Gamma}_i}) \,, \tag{14}$$

where the $\kappa_{\mathit{\Gamma}_i}$ are defined as in (4). However, we don't need this multi-round result in our work here; we only need the (improved) 3-round case of Section 3.4.

6.2 Preliminary Discussion

Given how we reduce the non-interactive to the interactive extractor for the 3-round case, a natural approach would be to do the same here for the multi-round case. Unfortunately, the running time would blow up too much. Indeed, in the 3-round case, having found one accepting transcript (a, c, z), we can hope for finding more transcripts *with the same* a by running $\mathcal{A}_a^{\mathsf{ro}}$ with different challenges (indeed, we have shown that $Q+1$ tries are sufficient in expectation); however, in the multi-round case, having found a transcript $(a_1, c_1, a_2, \ldots, a_\mu, c_\mu, a_{\mu+1})$ it is too much to hope for finding more transcripts with the same $(a_1, a_2, \ldots, a_\mu)$. It is also not necessary, since in different branches of a tree of transcripts the a_i's may well be different.

Another approach, also followed in prior works [ACK21, AF22, AFK22, AFR23, AFK23, KL23], is to handle the multi-round setting recursively. Informally, a $(2\mu + 1)$-round extractor invokes a 3-round extractor, but replaces the algorithm $\mathcal{A}$ by appropriate instantiations of a $(2\mu - 1)$-round extractor.

However, the analysis of this recursive approach exposes certain subtle issues, especially when considering Fiat–Shamir transformations. More precisely, when analyzing Fiat–Shamir transformations, the expected running time of a naive recursion scales linearly in Q^μ, where Q is the query complexity of the prover attacking the non-interactive proof, and thus exponentially in the number of rounds $2\mu + 1$. For a non-constant number of rounds, this renders the expected running time of the extractor superpolynomial.[13]

The same problem was encountered in the analysis of Fiat–Shamir transformations of $(k_1, \ldots, k_\mu)$-special-sound interactive proofs [AFK22, AFK23]. There, it is solved by making the following observation. An invocation of the $(2\mu - 1)$-round (sub)extractor is successful if it outputs a (sub)tree of accepting transcripts, *and* the (unique) first message of the tree corresponds to the first messages output by earlier invocations of this $(2\mu - 1)$-round extractor. The second requirement ensures that sufficiently many (sub)trees can be combined into a single larger tree.

[13] Assuming Q is polynomially-large, as is standard.

Hence, every invocation of a subextractor can fail for two reasons: (1) because it fails to output a subtree; or (2) because the first message output by the subextractor is incorrect. Since the first message is already determined in the first step of the (sub)extractor, a failure of type (2) can be identified before completing the execution of the subextractor. In other words, this observation allows for an *early-abort* strategy, where the subextractor aborts its execution after the first step if it outputs the wrong first message (cf. Remark 5). Hence, failures of type (2) are far less costly than failures of type (1); the latter can only be identified after the subextractor has finished. By applying this early-abort strategy and refining the running time analysis, it was shown that the recursively defined extractor of [AFK22, AFK23] can indeed be made to run in expected polynomial time. More precisely, the expected running time of their extractor scales linearly in Q.

We will use the same early-abort strategy to obtain an efficient knowledge extractor. To take this strategy into account, we exploit the refined running time analysis for the 3-round FS-extractor, shown in Section 5.3.

6.3 Setting Up the Stage

Let us now move to the analysis of multi-round interactive proofs. In this section, assume we are given μ monotone structures $\Gamma_1, \dots, \Gamma_\mu$, each $\Gamma_i \subseteq 2^{\mathcal{C}_i}$ for challenge sets $\mathcal{C}_i$. As before, we let $\mathbf{\Gamma} := (\Gamma_1, \dots, \Gamma_\mu)$ and $\boldsymbol{\mathcal{C}} := (\mathcal{C}_1, \dots, \mathcal{C}_\mu)$. Further we denote by $\overrightarrow{\mathsf{ro}} = (\mathsf{ro}_1, \dots, \mathsf{ro}_\mu)$ the vector of μ random oracles, where the understanding is that the i-th round challenge is determined via the i-th random oracle (cf. Definition 13). We define $\overrightarrow{\mathcal{RO}} := \mathcal{C}_1^{\mathcal{M}} \times \mathcal{C}_2^{\mathcal{M}^2} \times \cdots \times \mathcal{C}_\mu^{\mathcal{M}^\mu}$, the set from which the μ-tuple of random oracles $\overrightarrow{\mathsf{ro}}$ is sampled.[14]

To capture the behavior of a dishonest prover $\mathcal{P}^*$ attacking the non-interactive Fiat–Shamir transformation of a $(\Gamma_1, \dots, \Gamma_\mu)$-out-of-$(\mathcal{C}_1, \dots, \mathcal{C}_\mu)$ special-sound interactive proof (on input x), we consider an abstract algorithm of the form:

$$\mathcal{A}\colon \overrightarrow{\mathcal{RO}} \to \mathcal{M}^\mu \times \{0,1\}^* \,, \quad \overrightarrow{\mathsf{ro}} = (\mathsf{ro}_1, \dots, \mathsf{ro}_\mu) \mapsto \mathbf{a} = (a_1, \dots, a_\mu, a_{\mu+1}) \,.$$

Hence, we model the attacker as an algorithm $\mathcal{A}$ that, on input μ random oracles $\mathsf{ro}_1 \in \mathcal{C}_1^{\mathcal{M}}$, $\mathsf{ro}_2 \in \mathcal{C}_2^{\mathcal{M}^2}$, $\dots$, $\mathsf{ro}_\mu \in \mathcal{C}_\mu^{\mathcal{M}^\mu}$, outputs a proof $\pi = (a_1, \dots, a_\mu, a_{\mu+1})$. As for the 3-round case, we treat the random oracles $\mathsf{ro}_1, \dots, \mathsf{ro}_\mu$ for the different rounds as input with the understanding that $\mathcal{A}$ has oracle access only, and when considering an extractor that runs $\mathcal{A}$, the random oracle queries are answered using standard lazy sampling. When referring to the number of queries made by $\mathcal{A}$, we mean the total number of queries to the random oracles $\mathsf{ro}_1, \dots, \mathsf{ro}_\mu$.

The algorithm $\mathcal{A}$ is accompanied by a verification predicate

$$\mathcal{V}\colon \mathcal{M} \times \mathcal{C}_1 \times \mathcal{M} \times \cdots \times \mathcal{M} \times \mathcal{C}_\mu \times \{0,1\}^* \to \{0,1\} \,,$$

and the success of $\mathcal{A}$ is measured by checking if

$$\mathcal{V}\big(a_1, c_1, a_2, c_2, \dots, a_\mu, c_\mu, a_{\mu+1}\big) = 1 \quad \text{for} \quad c_i = \mathsf{ro}_i(a_1, \dots, a_i) \,.$$

We write the above as

$$\mathcal{V}^{\overrightarrow{\mathsf{ro}}}\big(a_1, a_2, \dots, a_\mu, a_{\mu+1}\big) = 1 \,,$$

and the success probability of $\mathcal{A}$ (as an attacker against the Fiat–Shamir transformed NIROP) is then given by

$$\epsilon^{\mathcal{V}}(\mathcal{A}) := \Pr\big(\mathcal{V}^{\overrightarrow{\mathsf{RO}}}(\mathcal{A}(\overrightarrow{\mathsf{RO}})) = 1\big) \,,$$

where $\overrightarrow{\mathsf{RO}}$ is distributed uniformly at random over $\overrightarrow{\mathcal{RO}}$. The natural instantiation of $\mathcal{V}$ is the verification predicate of the underlying interactive proof.

Below, we show the existence of a ($\mathcal{V}$-dependent, randomized) extractor for $\mathcal{A}$, denoted $\mathcal{F}_\mu^{\mathcal{A}}$, that aims to extract a tree $T \in \mathrm{TREE}_{\mathbf{\Gamma}}^{\mathcal{V}}(a_1)$ of $\mathcal{V}$-accepting transcripts $(a_1, c_1, \dots, a_\mu, c_\mu, a_{\mu+1})$ for some trunk a_1. Thus, the success probability of the extractor is given by $\Pr\big(\mathcal{W}_\mu(\mathcal{F}_\mu^{\mathcal{A}}) = 1\big)$, where $\mathcal{W}_\mu(T)$ checks

[14] Alternatively, the μ different random oracles could be implemented by (or modeled as) a single random oracle. The latter approach was followed in the multi-round Fiat–Shamir analysis of [AFK22]. Here, the above notation turned out to be more convenient.

whether $T \in \textsc{Tree}^{\mathcal{V}}_{\mathbf{\Gamma}}$. Note that, without loss of generality, we may assume that $\mathcal{F}^{\mathcal{A}}_{\mu}$'s output is indeed such a tree, or $\perp$ otherwise.

In order for the inductive analysis to go through, we need that $\mathcal{F}_{\mu}$ additionally satisfies the following technical property, which resembles the property on $\mathcal{A}$ necessary for Lemma 7 to be applicable (and this is of course no coincidence).

Definition 12 (Early-Choice Property). *An extractor $\mathcal{F}^{\mathcal{A}}_{\mu}$ satisfies the* early-choice *property, if*

1. *it runs $(a_1, \ldots, a_{\mu+1}) \leftarrow \mathcal{A}(\vec{\mathsf{ro}})$ as first step for uniformly random $\vec{\mathsf{ro}}$, and*
2. *if it does not abort (i.e., $\mathcal{F}^{\mathcal{A}}_{\mu}$ does not output $\perp$), it outputs $T \in \textsc{Tree}^{\mathcal{V}}_{\mathbf{\Gamma}}(a_1)$ with trunk a_1.*

The above early-choice property implies that, already after its first $\mathcal{A}$-invocation, the extractor $\mathcal{F}^{\mathcal{A}}_{\mu}$ knows the trunk a_1 of the tree it is going to output if it does not abort.

6.4 Recursive Extractor Construction

We show the existence of the extractor $\mathcal{F}_{\mu}$ by means of a recursive construction. For $\mu = 1$, $\mathcal{F}^{\mathcal{A}}_{1}$ is simply the extractor from the 3-round case (Figure 2). The early-abort property is satisfied by construction.

For $\mu > 1$, we consider

$$\mathcal{A}[\mathsf{ro}_1] : (\mathsf{ro}_2, \ldots, \mathsf{ro}_{\mu}) \mapsto ((a_1, a_2), a_3, \ldots, a_{\mu}, a_{\mu+1}) = \mathcal{A}(\mathsf{ro}_1, \mathsf{ro}_2, \ldots, \mathsf{ro}_{\mu})$$

which acts as $\mathcal{A}$ but has the random oracle ro_1 fixed, and where the output $(a_1, a_2, a_3, \ldots, a_{\mu}, z)$ of $\mathcal{A}$ is parsed as indicated, i.e., the first two messages are combined so that $\mathcal{A}[\mathsf{ro}_1]$ is viewed as an algorithm that outputs μ messages (instead of $\mu + 1$). The algorithm $\mathcal{A}[\mathsf{ro}_1]$ comes with the obvious verification predicate $\mathcal{V}[\mathsf{ro}_1]$ that has ro_1 fixed as well, i.e.,

$$\mathcal{V}[\mathsf{ro}_1]\big((a_1, a_2), c_2, a_3, \ldots, c_{\mu}, a_{\mu+1}\big) = \mathcal{V}\big(a_1, \mathsf{ro}_1(a_1), a_2, c_2, a_3, \ldots, c_{\mu}, a_{\mu+1}\big)\,.$$

Remark 8. Note that, by exploiting our particular choice of the multi-round Fiat–Shamir transformation where the second challenge is computed as $c_2 = \mathsf{ro}_2(a_1, a_2)$, it holds that

$$\mathcal{V}[\mathsf{ro}_1]^{\mathsf{ro}_2, \ldots, \mathsf{ro}_{\mu}}\big((a_1, a_2), a_3, \ldots, a_{\mu+1}\big) = \mathcal{V}^{\mathsf{ro}_1, \ldots, \mathsf{ro}_{\mu}}(a_1, a_2, a_3, \ldots, a_{\mu+1})\,.$$

The above inequality would not hold if the second challenge would be computed differently, e.g., as $\mathsf{ro}_2(a_2)$.

By applying recursion to $\mathcal{A}[\mathsf{ro}_1]$ and $\mathcal{V}[\mathsf{ro}_1]$, there exists a $\mathcal{V}[\mathsf{ro}_1]$-dependent, and thus ro_1-dependent, extractor

$$\mathcal{F}^{\mathcal{A}[\mathsf{ro}_1]}_{\mu-1} \to T'$$

that (unless it aborts) outputs a tree $T' \in \textsc{Tree}^{\mathcal{V}[\mathsf{ro}_1]}_{\Gamma_2, \ldots, \Gamma_{\mu}}(a_1, a_2)$ for some trunk (a_1, a_2); the corresponding (ro_1-dependent) verification predicate $\mathcal{W}_{\mu-1}[\mathsf{ro}_1]$ thus checks if the output of $\mathcal{F}_{\mu-1}$ is indeed such a tree.

We then consider the (probabilistic) algorithm

$$\mathcal{B}^{\mathcal{A}} \colon \mathcal{C}_1^{\mathcal{M}} \to \big(\mathcal{M} \times \textsc{Tree}_{\Gamma_2, \ldots, \Gamma_{\mu}}\big) \cup \{\perp\}\,, \quad \mathsf{ro}_1 \mapsto \mathcal{F}^{\mathcal{A}[\mathsf{ro}_1]}_{\mu-1}\,,$$

that, on input ro_1, runs $\mathcal{F}^{\mathcal{A}[\mathsf{ro}_1]}_{\mu-1}$ and where, unless it aborts, the output $T' = \{(a_1, a_2)\} \times T'' \in \textsc{Tree}_{\Gamma_2, \ldots, \Gamma_{\mu}}(a_1, a_2)$ is parsed as $(a_1, z) = (a_1, \{a_2\} \times T'')$, where then $\{a_2\} \times T'' \in \textsc{Tree}_{\Gamma_2, \ldots, \Gamma_{\mu}}(a_2)$. We note that, here and below, we merely do some obvious reformatting, but spelling it out is somewhat cumbersome.

Now note that the algorithm $\mathcal{B}^{\mathcal{A}}$ is naturally equipped with a verification predicate, which we denote by $\mathcal{V}_1$. This predicate accepts a transcript $(a_1, c_1, z) = (a_1, c_1, \{a_2\} \times T'')$ if $\{a_2\} \times T'' \in \textsc{Tree}_{\Gamma_2, \ldots, \Gamma_{\mu}}(a_2)$ and all transcripts $(a_1, c_1, a_2, c_2, \ldots) \in \{(a_1, c_1, a_2)\} \times T''$ are $\mathcal{V}$-accepting.

We note that, by construction, the success probability of $\mathcal{B}^{\mathcal{A}}$ on input ro_1 equals the success probability of the extractor $\mathcal{F}^{\mathcal{A}[\mathsf{ro}_1]}_{\mu-1}$, i.e., formally, $\mathcal{V}_1^{\mathsf{ro}_1}(a_1, \{a_2\} \times T'') = \mathcal{W}_{\mu-1}[\mathsf{ro}_1](\{(a_1, a_2)\} \times T'')$.

Note that the algorithm $\mathcal{B}^{\mathcal{A}}$ and its verification predicate $\mathcal{V}_1$ are of precisely the form required by our base case extractor $\mathcal{F}_1$. For this reason, the extractor $\mathcal{F}_\mu$ is given by

$$\mathcal{F}_\mu^{\mathcal{A}} := \mathcal{F}_1^{\mathcal{B}^{\mathcal{A}}} ,$$

with the early-abort strategy as specified in Figure 2. The early-abort strategy can be exploited in Lemma 7), using that, by induction, $\mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{ro}_1]}$ has the early-choice property (see Lemma 8), and therefore $\mathcal{B}^{\mathcal{A}}$ satisfies the requirement of having unit cost (measured in the number of queries to $\mathcal{A}$) to compute a_1.

By construction, $\mathcal{F}_1^{\mathcal{B}^{\mathcal{A}}}$ aims to output a $\mathit{\Gamma}_1$-tree of $\mathcal{V}_1$-accepting transcripts with a trunk a_1 (but different challenges c_1), which then forms a $\mathbf{\Gamma}$-tree of $\mathcal{V}$-accepting transcripts.

6.5 Success Probability

We begin by inductively arguing that our extractor succeeds with the desired probability. First, we require a lemma bounding the q-value $q(\mathcal{B}^{\mathcal{A}})$, as defined in Definition 8. Recall that, for the base case $\mu = 1$, i.e., when considering Fiat–Shamir transformations of $\mathit{\Sigma}$-protocols, Lemma 6 shows that $q(\mathcal{A}) \leq Q + 1$, where Q denotes the number of random oracle queries made by $\mathcal{A}$. More generally, we wish to show that for any μ and for $\mathcal{B}^{\mathcal{A}}$ defined as above it holds that $q(\mathcal{B}^{\mathcal{A}}) \leq Q + 1$, where again Q denotes the query complexity of $\mathcal{A}$. In fact, this bound will be a consequence of Lemma 6, as our multi-round extractor satisfies the *early-choice property.*

Lemma 8. *For any $\mu \geq 1$, the extractor $\mathcal{F}_\mu^{\mathcal{A}}$ $(= \mathcal{F}_1^{\mathcal{B}^{\mathcal{A}}})$ satisfies the early-choice property of Definition 12.*

Proof. For $\mu = 1$, $\mathcal{F}_1^{\mathcal{A}}$ is simply the extractor from the 3-round case (Figure 2). The early-choice property is therefore satisfied by construction.

Via induction, the claim for $\mu > 1$ then follows from transitivity. Namely, the extractor $\mathcal{F}_1^{\mathcal{B}^{\mathcal{A}}}$ runs $\mathcal{B}^{\mathcal{A}}$ as a first step, with ro_1 sampled uniformly at random, and eventually outputs the obtained a_1. Further, by the induction hypothesis $\mathcal{B}^{\mathcal{A}}(\mathsf{ro}_1) = \mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{ro}_1]}$ runs $\mathcal{A}$ as a first step, with the given ro_1 and uniformly random $\mathsf{ro}_2, \ldots, \mathsf{ro}_\mu$, and eventually outputs the obtained (a_1, a_2). It thus follows that $\mathcal{F}_1^{\mathcal{B}^{\mathcal{A}}}$ runs $\mathcal{A}(\mathsf{ro}_1, \ldots, \mathsf{ro}_\mu)$, with uniformly random $\mathsf{ro}_1, \ldots, \mathsf{ro}_\mu$, as a first step and eventually outputs the obtained a_1. □

Lemma 9. *For any algorithm $\mathcal{A}\colon \mathcal{C}_1^{\mathcal{M}} \times \mathcal{C}_2^{\mathcal{M}^2} \times \cdots \times \mathcal{C}_\mu^{\mathcal{M}^\mu} \to \mathcal{M}^\mu \times \{0,1\}^*$ making at most Q queries (in total) to its random oracles, and for $\mathcal{B}$ as in the above recursive construction,*

$$q(\mathcal{B}^{\mathcal{A}}) \leq Q + 1 .$$

Proof. By definition, $\mathcal{B}^{\mathcal{A}}$ takes only one random oracle ro_1 as input. Further, by the early-choice property of $\mathcal{F}_{\mu-1}$ (Lemma 8), $\mathcal{B}^{\mathcal{A}}(\mathsf{ro}_1) = \mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{ro}_1]}$ either aborts or its first output equals the first output of $\mathcal{A}(\mathsf{ro}_1, \mathsf{ro}_2, \ldots, \mathsf{ro}_\mu)$, for $\mathsf{ro}_2, \ldots, \mathsf{ro}_\mu$ sampled uniformly at random. In particular, $\mathsf{ro}_2, \ldots, \mathsf{ro}_\mu$ are sampled independently from the input ro_1, and thus uniquely determined by the random coins r of $\mathcal{B}^{\mathcal{A}}$. Let us now fix these random coins r and write $\mathcal{B}^{\mathcal{A}}[r](\mathsf{ro}_1)$ for the evaluation of $\mathcal{B}^{\mathcal{A}}$ with random coins r and input ro_1.

Then, by the above, the first output of $\mathcal{B}^{\mathcal{A}}[r](\mathsf{ro}_1)$ equals $\mathcal{A}(\mathsf{ro}_1, \mathsf{ro}_2^r, \ldots, \mathsf{ro}_\mu^r)$ for some fixed random oracles $\mathsf{ro}_2^r, \ldots, \mathsf{ro}_\mu^r$. For this reason,

$$q(\mathcal{B}^{\mathcal{A}}[r]) \leq q(\mathcal{A}(\cdot, \mathsf{ro}_2^r, \ldots, \mathsf{ro}_\mu^r)) ,$$

where $\mathcal{A}(\cdot, \mathsf{ro}_2^r, \ldots, \mathsf{ro}_\mu^r)\colon \mathcal{C}_1^{\mathcal{M}} \to \mathcal{M}^\mu \times \{0,1\}^*$ makes at most Q queries to the random oracles $\mathsf{ro}_1, \mathsf{ro}_2^r, \ldots, \mathsf{ro}_\mu^r$; in particular, it makes at most Q queries to ro_1.[15] Finally, by Lemma 6, the right-hand side of the above inequality can thus be bounded by $Q + 1$. The lemma then follows by taking the expectation over the random coins r. □

We are now ready to provide a lower-bound for success probability of the recursive extractor.

[15] It is thus easily seen that, in the lemma statement, a bound Q on the queries to each random oracle (instead of bound on the total number of queries) would have sufficed.

Proposition 1. *Let* $\mathcal{A}\colon \mathcal{C}_1^{\mathcal{M}} \times \mathcal{C}_2^{\mathcal{M}^2} \times \cdots \times \mathcal{C}_\mu^{\mathcal{M}^\mu} \to \mathcal{M}^\mu \times \{0,1\}^*$ *be a Q-query algorithm. Then, the success probability of the extractor* $\mathcal{F}_\mu^{\mathcal{A}}$ *satisfies*

$$\Pr\bigl(\mathcal{F}_\mu^{\mathcal{A}} \neq \bot\bigr) = \Pr\bigl(\mathcal{W}_\mu(\mathcal{F}_\mu^{\mathcal{A}}) = 1\bigr) \geq \frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - (Q+1)\cdot\kappa_{\mathbf{\Gamma}}}{1-\kappa_{\mathbf{\Gamma}}}\,.$$

Proof. The proof is by induction. The base case $\mu = 1$ follows from Theorem 2 and Lemma 6. Thus, consider now $\mu > 1$. First, setting $\mathbf{\Gamma}' = (\mathit{\Gamma}_2, \ldots, \mathit{\Gamma}_\mu)$ and using that $\mathcal{B}^{\mathcal{A}}(\mathsf{ro}_1) = \mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{ro}_1]}$, we note that by the induction hypothesis

$$\Pr\Bigl(\mathcal{V}_1^{\mathsf{ro}_1}\bigl(\mathcal{B}^{\mathcal{A}}(\mathsf{ro}_1)\bigr) = 1\Bigr) = \Pr\Bigl(\mathcal{W}_{\mu-1}[\mathsf{ro}_1]\bigl(\mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{ro}_1]}\bigr) = 1\Bigr) \geq \frac{\epsilon^{\mathcal{V}[\mathsf{ro}_1]}(\mathcal{A}[\mathsf{ro}_1]) - (Q+1)\cdot\kappa_{\mathbf{\Gamma}'}}{1-\kappa_{\mathbf{\Gamma}'}}\,.$$

Hence,

$$\begin{aligned}
\epsilon^{\mathcal{V}_1}(\mathcal{B}^{\mathcal{A}}) &= \Pr\bigl(\mathcal{V}_1^{\mathsf{RO}_1}(\mathcal{B}^{\mathcal{A}}(\mathsf{RO}_1)) = 1\bigr) = \Pr\Bigl(\mathcal{W}_{\mu-1}[\mathsf{RO}_1]\bigl(\mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{RO}_1]}\bigr) = 1\Bigr) \\
&= \mathop{\mathbb{E}}_{\mathsf{ro}_1}\Bigl[\Pr\Bigl(\mathcal{W}_{\mu-1}[\mathsf{ro}_1]\bigl(\mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{ro}_1]}\bigr) = 1\Bigr)\Bigr] \\
&\geq \frac{\mathbb{E}_{\mathsf{ro}_1}\bigl[\epsilon^{\mathcal{V}[\mathsf{ro}_1]}(\mathcal{A}[\mathsf{ro}_1])\bigr] - (Q+1)\cdot\kappa_{\mathbf{\Gamma}'}}{1-\kappa_{\mathbf{\Gamma}'}}\,.
\end{aligned}$$

By using that

$$\begin{aligned}
\epsilon^{\mathcal{V}[\mathsf{ro}_1]}(\mathcal{A}[\mathsf{ro}_1]) &= \Pr\bigl[\mathcal{V}[\mathsf{ro}_1]^{\mathsf{RO}_2,\ldots,\mathsf{RO}_\mu}(\mathcal{A}[\mathsf{ro}_1](\mathsf{RO}_2,\ldots,\mathsf{RO}_\mu)) = 1\bigr] \\
&= \Pr\bigl[\mathcal{V}^{\mathsf{ro}_1,\mathsf{RO}_2,\ldots,\mathsf{RO}_\mu}(\mathcal{A}(\mathsf{ro}_1,\mathsf{RO}_2,\ldots,\mathsf{RO}_\mu)) = 1\bigr]\,,
\end{aligned}$$

where the second equality exploits the particular definition of the Fiat–Shamir transformation (see Remark 8), we see that $\mathbb{E}_{\mathsf{ro}_1}\bigl[\epsilon^{\mathcal{V}[\mathsf{ro}_1]}(\mathcal{A}[\mathsf{ro}_1])\bigr] = \epsilon^{\mathcal{V}}(\mathcal{A})$. Hence,

$$\epsilon^{\mathcal{V}_1}(\mathcal{B}^{\mathcal{A}}) \geq \frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - (Q+1)\cdot\kappa_{\mathbf{\Gamma}'}}{1-\kappa_{\mathbf{\Gamma}'}}\,.$$

Furthermore, by Theorem 2, exploiting the bound on the q-value from Lemma 9,

$$\Pr\bigl(\mathcal{F}_\mu^{\mathcal{A}} \neq \bot\bigr) = \Pr\bigl(\mathcal{F}_1^{\mathcal{B}^{\mathcal{A}}} \neq \bot\bigr) \geq \frac{\epsilon^{\mathcal{V}_1}(\mathcal{B}^{\mathcal{A}}) - (Q+1)\cdot\kappa_{\mathit{\Gamma}_1}}{1-\kappa_{\mathit{\Gamma}_1}}\,.$$

Thus, writing ϵ for $\epsilon^{\mathcal{V}}(\mathcal{A})$ and exploiting that $(1-\kappa_{\mathit{\Gamma}_1})(1-\kappa_{\mathbf{\Gamma}'}) = 1-\kappa_{\mathbf{\Gamma}}$,

$$\begin{aligned}
\Pr\bigl(\mathcal{F}_\mu^{\mathcal{A}} \neq \bot\bigr) &\geq \frac{\frac{\epsilon-(Q+1)\cdot\kappa_{\mathbf{\Gamma}'}}{1-\kappa_{\mathbf{\Gamma}'}} - (Q+1)\cdot\kappa_{\mathit{\Gamma}_1}}{1-\kappa_{\mathit{\Gamma}_1}} \\
&= \frac{\epsilon - (Q+1)\cdot\kappa_{\mathbf{\Gamma}'} - (Q+1)\cdot\kappa_{\mathit{\Gamma}_1}\cdot(1-\kappa_{\mathbf{\Gamma}'})}{1-\kappa_{\mathbf{\Gamma}}} \\
&= \frac{\epsilon - (Q+1)\bigl(\kappa_{\mathbf{\Gamma}'} + \kappa_{\mathit{\Gamma}_1}\cdot(1-\kappa_{\mathbf{\Gamma}'})\bigr)}{1-\kappa_{\mathbf{\Gamma}}} \\
&= \frac{\epsilon - (Q+1)\bigl(1-(1-\kappa_{\mathbf{\Gamma}'})(1-\kappa_{\mathit{\Gamma}_1})\bigr)}{1-\kappa_{\mathbf{\Gamma}}} \\
&= \frac{\epsilon - (Q+1)\cdot\kappa_{\mathbf{\Gamma}}}{1-\kappa_{\mathbf{\Gamma}}}\,,
\end{aligned}$$

which completes the proof. □

6.6 Running Time Analysis

Let us now analyze the expected running time of our knowledge extractor.

Proposition 2. *Let $\Gamma_1, \dots, \Gamma_\mu$ be monotone structures and let $\mathbf{\Gamma} = (\Gamma_1, \dots, \Gamma_\mu)$. Let $\mathcal{A}\colon \overrightarrow{\mathcal{RO}} \to \mathcal{M}^\mu \times \{0,1\}^*$ be a Q-query algorithm, and let the extractor $\mathcal{F}_\mu$ be defined as above. Let the random variable $X_{\mathbf{\Gamma}}$ denote the number of $\mathcal{A}$-queries extractor $\mathcal{F}_\mu^{\mathcal{A}}$ makes. Then*

$$\mathbb{E}[X_{\mathbf{\Gamma}}] \leq \frac{T_{\mathbf{\Gamma}} + (T_{\mathbf{\Gamma}} - 1)Q}{1 - \kappa_{\mathbf{\Gamma}}} ,$$

where $T_{\mathbf{\Gamma}} = \prod_{j=1}^{\mu} (t_{\Gamma_j} + 1)$.

Proof. The proof is by induction over μ. The base case $\mu = 1$ follows from Theorem 2 together with the bound $q(\mathcal{A}) \leq Q + 1$ from Lemma 6. These results namely state that the expected number of $\mathcal{A}$-queries made by $\mathcal{F}_1^{\mathcal{A}} = \mathcal{F}_{\Gamma_1}^{\mathcal{A}}$ is at most

$$\begin{aligned} 1 + q(\mathcal{A}) \cdot t_{\Gamma_1} \cdot (1 + \kappa_{\Gamma_1}) &\leq 1 + \frac{q(\mathcal{A}) \cdot t_{\Gamma_1}}{1 - \kappa_{\Gamma_1}} \leq \frac{1 + q(\mathcal{A}) \cdot t_{\Gamma_1}}{1 - \kappa_{\Gamma_1}} \\ &\leq \frac{1 + (Q+1) \cdot t_{\Gamma_1}}{1 - \kappa_{\Gamma_1}} = \frac{t_{\Gamma_1} + 1 + t_{\Gamma_1} Q}{1 - \kappa_{\Gamma_1}} = \frac{T_{\Gamma_1} + (T_{\Gamma_1} - 1)(Q+1)}{1 - \kappa_{\Gamma_1}} , \end{aligned}$$

where we use that $1/(1-\kappa_1) \geq 1 + \kappa_{\Gamma_1} \geq 1$. We note that this bound could also have been obtained by using the refined running time analysis of Lemma 7, then using that $\theta(\mathsf{ro}) = 1$ is constant.

For the induction step, let $\mathbf{\Gamma}' = (\Gamma_2, \dots, \Gamma_\mu)$ and thus $T_{\mathbf{\Gamma}'} = \prod_{j=2}^{\mu} (t_{\Gamma_j} + 1)$. By construction, $\mathcal{F}_\mu^{\mathcal{A}} = \mathcal{F}_1^{\mathcal{B}^{\mathcal{A}}}$ where $\mathcal{B}^{\mathcal{A}}(\mathsf{ro}_1) = \mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{ro}_1]}$, up to some reformatting of the output.

By the early-choice property of $\mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{ro}_1]}$, the algorithm $\mathcal{B}^{\mathcal{A}}$ satisfies the requirement on $\mathcal{A}$ in Lemma 7, with the cost θ of $\mathcal{B}^{\mathcal{A}}$ measuring the expected number of calls to $\mathcal{A}$. Indeed, finding a_1 as in $(a_1, z) \leftarrow \mathcal{B}^{\mathcal{A}}(\mathsf{ro}_1)$ can be done with one query to $\mathcal{A}$, while finding a subtree requires a full run of $\mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{ro}_1]}$, and thus costs (in expectation) $\theta(\mathsf{ro}_1)$ calls to $\mathcal{A}$, where, by induction hypothesis,

$$\mathbb{E}[\theta(\mathsf{RO}_1)] = \mathbb{E}[X_{\mathbf{\Gamma}'}] \leq \frac{T_{\mathbf{\Gamma}'} + (T_{\mathbf{\Gamma}'} - 1)Q}{1 - \kappa_{\mathbf{\Gamma}'}} .$$

Thus, by Lemma 7, and using the bound on $q(\mathcal{B}^{\mathcal{A}})$ from Lemma 9, we find

$$\begin{aligned} \mathbb{E}[X_{\mathbf{\Gamma}}] &\leq \mathbb{E}[\theta(\mathsf{RO}_1)] + t_{\Gamma_1} \cdot \frac{\mathbb{E}[\theta(\mathsf{RO}_1)] - 1 + q(\mathcal{B}^{\mathcal{A}})}{1 - \kappa_{\Gamma_1}} \\ &\leq \frac{(t_{\Gamma_1} + 1)\, \mathbb{E}[\theta(\mathsf{RO}_1)] + t_{\Gamma_1} Q}{1 - \kappa_{\Gamma_1}} \\ &\leq \frac{(t_{\Gamma_1} + 1)(\frac{T_{\mathbf{\Gamma}'} + (T_{\mathbf{\Gamma}'} - 1)Q}{1 - \kappa_{\mathbf{\Gamma}'}}) + t_{\Gamma_1} Q}{1 - \kappa_{\Gamma_1}} \\ &\leq \frac{(t_{\Gamma_1} + 1)(T_{\mathbf{\Gamma}'} + (T_{\mathbf{\Gamma}'} - 1)Q) + t_{\Gamma_1} Q}{1 - \kappa_{\mathbf{\Gamma}}} \\ &= \frac{(t_{\Gamma_1} + 1)T_{\mathbf{\Gamma}'} + ((t_{\Gamma_1} + 1)(T_{\mathbf{\Gamma}'} - 1) + t_{\Gamma_1})Q}{1 - \kappa_{\mathbf{\Gamma}}} \\ &= \frac{T_{\mathbf{\Gamma}} + (T_{\mathbf{\Gamma}} - 1)Q}{1 - \kappa_{\mathbf{\Gamma}}} , \end{aligned}$$

which completes the proof. □

6.7 Knowledge Soundness of the Fiat–Shamir Transformation

The following theorem summarizes the properties of the recursive extraction algorithm defined in Section 6.4. In fact, from the success probability bound derived in Section 6.5 and the running time bound derived in Section 6.6, it immediately follows that the Fiat–Shamir transformation of a $(\Gamma_1, \dots, \Gamma_\mu)$-out-of-$(\mathcal{C}_1, \dots, \mathcal{C}_\mu)$ special-sound interactive proof is knowledge sound (under certain conditions). Moreover, the knowledge error only grows linearly in the query complexity Q of dishonest provers attacking the non-interactive proof, and is independent of the number of rounds. This is the main result of this paper.

Theorem 4 (Knowledge Soundness - Fiat–Shamir Transformation of Multi-Round Interactive Proofs). *Let $(\mathit{\Gamma}_i, \mathcal{C}_i)$ be nonempty monotone structures, and let $\mathbf{\Gamma} := (\mathit{\Gamma}_1, \ldots, \mathit{\Gamma}_\mu)$ and $\boldsymbol{\mathcal{C}} := (\mathcal{C}_1, \ldots, \mathcal{C}_\mu)$. Let $\mathit{\Pi}$ be a $\mathbf{\Gamma}$-out-of-$\boldsymbol{\mathcal{C}}$ special-sound interactive proof. Suppose that $T_{\mathbf{\Gamma}} = \prod_{i=1}^{\mu}(t_{\mathit{\Gamma}_i}+1)$ is polynomial in the size $|x|$ of the statement x and that, for all i, sampling from $\mathcal{U}_{\mathit{\Gamma}_i}(S_i)$ takes polynomial time (in $|x|$) for all $S_i \subseteq \mathcal{C}_i$ with $|S_i| \leq t_{\mathit{\Gamma}_i}$. Then the Fiat–Shamir transformation $\mathsf{FS}[\mathit{\Pi}]$ of $\mathit{\Pi}$ is knowledge sound with knowledge error*

$$(Q+1) \cdot \kappa_{\mathbf{\Gamma}} \,,$$

where $\kappa_{\mathbf{\Gamma}} = 1 - \prod_{i=1}^{\mu}(1 - \kappa_{\mathit{\Gamma}_i})$ is the knowledge error[16] *of the interactive proof $\mathit{\Pi}$ and Q denotes the query complexity of a prover attacking $\mathsf{FS}[\mathit{\Pi}]$.*

More precisely, $\mathsf{FS}[\mathit{\Pi}]$ admits a knowledge extractor that, on input a statement x and given oracle access to a Q-query prover $\mathcal{P}^*$ makes at most

$$\frac{T_{\mathbf{\Gamma}} + (T_{\mathbf{\Gamma}} - 1)Q}{1 - \kappa_{\mathbf{\Gamma}}}$$

queries to $\mathcal{P}^*$ in expectation, and succeeds to extract a witness for x with probability at least

$$\frac{\epsilon(\mathcal{P}^*, x) - (Q+1) \cdot \kappa_{\mathbf{\Gamma}}}{1 - \kappa_{\mathbf{\Gamma}}} \,.$$

Proof. This follows immediately from Propositions 1 and 2. □

Acknowledgments

The first author has been supported by TNO's Early Research Program - Next Generation Cryptography, the Vraaggestuurd Programma Cyber Security & Resilience, part of the Dutch Top Sector High Tech Systems and Materials program, and by the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe Research and Innovation Programme in the scope of the CONFIDENTIAL6G project under Grant Agreement 101096435. The contents of this publication are the sole responsibility of the authors and do not in any way reflect the views of the EU. The third author has been supported by Helsinki Institute for Information Technology (HIIT).

References

ACK21. Thomas Attema, Ronald Cramer, and Lisa Kohl. A compressed Σ-protocol theory for lattices. In Tal Malkin and Chris Peikert, editors, *CRYPTO*, volume 12826 of *Lecture Notes in Computer Science*, pages 549–579. Springer, 2021.

ACY23. Gal Arnon, Alessandro Chiesa, and Eylon Yogev. Iops with inverse polynomial soundness error. *IACR Cryptol. ePrint Arch.*, page 1062, 2023.

AF22. Thomas Attema and Serge Fehr. Parallel repetition of $(k_1, \ldots, k_\mu)$-special-sound multi-round interactive proofs. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO*, volume 13507 of *Lecture Notes in Computer Science*, pages 415–443. Springer, 2022.

AFK22. Thomas Attema, Serge Fehr, and Michael Klooß. Fiat-shamir transformation of multi-round interactive proofs. In Eike Kiltz and Vinod Vaikuntanathan, editors, *Theory of Cryptography Conference (TCC)*, volume 13747 of *Lecture Notes in Computer Science*, pages 113–142. Springer, 2022.

AFK23. Thomas Attema, Serge Fehr, and Michael Klooß. Fiat-shamir transformation of multi-round interactive proofs (extended version). *Journal of Cryptology*, 36(4):36, 2023.

AFR23. Thomas Attema, Serge Fehr, and Nicolas Resch. Generalized special-sound interactive proofs and their knowledge soundness. In Guy N. Rothblum and Hoeteck Wee, editors, *Theory of Cryptography - 21st International Conference, TCC 2023, Taipei, Taiwan, November 29 - December 2, 2023, Proceedings, Part III*, volume 14371 of *Lecture Notes in Computer Science*, pages 424–454. Springer, 2023.

ALS20. Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. Practical product proofs for lattice commitments. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO*, volume 12171 of *Lecture Notes in Computer Science*, pages 470–499. Springer, 2020.

Att23. Thomas Attema. *Compressed Σ-Protocol Theory.* PhD thesis, Leiden University, 2023.

[16] Recall that $\kappa_{\mathit{\Gamma}_i} = \max_{S_i \notin \mathit{\Gamma}_i} |S_i| \,/\, |\mathcal{C}|$.

BBB$^+$18. Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Gregory Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *IEEE Symposium on Security and Privacy (S&P)*, pages 315–334. IEEE Computer Society, 2018.

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 Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO*, volume 10992 of *Lecture Notes in Computer Science*, pages 669–699. Springer, 2018.

BBHR18. Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast reed-solomon interactive oracle proofs of proximity. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

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 Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT*, volume 9666 of *Lecture Notes in Computer Science*, pages 327–357. Springer, 2016.

BPW12. David Bernhard, Olivier Pereira, and Bogdan Warinschi. How not to prove yourself: Pitfalls of the fiat-shamir heuristic and applications to helios. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 626–643. Springer, 2012.

CD98. Ronald Cramer and Ivan Damgård. Zero-knowledge proofs for finite field arithmetic; or: Can zero-knowledge be for free? In Hugo Krawczyk, editor, *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 424–441. Springer, 1998.

Cra96. Ronald Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, CWI and University of Amsterdam, 1996.

ENS20. Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 259–288. Springer, 2020.

KL23. Michael Klooß and Russel Lai. Personal communication: Adaptive special-soundness, 2023.

LNP22. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 71–101. Springer, 2022.

LW22. George Lu and Brent Waters. How to sample a discrete gaussian (and more) from a random oracle. In Eike Kiltz and Vinod Vaikuntanathan, editors, *Theory of Cryptography - 20th International Conference, TCC 2022, Chicago, IL, USA, November 7-10, 2022, Proceedings, Part II*, volume 13748 of *Lecture Notes in Computer Science*, pages 653–682. Springer, 2022.

RR22. Noga Ron-Zewi and Ron D. Rothblum. Proving as fast as computing: succinct arguments with constant prover overhead. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1353–1363. ACM, 2022.

Wik18. Douglas Wikström. Special soundness revisited. *IACR Cryptology ePrint Archive*, 2018.

Wik21. Douglas Wikström. Special soundness in the random oracle model. *IACR Cryptology ePrint Archive*, 2021.

Appendix

A Definition - Fiat–Shamir Transformation

We provide a formal definition of the Fiat–Shamir transform, as discussed in Section 2.3. For simplicity, we assume access to multiple independent random oracles $\mathsf{ro}_1, \ldots, \mathsf{ro}_\mu$, where ro_i maps tuples of i prover messages to the appropriate challenge space.

Definition 13 (Static Fiat–Shamir Transformation). *Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a $2\mu+1$-move public-coin interactive proof for a relation R with i-th challenge set $\mathcal{C}_i$, and such that all potential prover messages are contained in $\mathcal{M}$. Let $\mathsf{ro}_1, \ldots, \mathsf{ro}_\mu$ be independent random oracles such that $\mathsf{ro}_i \colon \mathcal{M}^i \to \mathcal{C}_i$ for all $i = 1, \ldots, \mu$.*

The static Fiat–Shamir transformation $\mathsf{FS}[\Pi] = (\mathcal{P}_{\mathsf{fs}}, \mathcal{V}_{\mathsf{fs}})$ *of the protocol Π is the following NIROP. Given $(x; w) \in R$, the NIROP prover $\mathcal{P}_{\mathsf{fs}}^{\mathsf{ro}_1,\ldots,\mathsf{ro}_\mu}$ simulates the interactive proof prover $\mathcal{P}(x; w)$ such that, after it outputs the i-th message a_i, it is provided with the challenge*

$$c_i = \mathsf{ro}_i(a_1, \ldots, a_{i-1}, a_i) \tag{15}$$

for all i. The prover $\mathcal{P}_{\mathsf{fs}}^{\mathsf{ro}_1,\ldots,\mathsf{ro}_\mu}$ then outputs the proof $\pi = (a_1, a_2, \ldots, a_{\mu+1})$. On input a statement x and a proof $\pi = (a_1, a_2, \ldots, a_{\mu+1})$, the verifier $\mathcal{V}_{\mathsf{fs}}^{\mathsf{ro}_1,\ldots,\mathsf{ro}_\mu}$ accepts if and only if $\mathcal{V}$ accepts the transcript $(a_1, c_1, a_2, \ldots, a_\mu, c_\mu, a_{\mu+1})$ on input x, where each c_i is computed as in (15).

Remark 9 (Adaptive Fiat–Shamir Transformation). In the adaptive setting, a dishonest prover may adaptively choose the statement x. Thus, in the adaptive Fiat–Shamir transformation, the challenges from (15) are instead computed as

$$c_i = \mathsf{ro}_i(x, a_1, \ldots, a_{i-1}, a_i) \tag{16}$$

for $i = 1, \ldots, \mu$. Proving and verification proceed mutatis mutandis as before.

B Analysis of the Extractor for Interactive Proofs

This appendix provides the proof of Theorem 1, which describes the properties of the improved extractor for Γ-out-of-$\mathcal{C}$ special-sound Σ-protocols of [KL23]. We have merely adapted the results from [KL23] to our notation. Before we prove Theorem 1, we present two auxiliary lemmas. Further, this section makes use of the notation introduced in Section 4.

The following lemma provides an useful bound for a specific composition of consecutive Δ-values $\Delta(\mathbf{c}_0), \ldots, \Delta(\mathbf{c}_k)$, as defined in Equation (6). This lemma crucially relies on the fact that $U_i \implies U_{i-1}$ for all i (Equation (5)). Recall that U_i denotes the event $C \in \mathcal{U}_\Gamma(\mathbf{c}_i)$, where C is uniformly distributed in $\mathcal{C}$. To give some intuition, we note that for a trivial V that occurs with certainty, so that $\Delta(\mathbf{c}_k) = \Pr(\neg U_k \mid U_{k-1})$, the inequality becomes the (trivial) equality

$$\Pr\bigl(U_k\bigr) = \Pr\bigl(U_k \wedge \ldots \wedge U_1\bigr) = \prod_{i=1}^{k} \Pr\bigl(U_i \mid U_1 \wedge \ldots \wedge U_{i-1}\bigr) = \prod_{i=1}^{k} \Pr\bigl(U_i \mid U_{i-1}\bigr) .$$

For a non-trivial V, the proof is less straightforward.

Lemma 10. *Let $\mathbf{c} \in \mathcal{C}^k$ for some k and let $\mathbf{c}_i = (c_1, \ldots, c_i)$ for all $1 \leq i \leq k$. Then*

$$\prod_{i=1}^{k} \bigl(1 - \Delta(\mathbf{c}_i)\bigr) \leq 1 - \Pr\bigl(V \wedge \neg U_k\bigr) .$$

Proof. The proof of the lemma will now proceed inductively over k. The lemma trivially holds for the base case $k = 1$. So let us assume the lemma is proven for $k' = k - 1$. Then, by the induction hypothesis

$$\begin{aligned}
\prod_{i=1}^{k}\bigl(1 - \mathit{\Delta}(\mathbf{c}_i)\bigr) &\le \bigl(1 - \Pr(V \wedge \neg U_{k-1})\bigr) \cdot \bigl(1 - \mathit{\Delta}(\mathbf{c}_k)\bigr) \\
&= \bigl(1 - \Pr(V \wedge \neg U_{k-1})\bigr) \cdot \bigl(1 - \Pr(V \wedge \neg U_k \mid U_{k-1})\bigr) \\
&= \Pr(\neg V \vee U_{k-1}) \cdot \Pr(\neg V \vee U_k \mid U_{k-1}) \\
&= \bigl(\Pr(U_{k-1}) + \Pr(\neg V \wedge \neg U_{k-1})\bigr) \cdot \Pr(\neg V \vee U_k \mid U_{k-1}) \\
&\le \Pr\bigl((\neg V \vee U_k) \wedge U_{k-1}\bigr) + \Pr(\neg V \wedge \neg U_{k-1}) \\
&= \Pr\bigl((\neg V \wedge U_{k-1}) \vee (U_k \wedge U_{k-1})\bigr) + \Pr(\neg V \wedge \neg U_{k-1}) \\
&= \Pr\bigl((\neg V \wedge U_{k-1}) \vee U_k \vee (\neg V \wedge \neg U_{k-1})\bigr) \\
&= \Pr\bigl(\neg V \vee U_k\bigr) \\
&= 1 - \Pr\bigl(V \wedge \neg U_k\bigr) ,
\end{aligned}$$

which completes the proof of the lemma. □

The following auxiliary lemma provides a lower bound in terms of $\mathcal{A}$'s success probability $\epsilon(\mathcal{A})$ and the knowledge error κ_Γ. The left-hand side of this inequality will appear in the analysis of the success probability of the knowledge extractor.

Lemma 11. *Let $\mathbf{c}_k \in \mathcal{C}^k$ for some k with $\mathbf{c}_k \notin \Gamma$ and let $\mathbf{c}_i = (c_1, \ldots, c_i)$ for all $1 \le i \le k$. Then*

$$\delta(\mathbf{c}_0)\prod_{i=1}^{k}\frac{\tilde{\delta}(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} = \tilde{\delta}(\mathbf{c}_1)\prod_{i=2}^{k}\frac{\tilde{\delta}(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} \ge \frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - \kappa_\Gamma}{1 - \kappa_\Gamma} ,$$

where $\kappa_\Gamma := \max_{S \notin \Gamma} \frac{|S|}{|\mathcal{C}|}$.

Proof. First note that, for all $1 \le i \le k$,

$$\begin{aligned}
\delta(\mathbf{c}_{i-1}) - \mathit{\Delta}(\mathbf{c}_i) &= \Pr\bigl(V \mid U_{i-1}\bigr) - \Pr\bigl(V \wedge \neg U_i \mid U_{i-1}\bigr) = \Pr\bigl(V \wedge U_i \mid U_{i-1}\bigr) \\
&= \Pr\bigl(V \mid U_i \wedge U_{i-1}\bigr)\Pr\bigl(U_i \mid U_{i-1}\bigr) = \delta(\mathbf{c}_i) \cdot \Pr\bigl(U_i \mid U_{i-1}\bigr) ,
\end{aligned}$$

where we use that $U_i \implies U_{i-1}$.

Hence,

$$\tilde{\delta}(\mathbf{c}_i) = \frac{\delta(\mathbf{c}_{i-1}) - \mathit{\Delta}(\mathbf{c}_i)}{1 - \mathit{\Delta}(\mathbf{c}_i)} = \delta(\mathbf{c}_i)\frac{\Pr\bigl(U_i \mid U_{i-1}\bigr)}{1 - \mathit{\Delta}(\mathbf{c}_i)} ,$$

and

$$\begin{aligned}
\tilde{\delta}(\mathbf{c}_1)\prod_{i=2}^{k}\frac{\tilde{\delta}(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} &= \delta(\mathbf{c}_k)\prod_{i=1}^{k}\frac{\Pr\bigl(U_i \mid U_{i-1}\bigr)}{1 - \mathit{\Delta}(\mathbf{c}_i)} \\
&= \delta(\mathbf{c}_k)\Pr\bigl(U_k \mid U_0\bigr)\prod_{i=1}^{k}\frac{1}{1 - \mathit{\Delta}(\mathbf{c}_i)} \\
&= \delta(\mathbf{c}_k)\Pr\bigl(U_k\bigr)\prod_{i=1}^{k}\frac{1}{1 - \mathit{\Delta}(\mathbf{c}_i)} \\
&= \Pr\bigl(V \wedge U_k\bigr)\prod_{i=1}^{k}\frac{1}{1 - \mathit{\Delta}(\mathbf{c}_i)} ,
\end{aligned} \tag{17}$$

where the second equality follows again because $U_j \implies U_i$ for all $i \le j$. By Lemma 10, it thus follows that

$$\tilde{\delta}(\mathbf{c}_1)\prod_{i=2}^{k}\frac{\tilde{\delta}(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} \ge \frac{\Pr\bigl(V \wedge U_k\bigr)}{1 - \Pr\bigl(V \wedge \neg U_k\bigr)} = \frac{\Pr\bigl(V\bigr) - \Pr\bigl(V \wedge \neg U_k\bigr)}{1 - \Pr\bigl(V \wedge \neg U_k\bigr)} . \tag{18}$$

Now note that, since $\mathcal{C} \setminus \mathcal{U}_\Gamma(\mathbf{c}) \notin \Gamma$ for all $\mathbf{c} \notin \Gamma$ (see Equation (3)),

$$\Pr\bigl(V \wedge \neg U_k\bigr) \leq \Pr\bigl(\neg U_k\bigr) = \Pr\bigl(C \notin \mathcal{U}_\Gamma(\mathbf{c}_k)\bigr) \leq \max_{S \notin \Gamma} \frac{|S|}{|\mathcal{C}|} = \kappa_\Gamma \,.$$

Hence, by Equation (18) and the monotonicity (decreasing) of the function $x \mapsto \frac{q-x}{1-x}$ for all $0 \leq q \leq 1$, it follows that

$$\tilde{\delta}(\mathbf{c}_1) \prod_{i=2}^{k} \frac{\tilde{\delta}(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} \geq \frac{\Pr(V) - \kappa_\Gamma}{1 - \kappa_\Gamma} \geq \frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - \kappa_\Gamma}{1 - \kappa_\Gamma} \,,$$

which completes the proof of the lemma. □

We are no ready to present the proof of Theorem 1.

Proof of Theorem 1. The extractor $\mathcal{E}_\Gamma^{\mathcal{A}}$ is formally defined in Figure 1. It iteratively tries to find new challenges c_i, together with outputs y_i, such that $\mathcal{V}(c_i, y_i) = 1$, until it has collected a subset $\{c_1, \ldots, c_k\} \in \Gamma$. Further, for $i \geq 2$, every c_i is sampled from $\mathcal{U}_\Gamma(\{c_1, \ldots, c_{i-1}\})$, i.e., for all extractor outputs it holds that $k \leq t_\Gamma + 1$. Note that, $k = t_\Gamma + 1$ can only occur if the first challenge c_1 is in $\mathcal{C} \setminus \mathcal{U}_\Gamma(\emptyset)$, i.e., if c_1 is useless. Further, only in contrived interactive proofs $\mathcal{C} \neq \mathcal{U}_\Gamma(\emptyset)$, hence typically it even holds that $k \leq t_\Gamma$. See also Remark 3.

As before, for any $\mathbf{c} = (c_1, \ldots, c_k) \in \mathcal{C}^k$ and $1 \leq i \leq k$, we write $\mathbf{c}_i = (c_1, \ldots, c_i)$. Let us now analyze the success probability and the expected number of $\mathcal{A}$-queries of the extractor.

Success Probability. Let us write $t = t_\Gamma$. Let the random variable C_i denote the i-th successful challenge found by the extractor, where we let $C_i = \bot$ if the extractor finishes before finding i challenges, i.e., C_i has support in $\mathcal{C} \cup \{\bot\}$. Note that if $(C_1, \ldots, C_i) \in \Gamma$ then $C_{i+1} = \cdots = C_t = \bot$. Vice versa, $C_i = \bot$ implies that the extractor was either successful before the i-th iteration, or that it aborted and failed before the i-th iteration. Further, we write $\boldsymbol{C}_i = (C_1, \ldots, C_i)$ and $\boldsymbol{C} = \boldsymbol{C}_{t+1}$. Additionally, for $k \in \mathbb{N}$, we define

$$\mathcal{G}_k = \{\mathbf{c} \in (\mathcal{C} \cup \{\bot\})^{t+1} : \Pr(\boldsymbol{C} = \mathbf{c}) > 0 \wedge \mathbf{c}_{k-1} \notin \Gamma \wedge \mathbf{c}_k \in \Gamma\}$$

and then set $\mathcal{G} = \mathcal{G}_1 \cup \cdots \cup \mathcal{G}_{t+1}$. Note that, $\mathcal{G}_k$ contains those challenge sequences $\mathbf{c}$ which occur during a *successful* run of $\mathcal{E}$ and for which $\mathbf{c}_k \in \Gamma$ for the first time. Moreover, for all $\mathbf{c} \in \mathcal{G}_k$, it holds that $c_i \neq \bot$ for all $i \leq k$ and $c_i = \bot$ for all $i > k$. Further, observe that $\mathcal{G}$ is a disjoint union, and that the extractor succeeds if and only if $\boldsymbol{C} \in \mathcal{G}$.

We make the following case distinction. If $T := \{c \in \mathcal{C} : \Pr\bigl(\mathcal{V}(c, \mathcal{A}(c)) = 1\bigr) > 0\} \notin \Gamma$ then

$$0 = \Pr(V \mid C \notin T) \geq \Pr(V) - \Pr(C \in T) \geq \epsilon^{\mathcal{V}}(\mathcal{A}) - \kappa_\Gamma \,,$$

i.e., $\epsilon^{\mathcal{V}}(\mathcal{A}) \leq \kappa_\Gamma$. So, in this case, the extractor trivially succeeds with the claimed probability. In the remainder, we thus consider the case where $T \in \Gamma$, which in particular implies that for all i and all $\mathbf{c}_i \in \mathcal{C}^i$ with $\Pr\bigl(\boldsymbol{C}_i = \mathbf{c}_i\bigr) > 0$ and $\mathbf{c}_i \notin \Gamma$ it holds that $\Pr\bigl(C_{i+1} \neq \bot \mid \boldsymbol{C}_i = \mathbf{c}_i\bigr) > 0$, and so conditioning on the event "$C_{i+1} \neq \bot \wedge \boldsymbol{C}_i = \mathbf{c}_i$" is well defined then.

Towards analyzing the success probability $\Pr(\boldsymbol{C} \in \mathcal{G})$ of the extractor $\mathcal{E}$ in Fig. 1, it will be convenient to consider the variant $\mathcal{E}'$ of $\mathcal{E}$ that is not stopped by the coin, but keeps on searching for a next challenge $c_{i+1} \in \mathcal{U}_\Gamma(\mathbf{c}_i)$ for which $\mathcal{A}$ succeeds. Formally, for $i \in \{1, \ldots, t\}$ and $\mathbf{c}_i \in \mathcal{C}^i$ with $\Pr\bigl(\boldsymbol{C}_i = \mathbf{c}_i\bigr) > 0$ and $\mathbf{c}_i \notin \Gamma$, $\mathcal{E}'$ finds $c_{i+1} \in \mathcal{C}$ with probability

$$\tau(c_{i+1} \mid \mathbf{c}_i) := \Pr\bigl(C_{i+1} = c_{i+1} \mid C_{i+1} \neq \bot \wedge \boldsymbol{C}_i = \mathbf{c}_i\bigr) \,.$$

In all other cases, we set $\tau(c_{i+1} \mid \mathbf{c}_i) := 0$ for $\mathbf{c}_i \in (\mathcal{C} \cup \{\bot\})^i$ and $c_{i+1} \in \mathcal{C}$, and $\tau(\bot \mid \mathbf{c}_i) := 1$ instead, to indicate that c_{i+1} is set to $\bot$ then by $\mathcal{E}'$. Thus, by construction, $\mathcal{E}'$ keeps finding useful challenges $c_1, c_2, \ldots \in \mathcal{C}$ until it has collected $\mathbf{c}_i \in \Gamma$ which happens at the latest for $i = t+1$. Thus, the distribution

$$D(\mathbf{c}) := \prod_{i=0}^{t} \tau(c_{i+1} \mid \mathbf{c}_i)$$

of the list $\mathbf{c}$ of challenges produced by $\mathcal{E}'$ has its support contained in $\mathcal{G}$.

Below, we show how we can relate the probability of finding a certain $\mathbf{c}$ among the two extractors.

Let us now go back to analyzing the extractor. We observe that, for $\mathbf{c} \in \mathcal{G}_k$ and $i < k$, i.e., $\mathbf{c}_i \notin \Gamma$, conditioned on $[\boldsymbol{C}_i = \mathbf{c}_i]$ the extractor tries to find an $(i+1)$-th $c_{i+1} \in \mathcal{U}_\Gamma(\mathbf{c}_i)$ with $\mathcal{V}\big(c_{i+1}, \mathcal{A}(c_{i+1})\big) = 1$. To this end, it starts running two geometric experiments in parallel until either of them finishes. The first geometric experiment repeatedly runs $y_{i+1} \leftarrow \mathcal{A}(c_{i+1})$ for $c_{i+1} \in \mathcal{U}_\Gamma(\mathbf{c}_i)$ sampled uniformly at random, and thus has parameter $\delta(\mathbf{c}_i)$. The second geometric experiment repeatedly runs $y \leftarrow \mathcal{A}(d)$ for $d \leftarrow_R \mathcal{U}_\Gamma(\mathbf{c}_{i-1})$ until $d \notin \mathcal{U}_\Gamma(\mathbf{c}_i)$ and $\mathcal{V}(d, y) = 1$, i.e., it has parameter $\Delta(\mathbf{c}_i)$. The extractor succeeds in finding the $(i+1)$-th challenge if the first geometric experiment finishes before the second. Hence, by Lemma 1 and Equation (9),

$$\Pr\big(C_{i+1} \neq \bot \mid \boldsymbol{C}_i = \mathbf{c}_i\big) = \frac{\delta(\mathbf{c}_i)}{\delta(\mathbf{c}_i) + \Delta(\mathbf{c}_i) - \Delta(\mathbf{c}_i)\delta(\mathbf{c}_i)} \geq \frac{\tilde{\delta}(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})}\,,$$

and (noting that $c_{i+1} \neq \bot$ by definition of $\mathcal{G}$)

$$\begin{aligned}\Pr\big(C_{i+1} = c_{i+1} \mid \boldsymbol{C}_i = \mathbf{c}_i\big) &= \Pr\big(C_{i+1} = c_{i+1} \wedge C_{i+1} \neq \bot \mid \boldsymbol{C}_i = \mathbf{c}_i\big) \\ &= \tau(c_{i+1} \mid \mathbf{c}_i) \cdot \Pr\big(C_{i+1} \neq \bot \mid \boldsymbol{C}_i = \mathbf{c}_i\big) \\ &\geq \tau(c_{i+1} \mid \mathbf{c}_i) \cdot \frac{\tilde{\delta}(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})}\,.\end{aligned}$$

It thus follows that for k such that $\mathbf{c} \in \mathcal{G}_k$

$$\begin{aligned}\Pr(\boldsymbol{C} = \mathbf{c}) = \Pr(\boldsymbol{C}_k = \mathbf{c}_k) &= \prod_{i=1}^{k} \Pr(\boldsymbol{C}_i = \mathbf{c}_i \mid \boldsymbol{C}_{i-1} = \mathbf{c}_{i-1}) \\ &\geq \delta(\mathbf{c}_0)\tau(c_1 \mid \mathbf{c}_0) \prod_{i=1}^{k-1} \frac{\tilde{\delta}(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} \tau(c_{i+1} \mid \mathbf{c}_i) \qquad (19) \\ &\geq \frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - \kappa_\Gamma}{1 - \kappa_\Gamma} \prod_{i=1}^{k} \tau(c_i \mid \mathbf{c}_{i-1})\,,\end{aligned}$$

where the final inequality follows from Lemma 11. Hence, for $\mathbf{c} \in \mathcal{G}_k$, exploiting $\tau(c_i \mid \mathbf{c}_{i-1}) = \tau(\bot \mid \mathbf{c}_{i-1}) = 1$ for $i > k$, we find

$$\Pr(\boldsymbol{C} = \mathbf{c}) \geq \frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - \kappa_\Gamma}{1 - \kappa_\Gamma} \prod_{i=1}^{t+1} \tau(c_i \mid \mathbf{c}_{i-1}) = \frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - \kappa_\Gamma}{1 - \kappa_\Gamma} \cdot D(\mathbf{c})\,.$$

Thus, the extractor's success probability equals

$$\Pr(\boldsymbol{C} \in \mathcal{G}) = \sum_{\mathbf{c} \in \mathcal{G}} \Pr(\boldsymbol{C} = \mathbf{c}) \geq \frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - \kappa_\Gamma}{1 - \kappa_\Gamma} \cdot \sum_{\mathbf{c} \in \mathcal{G}} D(\mathbf{c}) = \frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - \kappa_\Gamma}{1 - \kappa_\Gamma}$$

where the final equality follows from the support of D being contained in $\mathcal{G}$. This proves the claimed success probability.

Expected Number of $\mathcal{A}$-Queries. Let us now continue with the expected running time analysis. For $1 \leq i < t$, let $\mathbf{c}_i \in \mathcal{C}^i$ with $\mathbf{c}_i \notin \Gamma$ and $\Pr[\boldsymbol{C}_i = \mathbf{c}_i] > 0$. As before, we use that in its $(i+1)$-th iteration, conditioned on $[\boldsymbol{C}_i = \mathbf{c}_i]$, the extractor runs two geometric experiments in parallel with parameters $\delta(\mathbf{c}_i)$ and $\Delta(\mathbf{c}_i)$, trying to find the $(i+1)$-th challenge. The probability that one of the experiments finishes in a single trial equals

$$1 - \big(1 - \delta(\mathbf{c}_i)\big)\big(1 - \Delta(\mathbf{c}_i)\big) = \delta(\mathbf{c}_i) + \Delta(\mathbf{c}_i) - \Delta(\mathbf{c}_i)\delta(\mathbf{c}_i) \geq \delta(\mathbf{c}_{i-1})\,,$$

where the inequality follows from Equation (8).

Now note that, to determine the value of the coin, the extractor only needs to invoke $\mathcal{A}$ if $d \notin \mathcal{U}_\Gamma(\mathbf{c}_i)$. Hence, in expectation each trial invokes $\mathcal{A}$ at most

$$1 + \Pr\big(C \notin \mathcal{U}_\Gamma(\mathbf{c}_i) \mid C \in \mathcal{U}_\Gamma(\mathbf{c}_{i-1})\big) \leq 1 + \Pr\big(C \notin \mathcal{U}_\Gamma(\mathbf{c}_i)\big) \leq 1 + \kappa_\Gamma$$

times, where the first inequality follows since $\mathcal{U}_\Gamma(\mathbf{c}_i) \subseteq \mathcal{U}_\Gamma(\mathbf{c}_{i-1})$, and the second since $\mathcal{C} \setminus \mathcal{U}_\Gamma(\mathbf{c}_i) \notin \Gamma$ for all $\mathbf{c}_i \notin \Gamma$ (Equation (3)). Hence, if we let Y_i denote the number of $\mathcal{A}$-queries that the extractor requires in its i-th iteration, i.e., when trying to find the i-th challenge C_i, then (by Lemma 2)

$$\mathbb{E}[Y_1] = 1 \quad \text{and} \quad \mathbb{E}[Y_{i+1} \mid \boldsymbol{C}_i = \mathbf{c}_i] \leq \frac{1+\kappa_\Gamma}{\delta(\mathbf{c}_{i-1})}\,.$$

On the other hand, if $\mathbf{c}_i$ is such that $\mathbf{c}_i \in \Gamma$ or $c_i = \bot$ (in either case, the extractor is done) then $\mathbb{E}[Y_{i+1} \mid \boldsymbol{C}_i = \mathbf{c}_i] = 0$. Putting these observations together shows that, for all $1 \leq k \leq t$,

$$\mathbb{E}[Y_{k+1}] = \sum_{\mathbf{c}_k \in T_k} \Pr(\boldsymbol{C}_k = \mathbf{c}_k) \cdot \mathbb{E}[Y_{k+1} \mid \boldsymbol{C}_k = \mathbf{c}_k] \leq (1+\kappa_\Gamma) \cdot \sum_{\mathbf{c}_k \in T_k} \frac{\Pr(\boldsymbol{C}_k = \mathbf{c}_k)}{\delta(\mathbf{c}_{k-1})}\,,$$

where $T_k := \{\mathbf{c}_k \in \mathcal{C}^k : \mathbf{c}_k \notin \Gamma \wedge \Pr(\boldsymbol{C}_k = \mathbf{c}_k) > 0\}$.

As in the success probability analysis, we will now expand the probability $\Pr(\boldsymbol{C}_k = \mathbf{c}_k)$, but now aiming for an upper bound. Then, for $\mathbf{c}_k \in T_k$ and $i < k$, by Equation (8)

$$\Pr\big(C_{i+1} \neq \bot \mid \boldsymbol{C}_i = \mathbf{c}_i\big) = \frac{\delta(\mathbf{c}_i)}{\delta(\mathbf{c}_i) + \Delta(\mathbf{c}_i) - \Delta(\mathbf{c}_i)\delta(\mathbf{c}_i)} \leq \frac{\delta(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})}\,,$$

and (noting that $c_{i+1} \neq \bot$ by definition of $\mathcal{G}$)

$$\begin{aligned}\Pr\big(C_{i+1} = c_{i+1} \mid \boldsymbol{C}_i = \mathbf{c}_i\big) &= \Pr\big(C_{i+1} = c_{i+1} \wedge C_{i+1} \neq \bot \mid \boldsymbol{C}_i = \mathbf{c}_i\big) \\ &= \tau(c_{i+1} \mid \mathbf{c}_i) \cdot \Pr\big(C_{i+1} \neq \bot \mid \boldsymbol{C}_i = \mathbf{c}_i\big) \\ &\leq \tau(c_{i+1} \mid \mathbf{c}_i) \cdot \frac{\delta(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})}\,.\end{aligned}$$

where, as before,

$$\tau(c_{i+1} \mid \mathbf{c}_i) := \Pr\big(C_{i+1} = c_{i+1} \mid C_{i+1} \neq \bot \wedge \boldsymbol{C}_i = \mathbf{c}_i\big)\,.$$

Similar to Equation (19), but now using the upper bound, it therefore follows that

$$\begin{aligned}\Pr(\boldsymbol{C}_k = \mathbf{c}_k) &= \prod_{i=1}^{k} \Pr(C_i = c_i \mid \boldsymbol{C}_{i-1} = \mathbf{c}_{i-1}) \\ &\leq \delta(\mathbf{c}_0) \cdot \tau(c_1 \mid \mathbf{c}_0) \cdot \prod_{i=1}^{k-1} \frac{\delta(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} \cdot \tau(c_{i+1} \mid \mathbf{c}_i) \\ &= \delta(\mathbf{c}_{k-1}) \cdot \prod_{i=1}^{k} \tau(c_i \mid \mathbf{c}_{i-1})\,.\end{aligned}$$

Hence, for all $k \geq 1$,

$$\begin{aligned}\mathbb{E}[Y_{k+1}] &\leq (1+\kappa_\Gamma) \cdot \sum_{\mathbf{c}_k \in T_k} \prod_{i=1}^{k} \tau(c_i \mid \mathbf{c}_{i-1}) \\ &\leq (1+\kappa_\Gamma) \cdot \sum_{c_1 \in \mathcal{C}} \tau(c_1 \mid \mathbf{c}_0) \sum_{c_2 \in \mathcal{C}} \tau(c_2 \mid \mathbf{c}_1) \cdots \sum_{c_k \in \mathcal{C}} \tau(c_k \mid \mathbf{c}_{k-1}) \\ &\leq (1+\kappa_\Gamma)\,,\end{aligned}$$

where the ill-defined terms $\tau(c_i \mid \mathbf{c}_{i-1})$, i.e., with $\Pr(\boldsymbol{C}_{i-1} = \mathbf{c}_{i-1}) = 0$, are simply defined to be 0.

Hence, the expected running time of the extractor is at most

$$1 + \sum_{k=1}^{t} \mathbb{E}[Y_{k+1}] \leq 1 + t \cdot (1+\kappa_\Gamma)\,,$$

which completes the proof of the lemma.

□