J Cryptol (2026) 39:19
https://doi.org/10.1007/s00145-025-09569-y

Journal of CRYPTOLOGY

Research Article

The Fiat—Shamir Transformation of $(\Gamma_1, \ldots, \Gamma_\mu)$-Special-Sound Interactive Proofs*

Thomas Attema
CWI, Cryptology Group, Amsterdam, The Netherlands
TNO, Applied Cryptography and Quantum Applications, The Hague, The Netherlands
thomas.attema@tno.nl

Serge Fehr
CWI, Cryptology Group, Amsterdam, The Netherlands
Mathematical Institute, Leiden University, Leiden, The Netherlands
serge.fehr@cwi.nl

Michael Klooß
Karlsruhe Institute of Technology, Karlsruhe, Germany
michael.klooss@kit.edu

Nicolas Resch
Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands
n.a.resch@uva.nl

Communicated by Stefano Tessaro

Received 20 November 2024 / Revised 30 October 2025 / Accepted 15 December 2025

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 Σ-protocols, which is a natural multi-round generalization of the well-known class of 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 $(\Gamma_1, \ldots, \Gamma_\mu)$-special-sound protocols. This notion was recently defined and analyzed in the interactive case; it captures a larger class of protocols, namely where the special-soundness property is characterized by a general access structure, rather than a threshold. We show in this work that this is indeed the case. Concretely, we show that the Fiat–Shamir transformation of any $(\Gamma_1, \ldots, \Gamma_\mu)$-special-sound interactive proof is knowledge sound under the same condition on $\Gamma_1, \ldots, \Gamma_\mu$ for which the

*Michael Klooß: Work done at Aalto University, Espoo, Finland.
This paper was reviewed by Eylon Yogev and an anonymous reviewer.

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. Thus, new techniques are necessary to handle the above complications.

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. Further desiderata may include not revealing any additional information (*zero-knowledge*), or the proof being (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 must x be in L, the prover actually needs to know a witness w attesting to $x \in L$ (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*: an efficient (i.e., expected poly-time) algorithm that is given rewindable oracle access to a (possibly dishonest) prover $\mathcal{P}^*$, and it outputs a witness w for x with a probability that is closely related to the probability of $\mathcal{P}^*$ convincing the verifier $\mathcal{V}$. In certain cases, ordinary soundness is meaningless; 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 non-trivial requirement is that the prover *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 recently proposed protocols, such as Bullet-

proofs [11,14]. Motivated by this, the original result on special-sound Σ-protocols was extended in [2,4,10] to k-special-sound Σ-protocols and to their multi-round variants, $(k_1, \ldots, k_\mu)$-special-sound protocols. However, many modern interactive proofs [3,13, 33]—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 [7], 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, [7] 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, [7] identifies two relevant parameters $\kappa_{\mathbf{\Gamma}}$ and $t_{\mathbf{\Gamma}}$, determined 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. While precise definitions are provided later, we now informally define these parameters. The parameter $\kappa_{\mathbf{\Gamma}}$ captures the trivial cheating probability, in that in a typical $\mathbf{\Gamma}$-special-sound interactive proof the prover can cheat by making a guess on the challenge, and $\kappa_{\mathbf{\Gamma}}$ is then precisely the probability that in at least one of the rounds the prover guessed (sufficiently) correctly. The parameter $t_{\mathbf{\Gamma}}$ on the other hand is related to the worst case number of challenges that one may encounter until the set lies in the access structure, when avoiding challenges that are "useless". Whether $t_{\mathbf{\Gamma}}$ is polynomial or not depends on $\mathbf{\Gamma}$; we discuss some examples later.

1.2. *The Fiat–Shamir Transformation*

The *Fiat–Shamir transformation* is a powerful technique for turning (public-coin) *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 many of those have a non-constant number of rounds; the Fiat–Shamir transformation is then often used to avoid the increased round complexity (which would then actually form the efficiency bottleneck) 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 in 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 recently, 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 [5] (see [6] for the full version). 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.

1.3. *On the Relevance of* $(\Gamma_1, \ldots, \Gamma_\mu)$*-Special-Soundness*

Unfortunately, while the notion of $(k_1, \ldots, k_\mu)$-special-soundness covers certain interactive proofs with succinct communication, e.g., Bulletproofs, there are many interesting examples that fall outside this class. To highlight the relevance of the more general $(\Gamma_1, \ldots, \Gamma_\mu)$-special-soundness notion, we analyze three examples in detail. To demonstrate the utility of this notion, it suffices to focus on 3-round interactive protocols, i.e., Σ-protocols. Accordingly, all three examples are simple Σ-protocols, although they are often used as building blocks in more complex multi-round protocols.

1.3.1. *Amortization Technique for Proving Knowledge of* $\mathbb{F}$*-Linear Map Preimages*

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., [9,12,25,30]. Unfortunately, it renders the corresponding threshold special-soundness parameter k too large to provide reasonable security guarantees. However, the amortization technique is Γ-special-sound for a Γ with small parameters.

In more detail, consider the following simple interactive proof for proving knowledge of preimages of n given values $y_1, \ldots, y_n$ under a $\mathbb{F}$-linear map, with $\mathbb{F}$ being a finite field: the prover simply announces a preimage x of $y = \sum_i c_i y_i$, where $c = (c_1, \ldots, c_n) \in \mathbb{F}^n$ is chosen at random by the verifier. [1] Regarding special-soundness, in the worst case, the preimages of $y = \sum_i c_i y_i$ for $|\mathbb{F}|^{n-1}$ choices of $c \in \mathbb{F}^n$ are not sufficient to recover preimages of $y_1, \ldots, y_n$; namely, when the c's do not span the entire space $\mathbb{F}^n$. Hence, this interactive proof is k-special-sound with $k = |\mathbb{F}|^{n-1} + 1$, i.e., the threshold special-soundness parameter k is typically exponentially large rendering generic extractors inefficient. [2] On the other hand, the preimages can be recovered as soon as the challenges c span the entire space. Thus, this protocol is also Γ-special-sound with Γ consisting of the sets of c's that span $\mathbb{F}^n$.

As for the parameters of interest, κ_Γ is defined as the maximum fraction of the challenge space that does not yet allow extraction, which here is

$$\frac{\max\{|S| : S \subseteq \mathbb{F}^n \text{ s.t. } S \text{ does not span } \mathbb{F}^n\}}{|\mathbb{F}^n|} = \frac{|\mathbb{F}|^{n-1}}{|\mathbb{F}|^n} = \frac{1}{|\mathbb{F}|},$$

[1] Typically, the prover would prove knowledge of x instead of announcing it, but we keep it simple here.

[2] The exponential parameter k can be avoided by defining the challenges as $c_i = \gamma^{i-1}$ for $1 \le i \le n$ and uniform $\gamma \in \mathbb{F}$, see for example [26]. This modified amortization is widely used and has $k = n$ and knowledge error $\frac{n-1}{|\mathbb{F}|} \ge \frac{1}{|\mathbb{F}|}$. However, this approach is not always applicable, e.g., it requires $|\mathcal{C}| \ge n$, which is not always the case. For instance, in lattice-based proof systems the challenges c are typically "short", i.e., $c \in [-C, C]$ for $C \ll |\mathbb{F}|$. For instance, [12] uses binary challenges c.

which matches the trivial cheating probability. The parameter t_Γ, on the other hand, is the worst case number of challenges that one may encounter until the set lies in Γ, when avoiding challenges that are "useless". Here, a challenge is useless when it is in the span of the previous challenges, and so when avoiding those we get a set of challenges that span $\mathbb{F}^n$ after n choices, so $t_\Gamma = n$.

1.3.2. *Proving Knowledge of Merkle Tree Commitment Opening*

This Γ-soundness notion leads to an effective analysis of a broader class of proof systems. For example, 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 [15,28,31,32]. By construction, an interactive proof obtained via this recipe is not $(k_1, \ldots, k_\mu)$-special-sound, at least not for reasonable parameters k_i, whereas it *can* be cast as a $(\Gamma_1, \ldots, \Gamma_\mu)$-special-sound protocol with reasonable parameters.[3]

More concretely, consider the following simple protocol for proving knowledge of the opening of a Merkle tree commitment. Suppose the Merkle tree has n leaves, and the verifier randomly selects ℓ leaves for the prover to open. The total number of possible challenges is therefore $\binom{n}{\ell}$. A dishonest prover might be able to open all but one leaf of the Merkle tree, in which case it can successfully respond to $\binom{n-1}{\ell}$ challenges. In other words, the threshold special-soundness parameter equals $\binom{n-1}{\ell}$.

On the other hand, this interactive proof is also Γ-special-sound, where Γ is the collection of challenge sets that cover all n leaves. The parameter κ_Γ is defined as the maximum fraction of the challenge space for which extraction is not yet possible. In this setting, it is given by

$$\frac{\binom{n-1}{\ell}}{\binom{n}{\ell}} = 1 - \frac{\ell}{n}.$$

Finally, the parameter t_Γ denotes the worst-case number of "useful" challenges required until the set of opened leaves belongs to Γ. Here, a challenge is considered useful if it opens at least one new leaf not previously revealed. The first challenge opens ℓ new leaves, and each subsequent useful challenge opens at least one additional new leaf. Thus, we have $t_\Gamma = n - \ell + 1$.

1.3.3. *Parallel Repetition*

As a final example, consider the t-fold parallel repetition Π^t of a k-special-sound Σ-protocol Π. It is easy to see that Π^t is $((k-1)^t + 1)$-special-sound, i.e., the threshold special-soundness parameter grows exponentially in t. Consequently, this property alone does not imply knowledge soundness.

[3] 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 $(\Gamma_1, \ldots, \Gamma_\mu)$-special-soundness, together with its knowledge extractors, provides an alternative for analyzing their knowledge soundness.

Table 1. The parameters κ_Γ and t_Γ for the access structures in the considered example Σ-Protocols. Additionally, the threshold-special-soundness is provided..

Γ-Special-sound interactive protocol	κ_Γ	t_Γ	Threshold special-soundness parameter
Amortization technique for proving knowledge of n $\mathbb{F}$-Linear map preimages	$\frac{1}{\lvert\mathbb{F}\rvert}$	n	$\lvert\mathbb{F}\rvert^{n-1}+1$
Proving knowledge of Merkle tree commitment opening	$1-\frac{\ell}{n}$	$n-\ell+1$	$\binom{n-1}{\ell}+1 \geq \left(\frac{n-1}{\ell}\right)^{\ell}+1$
t-Fold parallel repetition of k-special-sound Σ-protocol	$\left(\frac{k-1}{\lvert\mathcal{C}\rvert}\right)^{t}$	$(k-1)\cdot t+1$	$(k-1)^t+1$

However, Π^t is also Γ-special-sound, where Γ consists of the sets of challenge vectors for which there exists at least one position such that we encounter at least k distinct entries in this position across the challenge vectors. The maximum size of a set of challenges outside Γ is $(k-1)^t$, so that

$$\kappa_\Gamma = \left(\frac{k-1}{\lvert\mathcal{C}\rvert}\right)^{t}.$$

Moreover, a new challenge is considered useful if it introduces at least one previously unseen challenge in one of the t coordinates, i.e., $t_\Gamma = (k-1)\cdot t+1$. Hence, in contrast to the threshold special-soundness parameter, t_Γ grows only linearly in t, and thus the Γ-special-soundness property immediately implies knowledge soundness.

Overall, the table below summarizes the different parameters for the above examples. Recall that the extractor's running time for k-special-sound interactive proofs scales linearly in the threshold parameter k, whereas it scales linearly in t_Γ for Γ-special-sound interactive proofs. Therefore, the difference between the final two columns of the table highlights the advantage of the Γ-special-soundness notion (Table 1).

1.4. *Our Results*

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 [5] on the Fiat–Shamir transformation of $(k_1,\ldots,k_\mu)$-special-sound protocols extend to this general notion of $(\Gamma_1,\ldots,\Gamma_\mu)$-special-soundness introduced in [7].

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

Indeed, in this paper, we show that the results of [5], which show the security of the Fiat–Shamir transform of multi-round $(k_1,\ldots,k_\mu)$-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 [7]. In more detail, we show the following.

Theorem 1. *(Informal; see Theorem 5 for a precise statement) The Fiat–Shamir transformation* $\mathsf{FS}[\Pi]$ *of any* $(\Gamma_1, \ldots, \Gamma_\mu)$*-special-sound interactive proof* Π*, where* $\mathbf{\Gamma} := (\Gamma_1, \ldots, \Gamma_\mu)$ *is such that the parameter* $t_{\mathbf{\Gamma}}$ *is polynomial and "useful challenges are efficiently sampleable", 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* Π*.*

We stress the independence of the bound on μ, the number of challenge rounds of the protocol.

As we discuss below in more detail, a crucial ingredient in our extractor construction for the Fiat–Shamir transformation is a new, improved extractor for the *interactive* case, which is then used as a subroutine in our Fiat–Shamir extractor. See Table 2 and Table 3 for the characteristics of the respective extractors (for the interactive and the Fiat–Shamir transformation case).

To illustrate Theorem 1, let us return to the example of batch verification of n linear claims, as discussed in Sect. 1.3. Recall $\kappa_\Gamma = \frac{1}{|\mathbb{F}|}$ and that $t_\Gamma = n$. In particular, as t_Γ is polynomial in n Theorem 1 applies, and we find that the FS-transform is knowledge sound with knowledge error $\frac{Q+1}{\mathbb{F}}$. Recalling that in a single instantiation a prover can cheat with probability $\frac{1}{|\mathbb{F}|}$, and so with Q oracle queries (and an additional attempt by outputting a forgery without querying the oracle) one could cheat with probability $1-(1-1/|\mathbb{F}|)^{Q+1} \approx \frac{Q+1}{|\mathbb{F}|}$, our bound is essentially tight.

1.5. *Technical Overview*

One might expect that our result can be obtained as a direct generalization of the reasoning of [5] to the notion of $(\Gamma_1, \ldots, \Gamma_\mu)$-special-soundness. However, the extractor considered in [5] is based on sampling *without replacement* and properties of the *negative hypergeometric distribution* are used for the analysis; 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.

Our new Fiat–Shamir extractor. As a first step towards proving the claimed knowledge soundness of the Fiat–Shamir transformation, we show that for the 3-round case (i.e., for Σ-protocols), we can actually reduce the analysis of the considered extractor for the non-interactive Fiat–Shamir transformed protocol $\mathsf{FS}[\Pi]$ to that of (some variant of) the underlying interactive proof Π. Simply relying on the extractors proposed and analyzed in [4,7] for the interactive proof Π, would then settle the 3-round case, showing that the Fiat–Shamir transformation of a Γ-special-sound Σ-protocol is a proof of knowledge with knowledge soundness κ_Γ if the parameter t_Γ is polynomial.

Unfortunately, doing a similar reduction from $\mathsf{FS}[\Pi]$ to Π does not work for the multi-round case, as we explain in Sect. 6.2. For this reason, our approach for the multi-round case is to follow a similar recursive strategy as in [5], 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, and a refined analysis of the 3-round case, to achieve the required round-independent expected running time.

However, following this approach but using the above 3-round extractor that relies on the extractors from [4,7] has the problem that the success probability of the 3-round extractor suffers a factor t_Γ-loss (see Table 2), compared to the 3-round extractor of [5]. 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 of the extractor's success probability, the t_Γ-loss negatively impacts the knowledge error. More precisely, this approach would introduce a factor (roughly) $t_{\Gamma_2,\ldots,\Gamma_\mu} = t_{\Gamma_2} \cdots t_{\Gamma_\mu}$ blow-up in the knowledge error. Since $t_{\Gamma_2,\ldots,\Gamma_\mu} \leq t_{\mathbf{\Gamma}}$ and the latter is required to be polynomial, this would be sufficient for proving knowledge soundness. However, we aim to derive an *optimal* knowledge error that is equal to the trivial cheating probability of a dishonest prover, and therefore, we wish to avoid this loss.

An improved interactive extractor Our solution is to go back to square one and not use the extractor from [4,7] for the 3-round interactive case, but instead construct a new extractor. Our new extractor improves over the ones proposed in [4,7] in that it avoids the factor t_Γ loss. Roughly speaking, this is achieved by avoiding the recursive construction from [4,7] — which offers a rather simple analysis but is (quite obviously) suboptimal — and instead following a more straightforward, greedy strategy, together with a delicate choice of stopping criterion, which though necessitates a more complex analysis.

Table 2 provides an overview of the (expected) running times and success probabilities of the different extractors for special-sound interactive proofs. Note that, in this work we only provide the extractor for Γ-special-sound Σ-protocols, i.e., 3-round interactive proofs (see Theorem 2 and Lemma 6). Indeed, our main goal is to analyze the Fiat–Shamir transformation of multi-round interactive proofs, and to achieve this goal we do not need an extractor for multi-round *interactive* proofs. However, for a meaningful comparison with prior works, Table 2 includes the generalized multi-round version of our extractor, which can be obtained by leveraging the recursive techniques from [7]. The two different upper bounds on our expected running time reflect both a standard and a refined analysis, the latter of which is crucial for analyzing the Fiat–Shamir transformation.

In order to be suitable for our actual goal, analyzing Fiat–Shamir transformations of multi-round interactive proofs, we additionally need a refined running time analysis. More precisely, similar to the Fiat–Shamir analysis of [5], we need to be able to consider attackers that are less or more "costly" dependent on the challenge. Using this extractor and the refined running time analysis as the base case, we are then ready to work out the recursive extractor construction and the recursive analysis to argue the multi-round case.

Putting it together Altogether, this then leads to our main result: the Fiat–Shamir transformation of a $(\Gamma_1, \ldots, \Gamma_\mu)$-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. [4] Table 3 compares our new extractor with the extractor of [5] for the Fiat–Shamir transformation of $(k_1, \ldots, k_\mu)$-special-sound interactive proofs.

[4] Our main result (Theorem 5) is expressed in terms of $T_{\mathbf{\Gamma}}$ instead of $t_{\mathbf{\Gamma}}$, see also Remark 6. Obviously, $t_{\mathbf{\Gamma}}$ is polynomial if and only if $T_{\mathbf{\Gamma}}$ is (for non-trivial $t_{\Gamma_i} > 1$).

Table 2. Different knowledge extractors for $(\Gamma_1, \ldots, \Gamma_\mu)$-special-sound *interactive* proofs. Here, $\epsilon = \epsilon(\mathcal{A})$ denotes the success probability of the adversary $\mathcal{A}$, $\kappa_{\mathbf{\Gamma}}$ is the knowledge error, and $t_{\mathbf{\Gamma}} = \prod_{i=1}^{\mu} t_{\Gamma_i}$ and $T_{\mathbf{\Gamma}} = \prod_{i=1}^{\mu}(t_{\Gamma_i} + 1)$, where t_{Γ_i} depends on the access structure Γ_i (Definition 8).

Extractor	Running time (Expected number of $\mathcal{A}$-queries)	Success probability	Access structures Γ_i
[2]	$\leq t_{\mathbf{\Gamma}}$	$\geq \frac{\epsilon - \kappa_{\mathbf{\Gamma}}}{1 - \kappa_{t_{\mathbf{\Gamma}}}}$	Threshold
[4,7]	$\leq 2^{\mu} \cdot t_{\mathbf{\Gamma}}$	$\geq \frac{1}{t_{\mathbf{\Gamma}}} \frac{\epsilon - \kappa_{t_{\mathbf{\Gamma}}}}{1 - \kappa_{t_{\mathbf{\Gamma}}}}$	Arbitrary
This work	$\leq \min\left(2^{\mu} \cdot T_{\mathbf{\Gamma}}, \frac{T_{\mathbf{\Gamma}}}{1 - \kappa_{\mathbf{\Gamma}}}\right)$	$\geq \frac{\epsilon - \kappa_{t_{\mathbf{\Gamma}}}}{1 - \kappa_{t_{\mathbf{\Gamma}}}}$	Arbitrary

The slightly larger parameter $T_{\mathbf{\Gamma}}$ in the running time in our work is due to making the extractor (for the interactive proof) ready to be used as a building block in the Fiat–Shamir analysis. If we are interested in the interactive case only, we can replace $T_{\mathbf{\Gamma}}$ with $t_{\mathbf{\Gamma}}$, see Theorem 2

Table 3. Knowledge extractors for the Fiat–Shamir transformation of $(\Gamma_1, \ldots, \Gamma_\mu)$-special-sound *interactive* proofs. Here, $\epsilon = \epsilon(\mathcal{A})$ denotes the success probability of the adversary $\mathcal{A}$, Q the adversary's query complexity, $\kappa_{\mathbf{\Gamma}}$ the knowledge error, and $t_{\mathbf{\Gamma}} = \prod_{i=1}^{\mu} t_{\Gamma_i}$ and $T_{\mathbf{\Gamma}} = \prod_{i=1}^{\mu}(t_{\Gamma_i} + 1)$, where t_{Γ_i} depends on the access structure Γ_i (Definition 8). The final column shows whether the extractor applies to arbitrary access structures or only to threshold access structures.

Extractor	Running time (Expected number of $\mathcal{A}$-Queries)	Success probability	Access structures Γ_i
[5]	$\leq (Q+1) \cdot t_{\mathbf{\Gamma}}$	$\geq \frac{\epsilon - (Q+1)\kappa_{\mathbf{\Gamma}}}{1 - \kappa_{t_{\mathbf{\Gamma}}}}$	Threshold
This work (Theorem 5)	$\leq (Q+1) \cdot \frac{T_{\mathbf{\Gamma}}}{1 - \kappa_{\mathbf{\Gamma}}}$	$\geq \frac{\epsilon - (Q+1) \cdot \kappa_{t_{\mathbf{\Gamma}}}}{1 - \kappa_{t_{\mathbf{\Gamma}}}}$	Arbitrary

1.6. *Related Work*

We already discussed the related work [4,5,7], and their relevance and relation to this work, above.

In concurrent work [8], our extractor for Σ-protocols (Theorem 2) is extended to the newly introduced notion of adaptive special-soundness, which is used to improve soundness slack that arises specifically in lattice-based instantiations of multi-round interactive proofs.

Another closely related work is [35], 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 $(\Gamma_1, \ldots, \Gamma_\mu)$-special-soundness studied in [7], 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 [35, 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 [35] is establishing good tail bounds (instead of only expected query bounds) with a precise analysis. The presented bounds are "convoluted expressions" [35,

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 terminology and formalism used in [35], makes a comparison of [35] and [7] that goes beyond the high-level similarities challenging.

In [36], Wikström extends the terminology and results of [35] to the Fiat–Shamir transformation. This amplifies the difficulties in comparing our work (which extends [7]) with [36] (which extends [35]). We stress that [36, Sect. 7] claims similar result as ours, i.e., security loss linear in the number of queries; however, the connection between the actual proven result [36, Thm. 3] and the claimed implications on the knowledge soundness of the Fiat–Shamir transformation is implicit and non-obvious.

To expand a little: [36] chooses to rephrase the setting into "grafting protocols", which seems closely related to *state restoration (knowledge) soundness* [15,27]. In that language, the progress of a prover can be described using "grafted sequences", which at a high level correspond to queries along the proof to be extracted, and "shadow sequences" which are obtained from grafted sequences by ignoring non-proof-related queries. Intuitively, if one could directly sample shadow sequences, then the usual interactive extraction should work. However, only grafted sequences can be sampled directly. Thus [36] proceeds to provide a sampler for shadow sequences, using a concurrent but similar approach as [5]. Compared to our approach, the extractor in [36] (based on [35]) requires additional amplification, which results in a suboptimal knowledge error (which [35] already exhibits in the interactive case, hence this is of no concern in [36]).[5]

In summary, next to the above mentioned specific (and partly minor) differences, a major differentiation is the standard terminology that we use in our work to express and prove our results. The articles [35,36] on the other hand are phrased in terms of new abstraction and terminology, and the precise connection to the security of the Fiat–Shamir transformation is left implicit.

A very recent related work is [1], which introduces predicate special-soundness as an extension to $(k_1, \ldots, k_\mu)$-special-soundness. They reuse the extractor from [5] and extend its analysis. The notion of [1] allows to track certain failure events during an extraction, which occurs in advanced lattice-based proof systems. As such, predicate special-soundness and generalized special-soundness are rather orthogonal. It is a natural question whether both concepts could be combined using our new extractor.

In cases where the original interactive proof itself is modeled in some idealized setting, such as the random oracle model or the generic group model, rewinding can often be avoided in the extraction procedure by exploiting the features of the idealized setting. This then leads to straightline extraction. For example, this is typically the case for IOPs [15,34]. The security of the Fiat–Shamir transformation for multi-round protocols is the often shown via round-by-round soundness [15,17,27]. Very recently, a family of such proof systems has been studied using rewinding and a concrete security framework [19,20] that avoids the idealized setting for the interactive proofs.

Analyzing the Fiat–Shamir transformation in the quantum setting, i.e., when the prover is modeled as a quantum algorithm, is challenging, due to the fact that taking a snapshot of the prover's state (with the goal to rewind back to that state again later) is not possible, due to the no-cloning theorem. There has been progress in different directions: [23] reduced

[5] Had we based our work directly on the extractor from [7], we would have the same problem, cf. Sect. 1.5.

the quantum security of the Fiat–Shamir transformation to the quantum security of the underlying interactive proof (but with an exponential loss in the number of rounds for multi-round protocols), [29] showed that certain Bulletproofs-like *interactive* protocols are knowledge sound against quantum adversaries (for a suitably adjusted definition of knowledge soundness), [21,24] showed online extractability in the quantum random oracle model for the Fiat–Shamir transformation of certain protocols, etc.; however, many of the classical knowledge soundness results for (the Fiat–Shamir transformation of) interactive proofs do not have closely-matching quantum counterparts (yet).

2. Preliminaries

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 *knowledge 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 ro, called the *random oracle*, which is instantiated as a uniformly random function. Typically $\mathsf{ro} : \{0,1\}^* \to \{0,1\}^\eta$ for some $\eta \in \mathbb{N}$ related to the security parameter; however, we find it more convenient to consider an arbitrary finite codomain $\mathcal{C}$, and to limit the domain to a (sufficiently large) *finite*[6] message space $\mathcal{M}$, so that $\mathsf{ro} : \mathcal{M} \to \mathcal{C}$ (we will also write $\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}$).

[6] The restriction of the domain to be finite is because otherwise, strictly speaking, one cannot speak of a uniformly random such function (without specifying a σ-algebra), given that there are infinitely many.

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. 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 verifier 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 $x \in \{0, 1\}^*$,

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

where $\epsilon(\mathcal{P}^*, x) = \Pr\left(\mathcal{V}^{\mathsf{ro}}(x, \mathcal{P}^{*,\mathsf{ro}}) = 1\right)$. 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.

Remark 2. **(Adaptive Knowledge Soundness)** We note that Definition 3 above captures a *static* dishonest prover, which is given the instance x as input. A stronger notion considers an *adaptive* prover, which produces and outputs the instance x along with the non-interactive proof. For the latter, knowledge soundness is trickier to define though, since $\mathcal{P}^*$ may then output different instances in different runs, and so it is per-se not clear for which instance $\mathcal{E}$ should find a witness. A solution is to explicitly require that in a first step $\mathcal{E}$ must run $\mathcal{P}^*$ with a uniformly random choice of ro (implemented using lazy sampling), and if in this run $\mathcal{P}^*$ produces an instance x with a valid proof then $\mathcal{E}$ must find a witness for *that* x, else it must abort (see [6, Definition 10] for a more natural yet slightly weaker definition). It is a matter of inspection to see that all our results carry over to this adaptive variant. Indeed, by construction, our extractors run $\mathcal{P}^*$ with a uniformly random choice of ro as a first step, and abort if this run is not successful. Furthermore, by

considering the instance x, output by $\mathcal{P}^*$, to be part of the first message a_1 (see below for the meaning of a_1), it follows immediately that our analyses carry over to the adaptive variant. After all, after the initial run, our main challenge is to get $\mathcal{P}^*$ to output further proofs with *the same* a_1, and thus the same x then.

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 [6, Definition 10] where the malicious prover may first choose the input x and subsequently attempt to forge a false proof [16].

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 $\mathcal{C}_i$. However, our results also apply to other variants of the Fiat-Shamir transformation. Two widely used alternatives are discussed below. 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.

For completeness, we provide here the formal definition, where, 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 4. (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 $\mathsf{ro}_i : \mathcal{M}^i \to \mathcal{C}_i$ 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

$$c_i = \mathsf{ro}_i(a_1, \ldots, a_{i-1}, a_i) \tag{2}$$

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 (2).

Remark 3. **(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 (2) are instead computed as

$$c_i = \mathsf{ro}_i(x, a_1, \ldots, a_{i-1}, a_i) \tag{3}$$

for $i = 1, \ldots, \mu$; this is very much in line with treating x as part of the first message a_1 (Remark 2). Proving and verification proceed mutatis mutandis as before.

For concreteness, we consider this particular version of the Fiat–Shamir transformation, where the entire history is hashed. A widely used alternative to the above Fiat–Shamir transformation is a chained version that computes the i-th challenge as $c_i = \mathsf{ro}_i(c_{i-1}, a_i)$ (or $c_i = \mathsf{ro}_i(x, c_{i-1}, a_i)$ in the adaptive case). It is easy to see that an attack against the chained variant can be turned into an attack against the version we consider and analyze. The reduction only fails if the dishonest prover in the chained variant finds a collision in the random oracle, or succeeds in inverting the random oracle on a non-trivial point (i.e., a point that was not the reply to a random oracle query). Thus, security carries over (up to a small loss). In the reduction, one simply runs the original attack but extracts the "fitting" $a_1, \ldots, a_{i-1}$ from any query of the form (c_{i-1}, a_i) by inspecting former oracle queries.

In some applications of the Fiat–Shamir transformation it is important that the input to the hash computation for computing the challenges has sufficient entropy, and so a random salt may then be added to the hash input. Our results remain fully applicable to this salted variant as well.

2.4. *Geometric Distribution*

Finally, as in prior works [4,7], 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. *([7, 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 Bernoulli trial i is associated with a cost Z_i 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 $X \sim \mathrm{Geo}(p)$, and let $Z_1, Z_2, \ldots$ be arbitrary non-negative random variables, subject to the constraint that there exists $\theta \geq 0$ such that $\mathbb{E}[Z_i \mid X \geq i] \leq \theta$ for all i. Then, $Z = Z_1 + Z_2 + \cdots + Z_X$ satisfies*

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

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: Γ-Special-Sound Protocols

The standard concept of *special-soundness* for Σ-protocols [18,22] has recently seen many generalizations. Firstly, a Σ-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 Γ-special-soundness (and $(\Gamma_1, \ldots, \Gamma_\mu)$-special-soundness in the multi-round case) was defined and shown to imply knowledge soundness [7] 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 $(\Gamma_1, \ldots, \Gamma_\mu)$-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 5. *(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 6. *(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 $(a, c_1, z_1), \ldots, (a, c_k, z_k)$ with common first message a and such that $\{c_1, \ldots, c_k\} \in \Gamma$, 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 4. Technically, the monotone structure $(\Gamma, \mathcal{C})$ of Definition 6 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 [7] 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 $(a, c_1, z_1), \ldots, (a, c_k, z_k)$ with $\{c_1, \ldots, c_k\} \in \Gamma$.

Definition 7. *(Useful Elements)* For a monotone structure $(\Gamma, \mathcal{C})$ we define the following function:

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

It is easily seen that

$$\mathcal{U}_\Gamma(B) \subseteq \mathcal{U}_\Gamma(A) \quad \text{for all } A \subseteq B\,. \tag{4}$$

Further, [7, Lemma 3] shows that

$$\mathcal{C} \setminus \mathcal{U}_\Gamma(S) \notin \Gamma \quad \text{for all } S \notin \Gamma\,. \tag{5}$$

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

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

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. *([7, 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 Γ-Special-Sound Σ-Protocols*

For any Γ-special-sound Σ-protocol Π, Theorem 1 of [7] proves the existence of a knowledge extractor that makes an expected number of at most $2t_\Gamma - 1$ queries to the considered prover and successfully extracts a witness with probability at least

$$\frac{1}{t_\Gamma} \cdot \frac{\epsilon - \kappa_\Gamma}{1 - \kappa_\Gamma}\,,$$

where ϵ is the success probability of the considered prover and

$$\kappa_\Gamma := \max_{S \notin \Gamma} \frac{|S|}{|\mathcal{C}|}\,. \tag{6}$$

Looking ahead, we will reduce the existence of a knowledge extractor for the Fiat–Shamir transformation of a Γ-special-sound Σ-protocol to the existence of a knowledge extractor for the original protocol Π, which is provided by Theorem 1 of [7] (if t_Γ 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_Γ 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 first construct a new and improved version of the above extractor, which avoids the t_Γ in the denominator.

4. An Improved Extractor for Γ-Special-Sound Σ-Protocols

Here, we introduce a new and improved knowledge extractor for Γ-special-sound Σ-protocols. Our main objective is to avoid the factor $1/t_\Gamma$ loss in success probability exhibited by the aforementioned extractor of [7]. As explained earlier, this new extractor will be essential for our analysis of the knowledge soundness of the Fiat–Shamir transformation of multi-round protocols then in the later sections.

Remark 5. In the full version of this paper [7, Appendix A], we also present an alternative extractor for Σ-protocols. This construction admits a substantially simpler analysis and a slightly improved expected running time, at the cost of a minor loss in success probability. For 3-round Σ-protocols, this loss has no effect on the derived knowledge error. However, when applied to multi-round interactive proofs, the resulting knowledge error is slightly larger than the optimal cheating probability of dishonest provers, and is therefore suboptimal. For this reason, we do not rely on this alternative construction in our main proofs.

4.1. *The Extractor*

Similar to the recent works on the topic, we now continue the discussion in a more abstract language. Consider an algorithm $\mathcal{A} : \mathcal{C} \to \{0,1\}^*$, as well as a verification predicate $\mathcal{V} : \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. Naturally, we would instantiate $\mathcal{A}$ with a dishonest prover $\mathcal{P}^*$, attacking a Γ-out-of-$\mathcal{C}$ special-sound Σ-protocol, with the understanding that the output $y = (a, z)$ of $\mathcal{A}(c)$ consists of the first messages a sent by $\mathcal{P}^*$ and its response z produced on challenge c. By a standard averaging argument,[7] we may assume the dishonest prover $\mathcal{P}^*$, and thus the algorithm $\mathcal{A}$, to be deterministic; see for instance [4] 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

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

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 $c_1, \ldots, c_k$ that form a set in Γ.

We propose a new extractor in Fig. 1. In spirit, it is somewhat similar to the extractor considered in [7]: Initially, it first runs $\mathcal{A}$ on a random challenge c and aborts if $\mathcal{A}$ fails to produce an accepting transcript. If successful, i.e., if $\mathcal{A}$ has produced an accepting transcript, the extractor starts a search by running two geometric experiments: one has the aim of finding more accepting transcripts, and the other is a coin toss used to control the (expected) running time. These two phases of the extractor (the initial run of $\mathcal{A}$ on a random challenge, and the search phase) are respectively denoted $\mathcal{E}^{\mathcal{A}}_{\mathrm{init},\Gamma}$ and $\mathcal{E}^{\mathcal{A}}_{\mathrm{search},\Gamma}$.

A crucial difference is that the extractor of [7] recursively invokes a subextractor that needs to find one less accepting transcript, trying that subextractor multiple times. This is obviously suboptimal: an invocation of the subextractor may fail after some (but not enough) accepting transcripts have been found, in which case these accepting transcripts are then "forgotten," i.e., not considered anymore by the next try of running the subextractor. By contrast, our new extractor is more straightforward in that way in that it simply collects the challenges iteratively, and it never throws away an accepting transcript.

[7] This argument only relies on the definition of knowledge soundness (Definitions 1 and 3), namely the linearity of the knowledge error bound in $\epsilon^{\mathcal{V}}(\mathcal{A})$, and the standard notion of oracle access, where the extractor controls the randomness of the prover (in particular, it can choose it at random and then keep it fixed for further invocations).

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 $(c_1, \ldots, c_k, y_1, \ldots, y_k)$ with $\{c_1, \ldots, c_k\} \in \Gamma$ and $\mathcal{V}(c_j, y_j) = 1 \quad \forall j$.

1. Sample $c_1 \in \mathcal{C}$ uniformly at random and evaluate $y_1 \leftarrow \mathcal{A}(c_1)$. $\boxed{\mathcal{E}^{\mathcal{A}}_{\text{init},\Gamma}}$
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 $(c_1, y_1) \in \mathcal{C} \times \{0,1\}^*$.

4. Else set $S_1 = \{c_1\} \subseteq \mathcal{C}$, $i = 1$ and $\text{COIN} = 0$. $\boxed{\mathcal{E}^{\mathcal{A}}_{\text{search},\Gamma}(c_1, y_1)}$
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 $d \in \mathcal{C}$ in case $i = 1$) uniformly at random and set $\text{COIN} = 1$ 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 $\text{COIN} = 1$.
6. If $\text{COIN} = 1$, abort and output $\perp$.
7. If $S_i = \{c_1, \ldots, c_i\} \in \Gamma$, output $(c_1, \ldots, c_i, y_1, \ldots, y_i)$ with $\mathcal{V}(c_j, y_j) = 1$ for all $1 \le j \le i$.

Fig. 1. Extractor $\mathcal{E}^{\mathcal{A}}_{\Gamma}$ for Γ-Special-Sound Σ-Protocols.

Another difference is that our approach uses a more delicate stopping criterion. Both our extraction approach and prior ones require a stopping criterion, or a coin toss, that allows the extractor to abort if it spends too much time on a particular step. Without such a criterion, the extractor might otherwise run indefinitely when attempting to extract a witness from certain provers. On the other hand, choosing the stopping criterion too aggressively risks aborting too early, leading to a suboptimal knowledge error. Thus, selecting an appropriate stopping criterion is a careful balancing act.

In [7], the stopping criterion is defined as follows. In iteration $i + 1$, when trying to find the $(i + 1)$-th transcript, the coin is set to return 1 (and thus the extractor aborts) if $\mathcal{V}\big(d, \mathcal{A}(d)\big) = 1$ for a challenge $d \in \mathcal{U}_\Gamma(S_{i-1})$ sampled uniformly at random. In other words, the coin returns 1 with probability ϵ_i that an accepting transcript was found in an arbitrary trial of the *previous* (i-th) iteration. This is a natural choice; the $(i + 1)$-th iteration is entered with probability roughly ϵ_i, so allowing the extractor at most $1/\epsilon_i$ trials to find the next transcript provides an elegant running time analysis in which the ϵ_i's cancel out.

However, because an iteration can also end when the next transcript is found, the expected number of trials per iteration is in fact *less* than $1/\epsilon_i$. This observation shows that a more relaxed stopping criterion may be possible, without sacrificing the expected polynomial running time of the extractor. Note that, the extractor does not know the values of the ϵ_i's, making it non-trivial to define an optimal stopping criterion.

In our approach, the coin returns 1 with a (much) smaller probability. Concretely, we set the coin to 1 if and only if the sampled challenge $d \in \mathcal{U}_\Gamma(S_{i-1})$ satisfies $d \notin \mathcal{U}_\Gamma(S_i)$ and $\mathcal{V}\big(d, \mathcal{A}(d)\big) = 1$. Subsequently, we show that the probability that either the next transcript has been found *or* the coin returns 1 is at least ϵ_i, providing the required bound on the expected number of trials.

The approach of [7] yields a simple running time analysis, but incurs the factor t_Γ loss in the success probability. The (much) smaller probability for the coin returning 1 in our approach reduces the probability that the extractor aborts before succeeding, which

in turn improves the success probability of the extractor by a factor t_Γ (while still being able to control the expected running time sufficiently).

Compared to the recursive approach in [7], analyzing the iterative search of our new extractor is quite elaborate. Furthermore, in order to rigorously exploit the above intuitive improvements in the extractor design, we have to be more careful in the analysis. For instance, when analyzing the geometric experiment, we express the relevant bounds in terms of the concrete challenges collected so far, and then average in the end, while [7] uses a worst case bound over all possibilities.

Altogether, this allows us to argue the following extractability result, which, compared to [7, Theorem 1], avoids the t_Γ in the denominator of the success probability.

Theorem 2. *(Extraction Algorithm - Σ-Protocols) Let $(\Gamma, \mathcal{C})$ be a nonempty monotone structure and let $\mathcal{V} : \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. 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 $\mathcal{A} : \mathcal{C} \to \{0,1\}^*$, has its expected number of queries to $\mathcal{A}$ bounded above by $1 + t_\Gamma \cdot (1 + \kappa_\Gamma)$ and, with probability at least*

$$\frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - \kappa_\Gamma}{1 - \kappa_\Gamma}, \tag{7}$$

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

Remark 6. In contrast to the extractor of [7], the knowledge extractor of Fig. 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 2 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 $\mathcal{C} = \mathcal{U}_\Gamma(\emptyset)$.

4.2. *Some Helpful Notation and Auxiliary Results*

For the purpose of proving Theorem 2 (and the improved running time analysis later), we introduce some notation and show a couple of technical results.

Let $0 < k \in \mathbb{Z}$, and let $\mathbf{c}_k = (c_1, \ldots, c_k)$ be a vector with entries in $\mathcal{C}$ (later, we also allow $c_i = \perp$). Here and in the remainder, if k is fixed and clear from the context, we may also write $\mathbf{c}$ instead of $\mathbf{c}_k$; furthermore, when k and $\mathbf{c}_k$ are given and $1 \leq i \leq k$, we write $\mathbf{c}_i$ for the restriction of $\mathbf{c}_k$ to the first i entries, i.e., $\mathbf{c}_i = (c_1, \ldots, c_i)$. We will abuse notation by occasionally interpreting the vector $\mathbf{c}_i$ as the subset of challenges containing the coordinates of $\mathbf{c}_i$, i.e., the statement $\mathbf{c}_i \in \Gamma \subseteq 2^{\mathcal{C}}$ is interpreted as $\{c_1, \ldots, c_i\} \in \Gamma$. 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 $C \in \mathcal{U}_\Gamma(\mathbf{c}_i)$, 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., $\Pr(U_0) = 1$. [8] Recall that $\mathcal{U}_\Gamma(\mathbf{c}_k) \subseteq \mathcal{U}_\Gamma(\mathbf{c}_{k-1}) \subseteq \cdots \subseteq \mathcal{U}_\Gamma(\mathbf{c}_1) \subseteq \mathcal{C}$ (Eq. (4)), or, in terms of probability events,

$$U_k \implies U_{k-1} \implies \cdots \implies U_0\,. \tag{8}$$

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

$$\begin{aligned}
\delta(\mathbf{c}_i) &:= \Pr\big(V \mid U_i\big) = \Pr\big(\,\mathcal{V}\big(C, \mathcal{A}(C)\big) = 1 \mid C \in \mathcal{U}_\Gamma(\mathbf{c}_i)\,\big)\,,\\
\Delta(\mathbf{c}_i) &:= \Pr\big(V \wedge \neg U_i \mid U_{i-1}\big) = \Pr\big(\,\mathcal{V}\big(C, \mathcal{A}(C)\big)\\
&\qquad = 1 \wedge C \notin \mathcal{U}_\Gamma(\mathbf{c}_i) \mid C \in \mathcal{U}_\Gamma(\mathbf{c}_{i-1})\,\big)\,,\\
\tilde{\delta}(\mathbf{c}_i) &:= \frac{\delta(\mathbf{c}_{i-1}) - \Delta(\mathbf{c}_i)}{1 - \Delta(\mathbf{c}_i)}\,.
\end{aligned} \tag{9}$$

Furthermore, by convention, $\delta(\mathbf{c}_0) = \Pr(V \mid U_0) = \Pr(V)$.

The first two quantities are the parameters of the two geometric experiments that are run in parallel in step 5 in the extractor from Fig. 1, considering the challenges collected so far. 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 $\mathbf{c}_i = (c_1, \ldots, c_i)$, and $\Delta(\mathbf{c}_i)$ is the probability for the considered coin to become 1 (which then means that the extractor stops unsuccessfully). 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 $\delta(\mathbf{c}_i)$:

$$\begin{aligned}
\delta(\mathbf{c}_i) = \Pr(V \mid U_i) &= \frac{\Pr(V \wedge U_i \mid U_{i-1})}{\Pr(U_i \mid U_{i-1})} = \frac{\Pr(V \wedge U_i \mid U_{i-1})}{1 - \Pr(\neg U_i \mid U_{i-1})}\\
&\geq \frac{\Pr(V \wedge U_i \mid U_{i-1})}{1 - \Pr(V \wedge \neg U_i \mid U_{i-1})}\\
&= \frac{\Pr(V \wedge U_i \mid U_{i-1})}{1 - \Delta(\mathbf{c}_i)} = \frac{\Pr(V \mid U_{i-1}) - \Pr(V \wedge \neg U_i \mid U_{i-1})}{1 - \Delta(\mathbf{c}_i)}\\
&= \frac{\delta(\mathbf{c}_{i-1}) - \Delta(\mathbf{c}_i)}{1 - \Delta(\mathbf{c}_i)} = \tilde{\delta}(\mathbf{c}_i)\,.
\end{aligned} \tag{10}$$

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

$$\begin{aligned}
\delta(\mathbf{c}_i) + \Delta(\mathbf{c}_i) - \Delta(\mathbf{c}_i)\delta(\mathbf{c}_i) &= \delta(\mathbf{c}_i)\big(1 - \Delta(\mathbf{c}_i)\big) + \Delta(\mathbf{c}_i)\\
&\geq \tilde{\delta}(\mathbf{c}_i)\big(1 - \Delta(\mathbf{c}_i)\big) + \Delta(\mathbf{c}_i) = \delta(\mathbf{c}_{i-1})\,,
\end{aligned} \tag{11}$$

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

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

$$\begin{aligned}\frac{\delta(\mathbf{c}_i)+\Delta(\mathbf{c}_i)-\Delta(\mathbf{c}_i)\delta(\mathbf{c}_i)}{\delta(\mathbf{c}_i)} &= 1-\Delta(\mathbf{c}_i)+\frac{\Delta(\mathbf{c}_i)}{\delta(\mathbf{c}_i)} \leq 1-\Delta(\mathbf{c}_i)+\frac{\Delta(\mathbf{c}_i)}{\tilde{\delta}(\mathbf{c}_i)} \\ &= \frac{\delta(\mathbf{c}_{i-1})}{\tilde{\delta}(\mathbf{c}_i)}\,.\end{aligned} \tag{12}$$

Looking ahead, Eq. (11) will be used in the running time analysis (here and for the improved variant), while Eq. (12) will be used for the analysis of the success probability.

The following lemma provides a useful bound for a specific composition of consecutive Δ-values $\Delta(\mathbf{c}_0), \ldots, \Delta(\mathbf{c}_k)$. This lemma crucially relies on the fact that $U_i \implies U_{i-1}$ for all i. 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

$$\Pr\big(U_k\big) = \Pr\big(U_k \wedge \ldots \wedge U_1\big) = \prod_{i=1}^{k} \Pr\big(U_i \mid U_1 \wedge \ldots \wedge U_{i-1}\big) = \prod_{i=1}^{k} \Pr\big(U_i \mid U_{i-1}\big)\,. \tag{13}$$

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

Lemma 4. *Let $\mathbf{c}_k \in \mathcal{C}^k$ for some k and let $\mathbf{c}_i = (c_1, \ldots, c_i)$ for all $1 \leq i \leq k$. Then*

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

Proof. The proof of the lemma will proceed inductively over k. The lemma trivially holds for the base case $k = 1$ (by definition of U_0 being always satisfied). So let us assume the lemma is proven for $k' = k-1$. Then, by the induction hypothesis

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

where $(*)$ holds since $U_k \implies U_{k-1}$ and hence $U_k \wedge \neg U_{k-1}$ is empty. This completes the proof of the lemma. □

The following auxiliary lemma makes the connection of these δ-quantities to $\mathcal{A}$'s success probability $\epsilon^{\mathcal{V}}(\mathcal{A})$ 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 5. *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 $1 \leq i \leq k$. Then*

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

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

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

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

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

Hence,

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

and

$$\begin{aligned}\delta(\mathbf{c}_0) \prod_{i=1}^{k} \frac{\tilde{\delta}(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} &= \delta(\mathbf{c}_0) \prod_{i=1}^{k} \frac{\delta(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} \frac{\Pr\big(U_i \mid U_{i-1}\big)}{1 - \Delta(\mathbf{c}_i)} \\ &= \delta(\mathbf{c}_k) \prod_{i=1}^{k} \frac{\Pr\big(U_i \mid U_{i-1}\big)}{1 - \Delta(\mathbf{c}_i)} \\ &= \delta(\mathbf{c}_k) \Pr\big(U_k\big) \prod_{i=1}^{k} \frac{1}{1 - \Delta(\mathbf{c}_i)} \\ &= \Pr\big(V \wedge U_k\big) \prod_{i=1}^{k} \frac{1}{1 - \Delta(\mathbf{c}_i)},\end{aligned} \tag{14}$$

where the third equality uses Eq. (13). By Lemma 4, it thus follows that

$$\delta(\mathbf{c}_0) \prod_{i=1}^{k} \frac{\tilde{\delta}(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} \geq \frac{\Pr\big(V \wedge U_k\big)}{1 - \Pr\big(V \wedge \neg U_k\big)} = \frac{\Pr\big(V\big) - \Pr\big(V \wedge \neg U_k\big)}{1 - \Pr\big(V \wedge \neg U_k\big)}. \tag{15}$$

Now note that, since $\mathcal{C} \setminus \mathcal{U}_\Gamma(\mathbf{c}) \notin \Gamma$ for all $\mathbf{c} \notin \Gamma$ (see Eq. (5)),

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

Hence, by Eq. (15) and the monotonicity (decreasing) of the function $x \mapsto \frac{q-x}{1-x}$ for all $0 \leq q \leq 1$, it follows that

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

which completes the proof of the lemma. □

4.3. *The Extractor Analysis (Proof of Theorem 2)*

We are now ready to analyze the success probability and the expected running time of our extractor. We note that the running time analysis provided here will be superseded by the refined variant discussed and proven in Sect. 4.4; we nevertheless offer the proof for the simpler variant here for didactic reasons, and for the reader who is only interested in the interactive case (where the refined variant is not needed).

Proof of Theorem 2. The extractor $\mathcal{E}_\Gamma^{\mathcal{A}}$ is formally defined in Fig. 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 6.

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 $C_i = \perp$ if the extractor $\mathcal{E}_\Gamma^{\mathcal{A}}$ finishes (successfully or unsuccessfully) before finding i challenges, i.e., C_i has support in $\mathcal{C} \cup \{\perp\}$. We write $\boldsymbol{C}_i = (C_1, \ldots, C_i)$ and $\boldsymbol{C} = \boldsymbol{C}_{t+1}$. By the way the extractor works, $\boldsymbol{C}$ is distributed over

$$\mathcal{G} := \{\mathbf{c} \in (\mathcal{C} \cup \{\perp\})^{t+1} : (\mathbf{c}_i \in \Gamma \vee c_i = \perp) \Rightarrow c_{i+1} = \perp \text{ for all } 1 \leq i \leq t\} ,$$

where, following our convention, $\mathbf{c}_i = (c_1, \ldots, c_i)$ consists of the first i entries of $\mathbf{c} = (c_1, \ldots, c_{t+1})$, and we slightly abuse notation and write $\mathbf{c}_i \in \Gamma$ to express that $\mathbf{c}_j \in \Gamma$ for some $j \leq i$ (in other words, we ignore $\perp$-entries). Hence, the extractor succeeds in the event $[\boldsymbol{C} \in \Gamma]$, and we are thus interested in the probability $\Pr(\boldsymbol{C} \in \Gamma)$.

We first consider the case where $\Omega := \{c \in \mathcal{C} : \Pr\big(\mathcal{V}\big(c, \mathcal{A}(c)\big) = 1\big) > 0\} \notin \Gamma$. In this case,

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

i.e., $\epsilon^{\mathcal{V}}(\mathcal{A}) \leq \kappa_\Gamma$, and so the bound on the success probability holds trivially. We will thus assume in the remainder that $\Omega \in \Gamma$.

We will relate the probability $\Pr(\boldsymbol{C} \in \Gamma)$ of the extractor succeeding to the probability $\Pr(\boldsymbol{C}' \in \Gamma)$, where $\boldsymbol{C}'$ is defined in a similar way, but for the *unbounded* variant of $\mathcal{E}^{\mathcal{A}}_{\Gamma}$ that has no coin toss forcing the extractor to stop: in each iteration it keeps trying until it has successfully found the next challenge. By assumption $\Omega \in \Gamma$, there is always a next challenge to be found, until it has collected a qualified set; thus, $\Pr(\boldsymbol{C}' \in \Gamma) = 1$. By definition of the unbounded extractor, $\Pr(\boldsymbol{C}' = \mathbf{c}) = \prod_i \Pr(C'_{i+1} = c_{i+1} \mid \boldsymbol{C}'_i = \mathbf{c}_i)$ for

$$\Pr(C'_{i+1} = c_{i+1} \mid \boldsymbol{C}'_i = \mathbf{c}_i) = \Pr(C_{i+1} = c_{i+1} \mid C_{i+1} \neq \perp \wedge \boldsymbol{C}_i = \mathbf{c}_i)\,,$$

with the understanding that $\Pr(C'_{i+1} = \perp \mid \boldsymbol{C}'_i = \mathbf{c}_i) = 1$ if $\mathbf{c}_i \in \Gamma$, i.e., $\Pr(C_{i+1} \neq \perp \wedge \boldsymbol{C}_i = \mathbf{c}_i) = 0$.

Consider $\mathbf{c} \in \mathcal{G}$ with $\mathbf{c} \in \Gamma$, and let $i \geq 1$ be such that $\mathbf{c}_i \notin \Gamma$ and $\Pr(\boldsymbol{C}_i = \mathbf{c}_i) > 0$. Then, conditioned on having found the challenges $\mathbf{c}_i$, i.e., 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, i.e., success probability, $\delta(\mathbf{c}_i)$. The second geometric experiment repeatedly runs $y \leftarrow \mathcal{A}(d)$ for a uniformly random $d \in \mathcal{U}_{\Gamma}(\mathbf{c}_{i-1})$ 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 Eq. (12),

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

and (noting that $c_{i+1} \neq \perp$ since $\mathbf{c}_i \notin \Gamma$ yet $\mathbf{c} \in \Gamma$)

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

Similarly, $\Pr\big(C_1 = c_1\big) = \Pr(C'_1 = c_1) \cdot \Pr\big(C_1 \neq \perp\big) = \Pr(C'_1 = c_1) \cdot \Pr\big(V\big) = \Pr(C'_1 = c_1) \cdot \delta(\mathbf{c}_0)$.

Let now k be such that $\mathbf{c}_k \in \Gamma$ but $\mathbf{c}_{k-1} \notin \Gamma$. Then

$$\begin{aligned}\Pr(\boldsymbol{C} = \mathbf{c}) = \Pr(\boldsymbol{C}_k = \mathbf{c}_k) &= \Pr(C_1 = c_1) \prod_{i=1}^{k-1} \Pr(C_{i+1} = c_{i+1} \mid \boldsymbol{C}_i = \mathbf{c}_i) \\ &\geq \Pr(C'_1 = c_1) \cdot \delta(\mathbf{c}_0) \cdot \prod_{i=1}^{k-1} \Pr(C'_{i+1} = c_{i+1} \mid \boldsymbol{C}'_i = \mathbf{c}_i) \cdot \frac{\tilde{\delta}(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} \\ &= \Pr(\boldsymbol{C}'_k = \mathbf{c}_k) \cdot \delta(\mathbf{c}_0) \prod_{i=1}^{k-1} \frac{\tilde{\delta}(\mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} \\ &\geq \Pr(\boldsymbol{C}' = \mathbf{c}) \cdot \frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - \kappa_\Gamma}{1 - \kappa_\Gamma}\,,\end{aligned} \tag{16}$$

where the final inequality follows from Lemma 5. The first equality holds since $C_i = \perp$ with certainty for $i > k$ when $\boldsymbol{C}_k = \mathbf{c}_k$ for $\mathbf{c}_k \in \Gamma$, and similarly for the final equality.

Hence, by summing over all $\mathbf{c} \in \mathcal{G}$ with $\mathbf{c} \in \Gamma$,

$$\Pr(\boldsymbol{C} \in \Gamma) \geq \Pr(\boldsymbol{C}' \in \Gamma) \cdot \frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - \kappa_\Gamma}{1 - \kappa_\Gamma} = \frac{\epsilon^{\mathcal{V}}(\mathcal{A}) - \kappa_\Gamma}{1 - \kappa_\Gamma}\,,$$

where the equality follows from $\Pr(\boldsymbol{C}' \in \Gamma) = 1$, which holds for the considered case $\Omega \in \Gamma$.

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 Eq. (11).

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 $\mathbf{c}_i \notin \Gamma$ (Eq. (5)). 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 $c_i = \perp$ (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 $1 \leq i \leq t$,

$$\mathbb{E}[Y_{i+1}] = \sum_{\mathbf{c}_i \in T_i} \Pr(\boldsymbol{C}_i = \mathbf{c}_i) \cdot \mathbb{E}[Y_{i+1} \mid \boldsymbol{C}_i = \mathbf{c}_i] \leq (1 + \kappa_\Gamma) \sum_{\mathbf{c}_i \in T_i} \frac{\Pr(\boldsymbol{C}_i = \mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})}, \tag{17}$$

where $T_i := \{\mathbf{c}_i \in \mathcal{C}^i : \mathbf{c}_i \notin \Gamma \wedge \Pr(\boldsymbol{C}_i = \mathbf{c}_i) > 0\}$.

As in the success probability analysis, we will now compare the probability $\Pr(\boldsymbol{C}_i = \mathbf{c}_i)$ with the probability $\Pr(\boldsymbol{C}'_i = \mathbf{c}_i)$ for the unbounded extractor variant, i.e., with $\boldsymbol{C}'_i$ defined as in the above success probability analysis, but now we aim for an *upper* bound. More precisely, we will show that

$$\Pr(\boldsymbol{C}_i = \mathbf{c}_i) \leq \Pr(\boldsymbol{C}'_i = \mathbf{c}_i) \cdot \delta(c_{i-1}).$$

To this end, note that by Eq. (11), for $\mathbf{c}_i \in T_i$ and $j < i$,

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

and thus

$$\begin{aligned}
&\Pr\big(C_{j+1} = c_{j+1} \mid \boldsymbol{C}_j = \mathbf{c}_j\big) \\
&= \Pr\big(C_{j+1} = c_{j+1} \wedge C_{j+1} \neq \perp \mid \boldsymbol{C}_j = \mathbf{c}_j\big) \\
&= \Pr\big(C_{j+1} \neq \perp \mid \boldsymbol{C}_j = \mathbf{c}_j\big) \cdot \Pr\big(C_{j+1} = c_{j+1} \mid C_{j+1} \neq \perp \wedge \boldsymbol{C}_j = \mathbf{c}_j\big) \\
&\leq \frac{\delta(\mathbf{c}_j)}{\delta(\mathbf{c}_{j-1})} \cdot \Pr\big(C_{j+1} = c_{j+1} \mid C_{j+1} \neq \perp \wedge \boldsymbol{C}_j = \mathbf{c}_j\big) \\
&= \frac{\delta(\mathbf{c}_j)}{\delta(\mathbf{c}_{j-1})} \cdot \Pr\big(C'_{j+1} = c_{j+1} \mid \boldsymbol{C}'_j = \mathbf{c}_j\big).
\end{aligned}$$

Hence,

$$\begin{aligned}
\Pr(\boldsymbol{C}_i = \mathbf{c}_i) &= \Pr\big(C_1 = c_1\big) \cdot \prod_{j=1}^{i-1} \Pr\big(C_{j+1} = c_{j+1} \mid \boldsymbol{C}_j = \mathbf{c}_j\big) \\
&\leq \Pr\big(C'_1 = c_1\big) \cdot \prod_{j=1}^{i-1} \frac{\delta(\mathbf{c}_j)}{\delta(\mathbf{c}_{j-1})} \cdot \Pr\big(C'_{j+1} = c_{j+1} \mid \boldsymbol{C}'_j = \mathbf{c}_j\big) \\
&= \Pr(\boldsymbol{C}'_i = \mathbf{c}_i) \cdot \delta(c_{i-1}),
\end{aligned}$$

which proves the claimed upper bound on $\Pr(\boldsymbol{C}_i = \mathbf{c}_i)$.

It now follows that

$$\sum_{\mathbf{c}_i \in T_i} \frac{\Pr(\boldsymbol{C}_i = \mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} \leq \sum_{\mathbf{c}_i \in T_i} \Pr(\boldsymbol{C}'_i = \mathbf{c}_i) \leq 1\,. \tag{18}$$

and thus, by Eq. (17),

$$\mathbb{E}[Y_{i+1}] \leq (1 + \kappa_\Gamma)\,.$$

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

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

which completes the proof. □

4.4. *Refined Running Time Analysis*

In this section, we refine the running time analysis of the knowledge extractor of Fig. 1. Instead of simply counting the (expected) number of $\mathcal{A}$-invocations, we associate $\mathcal{A}$ with a cost function $\theta : \mathcal{C} \to \mathbb{R}_{\geq 0}$, 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 6. (Refined Running Time Analysis - IP Extractor) *Let $(\Gamma, \mathcal{C})$ be a nonempty monotone structure and $\mathcal{A} : \mathcal{C} \to \{0,1\}^*$ an algorithm accompanied by a verification predicate $\mathcal{V} : \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. Further, let $\theta : \mathcal{C} \to \mathbb{R}_{\geq 0}$, such that $\theta(c)$ denotes the cost of evaluating $\mathcal{A}(c)$.*

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

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

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

The proof follows closely the running time analysis in the proof of Theorem 2, but generalizing it to the setting here, where the cost of evaluating $\mathcal{A}(c)$ depends on c.

Proof of Lemma 6. We keep the notations of the previous proof. For $1 \leq i < t$, let $\mathbf{c}_i \in \mathcal{C}^i$ with $\mathbf{c}_i \notin \Gamma$ and $\Pr[\boldsymbol{C}_i = \mathbf{c}_i] > 0$. 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

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

where the inequality follows from Eq. (11). Thus, this run of two geometric experiments is again a geometric experiment, but now with parameter p.

We now evaluate the cost of this (combined) geometric experiment. 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, in each trial the expected cost of the $\mathcal{A}$-invocations is at most

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

where, in the first inequality, we use that $U_i \implies U_{i-1}$ and thus $\Pr(U_i) \leq \Pr(U_{i-1})$. Moreover, the second inequality follows since $\theta(c) \geq 0$ 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 $\mathcal{C} \setminus \mathcal{U}_\Gamma(\mathbf{c}_i) \notin \Gamma$ (Eq. (5)).

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

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

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 $c_i = \perp$ (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 $1 \leq i \leq t$

$$\mathbb{E}[Y_{i+1}] = \sum_{\mathbf{c}_i \in T_i} \Pr(\boldsymbol{C}_i = \mathbf{c}_i) \mathbb{E}[Y_{i+1} \mid \boldsymbol{C}_i = \mathbf{c}_i] \leq \frac{\mathbb{E}[\theta(C)]}{1 - \kappa_\Gamma} \sum_{\mathbf{c}_i \in T_i} \frac{\Pr(\boldsymbol{C}_i = \mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})}$$

where $T_i := \{\mathbf{c}_i \in \mathcal{C}^i : \mathbf{c}_i \notin \Gamma \wedge \Pr(\boldsymbol{C}_i = \mathbf{c}_i) > 0\}$.

Recycling inequality (18) from the running time analysis of Sect. 4.3, i.e.,

$$\sum_{\mathbf{c}_i \in T_i} \frac{\Pr(\boldsymbol{C}_i = \mathbf{c}_i)}{\delta(\mathbf{c}_{i-1})} \leq 1 ,$$

it follows that

$$\mathbb{E}[Y_{i+1}] \leq \frac{\mathbb{E}[\theta(C)]}{1 - \kappa_\Gamma} .$$

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

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

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 Sect. 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. For comparison, in general the security loss of the Fiat–Shamir transformation may be exponential in the number of rounds.

In Sect. 5.1, we will set the stage and introduce the required notation and auxiliary lemmas. In Sect. 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 Sect. 5.3.

5.1. *Preliminary Discussion*

Following the notation of prior works, and in line with the notation used in Sect. 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 (for a fixed statement x). 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 $\mathsf{ro} : \mathcal{M} \to \mathcal{C}$, 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 $c = \mathsf{ro}(a)$.[9]

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

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

[9] 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 $c = \mathsf{ro}(x, a)$.

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,

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

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

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

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 7. **(Probabilistic Algorithms)** As discussed in Sect. 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 $(a, c_1, z_1), \ldots, (a, c_k, z_k)$, with common first message a and such that $\{c_1, \ldots, c_k\} \in \Gamma$. As we show below, this can be achieved by combining the extractor of Sect. 4 for (interactive) Σ-protocols with the techniques of [5,6]. Deriving its success probability and 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

$$\mathcal{A}_\alpha^{\mathsf{ro}} : \mathcal{C} \to \mathcal{M} \times \{0,1\}^*, \quad c \mapsto (a, z) := \mathcal{A}(\mathsf{ro}[\alpha \mapsto c]), \tag{19}$$

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

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

The corresponding verification predicate is

$$\mathcal{V}_\alpha(a,c,z) = \begin{cases} 1 & \text{if } \mathcal{V}(a,c,z)=1 \text{ and } a=\alpha; \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

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 $\mathcal{V}_\alpha(c, \mathcal{A}(\mathsf{ro}))$ 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 $\mathcal{V}_\alpha^{\mathsf{ro}}(\mathcal{A}(\mathsf{ro})) = \mathcal{V}_\alpha(\mathsf{ro}(\alpha), \mathcal{A}(\mathsf{ro}))$. The algorithm $\mathcal{A}_\alpha^{\mathsf{ro}}$ now has a naturally defined success probability

$$\epsilon^{\mathcal{V}_\alpha}(\mathcal{A}_\alpha^{\mathsf{ro}}) = \Pr\big(\mathcal{V}_\alpha\big(C, \mathcal{A}_\alpha^{\mathsf{ro}}(C)\big) = 1\big) = \Pr\big(\mathcal{V}_\alpha\big(C, \mathcal{A}(\mathsf{ro}[\alpha \mapsto C])\big) = 1\big)\,,$$

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

Remark 8. **(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 Sect. 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 7. *Let $\mathcal{A} : \mathcal{C}^{\mathcal{M}} \to \mathcal{M} \times \{0,1\}^*$ be an algorithm together with a verification predicate $\mathcal{V} : \mathcal{M} \times \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. Then*

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

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

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

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

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. □

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

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

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

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 $|S(\mathsf{ro})| \le Q$.

If $\mathcal{A}(\mathsf{ro})$ does not query message α, i.e., $\alpha \notin \mathcal{S}(\mathsf{ro})$, then $\mathcal{A}_\alpha^{\mathsf{ro}}(c) = \mathcal{A}(\mathsf{ro}[\alpha \mapsto c])$ 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

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

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 $\mathcal{A}_a^{\mathsf{ro}}$, i.e., it proceeds as in the interactive setting, but now using the algorithm $\mathcal{A}_a^{\mathsf{ro}}$ (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., $(a', z') \leftarrow \mathcal{A}(\mathsf{ro}[a \mapsto c']) = \mathcal{A}_a^{\mathsf{ro}}(c')$ is early-aborted if $a' \neq a$ (see also Remark 8). The extractor is formally described in Fig. 2. Its properties are summarized in Theorem 3.

We note that the extractor, as described here, is not efficient because it chooses a random oracle $\mathsf{ro} \in \mathcal{C}^{\mathcal{M}}$ uniformly at random. Thus, we take it as understood (here and later) that the sampling of ro is done using standard lazy sampling, i.e., the function values of ro are sampled on the fly (only) when needed.

We also note that the early-abort property is only exploited in the refined running time analysis of Sect. 5.3 and, subsequently, in the analysis of multi-round interactive proofs. For this reason, this property will not play a role in Theorem 3 and its proof.

Theorem 3. (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 $\mathcal{A} = (A, Z) : \mathcal{C}^{\mathcal{M}} \to \mathcal{M} \times \{0,1\}^*$ defined as above with verification predicate $\mathcal{V} : \mathcal{M} \times \mathcal{C} \times \{0,1\}^* \to \{0,1\}$, has its expected number of queries to $\mathcal{A}$ bounded above by $1 + q(\mathcal{A}) \cdot t_\Gamma \cdot (1 + \kappa_\Gamma)$ and, with probability at least*

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

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

Remark 9. We note that Theorem 3 makes no claim about the computational complexity of the extractor, only about the query complexity. From the description we see that for the extractor to be computationally efficient (which we ultimately want), it is necessary and sufficient that it is computationally feasible to sample from $\mathcal{U}_\Gamma(S)$. This is made explicit wherever this is crucial (e.g., in Theorem 4).

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 (19) and (20)) 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.

$\boxed{\mathcal{F}^{\mathcal{A}}_{\mathrm{init},\Gamma}}$

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 $(a, \mathsf{ro}(a), z)$.

$\boxed{\mathcal{F}^{\mathcal{A}}_{\mathrm{search},\Gamma}(a, z, \mathsf{ro}) = \mathcal{E}^{\mathcal{A}^{\mathsf{ro}}_{a}}_{\mathrm{search},\Gamma}(\mathsf{ro}(a), a, z)}$

4. Else, set $S_1 = \{\mathsf{ro}(a)\} \subseteq \mathcal{C}$, $i = 1$ and $\textsc{coin} = 0$.
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 $\textsc{coin} = 1$ 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 $\textsc{coin} = 1$.
6. If $\textsc{coin} = 1$, 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 $2 \leq j \leq i$.

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

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 $\mathbb{E}_r[\ldots]$ 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 $(a, z) = \mathcal{A}(\mathsf{ro}[\alpha \mapsto c]) = \mathcal{A}^{\mathsf{ro}}_{\alpha}(c)$. As such, conditioned on $\Psi_{\alpha}(\mathsf{ro})$, the first steps of the extractors $\mathcal{F}^{\mathcal{A}}$ and $\mathcal{E}^{\mathcal{A}^{\mathsf{ro}}_{\alpha}}$ are almost identical. The only difference between the two is that $\mathcal{E}^{\mathcal{A}^{\mathsf{ro}}_{\alpha}}$ aborts if its first $\mathcal{A}$-invocation outputs a message

$a \neq \alpha$, whereas $\mathcal{F}^{\mathcal{A}}$ would then proceed with running $\mathcal{E}^{\mathcal{A}_a^{\mathsf{ro}}}_{\text{search}}$. [10] It thus follows that

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

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

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

where the first inequality follows from Theorem 2 and the final inequality follows from Lemma 7 together with the observation that

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

and thus

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

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_α^{ro} denote the event that the first $\mathcal{A}$-invocation of the extractor $\mathcal{E}^{\mathcal{A}_\alpha^{\mathsf{ro}}}$, when applied to $\mathcal{A}_\alpha^{\mathsf{ro}}$, is successful. Further, we write X_α^{ro} for the

[10] 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 6.

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 2, if follows that

$$\Pr(V^{\mathsf{ro}}_\alpha) \cdot \mathbb{E}[X^{\mathsf{ro}}_\alpha \mid V^{\mathsf{ro}}_\alpha] = \mathbb{E}[X^{\mathsf{ro}}_\alpha] \leq t_\Gamma \cdot (1 + \kappa_\Gamma)\,. \tag{21}$$

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,

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

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 immediately follows from Lemma 8 and Theorem 3.

Theorem 4. (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 κ_Γ is the knowledge error of the (interactive) Σ-protocol Π, as defined in Eq. (6).

More precisely, $\mathsf{FS}[\Pi]$ admits a knowledge extractor that, on input a statement x and given oracle access to a Q-query prover $\mathcal{P}^*$, has its expected number of queries to $\mathcal{P}^*$

bounded above by $1 + (Q+1) \cdot t_\Gamma \cdot (1 + \kappa_\Gamma)$ *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 10. **(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 for the knowledge extractor of the Fiat–Shamir transformation is rather generic.

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

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

an algorithm with a verification predicate $\mathcal{V} : \mathcal{M} \times \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. *Let* $\theta : \mathcal{C}^{\mathcal{M}} \to \mathbb{R}_{\geq 1}$, *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 Fig. 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 3, 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 6.

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

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

As in the proof of Theorem 3, 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 $\mathcal{E}_{\Gamma}^{\mathcal{A}_\alpha^{\mathsf{ro}}}$, not counting the cost of the first $\mathcal{A}_\alpha^{\mathsf{ro}}$-query. Then, recalling (21), 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 3, we used Theorem 2 to bound $\mathbb{E}[X_\alpha^{\mathsf{ro}}]$. Here we use the refined running time bound of Lemma 6, from which it follows that

$$\begin{aligned} \mathbb{E}[X_\alpha^{\mathsf{ro}}] \leq t_\Gamma \cdot \frac{\mathbb{E}[\theta_\alpha^{\mathsf{ro}}(C)]}{1-\kappa_\Gamma} = \frac{t_\Gamma}{1-\kappa_\Gamma} \\ \Big(\Pr(A_\alpha^{\mathsf{ro}}(C) = \alpha)\mathbb{E}[\theta_\alpha^{\mathsf{ro}}(C) \mid A_\alpha^{\mathsf{ro}}(C) = \alpha] + \Pr(A_\alpha^{\mathsf{ro}}(C) \neq \alpha)\Big) \end{aligned} \tag{22}$$

where C is uniformly random in $\mathcal{C}$. Note that we ignore the cost of the first $\mathcal{A}$ invocation, and thus, in Eq. (22), omitted the first summand $\mathbb{E}[\theta_\alpha^{\mathsf{ro}}(C)]$ from the running time bound of Lemma 6.

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,

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

Hence, by Eq. (22),

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

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

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

which shows that

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

It therefore follows that

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

Further, since $\theta^{\mathsf{ro}}_\alpha(c)\ge 1$ 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 $\{\alpha:\epsilon^{\mathcal{V}_\alpha}(\mathcal{A}^{\mathsf{ro}}_\alpha)>0\}\subseteq\mathcal{M}$ to all of $\mathcal{M}$. It therefore follows that

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

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

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

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

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 (Fig. 3).

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 $\mathcal{C} := (\mathcal{C}_1, \ldots, \mathcal{C}_\mu)$. Then, towards defining multi-round special-soundness as defined in [7], we recall the following.

Definition 10. *(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

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

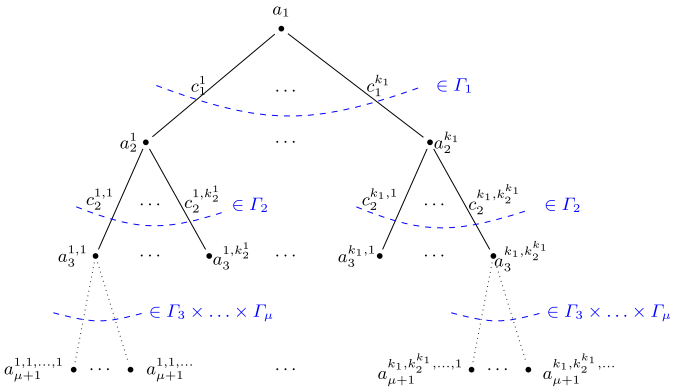

Fig. 3. $(\Gamma_1, \ldots, \Gamma_\mu)$-tree of transcripts of a $(2\mu+1)$-move interactive proof (Definition 10), arranged as a tree. Each element in $\text{TREE}_{\Gamma_1,\ldots,\Gamma_\mu}(a_1)$ consists of (all the a- and c-labels along) a path from the root a_1 to a leaf. The incident challenges for each a_i form a set in Γ_i, i.e, $\{c_1^1, \ldots, c_1^{k_1}\} \in \Gamma_1$ etc.

We write $\text{TREE}_\Gamma(a)$ and TREE_Γ 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 $T \in \text{TREE}_{\boldsymbol{\Gamma}}$ of triples $(a_1, c_1, T') \in \{a_1\} \times \mathcal{C}_1 \times \text{TREE}_{\Gamma_2,\ldots,\Gamma_\mu}$ such that

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

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

Remark 11. 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 11. *(Tree of Accepting Transcripts)* Let

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

be a verification predicate. A $\boldsymbol{\Gamma}$-tree T of transcripts (with arbitrary trunk a_1) is called a $\boldsymbol{\Gamma}$-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 $\text{TREE}_{\boldsymbol{\Gamma}}^{\mathcal{V}}$ and $\text{TREE}^{\mathcal{V}}(a_1)$ for the set of all $(\Gamma_1, \ldots, \Gamma_\mu)$-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 12. *($(\Gamma_1, \ldots, \Gamma_\mu)$-out-of-$(\mathcal{C}_1, \ldots, \mathcal{C}_\mu)$Special-Soundness)* Let $(\Gamma_i, \mathcal{C}_i)$ be monotone structures for $1 \leq i \leq \mu$, and let $\mathbf{\Gamma} := (\Gamma_1, \ldots, \Gamma_\mu)$ and $\mathcal{C} := (\mathcal{C}_1, \ldots, \mathcal{C}_\mu)$. 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-*$\mathcal{C}$ *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 $t_{\mathbf{\Gamma}} := \prod_{i=1}^{\mu} t_{\Gamma_i}$ is polynomially bounded, then [7] establishes that $\mathbf{\Gamma}$-out-of-$\mathcal{C}$ special-sound interactive proofs are knowledge sound with knowledge error

$$\kappa_{\mathbf{\Gamma}} = 1 - \prod_{i=1}^{\mu} (1 - \kappa_{\Gamma_i}) \,, \tag{23}$$

where the κ_{Γ_i} are defined as in (6). However, we don't need this multi-round result in our work here; we only need the (improved) 3-round case of Sect. 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 [2,4–7], 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.[11]

The same problem surfaced in the analysis of Fiat–Shamir transformations of $(k_1, \ldots, k_\mu)$-special-sound interactive proofs [5,6]. 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

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

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.

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 8). 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 [5,6] 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 Sect. 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 $\Gamma_1, \ldots, \Gamma_\mu$, each $\Gamma_i \subseteq 2^{\mathcal{C}_i}$ for challenge sets $\mathcal{C}_i$. As before, we let $\boldsymbol{\Gamma} := (\Gamma_1, \ldots, \Gamma_\mu)$ and $\mathcal{C} := (\mathcal{C}_1, \ldots, \mathcal{C}_\mu)$. Further we denote by $\overrightarrow{\mathsf{ro}} = (\mathsf{ro}_1, \ldots, \mathsf{ro}_\mu)$ 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 4). 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.[12]

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

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

Hence, we model the attacker as an algorithm $\mathcal{A}$ that, on input μ random oracles $\mathsf{ro}_1 \in \mathcal{C}_1^{\mathcal{M}}$, $\mathsf{ro}_2 \in \mathcal{C}_2^{\mathcal{M}^2}$, …, $\mathsf{ro}_\mu \in \mathcal{C}_\mu^{\mathcal{M}^\mu}$, outputs a proof $\pi = (a_1, \ldots, a_\mu, a_{\mu+1})$. As for the 3-round case, we treat the random oracles $\mathsf{ro}_1, \ldots, \mathsf{ro}_\mu$ 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 $\mathsf{ro}_1, \ldots, \mathsf{ro}_\mu$.

[12] 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 [5]. Here, the above notation turned out to be more convenient.

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

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

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

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

We write the above as

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

and the success probability of $\mathcal{A}$ (as 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 $T \in \textsc{Tree}_{\mathbf{\Gamma}}^{\mathcal{V}}(a_1)$ of $\mathcal{V}$-accepting transcripts $(a_1, c_1, \ldots, a_\mu, c_\mu, a_{\mu+1})$ 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 whether $T \in \textsc{Tree}_{\mathbf{\Gamma}}^{\mathcal{V}}$. Note that, without loss of generality, we may assume that $\mathcal{F}_\mu^{\mathcal{A}}$'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 9 to be applicable (and this is of course no coincidence).

Definition 13. *(Early-Choice Property)* An extractor $\mathcal{F}_\mu^{\mathcal{A}}$ satisfies the *early-choice* property, if

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

The above early-choice property implies that, already after its first $\mathcal{A}$-invocation, the extractor $\mathcal{F}_\mu^{\mathcal{A}}$ knows the trunk a_1 of the tree it will 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 $\mu = 1$, $\mathcal{F}_1^{\mathcal{A}}$ is simply the extractor from the 3-round case (Fig. 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 12. 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

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

that (unless it aborts) outputs a tree $T'\in\textsc{Tree}_{\Gamma_2,\ldots,\Gamma_\mu}^{\mathcal{V}[\mathsf{ro}_1]}(a_1,a_2)$ 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

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

that, on input ro_1, runs $\mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{ro}_1]}$ 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 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 $\mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{ro}_1]}$, i.e., formally, $\mathcal{V}_1^{\mathsf{ro}_1}(a_1,\{a_2\}\times T'')=\mathcal{W}_{\mu-1}[\mathsf{ro}_1](\{(a_1,a_2)\}\times T'')$.

The crucial point now is 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 Fig. 2. The early-abort strategy can be exploited in Lemma 9, using that, by induction, $\mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{ro}_1]}$ has the early-choice property

(see Lemma 10), 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 Γ_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 9. Recall that, for the base case $\mu = 1$, i.e., when considering Fiat–Shamir transformations of Σ-protocols, Lemma 8 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 8, as our multi-round extractor satisfies the *early-choice property*.

Lemma 10. *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 13.*

Proof. For $\mu = 1$, $\mathcal{F}_1^{\mathcal{A}}$ is simply the extractor from the 3-round case (Fig. 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 11. *For any algorithm $\mathcal{A} : \mathcal{C}_1^{\mathcal{M}} \times \mathcal{C}_2^{\mathcal{M}^2} \times \cdots \times \mathcal{C}_\mu^{\mathcal{M}^\mu} \to \mathcal{M}^\mu \times \{0,1\}^*$ making at most Q queries (in total) to its random oracles, and for $\mathcal{B}$ as in the above recursive construction, we have*

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

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 10), $\mathcal{B}^{\mathcal{A}}(\mathsf{ro}_1) = \mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{ro}_1]}$ 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 $\mathcal{A}(\cdot, \mathsf{ro}_2^r, \ldots, \mathsf{ro}_\mu^r) : \mathcal{C}_1^{\mathcal{M}} \to \mathcal{M}^\mu \times \{0,1\}^*$ 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.[13] Finally, by Lemma 8, 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.

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

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

Proof. The proof is by induction. The base case $\mu = 1$ follows from Theorem 3 and Lemma 8. Thus, consider now $\mu > 1$. First, setting $\mathbf{\Gamma}' = (\Gamma_2, \ldots, \Gamma_\mu)$ 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

$$\begin{aligned}\Pr\left(\mathcal{V}_1^{\mathsf{ro}_1}\left(\mathcal{B}^{\mathcal{A}}(\mathsf{ro}_1)\right) = 1\right) &= \Pr\left(\mathcal{W}_{\mu-1}[\mathsf{ro}_1]\left(\mathcal{F}_{\mu-1}^{\mathcal{A}[\mathsf{ro}_1]}\right) = 1\right)\\ &\geq \frac{\epsilon^{\mathcal{V}[\mathsf{ro}_1]}(\mathcal{A}[\mathsf{ro}_1]) - (Q+1)\cdot\kappa_{\mathbf{\Gamma}'}}{1-\kappa_{\mathbf{\Gamma}'}}.\end{aligned}$$

Hence,

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

By using that

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

where the second equality exploits the particular definition of the Fiat–Shamir transformation (see Remark 12), we see that $\mathbb{E}_{\mathsf{ro}_1}\left[\epsilon^{\mathcal{V}[\mathsf{ro}_1]}(\mathcal{A}[\mathsf{ro}_1])\right] = \epsilon^{\mathcal{V}}(\mathcal{A})$. Hence,

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

[13] 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.

Furthermore, by Theorem 3, exploiting the bound on the q-value from Lemma 11,

$$\Pr\left(\mathcal{F}_{\mu}^{\mathcal{A}} \neq \perp\right) = \Pr\left(\mathcal{F}_{1}^{\mathcal{B}^{\mathcal{A}}} \neq \perp\right) \geq \frac{\epsilon^{\mathcal{V}_1}(\mathcal{B}^{\mathcal{A}}) - (Q+1)\cdot\kappa_{\Gamma_1}}{1-\kappa_{\Gamma_1}}.$$

Thus, writing ϵ for $\epsilon^{\mathcal{V}}(\mathcal{A})$ and exploiting that $(1-\kappa_{\Gamma_1})(1-\kappa_{\mathbf{\Gamma}'}) = 1-\kappa_{\mathbf{\Gamma}}$,

$$\begin{aligned}
\Pr\left(\mathcal{F}_{\mu}^{\mathcal{A}} \neq \perp\right) &\geq \frac{\frac{\epsilon-(Q+1)\cdot\kappa_{\mathbf{\Gamma}'}}{1-\kappa_{\mathbf{\Gamma}'}} - (Q+1)\cdot\kappa_{\Gamma_1}}{1-\kappa_{\Gamma_1}} \\
&= \frac{\epsilon - (Q+1)\cdot\kappa_{\mathbf{\Gamma}'} - (Q+1)\cdot\kappa_{\Gamma_1}\cdot(1-\kappa_{\mathbf{\Gamma}'})}{1-\kappa_{\mathbf{\Gamma}}} \\
&= \frac{\epsilon - (Q+1)\left(\kappa_{\mathbf{\Gamma}'} + \kappa_{\Gamma_1}\cdot(1-\kappa_{\mathbf{\Gamma}'})\right)}{1-\kappa_{\mathbf{\Gamma}}} \\
&= \frac{\epsilon - (Q+1)\left(1-(1-\kappa_{\mathbf{\Gamma}'})(1-\kappa_{\Gamma_1})\right)}{1-\kappa_{\mathbf{\Gamma}}} \\
&= \frac{\epsilon - (Q+1)\cdot\kappa_{\mathbf{\Gamma}}}{1-\kappa_{\mathbf{\Gamma}}},
\end{aligned}$$

which completes the proof. □

6.6. *Running Time Analysis*

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

Proposition 2. *Let $\Gamma_1, \ldots, \Gamma_\mu$ be monotone structures and let $\mathbf{\Gamma} = (\Gamma_1, \ldots, \Gamma_\mu)$. Let $\mathcal{A} : \overrightarrow{\mathcal{RO}} \to \mathcal{M}^\mu \times \{0,1\}^*$ 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 3 together with the bound $q(\mathcal{A}) \leq Q+1$ from Lemma 8. 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 9, then using that $\theta(\mathsf{ro}) = 1$ is constant.

For the induction step, let $\mathbf{\Gamma}' = (\Gamma_2, \ldots, \Gamma_\mu)$ 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 9, 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 the 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 9, and using the bound on $q(\mathcal{B}^{\mathcal{A}})$ from Lemma 11, 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 Sect. 6.4. In fact, from the success probability bound derived in Sect. 6.5 and the running time bound derived in Sect. 6.6, it immediately follows that the Fiat–Shamir transformation of a $(\Gamma_1, \ldots, \Gamma_\mu)$-out-of-$(\mathcal{C}_1, \ldots, \mathcal{C}_\mu)$ 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; it follows immediately from Propositions 1 and 2.

Theorem 5. (Fiat–Shamir Transformation of Multi-Round Proofs) *Let $(\Gamma_i, \mathcal{C}_i)$ be nonempty monotone structures, and let $\mathbf{\Gamma} := (\Gamma_1, \ldots, \Gamma_\mu)$ and $\mathcal{C} := (\mathcal{C}_1, \ldots, \mathcal{C}_\mu)$. Let Π be a $\mathbf{\Gamma}$-out-of-$\mathcal{C}$ special-sound interactive proof. Suppose that $T_{\mathbf{\Gamma}} = \prod_{i=1}^{\mu}(t_{\Gamma_i} + 1)$ is polynomial in the size $|x|$ of the statement x and that, for all i, sampling from $\mathcal{U}_{\Gamma_i}(S_i)$ takes polynomial time (in $|x|$) for all $S_i \subseteq \mathcal{C}_i$ with $|S_i| \leq t_{\Gamma_i}$. Then the Fiat–Shamir transformation $\mathsf{FS}[\Pi]$ of Π is knowledge sound with knowledge error*

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

where $\kappa_{\mathbf{\Gamma}}$, as defined in Eq. (23), is the knowledge error of the interactive proof Π, and Q denotes the query complexity of a prover attacking $\mathsf{FS}[\Pi]$.

More precisely, $\mathsf{FS}[\Pi]$ admits a knowledge extractor that, on input a statement x and given oracle access to a Q-query prover $\mathcal{P}^*$, has its expected number of queries to $\mathcal{P}^*$ bounded above by

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

and succeeds to extract a witness for x with probability at least

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

Remark 13. **(Adaptive Knowledge Soundness of the Adaptive Fiat–Shamir Transformation)** If in Theorem 5 the *adaptive (a.k.a. strong)* Fiat–Shamir transformation is used, then the results of Theorem 5 also hold for *adaptive* knowledge soundness. See Remarks 2 and 3 for the respective notions and further discussion.

Acknowledgements

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). The fourth author is supported by a Veni grant (VI.Veni.222.347) from the Dutch Research Council (NWO). Part of this work was conducted while the author was visiting the Simons Institute for the Theory of Computing.

References

[1] M.A. Aardal, D.F. Aranha, K. Boudgoust, S. Kolby, and A. Takahashi. Aggregating Falcon Signatures with LaBRADOR. In: *CRYPTO*. Ed. by Leonid Reyzin and Douglas Stebila. Lecture Notes in Computer Science, vol. 14920 (Springer, 2024), pp. 71–106.

[2] T. Attema, R. Cramer, and L. Kohl. A Compressed Σ-Protocol Theory for Lattices. In: *CRYPTO*. Ed. by Tal Malkin and Chris Peikert. Lecture Notes in Computer Science, vol. 12826 (Springer, 2021), pp. 549–579.

[3] G. Arnon, A. Chiesa, and E. Yogev. IOPs with inverse polynomial soundness error. In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2023, pp. 752–761.

[4] T. Attema and S. Fehr. Parallel repetition of $(k_1, \ldots, k_\mu)$-special-sound multi-round interactive proofs". In: *CRYPTO*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Lecture Notes in Computer Science, vol. 13507 (Springer, 2022), pp. 415–443.

[5] T. Attema, S. Fehr, and M. Klooß. Fiat–Shamir transformation of multi-round interactive proofs. In: *Theory of Cryptography Conference (TCC)*. Ed. by Eike Kiltz and Vinod Vaikuntanathan. Lecture Notes in Computer Science, vol. 13747 (Springer, 2022), pp. 113–142.

[6] T. Attema, S. Fehr, and M. Klooß. Fiat–Shamir transformation of multi-round interactive proofs (Extended Version). *J. Cryptol.* 36.4 (2023), p. 36.

[7] T. Attema, S. Fehr, and N. Resch. Generalized special-sound interactive proofs and their knowledge soundness. In: *Theory of Cryptography Conference (TCC)*. Ed. by Guy N. Rothblum and Hoeteck Wee. Lecture Notes in Computer Science, vol. 14371 (Springer, 2023), pp. 424–454.

[8] T. Attema, M. Klooß, R.W.F. Lai, and P. Yatsyna. Adaptive special soundness: improved knowledge extraction by adaptive useful challenge sampling. In: *IACR Cryptology ePrint Archive 2024/2038* (2024).

[9] T. Attema, V. Lyubashevsky, and G. Seiler. Practical product proofs for lattice commitments. In: *CRYPTO*. Ed. by Daniele Micciancio and Thomas Ristenpart. Lecture Notes in Computer Science, vol. 12171 (Springer, 2020), pp. 470–499.

[10] T. Attema. Compressed Σ-Protocol Theory. PhD thesis. Leiden University, 2023.

[11] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: short proofs for confidential transactions and more. In: *IEEE Symposium on Security and Privacy (S&P)*. IEEE Computer Society, 2018, pp. 315–334.

[12] C. Baum, J. Bootle, A. Cerulli, R. del Pino, J. Groth, and V. Lyubashevsky. Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits. In: *CRYPTO*. Ed. by Hovav Shacham and Alexandra Boldyreva. Lecture Notes in Computer Science, vol. 10992 (Springer, 2018), pp. 669–699.

[13] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev. Fast reed-solomon interactive oracle proofs of proximity. In: *International Colloquium on Automata, Languages, and Programming, ICALP*. Ed. by Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, vol. 107, 2018, 14:1–14:17.

[14] J. Bootle, A. Cerulli, P. Chaidos, J. Groth, and C. Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In: *EUROCRYPT*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Lecture Notes in Computer Science, vol. 9666 (Springer, 2016), pp. 327–357.

[15] E. Ben-Sasson, A. Chiesa, and N. Spooner. Interactive oracle proofs. In: *Theory of Cryptography Conference (TCC)*. Ed. by Martin Hirt and Adam D. Smith. Lecture Notes in Computer Science, vol. 9986 2016, pp. 31–60.

[16] D. Bernhard, O. Pereira, and B. Warinschi. How not to prove yourself: pitfalls of the Fiat-Shamir heuristic and applications to helios. In: *ASIACRYPT*. Ed. by Xiaoyun Wang and Kazue Sako. Lecture Notes in Computer Science, vol. 7658 (Springer, 2012), pp. 626–643.

[17] R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G.N. Rothblum, R.D. Rothblum, and D. Wichs. Fiat–Shamir: from practice to theory. In: *ACM Symposium on Theory of Computing (STOC)*. Ed. by Moses Charikar and Edith Cohen. ACM, 2019, pp. 1082–1090.
[18] R. Cramer and I. Damgård. Zero-knowledge proofs for finite field arithmetic; or: Can zero-knowledge be for free? In: *CRYPTO*. Ed. by Hugo Krawczyk. Lecture Notes in Computer Science, vol. 1462 (Springer, 1998), pp. 424–441.
[19] A. Chiesa, M. Dall'Agnol, Z. Guan, N. Spooner, and E. Yogev. Untangling the security of Kilian's protocol: upper and lower bounds. In: *IACR Cryptology ePrint Archive 2024/1434* (2024).
[20] A. Chiesa, M. Dall'Agnol, Z. Guan, and N. Spooner. Concrete security for succinct arguments from vector commitments. In: *IACR Cryptology ePrint Archive 2023/1737* (2023).
[21] A. Chiesa, P. Manohar, and N. Spooner. Succinct arguments in the quantum random oracle model. In: *Theory of Cryptography Conference (TCC)*. Ed. by Dennis Hofheinz and Alon Rosen. Lecture Notes in Computer Science, vol. 11892 (Springer, 2019), pp. 1–29.
[22] R. Cramer. Modular design of secure yet practical cryptographic protocols. PhD thesis. CWI and University of Amsterdam, 1996.
[23] J. Don, S. Fehr, and C. Majenz. The Measure-and-Reprogram Technique 2.0: Multi-round Fiat-Shamir and More. In: *CRYPTO*. Ed. by Daniele Micciancio and Thomas Ristenpart. Lecture Notes in Computer Science, vol. 12172 (Springer, 2020), pp. 602–631.
[24] J. Don, S. Fehr, C. Majenz, and C. Schaffner. Efficient NIZKs and signatures from commit-and-open protocols in the QROM. In: *CRYPTO*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Lecture Notes in Computer Science, vol. 13508 (Springer, 2022), pp. 729–757.
[25] M.F. Esgin, N.K. Nguyen, and G. Seiler. Practical exact proofs from lattices: new techniques to exploit fully-splitting rings. In: *ASIACRYPT*. Ed. by Shiho Moriai and Huaxiong Wang. Lecture Notes in Computer Science, vol. 12492 (Springer, 2020), pp. 259–288.
[26] R. Gennaro, D. Leigh, R. Sundaram, and W.S. Yerazunis. Batching Schnorr identification scheme with applications to privacy-preserving authorization and low-bandwidth communication devices. In: *ASIACRYPT*. Ed. by Pil Joong Lee. Lecture Notes in Computer Science, vol. 3329 (Springer, 2004), pp. 276–292.
[27] J. Holmgren. On round-by-round soundness and state restoration attacks. In: *IACR Cryptology ePrint Archive 2019/1261* (2019).
[28] J. Kilian. A note on efficient zero-knowledge proofs and arguments (Extended Abstract). In: *ACM Symposium on Theory of Computing (STOC)*. Ed. by S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis. ACM, 1992, pp. 723–732.
[29] R.W.F. Lai, G. Malavolta, and N. Spooner. Quantum rewinding for many-round protocols. In: *Theory of Cryptography Conference (TCC)*. Ed. by Eike Kiltz and Vinod Vaikuntanathan. Lecture Notes in Computer Science, vol. 13747 (Springer, 2022), pp. 80–109.
[30] V. Lyubashevsky, N.K. Nguyen, and M. Plançon. Lattice-based zero-knowledge proofs and applications: shorter, simpler, and more general. In: *CRYPTO*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Lecture Notes in Computer Science, vol. 13508 (Springer, 2022), pp. 71–101.
[31] S. Micali. Computationally sound proofs. *SIAM J. Comput.* 30(4), pp. 1253–1298 (2000).
[32] S. Micali. CS proofs (Extended abstracts). In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 1994, pp. 436–453.
[33] N. Ron-Zewi and R.D. Rothblum. Proving as fast as computing: succinct arguments with constant prover overhead. In: *ACM Symposium on Theory of Computing (STOC)*. Ed. by Stefano Leonardi and Anupam Gupta. ACM, 2022, pp. 1353–1363.
[34] O. Reingold, G.N. Rothblum, and R.D. Rothblum. Constant-round interactive proofs for delegating computation. In: *ACM Symposium on Theory of Computing (STOC)*. Ed. by Daniel Wichs and Yishay Mansour. ACM, 2016, pp. 49–62.
[35] D. Wikström. Special soundness revisited. *IACR Commun. Cryptol.* 1(3), p. 25 (2024).
[36] D. Wikström. Special soundness in the random oracle model. In: *IACR Commun. Cryptol.* 1(3), p. 26 (2024).