The Cost of Rewinding for Succinct Arguments: Strict-Time Barriers and Expected-Time Security

Alessandro Chiesa
alessandro.chiesa@epfl.ch
EPFL

Ziyi Guan
ziyiguan@mit.edu
MIT

William Wang
ww@priv.pub
NYU

Yuetian Wu
yuetianw@stanford.edu
Stanford University

November 2, 2026

Abstract

Succinct arguments are central cryptographic objects, underlying various applications such as blockchains and image authentication. Almost all succinct arguments used in practice follow the commit-and-open paradigm introduced by Kilian (STOC 1992): commit to a probabilistic proof, then open the few locations the verifier queries. Decades of work have generalized and optimized this paradigm, yet its exact provable security in the standard model remains unclear. Existing analyses proceed by rewinding, and they leave gaps: strict-time analyses lose an inverse-polynomial factor, while expected-time analyses are sharper but apply only to special cases of this paradigm.

We resolve both gaps. On the negative side, we show that the strict-time loss is inherent: no strict polynomial-time reduction relying on a black-box extractor can achieve negligible knowledge error (unless the underlying language is easy). On the positive side, we give a new expected-time analysis, achieving negligible soundness and knowledge errors via an expected polynomial-time reduction against expected polynomial-time adversaries, at the full generality of the paradigm, capturing multi-round and functional variants such as polynomial and linear IOPs compiled with the corresponding commitments. This is the first standard-model analysis to reach this regime for succinct arguments deployed in practice.

Keywords: black-box separation; expected-time analysis; succinct arguments

Contents

1 Introduction **3**
- 1.1 Our results . . . 4
- 1.2 The cost of rewinding . . . 5
- 1.3 Related work . . . 6

2 Techniques **9**
- 2.1 Strict-time barriers for knowledge soundness . . . 9
- 2.2 Expected-time security of Kilian's protocol . . . 11
- 2.3 Expected-time security of the IBCS protocol . . . 17
- 2.4 Expected-time security of the Funky protocol . . . 21

3 Preliminaries **22**
- 3.1 Interactive arguments . . . 22
- 3.2 Query classes . . . 23
- 3.3 Functional commitments . . . 25
- 3.4 Functional interactive oracle proofs . . . 26
- 3.5 Funky protocol . . . 27

4 Barriers for knowledge soundness **29**
- 4.1 Additional preliminaries . . . 29
- 4.2 Strict-time barriers . . . 30
- 4.3 Expected-time barriers . . . 33

5 Expected-time soundness **36**
- 5.1 Adversary construction . . . 36
- 5.2 Security analysis . . . 37

6 Expected-time knowledge soundness **40**
- 6.1 Extractor construction . . . 41
- 6.2 Security analysis . . . 42

A Coverage factor bounds **48**

Acknowledgments **51**

AI disclosure **51**

References **51**

1 Introduction

A succinct argument for a relation R is a computationally sound interactive proof in which, for a given instance $\mathbb{x}$, the argument prover convinces the argument verifier that there exists a witness $\mathbb{w}$ with $(\mathbb{x}, \mathbb{w}) \in R$, while communicating far fewer than $|\mathbb{w}|$ bits. Succinct arguments are central cryptographic primitives with numerous applications. We study succinct arguments for NP in the standard model from falsifiable assumptions.

The commit-and-open paradigm. Kilian's protocol [Kil92], the first succinct argument, is the prototype for almost all practical constructions: the argument prover commits to a probabilistically checkable proof (PCP) via a vector commitment scheme (e.g., a Merkle tree) and sends the commitment to the verifier; the verifier replies with PCP randomness; and the prover opens the queried locations of the PCP. The verifier accepts if the openings are valid and the PCP verifier accepts. The same *commit-and-open* template has since been applied to interactive oracle proofs (IOPs), the multi-round analogue of PCPs, and to functional IOPs such as polynomial IOPs [BCS16; LM19; CHMMVW20; BFS20; CFFQR21; LPS24; CDGS23; LPS25; CGKY26]. In the standard model, security of every such protocol is proved by *rewinding*: the reduction replays the adversary against fresh verifier challenges, collecting a few opened locations per run, until it can reconstruct a full proof string and invoke the security of the underlying probabilistic proof. This paper studies how many rewinds are needed.

The strict-time setting. In the standard *strict-time setting*, all algorithms (the malicious prover, the extractor, the reductions between them) run in worst-case bounded time. The state of the art for Kilian's protocol is as follows [CDGSY24]. Let PCP be a probabilistically checkable proof with proof length ℓ, soundness error ϵ_{PCP}, and knowledge soundness error κ_{PCP}. Let VC be a vector commitment scheme with position binding error ϵ_{VC}. Then, for every $\epsilon > 0$, the interactive argument $\mathsf{ARG} \coloneqq \mathsf{Kilian}[\mathsf{PCP}, \mathsf{VC}]$ satisfies the following error bounds.

- *Soundness.*

$$\epsilon_{\mathsf{ARG}}(t_{\mathsf{ARG}}) \leq \epsilon_{\mathsf{PCP}} + \epsilon_{\mathsf{VC}}(t_{\mathsf{VC}}) + \epsilon \ , \text{ where } t_{\mathsf{VC}} = O\left(\frac{\ell}{\epsilon} \cdot t_{\mathsf{ARG}}\right) \ .$$

- *Knowledge soundness.*

$$\kappa_{\mathsf{ARG}}(t_{\mathsf{ARG}}) \leq \kappa_{\mathsf{PCP}} + \epsilon_{\mathsf{VC}}(t_{\mathsf{VC}}) + \epsilon \ , \text{ where } t_{\mathsf{VC}} = O\left(\frac{\ell}{\epsilon} \cdot t_{\mathsf{ARG}}\right) \text{ with extraction time } O\left(\frac{\ell}{\epsilon} \cdot t_{\mathsf{ARG}}\right) \ .$$

The additive error ϵ affects the time t_{VC} of the position-binding adversary (and extractor) multiplicatively as ℓ/ϵ; driving this error to negligible (for all polynomial-size adversaries) forces a *super-polynomial* rewinding budget.

Can this loss be avoided? There is partial evidence that suggests not: instantiating Kilian's protocol with a tailored PCP and a vector commitment based on Schnorr's identification scheme, [CDGSY24] showed that a sufficiently strong improvement to the argument's soundness bound would carry over to the standard-model security bound of Schnorr, which has resisted improvement for over two decades. This, however, is only a reduction to the *unproven* optimality of Schnorr's analysis.[1] We therefore ask:

Is the loss of ℓ/ϵ provably necessary for Kilian's protocol?

The expected-time setting. The connection to Schnorr's identification scheme suggests a potential way to sidestep the discussion above. Schnorr's canonical extractor rewinds only after a first accepting run, and then until it finds a second; its running time is therefore a random variable, and its guarantee is naturally stated in *expected* polynomial time. In the *expected-time setting*, all algorithms (adversary, extractor, and the reductions between them) are bounded in expected rather than worst-case running time, and the underlying cryptography is assumed secure against expected-time adversaries [JT20]. Analyzed this way, the Schnorr identification scheme loses essentially nothing relative to (the expected-time variant of the) discrete logarithm problem.

[1] The recent lower bounds for Schnorr against restricted reductions [HM26] do not apply; see Section 1.3.

There has been recent progress in analyzing the expected-time security of Kilian's protocol. [CDGSY24] gives an expected-time soundness analysis that replaces the ℓ/ϵ time overhead with $\ell \cdot \log(\mathsf{q}/\epsilon)$, where q is the query complexity of PCP; but the analysis does not extend to knowledge soundness. [AKLY24] obtains an essentially tight expected-time knowledge soundness bound via *adaptive special soundness*, but only for PCPs (that moreover satisfy a certain knowledge property); the analysis does not extend to the multi-round IOPs and functional IOPs on which deployed systems are built. This motivates our second question:

Can expected-time analysis achieve tight security at the full generality of the commit-and-open paradigm?

1.1 Our results

We answer both questions positively. First, the strict-time loss from rewinding is provably necessary: every natural strict polynomial-time extractor for Kilian's protocol, for *every* vector commitment scheme, incurs at least inverse polynomial knowledge soundness error. In fact, this lower bound holds for the commit-and-open paradigm in full generality (multiple rounds and functional queries).[2] Second, we obtain tight expected-time security for the commit-and-open paradigm, also in full generality.

For simplicity, we state the theorems below for the IBCS protocol, which compiles a public-coin IOP and a VC scheme into an interactive argument; later we discuss the extension to functional IOPs.

1.1.1 Strict-time barriers

We consider knowledge extractors that are *fully black-box* in the following sense: (i) there is a single "universal" extractor for all provers, running in *strict polynomial-time with an a priori rewinding budget* p (fixed independently of the prover); and (ii) *VC-oblivious*, i.e., the extractor accesses the prover only through an interface that returns, for any desired verifier challenge, the opened query-answer pairs and the bit denoting whether the opening proof is valid. For brevity we call such extractors *admissible*; every rewinding extractor we are aware of is admissible.

Theorem 1 (Informal)**.** *Let* IOP *be any IOP for an* NP *relation* R *with* $L(R) \notin \mathsf{BPP}$*, let* VC *be any vector commitment scheme, and set* $\mathsf{ARG} := \mathsf{IBCS}[\mathsf{IOP}, \mathsf{VC}]$*. Let* κ_{ARG} *be the non-adaptive knowledge soundness error of* ARG *against* $\mathsf{poly}(n)$*-time provers, with respect to any admissible extractor that rewinds the prover at most* $p = \mathsf{poly}(n)$ *times.*

- *If* IOP *has query complexity* $\mathsf{q} = O(\log n)$*, then* $\kappa_{\mathsf{ARG}} > \Theta(1/p)$*.*
- *If* IOP *is zero-knowledge, then* $\kappa_{\mathsf{ARG}} > \Theta(1/p)$*.*

The first bound holds for any IOP with small query complexity, while the second requires zero knowledge. Either way, no fixed polynomial budget achieves negligible knowledge soundness error: the strict-time extractor in [CDGSY24] loses roughly ℓ/p under a budget of p, and the $1/p$ dependence is necessary.

Remark 1. *Some restriction on how the extractor accesses the VC scheme in Theorem 1 (e.g., VC-obliviousness) is necessary. Indeed, if the VC scheme is* extractable *(e.g., Merkle commitments in the random oracle model [CY24] or various schemes proved under knowledge assumptions) then an extractor can obtain the committed IOP strings without any rewinding (and in strict polynomial time). A bound quantifying over* all *VC schemes must therefore exclude extractors that exploit a scheme's internals. Our barrier applies to every analysis that treats the VC scheme as a black-box position-binding primitive, which is every known standard-model analysis.*

[2] A lower bound for Kilian's protocol does not necessarily demonstrate a lower bound for commit-and-open arguments in general. For example, certain impossibility results only apply to constant-round interactive protocols [BL04], and hence (specific instantations of) Kilian's protocol. Yet, they say nothing about arguments with superconstant round complexity.

1.1.2 Expected-time security

We denote the IOP's round complexity as k, (total) proof length as ℓ, soundness error as ϵ_{IOP} and knowledge soundness error as κ_{IOP} (with associated extraction time $t_{\mathbf{E}}$); additionally, $\epsilon^{\star}_{\mathsf{VC}}(t^{\star}_{\mathsf{VC}})$ denotes the position binding error of the VC scheme against adversaries running in *expected-time* $t^{\star}_{\mathsf{VC}}$.

Theorem 2 (Informal)**.** $\mathsf{ARG} := \mathsf{IBCS}[\mathsf{IOP}, \mathsf{VC}]$ *has expected-time soundness error*

$$\epsilon^{\star}_{\mathsf{ARG}}(t^{\star}_{\mathsf{ARG}}) \leq \epsilon_{\mathsf{IOP}} + \epsilon^{\star}_{\mathsf{VC}}\big(t^{\star}_{\mathsf{VC}}\big) \ , \ \textit{where } t^{\star}_{\mathsf{VC}} = O\big(\ell \cdot t^{\star}_{\mathsf{ARG}}\big) \ ;$$

and expected-time knowledge soundness error

$$\kappa^{\star}_{\mathsf{ARG}}(t^{\star}_{\mathsf{ARG}}) \leq (1+\alpha) \cdot \kappa_{\mathsf{IOP}} + \epsilon^{\star}_{\mathsf{VC}}\big(t^{\star}_{\mathsf{VC}}\big) \ , \ \textit{where } t^{\star}_{\mathsf{VC}} = O\Big(\tfrac{\mathsf{k} \cdot \ell}{\log(1+\alpha)} \cdot t^{\star}_{\mathsf{ARG}}\Big)$$

and the extractor's expected running time is $t^{\star}_{\mathsf{VC}} + t_{\mathbf{E}}$.

On tightness of knowledge soundness. For a PCP ($\mathsf{k} = 1$), the extractor of Theorem 2 rewinds $O(\ell/\log(1+\alpha))$ times in expectation, compared to $O(\ell/\epsilon)$ rewinds of the strict-time extractor of [CDGSY24] at additive error ϵ. The dependence on ℓ is essentially unavoidable: we show that, for a suitable PCP, an expected-time VC-oblivious extractor rewinding $\ell/\mathsf{polylog}(\ell)$ times in expectation would yield an improved subexponential algorithm for $\mathsf{3SAT}$.[3] Hence the extractor in Theorem 2 seems optimal up to a polylogarithmic factor; eliminating this factor remains open, as does determining whether the cost is forced by VC-obliviousness.

We stress that this tightness holds for knowledge soundness *only*. Every known soundness analysis of the commit-and-open paradigm, including ours, proceeds by extracting a committed proof string and then invoking soundness of the underlying probabilistic proof; the reduction therefore pays the same $\Theta(\ell)$ rewinding cost as a knowledge extractor. Our lower bounds show that this cost is necessary for extraction, but not that extraction is necessary for soundness. Whether soundness can be established without extracting a full proof string, at a rewinding cost independent of ℓ, is an open problem that we leave for future work.

On functional IOPs. Many succinct arguments of interest are based on polynomial IOPs or other forms of *functional IOPs*; these yield distinct efficiency tradeoffs (e.g., between prover time and argument size). In a polynomial IOP, for example, the verifier treats the committed message as a polynomial and queries consist of desired evaluation points. The *Funky protocol* [CGKY26] captures the commit-and-open paradigm at this level of generality: a functional IOP and a functional commitment scheme both for the same family $\mathbf{Q}$ of query functions are combined into a succinct interactive argument. The *strict-time* security of the Funky protocol inherits the same rewinding overhead as Kilian's protocol [CGKY26], and we generalize Theorem 1 to establish a corresponding lower bound for knowledge soundness. We also establish tight *expected-time* security of the Funky protocol, generalizing Theorem 2. See Section 2 for details.

Remark 2 (Adaptive security)**.** *Our expected-time analyses are proved with adaptive security in the common reference string model, where a malicious argument prover chooses the instance after seeing the public parameters (as in [CDGSY24]); plain-model variants, where the verifier starts by sending public parameters, follow directly. Our barriers in Theorem 1 are for non-adaptive knowledge soundness, making the statements strictly stronger.*

1.2 The cost of rewinding

We discuss how the improvement on the number of rewindings in the expected-time regime affects concrete security. We instantiate the vector commitment scheme as a Merkle commitment scheme constructed from an ideal hash function with output length λ, and compare several *soundness* bounds of Kilian's protocol. Throughout,

[3] Recent AI-discovered results, still under review, claim subexponential algorithms for $\mathsf{3SAT}$ [Ope26; Cla26]. Our barrier would concern further improvements beyond these running time bounds (by choosing a larger polylogarithmic factor in the extractor's rewinding budget).

b denotes the target number of bits of security in the *cryptographic* contribution to the argument error; the information-theoretic term of the probabilistic proof (the PCP soundness error) is budgeted separately.

For an ideal hash function, a t-time adversary finds a collision with probability at most $t^2/2^\lambda$ (a birthday bound), whereas an adversary with *expected* running time $t^\star$ succeeds with probability at most $t^\star/2^{\lambda/2}$ [JT20]. This yields estimates for the position binding errors in the strict-time and expected-time settings:

$$\epsilon_{\mathsf{VC}}(\lambda, t_{\mathsf{VC}}) \leq \frac{t_{\mathsf{VC}}^2}{2^\lambda} \quad \text{and} \quad \epsilon_{\mathsf{VC}}^\star(\lambda, t_{\mathsf{VC}}^\star) \;\leq\; \frac{t_{\mathsf{VC}}^\star}{2^{\lambda/2}} \;\; .$$

We use these estimates to determine the minimal hash output length λ that yields b bits of cryptographic security in three cases for Kilian's protocol: the strict-time setting with [CDGSY24]'s bound, the expected-time setting with our bound in Theorem 2, and the straightline extraction setting in the random oracle model (ROM) [CY24].

- *Strict-time.* The position-binding adversary runs in time $t_{\mathsf{VC}} = O(\frac{\ell}{\epsilon} \cdot t_{\mathsf{ARG}})$, so the cryptographic contribution to the error is, for every tolerance $\epsilon \in (0, 1]$, $E(\epsilon) \coloneqq \frac{(4 \cdot \ell \cdot t_{\mathsf{ARG}})^2}{\epsilon^2 \cdot 2^\lambda} + \epsilon$, where the multiplicative factor 4 is the constant from the reduction in [CDGSY24]. Setting $\bar{\epsilon} \coloneqq (32 \cdot (\ell \cdot t_{\mathsf{ARG}})^2/2^\lambda)^{1/3}$ minimizes E, with value $E(\bar{\epsilon}) = \frac{3}{2}\bar{\epsilon}$. Requiring $E(\bar{\epsilon}) \leq 2^{-b}$ yields $\lambda \geq 3b + 2\log_2(\ell \cdot t_{\mathsf{ARG}}) + \log_2 108$.
- *Expected-time.* The position-binding adversary underlying Theorem 2 has expected running time $t_{\mathsf{VC}}^\star \leq (\ell+1) \cdot t_{\mathsf{ARG}}^\star$, so the error is at most $(\ell+1) \cdot t_{\mathsf{ARG}}^\star/2^{\lambda/2}$. This is at most 2^{-b} when $\lambda \geq 2b + 2\log_2((\ell+1) \cdot t_{\mathsf{ARG}}^\star)$.
- *Straightline in the ROM.* [CY24] shows that $\epsilon_{\mathsf{ARG}}(t_{\mathsf{ARG}}) \leq \epsilon_{\mathsf{PCP}} + t_{\mathsf{ARG}}^2/2^\lambda$, so the cryptographic term is at most 2^{-b} when $\lambda \geq b + 2\log_2 t_{\mathsf{ARG}}$.

Each requirement is affine in the target b with slope 3, 2, or 1: raising the security by one bit costs 3 extra bits of hash output under the strict-time analysis, 2 under the expected-time analysis, and 1 in the ROM. To see the improvement more concretely, we consider a target argument error of 2^{-40} against adversaries of size 2^{60}. Suppose that the underlying PCP has soundness error 2^{-42} and proof length 2^{30}. The table below collects the three requirements for the output length λ together with their values at this benchmark.

Analysis	required λ	at the benchmark
Strict-time [CDGSY24]	$3b + 2\log_2(\ell \cdot t_{\mathsf{ARG}}) + \log_2 108$	308
Expected-time (Theorem 2)	$2b + 2\log_2((\ell+1) \cdot t_{\mathsf{ARG}}^\star)$	261
Straightline in the ROM [CY24]	$b + 2\log_2 t_{\mathsf{ARG}}$	161

1.3 Related work

Strict vs. expected polynomial time. [BL04] proves that nontrivial constant-round zero-knowledge arguments admit neither strict-polynomial-time black-box simulators nor strict-polynomial-time black-box knowledge extractors. Theorem 1 adapts their techniques, but does not follow from their theorem: it applies to Kilian's protocol with an arbitrary vector commitment, in which case the argument need not be zero knowledge. Moreover, it extends to commit-and-open arguments with super-constant round complexity; see Remark 2.1 for a detailed comparison. Other studies of expected-time cryptography [Klo21; JT20; Wik24] do not study the commit-and-open paradigm.

Special soundness and IOPs. Special soundness of the commit-and-open compiler would immediately yield expected-time knowledge soundness, but Kilian's protocol is not k-special-sound for any polynomial k [CMSZ21]. *Generalized* special-soundness frameworks weaken this requirement and have been pushed toward IOPs: FRI [BBHR18] satisfies one [AFR23], with almost-optimal knowledge error but expected *quasi*-polynomial extraction time. None of them are known to yield an expected-polynomial-time knowledge extractor for the general commit-and-open paradigm.

Strict-time analyses of commit-and-open arguments. Kilian's original work [Kil92] gives no formal security analysis; later analyses cover increasingly general probabilistic proofs and commitment schemes: non-adaptive reverse-samplable PCPs [BG08], linear PCPs and linear commitments [LM19], the IOP-based compiler [CDGS23], and functional IOPs with functional commitments [CGKY26]; the same regime has also been analyzed against quantum adversaries, for PCP-based Kilian [CMSZ21] and for multi-round IOP-based arguments [CDDGS25]. All of them enforce a worst-case running-time bound and hence pay an additive loss in security. [CDGSY24] gives partial evidence that this loss is inherent, by reduction to the unproven optimality of Schnorr's standard-model analysis; the recent Schnorr lower bounds [HM26] cover only restricted, uniform-challenge reductions and thus do not apply. Our lower bounds instead make the loss precise for a natural class of black-box extractors that includes all extractors currently known in this line of work.

Expected-time analysis of Kilian's protocol. [CDGSY24] gives the first expected-time soundness analysis of Kilian's protocol, replacing the strict-time ℓ/ϵ rewinding overhead by $\ell \cdot \log(\mathsf{q}/\epsilon)$; it still pays the additive error ϵ and gives no knowledge soundness. [AKLY24] introduces adaptive special soundness (which is in turn inspired by almost special soundness [BF23]) and obtains the first lossless expected-time analysis of Kilian's protocol: it preserves the PCP knowledge error exactly, whereas our bound incurs a factor $1 + \alpha$. The two bounds are nevertheless *incomparable*, because theirs requires the PCP to be strongly extractable: a *deterministic* extractor $\mathbf{E}$ such that, for every instance $\mathbb{x}$ and proof string Π,

$$\Pr_{\rho}\left[\mathbf{V}^{\Pi}(\mathbb{x};\rho)=1\right] > \kappa_{\mathsf{PCP}} \implies (\mathbb{x}, \mathbf{E}(\mathbb{x},\Pi)) \in R \ .$$

We assume only the standard notion (Definition 3), in which the extractor may depend on the verifier randomness and only the joint failure probability is bounded. This is what enables our multi-round analysis: for IOPs the round-i proof $\Pi_i = \Pi_i(\rho_1, \ldots, \rho_{i-1})$ depends on earlier verifier randomness, so there is no verifier-independent tuple $(\Pi_1, \ldots, \Pi_{\mathsf{k}})$ on which deterministic per-string extraction can even be defined, and supplying the whole prover strategy would force exponential-time extraction. (Randomized extraction already arises at the PCP level under randomized code decoding [IJK09].) Generic composition of their extractor across rounds also blows up its running time superpolynomially. In sum, their exact knowledge term rests on a stronger, deterministic-extraction assumption, whereas our theorem uses the standard randomized notion, composes in expected polynomial time across rounds, and extends to functional IOPs.

Orthogonal models. Complementary lines of work escape the standard-model rewinding barrier by other means: straightline extraction from scheme-specific extractability in idealized models [CY24], and extractability assumptions for polynomial commitments in the standard model [CHMMVW20; BFS20]. Our lower bounds suggest this may be the only way out, since a black-box extractor must pay the rewinding cost (Remark 1). Finally, [CGKY26] treats strict-time Fiat–Shamir security of the Funky protocol, leaving the expected-time case open. Table 1 compares the standard-model rewinding analyses discussed above.

Analysis	Model	Security	Scope	Near-lossless?
[BG08]	strict-time	S, KS	PCP†	✗
[LM19]	strict-time	S, KS	PCP/LPCP	✗
[CDGS23]	strict-time	S, KS	IOP	✗
[CDGSY24]	strict-time	S, KS	PCP	✗
[CDGSY24]	expected-time	S	PCP	✗
[AKLY24]	expected-time	S, KS	PCP‡	✓
[CGKY26]	strict-time	$\mathsf{S}, \mathsf{KS}^{*}$	FIOP	✗
This work	expected-time	S, KS	FIOP	✓

† Non-adaptive, reverse-samplable PCPs.

‡ Strongly κ_{PCP}-extractable PCPs.

* State-restoration soundness and state-restoration knowledge soundness.

Table 1: Standard-model rewinding analyses of commit-and-open succinct arguments. S and KS denote soundness and knowledge soundness, and a bound is *near-lossless* if it has no additive truncation error.

2 Techniques

We overview the main ideas underlying our results. Section 2.1 explains our barriers for strict-time knowledge soundness. After that we discuss our expected-time analyses in increasing generality: first for Kilian's protocol in Section 2.2 (as a warmup); then for the IBCS protocol in Section 2.3 (which compiles IOPs instead of PCPs); and finally for the Funky protocol in Section 2.4 (which supports functional queries).

2.1 Strict-time barriers for knowledge soundness

We explain Theorem 1's lower bounds. For simplicity, we analyze Kilian's protocol and assume perfect completeness of the underlying probabilistic proofs and commitment schemes; the formal statements and proofs in Section 4 are for the Funky protocol and allow for imperfect completeness.

Definition 1. *An interactive argument* $\mathsf{ARG} = (\mathcal{P}, \mathcal{V})$ *for a relation* R *has* **non-adaptive knowledge soundness error** κ_{ARG} **with extraction time** $t_{\mathcal{E}}$ *if there exists a* $t_{\mathcal{E}}$*-time probabilistic algorithm* $\mathcal{E}$ *such that, for every instance* $\mathbb{x}$ *and time-bounded prover* $\widetilde{\mathcal{P}}$*,*

$$\Pr\left[\begin{array}{l|l} (\mathbb{x}, \mathbb{w}) \notin R & b \xleftarrow{\mathsf{tr}} \langle \widetilde{\mathcal{P}}, \mathcal{V}(\mathbb{x}) \rangle \\ \wedge\ b = 1 & \mathbb{w} \leftarrow \mathcal{E}^{\widetilde{\mathcal{P}}}(\mathbb{x}, \mathsf{tr}, b) \end{array}\right] \leq \kappa_{\mathsf{ARG}} \ .$$

Above, tr and b are the transcript and decision of the argument execution. The extractor $\mathcal{E}$ receives both, along with rewinding access to $\widetilde{\mathcal{P}}$.

Let R be the NP relation stated in Theorem 1. Suppose, toward a contradiction, that there is an admissible extractor (see Section 1.1.1) that makes at most p rewinds and has small knowledge soundness error. We show how to use this extractor to construct a BPP decider for the language $L(R)$, contradicting the assumption that $L(R) \notin \mathsf{BPP}$.

The main idea is to run the extractor against a prover that behaves like the honest argument prover except that, with probability $1 - \theta$, it refuses to answer the verifier's challenge. This aborting-prover strategy is due to Barak and Lindell [BL04]; we compare the two analyses at the end of this section. For properly chosen θ, we show that, with noticeable probability, the extractor sees only a single PCP view. Since this view can be simulated by a zero-knowledge simulator (or guessed if the PCP has small query complexity), we conclude that any sufficiently successful extractor can be used to construct a BPP decider for $L(R)$.

In more detail, fix an instance-witness pair $(\mathbb{x}, \tilde{\mathbb{w}}) \in R$ and parameter $\theta \in [0, 1]$. We define the malicious argument prover.

$\widetilde{\mathcal{P}}_{\theta, \mathbb{x}, \tilde{\mathbb{w}}}$:

1. Run the honest argument prover $(\mathsf{cm}, \mathsf{aux}) \leftarrow \mathcal{P}(\mathbb{x}, \tilde{\mathbb{w}})$. Send cm.
2. Receive the verifier's message ρ.
3. Flip a coin with heads probability θ:
   - If heads, send the honest response $\mathcal{P}(\mathsf{aux}, \rho)$.
   - If tails, send $\perp$.

Consider Definition 1's experiment with the above prover, and without loss of generality assume that the extractor rewinds p times. Observe that, with probability $\theta(1 - \theta)^p$, the prover behaves honestly in the initial execution but aborts in every one of the extractor's rewinds. We construct an algorithm which simulates the extractor's behavior *conditioned* on this event.

$A_{\theta, \mathbb{x}, \tilde{\mathbb{w}}}$:

1. Sample an honest argument execution $\langle \mathcal{P}(\mathbb{x}, \tilde{\mathbb{w}}), \mathcal{V}(\mathbb{x}) \rangle$; let tr be the resulting transcript; note that tr is accepting by completeness of PCP and VC.

2. Simulate $\mathcal{E}(\mathbb{x}, \mathsf{tr}, 1)$; whenever it rewinds with verifier message ρ, reply with $\perp$.
3. Output the witness produced by $\mathcal{E}$.

Observe that

$$\kappa_{\mathsf{ARG}} \geq \theta(1-\theta)^p \cdot \Pr_{\mathbb{w} \leftarrow A_{\theta,\mathbb{x},\tilde{\mathbb{w}}}}[(\mathbb{x}, \mathbb{w}) \notin R] \ ,$$

by completeness of PCP and VC. We now argue that the above algorithm does not need the hard-wired witness $\tilde{\mathbb{w}}$ when the underlying PCP satisfies certain properties:

- If PCP is honest-verifier zero-knowledge (with zero simulation error, for simplicity), then it suffices to give the extractor a simulated PCP verifier's view.
- If PCP has query complexity $\mathsf{q} = O(\log n)$, then we can efficiently enumerate over all possible views corresponding to a choice of PCP randomness.

Note that the algorithm can no longer runs the argument prover $\mathcal{P}$, and therefore cannot compute a proper commitment or opening proof. This is not a problem when the extractor $\mathcal{E}$ is *VC-oblivious*, as it does not use this data except to decide whether or not the opening proof is valid; see Definition 4.7 for a formal definition of VC-oblivious extractors.

To conclude the analysis, Bernoulli's inequality gives $\theta(1-\theta)^p \geq \theta(1-p\theta)$ for $\theta \in [0,1]$. Setting $\theta := 1/(2p)$ for $p \geq 1$, we obtain

$$\kappa_{\mathsf{ARG}} \geq \frac{1}{4p} \cdot \Pr_{\mathbb{w} \leftarrow A_{\theta,\mathbb{x}}}[(\mathbb{x}, \mathbb{w}) \notin R] \ ,$$

where $A_{\theta,\mathbb{x}}$ does not have the hard-wired witness as discussed above. Thus, if $\kappa_{\mathsf{ARG}} \leq 1/(5p)$, then $A_{\theta,\mathbb{x}}$ outputs a valid witness with probability at least $1/5$. Repeating the algorithm a constant number of times yields a BPP decider for $L(R)$.

Generalizing to the Funky protocol. The above analysis can be generalized to the Funky protocol in a straightforward manner. The main idea is to consider a prover that aborts (with probability $1-\theta$) in a *single round*: specifically, the round in which the prover should send FIOP answers. See Section 4 for details.

Remark 2.1 (Comparison with [BL04])**.** Barak and Lindell prove that no constant-round zero-knowledge argument of knowledge with negligible knowledge error for a language outside BPP has a strict polynomial-time black-box knowledge extractor. In their analysis, the malicious prover aborts with some (carefully chosen) probability in each round, such that the extractor only observes at most one interaction with noticeable probability; since the argument is zero-knowledge, this interaction simulatable. We take a similar approach, but our analysis differs in the following respects:

- *Any number of rounds.* Their lower bound degrades exponentially in the protocol's round complexity, and it is therefore restricted to constant-round protocols. Moreover, this restriction is inherent: with any super-constant number of rounds, sequential repetition of a zero-knowledge proof of knowledge yields strict polynomial-time black-box extraction [Gol01; BL04]. Our lower bound does not depend on the round complexity (and is therefore tighter, even for constant-round protocols) because we consider a malicious prover which only aborts in a single round. This suffices because the messages in other rounds only contain commitment data, and thus an oblivious extractor learns nothing from them alone.
- *Defining knowledge soundness.* Their lower bound is for a different variant of knowledge soundness, which is due to Bellare and Goldreich [BG93]. Roughly speaking, their definition requires the difference between the adversary and extractor's success probabilities to be at most κ_{ARG}; this is a weaker guarantee than Definition 1, which bounds the joint probability that the adversary succeeds and the extractor fails. We emphasize that these are both standard definitions which are satisfied by all known constructions. Our analysis can be adapted to work with either definition, but we observe that targeting Definition 1 has two advantages. First, we only require *honest-verifier* zero knowledge, whereas [BL04] requires malicious-verifier zero knowledge. Second, we achieve the tight lower bound $\kappa_{\mathsf{ARG}} \geq \Theta(1/p)$. Using the definition of [BG93], we would need to

bound the probability that any two rewinds of the extractor lead to accepting executions; this requires setting $\theta := \Theta(1/p^2)$ and leads to a looser bound $\kappa_{\mathsf{ARG}} \geq \Theta(1/p^2)$.

What about expected-time extractors? The analysis above exploits the worst-case nature of a strict rewinding budget: a strict extractor must obey the same p-rewind cap on every execution, including executions in which useful continuations occur only with probability θ. On the other hand, an expected-time extractor may instead enter a long rewinding phase only after an acceptance event of probability θ, and then spend $\Theta(1/\theta)$ rewinds on that branch: the conditional cost $1/\theta$ is canceled by the probability θ of entering the branch (an observation already made in [BL04, Section 5]). Expected-time analyses thus sidestep the lower bound, as we discuss next.

2.2 Expected-time security of Kilian's protocol

We discuss our expected-time soundness and knowledge soundness analyses for Kilian's protocol. The starting point is the expected-time soundness proof of [CDGSY24], which we recall in Section 2.2.1. In Section 2.2.2 we explain our soundness analysis, and in Section 2.2.3 we explain our knowledge soundness analysis.

2.2.1 Soundness analysis of [CDGSY24]

The analysis of [CDGSY24] formalizes the following intuition. First, the argument adversary can always be as successful as the best PCP adversary, since committing to a PCP string and honestly opening it is one available strategy. Second, if the argument adversary succeeds with probability higher than that, then it must be able to open the commitment inconsistently (with a similar advantage).

Optimal PCP string. Fix an instance $\mathbb{x} \notin L(R)$ and a malicious argument prover $\widetilde{\mathcal{P}}$ with expected running time $t^{\star}_{\mathsf{ARG}}$. For verifier randomness ρ, let $(\mathcal{Q}_\rho, \mathsf{ans}_\rho, \mathsf{pf}_\rho)$ denote the response given by $\widetilde{\mathcal{P}}(\mathsf{aux}, \rho)$, and let b_ρ denote the resulting decision bit of the Kilian verifier; we say that $(\mathcal{Q}_\rho, \mathsf{ans}_\rho, \mathsf{pf}_\rho)$ is *accepting* if $b_\rho = 1$. [CDGSY24] defines the "best" PCP string as follows. For every location $q \in [\ell]$ and symbol $\sigma \in \Sigma$, define

$$p(q, \sigma) := \Pr_{\rho \leftarrow \{0,1\}^{\mathsf{r}}}[q \in \mathcal{Q}_\rho \wedge \mathsf{ans}_\rho[q] = \sigma \wedge b_\rho = 1] \ .$$

The PCP string $\widetilde{\Pi}^{\star}$ is defined location-wise to be

$$\widetilde{\Pi}^{\star}[q] \in \arg\max_{\sigma \in \Sigma} p(q, \sigma) \ .$$

In other words, $\widetilde{\Pi}^{\star}[q]$ is the most likely value with which the malicious prover opens location q in an accepting execution. (If the prover never opens location q, then $\widetilde{\Pi}^{\star}[q]$ is defined to be the default symbol 0.) Note that $\widetilde{\Pi}^{\star}$ is not efficiently computable. Nevertheless, it is a well-defined PCP string.

By the law of total probability, the success probability of the argument adversary decomposes according to whether the PCP verifier, run with the fixed proof $\widetilde{\Pi}^{\star}$ on the same verifier challenge ρ, would also accept:

$$\Pr\left[\left\langle \widetilde{\mathcal{P}}, \mathcal{V}(\mathbb{x}) \right\rangle = 1\right] \leq \Pr\left[\mathbf{V}^{\widetilde{\Pi}^{\star}}(\mathbb{x}; \rho) = 1\right] + \Pr\left[\begin{array}{l} \left\langle \widetilde{\mathcal{P}}, \mathcal{V}(\mathbb{x}) \right\rangle = 1 \\ \wedge\ \mathbf{V}^{\widetilde{\Pi}^{\star}}(\mathbb{x}; \rho) = 0 \end{array}\right] \ . \tag{1}$$

The first term can be upper-bounded by ϵ_{PCP} according to PCP soundness.

VC adversary. It remains to bound the second term, where the argument verifier accepts but the PCP verifier rejects $\widetilde{\Pi}^{\star}$ on the same challenge. This event implies the existence of a query $q \in \mathcal{Q}_\rho$ such that $\mathsf{ans}_\rho[q] \neq \widetilde{\Pi}^{\star}[q]$. [CDGSY24] exploits this inconsistency to build the following position-binding adversary A_{VC}, which guesses the location of the disagreement.

$A_{\mathsf{VC}}(\mathbb{x})$:

1. Run the argument prover to get the commitment: $(\mathsf{cm}, \mathsf{aux}) \leftarrow \widetilde{\mathcal{P}}$.
2. Sample $\rho \leftarrow \{0,1\}^{\mathsf{r}}$ and compute $(\mathcal{Q}, \mathsf{ans}, \mathsf{pf}) \leftarrow \widetilde{\mathcal{P}}(\mathsf{aux}, \rho)$. If $b_\rho = 0$, output $\perp$.
3. Sample $i \leftarrow [|\mathcal{Q}|]$ uniformly and let q be the i-th location of $\mathcal{Q}$. Rewind the prover on uniformly sampled verifier randomness until M (to be chosen later) accepting openings whose query sets contain q are collected.
4. If some collected opening opens q to a value different from $\mathsf{ans}[q]$, output the two inconsistent openings; otherwise output $\perp$.

The second term of Equation 1 can be decomposed as

$$\begin{aligned}
\Pr\left[\begin{array}{l} \left\langle \widetilde{\mathcal{P}}, \mathcal{V}(\mathbb{x}) \right\rangle = 1 \\ \wedge\ \mathbf{V}^{\widetilde{\Pi}^\star}(\mathbb{x}; \rho) = 0 \end{array}\right] &\le \mathsf{q} \cdot \Pr\Big[A_{\mathsf{VC}} \text{ guesses } i \text{ such that } \mathsf{ans}[\mathcal{Q}[i]] \neq \widetilde{\Pi}^\star[\mathcal{Q}[i]]\Big] \\
&\le \mathsf{q} \cdot \Pr\left[\begin{array}{l} A_{\mathsf{VC}} \text{ guesses } i \text{ such that } \mathsf{ans}[\mathcal{Q}[i]] \neq \widetilde{\Pi}^\star[\mathcal{Q}[i]] \\ \wedge\ A_{\mathsf{VC}} \text{ outputs a binding violation} \end{array}\right] \\
&\quad + \mathsf{q} \cdot \Pr\left[\begin{array}{l} A_{\mathsf{VC}} \text{ guesses } i \text{ such that } \mathsf{ans}[\mathcal{Q}[i]] \neq \widetilde{\Pi}^\star[\mathcal{Q}[i]] \\ \wedge\ \text{all } M \text{ collected openings containing } \mathcal{Q}[i] \text{ open it to } \mathsf{ans}[\mathcal{Q}[i]] \end{array}\right] .
\end{aligned}$$

To see the first inequality, condition on the event on the left-hand side, so that a disagreeing location q exists; the probability of A_{VC} correctly guessing such a position is at least $1/\mathsf{q}$. The second inequality is a decomposition by total probability as follows. If one of the M collected openings opens q to a value different from $\mathsf{ans}[q]$, then A_{VC} outputs a binding violation; this gives the first term. Otherwise, they all open q to $\mathsf{ans}[q]$; this gives the second term.

To bound the first term, [CDGSY24] proves that A_{VC} rewinds at most $O(\ell M)$ times in expectation. Thus, assuming argument executions are the dominant cost,

$$\Pr\left[\begin{array}{l} A_{\mathsf{VC}} \text{ guesses } i \text{ such that } \mathsf{ans}[\mathcal{Q}[i]] \neq \widetilde{\Pi}^\star[\mathcal{Q}[i]] \\ \wedge\ A_{\mathsf{VC}} \text{ outputs a binding violation} \end{array}\right] \le \epsilon^\star_{\mathsf{VC}}(O(\ell M \cdot t^\star_{\mathsf{ARG}})) \ ,$$

where $\epsilon^\star_{\mathsf{VC}}(t^\star_{\mathsf{VC}})$ is the position-binding error of VC against adversaries of expected running time $t^\star_{\mathsf{VC}}$.

To bound the second term, observe that

$$\Pr\left[\begin{array}{l} A_{\mathsf{VC}} \text{ guesses } i \text{ such that } \mathsf{ans}[\mathcal{Q}[i]] \neq \widetilde{\Pi}^\star[\mathcal{Q}[i]] \\ \wedge\ \text{all } M \text{ collected openings containing } \mathcal{Q}[i] \text{ open it to } \mathsf{ans}[\mathcal{Q}[i]] \end{array}\right] \le 2^{-M} \ .$$

To see this, fix a queried location q and suppose that $\mathsf{ans}[q] \neq \widetilde{\Pi}^\star[q]$. The probability that an accepting opening opens q to $\mathsf{ans}[q]$ is at most $1/2$, since the distinct symbol $\widetilde{\Pi}^\star[q]$ is at least as likely. Hence, the probability that all M collected openings open q to $\mathsf{ans}[q]$ is at most 2^{-M}.

Final bound. Fix a parameter $\epsilon \in (0,1)$. Taking $M := \lceil \log_2(\mathsf{q}/\epsilon) \rceil$, [CDGSY24] concludes that

$$\Pr\Big[\left\langle \widetilde{\mathcal{P}}, \mathcal{V}(\mathbb{x}) \right\rangle = 1\Big] \le \epsilon_{\mathsf{PCP}} + \mathsf{q} \cdot \epsilon^\star_{\mathsf{VC}}(O(\ell \log(\mathsf{q}/\epsilon) \cdot t^\star_{\mathsf{ARG}})) + \epsilon \ . \tag{2}$$

The additive overhead. The above expected-time analysis of [CDGSY24] is not lossless: unlike the adaptive-special-soundness analysis of [AKLY24], it pays the additive error ϵ as shown in Equation 2. Intuitively, the error stems from the fact that A_{VC} never computes $\widetilde{\Pi}^\star[q]$, but rather approximates it from samples. Any finite number of samples is not guaranteed to contain $\widetilde{\Pi}^\star[q]$, and the reduction can only conclude that $\widetilde{\Pi}^\star[q]$ is missed with probability at most 2^{-M}; this is exactly the additive error ϵ.

2.2.2 Our lossless soundness analysis

Roughly speaking, we obtain Theorem 2's lossless bound by defining a *different* "best" PCP string and showing how to *exactly* compute its symbol at position q (in expected time), thereby avoiding any approximation error.

Optimal PCP string. We define $\widetilde{\Pi}$, which is not a fixed string but rather a *random variable*. Consider an experiment which rewinds $\widetilde{\mathcal{P}}$ on uniformly sampled verifier randomness indefinitely, keeping the accepting openings, and let $(\mathcal{Q}_i, \mathsf{ans}_i, \mathsf{pf}_i)$ be the i-th accepting opening:

$$(\mathcal{Q}_1, \mathsf{ans}_1, \mathsf{pf}_1),\ (\mathcal{Q}_2, \mathsf{ans}_2, \mathsf{pf}_2),\ \ldots\ .$$

Let $N_q \coloneqq \min\{i \geq 1 : q \in \mathcal{Q}_i\}$ be the index of the first opening that contains location q, and define the random PCP string $\widetilde{\Pi}$ by

$$\widetilde{\Pi}[q] \coloneqq \begin{cases} \mathsf{ans}_{N_q}[q] & \text{if } N_q < \infty \\ 0 & \text{otherwise.} \end{cases}$$

Although $\widetilde{\Pi}$ is random, PCP soundness still applies to it: the sequence of openings is sampled before, and independently of, the verifier randomness. Thus, averaging yields $\Pr\Big[\mathbf{V}^{\widetilde{\Pi}}(\mathbb{x}; \rho) = 1\Big] \leq \epsilon_{\mathsf{PCP}}$, and the decomposition in Equation 1 carries over with $\widetilde{\Pi}$ in place of $\widetilde{\Pi}^{\star}$:

$$\begin{aligned} \Pr\Big[\big\langle \widetilde{\mathcal{P}}, \mathcal{V}(\mathbb{x}) \big\rangle = 1\Big] &\leq \Pr\Big[\mathbf{V}^{\widetilde{\Pi}}(\mathbb{x}; \rho) = 1\Big] + \Pr\left[\begin{array}{l} \big\langle \widetilde{\mathcal{P}}, \mathcal{V}(\mathbb{x}) \big\rangle = 1 \\ \wedge\ \mathbf{V}^{\widetilde{\Pi}}(\mathbb{x}; \rho) = 0 \end{array}\right] \\ &\leq \epsilon_{\mathsf{PCP}} + \Pr\left[\begin{array}{l} \big\langle \widetilde{\mathcal{P}}, \mathcal{V}(\mathbb{x}) \big\rangle = 1 \\ \wedge\ \mathbf{V}^{\widetilde{\Pi}}(\mathbb{x}; \rho) = 0 \end{array}\right]. \end{aligned}$$

VC adversary. To bound the second term, we consider a new position-binding adversary A'_{VC}, which differs from Section 2.2.1's adversary in two ways. First, A'_{VC} does not have a cutoff: it continues rewinding until it finds an accepting opening whose query set contains q. Second, A'_{VC} in fact searches for all queries $q \in \mathcal{Q}$, instead of a random one. We present the construction below.

$A'_{\mathsf{VC}}(\mathbb{x})$:

1. Run the argument prover to get the commitment: $(\mathsf{cm}, \mathsf{aux}) \leftarrow \widetilde{\mathcal{P}}$.
2. Sample $\rho \leftarrow \{0,1\}^{\mathsf{r}}$ and compute $(\mathcal{Q}, \mathsf{ans}, \mathsf{pf}) \leftarrow \widetilde{\mathcal{P}}(\mathsf{aux}, \rho)$. If the Kilian verifier rejects, output $\perp$.
3. Rewind the prover on uniformly sampled verifier randomness, keeping the accepting openings, until every query in $\mathcal{Q}$ appears in some accepting opening.
4. If some collected opening opens a location $q \in \mathcal{Q}$ to a value different from $\mathsf{ans}[q]$, output the two inconsistent openings; otherwise output $\perp$.

For each $q \in \mathcal{Q}$, consider the *first* opening collected by A'_{VC} that contains q; let σ_q denote the symbol with which this opening opens q. The key fact which we leverage is that each σ_q is distributed as $\widetilde{\Pi}[q]$. Thus, the probability that some σ_q disagrees with $\mathsf{ans}[q]$ (which implies A'_{VC} outputs a binding violation) is exactly the probability that $\widetilde{\Pi}[\mathcal{Q}]$ disagrees with ans. We conclude that

$$\Pr\left[\begin{array}{l} \big\langle \widetilde{\mathcal{P}}, \mathcal{V}(\mathbb{x}) \big\rangle = 1 \\ \wedge\ \mathbf{V}^{\widetilde{\Pi}}(\mathbb{x}; \rho) = 0 \end{array}\right] \leq \Pr\big[A'_{\mathsf{VC}} \text{ outputs a binding violation}\big]\ .$$

Bounding the running time. Conditioned on the argument verifier accepting, the initial opening $(\mathcal{Q}, \mathsf{ans}, \mathsf{pf})$ is distributed identically to each rewound accepting opening. Thus, the probability that the first n rewound openings fail to *cover* $\mathcal{Q}$ (i.e., contain every query in $\mathcal{Q}$) equals the probability that the $(n+1)$-th opening is

new (i.e., reveals a location unseen so far). Summing over $n \in \mathbb{N}$, the expected number of accepting openings needed to cover $\mathcal{Q}$ equals the expected number of new openings. But there are at most ℓ new openings across the sequence (every new opening reveals at least one of the ℓ locations), so we deduce that A'_{VC} collects at most ℓ accepting openings in expectation. Each accepting opening in turn costs $1/\epsilon$ rewinds in expectation, where

$$\epsilon := \Pr_{\rho \leftarrow \{0,1\}^r}[b_\rho = 1] \ ,$$

denotes the prover's success probability. Thus, conditioned on Step 3 occurring, A'_{VC} makes at most ℓ/ϵ rewinds in expectation. But Step 3 only occurs with probability ϵ, so the unconditional expected number of rewinds is at most $\epsilon \cdot (\ell/\epsilon) = \ell$. The adversary A'_{VC} therefore runs in expected time $O(\ell \cdot t^\star_{\mathsf{ARG}})$. We conclude that

$$\Pr\left[\left\langle \widetilde{\mathcal{P}}, \mathcal{V}(\mathbb{x}) \right\rangle = 1\right] \leq \epsilon_{\mathsf{PCP}} + \epsilon^\star_{\mathsf{VC}}(O(\ell \cdot t^\star_{\mathsf{ARG}})) \ .$$

The above analysis does not suffice for knowledge soundness. One may expect our expected-time soundness analysis to carry over to expected-time knowledge soundness. However, difficulties arise when implementing the corresponding argument extractor. Since the soundness reduction only needs to determine whether one PCP verifier execution accepts, it suffices to obtain the local view $\widetilde{\Pi}[\mathcal{Q}]$ by sampling until $\mathcal{Q}$ is covered. In particular, values outside $\mathcal{Q}$ are never used. In contrast, the PCP extractor $\mathbf{E}$ must receive a *complete* PCP string, and may depend on locations that the PCP verifier did not query in its execution. A local view of $\widetilde{\Pi}$ is therefore insufficient.

Yet we also cannot simply reconstruct the entire $\widetilde{\Pi}$. An expected-time reduction can afford a long rewinding phase, but only when the cost of the phase is canceled by the probability of entering it, as in the analysis of A'_{VC} (where the ℓ/ϵ rewinds are paid for by the acceptance probability ϵ of the initial execution). It is not possible to reconstruct the entire proof in a similar fashion: revealing a location that appears in accepting openings with probability p takes $1/p$ accepting openings in expectation, and p can be arbitrarily small, with no event of matching probability to cancel this cost.

A tempting alternative is to follow the soundness reduction: sample ρ, sample openings until its query set is covered, and then fill all unopened locations with a default value. Unfortunately, the reconstructed proof (which is *not* equal to $\widetilde{\Pi}$) depends on ρ, and PCP knowledge soundness does not apply to PCP strings that depend on the verifier randomness.

In sum, a new analysis is needed to establish the knowledge soundness of Kilian's protocol.

2.2.3 Our knowledge soundness analysis

Our analysis draws inspiration from both of the soundness analyses presented above. From Section 2.2.1 we take the finite cutoff: fix a complete PCP string after finitely many rewinds. From Section 2.2.2 we take the formulation of $\widetilde{\Pi}$: define the PCP string according to the first answers seen across the sequence of rewound openings.

We first recall the definition of expected-time argument knowledge soundness. In contrast to the lower bounds, our upper bounds hold for *adaptive* knowledge soundness, where the malicious prover chooses the instance (see Remark 2).

Definition 2 (Adaptive expected-time knowledge soundness)**.** *An interactive argument* $\mathsf{ARG} = (\mathcal{P}, \mathcal{V})$ *for a relation* R *has* **adaptive expected-time knowledge soundness error** $\kappa^\star_{\mathsf{ARG}}$ **with extraction time** $t^\star_{\mathcal{E}}$ *if there exists an algorithm* $\mathcal{E}$ *with expected running time at most* $t^\star_{\mathcal{E}}$ *such that, for every algorithm* $\widetilde{\mathcal{P}}$ *with expected running time at most* $t^\star_{\mathsf{ARG}}$*,*

$$\Pr\left[\begin{array}{l|l} (\mathbb{x}, \mathbb{w}) \notin R & (\mathbb{x}, \mathsf{aux}) \leftarrow \widetilde{\mathcal{P}} \\ \wedge\ b = 1 & b \xleftarrow{\mathsf{tr}} \left\langle \widetilde{\mathcal{P}}(\mathsf{aux}), \mathcal{V}(\mathbb{x}) \right\rangle \\ & \mathbb{w} \leftarrow \mathcal{E}^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathbb{x}, \mathsf{tr}, b) \end{array}\right] \leq \kappa^\star_{\mathsf{ARG}} \ .$$

Above, the extractor receives rewinding access to $\widetilde{\mathcal{P}}(\mathsf{aux})$, where aux is the state saved just after the prover chooses $\mathbb{x}$.

Since there are various ways to define the PCP knowledge soundness (as noted in Section 1.3), we recall the standard definition to avoid confusion.

Definition 3 (PCP knowledge soundness)**.** $\mathsf{PCP} = (\mathbf{P}, \mathbf{V})$ *for a relation* R *has* **knowledge soundness error** κ_{PCP} **with extraction time** $t_{\mathbf{E}}$ *if there exists a* $t_{\mathbf{E}}$*-time probabilistic algorithm* $\mathbf{E}$ *such that, for every time-bounded prover* $\widetilde{\mathbf{P}}$*,*

$$\Pr\left[\begin{array}{l|l} (\mathbb{x},\mathbb{w}) \notin R & (\mathbb{x},\widetilde{\Pi}) \leftarrow \widetilde{\mathbf{P}} \\ \wedge\, b = 1 & \rho \leftarrow \{0,1\}^{\mathsf{r}} \\ & b := \mathbf{V}^{\widetilde{\Pi}}(\mathbb{x};\rho) \\ & \mathbb{w} \leftarrow \mathbf{E}(\mathbb{x},\widetilde{\Pi},\rho) \end{array}\right] \leq \kappa_{\mathsf{PCP}} \ .$$

Argument extractor. Our goal is to construct a knowledge extractor $\mathcal{E}$ for Kilian's protocol. The challenge is to efficiently sample a complete PCP string before the verifier randomness. To do this, our extractor follows the main ideas of the soundness analyses: sample a list T of M accepting openings, reconstruct a PCP string $\widetilde{\Pi}_T$ by taking the first answer seen at each location, and run the PCP extractor on $\widetilde{\Pi}_T$. Note that using a finite cutoff risks introducing an additive error, as seen in Section 2.2.1. To avoid this, our extractor carefully samples a final verifier randomness $\rho^\star$, which is passed into $\mathbf{E}$ alongside $\widetilde{\Pi}_T$, as follows. It continues rewinding until it finds an accepting opening $(\mathcal{Q}^\star, \mathsf{ans}^\star, \mathsf{pf}^\star)$ where $\mathcal{Q}^\star$ is *covered* by the previously sampled M openings; $\rho^\star$ is then the corresponding verifier randomness. We present the construction below.

$\mathcal{E}^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathbb{x}, \mathsf{tr}, b)$:

1. If $b = 0$, output $\perp$.
2. Read the commitment cm from tr. All subsequent rewinds return to the prover's state immediately after it sent this commitment.
3. Rewind the prover on uniformly sampled verifier randomness until M (to be chosen later) accepting transcripts $T = ((\rho_1, \mathcal{Q}_1, \mathsf{ans}_1, \mathsf{pf}_1), \ldots, (\rho_M, \mathcal{Q}_M, \mathsf{ans}_M, \mathsf{pf}_M))$ are collected.
4. Define the complete PCP string $\widetilde{\Pi}_T$ by taking the first answer seen at every location in $\mathcal{Q}_1 \cup \cdots \cup \mathcal{Q}_M$ and the default symbol elsewhere.
5. Rewind further until an accepting transcript $(\rho^\star, \mathcal{Q}^\star, \mathsf{ans}^\star, \mathsf{pf}^\star)$ with $\mathcal{Q}^\star \subseteq \mathcal{Q}_1 \cup \cdots \cup \mathcal{Q}_M$ is found.
6. Output $\mathbf{E}(\mathbb{x}, \widetilde{\Pi}_T, \rho^\star)$.

By the law of total probability, we can decompose the target probability as follows:

$$\Pr\left[\begin{array}{l} (\mathbb{x},\mathbb{w}) \notin R \\ \wedge\, b = 1 \end{array}\right] \leq \Pr\left[\begin{array}{l} (\mathbb{x}, \mathbf{E}(\mathbb{x}, \widetilde{\Pi}_T, \rho^\star)) \notin R \\ \wedge\, \mathbf{V}^{\widetilde{\Pi}_T}(\mathbb{x};\rho^\star) = 1 \end{array}\right] + \Pr\left[\mathsf{ans}^\star \neq \widetilde{\Pi}_T[\mathcal{Q}^\star]\right] \ . \tag{3}$$

Reducing to PCP knowledge soundness. At first glance, the first term of Equation 3 can be bounded by the PCP knowledge soundness error κ_{PCP}. This is not the case, however, since the verifier randomness $\rho^\star$ is *not* uniformly sampled.

Instead, our analysis proceeds as follows. First, condition on T in Step 3. We say that T *covers* ρ if $\mathcal{Q}_\rho \subseteq \mathcal{Q}_1 \cup \cdots \cup \mathcal{Q}_M$. Define

$$\delta := \Pr_\rho[T \text{ covers } \rho \mid b_\rho = 1] \ ,$$

so that a uniformly sampled ρ is accepting and covered with probability $\epsilon \cdot \delta$. Let B_T be the set of bad randomness for $\widetilde{\Pi}_T$: the randomness ρ on which the PCP verifier $\mathbf{V}^{\widetilde{\Pi}_T}(\mathbb{x};\rho)$ accepts but the PCP extractor $\mathbf{E}(\mathbb{x}, \widetilde{\Pi}_T, \rho)$ fails to output a valid witness; by PCP knowledge soundness, it holds that $\Pr_\rho[\rho \in B_T] \leq \kappa_{\mathsf{PCP}}$. Thus,

$$\Pr[\rho^\star \in B_T \mid T] = \frac{\Pr_\rho[\rho \in B_T \wedge T \text{ covers } \rho]}{\epsilon \cdot \delta} \leq \frac{\kappa_{\mathsf{PCP}}}{\epsilon \cdot \delta} \ .$$

Note that $\mathcal{E}$ only starts rewinding if the initial interaction accepts, i.e., $b = 1$, which happens with probability ϵ. Averaging over the M accepting openings,

$$\Pr\left[\begin{array}{l}(\mathbb{x}, \mathbf{E}(\mathbb{x}, \widetilde{\Pi}_T, \rho^\star)) \notin R \\ \wedge\ \mathbf{V}^{\widetilde{\Pi}_T}(\mathbb{x}; \rho^\star) = 1\end{array}\right] \leq \epsilon \cdot \mathbb{E}\left[\frac{\kappa_{\mathsf{PCP}}}{\epsilon \cdot \delta}\right] = \kappa_{\mathsf{PCP}} \cdot \mathbb{E}\left[\frac{1}{\delta}\right] \ .$$

It remains to bound $\mathbb{E}[1/\delta]$, which we call the *coverage factor*.

Bounding the coverage factor. For $W \subseteq [\ell]$, write p_W for the probability that a uniformly sampled accepting opening has its query set contained in W. In other words, $\delta = p_{\mathcal{Q}_1 \cup \cdots \cup \mathcal{Q}_M}$. For every $z \in (0, 1]$, a union bound over the possible covered domains gives

$$\Pr[\delta \leq z] \leq \sum_{\substack{W \subseteq [\ell]: \\ p_W \leq z}} \Pr[\mathcal{Q}_1 \cup \cdots \cup \mathcal{Q}_M = W] \leq \sum_{\substack{W \subseteq [\ell]: \\ p_W \leq z}} (p_W)^M \leq 2^\ell \cdot z^M \ ,$$

where the second inequality holds because $\mathcal{Q}_1 \cup \cdots \cup \mathcal{Q}_M = W$ requires all M independent query sets to land inside W, an event of probability $(p_W)^M$. Then,

$$\mathbb{E}[1/\delta] = \int_0^\infty \Pr[1/\delta \geq u]\,\mathrm{d}u \leq 2^{\ell/M} + \int_{2^{\ell/M}}^\infty 2^\ell \cdot u^{-M}\,\mathrm{d}u = \frac{M}{M-1} \cdot 2^{\ell/M} \ ,$$

where we bound $\Pr[1/\delta \geq u]$ by 1 for $u \leq 2^{\ell/M}$ and by $2^\ell \cdot u^{-M}$ beyond. Fix a parameter $\alpha > 0$. Setting $M := \lceil (\ell + 2)/\log(1+\alpha) \rceil$ makes the right-hand side at most $1 + \alpha$, so the first term of Equation 3 is at most $(1+\alpha) \cdot \kappa_{\mathsf{PCP}}$.[4]

VC adversary. To bound the second term of Equation 3, observe that, whenever the final answers $\mathsf{ans}^\star$ disagree with $\widetilde{\Pi}_T$ at a location of $\mathcal{Q}^\star$, the extractor has found a violation of the VC position-binding property. To bound the extractor's running time, observe the following:

- Conditioned on $b = 1$, Step 3 requires M/ϵ rewinds in expectation.
- Conditioned on $b = 1$ and T, Step 5 requires $1/(\epsilon \cdot \delta)$ rewinds in expectation.

It follows that the extractor needs at most

$$\epsilon \cdot \left(\frac{M}{\epsilon} + \frac{1}{\epsilon} \cdot \mathbb{E}[1/\delta]\right) = M + \mathbb{E}[1/\delta] \leq M + 1 + \alpha = O\left(\frac{\ell}{\log(1+\alpha)}\right)$$

rewinds in expectation, and the second term of Equation 3 is bounded by $\epsilon^\star_{\mathsf{VC}}(O(\ell/\log(1+\alpha) \cdot t^\star_{\mathsf{ARG}}))$.

Final bound. Altogether, for every constant $\alpha > 0$,

$$\kappa^\star_{\mathsf{ARG}} = (1+\alpha) \cdot \kappa_{\mathsf{PCP}} + \epsilon^\star_{\mathsf{VC}}\left(O\left(\frac{\ell}{\log(1+\alpha)} \cdot t^\star_{\mathsf{ARG}}\right)\right) \ ,$$

with the argument extractor running in expected time $O(\ell/\log(1+\alpha) \cdot t^\star_{\mathsf{ARG}}) + t_{\mathbf{E}}$.

Tightness of the expected-time knowledge soundness analysis. Our extractor rewinds the prover $O(\ell)$ times in expectation. Is this necessary? The lower bounds of Section 2.1 for strict-time knowledge extractors do not answer this question. Nevertheless, we show that $O(\ell)$ rewinds are at most a $\mathsf{polylog}(\ell)$ factor away from optimal, unless we obtain substantially faster subexponential-time algorithms for NP-hard problems (e.g., $\mathsf{3SAT}$) than currently known.

Consider a PCP with query complexity q, alphabet Σ, and proof length ℓ for an NP relation R. Suppose there exists a VC-oblivious extractor for Kilian's protocol with knowledge soundness error bounded away from 1 by a constant that runs in expected polynomial time and rewinds the prover $p^\star$ times in expectation. Then, we show

[4] In Lemma A.2 we prove the bound $\mathbb{E}[1/\delta] \leq 2^{\ell/M}$, which enables setting $M := \lceil \ell/\log(1+\alpha) \rceil$.

that there exists an algorithm that runs in strict time $\mathsf{poly}(n) \cdot 2^{O(p^\star \cdot \mathsf{q} \log|\Sigma|)}$ and decides $L(R)$ with probability at least $2/3$, where n is the instance size. Consider the PCP for 3SAT from [Din07], which has $\mathsf{q} = O(1)$, $|\Sigma| = O(1)$, and $\ell = n \cdot \mathsf{polylog}(n)$. An extractor that rewinds $p^\star = O(n/(\log n)^c)$ times in expectation would yield an algorithm that decides 3SAT in time $2^{O(n/(\log n)^c)}$; this is currently unknown for, say, $c = 2$.

The proof follows a strategy similar to that of our strict-time lower bound for low-query PCPs. The idea is to consider running the extractor against the honest prover, which receives a valid witness. Since the honest prover succeeds with probability 1, the extractor must output a valid witness with probability at least $1 - \kappa^\star_{\mathsf{ARG}}$. Assuming this is constant, Markov's inequality implies that truncating the execution after $O(p^\star)$ rewinds preserves constant success probability. Thus, we construct a decider which simulates the extractor's execution by enumerating over all possible views for each of the $O(p^\star)$ rewinds; the number of possible choices is bounded by $|\Sigma^{\mathsf{q}}|^{O(p^\star)} = 2^{O(p^\star \cdot \mathsf{q} \log|\Sigma|)}$. See Section 4.3 for the formal statement and proof.

2.3 Expected-time security of the IBCS protocol

We discuss our expected-time security analyses for the IBCS protocol, which extends Kilian's protocol to work with any public-coin IOP (a multiple-round PCP). In a k-round IOP, the prover sends a proof string Π_i and receives the verifier randomness ρ_i in each round $i \in [\mathsf{k}]$. The IBCS protocol transforms this into a k-round interactive argument where the prover instead sends a commitment cm_i to Π_i; after the last round, the prover opens the queried positions of every committed string.

2.3.1 Soundness

The analysis of [CDGSY24] (recalled in Section 2.2.1) directly extends to the IBCS protocol, still incurring the additive error ϵ and the factor q. Likewise, our new lossless analysis (overviewed in Section 2.2.2) directly extends to the IBCS protocol.

Indeed, given a malicious argument prover $\widetilde{\mathcal{P}}$ we define a "best" IOP strategy as follows. For each round i and verifier randomness prefix $\boldsymbol{\rho}_{<i}$, independently sample an infinite sequence of continuations of $\widetilde{\mathcal{P}}$ on uniformly sampled randomness for the remaining rounds, conditioned on the IBCS verifier accepting. Define the round-i proof string $\widetilde{\Pi}_i(\boldsymbol{\rho}_{<i})$ by taking, at each location, the first answer seen in the round-i openings of this sequence, and the default symbol if the location never appears.

The position-binding adversary first samples an execution of the IBCS protocol with $\widetilde{\mathcal{P}}$; let $(\rho_1, \ldots, \rho_{\mathsf{k}})$ be the execution's verifier randomness. If the verifier rejects, the adversary halts. Otherwise, for each round i, it rewinds from the prover's round-i state to reconstruct the view of $\widetilde{\Pi}_i(\boldsymbol{\rho}_{<i})$ on the locations queried in this execution. With a similar analysis as in Section 2.2.2, we show that the expected number of rewinds for the i-th round is the round-i proof length ℓ_i. The expected running time of the adversary is therefore $O(\ell \cdot t^\star_{\mathsf{ARG}})$, where $\ell = \sum_{i=1}^{\mathsf{k}} \ell_i$ is the IOP's total proof length. Thus, we obtain the expected-time soundness error bound

$$\epsilon^\star_{\mathsf{ARG}} \leq \epsilon_{\mathsf{IOP}} + \epsilon^\star_{\mathsf{VC}}(O(\ell \cdot t^\star_{\mathsf{ARG}})) \ .$$

2.3.2 Knowledge soundness

Generalizing our expected-time knowledge soundness to multiple rounds is less straightforward.

Prior art: trees of accepting transcripts. Previous rewinding extractors for multi-round protocols assemble a *tree of accepting transcripts*, i.e., a depth-k tree where:

- each i-th layer node is labeled with a prover message and has m_i children;
- each edge is labeled with verifier randomness;
- each path down the tree represents an accepting argument transcript.

Observe that such a tree has $\prod_i m_i$ leaves. In the IBCS protocol, the i-th round would seemingly require m_i to be on the order of ℓ_i. Hence, producing a tree of accepting transcripts would require on the order of $\prod_i \ell_i$ prover executions, which is prohibitively large. Indeed, the authors of [AKLY24], whose *adaptive special soundness*

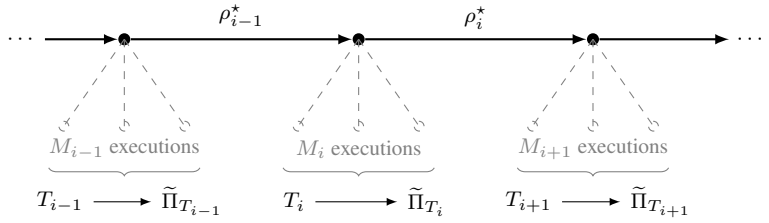

Figure 1: Visualization of the rewinding strategy in $\mathcal{E}$. The M_i executions sampled in Step 2b determine $\widetilde{\Pi}_{T_i}$ but are not used otherwise. After T_i is assembled, the execution sampled in Step 2d determines $\rho^\star_i$, which extends the main transcript.

framework works with such trees, are only able to give a one-round analysis of Kilian's protocol, leaving the multi-round IBCS analysis open. We remark that running time is not the only obstruction: as discussed in Section 1.3, their extractor also requires strong PCP knowledge soundness, which has no natural analogue for IOPs.

Our approach. Recall that Section 2.2.3's extractor performs two distinct steps: it samples a list T of M accepting openings, and then samples one more accepting opening, covered by T, to pick $\rho^\star$. At a high level, our extractor runs this two-step procedure once per round. At the start of round i, it has fixed the verifier randomness $(\rho^\star_1, \ldots, \rho^\star_{i-1})$ and obtained the lists $T_1, \ldots, T_{i-1}$. It proceeds by sampling M_i accepting executions to form T_i; crucially, each execution's queries to the first $i-1$ proof strings must be covered by the existing samples $T_1, \ldots, T_{i-1}$, respectively. It then samples one more accepting execution, whose queries to the first i proof strings are covered by $T_1, \ldots, T_i$, to pick $\rho^\star_i$. In this way the extractor ensures that the final IOP transcript $(\widetilde{\Pi}_{T_1}, \rho^\star_1, \ldots, \widetilde{\Pi}_{T_\mathsf{k}}, \rho^\star_\mathsf{k})$ is accepting. We present the construction below.

$\mathcal{E}^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathbb{x}, \mathsf{tr}, b)$:

1. If $b = 0$, output $\perp$.
2. For $i = 1, \ldots, \mathsf{k}$:
   (a) Initialize an empty list T_i.
   (b) Repeat M_i (to be chosen later) times: rewind the prover after the prefix $(\rho^\star_1, \ldots, \rho^\star_{i-1})$ and sample uniformly from all verifier randomness $(\rho'_i, \ldots, \rho'_\mathsf{k})$ that satisfies the following:
   – $\langle \widetilde{\mathcal{P}}(\mathsf{aux}), \mathcal{V}(\mathbb{x}; \rho^\star_1, \ldots, \rho^\star_{i-1}, \rho'_i, \ldots, \rho'_\mathsf{k}) \rangle = 1$; and
   – for every $j < i$, the round-j opening is covered by T_j.
   Append the round-i opening to T_i.
   (c) Define the round-i proof $\widetilde{\Pi}_{T_i}$ by taking the first answer seen in T_i at every covered location and the default symbol elsewhere.
   (d) Sample $(\rho'_i, \ldots, \rho'_\mathsf{k})$ as in Step 2b, with the additional requirement that the round-i opening is covered by T_i. Fix the round-i verifier randomness as $\rho^\star_i := \rho'_i$.
3. Output $\mathbf{E}(\mathbb{x}, \widetilde{\Pi}_{T_1}, \rho^\star_1, \ldots, \widetilde{\Pi}_{T_\mathsf{k}}, \rho^\star_\mathsf{k})$.

Our extractor does not follow the "tree of accepting transcripts" approach outlined earlier. Rather, it aims to assemble a "path of accepting transcripts" (Figure 1): once Step 2d fixes $\rho^\star_i$, every later sample extends the prefix $(\rho^\star_1, \ldots, \rho^\star_i)$. This enables reconstructing a single (accepting) IOP transcript, while keeping the total cost small (roughly a sum over ℓ_i).

Why does a path suffice? Essentially it is because the standard definition of IOP knowledge soundness, presented below, only gives the IOP extractor a single transcript.

Definition 4 (IOP knowledge soundness)**.** $\mathsf{IOP} = (\mathbf{P}, \mathbf{V})$ *for a relation* R *has* **knowledge soundness error** κ_{IOP}

with extraction time $t_{\mathbf{E}}$ *if there exists a* $t_{\mathbf{E}}$*-time algorithm* $\mathbf{E}$ *such that, for every prover* $\widetilde{\mathbf{P}}$*,*

$$\Pr\left[\begin{array}{l|l} & (\mathbb{x}, \widetilde{\Pi}_1, \mathbf{aux}_1) \leftarrow \widetilde{\mathbf{P}} \\ (\mathbb{x}, \mathbb{w}) \notin R & \textit{for } i = 1, \ldots, \mathsf{k}-1\textit{:} \\ \wedge\, b = 1 & \quad \rho_i \leftarrow \{0,1\}^{\mathsf{r}_i} \\ & \quad (\widetilde{\Pi}_{i+1}, \mathbf{aux}_{i+1}) \leftarrow \widetilde{\mathbf{P}}(\mathbf{aux}_i, \rho_i) \\ & \rho_{\mathsf{k}} \leftarrow \{0,1\}^{\mathsf{r}_{\mathsf{k}}} \\ & b := \mathbf{V}^{\widetilde{\Pi}_1, \ldots, \widetilde{\Pi}_{\mathsf{k}}}(\mathbb{x}; \rho_1, \ldots, \rho_{\mathsf{k}}) \\ & \mathbb{w} \leftarrow \mathbf{E}(\mathbb{x}, \widetilde{\Pi}_1, \rho_1, \ldots, \widetilde{\Pi}_{\mathsf{k}}, \rho_{\mathsf{k}}) \end{array}\right] \leq \kappa_{\mathsf{IOP}} \ .$$

We now bound the adversary's success probability. As in Section 2.2.3, we have the following decomposition:

$$\begin{aligned} \Pr\left[\begin{array}{l} (\mathbb{x}, \mathbb{w}) \notin R \\ \wedge\, b = 1 \end{array}\right] \leq \Pr\left[\begin{array}{l} (\mathbb{x}, \mathbb{w}) \notin R \\ \wedge\, \mathbf{V}^{\widetilde{\Pi}_{T_1}, \ldots, \widetilde{\Pi}_{T_{\mathsf{k}}}}(\mathbb{x}; \rho_1^\star, \ldots, \rho_{\mathsf{k}}^\star) = 1 \end{array}\right] \\ + \Pr\left[\begin{array}{l} \text{an accepting opening sampled by the extractor disagrees} \\ \text{with some } \widetilde{\Pi}_{T_i} \text{ at a location covered by } T_i \end{array}\right] . \end{aligned} \tag{4}$$

The second term is bounded by the position-binding error of VC against adversaries with the expected running time of the extractor, as before. It remains to bound the first term.

Reducing to IOP knowledge soundness. As in Section 2.2.3, the first term of Equation 4 is not necessarily bounded by κ_{IOP}, since the verifier randomness $\rho_1^\star, \ldots, \rho_{\mathsf{k}}^\star$ is not uniformly distributed.

Instead, our analysis proceeds as follows. Define

$$\begin{aligned} \kappa_{\mathsf{k}}(\Pi_1, \rho_1, \ldots, \Pi_{\mathsf{k}}, \rho_{\mathsf{k}}) &:= \mathbb{1}\left[\begin{array}{l} \mathbf{V}^{\Pi_1, \ldots, \Pi_{\mathsf{k}}}(\mathbb{x}; \rho_1, \ldots, \rho_{\mathsf{k}}) = 1 \\ \wedge\, (\mathbb{x}, \mathbf{E}(\mathbb{x}, \Pi_1, \rho_1, \ldots, \Pi_{\mathsf{k}}, \rho_{\mathsf{k}})) \notin R \end{array}\right] , \\ \kappa_{i-1}(\Pi_1, \rho_1, \ldots, \Pi_{i-1}, \rho_{i-1}) &:= \max_{\Pi_i \in \Sigma^{\ell_i}} \mathop{\mathbb{E}}_{\rho_i \leftarrow \{0,1\}^{\mathsf{r}_i}} [\kappa_i(\Pi_1, \rho_1, \ldots, \Pi_i, \rho_i)] \ . \end{aligned}$$

In other words, $\kappa_i(\Pi_1, \rho_1, \ldots, \Pi_i, \rho_i)$ is the largest probability of accepting without extracting a witness that an IOP prover can achieve from a fixed partial transcript $(\Pi_1, \rho_1, \ldots, \Pi_i, \rho_i)$. By IOP knowledge soundness, it holds that $\kappa_0 \leq \kappa_{\mathsf{IOP}}$. Conditioned on $b = 1$, the event in the first term of Equation 4 occurs with probability $\mathbb{E}\Big[\kappa_{\mathsf{k}}(\widetilde{\Pi}_{T_1}, \rho_1^\star, \ldots, \widetilde{\Pi}_{T_{\mathsf{k}}}, \rho_{\mathsf{k}}^\star)\Big]$. It remains to relate this expectation to κ_0.

Fix the instance, first commitment, and corresponding prover state $(\mathbb{x}, \mathsf{cm}_1, \mathsf{aux}_1)$ from the initial interaction and, for round i, condition on the verifier randomness $\rho_1^\star, \ldots, \rho_{i-1}^\star$ fixed in the earlier rounds and on $T_1, \ldots, T_i$. Let ϵ_i be the probability that uniformly sampled verifier randomness $(\rho_i', \ldots, \rho_{\mathsf{k}}')$ satisfies the two conditions of Step 2b in $\mathcal{E}$: (i) the corresponding execution of the argument accepts; and (ii) for every $j < i$, the round-j opening is covered by T_j. Let δ_i be the probability, conditioned on these two conditions, that the round-i opening is covered by T_i, as required in Step 2d of $\mathcal{E}$. As in the analysis for Kilian's protocol, the verifier randomness $\rho_i^\star$ is not uniformly sampled; in particular, for every $\rho_i \in \{0,1\}^{\mathsf{r}_i}$, the probability that $\rho_i^\star = \rho_i$ is proportional to the probability that ρ_i leads to an accepting covered continuation (Figure 2):

$$\Pr[\rho_i^\star = \rho_i] = \frac{\epsilon_{i+1}(\rho_i)}{2^{\mathsf{r}_i} \cdot \epsilon_i \cdot \delta_i} \ ,$$

where $\epsilon_{i+1}(\rho_i)$ is ϵ_{i+1} with the round-i verifier randomness set to ρ_i: the probability that, given the prefix $(\rho_1^\star, \ldots, \rho_{i-1}^\star, \rho_i)$, uniformly sampled verifier randomness for the later rounds leads to an accepting execution whose round-j opening is covered by T_j for every $j \leq i$. Consequently, suppressing the transcript prefix $(\widetilde{\Pi}_{T_1}, \rho_1^\star, \ldots, \widetilde{\Pi}_{T_{i-1}}, \rho_{i-1}^\star)$ from the arguments of κ_{i-1} and κ_i,

$$\mathop{\mathbb{E}}_{\rho_i^\star}\left[\frac{\kappa_i(\widetilde{\Pi}_{T_i}, \rho_i^\star)}{\epsilon_{i+1}(\rho_i^\star)}\right] \leq \frac{\mathbb{E}_{\rho_i \leftarrow \{0,1\}^{\mathsf{r}_i}}\Big[\kappa_i(\widetilde{\Pi}_{T_i}, \rho_i)\Big]}{\epsilon_i \cdot \delta_i} \leq \frac{\kappa_{i-1}}{\epsilon_i \cdot \delta_i} \ . \tag{5}$$

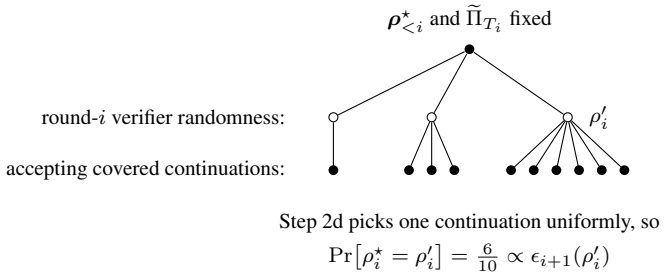

Figure 2: The distribution of $\rho^\star_i$ in a toy example: the accepting covered continuations of the prefix $\boldsymbol{\rho}^\star_{<i}$, grouped by their round-i verifier randomness; the marked value ρ'_i has six of the ten continuations. Step 2d picks one continuation uniformly, so each value is selected with probability proportional to its number of continuations, that is, proportional to ϵ_{i+1}.

Let

$$C_i \coloneqq \frac{M_i}{M_i - 1} \cdot 2^{\ell_i / M_i} \geq \mathbb{E}[1/\delta_i] \ ,$$

where the inequality follows from the calculation in Section 2.2.3.

We apply Equation 5 from round k down to round 1. Observe that below, $\epsilon_{\mathsf{k}+1} = 1$ because the execution sampled in Step 2d at round k accepts and its round-j opening is covered by T_j for every $j \leq \mathsf{k}$. Therefore, conditioned on $b = 1$, using Equation 5, we have

$$\begin{aligned}
\mathbb{E}\Big[\kappa_{\mathsf{k}}(\widetilde{\Pi}_{T_1}, \rho^\star_1, \ldots, \widetilde{\Pi}_{T_{\mathsf{k}}}, \rho^\star_{\mathsf{k}})\Big] &= \mathbb{E}\left[\frac{\kappa_{\mathsf{k}}(\widetilde{\Pi}_{T_1}, \rho^\star_1, \ldots, \widetilde{\Pi}_{T_{\mathsf{k}}}, \rho^\star_{\mathsf{k}})}{\epsilon_{\mathsf{k}+1}}\right] \\
&\leq C_{\mathsf{k}} \cdot \mathbb{E}\left[\frac{\kappa_{\mathsf{k}-1}(\widetilde{\Pi}_{T_1}, \rho^\star_1, \ldots, \widetilde{\Pi}_{T_{\mathsf{k}-1}}, \rho^\star_{\mathsf{k}-1})}{\epsilon_{\mathsf{k}}}\right] \\
&\leq C_{\mathsf{k}} \cdot C_{\mathsf{k}-1} \cdot \mathbb{E}\left[\frac{\kappa_{\mathsf{k}-2}(\widetilde{\Pi}_{T_1}, \rho^\star_1, \ldots, \widetilde{\Pi}_{T_{\mathsf{k}-2}}, \rho^\star_{\mathsf{k}-2})}{\epsilon_{\mathsf{k}-1}}\right] \\
&\leq \cdots \\
&\leq \Big(\prod_{i=1}^{\mathsf{k}} C_i\Big) \cdot \frac{\kappa_0}{\epsilon_1} \ .
\end{aligned}$$

Finally, ϵ_1 is exactly the probability that the initial interaction accepts. Since the bound above is conditioned on this event, the first term of Equation 4 equals

$$\epsilon_1 \cdot \mathbb{E}\Big[\kappa_{\mathsf{k}}(\widetilde{\Pi}_{T_1}, \rho^\star_1, \ldots, \widetilde{\Pi}_{T_{\mathsf{k}}}, \rho^\star_{\mathsf{k}})\Big] \leq \epsilon_1 \cdot \Big(\prod_{i=1}^{\mathsf{k}} C_i\Big) \cdot \frac{\kappa_0}{\epsilon_1} \leq \Big(\prod_{i=1}^{\mathsf{k}} C_i\Big) \cdot \kappa_{\mathsf{IOP}} \ .$$

Note that C_i decreases to 1 as M_i grows. We set $M_i \coloneqq \lceil \mathsf{k} \cdot (\ell_i + 2) / \log_2(1+\alpha) \rceil$, which gives $C_i \leq (1+\alpha)^{1/\mathsf{k}}$ and hence $\prod_i C_i \leq 1 + \alpha$, independent of the round complexity.

Bounding the running time. The same calculation as in Equation 5 applies to any nonnegative functions $F_0, \ldots, F_{\mathsf{k}}$ satisfying $\mathbb{E}_{\rho_i}[F_i] \leq F_{i-1}$ for uniformly sampled ρ_i. Under the same conditioning on the preceding history and T_i, we obtain

$$\mathop{\mathbb{E}}_{\rho^\star_i}\left[\frac{F_i(\rho^\star_i)}{\epsilon_{i+1}(\rho^\star_i)}\right] \leq \frac{\mathbb{E}_{\rho_i \leftarrow \{0,1\}^{\mathsf{r}_i}}[F_i(\rho_i)]}{\epsilon_i \cdot \delta_i} \leq \frac{F_{i-1}}{\epsilon_i \cdot \delta_i} \ . \tag{6}$$

We now take F_i to be the prover's conditional expected running time given the round-i prefix, so that $F_0 = t^\star_{\mathsf{ARG}}$. Conditioned on $b = 1$, each entry of T_i costs expected time F_{i-1}/ϵ_i by rejection sampling, and the sample of

Step 2d costs $F_{i-1}/(\epsilon_i \cdot \delta_i)$. Applying Equation 6 as before,

$$\mathbb{E}\left[\frac{F_{i-1}}{\epsilon_i}\right] \leq C_{i-1} \cdot \mathbb{E}\left[\frac{F_{i-2}}{\epsilon_{i-1}}\right] \leq \cdots \leq \Big(\prod_{j<i} C_j\Big) \cdot \frac{t^\star_{\mathsf{ARG}}}{\epsilon_1} \ ,$$

and averaging over T_i bounds the cost of the sample of Step 2d by $C_i \cdot \mathbb{E}[F_{i-1}/\epsilon_i]$. Summing over the rounds, the expected running time conditioned on $b = 1$ is at most

$$\sum_{i=1}^{\mathsf{k}} (M_i + C_i) \cdot \Big(\prod_{j<i} C_j\Big) \cdot \frac{t^\star_{\mathsf{ARG}}}{\epsilon_1} \leq (1+\alpha) \cdot \Big(\sum_{i=1}^{\mathsf{k}} (M_i + C_i)\Big) \cdot \frac{t^\star_{\mathsf{ARG}}}{\epsilon_1} \ .$$

Multiplying by the success probability ϵ_1 cancels the $1/\epsilon_1$, and $\sum_{i=1}^{\mathsf{k}}(M_i + C_i) = O(\mathsf{k} \cdot \ell/\log(1+\alpha))$ by our choice of M_i, so unconditionally the extractor spends expected time $O(\mathsf{k} \cdot \ell/\log(1+\alpha) \cdot t^\star_{\mathsf{ARG}})$ running the prover.

Final bound. Altogether, for every constant $\alpha > 0$,

$$\kappa^\star_{\mathsf{ARG}} \leq (1+\alpha) \cdot \kappa_{\mathsf{IOP}} + \epsilon^\star_{\mathsf{VC}}\left(O\left(\frac{\mathsf{k} \cdot \ell}{\log(1+\alpha)} \cdot t^\star_{\mathsf{ARG}}\right)\right) \ ,$$

with the argument extractor running in expected time $O(\mathsf{k} \cdot \ell/\log(1+\alpha) \cdot t^\star_{\mathsf{ARG}}) + t_{\mathbf{E}}$.

2.4 Expected-time security of the Funky protocol

We discuss how our expected-time analyses extend to the Funky protocol [CGKY26], which combines a functional IOP (FIOP) with a functional commitment scheme. In an FIOP, the verifier reads the round-i committed string Π_i only through a class $\mathbf{Q}$ of query functions. Thus, an opening no longer assigns symbols to locations; rather, it induces *constraints* (α, β) such that the committed string Π satisfies $\alpha(\Pi) = \beta$.

In Section 2.2.2, the sampled openings defined the proof through the first answer seen at every covered location; here, to construct the best FIOP string $\Pi^\star_i$ from the openings sampled in round i, process their constraints in order, append each constraint not already determined by the previous accepting openings, and let $\Pi^\star_i$ be any string that satisfies the appended constraints. This is the functional analogue of taking the first answer seen at every covered location.

For point queries, a contradiction among collected constraints is two valid openings that disagree at a location, which implies a position-binding violation. For general query classes, a contradiction means that no single string satisfies the constraints of all the collected openings, even though every one of them was accepted with respect to the same commitment; this constitutes a violation of the *function binding* property of the functional commitment (Definition 3.18).

Recall from Section 2.2.2 that the expected number of rewinds for Kilian's protocol is the proof length ℓ, because only ℓ openings can ever reveal a new location. For a general query class, our construction of the FIOP strings gives the same bound with the same proof: each appended constraint strictly shrinks a nonempty solution space, so the number of appends is at most the length of the longest strictly decreasing chain of solution spaces, the diameter $d_{\mathbf{Q}}$ of the class (Definition 3.12). (For example, for linear queries the appended constraints form a row basis and the diameter is again the proof length.)

Let $d := \sum_{i=1}^{\mathsf{k}} d_{\mathbf{Q}}(\ell_i)$ be the sum of diameters from each round's query class. We obtain, for every constant $\alpha > 0$,

$$\epsilon^\star_{\mathsf{ARG}} \leq \epsilon_{\mathsf{FIOP}} + \epsilon^\star_{\mathsf{FC}}(O(d \cdot t^\star_{\mathsf{ARG}})) \ ,$$
$$\kappa^\star_{\mathsf{ARG}} \leq (1+\alpha) \cdot \kappa_{\mathsf{FIOP}} + \epsilon^\star_{\mathsf{FC}}(O_\alpha(\mathsf{k} \cdot d \cdot t^\star_{\mathsf{ARG}})) \ ,$$

with the argument extractor running in expected time $O_\alpha(\mathsf{k} \cdot d \cdot t^\star_{\mathsf{ARG}}) + t_{\mathbf{E}}$. For point queries the diameter equals the proof length and we recover the bounds in Theorem 2. The formal theorems and proofs are in Sections 5 and 6.

3 Preliminaries

We define the interactive argument security notions used throughout the paper and the functional commitments and oracle proofs used in the expected-time analyses, and then recall the Funky compiler that combines them. The vector commitment definitions and the IOP notions used only for the lower bounds, which specialize the functional commitment and oracle proof definitions, are stated in Section 4.

Definition 3.1. *A* **relation** R *is a set of pairs* $(\mathbb{x}, \mathbb{w})$ *where* $\mathbb{x}$ *is an instance and* $\mathbb{w}$ *a witness. The corresponding* **language** $L(R)$ *is the set of instances* $\mathbb{x}$ *for which there exists a witness* $\mathbb{w}$ *such that* $(\mathbb{x}, \mathbb{w}) \in R$*.*

3.1 Interactive arguments

An *interactive argument* (in the common reference string model) for a relation R is a tuple of polynomial-time algorithms $\mathsf{ARG} = (\mathcal{G}, \mathcal{P}, \mathcal{V})$ that satisfies the following properties.

Definition 3.2 (Completeness)**.** $\mathsf{ARG} = (\mathcal{G}, \mathcal{P}, \mathcal{V})$ *for a relation* R *has* **completeness error** γ_{ARG} *if for every security parameter* $\lambda \in \mathbb{N}$*, instance size bound* $n \in \mathbb{N}$*, and instance-witness pair* $(\mathbb{x}, \mathbb{w}) \in R$ *with* $|\mathbb{x}| \leq n$*,*

$$\Pr\left[\left\langle \mathcal{P}(\mathsf{pp}, \mathbb{x}, \mathbb{w}), \mathcal{V}(\mathsf{pp}, \mathbb{x})\right\rangle = 1 \;\middle|\; \mathsf{pp} \leftarrow \mathcal{G}(1^\lambda, n)\right] \geq 1 - \gamma_{\mathsf{ARG}}(\lambda, n) \enspace .$$

It has **perfect completeness** *if* $\gamma_{\mathsf{ARG}} = 0$*.*

Definition 3.3 (Adaptive soundness)**.** $\mathsf{ARG} = (\mathcal{G}, \mathcal{P}, \mathcal{V})$ *for a relation* R *has* **(adaptive strict-time) soundness error** ϵ_{ARG} *if for every security parameter* $\lambda \in \mathbb{N}$*, instance size bound* $n \in \mathbb{N}$*, auxiliary input distribution* $\mathcal{D}$*, adversary time bound* $t_{\mathsf{ARG}} \in \mathbb{N}$*, and* t_{ARG}*-time algorithm* $\widetilde{\mathcal{P}}$*,*

$$\Pr\left[\begin{array}{l|l} |\mathbb{x}| \leq n & \mathsf{pp} \leftarrow \mathcal{G}(1^\lambda, n) \\ \wedge\, \mathbb{x} \notin L(R) & \eta \leftarrow \mathcal{D} \\ \wedge\, b = 1 & (\mathbb{x}, \mathsf{aux}) \leftarrow \widetilde{\mathcal{P}}(\mathsf{pp}, \eta) \\ & b \leftarrow \left\langle \widetilde{\mathcal{P}}(\mathsf{aux}), \mathcal{V}(\mathsf{pp}, \mathbb{x})\right\rangle \end{array}\right] \leq \epsilon_{\mathsf{ARG}}(\lambda, n, t_{\mathsf{ARG}}) \enspace .$$

Definition 3.4 (Adaptive expected-time soundness)**.** $\mathsf{ARG} = (\mathcal{G}, \mathcal{P}, \mathcal{V})$ *for a relation* R *has* **(adaptive) expected-time soundness error** $\epsilon^\star_{\mathsf{ARG}}$ *if for every security parameter* $\lambda \in \mathbb{N}$*, instance size bound* $n \in \mathbb{N}$*, auxiliary input distribution* $\mathcal{D}$*, adversary expected-time bound* $t^\star_{\mathsf{ARG}} \in \mathbb{N}$*, and algorithm* $\widetilde{\mathcal{P}}$ *with expected running time at most* $t^\star_{\mathsf{ARG}}$*,*

$$\Pr\left[\begin{array}{l|l} |\mathbb{x}| \leq n & \mathsf{pp} \leftarrow \mathcal{G}(1^\lambda, n) \\ \wedge\, \mathbb{x} \notin L(R) & \eta \leftarrow \mathcal{D} \\ \wedge\, b = 1 & (\mathbb{x}, \mathsf{aux}) \leftarrow \widetilde{\mathcal{P}}(\mathsf{pp}, \eta) \\ & b \leftarrow \left\langle \widetilde{\mathcal{P}}(\mathsf{aux}), \mathcal{V}(\mathsf{pp}, \mathbb{x})\right\rangle \end{array}\right] \leq \epsilon^\star_{\mathsf{ARG}}(\lambda, n, t^\star_{\mathsf{ARG}}) \enspace .$$

Definition 3.5 (Adaptive knowledge soundness)**.** $\mathsf{ARG} = (\mathcal{G}, \mathcal{P}, \mathcal{V})$ *for a relation* R *has* **(adaptive) knowledge soundness error** κ_{ARG} **with extraction time** $t_\mathcal{E}$ *if there exists a probabilistic algorithm* $\mathcal{E}$ *such that for every security parameter* $\lambda \in \mathbb{N}$*, instance size bound* $n \in \mathbb{N}$*, auxiliary input distribution* $\mathcal{D}$*, adversary time bound* $t_{\mathsf{ARG}} \in \mathbb{N}$*, and* t_{ARG}*-time algorithm* $\widetilde{\mathcal{P}}$*,*

$$\Pr\left[\begin{array}{l|l} & \mathsf{pp} \leftarrow \mathcal{G}(1^\lambda, n) \\ |\mathbb{x}| \leq n & \eta \leftarrow \mathcal{D} \\ \wedge\, (\mathbb{x}, \mathbb{w}) \notin R & (\mathbb{x}, \mathsf{aux}) \leftarrow \widetilde{\mathcal{P}}(\mathsf{pp}, \eta) \\ \wedge\, b = 1 & b \xleftarrow{\mathsf{tr}} \left\langle \widetilde{\mathcal{P}}(\mathsf{aux}), \mathcal{V}(\mathsf{pp}, \mathbb{x})\right\rangle \\ & \mathbb{w} \leftarrow \mathcal{E}^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, \mathsf{tr}, b) \end{array}\right] \leq \kappa_{\mathsf{ARG}}(\lambda, n, t_{\mathsf{ARG}}) \enspace ;$$

moreover, $\mathcal{E}$ *runs in time* $t_\mathcal{E}(\lambda, n, t_{\mathsf{ARG}})$*.*

Definition 3.6 (Adaptive expected-time knowledge soundness)**.** $\mathsf{ARG} = (\mathcal{G}, \mathcal{P}, \mathcal{V})$ *for a relation* R *has* **(adaptive) expected-time knowledge soundness error** $\kappa^\star_{\mathsf{ARG}}$ **with extraction time** $t^\star_{\mathcal{E}}$ *if there exists a probabilistic algorithm* $\mathcal{E}$ *such that for every security parameter* $\lambda \in \mathbb{N}$*, instance size bound* $n \in \mathbb{N}$*, auxiliary input distribution* $\mathcal{D}$*, adversary expected-time bound* $t^\star_{\mathsf{ARG}} \in \mathbb{N}$*, and algorithm* $\widetilde{\mathcal{P}}$ *with expected running time at most* $t^\star_{\mathsf{ARG}}$*,*

$$\Pr\left[\begin{array}{l|l} & \mathsf{pp} \leftarrow \mathcal{G}(1^\lambda, n) \\ |\mathbb{x}| \leq n & \eta \leftarrow \mathcal{D} \\ \wedge\, (\mathbb{x}, \mathbb{w}) \notin R & (\mathbb{x}, \mathsf{aux}) \leftarrow \widetilde{\mathcal{P}}(\mathsf{pp}, \eta) \\ \wedge\, b = 1 & b \xleftarrow{\mathsf{tr}} \langle \widetilde{\mathcal{P}}(\mathsf{aux}), \mathcal{V}(\mathsf{pp}, \mathbb{x}) \rangle \\ & \mathbb{w} \leftarrow \mathcal{E}^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, \mathsf{tr}, b) \end{array}\right] \leq \kappa^\star_{\mathsf{ARG}}(\lambda, n, t^\star_{\mathsf{ARG}}) \ ;$$

moreover, $\mathcal{E}$ *has expected running time at most* $t^\star_{\mathcal{E}}(\lambda, n, t^\star_{\mathsf{ARG}})$*.*

Above, $b \xleftarrow{\mathsf{tr}} \langle \widetilde{\mathcal{P}}(\mathsf{aux}), \mathcal{V}(\mathsf{pp}, \mathbb{x}) \rangle$ denotes the fact that tr is the transcript of the interaction (i.e., messages exchanged between $\widetilde{\mathcal{P}}$ and $\mathcal{V}$). Moreover, $\mathcal{E}^{\widetilde{\mathcal{P}}}$ means that $\mathcal{E}$ has black-box access to (each next-message function of) $\widetilde{\mathcal{P}}$; in particular $\mathcal{E}$ can send verifier messages to $\widetilde{\mathcal{P}}$ in order to obtain the next message of $\widetilde{\mathcal{P}}$ (for a partial interaction where $\mathcal{V}$ sent those messages).

For security parameter $\lambda \in \mathbb{N}$ and instance size bound $n \in \mathbb{N}$, we write $t_{\mathcal{P}} \coloneqq t_{\mathcal{P}}(\lambda, n)$ to denote the running time of $\mathcal{P}$, and we write $t_{\mathcal{V}} \coloneqq t_{\mathcal{V}}(\lambda, n)$ to denote the worst-case running time of $\mathcal{V}$ (even when interacting with a malicious prover $\widetilde{\mathcal{P}}$).

Remark 3.7. The argument generator $\mathcal{G}$ receives two inputs: the security parameter λ and an instance size bound n. This means that the public parameter sampled by $\mathcal{G}$ may work only for instances of size at most n. However, one could consider the stronger notion where the sampled public parameter works for all instance sizes; in this case $\mathcal{G}$ receives only λ as input. Our analysis works for both cases.

Remark 3.8 (Deterministic adversaries)**.** Following prior work, we assume that $\widetilde{\mathcal{P}}$ is deterministic when proving upper bounds on (expected-time) soundness and knowledge soundness. This assumption is purely for notational simplicity (specifically, it allows us to avoid tracking the prover's intermediate states), and our approach works for probabilistic adversaries as well.

Remark 3.9 (Plain model variant)**.** The above definitions consider interactive arguments in the *common reference string model*, where a generator samples a public parameter used by the argument prover and the argument verifier. One could also consider interactive arguments in the *plain model*, where there is no generator. This latter notion is implied, at the cost of an additional verifier message, as we now explain.

Suppose that $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is an interactive argument in the common reference string model. We describe an interactive argument $(\mathcal{P}', \mathcal{V}')$ in the plain model with an additional verifier message. The argument prover $\mathcal{P}'$ receives as input an instance $\mathbb{x}$ and witness $\mathbb{w}$, and the argument verifier $\mathcal{V}'$ receives as input the instance $\mathbb{x}$; both also receive as input the security parameter λ (in unary). They interact as follows:

- $\mathcal{V}'$ samples a public parameter $\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda, |\mathbb{x}|)$ and sends pp to $\mathcal{P}'$;
- $\mathcal{P}'$ and $\mathcal{V}'$ simulate an interaction of $\mathcal{P}(\mathsf{pp}, \mathbb{x}, \mathbb{w})$ and $\mathcal{V}(\mathsf{pp}, \mathbb{x})$.

It is straightforward to see that $(\mathcal{P}', \mathcal{V}')$ satisfies the standard definitions of completeness, soundness, and knowledge soundness for interactive arguments in the plain model.[5] In fact, it would suffice for $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ to satisfy the non-adaptive relaxations of soundness and knowledge soundness.

3.2 Query classes

Definition 3.10. *A* **query class Q** *is a set of functions of the form* $\alpha \colon \Sigma^\ell \to \mathbb{D}$*, where* Σ *is the input alphabet,* $\ell \in \mathbb{N}$ *is the input length, and* $\mathbb{D}$ *is the answer domain. Here,* Σ *and* $\mathbb{D}$ *are finite sets of cardinality* ≥ 2*.*

[5] These standard definitions can be derived from Definitions 3.2, 3.3 and 3.5 by setting pp to be empty.

Definition 3.11. *A* **constraint** *of a query class* $\mathbf{Q}$ *is a pair* $(\alpha, \beta) \in \mathbf{Q} \times \mathbb{D}$*. For a set of constraints* $\mathcal{C}$*, its solution space is*

$$\mathrm{Sol}(\mathcal{C}) \coloneqq \left\{ \Pi \in \Sigma^\ell : \forall (\alpha, \beta) \in \mathcal{C}, \alpha(\Pi) = \beta \right\} \ ;$$

we say $\mathcal{C}$ *is* **satisfiable** *if* $\mathrm{Sol}(\mathcal{C}) \neq \emptyset$*. For satisfiable* $\mathcal{C}$*, a constraint* (α, β) *is:*
- **entailed** *by* $\mathcal{C}$ *if* $\alpha(\Pi) = \beta$ *for every* $\Pi \in \mathrm{Sol}(\mathcal{C})$*.*
- **refuted** *by* $\mathcal{C}$ *if* $\alpha(\Pi) \neq \beta$ *for every* $\Pi \in \mathrm{Sol}(\mathcal{C})$*.*
- **decided** *by* $\mathcal{C}$ *if it is entailed or refuted, and* **undecided** *otherwise.*

Definition 3.12. *The* **diameter** *of a query class* $\mathbf{Q}$*, denoted* $d_{\mathbf{Q}}$*, is the largest integer* $d \geq 0$ *for which there exist constraints* $p_1, \ldots, p_d$ *satisfying*

$$\emptyset \neq \mathrm{Sol}(\{p_1, \ldots, p_i\}) \subsetneq \mathrm{Sol}(\{p_1, \ldots, p_{i-1}\}) \quad \text{for every } i \in [d] \ ,$$

or ∞ *if no largest* d *exists.*

Definition 3.13. *For a satisfiable constraint set* $\mathcal{C}$ *and a constraint* (α, β)*, define*

$$\mathcal{C} \oplus (\alpha, \beta) \coloneqq \begin{cases} \mathcal{C} \cup \{(\alpha, \beta)\} & \text{if } (\alpha, \beta) \text{ is undecided by } \mathcal{C}, \\ \mathcal{C} & \text{otherwise.} \end{cases}$$

The **refinement** *of a constraint list* $L = (c_1, \ldots, c_n)$ *is* $\mathcal{C}_L \coloneqq \emptyset \oplus c_1 \oplus \cdots \oplus c_n$*.*

Definition 3.14. *A* **constraint solver** *for a query class* $\mathbf{Q}$ *is a stateful algorithm* $\mathsf{Solver}_{\mathbf{Q}}$ *that supports the following operations on an (initially empty) list of constraints* L*:*
- *Given a constraint, report whether it is entailed, refuted, or undecided by* $\mathcal{C}_L$*.*
- *Append an undecided constraint to* L*.*
- *Output an element* $\Pi \in \mathrm{Sol}(\mathcal{C}_L)$*; for simplicity, we assume the choice is deterministic.*

We write $t_{\mathbf{Q}}$ *to denote the running time of one operation.*

Query sets. Let $\boldsymbol{\alpha} \subseteq \mathbf{Q}$ be an (ordered) query set. We write $\boldsymbol{\alpha}(\Pi) \in \mathbb{D}^{\boldsymbol{\alpha}}$ to denote the map sending $\alpha \in \boldsymbol{\alpha}$ to $\alpha(\Pi)$. Given an answer set $\boldsymbol{\beta} \in \mathbb{D}^{\boldsymbol{\alpha}}$, we call $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ a *query-answer tuple*. By convention, we say:
- "$\mathcal{C}$ entails $(\boldsymbol{\alpha}, \boldsymbol{\beta})$" if $\mathcal{C}$ entails every constraint in $(\boldsymbol{\alpha}, \boldsymbol{\beta})$.
- "$\mathcal{C}$ refutes $(\boldsymbol{\alpha}, \boldsymbol{\beta})$" if $\mathcal{C}$ refutes at least one constraint in $(\boldsymbol{\alpha}, \boldsymbol{\beta})$.
- "$\mathcal{C}$ decides $(\boldsymbol{\alpha}, \boldsymbol{\beta})$" if $\mathcal{C}$ decides every constraint in $(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

We often interpret a query-answer tuple as a list of constraints. Similarly, we interpret a list of query-answer tuples as a list of constraints, i.e., the list obtained by concatenating the constraints from each tuple.

Families of query classes. We often say "query class" to mean a *family* of query classes $\mathbf{Q} = (\mathbf{Q}_\ell)_{\ell \in \mathbb{N}}$ over a common alphabet Σ and answer domain $\mathbb{D}$, where each $\mathbf{Q}_\ell$ is a set of functions $\alpha \colon \Sigma^\ell \to \mathbb{D}$. Consequently, $d_{\mathbf{Q}}$ and $t_{\mathbf{Q}}$ are functions of $\ell \in \mathbb{N}$.

Definition 3.15 (Point queries)**.** *The class of point queries* $\mathbf{Q}_{\mathsf{Point}}$ *is defined to be*

$$\mathbf{Q}_{\mathsf{Point}}(\ell) \coloneqq \{\alpha_i \colon \Sigma^\ell \to \Sigma\}_{i \in [\ell]} \ ,$$

where α_i *maps* Π *to its* i*-th symbol* $\Pi[i]$*.*

3.3 Functional commitments

An (interactive) *functional commitment scheme* for a query class $\mathbf{Q} = (\mathbf{Q}_\ell)_{\ell\in\mathbb{N}}$ over Σ is a tuple of algorithms

$$\mathsf{FC} = (\mathsf{FC.Gen}, \mathsf{FC.Commit}, \mathcal{P}_{\mathsf{FC}}, \mathcal{V}_{\mathsf{FC}})$$

with the following syntax.

- $\mathsf{FC.Gen}(1^\lambda, \ell_{\mathsf{max}}) \to \mathsf{pp}$: On input a security parameter $\lambda \in \mathbb{N}$ and message length bound $\ell_{\mathsf{max}} \in \mathbb{N}$, $\mathsf{FC.Gen}$ samples public parameter pp.
- $\mathsf{FC.Commit}(\mathsf{pp}, \Pi) \to (\mathsf{cm}, \mathsf{aux})$: On input a public parameter pp and a message $\Pi \in \Sigma^\ell$, $\mathsf{FC.Commit}$ produces a commitment cm and the corresponding auxiliary state aux.
- $\langle \mathcal{P}_{\mathsf{FC}}(\mathsf{aux}, \boldsymbol{\alpha}, \boldsymbol{\beta}), \mathcal{V}_{\mathsf{FC}}(\mathsf{pp}, \mathsf{cm}, \boldsymbol{\alpha}, \boldsymbol{\beta})\rangle \to b$: The *opening phase* is an interactive protocol between the prover $\mathcal{P}_{\mathsf{FC}}$ and verifier $\mathcal{V}_{\mathsf{FC}}$. The prover receives as input an auxiliary state aux, a query set $\boldsymbol{\alpha} \subseteq \mathbf{Q}_\ell$, and an answer set $\boldsymbol{\beta} \in \mathbb{D}^{\boldsymbol{\alpha}}$. The verifier receives as input the public parameter pp, a commitment cm, a query set $\boldsymbol{\alpha}$, and an answer set $\boldsymbol{\beta}$; after interacting, it accepts or rejects, which is represented by the bit $b \in \{0, 1\}$.

Definition 3.16 (Completeness)**.** FC *has* **completeness error** γ_{FC} *if for every security parameter $\lambda \in \mathbb{N}$, message length bound $\ell_{\mathsf{max}} \in \mathbb{N}$, query bound $s \in \mathbb{N}$, message $\Pi \in \Sigma^\ell$ where $\ell \le \ell_{\mathsf{max}}$, and query set $\boldsymbol{\alpha} \subseteq \mathbf{Q}_\ell$ with $|\boldsymbol{\alpha}| \le s$,*

$$\Pr\left[b = 1 \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{FC.Gen}(1^\lambda, \ell_{\mathsf{max}}) \\ (\mathsf{cm}, \mathsf{aux}) \leftarrow \mathsf{FC.Commit}(\mathsf{pp}, \Pi) \\ b \leftarrow \langle \mathcal{P}_{\mathsf{FC}}(\mathsf{aux}, \boldsymbol{\alpha}, \boldsymbol{\alpha}(\Pi)), \mathcal{V}_{\mathsf{FC}}(\mathsf{pp}, \mathsf{cm}, \boldsymbol{\alpha}, \boldsymbol{\alpha}(\Pi))\rangle \end{array} \right] \ge 1 - \gamma_{\mathsf{FC}}(\lambda, \ell_{\mathsf{max}}, s) \ .$$

It has **perfect completeness** *if* $\gamma_{\mathsf{FC}} = 0$.

Definition 3.17 (Function binding game)**.** FCGame *is an interactive algorithm that works as follows.*

$\mathsf{FCGame}(\mathsf{pp})$*:*

1. *For $i = 1, 2, \ldots$, until the interaction terminates:*
   (a) *Receive* $(\mathsf{cm}_i, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$.
   (b) *Run the FC verifier* $\mathcal{V}_{\mathsf{FC}}(\mathsf{pp}, \mathsf{cm}_i, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$*, and let b_i be the resulting decision bit.*
   (c) *Send b_i.*

At the end of the interaction, for every commitment cm*, $\ell \in \mathbb{N}$, and $s \in \mathbb{N}$, we define*

$$\mathcal{C}_{\mathsf{cm},\ell,s} := \left\{ (\alpha, \boldsymbol{\beta}_i(\alpha)) : \begin{array}{l} i \ge 1,\ \alpha \in \boldsymbol{\alpha}_i, \\ \mathsf{cm}_i = \mathsf{cm},\ \boldsymbol{\alpha}_i \subseteq \mathbf{Q}_\ell, \\ b_i = 1,\ |\boldsymbol{\alpha}_i| \le s \end{array} \right\} \ .$$

Definition 3.18 (Function binding)**.** FC *has* **(strict-time) function binding error** ϵ_{FC} *if for every security parameter $\lambda \in \mathbb{N}$, message length bound $\ell_{\mathsf{max}} \in \mathbb{N}$, query bound $s \in \mathbb{N}$, auxiliary input distribution $\mathcal{D}_{\mathsf{FC}}$, adversary time bound t_{FC}, and t_{FC}-time algorithm $\mathcal{A}_{\mathsf{FC}}$,*

$$\Pr\left[\exists\, \mathsf{cm}, \ell \le \ell_{\mathsf{max}} :\ \mathrm{Sol}(\mathcal{C}_{\mathsf{cm},\ell,s}) = \emptyset \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{FC.Gen}(1^\lambda, \ell_{\mathsf{max}}) \\ \mathsf{ai}_{\mathsf{FC}} \leftarrow \mathcal{D}_{\mathsf{FC}} \\ \langle \mathcal{A}_{\mathsf{FC}}(\mathsf{pp}, \mathsf{ai}_{\mathsf{FC}}), \mathsf{FCGame}(\mathsf{pp})\rangle \end{array} \right] \le \epsilon_{\mathsf{FC}}(\lambda, \ell_{\mathsf{max}}, s, t_{\mathsf{FC}}) \ .$$

Definition 3.19 (Expected-time function binding)**.** FC *has* **expected-time function binding error** $\epsilon^\star_{\mathsf{FC}}$ *if for every security parameter $\lambda \in \mathbb{N}$, message length bound $\ell_{\mathsf{max}} \in \mathbb{N}$, query bound $s \in \mathbb{N}$, auxiliary input distribution $\mathcal{D}_{\mathsf{FC}}$, adversary expected-time bound $t^\star_{\mathsf{FC}}$, and algorithm $\mathcal{A}_{\mathsf{FC}}$ with expected running time at most $t^\star_{\mathsf{FC}}$,*

$$\Pr\left[\exists\, \mathsf{cm}, \ell \le \ell_{\mathsf{max}} :\ \mathrm{Sol}(\mathcal{C}_{\mathsf{cm},\ell,s}) = \emptyset \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{FC.Gen}(1^\lambda, \ell_{\mathsf{max}}) \\ \mathsf{ai}_{\mathsf{FC}} \leftarrow \mathcal{D}_{\mathsf{FC}} \\ \langle \mathcal{A}_{\mathsf{FC}}(\mathsf{pp}, \mathsf{ai}_{\mathsf{FC}}), \mathsf{FCGame}(\mathsf{pp})\rangle \end{array} \right] \le \epsilon^\star_{\mathsf{FC}}(\lambda, \ell_{\mathsf{max}}, s, t^\star_{\mathsf{FC}}) \ .$$

Remark 3.20 (Monotonicity of ϵ_{FC})**.** We assume hereafter that the function binding error ϵ_{FC} is monotone in each coordinate in the natural direction:

- $\epsilon_{\mathsf{FC}}(\cdot, \ell, s, t_{\mathsf{FC}})$ is non-increasing (larger security parameters decrease an adversary's success);
- $\epsilon_{\mathsf{FC}}(\lambda, \cdot, s, t_{\mathsf{FC}})$ is non-decreasing (larger messages increase an adversary's success);
- $\epsilon_{\mathsf{FC}}(\lambda, \ell, \cdot, t_{\mathsf{FC}})$ is non-decreasing (more constraints increase an adversary's success); and
- $\epsilon_{\mathsf{FC}}(\lambda, \ell, s, \cdot)$ is non-decreasing (the success of an adversary increases with its computational power).

The last condition is trivially satisfied, while the first should also hold in any reasonable commitment scheme. The remaining two assumptions are natural; moreover, we may forgo them by replacing expressions of the form $\epsilon_{\mathsf{FC}}(\lambda, \ell_{\mathsf{max}}, s_{\mathsf{max}}, t_{\mathsf{FC}})$, where $\ell_{\mathsf{max}} = \max_i \{\ell_i\}$ and $s_{\mathsf{max}} = \max_j \{s_j\}$, with

$$\max_{i,j} \{\epsilon_{\mathsf{FC}}(\lambda, \ell_i, s_j, t_{\mathsf{FC}})\} \quad .$$

Analogously, we assume the expected-time function binding error $\epsilon^{\star}_{\mathsf{FC}}$ is monotone.

3.4 Functional interactive oracle proofs

A *functional interactive oracle proof* (FIOP) is an interactive proof in which the verifier accesses each prover string through a prescribed query class. The definition below is adapted from the definition of IOPs with special queries in [BCG20].

Definition 3.21. *A (public-coin) FIOP for a relation R with query class $\mathbf{Q} = (\mathbf{Q}_\ell)_{\ell \in \mathbb{N}}$ over alphabet Σ and answer domain $\mathbb{D}$ is a tuple $\mathsf{FIOP} = (\mathbf{P}, \mathbf{V})$. The prover receives an instance-witness pair $(\mathbb{x}, \mathbb{w})$ and the verifier receives the instance $\mathbb{x}$. The protocol has k rounds. In round $i \in [\mathsf{k}]$, the prover sends a proof string $\Pi_i \in \Sigma^{\ell_i}$, after which the verifier sends uniformly random coins $\rho_i \in \{0,1\}^{\mathsf{r}_i}$. After the interaction, the verifier accesses each Π_i only through queries in $\mathbf{Q}_{\ell_i}$: the answer to $\alpha \in \mathbf{Q}_{\ell_i}$ is $\alpha(\Pi_i) \in \mathbb{D}$. It then outputs a decision bit. We write $\mathbf{tr} = (\Pi_1, \rho_1, \ldots, \Pi_{\mathsf{k}}, \rho_{\mathsf{k}})$ for the interaction transcript.*

An *interactive oracle proof* (IOP) is the special case of an FIOP where $\mathbf{Q} = \mathbf{Q}_{\mathsf{Point}}$ is the class of point queries.

Definition 3.22 (Completeness)**.** $\mathsf{FIOP} = (\mathbf{P}, \mathbf{V})$ *for a relation R has* **completeness error** γ_{FIOP} *if, for every instance size bound $n \in \mathbb{N}$ and every instance-witness pair $(\mathbb{x}, \mathbb{w}) \in R$ with $|\mathbb{x}| \leq n$,*

$$\Pr\left[\left\langle \mathbf{P}(\mathbb{x}, \mathbb{w}), \mathbf{V}(\mathbb{x}) \right\rangle = 1\right] \geq 1 - \gamma_{\mathsf{FIOP}}(n) \quad .$$

It has **perfect completeness** *if $\gamma_{\mathsf{FIOP}} = 0$.*

Definition 3.23 (Soundness)**.** $\mathsf{FIOP} = (\mathbf{P}, \mathbf{V})$ *for a relation R has* **soundness error** ϵ_{FIOP} *if, for every instance size bound $n \in \mathbb{N}$, (unbounded) circuit $\widetilde{\mathbf{P}}$, and auxiliary input distribution $\mathbf{D}$,*

$$\Pr\left[\begin{array}{l|l} |\mathbb{x}| \leq n & \mathbf{ai} \leftarrow \mathbf{D} \\ \wedge\, \mathbb{x} \notin L(R) & (\mathbb{x}, \mathbf{aux}) \leftarrow \widetilde{\mathbf{P}}(\mathbf{ai}) \\ \wedge\, b = 1 & b \leftarrow \left\langle \widetilde{\mathbf{P}}(\mathbf{aux}), \mathbf{V}(\mathbb{x}) \right\rangle \end{array}\right] \leq \epsilon_{\mathsf{FIOP}}(n) \quad .$$

Definition 3.24 (Knowledge soundness)**.** $\mathsf{FIOP} = (\mathbf{P}, \mathbf{V})$ *for a relation R has* **(straightline) knowledge soundness error** κ_{FIOP} **with extraction time** $t_{\mathbf{E}}$ *if there exists a deterministic algorithm $\mathbf{E}$ such that, for every instance size bound $n \in \mathbb{N}$, adversary $\widetilde{\mathbf{P}}$, and auxiliary input distribution $\mathbf{D}$,*

$$\Pr\left[\begin{array}{l|l} |\mathbb{x}| \leq n & \mathbf{ai} \leftarrow \mathbf{D} \\ \wedge\, (\mathbb{x}, \mathbb{w}) \notin R & (\mathbb{x}, \mathbf{aux}) \leftarrow \widetilde{\mathbf{P}}(\mathbf{ai}) \\ \wedge\, b = 1 & b \xleftarrow{\mathbf{tr}} \left\langle \widetilde{\mathbf{P}}(\mathbf{aux}), \mathbf{V}(\mathbb{x}) \right\rangle \\ & \mathbb{w} \leftarrow \mathbf{E}(\mathbb{x}, \mathbf{tr}) \end{array}\right] \leq \kappa_{\mathsf{FIOP}}(n) \ ;$$

moreover, $\mathbf{E}$ runs in time $t_{\mathbf{E}}(n)$.

We consider the following efficiency measures for an FIOP:

- the *round complexity* k is the number of rounds (back-and-forth interactions) between the prover and verifier;
- the *alphabet* Σ is the set over which symbols of proof strings are defined;
- the *proof length* ℓ is the total number of alphabet symbols sent by the prover; moreover, ℓ_i is the length of the round-i proof string and $\ell_{\mathsf{max}} := \max_{i \in [\mathsf{k}]} \ell_i$;
- the *query complexity* q is the total number of queries issued by the verifier; moreover, q_i is the number of queries issued to the round-i proof string and $\mathsf{q}_{\mathsf{max}} := \max_{i \in [\mathsf{k}]} \mathsf{q}_i$;
- the *randomness complexity* r is the total number of random bits sent by the verifier; moreover, r_i is the number of random bits sent in round i.

Each efficiency measure may depend on the instance size $|\mathbb{x}|$.

3.5 Funky protocol

The Funky protocol combines an FIOP with a functional commitment scheme to construct an interactive argument. In more detail, consider the following ingredients:

- $\mathsf{FIOP} = (\mathbf{P}, \mathbf{V})$, an FIOP for a relation R with query class $\mathbf{Q}$ and round complexity k. We use explicit state-passing notation for $\mathbf{P}$, via the auxiliary state $\mathbf{aux}$.
- $\mathsf{FC} = (\mathsf{FC.Gen}, \mathsf{FC.Commit}, \mathcal{P}_{\mathsf{FC}}, \mathcal{V}_{\mathsf{FC}})$, a functional commitment scheme for $\mathbf{Q}$.

The construction $\mathsf{ARG} := \mathsf{Funky}[\mathsf{FIOP}, \mathsf{FC}]$, specified below, is adapted from [CGKY26].

Construction 3.25 (Funky protocol)**.** The argument generator $\mathcal{G}$ receives as input a security parameter $\lambda \in \mathbb{N}$ (in unary) and an instance size bound $n \in \mathbb{N}$, and works as follows.

$\mathcal{G}(1^\lambda, n)$:
1. Sample $\mathsf{pp} \leftarrow \mathsf{FC.Gen}(1^\lambda, \ell_{\mathsf{max}}(n))$.
2. Return pp.

The argument prover $\mathcal{P}$ receives as input the public parameter pp, an instance $\mathbb{x}$, and a witness $\mathbb{w}$, and the argument verifier $\mathcal{V}$ receives as input the public parameter pp and the instance $\mathbb{x}$. Then $\mathcal{P}$ and $\mathcal{V}$ interact as follows.

1. **FC commitments and FIOP challenges.** For each round $i \in [\mathsf{k}]$:
   (a) The prover:
      i. Computes the i-th FIOP string and auxiliary state
      $$(\Pi_i, \mathbf{aux}_i) \leftarrow \begin{cases} \mathbf{P}(\mathbb{x}, \mathbb{w}) & i = 1 \\ \mathbf{P}(\mathbf{aux}_{i-1}, \rho_{i-1}) & \text{otherwise.} \end{cases}$$
      ii. Computes a functional commitment to the FIOP string:
      $$(\mathsf{cm}_i, \mathsf{aux}_i) \leftarrow \mathsf{FC.Commit}(\mathsf{pp}, \Pi_i) \ .$$
      iii. Sends cm_i to the verifier.
   (b) The verifier samples $\rho_i \leftarrow \{0,1\}^{\mathsf{r}_i}$ and sends ρ_i to the prover.

2. **FIOP answers.** The prover:
   (a) Runs the FIOP verifier
   $$\mathbf{V}^{\Pi_1, \ldots, \Pi_{\mathsf{k}}}(\mathbb{x}; \rho_1, \ldots, \rho_{\mathsf{k}})$$
   to determine $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{\mathsf{k}}$, where $\boldsymbol{\alpha}_i \subseteq \mathbf{Q}_{\ell_i}$ is the set of queries issued by $\mathbf{V}$ to Π_i.

(b) Computes the evaluation $\boldsymbol{\beta}_i(\alpha) \coloneqq \alpha(\Pi_i)$ for every $i \in [\mathsf{k}]$ and $\alpha \in \boldsymbol{\alpha}_i$.
(c) Sends $(\tau_1, \ldots, \tau_{\mathsf{k}})$ to the verifier, where τ_i denotes the query-answer tuple $(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$.

3. **FC openings.** For each round $i \in [\mathsf{k}]$, sequentially, the prover and verifier engage in the FC opening phase for the i-th commitment:

$$\left\langle \mathcal{P}_{\mathsf{FC}}(\mathsf{aux}_i, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i), \mathcal{V}_{\mathsf{FC}}(\mathsf{pp}, \mathsf{cm}_i, \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) \right\rangle \to b_i \ .$$

Let pf_i denote the resulting transcript.

4. **Decision.** The verifier accepts if and only if the following hold:
(a) The FIOP verifier $\mathbf{V}(\mathbb{x}; \rho_1, \ldots, \rho_{\mathsf{k}})$, receiving answers according to $\tau_1, \ldots, \tau_{\mathsf{k}}$, accepts;
(b) $b_i = 1$ for every $i \in [\mathsf{k}]$.

The resulting interaction transcript is

$$\mathsf{tr} = (\mathsf{cm}_1, \rho_1, \ldots, \mathsf{cm}_{\mathsf{k}}, \rho_{\mathsf{k}}, (\tau_1, \ldots, \tau_{\mathsf{k}}), \mathsf{pf}_1, \ldots, \mathsf{pf}_{\mathsf{k}}) \ .$$

4 Barriers for knowledge soundness

We prove lower bounds on the non-adaptive knowledge soundness error of strict-time extractors. We also prove a lower bound on the expected number of rewinds for the Funky protocol.

4.1 Additional preliminaries

We define non-adaptive knowledge soundness, which is the notion of security considered in the lower bounds.

Definition 4.1 (Non-adaptive knowledge soundness)**.** $\mathsf{ARG} = (\mathcal{G}, \mathcal{P}, \mathcal{V})$ *for a relation* R *has* **non-adaptive knowledge soundness error** κ_{ARG} **with extraction time** $t_{\mathcal{E}}$ *if there exists a probabilistic algorithm* $\mathcal{E}$ *such that for every security parameter* $\lambda \in \mathbb{N}$*, instance* $\mathbb{x}$*, auxiliary input distribution* $\mathcal{D}$*, adversary time bound* $t_{\mathsf{ARG}} \in \mathbb{N}$*, and* t_{ARG}*-time algorithm* $\widetilde{\mathcal{P}}$*,*

$$\Pr\left[\begin{array}{l|l} & \mathsf{pp} \leftarrow \mathcal{G}(1^\lambda, |\mathbb{x}|) \\ (\mathbb{x}, \mathbb{w}) \notin R & \eta \leftarrow \mathcal{D} \\ \wedge\, b = 1 & b \xleftarrow{\mathsf{tr}} \left\langle \widetilde{\mathcal{P}}(\mathsf{pp}, \eta), \mathcal{V}(\mathsf{pp}, \mathbb{x}) \right\rangle \\ & \mathbb{w} \leftarrow \mathcal{E}^{\widetilde{\mathcal{P}}(\mathsf{pp},\eta)}(\mathsf{pp}, \mathbb{x}, \mathsf{tr}, b) \end{array}\right] \leq \kappa_{\mathsf{ARG}}(\lambda, \mathbb{x}, t_{\mathsf{ARG}}) \ ;$$

moreover, $\mathcal{E}$ *runs in time* $t_{\mathcal{E}}(\lambda, \mathbb{x}, t_{\mathsf{ARG}})$*.*

Definition 4.2 (Non-adaptive expected-time knowledge soundness)**.** $\mathsf{ARG} = (\mathcal{G}, \mathcal{P}, \mathcal{V})$ *for a relation* R *has* **non-adaptive expected-time knowledge soundness error** $\kappa^\star_{\mathsf{ARG}}$ **with extraction time** $t^\star_{\mathcal{E}}$ *if there exists a probabilistic algorithm* $\mathcal{E}$ *such that for every security parameter* $\lambda \in \mathbb{N}$*, instance* $\mathbb{x}$*, auxiliary input distribution* $\mathcal{D}$*, adversary expected-time bound* $t^\star_{\mathsf{ARG}} \in \mathbb{N}$*, and algorithm* $\widetilde{\mathcal{P}}$ *with expected running time at most* $t^\star_{\mathsf{ARG}}$*,*

$$\Pr\left[\begin{array}{l|l} & \mathsf{pp} \leftarrow \mathcal{G}(1^\lambda, |\mathbb{x}|) \\ (\mathbb{x}, \mathbb{w}) \notin R & \eta \leftarrow \mathcal{D} \\ \wedge\, b = 1 & b \xleftarrow{\mathsf{tr}} \left\langle \widetilde{\mathcal{P}}(\mathsf{pp}, \eta), \mathcal{V}(\mathsf{pp}, \mathbb{x}) \right\rangle \\ & \mathbb{w} \leftarrow \mathcal{E}^{\widetilde{\mathcal{P}}(\mathsf{pp},\eta)}(\mathsf{pp}, \mathbb{x}, \mathsf{tr}, b) \end{array}\right] \leq \kappa^\star_{\mathsf{ARG}}(\lambda, \mathbb{x}, t^\star_{\mathsf{ARG}}) \ ;$$

moreover, $\mathcal{E}$ *has expected running time at most* $t^\star_{\mathcal{E}}(\lambda, \mathbb{x}, t^\star_{\mathsf{ARG}})$*.*

We define (honest-verifier) zero-knowledge for FIOPs.

Definition 4.3. *Let* $\mathsf{FIOP} = (\mathbf{P}, \mathbf{V})$ *be an FIOP with query class* $\mathbf{Q}$*, and let* $(\mathbb{x}, \mathbb{w})$ *be an instance-witness pair. The view of the honest verifier, denoted by* $\mathsf{View}(\mathbf{P}, \mathbf{V}, \mathbb{x}, \mathbb{w})$*, is the random variable* $((\rho_1, \ldots, \rho_{\mathsf{k}}), (\tau_1, \ldots, \tau_{\mathsf{k}}))$ *where:*

- *each* ρ_i *is uniformly sampled from* $\{0,1\}^{\mathsf{r}_i}$*;*
- *each* $\tau_i = (\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$ *is a query-answer tuple, where* $\boldsymbol{\alpha}_i \subseteq \mathbf{Q}_{\ell_i}$ *is the set of queries issued to* Π_i *by the FIOP verifier* $\mathbf{V}^{\Pi_1,\ldots,\Pi_{\mathsf{k}}}(\mathbb{x}; \rho_1, \ldots, \rho_{\mathsf{k}})$ *and* $\boldsymbol{\beta}_i := \boldsymbol{\alpha}_i(\Pi_i)$ *is the corresponding evaluation; here,* $\Pi_i \in \Sigma^{\ell_i}$ *is the* i*-th proof string output by the FIOP prover* $\mathbf{P}(\mathbb{x}, \mathbb{w}, \rho_1, \ldots, \rho_{i-1})$*.*

Definition 4.4 (Honest-verifier zero-knowledge)**.** $\mathsf{FIOP} = (\mathbf{P}, \mathbf{V})$ *for a relation* R *has* **honest-verifier zero-knowledge error** ζ_{FIOP} *if there exists a probabilistic polynomial-time algorithm* $\mathbf{S}$ *such that, for every instance-witness pair* $(\mathbb{x}, \mathbb{w}) \in R$*, the random variables* $\mathsf{View}(\mathbf{P}, \mathbf{V}, \mathbb{x}, \mathbb{w})$ *and* $\mathbf{S}(\mathbb{x})$ *are* $\zeta_{\mathsf{FIOP}}(\mathbb{x})$*-close in statistical distance.*

We formalize the notion of FC-oblivious extractors. Intuitively, these are extractors for the Funky protocol that do not use the functional commitment scheme aside from checking the argument prover's openings.

Below, let $\mathsf{ARG} = \mathsf{Funky}[\mathsf{FIOP}, \mathsf{FC}]$, where:

- FIOP is a public-coin FIOP for a relation R with query class $\mathbf{Q}$ and round complexity k.

- $\mathsf{FC} = (\mathsf{FC.Gen}, \mathsf{FC.Commit}, \mathcal{P}_{\mathsf{FC}}, \mathcal{V}_{\mathsf{FC}})$ is an interactive functional commitment scheme for $\mathbf{Q}$.

Definition 4.5. *For public parameter* pp *and argument transcript*

$$\mathsf{tr} = (\mathsf{cm}_1, \rho_1, \ldots, \mathsf{cm}_{\mathsf{k}}, \rho_{\mathsf{k}}, (\tau_1, \ldots, \tau_{\mathsf{k}}), \mathsf{pf}_1, \ldots, \mathsf{pf}_{\mathsf{k}}) \ ,$$

the **projected transcript***, denoted* tr°*, is defined to be* $(\rho_1, \ldots, \rho_{\mathsf{k}}, (\tau_1, \ldots, \tau_{\mathsf{k}}), b_1, \ldots, b_{\mathsf{k}})$*, where* b_i *is the decision bit of* $\mathcal{V}_{\mathsf{FC}}(\mathsf{pp}, \mathsf{cm}_i, \tau_i, \mathsf{pf}_i)$.

Construction 4.6. For an argument prover $\widetilde{\mathcal{P}}$, the interactive algorithm $\widetilde{\mathcal{P}}^\circ$ works as follows.

$\widetilde{\mathcal{P}}^\circ(\mathsf{pp}, \eta)$:

1. For each round $i \in [\mathsf{k}]$:
   (a) Compute the i-th commitment and auxiliary state
   $$(\mathsf{cm}_i, \mathsf{aux}_i) \leftarrow \begin{cases} \widetilde{\mathcal{P}}(\mathsf{pp}, \eta) & i = 1 \\ \widetilde{\mathcal{P}}(\mathsf{aux}_{i-1}, \rho_{i-1}) & \text{otherwise.} \end{cases}$$
   (b) Send $\perp$ and receive the verifier's message ρ_i.
2. Compute $(\boldsymbol{\tau}, \mathsf{aux}) \leftarrow \widetilde{\mathcal{P}}(\mathsf{aux}_{\mathsf{k}}, \rho_{\mathsf{k}})$ and parse the prover message $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{\mathsf{k}})$; if this is not well-formed, send $\perp$ and halt.
3. For each round $i \in [\mathsf{k}]$, sequentially, simulate the FC opening phase
   $$\langle \widetilde{\mathcal{P}}(\mathsf{aux}), \mathcal{V}_{\mathsf{FC}}(\mathsf{pp}, \mathsf{cm}_i, \tau_i) \rangle \to b_i \ ,$$
   and let aux denote the prover's resulting auxiliary state.
4. Send $((\tau_1, \ldots, \tau_{\mathsf{k}}), b_1, \ldots, b_{\mathsf{k}})$.

Definition 4.7 (FC-oblivious extractor)**.** *An* **FC-oblivious extractor for** ARG *is an algorithm* $\mathcal{E}$ *that receives an instance* $\mathbb{x}$*, projected transcript* tr°*, and decision bit* b*, along with rewinding access to* $\widetilde{\mathcal{P}}^\circ(\mathsf{pp}, \eta)$ *for some argument prover* $\widetilde{\mathcal{P}}$*, public parameters* pp*, and auxiliary input* η*. We write* $t_{\mathcal{E}}(\lambda, |\mathbb{x}|)$ *to denote the running time of* $\mathcal{E}$*, excluding time spent in* $\widetilde{\mathcal{P}}^\circ$.

An FC-oblivious extractor for ARG induces a (standard) extractor for ARG which computes the projected transcript from the original transcript and simulates rewinding access to $\widetilde{\mathcal{P}}^\circ(\mathsf{pp}, \eta)$ using rewinding access to $\widetilde{\mathcal{P}}(\mathsf{pp}, \eta)$. We say that an FC-oblivious extractor achieves non-adaptive knowledge soundness error κ_{ARG} if its induced extractor does, according to Definitions 4.1 and 4.2.

4.2 Strict-time barriers

Throughout, we write

$$\gamma_{\mathsf{ARG}}(\lambda, n) \coloneqq 1 - (1 - \gamma_{\mathsf{FIOP}}(n)) \left(1 - \sum_{i=1}^{\mathsf{k}} \gamma_{\mathsf{FC}}(\lambda, \ell_{\mathsf{max}}, \mathsf{q}_i)\right)$$

as a bound on the completeness error of $\mathsf{ARG} \coloneqq \mathsf{Funky}[\mathsf{FIOP}, \mathsf{FC}]$.

Theorem 4.8. *Let* $\mathsf{FIOP} = (\mathbf{P}, \mathbf{V})$ *be an FIOP for an* NP *relation* R *with query class* $\mathbf{Q}$*, completeness error* γ_{FIOP}*, and honest-verifier zero-knowledge error* ζ_{FIOP}*. Let* FC *be a public-coin functional commitment scheme for* $\mathbf{Q}$ *with completeness error* γ_{FC}*. Let* $\theta \in (0, 1)$.

Suppose there exists an FC-oblivious extractor $\mathcal{E}$ *for* $\mathsf{ARG} \coloneqq \mathsf{Funky}[\mathsf{FIOP}, \mathsf{FC}]$ *with running time* $t_{\mathcal{E}}$ *that rewinds the prover at most* p *times and achieves non-adaptive knowledge soundness error* κ_{ARG}*. Then there exists*

a probabilistic algorithm $\mathcal{A}$ that, on input $\mathbb{x}$, runs in time $t_{\mathcal{E}} + \mathsf{poly}(|\mathbb{x}|)$ and, for every $\mathbb{x} \in L(R)$, accepts $\mathbb{x}$ with probability at least

$$1 - \gamma_{\mathsf{ARG}}(\lambda, |\mathbb{x}|) - \frac{\kappa_{\mathsf{ARG}}(\lambda, \mathbb{x}, t_{\mathsf{ARG}})}{\theta(1-\theta)^p} - \zeta_{\mathsf{FIOP}}(\mathbb{x}) \ ,$$

where $t_{\mathsf{ARG}} = O(t_{\mathcal{P}})$. Moreover, $\mathcal{A}$ always rejects instances outside $L(R)$.

Corollary 4.9. *In the setting of Theorem 4.8, fix $\lambda := \lambda(|\mathbb{x}|)$ and suppose that FIOP and FC have perfect completeness. Suppose that $t_{\mathcal{E}}(\lambda, \mathbb{x}) = \mathsf{poly}(|\mathbb{x}|)$ and $p = \mathsf{poly}(|\mathbb{x}|) \geq 1$ for every instance $\mathbb{x}$. Further suppose $\zeta_{\mathsf{FIOP}}(\mathbb{x}) \leq 1/10$ for every $\mathbb{x} \in L(R)$. Then unless $L(R) \in \mathsf{BPP}$, the bound*

$$\kappa_{\mathsf{ARG}}(\lambda, \mathbb{x}, t_{\mathsf{ARG}}) > \frac{1}{5p}$$

holds for infinitely many $\mathbb{x} \in L(R)$, where $t_{\mathsf{ARG}} = O(t_{\mathcal{P}})$.

Theorem 4.10. *Let $\mathsf{FIOP} = (\mathbf{P}, \mathbf{V})$ be an FIOP for an NP relation R with query class $\mathbf{Q}$, completeness error γ_{FIOP}, answer domain $\mathbb{D}$, and query complexity q such that $|\mathbb{D}|^{\mathsf{q}} = \mathsf{poly}(|\mathbb{x}|)$ for every instance $\mathbb{x}$. Let FC be a public-coin functional commitment scheme for $\mathbf{Q}$ with completeness error γ_{FC}. Let $\theta \in (0, 1)$.*

Suppose there exists an FC-oblivious extractor $\mathcal{E}$ for $\mathsf{ARG} := \mathsf{Funky}[\mathsf{FIOP}, \mathsf{FC}]$ with running time $t_{\mathcal{E}}$ that rewinds the prover at most p times and achieves non-adaptive knowledge soundness error κ_{ARG}. Then there exists a probabilistic algorithm $\mathcal{A}$ that, on input $\mathbb{x}$, runs in time $\mathsf{poly}(|\mathbb{x}|, t_{\mathcal{E}})$ and, for every $\mathbb{x} \in L(R)$, accepts $\mathbb{x}$ with probability at least

$$1 - \gamma_{\mathsf{ARG}}(\lambda, |\mathbb{x}|) - \frac{\kappa_{\mathsf{ARG}}(\lambda, \mathbb{x}, t_{\mathsf{ARG}})}{\theta(1-\theta)^p} \ ,$$

where $t_{\mathsf{ARG}} = O(t_{\mathcal{P}})$. Moreover, $\mathcal{A}$ always rejects instances outside $L(R)$.

Corollary 4.11. *In the setting of Theorem 4.10, fix $\lambda := \lambda(|\mathbb{x}|)$ and suppose that FIOP and FC have perfect completeness. Suppose that $t_{\mathcal{E}}(\lambda, \mathbb{x}) = \mathsf{poly}(|\mathbb{x}|)$ and $p = \mathsf{poly}(|\mathbb{x}|) \geq 1$ for every instance $\mathbb{x}$. Then unless $L(R) \in \mathsf{BPP}$, the bound*

$$\kappa_{\mathsf{ARG}}(\lambda, \mathbb{x}, t_{\mathsf{ARG}}) > \frac{1}{5p}$$

holds for infinitely many $\mathbb{x} \in L(R)$, where $t_{\mathsf{ARG}} = O(t_{\mathcal{P}})$.

The following lemma is the main step towards proving Theorems 4.8 and 4.10.

Lemma 4.12. *Let $\mathcal{E}$ be an FC-oblivious extractor that rewinds at most p times and achieves non-adaptive knowledge soundness error κ_{ARG}. For any security parameter λ, instance-witness pair $(\mathbb{x}, \mathbb{w}) \in R$, and parameter $\theta \in (0, 1)$,*

$$\Pr\left[(\mathbb{x}, \mathbb{w}') \in R \;\middle|\; \begin{array}{l} (\boldsymbol{\rho}, \boldsymbol{\tau}) \leftarrow \mathsf{View}(\mathbf{P}, \mathbf{V}, \mathbb{x}, \mathbb{w}) \\ \mathbb{w}' \leftarrow \mathcal{E}^{\perp}(\mathbb{x}, (\boldsymbol{\rho}, \boldsymbol{\tau}, \mathbf{1}), 1) \end{array}\right] \geq 1 - \gamma_{\mathsf{ARG}}(\lambda, |\mathbb{x}|) - \frac{\kappa_{\mathsf{ARG}}(\lambda, \mathbb{x}, t_{\mathsf{ARG}})}{\theta(1-\theta)^p} \ , \tag{7}$$

where $t_{\mathsf{ARG}} = O(t_{\mathcal{P}})$. Above, $\mathcal{E}^{\perp}$ denotes an execution of $\mathcal{E}$ in which every rewind is answered with $\perp$.

Proof. Fix λ, $(\mathbb{x}, \mathbb{w})$, and θ as in the lemma statement. Consider the following argument prover.

$\widetilde{\mathcal{P}}_{\theta, \mathbb{x}, \mathbb{w}}(\mathsf{pp}, \eta = \perp)$:

1. Run $\mathcal{P}(\mathsf{pp}, \mathbb{x}, \mathbb{w})$ for the first k rounds, and let $\mathsf{aux}_{\mathsf{k}}$ denote the prover's resulting auxiliary state.
2. Flip a coin with heads probability θ:
   - On heads, run $\mathcal{P}(\mathsf{aux}_{\mathsf{k}})$.
   - On tails, halt (i.e., all future messages are $\perp$).

The running time of $\widetilde{\mathcal{P}}_{\theta,\mathbb{x},\mathbb{w}}$ is $t_{\mathsf{ARG}} = O(t_{\mathcal{P}})$. By knowledge soundness of $\mathcal{E}$ (Definition 4.1),

$$\Pr\left[\begin{array}{l|l} (\mathbb{x},\mathbb{w}') \notin R & \mathsf{pp} \leftarrow \mathcal{G}(1^\lambda, |\mathbb{x}|) \\ \wedge\, b = 1 & b \xleftarrow{\mathsf{tr}} \left\langle \widetilde{\mathcal{P}}_{\theta,\mathbb{x},\mathbb{w}}(\mathsf{pp},\bot), \mathcal{V}(\mathsf{pp},\mathbb{x}) \right\rangle \\ & \mathbb{w}' \leftarrow \mathcal{E}^{\widetilde{\mathcal{P}}^{\circ}_{\theta,\mathbb{x},\mathbb{w}}(\mathsf{pp},\bot)}(\mathbb{x},\mathsf{tr}^{\circ},b) \end{array}\right] \leq \kappa_{\mathsf{ARG}}(\lambda,\mathbb{x},t_{\mathsf{ARG}}) \ . \tag{8}$$

In the above experiment, let C_0 denote the prover's coin flip in the initial execution, and let $C_1,\ldots,C_p$ denote the prover's coin flips during extraction (unused coins are ignored). Since each coin flip is independent, the event

$$E \coloneqq \{C_0 \text{ heads},\ C_1,\ldots,C_p \text{ tails}\}$$

has probability $\theta(1-\theta)^p$.

Let σ be the success probability in the lemma statement; by construction, $\Pr[b=1 \wedge (\mathbb{x},\mathbb{w}') \in R \mid E] \leq \sigma$. By completeness, $\Pr[b=1 \mid E] \geq 1 - \gamma_{\mathsf{ARG}}(\lambda,|\mathbb{x}|)$. Hence, by Equation 8,

$$\begin{aligned} \kappa_{\mathsf{ARG}}(\lambda,\mathbb{x},t_{\mathsf{ARG}}) &\geq \Pr[E]\cdot\Pr[b=1\wedge(\mathbb{x},\mathbb{w}')\notin R \mid E] \\ &= \theta(1-\theta)^p\cdot\big(\Pr[b=1\mid E] - \Pr[b=1\wedge(\mathbb{x},\mathbb{w}')\in R\mid E]\big) \\ &\geq \theta(1-\theta)^p\cdot(1-\gamma_{\mathsf{ARG}}(\lambda,|\mathbb{x}|)-\sigma) \ . \end{aligned}$$

Rearranging proves the claim. □

Proof of Theorem 4.8. Let $\mathbf{S}$ be a simulator which achieves honest-verifier zero-knowledge error ζ_{FIOP} according to Definition 4.4. The decider $\mathcal{A}$ runs the FC-oblivious extractor on a simulated FIOP view.

$\mathcal{A}(\mathbb{x})$:
1. Sample $(\boldsymbol{\rho},\boldsymbol{\tau}) \leftarrow \mathbf{S}(\mathbb{x})$. If the output is not well-formed, return 0.
2. Run $\mathcal{E}^{\bot}(\mathbb{x},(\boldsymbol{\rho},\boldsymbol{\tau},\mathbf{1}),1)$. If it returns $\mathbb{w}$ with $(\mathbb{x},\mathbb{w})\in R$, return 1.
3. Return 0.

Since $\mathbf{S}$ runs in time $\mathsf{poly}(|\mathbb{x}|)$ and R is an NP relation, $\mathcal{A}$ runs in time $t_{\mathcal{E}} + \mathsf{poly}(|\mathbb{x}|)$. For $\mathbb{x}\notin L(R)$, no valid witness can be found, so $\mathcal{A}(\mathbb{x})$ always outputs 0.

Now fix $(\mathbb{x},\mathbb{w})\in R$. By Definition 4.4, the simulated view $\mathbf{S}(\mathbb{x})$ is $\zeta_{\mathsf{FIOP}}(\mathbb{x})$-close in statistical distance to $\mathsf{View}(\mathbf{P},\mathbf{V},\mathbb{x},\mathbb{w})$. Hence, by Lemma 4.12,

$$\Pr[\mathcal{A}(\mathbb{x})=1] \geq 1 - \gamma_{\mathsf{ARG}}(\lambda,|\mathbb{x}|) - \frac{\kappa_{\mathsf{ARG}}(\lambda,\mathbb{x},t_{\mathsf{ARG}})}{\theta(1-\theta)^p} - \zeta_{\mathsf{FIOP}}(\mathbb{x}) \ .$$

□

Proof of Corollary 4.9. Perfect completeness means that $\gamma_{\mathsf{ARG}}(\lambda,|\mathbb{x}|)=0$. By Bernoulli's inequality, $\theta(1-\theta)^p \geq \theta(1-p\theta)$ holds for every $\theta\in(0,1)$. Setting $\theta \coloneqq 1/(2p)$, we obtain

$$\theta(1-\theta)^p \geq \frac{1}{2p}\Big(1-\frac{p}{2p}\Big) = \frac{1}{4p} \ . \tag{9}$$

Suppose there is a finite set $S\subseteq L(R)$ such that $\kappa_{\mathsf{ARG}}(\lambda,\mathbb{x},t_{\mathsf{ARG}}) \leq 1/(5p)$ holds for every $\mathbb{x}\in L(R)\setminus S$. By Theorem 4.8, there exists a strict polynomial-time algorithm $\mathcal{A}$ that always rejects instances outside $L(R)$ and, for every $\mathbb{x}\in L(R)\setminus S$, satisfies

$$\Pr[\mathcal{A}(\mathbb{x})=1] \geq 1 - \frac{1}{5p}\cdot 4p - \frac{1}{10} = \frac{1}{10} \ .$$

This yields a BPP decider for $L(R)$ as follows. Hard-code S and accept any input in S. On all other inputs, repeat $\mathcal{A}$ independently a constant number of times and accept if any repetition is accepting. □

Proof of Theorem 4.10. The decider $\mathcal{A}$ runs the FC-oblivious extractor on all possible FIOP views.

$\mathcal{A}(\mathbb{x})$:
1. Sample $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_{\mathsf{k}})$, where $\rho_i \leftarrow \{0,1\}^{\mathsf{r}_i}$. Additionally sample random tape r.
2. For each $(\beta_1, \ldots, \beta_{\mathsf{q}}) \in \mathbb{D}^{\mathsf{q}}$:
   (a) Run $\mathbf{V}(\mathbb{x}; \boldsymbol{\rho})$, answering the i-th query with β_i. Let $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{\mathsf{k}})$ be the resulting query-answer tuples.
   (b) Compute $\mathbb{w} \leftarrow \mathcal{E}^{\perp}(\mathbb{x}, (\boldsymbol{\rho}, \boldsymbol{\tau}, \mathbf{1}), 1; r)$.
   (c) If $(\mathbb{x}, \mathbb{w}) \in R$, return 1.
3. Return 0.

Since $|\mathbb{D}|^{\mathsf{q}} = \mathsf{poly}(|\mathbb{x}|)$, $\mathcal{A}$ runs in time $\mathsf{poly}(t_{\mathcal{E}}, |\mathbb{x}|)$. For $\mathbb{x} \notin L(R)$, no valid witness can be found, so $\mathcal{A}(\mathbb{x})$ always outputs 0.

Now fix $(\mathbb{x}, \mathbb{w}) \in R$. Then in some iteration, $\mathcal{A}$ runs $\mathcal{E}$ on a view distributed according to $\mathsf{View}(\mathbf{P}, \mathbf{V}, \mathbb{x}, \mathbb{w})$ (or it has already found a valid witness). Hence, by Lemma 4.12,

$$\Pr[\mathcal{A}(\mathbb{x}) = 1] \geq 1 - \gamma_{\mathsf{ARG}}(\lambda, |\mathbb{x}|) - \frac{\kappa_{\mathsf{ARG}}(\lambda, \mathbb{x}, t_{\mathsf{ARG}})}{\theta(1-\theta)^p} \quad .$$

□

Proof of Corollary 4.11. Perfect completeness means that $\gamma_{\mathsf{ARG}}(\lambda, |\mathbb{x}|) = 0$. Set $\theta := 1/(2p)$, so that $\theta(1-\theta)^p \geq 1/(4p)$ by Equation 9. Suppose there is a finite set $S \subseteq L(R)$ such that $\kappa_{\mathsf{ARG}}(\lambda, \mathbb{x}, t_{\mathsf{ARG}}) \leq 1/(5p)$ holds for every $\mathbb{x} \in L(R) \setminus S$. By Theorem 4.10, there exists a strict polynomial-time algorithm $\mathcal{A}$ that always rejects instances outside $L(R)$ and, for every $\mathbb{x} \in L(R) \setminus S$, satisfies

$$\Pr[\mathcal{A}(\mathbb{x}) = 1] \geq 1 - \frac{1}{5p} \cdot 4p = \frac{1}{5} \quad .$$

This yields a BPP decider for $L(R)$ as follows. Hard-code S and accept any input in S. On all other inputs, repeat $\mathcal{A}$ independently a constant number of times and accept if any repetition is accepting. □

Remark 4.13 (Deterministic provers)**.** The prover $\widetilde{\mathcal{P}}_{\theta,\mathbb{x},\mathbb{w}}$ defined in the proof of Lemma 4.12 samples a new coin on every rewinding call on a challenge of the last round. Our barriers also hold for deterministic provers (see Remark 3.8). In particular, consider an auxiliary input distribution $\mathcal{D}$ which samples a key for a $(p+1)$-wise independent hash function H with $\Pr[H(\boldsymbol{\rho}) = 1] = \theta$ for every $\boldsymbol{\rho}$. The use of $(p+1)$-wise independent hash functions follows [GK96; BL04].

Remark 4.14 (Comparison with prior extractors)**.** Our barriers do not contradict the knowledge soundness analyses of Kilian's protocol in [CMSZ21; LM19]. The extractor in [CMSZ21] runs in expected polynomial time and receives as input $1^{1/\epsilon}$, where ϵ is a lower bound on the probability that the prover convinces the verifier, and it outputs a witness with probability $\Omega(\epsilon - \kappa_{\mathsf{PCP}})$. In [LM19], for every polynomial-time prover there is a polynomial-time extractor that outputs a witness with non-negligible probability whenever the prover convinces the verifier with non-negligible probability, and the polynomial bounding the running time of the extractor depends on the prover. Neither is a single strict polynomial-time extractor whose running time and number of rewinds are independent of the prover and of its success probability, which is what Theorems 4.8 and 4.10 rule out.

4.3 Expected-time barriers

Theorem 4.15. *Let* $\mathsf{FIOP} = (\mathbf{P}, \mathbf{V})$ *be an FIOP for an* NP *relation* R *with query class* $\mathbf{Q}$*, completeness error* γ_{FIOP}*, answer domain* $\mathbb{D}$*, and query complexity* q*; suppose that* q *is at least the round complexity of* FIOP*. Let* FC *be a public-coin functional commitment scheme for* $\mathbf{Q}$ *with completeness error* γ_{FC}*.*

Suppose there exists an FC-oblivious extractor $\mathcal{E}$ for $\mathsf{ARG} \coloneqq \mathsf{Funky}[\mathsf{FIOP}, \mathsf{FC}]$ with expected running time $t^\star_{\mathcal{E}}$ that rewinds the prover $p^\star \geq 1$ times in expectation and achieves non-adaptive knowledge soundness error $\kappa^\star_{\mathsf{ARG}}$ satisfying

$$\kappa^\star_{\mathsf{ARG}}(\lambda, \mathbb{x}, t^\star_{\mathsf{ARG}}) \leq 1 - \gamma_{\mathsf{ARG}}(\lambda, |\mathbb{x}|) - c$$

for some constant $c > 0$ and every $\mathbb{x} \in L(R)$, where $t^\star_{\mathsf{ARG}} = O(t_{\mathcal{P}})$. Then there exists a probabilistic algorithm $\mathcal{A}$ that, on input $\mathbb{x}$, runs in (strict) time at most

$$\mathsf{poly}(t^\star_{\mathcal{E}}, |\mathbb{x}|) \cdot 2^{O(p^\star \cdot \mathsf{q} \log|\mathbb{D}|)}$$

and, for every $\mathbb{x} \in L(R)$, accepts $\mathbb{x}$ with probability at least $c/2$. Moreover, $\mathcal{A}$ always rejects instances outside $L(R)$.

Corollary 4.16. *In the setting of Theorem 4.15, fix $\lambda \coloneqq \lambda(|\mathbb{x}|)$ and suppose that FIOP and FC have perfect completeness. Suppose that*

$$t^\star_{\mathcal{E}}(\lambda, \mathbb{x}) = \mathsf{poly}(|\mathbb{x}|), \quad p^\star = \frac{f(|\mathbb{x}|)}{\mathsf{q} \log|\mathbb{D}|}$$

for every instance $\mathbb{x}$, where $f(n) = \Omega(\log n)$. Further suppose $\kappa^\star_{\mathsf{ARG}}(\lambda, \mathbb{x}, t^\star_{\mathsf{ARG}}) \leq 1 - c$ for some constant $c > 0$ and every $\mathbb{x} \in L(R)$, where $t^\star_{\mathsf{ARG}} = O(t_{\mathcal{P}})$. Then there exists a probabilistic algorithm that, on input $\mathbb{x}$, runs in (strict) time $2^{O(f(|\mathbb{x}|))}$ and decides $L(R)$ with probability at least $2/3$.

For Kilian's protocol, the extractor of Corollary 6.2 rewinds the prover $O(\ell)$ times, where ℓ is the PCP length (here we set α to be a constant). Compared with Corollary 4.16, the multiplicative gap is

$$O\left(\frac{\ell \cdot \mathsf{q} \log|\Sigma|}{f(n)}\right) \ .$$

Consider the PCP for 3SAT from [Din07], which has $\mathsf{q} = O(1)$, $|\Sigma| = O(1)$, and $\ell = n \cdot \mathsf{polylog}(n)$ for $n = |\mathbb{x}|$. Let $f(n) = n/g(n)$ for some $g(n) = \mathsf{polylog}(n)$. Then by Corollary 4.16, the number of rewinds of our extractor in Corollary 6.2 is at most a $\mathsf{polylog}(n)$ factor away from optimal, unless there exists a probabilistic algorithm that decides 3SAT in time $2^{O(n/g(n))}$.

Proof of Theorem 4.15. Let $\bar{p} \coloneqq \lceil 4p^\star/c \rceil$ and $\bar{t} \coloneqq \lceil 4t^\star_{\mathcal{E}}/c \rceil$. The decider $\mathcal{A}$ runs the FC-oblivious extractor on all possible FIOP views, as in the proof of Theorem 4.10, and answers each rewind in all possible ways, while truncating the number of rewinds at $\bar{p}$ and the running time at $\bar{t}$.

$\mathcal{A}(\mathbb{x})$:

1. Sample $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_{\mathsf{k}})$, where $\rho_i \leftarrow \{0,1\}^{\mathsf{r}_i}$. Additionally sample random tape r.
2. For each $\boldsymbol{\beta} \in \mathbb{D}^{\mathsf{q}}$ and $(\boldsymbol{\beta}'_1, \boldsymbol{b}'_1, \ldots, \boldsymbol{\beta}'_{\bar{p}}, \boldsymbol{b}'_{\bar{p}}) \in (\mathbb{D}^{\mathsf{q}} \times \{0,1\}^{\mathsf{k}})^{\bar{p}}$:
   (a) Run $\mathbf{V}(\mathbb{x}; \boldsymbol{\rho})$, answering the i-th query with the i-th entry of $\boldsymbol{\beta}$. Let $\boldsymbol{\tau}$ be the resulting query-answer tuples.
   (b) Run $\mathcal{E}(\mathbb{x}, (\boldsymbol{\rho}, \boldsymbol{\tau}, \mathbf{1}), 1; r)$ for at most $\bar{t}$ steps and at most $\bar{p}$ rewinds, answering its j-th rewind as follows. If the extractor rewinds with a full set of challenges $\boldsymbol{\rho}' = (\rho'_1, \ldots, \rho'_{\mathsf{k}})$, run $\mathbf{V}(\mathbb{x}; \boldsymbol{\rho}')$, answering the i-th query with the i-th entry of $\boldsymbol{\beta}'_j$ and give the extractor $(\boldsymbol{\tau}', \boldsymbol{b}'_j)$, where $\boldsymbol{\tau}' = (\tau'_1, \ldots, \tau'_{\mathsf{k}})$ consists of the resulting query-answer tuples; otherwise, give the extractor $\perp$.
   (c) If the extractor returns $\mathbb{w}$ with $(\mathbb{x}, \mathbb{w}) \in R$, return 1.
3. Return 0.

Step 2 takes at most $|\mathbb{D}|^{O(\bar{p} \cdot \mathsf{q})}$ iterations, and each iteration runs in at most $\mathsf{poly}(\bar{t}, |\mathbb{x}|)$ time. Since c is a constant, $\bar{t} = O(t^\star_{\mathcal{E}})$ and $\bar{p} = O(p^\star)$. Thus, $\mathcal{A}$ runs in time at most

$$\mathsf{poly}(t^\star_{\mathcal{E}}, |\mathbb{x}|) \cdot 2^{O(p^\star \cdot \mathsf{q} \cdot \log|\mathbb{D}|)} \ .$$

For $\mathbb{x} \notin L(R)$, no valid witness can be found, so $\mathcal{A}(\mathbb{x})$ always outputs 0.

Now fix $(\mathbb{x}, \mathbb{w}) \in R$, and let $\widetilde{\mathcal{P}}$ be the prover that, on input $(\mathsf{pp}, \eta = \bot)$, runs the honest prover $\mathcal{P}(\mathsf{pp}, \mathbb{x}, \mathbb{w})$; it runs in time $t^\star_{\mathsf{ARG}} = O(t_{\mathcal{P}})$. Let $\boldsymbol{\rho}$ and r be sampled by $\mathcal{A}$. Consider the experiment of Definition 4.2 with instance $\mathbb{x}$, auxiliary input $\bot$, prover $\widetilde{\mathcal{P}}$, and the extractor induced by $\mathcal{E}$, in which $\mathcal{V}$ sends the challenges $\boldsymbol{\rho}$ and $\mathcal{E}$ uses the random tape r. Let $\mathbb{w}'$, $P^\star$, and $T^\star$ denote the output, number of rewinds, and number of steps of $\mathcal{E}$, respectively. Condition on $b = 1$, $P^\star \leq \bar{p}$, and $T^\star \leq \bar{t}$. Then in some iteration, $\mathcal{A}$ runs $\mathcal{E}$ on the same input and answers each rewind in the same way as in the experiment (or it has already found a valid witness). Thus,

$$\begin{aligned}\Pr[\mathcal{A}(\mathbb{x}) = 1] &\geq \Pr\big[(\mathbb{x}, \mathbb{w}') \in R \wedge b = 1 \wedge P^\star \leq \bar{p} \wedge T^\star \leq \bar{t}\big] \\ &\geq \Pr[b = 1] - \Pr\big[(\mathbb{x}, \mathbb{w}') \notin R \wedge b = 1\big] - \Pr[P^\star > \bar{p}] - \Pr[T^\star > \bar{t}] \ .\end{aligned}$$

By completeness, $\Pr[b = 1] \geq 1 - \gamma_{\mathsf{ARG}}(\lambda, |\mathbb{x}|)$. By expected-time knowledge soundness of $\mathcal{E}$ (Definition 4.2), $\Pr[(\mathbb{x}, \mathbb{w}') \notin R \wedge b = 1] \leq \kappa^\star_{\mathsf{ARG}}(\lambda, \mathbb{x}, t^\star_{\mathsf{ARG}})$. Finally, recall that $\mathbb{E}[P^\star] \leq p^\star$ and $\mathbb{E}[T^\star] \leq t^\star_{\mathcal{E}}$. By Markov's inequality,

$$\begin{aligned}\Pr[\mathcal{A}(\mathbb{x}) = 1] &\geq 1 - \gamma_{\mathsf{ARG}}(\lambda, |\mathbb{x}|) - \kappa^\star_{\mathsf{ARG}}(\lambda, \mathbb{x}, t^\star_{\mathsf{ARG}}) - \frac{c}{4} - \frac{c}{4} \\ &\geq c - \frac{c}{4} - \frac{c}{4} = \frac{c}{2} \ .\end{aligned}$$

□

Proof of Corollary 4.16. Perfect completeness means that $\gamma_{\mathsf{ARG}}(\lambda, |\mathbb{x}|) = 0$. By Theorem 4.15, there exists a probabilistic algorithm $\mathcal{A}$ that always rejects instances outside $L(R)$ and accepts every $\mathbb{x} \in L(R)$ with probability at least $c/2$. On input $\mathbb{x}$, its running time is at most

$$\mathsf{poly}(t^\star_{\mathcal{E}}, |\mathbb{x}|) \cdot 2^{O(p^\star \cdot \mathsf{q} \cdot \log|\mathbb{D}|)} = 2^{O(f(|\mathbb{x}|))} \ .$$

Repeating $\mathcal{A}$ independently $\lceil 4/c \rceil$ times (and accepting if any repetition is accepting) yields a decider for $L(R)$ that runs in time at most $2^{O(f(|\mathbb{x}|))}$ and succeeds with probability at least $2/3$. □

5 Expected-time soundness

Theorem 5.1. *Let* FIOP *be a public-coin FIOP for a relation* R *with query class* $\mathbf{Q}$*, round complexity* k*, proof length* ℓ*, query complexity* q*, and soundness error* ϵ_{FIOP}*; suppose that* $\mathbf{Q}$ *has finite diameter* $d_{\mathbf{Q}}$ *and admits a constraint solver with operation time* $t_{\mathbf{Q}}$*. Let* FC *be a public-coin interactive functional commitment scheme for* $\mathbf{Q}$ *with expected-time function binding error* $\epsilon^{\star}_{\mathsf{FC}}$*.*

Then $\mathsf{ARG} \coloneqq \mathsf{Funky}[\mathsf{FIOP}, \mathsf{FC}]$ *(Construction 3.25) is a public-coin interactive argument for* R *whose expected-time soundness error* $\epsilon^{\star}_{\mathsf{ARG}}$ *satisfies the following for every* $t^{\star}_{\mathsf{ARG}} \geq t_{\mathcal{V}} + \max_{i \in [\mathsf{k}]} \mathsf{q}_i \cdot t_{\mathbf{Q}}(\ell_i)$*:*

$$\epsilon^{\star}_{\mathsf{ARG}}(\lambda, n, t^{\star}_{\mathsf{ARG}}) \leq \epsilon_{\mathsf{FIOP}}(n) + \epsilon^{\star}_{\mathsf{FC}}(\lambda, \ell_{\mathsf{max}}, \mathsf{q}_{\mathsf{max}}, t^{\star}_{\mathsf{FC}}) \ ,$$

where $t^{\star}_{\mathsf{FC}} = O(N \cdot t^{\star}_{\mathsf{ARG}})$ *for* $N \coloneqq 1 + \sum_{i=1}^{\mathsf{k}} d_{\mathbf{Q}}(\ell_i)$*. If the verifier of* FIOP *always issues a* q^*_i*-independent query set (Definition 5.2) to the* i*-th proof string, then the above holds for* $N \coloneqq 1 + \sum_{i=1}^{\mathsf{k}} \big(d_{\mathbf{Q}}(\ell_i) - \max\{\mathsf{q}^*_i - 1, 0\}\big)$*.*

Theorem 5.1 establishes slightly tighter bounds for FIOPs with *independent queries*, which we now define.

Definition 5.2 (Independent queries)**.** *A query-answer tuple* τ *is* m**-independent** *if* $|\mathcal{C}_\tau| \geq m$*. A query set* $\boldsymbol{\alpha}$ *is* m*-independent if* $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ *is* m*-independent for every* $\boldsymbol{\beta} \in \mathbb{D}^{\boldsymbol{\alpha}}$*.*

Throughout, let $\mathsf{FIOP} = (\mathbf{P}, \mathbf{V})$ and $\mathsf{ARG} = (\mathcal{G}, \mathcal{P}, \mathcal{V})$. Let $\mathsf{Solver}_{\mathbf{Q}}$ be a constraint solver for $\mathbf{Q}$ with operation time $t_{\mathbf{Q}}$. We prove Theorem 5.1's second bound, which implies the first bound by setting all $\mathsf{q}^*_i = 0$.

5.1 Adversary construction

We construct an adversary for FC which will be used to analyze the expected-time soundness of ARG.

Construction 5.3 (Sampler)**.** For a round $i \in [\mathsf{k}]$, the sampler $\mathcal{S}_i$ works as follows.

$\mathcal{S}_i^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, \rho_1, \ldots, \rho_{i-1}, \mathsf{cm}_i)$:

1. Loop:
   (a) For each $j \in [i, \mathsf{k}]$, sample verifier randomness $\rho_j \leftarrow \{0,1\}^{\mathsf{r}_j}$.
   (b) Sample an execution of the prover $\widetilde{\mathcal{P}}(\mathsf{aux})$ as follows. In the FIOP phase, use randomness $\rho_1, \ldots, \rho_{\mathsf{k}}$; the resulting (partial) interaction transcript is of the form

   $$(\mathsf{cm}_1, \rho_1, \ldots, \mathsf{cm}_{\mathsf{k}}, \rho_{\mathsf{k}}, (\tau_1, \ldots, \tau_{\mathsf{k}})) \ ,$$

   where $\tau_j = (\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j)$ is the query-answer tuple for the j-th commitment. In the opening phase, for each $j \in [\mathsf{k}]$, interact with FCGame: send $(\mathsf{cm}_j, \boldsymbol{\alpha}_j, \boldsymbol{\beta}_j)$ and then relay messages between $\widetilde{\mathcal{P}}(\mathsf{aux})$ and FCGame.
   (c) If the above execution is accepting, return τ_i.

Construction 5.4 (Reductor)**.** The reductor $\mathcal{R}$ works as follows.

$\mathcal{R}^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, \mathsf{tr})$:

1. Parse the transcript $\mathsf{tr} = (\mathsf{cm}_1, \rho_1, \ldots, \mathsf{cm}_{\mathsf{k}}, \rho_{\mathsf{k}}, (\tau_1, \ldots, \tau_{\mathsf{k}}), \mathsf{pf}_1, \ldots, \mathsf{pf}_{\mathsf{k}})$.
2. For each $i \in [\mathsf{k}]$:
   (a) Initialize an empty list $T_i \coloneqq \emptyset$.
   (b) Loop while $\mathcal{C}_{T_i}$ does not decide τ_i:
      i. Sample $\tau'_i \leftarrow \mathcal{S}_i^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, \rho_1, \ldots, \rho_{i-1}, \mathsf{cm}_i)$.
      ii. Append $T_i \coloneqq T_i \parallel \tau'_i$.

Construction 5.5 (FC adversary)**.** For an argument prover $\widetilde{\mathcal{P}}$, the adversary $\mathcal{A}^{\widetilde{\mathcal{P}}}_{\mathsf{FC}}$ interacts with $\mathsf{FCGame}(\mathsf{pp})$ as follows.

$\mathcal{A}^{\widetilde{\mathcal{P}}}_{\mathsf{FC}}(\mathsf{pp}, \eta)$:

1. Compute $(\mathbb{x}, \mathsf{aux}) \leftarrow \widetilde{\mathcal{P}}(\mathsf{pp}, \eta)$. If $|\mathbb{x}| > n$, halt.
2. Sample an execution of the prover $\widetilde{\mathcal{P}}(\mathsf{aux})$ as follows. In the FIOP phase, sample verifier randomness $\rho_1, \ldots, \rho_{\mathsf{k}}$. In the opening phase, interact with FCGame as in Step 1b. Let b and tr be the resulting decision bit and interaction transcript, respectively.
3. If $b = 1$, run $\mathcal{R}^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, \mathsf{tr})$.

5.2 Security analysis

Throughout, fix an argument prover $\widetilde{\mathcal{P}}$ and auxiliary input distribution $\mathcal{D}$. We consider the experiment

$$E \coloneqq \left\{ \begin{array}{l} \mathsf{pp} \leftarrow \mathcal{G}(1^\lambda, n) \\ \eta \leftarrow \mathcal{D} \\ (\mathbb{x}, \mathsf{aux}) \leftarrow \widetilde{\mathcal{P}}(\mathsf{pp}, \eta) \\ b \xleftarrow{\mathsf{tr}} \left\langle \widetilde{\mathcal{P}}(\mathsf{aux}), \mathcal{V}(\mathsf{pp}, \mathbb{x}) \right\rangle \end{array} \right\},$$

where $\mathsf{tr} = (\mathsf{cm}_1, \rho_1, \ldots, \mathsf{cm}_{\mathsf{k}}, \rho_{\mathsf{k}}, (\tau_1, \ldots, \tau_{\mathsf{k}}), \mathsf{pf}_1, \ldots, \mathsf{pf}_{\mathsf{k}})$. Let $t^\star_{\mathsf{ARG}}$ denote the expected running time of $\widetilde{\mathcal{P}}$ above; throughout, we assume the lower bound on $t^\star_{\mathsf{ARG}}$ from Theorem 5.1.

We also consider the experiment

$$E_{\mathsf{FC}} \coloneqq \left\{ \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{FC.Gen}(1^\lambda, \ell_{\mathsf{max}}) \\ \eta \leftarrow \mathcal{D} \\ \left\langle \mathcal{A}^{\widetilde{\mathcal{P}}}_{\mathsf{FC}}(\mathsf{pp}, \eta), \mathsf{FCGame}(\mathsf{pp}) \right\rangle \end{array} \right\}.$$

When $\mathcal{A}^{\widetilde{\mathcal{P}}}_{\mathsf{FC}}$ does not run $\mathcal{R}$, the lists $T_1, \ldots, T_{\mathsf{k}}$ are taken to be empty.

Definition 5.6. *For a public parameter* pp *and auxiliary input* η*, we define:*

$$t^\star_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<i}) \coloneqq \mathop{\mathbb{E}}_{E}\Big[\textit{running time of } \widetilde{\mathcal{P}} \;\Big|\; \mathsf{pp}, \eta, \boldsymbol{\rho}_{<i}\Big]$$
$$\epsilon_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<i}) \coloneqq \Pr_{E}[b = 1 \mid \mathsf{pp}, \eta, \boldsymbol{\rho}_{<i}] \ ,$$

where $\boldsymbol{\rho}_{<i}$ *denotes* $(\rho_1, \ldots, \rho_{i-1})$*. When* $i = 1$*, we write* $t^\star_{\mathsf{pp},\eta}$ *and* $\epsilon_{\mathsf{pp},\eta}$ *for brevity.*

5.2.1 Running time

We analyze the expected running time of the adversary $\mathcal{A}_{\mathsf{FC}}$ (Construction 5.5).

Lemma 5.7. *Let* $\mathbf{Q}$ *be a query class with finite diameter* $d_{\mathbf{Q}}$*, and let* D *be a distribution over query-answer tuples. Let* $X_0, X_1, X_2, \ldots$ *be an infinite sequence of samples independently drawn from* D*, and, for* $n \geq 0$*, let* $\mathcal{C}_n \coloneqq \mathcal{C}_{(X_1,\ldots,X_n)}$ *(with* $\mathcal{C}_0 = \emptyset$*). Then the random variable*

$$N \coloneqq \min\{n \geq 0 : \mathcal{C}_n \textit{ decides } X_0\}$$

satisfies $\mathbb{E}[N] \leq d_{\mathbf{Q}}$*. If every query-answer tuple in the support of* D *is* m*-independent, then* $\mathbb{E}[N] \leq d_{\mathbf{Q}} - m + 1$*.*

Proof. For $n \geq 0$, we say that X_{n+1} is *new* if $\mathcal{C}_n$ does not decide X_{n+1}. Observe that the event $N > n$ occurs if and only if some constraint of X_0 is undecided by $\mathcal{C}_n$. Since the samples are i.i.d. and $\mathcal{C}_n$ is a function of $(X_1, \ldots, X_n)$ only,

$$\Pr[N > n] = \Pr[\mathcal{C}_n \text{ does not decide } X_0] = \Pr[\mathcal{C}_n \text{ does not decide } X_{n+1}] = \Pr[X_{n+1} \text{ new}] \ .$$

It follows that

$$\mathbb{E}[N] = \sum_{n \geq 0} \Pr[N > n] = \sum_{n \geq 0} \Pr[X_{n+1} \text{ new}] = \mathbb{E}[\text{number of new samples}] \ ,$$

where the final equality is by linearity of expectation. We conclude by observing that there are at most $d_{\mathbf{Q}}$ new samples across the sequence (each new sample appends at least one constraint, and at most $d_{\mathbf{Q}}$ constraints can be appended in total). If every query-answer tuple in the support of D is m-independent, then there are at most $d_{\mathbf{Q}} - m + 1$ new samples (since X_1 always contributes at least m constraints). □

Lemma 5.8 (Expected running time)**.** *The expected running time of $\mathcal{A}^{\widetilde{\mathcal{P}}}_{\mathsf{FC}}$ in E_{FC} is at most $O(N \cdot t^{\star}_{\mathsf{ARG}})$ for $N \coloneqq 1 + \sum_{i=1}^{\mathsf{k}} \left(d_{\mathbf{Q}}(\ell_i) - \max\{\mathsf{q}_i^* - 1, 0\}\right)$.*

Proof. For simplicity, we focus on the time spent executing $\widetilde{\mathcal{P}}$; the remaining time can be bounded with some additional accounting. Fix a public parameter pp and auxiliary input η, which together determine the instance $\mathbb{x}$ and prover state aux. Without loss of generality, assume $|\mathbb{x}| \leq n$.

Fix a round $i \in [\mathsf{k}]$, and let N_i denote the number of times the sampler $\mathcal{S}_i$ is invoked. Fix verifier randomness $\boldsymbol{\rho}_{<i} = (\rho_1, \ldots, \rho_{i-1})$, which determines the commitment cm_i. If $\epsilon_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<i}) = 0$, then $N_i = 0$; hence assume it is positive. The expected running time of an invocation

$$\mathcal{S}_i^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, \rho_1, \ldots, \rho_{i-1}, \mathsf{cm}_i)$$

is $t^{\star}_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<i}) / \epsilon_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<i})$. By Wald's identity,

$$\mathbb{E}[\text{time spent in } \mathcal{S}_i \mid \mathsf{pp}, \eta, \boldsymbol{\rho}_{<i}] = \mathbb{E}[N_i \mid \mathsf{pp}, \eta, \boldsymbol{\rho}_{<i}] \cdot \frac{t^{\star}_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<i})}{\epsilon_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<i})} \ .$$

Observe that $N_i = 0$ unless $b = 1$. Thus, by total probability,

$$\mathbb{E}[N_i \mid \mathsf{pp}, \eta, \boldsymbol{\rho}_{<i}] \leq \mathbb{E}[N_i \mid \mathsf{pp}, \eta, \boldsymbol{\rho}_{<i}, b = 1] \cdot \epsilon_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<i}) \ .$$

Conditioned on $b = 1$, the initial query-answer tuple τ_i follows the output distribution of $\mathcal{S}_i$. By Lemma 5.7 (where D is the output distribution of $\mathcal{S}_i$),

$$\mathbb{E}[N_i \mid \mathsf{pp}, \eta, \boldsymbol{\rho}_{<i}, b = 1] \leq d_{\mathbf{Q}}(\ell_i) - \max\{\mathsf{q}_i^* - 1, 0\} \ .$$

Putting everything together and averaging over $\boldsymbol{\rho}_{<i}$, we get

$$\mathbb{E}[\text{time spent in } \mathcal{S}_i \mid \mathsf{pp}, \eta] \leq (d_{\mathbf{Q}}(\ell_i) - \max\{\mathsf{q}_i^* - 1, 0\}) \cdot t^{\star}_{\mathsf{pp},\eta} \ ;$$

summing over $i \in [\mathsf{k}]$ and averaging over pp and η bounds the expected running time of the reductor $\mathcal{R}$. Accounting also for the initial execution of $\widetilde{\mathcal{P}}$ gives the lemma. □

5.2.2 Success probability

Definition 5.9. *For a (possibly infinite) list of query-answer tuples T, let $\widetilde{\Pi}_T$ denote the lexicographically first element of* $\mathrm{Sol}(\mathcal{C}_T)$. *(When $T = (\tau_n)_{n\in\mathbb{N}}$ is infinite, $\mathcal{C}_T$ is defined to be the eventual value of $\mathcal{C}_{(\tau_1,\ldots,\tau_n)}$, which exists since d_{Q} is finite.)*

Definition 5.10. *For a public parameter* pp *and auxiliary input η, we define $\mathbf{X}_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<i})$ to be an infinite sequence of samples independently drawn from the output distribution of*

$$\mathcal{S}_i^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, \rho_1, \ldots, \rho_{i-1}, \mathsf{cm}_i) \ ,$$

where $\boldsymbol{\rho}_{<i} = (\rho_1, \ldots, \rho_{i-1})$ for some round $i \in [\mathsf{k}]$. (If the sampler never halts, we take $\mathbf{X}_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<i})$ to be empty.) When pp *and η are clear from context, we write $\mathbf{X}(\boldsymbol{\rho}_{<i})$.*

Proof of Theorem 5.1. Fix an argument prover $\widetilde{\mathcal{P}}$ and auxiliary input distribution $\mathcal{D}$. Let E^+ be the extension of E that additionally samples

$$b' \leftarrow \mathbf{V}^{\widetilde{\Pi}_{\mathbf{X}(\boldsymbol{\rho}_{<1})},\ldots,\widetilde{\Pi}_{\mathbf{X}(\boldsymbol{\rho}_{<\mathsf{k}})}}(\mathbb{x}; \rho_1, \ldots, \rho_{\mathsf{k}}) \ .$$

By total probability,

$$\Pr_{E}\left[\begin{array}{l} |\mathbb{x}| \le n \\ \wedge\, \mathbb{x} \notin L(R) \\ \wedge\, b = 1 \end{array}\right] \le \underbrace{\Pr_{E^+}\left[\begin{array}{l} |\mathbb{x}| \le n \\ \wedge\, \mathbb{x} \notin L(R) \\ \wedge\, b' = 1 \end{array}\right]}_{A} + \underbrace{\Pr_{E^+}\left[\begin{array}{l} |\mathbb{x}| \le n \\ \wedge\, b = 1 \\ \wedge\, b' = 0 \end{array}\right]}_{B} \ .$$

Bounding A. Define the FIOP prover $\widetilde{\mathbf{P}}$ which, on input pp and η, computes $(\mathbb{x}, \mathsf{aux}) \leftarrow \widetilde{\mathcal{P}}(\mathsf{pp}, \eta)$ and outputs the instance $\mathbb{x}$. In the i-th round, having received $\boldsymbol{\rho}_{<i}$, it samples and sends a proof string according to the distribution $\widetilde{\Pi}_{\mathbf{X}(\boldsymbol{\rho}_{<i})}$. Observe that A is equal to the success probability of $\widetilde{\mathbf{P}}$ (with an auxiliary input distribution $\mathbf{D}$ that samples public parameter $\mathsf{pp} \leftarrow \mathcal{G}(1^\lambda, n)$, auxiliary input $\eta \leftarrow \mathcal{D}$, and additional random coins). By soundness of FIOP (Definition 3.23),

$$A \le \epsilon_{\mathsf{FIOP}}(n) \ .$$

Bounding B. Define the random variables $T'_1, \ldots, T'_{\mathsf{k}}$ in E^+ as follows. If $|\mathbb{x}| \le n$ and $b = 1$, then T'_i is the shortest prefix of $\mathbf{X}(\boldsymbol{\rho}_{<i})$ such that $\mathcal{C}_{T'_i}$ decides τ_i; such a prefix exists by Lemma 5.7. Otherwise, T'_i is an empty list. Observe that the following distributions are identical:

$$\left\{\mathsf{tr}, T'_1, \ldots, T'_{\mathsf{k}} \;\middle|\; E^+, |\mathbb{x}| \le n, b = 1\right\} \sim \left\{\mathsf{tr}, T_1, \ldots, T_{\mathsf{k}} \;\middle|\; E_{\mathsf{FC}}, |\mathbb{x}| \le n, b = 1\right\} \ .$$

In E^+, suppose that $b = 1$ and $b' = 0$. Then in some round $i \in [\mathsf{k}]$, it must be that $\widetilde{\Pi}_{\mathbf{X}(\boldsymbol{\rho}_{<i})}$ disagrees with τ_i, and hence $\mathcal{C}_{T'_i}$ *refutes* τ_i. Thus,

$$B \le \Pr_{E^+}\left[\begin{array}{l} |\mathbb{x}| \le n \\ \wedge\, b = 1 \\ \wedge\, \exists i, \mathcal{C}_{T'_i} \text{ refutes } \tau_i \end{array}\right] = \Pr_{E_{\mathsf{FC}}}\left[\begin{array}{l} |\mathbb{x}| \le n \\ \wedge\, b = 1 \\ \wedge\, \exists i, \mathcal{C}_{T_i} \text{ refutes } \tau_i \end{array}\right] \ ,$$

In E_{FC}, observe that $\mathcal{C}_{T_i}$ refuting τ_i implies $\mathrm{Sol}(\mathcal{C}_{\mathsf{cm}_i,\ell_i,\mathsf{q}_i}) = \emptyset$ (see Definition 3.17). By expected-time function binding of FC (Definition 3.19) and Lemma 5.8,

$$B \le \epsilon^\star_{\mathsf{FC}}(\lambda, \ell_{\mathsf{max}}, \mathsf{q}_{\mathsf{max}}, t^\star_{\mathsf{FC}}) \ ,$$

where $t^\star_{\mathsf{FC}}$ is as stated in the theorem. □

6 Expected-time knowledge soundness

Theorem 6.1. *Let* FIOP *be a public-coin FIOP for a relation* R *with query class* $\mathbf{Q}$*, round complexity* k*, proof length* ℓ*, query complexity* q*, and knowledge soundness error* κ_{FIOP}*; suppose that* $\mathbf{Q}$ *has coverage factor* $C_{\mathbf{Q}}$ *(Definition 6.11) and admits a constraint solver with operation time* $t_{\mathbf{Q}}$*. Let* FC *be a public-coin interactive functional commitment scheme for* $\mathbf{Q}$ *with expected-time function binding error* $\epsilon^{\star}_{\mathsf{FC}}$*.*

Then $\mathsf{ARG} \coloneqq \mathsf{Funky}[\mathsf{FIOP}, \mathsf{FC}]$ *(Construction 3.25) is a public-coin interactive argument for* R *whose expected-time knowledge soundness error* $\kappa^{\star}_{\mathsf{ARG}}$ *satisfies the following for every* $t^{\star}_{\mathsf{ARG}} \geq t_{\mathcal{V}} + \sum_{i=1}^{\mathsf{k}} \mathsf{q}_i \cdot t_{\mathbf{Q}}(\ell_i)$ *and every* $M_1, \ldots, M_{\mathsf{k}} \in \mathbb{N}$*:*

$$\kappa^{\star}_{\mathsf{ARG}}(\lambda, n, t^{\star}_{\mathsf{ARG}}) \leq \left(\prod_{i=1}^{\mathsf{k}} C_{\mathbf{Q}}(\ell_i, M_i) \right) \cdot \kappa_{\mathsf{FIOP}}(n) + \epsilon^{\star}_{\mathsf{FC}}(\lambda, \ell_{\mathsf{max}}, \mathsf{q}_{\mathsf{max}}, t^{\star}_{\mathsf{FC}}) \ ,$$

where $t^{\star}_{\mathsf{FC}} = O(N \cdot t^{\star}_{\mathsf{ARG}})$ *for*

$$N \coloneqq 1 + \sum_{i=1}^{\mathsf{k}} \big(M_i + C_{\mathbf{Q}}(\ell_i, M_i) \big) \prod_{j=1}^{i-1} C_{\mathbf{Q}}(\ell_j, M_j) \ .$$

Moreover, the expected running time of the argument extractor is $O(N \cdot t^{\star}_{\mathsf{ARG}} + t_{\mathbf{E}})$*.*

Theorem 6.1's bounds depend on the *coverage factor* of $\mathbf{Q}$, a quantity which we define in the security analysis. In Section A we bound coverage factors for various query classes; these yield the following corollaries.

Corollary 6.2. *In the setting of Theorem 6.1, suppose that* $\mathbf{Q} = \mathbf{Q}_{\mathsf{Point}}$ *is the class of point queries. Then* ARG *is a public-coin interactive argument for* R *whose expected-time knowledge soundness error* $\kappa^{\star}_{\mathsf{ARG}}$ *satisfies the following for every* $t^{\star}_{\mathsf{ARG}} \geq t_{\mathcal{V}} + \sum_{i=1}^{\mathsf{k}} \mathsf{q}_i \cdot t_{\mathbf{Q}_{\mathsf{Point}}}(\ell_i)$ *and constant* $\alpha > 0$*:*

$$\kappa^{\star}_{\mathsf{ARG}}(\lambda, n, t^{\star}_{\mathsf{ARG}}) \leq (1 + \alpha) \cdot \kappa_{\mathsf{FIOP}}(n) + \epsilon^{\star}_{\mathsf{FC}}(\lambda, \ell_{\mathsf{max}}, \mathsf{q}_{\mathsf{max}}, t^{\star}_{\mathsf{FC}}) \ ,$$

where $t^{\star}_{\mathsf{FC}} = O(N \cdot t^{\star}_{\mathsf{ARG}})$ *for*

$$N \coloneqq 1 + \frac{\alpha\rho}{\rho - 1} + \sum_{i=1}^{\mathsf{k}} \max\left\{ 1, \left\lceil \frac{\mathsf{k} \cdot \ell_i}{\log(1+\alpha)} \right\rceil \right\} \cdot \rho^{i-1} \quad , \quad \rho \coloneqq (1+\alpha)^{1/\mathsf{k}} \ .$$

Moreover, the expected running time of the argument extractor is $O(N \cdot t^{\star}_{\mathsf{ARG}} + t_{\mathbf{E}})$*.*

Corollary 6.3. *In the setting of Theorem 6.1, suppose that* $\mathbf{Q}$ *has finite positive diameter* $d_{\mathbf{Q}}$*. Then* ARG *is a public-coin interactive argument for* R *whose expected-time knowledge soundness error* $\kappa^{\star}_{\mathsf{ARG}}$ *satisfies the following for every* $t^{\star}_{\mathsf{ARG}} \geq t_{\mathcal{V}} + \sum_{i=1}^{\mathsf{k}} \mathsf{q}_i \cdot t_{\mathbf{Q}}(\ell_i)$ *and constant* $\alpha > 0$*:*

$$\kappa^{\star}_{\mathsf{ARG}}(\lambda, n, t^{\star}_{\mathsf{ARG}}) \leq (1 + \alpha) \cdot \kappa_{\mathsf{FIOP}}(n) + \epsilon^{\star}_{\mathsf{FC}}(\lambda, \ell_{\mathsf{max}}, \mathsf{q}_{\mathsf{max}}, t^{\star}_{\mathsf{FC}}) \ ,$$

where $t^{\star}_{\mathsf{FC}} = O_{\alpha}(\mathsf{k} \cdot d \cdot t^{\star}_{\mathsf{ARG}})$ *for* $d \coloneqq \sum_{i=1}^{\mathsf{k}} d_{\mathbf{Q}}(\ell_i)$*. Moreover, the expected running time of the argument extractor is* $O_{\alpha}(\mathsf{k} \cdot d \cdot t^{\star}_{\mathsf{ARG}} + t_{\mathbf{E}})$*.*

Remark 6.4. Making explicit the dependence on α, Corollary 6.3's bound takes $t^{\star}_{\mathsf{FC}} = O\Big(\frac{1+\alpha}{\log(1+\alpha)} \cdot \mathsf{k} \cdot d \cdot t^{\star}_{\mathsf{ARG}}\Big)$, and the expected running time of the argument extractor is $O\Big(\frac{1+\alpha}{\log(1+\alpha)} \cdot \mathsf{k} \cdot d \cdot t^{\star}_{\mathsf{ARG}} + t_{\mathbf{E}}\Big)$.

Throughout, let $\mathsf{FIOP} = (\mathbf{P}, \mathbf{V})$ and $\mathsf{ARG} = (\mathcal{G}, \mathcal{P}, \mathcal{V})$. Let $\mathbf{E}$ be an extractor for FIOP with knowledge soundness error κ_{FIOP}. Let $\mathsf{Solver}_{\mathbf{Q}}$ be a constraint solver for $\mathbf{Q}$ with operation time $t_{\mathbf{Q}}$.

6.1 Extractor construction

We construct an extractor $\mathcal{E}$ and adversary $\mathcal{A}_{\mathsf{FC}}$ which will be used to analyze the expected-time knowledge soundness of ARG.

Construction 6.5 (Sampler)**.** For a round $i \in [\mathsf{k}]$, the sampler $\mathcal{S}_i$ works as follows.

$\mathcal{S}_i^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, T_1, \rho_1, \ldots, T_{i-1}, \rho_{i-1})$:

1. Loop:
   (a) For each $j \in [i, \mathsf{k}]$, sample verifier randomness $\rho_j \leftarrow \{0,1\}^{\mathsf{r}_j}$.
   (b) Sample an execution of the prover $\widetilde{\mathcal{P}}(\mathsf{aux})$ as follows. In the FIOP phase, use randomness $\rho_1, \ldots, \rho_{\mathsf{k}}$; the resulting (partial) interaction transcript is of the form

$$(\mathsf{cm}_1, \rho_1, \ldots, \mathsf{cm}_{\mathsf{k}}, \rho_{\mathsf{k}}, (\tau_1, \ldots, \tau_{\mathsf{k}})) \ ,$$

   where $\tau_j = (\boldsymbol{\alpha}_j, \boldsymbol{\beta}_j)$ is the query-answer tuple for the j-th commitment. In the opening phase, for each $j \in [\mathsf{k}]$, interact with FCGame: send $(\mathsf{cm}_j, \boldsymbol{\alpha}_j, \boldsymbol{\beta}_j)$ and then relay messages between $\widetilde{\mathcal{P}}(\mathsf{aux})$ and FCGame. Let tr be the resulting interaction transcript.
   (c) If the above execution is accepting and $\mathcal{C}_{T_j}$ decides τ_j for every $j < i$, return tr.

Construction 6.6 (Terminal sampler)**.** For a round $i \in [\mathsf{k}]$, the sampler $\mathcal{S}_{i,\mathsf{fin}}$ works as follows.

$\mathcal{S}_{i,\mathsf{fin}}^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, T_1, \rho_1, \ldots, T_{i-1}, \rho_{i-1}, T_i)$:

1. Loop:
   (a) Sample $\mathsf{tr} \leftarrow \mathcal{S}_i^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, T_1, \rho_1, \ldots, T_{i-1}, \rho_{i-1})$.
   (b) Parse the i-th-round randomness ρ_i and query-answer tuple τ_i from tr.
   (c) If $\mathcal{C}_{T_i}$ decides τ_i, return tr.

Construction 6.7 (Reductor)**.** Fix repetition parameters $M_1, \ldots, M_{\mathsf{k}} \in \mathbb{N}$. The reductor $\mathcal{R}$ works as follows.

$\mathcal{R}^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x})$:

1. For each $i \in [\mathsf{k}]$:
   (a) Initialize an empty list $T_i := \emptyset$.
   (b) Repeat M_i times:
      i. Sample $\mathsf{tr} \leftarrow \mathcal{S}_i^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, T_1, \rho_1, \ldots, T_{i-1}, \rho_{i-1})$ and parse τ_i from tr.
      ii. Append $T_i := T_i \parallel \tau_i$.
   (c) Sample $\mathsf{tr}_i \leftarrow \mathcal{S}_{i,\mathsf{fin}}^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, T_1, \rho_1, \ldots, T_{i-1}, \rho_{i-1}, T_i)$ and parse ρ_i from tr_i.
2. Return $(T_1, \rho_1, \ldots, T_{\mathsf{k}}, \rho_{\mathsf{k}})$.

Construction 6.8 (Argument extractor)**.** The extractor $\mathcal{E}$ works as follows.

$\mathcal{E}^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, \mathsf{tr}, b)$:

1. If $|\mathbb{x}| > n$ or $b = 0$, output $\perp$.
2. Sample $(T_1, \rho_1, \ldots, T_{\mathsf{k}}, \rho_{\mathsf{k}}) \leftarrow \mathcal{R}^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x})$.
3. Return $\mathbf{E}(\mathbb{x}, \widetilde{\Pi}_{T_1}, \rho_1, \ldots, \widetilde{\Pi}_{T_{\mathsf{k}}}, \rho_{\mathsf{k}})$, where each $\widetilde{\Pi}_{T_i} \in \mathrm{Sol}(\mathcal{C}_{T_i})$ is output by $\mathsf{Solver}_{\mathbf{Q}}$.

Construction 6.9 (FC adversary)**.** For an argument prover $\widetilde{\mathcal{P}}$, the adversary $\mathcal{A}_{\mathsf{FC}}^{\widetilde{\mathcal{P}}}$ interacts with $\mathsf{FCGame}(\mathsf{pp})$ as follows.

$\mathcal{A}_{\mathsf{FC}}^{\widetilde{\mathcal{P}}}(\mathsf{pp}, \eta)$:

1. Compute $(\mathbb{x}, \mathsf{aux}) \leftarrow \widetilde{\mathcal{P}}(\mathsf{pp}, \eta)$. If $|\mathbb{x}| > n$, halt.

2. Sample an execution of the prover $\widetilde{\mathcal{P}}(\mathsf{aux})$ as follows. In the FIOP phase, sample verifier randomness $\rho_1, \ldots, \rho_{\mathsf{k}}$. In the opening phase, interact with FCGame as in Step 1b. Let b and tr be the resulting decision bit and interaction transcript, respectively.
3. If $b = 1$, run $\mathcal{R}^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x})$.

6.2 Security analysis

Throughout, fix an argument prover $\widetilde{\mathcal{P}}$ and auxiliary input distribution $\mathcal{D}$. We consider the experiment

$$E \coloneqq \left\{ \begin{array}{l} \mathsf{pp} \leftarrow \mathcal{G}(1^\lambda, n) \\ \eta \leftarrow \mathcal{D} \\ (\mathbb{x}, \mathsf{aux}) \leftarrow \widetilde{\mathcal{P}}(\mathsf{pp}, \eta) \\ b \xleftarrow{\mathsf{tr}} \langle \widetilde{\mathcal{P}}(\mathsf{aux}), \mathcal{V}(\mathsf{pp}, \mathbb{x}) \rangle \end{array} \right\} ,$$

where $\mathsf{tr} = (\mathsf{cm}_1, \rho_1, \ldots, \mathsf{cm}_{\mathsf{k}}, \rho_{\mathsf{k}}, (\tau_1, \ldots, \tau_{\mathsf{k}}), \mathsf{pf}_1, \ldots, \mathsf{pf}_{\mathsf{k}})$. Let $t^\star_{\mathsf{ARG}}$ denote the expected running time of $\widetilde{\mathcal{P}}$ above; throughout, we assume the lower bound on $t^\star_{\mathsf{ARG}}$ from Theorem 6.1.

We also consider the experiment

$$E_{\mathsf{FC}} \coloneqq \left\{ \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{FC.Gen}(1^\lambda, \ell_{\mathsf{max}}) \\ \eta \leftarrow \mathcal{D} \\ \langle \mathcal{A}^{\widetilde{\mathcal{P}}}_{\mathsf{FC}}(\mathsf{pp}, \eta), \mathsf{FCGame}(\mathsf{pp}) \rangle \end{array} \right\} .$$

When $\mathcal{A}^{\widetilde{\mathcal{P}}}_{\mathsf{FC}}$ does not run $\mathcal{R}$, the lists $T_1, \ldots, T_{\mathsf{k}}$ are taken to be empty.

Definition 6.10. *For a public parameter* pp *and auxiliary input* η*, define:*

$$\begin{aligned} t^\star_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<i}) &\coloneqq \mathop{\mathbb{E}}_{E}\Big[\textit{running time of } \widetilde{\mathcal{P}} \;\Big|\; \mathsf{pp}, \eta, \boldsymbol{\rho}_{<i}\Big] \\ \epsilon_{\mathsf{pp},\eta}(\boldsymbol{T}_{<i}, \boldsymbol{\rho}_{<i}) &\coloneqq \Pr_{E}\big[b = 1 \wedge \forall j < i, \mathcal{C}_{T_j} \textit{ decides } \tau_j \;\big|\; \mathsf{pp}, \eta, \boldsymbol{\rho}_{<i}\big] \ , \\ \delta_{\mathsf{pp},\eta}(\boldsymbol{T}_{<i+1}, \boldsymbol{\rho}_{<i}) &\coloneqq \Pr_{E}\big[\mathcal{C}_{T_i} \textit{ decides } \tau_i \;\big|\; \mathsf{pp}, \eta, \boldsymbol{\rho}_{<i}, b = 1 \wedge \forall j < i, \mathcal{C}_{T_j} \textit{ decides } \tau_j\big] \ , \end{aligned}$$

where $\boldsymbol{T}_{<i}$ *and* $\boldsymbol{\rho}_{<i}$ *denote* $(T_1, \ldots, T_{i-1})$ *and* $(\rho_1, \ldots, \rho_{i-1})$*, respectively. We additionally define* $\epsilon_{\mathsf{pp},\eta}(\boldsymbol{T}_{<i}, \boldsymbol{\rho}_{<i})$ *for* $i \coloneqq \mathsf{k} + 1$*. When* $i = 1$*, we write* $t^\star_{\mathsf{pp},\eta}$ *and* $\epsilon_{\mathsf{pp},\eta}$ *for brevity.*

6.2.1 Coverage factors

To facilitate the security analysis, we introduce coverage factors and prove a technical lemma.

Definition 6.11 (Coverage factor)**.** *Let* $\ell, M \in \mathbb{N}$*. For a distribution* D *over query-answer tuples of* $\mathbf{Q}_\ell$ *and a satisfiable constraint set* $\mathcal{C}$*, define*

$$c_D(\mathcal{C}) \coloneqq \Pr_{X \leftarrow D}[\mathcal{C} \textit{ decides } X] \ .$$

The coverage factor *of* $\mathbf{Q}$ *is*

$$C_{\mathbf{Q}}(\ell, M) \coloneqq \sup_{D} \mathop{\mathbb{E}}_{T=(X_1,\ldots,X_M)\leftarrow D^M}\left[\frac{1}{c_D(\mathcal{C}_T)}\right] \ ,$$

where the supremum ranges over all such distributions D*. (Note that* $c_D(\mathcal{C}_T) > 0$ *with probability one, since* $\mathcal{C}_T$ *decides each* X_i *and the support of* D *is finite.)*

Lemma 6.12. *Let* pp *be a public parameter and* η *be auxiliary input where* $|\mathbb{x}| \leq n$ *and* $\epsilon_{\mathsf{pp},\eta} > 0$*. Let* $F_0, F_1, \ldots, F_i$ *be non-negative functions for* $i \leq \mathsf{k}$*, where* F_m *takes as input* $(\boldsymbol{T}_{<m+1}, \boldsymbol{\rho}_{<m+1})$*, such that, for every* $m \in [i]$*, all* $\boldsymbol{T}_{<m+1}$*, and all* $\boldsymbol{\rho}_{<m}$*,*

$$\mathop{\mathbb{E}}_{\rho_m \leftarrow \{0,1\}^{\mathsf{r}_m}} [F_m(\boldsymbol{T}_{<m+1}, \boldsymbol{\rho}_{<m+1})] \leq F_{m-1}(\boldsymbol{T}_{<m}, \boldsymbol{\rho}_{<m}) \ .$$

It holds that

$$\mathop{\mathbb{E}}_{E_{\mathsf{FC}}} \left[\frac{F_i(\boldsymbol{T}_{<i+1}, \boldsymbol{\rho}_{<i+1})}{\epsilon_{\mathsf{pp},\eta}(\boldsymbol{T}_{<i+1}, \boldsymbol{\rho}_{<i+1})} \;\middle|\; \mathsf{pp}, \eta, b = 1 \right] \leq \left(\prod_{m=1}^{i} C_{\mathbf{Q}}(\ell_m, M_m) \right) \cdot \frac{F_0}{\epsilon_{\mathsf{pp},\eta}} \ .$$

Proof. It suffices to show, for all $j \leq i$,

$$\begin{aligned} &\mathop{\mathbb{E}}_{E_{\mathsf{FC}}} \left[\frac{F_j(\boldsymbol{T}_{<j+1}, \boldsymbol{\rho}_{<j+1})}{\epsilon_{\mathsf{pp},\eta}(\boldsymbol{T}_{<j+1}, \boldsymbol{\rho}_{<j+1})} \;\middle|\; \mathsf{pp}, \eta, \boldsymbol{T}_{<j}, \boldsymbol{\rho}_{<j}, b = 1 \right] \\ &\qquad \leq C_{\mathbf{Q}}(\ell_j, M_j) \cdot \frac{F_{j-1}(\boldsymbol{T}_{<j}, \boldsymbol{\rho}_{<j})}{\epsilon_{\mathsf{pp},\eta}(\boldsymbol{T}_{<j}, \boldsymbol{\rho}_{<j})} \ ; \end{aligned} \tag{10}$$

the lemma follows by successively averaging over each round $j = i, \ldots, 1$, applying Equation 10 in each step.

Let $j \in [i]$. Fix (pp, η) as in the statement, and fix reachable $(\boldsymbol{T}_{<j}, \boldsymbol{\rho}_{<j})$. Let H denote the experiment E conditioned on $(\mathsf{pp}, \eta, \boldsymbol{\rho}_{<j})$, and let H_{FC} denote the experiment E_{FC} conditioned on $(\mathsf{pp}, \eta, \boldsymbol{T}_{<j}, \boldsymbol{\rho}_{<j})$ and the event $\{b = 1\}$. For brevity, write

$$\begin{aligned} F_{j-1} &\coloneqq F_{j-1}(\boldsymbol{T}_{<j}, \boldsymbol{\rho}_{<j}), & F_j(T_j, \rho_j) &\coloneqq F_j(\boldsymbol{T}_{<j+1}, \boldsymbol{\rho}_{<j+1}), \\ \epsilon_j &\coloneqq \epsilon_{\mathsf{pp},\eta}(\boldsymbol{T}_{<j}, \boldsymbol{\rho}_{<j}), & \epsilon_{j+1}(T_j, \rho_j) &\coloneqq \epsilon_{\mathsf{pp},\eta}(\boldsymbol{T}_{<j+1}, \boldsymbol{\rho}_{<j+1}), \\ \delta_j(T_j) &\coloneqq \delta_{\mathsf{pp},\eta}(\boldsymbol{T}_{<j+1}, \boldsymbol{\rho}_{<j}) \ . \end{aligned}$$

Fix T_j reachable in H_{FC}. We have

$$\mathop{\mathbb{E}}_{H_{\mathsf{FC}}} \left[\frac{F_j(T_j, \rho_j)}{\epsilon_{j+1}(T_j, \rho_j)} \;\middle|\; T_j \right] = \sum_{\substack{\rho_j \in \{0,1\}^{\mathsf{r}_j} \\ \epsilon_{j+1}(T_j, \rho_j) > 0}} \Pr_{H_{\mathsf{FC}}}[\rho_j \mid T_j] \cdot \frac{F_j(T_j, \rho_j)}{\epsilon_{j+1}(T_j, \rho_j)} \ .$$

Bayes' rule gives

$$\begin{aligned} \Pr_{H_{\mathsf{FC}}}[\rho_j \mid T_j] &= \Pr_{H}[\rho_j \mid b = 1 \wedge \forall m \leq j, \mathcal{C}_{T_m} \text{ decides } \tau_m] \\ &= \frac{\Pr_H[b = 1 \wedge \forall m \leq j, \mathcal{C}_{T_m} \text{ decides } \tau_m \mid \rho_j] \cdot 2^{-\mathsf{r}_j}}{\Pr_H[b = 1 \wedge \forall m \leq j, \mathcal{C}_{T_m} \text{ decides } \tau_m]} \\ &= \frac{\epsilon_{j+1}(T_j, \rho_j) \cdot 2^{-\mathsf{r}_j}}{\epsilon_j \cdot \delta_j(T_j)} \ , \end{aligned}$$

where the final equality is by the definitions of ϵ and δ. It follows that

$$\begin{aligned} \mathop{\mathbb{E}}_{H_{\mathsf{FC}}} \left[\frac{F_j(T_j, \rho_j)}{\epsilon_{j+1}(T_j, \rho_j)} \;\middle|\; T_j \right] &\leq \sum_{\rho_j \in \{0,1\}^{\mathsf{r}_j}} \frac{F_j(T_j, \rho_j) \cdot 2^{-\mathsf{r}_j}}{\epsilon_j \cdot \delta_j(T_j)} \\ &= \frac{\mathbb{E}_{\rho_j \leftarrow \{0,1\}^{\mathsf{r}_j}}[F_j(T_j, \rho_j)]}{\epsilon_j \cdot \delta_j(T_j)} \leq \frac{F_{j-1}}{\epsilon_j \cdot \delta_j(T_j)} \ . \end{aligned}$$

Let D be the distribution of query-answer tuples defined by

$$D(\tau) \coloneqq \Pr_H[\tau_j = \tau \mid b = 1 \wedge \forall m < j, \mathcal{C}_{T_m} \text{ decides } \tau_m] \ .$$

By construction, $T_j \leftarrow D^{M_j}$ in H_{FC} and $\delta_j(T_j) = c_D(\mathcal{C}_{T_j})$. Thus, by Definition 6.11,

$$\mathop{\mathbb{E}}_{H_{\mathsf{FC}}}\left[\frac{1}{\delta_j(T_j)}\right] \leq C_{\mathbf{Q}}(\ell_j, M_j) \ ,$$

and Equation 10 follows by averaging over T_j in H_{FC}. □

6.2.2 Running time

We analyze the expected running time of the adversary $\mathcal{A}_{\mathsf{FC}}$ (Construction 6.9).

Lemma 6.13 (Expected running time)**.** *The expected running time of $\mathcal{A}_{\mathsf{FC}}^{\widetilde{\mathcal{P}}}$ in E_{FC} is at most $O(N \cdot t^{\star}_{\mathsf{ARG}})$ for N as defined in Theorem 6.1.*

Proof. For simplicity, we focus on the time spent executing $\widetilde{\mathcal{P}}$; the remaining time can be bounded with some additional accounting. Fix a public parameter pp and auxiliary input η, which together determine the instance $\mathbb{x}$ and prover state aux. Without loss of generality, assume $|\mathbb{x}| \leq n$ and $\epsilon_{\mathsf{pp},\eta} > 0$.

Claim 6.14. *For a round $i \in [\mathsf{k}]$, let $\mathcal{T}_i$ denote the time spent executing $\widetilde{\mathcal{P}}$ during one invocation of $\mathcal{S}_i$ in the i-th iteration of Loop 1. It holds that*

$$\mathbb{E}[\mathcal{T}_i \mid \mathsf{pp}, \eta, b = 1] \leq \left(\prod_{j=1}^{i-1} C_{\mathbf{Q}}(\ell_j, M_j)\right) \cdot \frac{t^{\star}_{\mathsf{pp},\eta}}{\epsilon_{\mathsf{pp},\eta}} \ .$$

Proof. By construction,

$$\mathbb{E}[\mathcal{T}_i \mid \mathsf{pp}, \eta, b = 1, \boldsymbol{T}_{<i}, \boldsymbol{\rho}_{<i}] \leq \frac{t^{\star}_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<i})}{\epsilon_{\mathsf{pp},\eta}(\boldsymbol{T}_{<i}, \boldsymbol{\rho}_{<i})} \ .$$

Averaging over $(\boldsymbol{T}_{<i}, \boldsymbol{\rho}_{<i})$ and applying Lemma 6.12 with

$$F_j(\boldsymbol{T}_{<j+1}, \boldsymbol{\rho}_{<j+1}) \coloneqq t^{\star}_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<j+1}) \quad \text{for } j \in \{0, \ldots, i-1\}$$

yields the claim. □

Claim 6.15. *For a round $i \in [\mathsf{k}]$, let $\mathcal{T}_{i,\mathsf{fin}}$ denote the time spent executing $\widetilde{\mathcal{P}}$ during the invocation of $\mathcal{S}_{i,\mathsf{fin}}$ in the i-th iteration of Loop 1. It holds that*

$$\mathbb{E}[\mathcal{T}_{i,\mathsf{fin}} \mid \mathsf{pp}, \eta, b = 1] \leq \left(\prod_{j=1}^{i} C_{\mathbf{Q}}(\ell_j, M_j)\right) \cdot \frac{t^{\star}_{\mathsf{pp},\eta}}{\epsilon_{\mathsf{pp},\eta}} \ .$$

Proof. By construction,

$$\mathbb{E}[\mathcal{T}_{i,\mathsf{fin}} \mid \mathsf{pp}, \eta, b = 1, \boldsymbol{T}_{<i+1}, \boldsymbol{\rho}_{<i}] \leq \frac{t^{\star}_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<i})}{\epsilon_{\mathsf{pp},\eta}(\boldsymbol{T}_{<i}, \boldsymbol{\rho}_{<i}) \cdot \delta_{\mathsf{pp},\eta}(\boldsymbol{T}_{<i+1}, \boldsymbol{\rho}_{<i})} \ .$$

Averaging over T_i and applying the definition of $C_{\mathbf{Q}}$ in Definition 6.11, we obtain

$$\mathbb{E}[\mathcal{T}_{i,\mathsf{fin}} \mid \mathsf{pp}, \eta, b = 1, \boldsymbol{T}_{<i}, \boldsymbol{\rho}_{<i}] \leq C_{\mathbf{Q}}(\ell_i, M_i) \cdot \frac{t^{\star}_{\mathsf{pp},\eta}(\boldsymbol{\rho}_{<i})}{\epsilon_{\mathsf{pp},\eta}(\boldsymbol{T}_{<i}, \boldsymbol{\rho}_{<i})} \ .$$

Averaging over $(\boldsymbol{T}_{<i}, \boldsymbol{\rho}_{<i})$ and applying Lemma 6.12 (as in Claim 6.14) yields the claim. □

By Claims 6.14 and 6.15,

$$\mathbb{E}[\text{running time of } \mathcal{R} \mid \mathsf{pp}, \eta, b = 1] \leq \left(\sum_{i=1}^{\mathsf{k}} \left(M_i + C_{\mathbf{Q}}(\ell_i, M_i) \right) \cdot \prod_{j=1}^{i-1} C_{\mathbf{Q}}(\ell_j, M_j) \right) \cdot \frac{t^{\star}_{\mathsf{pp},\eta}}{\epsilon_{\mathsf{pp},\eta}} \ .$$

Since the reductor only executes if $b = 1$, we conclude that

$$\begin{aligned} \mathbb{E}[\text{running time of } \mathcal{R} \mid \mathsf{pp}, \eta] &= \mathbb{E}[\text{running time of } \mathcal{R} \mid \mathsf{pp}, \eta, b = 1] \cdot \Pr[b = 1 \mid \mathsf{pp}, \eta] \\ &\leq \left(\sum_{i=1}^{\mathsf{k}} \left(M_i + C_{\mathbf{Q}}(\ell_i, M_i) \right) \cdot \prod_{j=1}^{i-1} C_{\mathbf{Q}}(\ell_j, M_j) \right) \cdot t^{\star}_{\mathsf{pp},\eta} \ ; \end{aligned}$$

averaging over pp and η bounds the expected running time of the reductor $\mathcal{R}$. Accounting also for the initial execution of $\widetilde{\mathcal{P}}$ gives the lemma. □

6.2.3 Success probability

We analyze the probability that the extractor $\mathcal{E}$ fails to output a witness.

When the reductor executes in E_{FC}, let $(\mathsf{cm}_1, \rho_1, \ldots, \mathsf{cm}_{\mathsf{k}}, \rho_{\mathsf{k}}, (\tau_1, \ldots, \tau_{\mathsf{k}}))$ denote the transcript tr_{k} produced in the final loop iteration of the round-k terminal sampler. Define the following events:

- Ext: the reductor executes and $(\mathbb{x}, \mathbf{E}(\mathbb{x}, \widetilde{\Pi}_{T_1}, \rho_1, \ldots, \widetilde{\Pi}_{T_{\mathsf{k}}}, \rho_{\mathsf{k}})) \in R$.
- Coll: the reductor executes and, for some $i \in [\mathsf{k}]$, $\mathcal{C}_{T_i}$ *refutes* τ_i.

Definition 6.16. *For an instance $\mathbb{x}$, inductively define*

$$\begin{aligned} \kappa_{\mathsf{k}}(\mathbb{x}, \Pi_1, \rho_1, \ldots, \Pi_{\mathsf{k}}, \rho_{\mathsf{k}}) &\coloneqq \mathbb{1}\left[\begin{array}{l} \mathbf{V}^{\Pi_1, \ldots, \Pi_{\mathsf{k}}}(\mathbb{x}; \rho_1, \ldots, \rho_{\mathsf{k}}) = 1 \\ \wedge\, (\mathbb{x}, \mathbf{E}(\mathbb{x}, \Pi_1, \rho_1, \ldots, \Pi_{\mathsf{k}}, \rho_{\mathsf{k}})) \notin R \end{array} \right] , \\ \kappa_{i-1}(\mathbb{x}, \Pi_1, \rho_1, \ldots, \Pi_{i-1}, \rho_{i-1}) &\coloneqq \max_{\Pi_i \in \Sigma^{\ell_i}} \mathop{\mathbb{E}}_{\rho_i \leftarrow \{0,1\}^{\mathsf{r}_i}} [\kappa_i(\mathbb{x}, \Pi_1, \rho_1, \ldots, \Pi_i, \rho_i)] \quad \textit{for } i = \mathsf{k}, \ldots, 1 \ . \end{aligned}$$

Lemma 6.17 (Success probability)**.** *It holds that*

$$\Pr\left[\begin{array}{l} |\mathbb{x}| \leq n \\ \wedge\, \neg \mathsf{Ext} \\ \wedge\, \neg \mathsf{Coll} \\ \wedge\, b = 1 \end{array} \right] \leq \left(\prod_{i=1}^{\mathsf{k}} C_{\mathbf{Q}}(\ell_i, M_i) \right) \cdot \kappa_{\mathsf{FIOP}}(n) \ .$$

Proof. Fix a public parameter pp and auxiliary input η, which together determine the instance $\mathbb{x}$ and prover state aux. Without loss of generality, assume $|\mathbb{x}| \leq n$.

Conditioned on $b = 1$, the following hold:

- If Ext does not occur, then
$$(\mathbb{x}, \mathbf{E}(\mathbb{x}, \widetilde{\Pi}_{T_1}, \rho_1, \ldots, \widetilde{\Pi}_{T_{\mathsf{k}}}, \rho_{\mathsf{k}})) \notin R \ .$$
- Recall that each $\mathcal{C}_{T_i}$ decides τ_i. Thus, if Coll does not occur, then each $\mathcal{C}_{T_i}$ *entails* τ_i, and hence each $\widetilde{\Pi}_{T_i}$ agrees with τ_i. Since tr_{k} is accepting,
$$\mathbf{V}^{\widetilde{\Pi}_{T_1}, \ldots, \widetilde{\Pi}_{T_{\mathsf{k}}}}(\mathbb{x}; \rho_1, \ldots, \rho_{\mathsf{k}}) = 1 \ .$$

Therefore,

$$\begin{aligned}\Pr[\neg\mathsf{Ext} \wedge \neg\mathsf{Coll} \mid \mathsf{pp}, \eta, b = 1] &\le \mathbb{E}\Big[\kappa_{\mathsf{k}}(\mathbb{x}, \widetilde{\Pi}_{T_1}, \rho_1, \ldots, \widetilde{\Pi}_{T_{\mathsf{k}}}, \rho_{\mathsf{k}}) \;\Big|\; \mathsf{pp}, \eta, b = 1\Big] \\ &\le \mathbb{E}\left[\frac{\kappa_{\mathsf{k}}(\mathbb{x}, \widetilde{\Pi}_{T_1}, \rho_1, \ldots, \widetilde{\Pi}_{T_{\mathsf{k}}}, \rho_{\mathsf{k}})}{\epsilon_{\mathsf{pp},\eta}(\boldsymbol{T}_{<\mathsf{k}+1}, \boldsymbol{\rho}_{<\mathsf{k}+1})} \;\middle|\; \mathsf{pp}, \eta, b = 1\right] \ ,\end{aligned}$$

where the final quantity is well-defined since $\epsilon_{\mathsf{pp},\eta}(\boldsymbol{T}_{<\mathsf{k}+1}, \boldsymbol{\rho}_{<\mathsf{k}+1}) > 0$ for any $(\boldsymbol{T}_{<\mathsf{k}+1}, \boldsymbol{\rho}_{<\mathsf{k}+1})$ reachable in E_{FC}. Applying Lemma 6.12 with

$$F_j(\boldsymbol{T}_{<j+1}, \boldsymbol{\rho}_{<j+1}) \coloneqq \kappa_j(\mathbb{x}, \widetilde{\Pi}_{T_1}, \rho_1, \ldots, \widetilde{\Pi}_{T_j}, \rho_j) \quad \text{for } j = 0, \ldots, \mathsf{k} \ ,$$

we obtain

$$\mathbb{E}\Big[\kappa_{\mathsf{k}}(\mathbb{x}, \widetilde{\Pi}_{T_1}, \rho_1, \ldots, \widetilde{\Pi}_{T_{\mathsf{k}}}, \rho_{\mathsf{k}}) \;\Big|\; \mathsf{pp}, \eta, b = 1\Big] \le \left(\prod_{i=1}^{\mathsf{k}} C_{\mathbf{Q}}(\ell_i, M_i)\right) \cdot \frac{\kappa_0(\mathbb{x})}{\epsilon_{\mathsf{pp},\eta}} \ .$$

By knowledge soundness of FIOP (Definition 3.24), it holds that $\kappa_0(\mathbb{x}) \le \kappa_{\mathsf{FIOP}}(n)$. We conclude that

$$\begin{aligned}\Pr[\neg\mathsf{Ext} \wedge \neg\mathsf{Coll} \wedge b = 1 \mid \mathsf{pp}, \eta] &= \Pr[\neg\mathsf{Ext} \wedge \neg\mathsf{Coll} \mid \mathsf{pp}, \eta, b = 1] \cdot \Pr[b = 1 \mid \mathsf{pp}, \eta] \\ &\le \left(\prod_{i=1}^{\mathsf{k}} C_{\mathbf{Q}}(\ell_i, M_i)\right) \cdot \kappa_{\mathsf{FIOP}}(n) \ ,\end{aligned}$$

and the lemma follows by averaging over pp and η. □

Proof of Theorem 6.1. Instantiate the reductor $\mathcal{R}$ with repetition parameters $M_1, \ldots, M_{\mathsf{k}}$ as in the statement. Let $E_{\mathcal{E}}$ be the extension of E that additionally samples

$$\mathbb{w} \leftarrow \mathcal{E}^{\widetilde{\mathcal{P}}(\mathsf{aux})}(\mathsf{pp}, \mathbb{x}, \mathsf{tr}, b) \ .$$

The reductor execution in this experiment is distributed as in E_{FC}. The argument extractor runs this reductor and then $\mathbf{E}$, so by Lemma 6.13, its expected running time is at most

$$O(N \cdot t^{\star}_{\mathsf{ARG}} + t_{\mathbf{E}}) \ .$$

By total probability,

$$\begin{aligned}\Pr_{E_{\mathcal{E}}}\left[\begin{array}{l} |\mathbb{x}| \le n \\ \wedge\, (\mathbb{x}, \mathbb{w}) \notin R \\ \wedge\, b = 1 \end{array}\right] &= \Pr_{E_{\mathsf{FC}}}\left[\begin{array}{l} |\mathbb{x}| \le n \\ \wedge\, \neg\mathsf{Ext} \\ \wedge\, b = 1 \end{array}\right] \\ &\le \underbrace{\Pr_{E_{\mathsf{FC}}}\left[\begin{array}{l} |\mathbb{x}| \le n \\ \wedge\, \neg\mathsf{Ext} \\ \wedge\, \neg\mathsf{Coll} \\ \wedge\, b = 1 \end{array}\right]}_{A} + \underbrace{\Pr_{E_{\mathsf{FC}}}\left[\begin{array}{l} |\mathbb{x}| \le n \\ \wedge\, \mathsf{Coll} \\ \wedge\, b = 1 \end{array}\right]}_{B} \ .\end{aligned}$$

Bounding A. By Lemma 6.17,

$$A \le \left(\prod_{i=1}^{\mathsf{k}} C_{\mathbf{Q}}(\ell_i, M_i)\right) \cdot \kappa_{\mathsf{FIOP}}(n) \ .$$

Bounding B. In E_{FC}, observe that Coll implies $\mathrm{Sol}(\mathcal{C}_{\mathsf{cm}_i, \ell_i, \mathsf{q}_i}) = \emptyset$ for some $i \in [\mathsf{k}]$ (see Definition 3.17). By expected-time function binding of FC (Definition 3.19) and Lemma 6.13,

$$B \le \epsilon^{\star}_{\mathsf{FC}}(\lambda, \ell_{\mathsf{max}}, \mathsf{q}_{\mathsf{max}}, t^{\star}_{\mathsf{FC}}) \ ,$$

where $t^{\star}_{\mathsf{FC}}$ is as stated in the theorem. □

Proof of Corollary 6.2. We instantiate Theorem 6.1 with

$$M_i \coloneqq \max\left\{1, \left\lceil \frac{\mathsf{k} \cdot \ell_i}{\log(1+\alpha)} \right\rceil \right\} \qquad \text{for } i \in [\mathsf{k}] \ .$$

By Lemma A.2, this yields

$$C_{\mathbf{Q}_{\mathsf{Point}}}(\ell_i, M_i) \leq 2^{\ell_i/M_i} \leq (1+\alpha)^{1/\mathsf{k}} \ ,$$

Writing $\rho \coloneqq (1+\alpha)^{1/\mathsf{k}}$, we have

$$\begin{aligned} N &= 1 + \sum_{i=1}^{\mathsf{k}} \left(M_i + C_{\mathbf{Q}_{\mathsf{Point}}}(\ell_i, M_i)\right) \prod_{j=1}^{i-1} C_{\mathbf{Q}_{\mathsf{Point}}}(\ell_j, M_j) \\ &\leq 1 + \sum_{i=1}^{\mathsf{k}} (M_i + \rho)\rho^{i-1} \\ &= 1 + \frac{\alpha\rho}{\rho - 1} + \sum_{i=1}^{\mathsf{k}} M_i \rho^{i-1} \ . \end{aligned}$$

□

Proof of Corollary 6.3. By Lemma A.3, there exist repetition parameters $M_1, \ldots, M_{\mathsf{k}}$ satisfying

$$M_i = O_\alpha(\mathsf{k} \cdot d_{\mathbf{Q}}(\ell_i)) \quad \text{and} \quad C_{\mathbf{Q}}(\ell_i, M_i) \leq (1+\alpha)^{1/\mathsf{k}} \quad \text{for } i \in [\mathsf{k}] \ .$$

Instantiating Theorem 6.1 with these parameters yields

$$\begin{aligned} N &= 1 + \sum_{i=1}^{\mathsf{k}} \left(M_i + C_{\mathbf{Q}}(\ell_i, M_i)\right) \prod_{j=1}^{i-1} C_{\mathbf{Q}}(\ell_j, M_j) \\ &= O_\alpha\left(\mathsf{k} \cdot \sum_{i=1}^{\mathsf{k}} d_{\mathbf{Q}}(\ell_i)\right) = O_\alpha(\mathsf{k} \cdot d) \ . \end{aligned}$$

□

A Coverage factor bounds

Fact A.1 (Hölder's inequality for sums)**.** For every finite set I, non-negative reals $(a_i)_{i\in I}$ and $(b_i)_{i\in I}$, and reals $p, q > 1$ satisfying $1/p + 1/q = 1$,

$$\sum_{i\in I} a_i b_i \le \left(\sum_{i\in I} a_i^p\right)^{1/p} \left(\sum_{i\in I} b_i^q\right)^{1/q} .$$

Lemma A.2 (Point query bound)**.** *For every $\ell, M \in \mathbb{N}$ with $M \ge 1$,*

$$C_{\mathbf{Q}_{\mathsf{Point}}}(\ell, M) \le 2^{\ell/M} .$$

Proof. Fix a distribution D over query-answer tuples of $\mathbf{Q}_{\mathsf{Point}}$, and let Q be the induced distribution over their query sets, viewed as subsets of $[\ell]$. Draw $\tau_1, \ldots, \tau_M$ independently from D, let $T \coloneqq (\tau_1, \ldots, \tau_M)$, and let U be the union of their query sets. For every $S \subseteq [\ell]$, write

$$c(S) \coloneqq \Pr_{X\leftarrow Q}[X \subseteq S] \text{ and } p(S) \coloneqq \Pr[U = S] .$$

The event $U = S$ implies that all M sampled query sets are contained in S, and hence

$$p(S) \le c(S)^M .$$

Observe that $c(U) = c_D(\mathcal{C}_T)$. By Fact A.1, the following holds for $M \ge 2$:

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{c_D(\mathcal{C}_T)}\right] &= \sum_{\substack{S\subseteq[\ell]\\ p(S)>0}} \frac{p(S)}{c(S)} \\
&\le \sum_{S\subseteq[\ell]} p(S)^{1-1/M} \\
&\le \left(\sum_{S\subseteq[\ell]} 1^M\right)^{1/M} \left(\sum_{S\subseteq[\ell]} p(S)\right)^{1-1/M} \\
&= 2^{\ell/M} .
\end{aligned}$$

For $M = 1$, the same bound follows from $p(S)/c(S) \le 1$ and there being at most 2^ℓ possible sets S. □

Lemma A.3 (Diameter-based bound)**.** *For every $\ell, M \in \mathbb{N}$ with $d \coloneqq d_{\mathbf{Q}}(\ell)$ positive and $M \ge d$,*

$$C_{\mathbf{Q}}(\ell, M) \le 1 + \frac{2d}{M+1} + \sum_{j=0}^{d-1} \binom{M}{j} 2^{-(M-j-1)} ,$$

and, for all $M \ge 5d$,

$$C_{\mathbf{Q}}(\ell, M) \le 1 + O\left(\frac{d}{M}\right) .$$

Proof. Fix a distribution D over query-answer tuples of $\mathbf{Q}$. Draw $\tau_1, \ldots, \tau_M, \tau_{M+1}$ independently from D and let $T \coloneqq (\tau_1, \ldots, \tau_M)$. Set $Z \coloneqq 1/c_D(\mathcal{C}_T)$ and fix $\theta \in (0, 1)$; we have

$$\mathbb{E}[Z] = \mathbb{E}\Big[Z \cdot \mathbb{1}[c_D(\mathcal{C}_T) \ge \theta]\Big] + \mathbb{E}\Big[Z \cdot \mathbb{1}[c_D(\mathcal{C}_T) < \theta]\Big] .$$

For $n \in \{0, \ldots, M\}$, let $\mathcal{C}_n \coloneqq \mathcal{C}_{(\tau_1,\ldots,\tau_n)}$, with $\mathcal{C}_0 \coloneqq \emptyset$. As in Lemma 5.7, say that τ_{n+1} is *new* if $\mathcal{C}_n$ does not decide τ_{n+1}.

Claim A.4.

$$\mathbb{E}\Big[Z \cdot \mathbb{1}[c_D(\mathcal{C}_T) \geq \theta]\Big] \leq 1 + \frac{d}{\theta(M+1)} \ .$$

Proof. On $c_D(\mathcal{C}_T) \geq \theta$,

$$Z = 1 + \frac{1 - c_D(\mathcal{C}_T)}{c_D(\mathcal{C}_T)} \leq 1 + \frac{1 - c_D(\mathcal{C}_T)}{\theta} \ .$$

It follows that

$$\begin{aligned}\mathbb{E}\Big[Z \cdot \mathbb{1}[c_D(\mathcal{C}_T) \geq \theta]\Big] &\leq \mathbb{E}\left[\left(1 + \frac{1 - c_D(\mathcal{C}_T)}{\theta}\right) \cdot \mathbb{1}[c_D(\mathcal{C}_T) \geq \theta]\right] \\ &\leq 1 + \frac{1}{\theta} \cdot \mathbb{E}[1 - c_D(\mathcal{C}_T)] \ .\end{aligned}$$

For $n \in \{0, \ldots, M\}$, let $a_n \coloneqq \mathbb{E}[1 - c_D(\mathcal{C}_n)]$. Since τ_{n+1} is independent of $\mathcal{C}_n$, a_n is the probability that τ_{n+1} is new. The sequence $(a_n)_{n=0}^{M}$ is non-increasing, and at most d samples can be new. Hence

$$\mathbb{E}[1 - c_D(\mathcal{C}_T)] = a_M \leq \frac{1}{M+1} \sum_{n=0}^{M} a_n \leq \frac{d}{M+1} \ .$$

□

Claim A.5.

$$\mathbb{E}\Big[Z \cdot \mathbb{1}[c_D(\mathcal{C}_T) < \theta]\Big] \leq \sum_{j=0}^{d-1} \binom{M}{j} \theta^{M-j-1} \ .$$

Proof. Let $R(T) \subseteq [M]$ be the set of indices j for which τ_j is new. Observe that $c_D(\mathcal{C}_T) < \theta$ can occur only when $|R(T)| \leq d-1$. Accordingly, fix $I \subseteq [M]$ with $|I| \leq d-1$, and let T_I denote the sublist of T indexed by I. Conditional on T_I, the remaining tuples are independent samples from D. The event $R(T) = I$ implies that $\mathcal{C}_T = \mathcal{C}_{T_I}$ and that $\mathcal{C}_{T_I}$ decides every tuple outside T_I. Therefore,

$$\Pr[R(T) = I \mid T_I] \leq c_D(\mathcal{C}_{T_I})^{M-|I|} \ .$$

On the event $R(T) = I$, we have $Z = 1/c_D(\mathcal{C}_{T_I})$. Thus, for every such I,

$$\begin{aligned}&\mathbb{E}\Big[Z \cdot \mathbb{1}[c_D(\mathcal{C}_T) < \theta] \cdot \mathbb{1}[R(T) = I] \ \Big| \ T_I\Big] \\ &\quad \leq \mathbb{1}[c_D(\mathcal{C}_{T_I}) < \theta] \cdot c_D(\mathcal{C}_{T_I})^{M-|I|-1} \leq \theta^{M-|I|-1} \ .\end{aligned}$$

Averaging over T_I and summing over the possible values of $R(T)$ proves the claim. □

Applying the two claims, we get

$$\mathbb{E}[Z] \leq 1 + \frac{d}{\theta(M+1)} + \sum_{j=0}^{d-1} \binom{M}{j} \theta^{M-j-1} \ .$$

Setting $\theta = 1/2$ establishes the statement's first bound. The second bound follows from the fact that setting $M \geq 5d$ yields

$$\sum_{j=0}^{d-1} \binom{M}{j} 2^{-(M-j-1)} \leq 2^{-(M-d)} \sum_{j=0}^{d} \binom{M}{j}$$

$$\leq 2^{-(M-d)} \left(\frac{eM}{d} \right)^d$$
$$\leq \left(\frac{(5e)^{1/5}}{2^{4/5}} \right)^M < 0.968^M \quad .$$

□

Acknowledgments

Ziyi Guan thanks Christian Knabenhans, Nicholas Spooner, and Eylon Yogev for helpful discussions in the early stages of this work.

AI disclosure

This paper has three main technical components: the lower bounds (Section 4), the soundness analysis of the Funky protocol (Section 5), and the knowledge soundness analysis of Funky (Section 6). The ideas and proofs in the lower bound section are entirely the work of the authors. For both analyses of Funky, the authors first established slightly weaker bounds without AI assistance. The bounds presented here were subsequently tightened through iterative discussions with AI models, drawing on the authors' insights. The proofs of the coverage factor bounds in Section A were produced primarily by an AI model, with subsequent edits by the authors. AI tools also provided assistance in polishing the text throughout the paper.

References

[AFR23] Thomas Attema, Serge Fehr, and Nicolas Resch. "Generalized Special-Sound Interactive Proofs and Their Knowledge Soundness". In: *Proceedings of the 21st Theory of Cryptography Conference*. TCC '23. 2023, pp. 424–454. DOI: 10.1007/978-3-031-48621-0_15.

[AKLY24] Thomas Attema, Michael Klooß, Russell W. F. Lai, and Pavlo Yatsyna. *Adaptive Special Soundness: Improved Knowledge Extraction by Adaptive Useful Challenge Sampling*. Cryptology ePrint Archive, Paper 2024/2038. 2024. URL: https://eprint.iacr.org/2024/2038.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. "Fast Reed–Solomon Interactive Oracle Proofs of Proximity". In: *Proceedings of the 45th International Colloquium on Automata, Languages and Programming*. ICALP '18. 2018, 14:1–14:17. DOI: 10.4230/LIPIcs.ICALP.2018.14.

[BCG20] Jonathan Bootle, Alessandro Chiesa, and Jens Groth. "Linear-Time Arguments with Sublinear Verification from Tensor Codes". In: *Proceedings of the 18th Theory of Cryptography Conference*. TCC '20. 2020, pp. 19–46. DOI: 10.1007/978-3-030-64378-2_2.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. "Interactive Oracle Proofs". In: *Proceedings of the 14th Theory of Cryptography Conference*. TCC '16-B. 2016, pp. 31–60. DOI: 10.1007/978-3-662-53644-5_2.

[BF23] Benedikt Bünz and Ben Fisch. "Multilinear Schwartz-Zippel Mod N and Lattice-Based Succinct Arguments". In: *Proceedings of the 21st Theory of Cryptography Conference*. TCC '23. 2023, 394–423. DOI: 10.1007/978-3-031-48621-0_14.

[BFS20] Benedikt Bünz, Ben Fisch, and Alan Szepieniec. "Transparent SNARKs from DARK Compilers". In: *Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '20. 2020, pp. 677–706. DOI: 10.1007/978-3-030-45721-1_24.

[BG08] Boaz Barak and Oded Goldreich. "Universal Arguments and their Applications". In: *SIAM Journal on Computing* 38.5 (2008). Preliminary version appeared in CCC '02., pp. 1661–1694. DOI: 10.1137/070709244.

[BG93] Mihir Bellare and Oded Goldreich. "On Defining Proofs of Knowledge". In: *Proceedings of the 12th Annual International Cryptology Conference*. CRYPTO '92. 1993, pp. 390–420. DOI: 10.1007/3-540-48071-4_28.

[BL04] Boaz Barak and Yehuda Lindell. "Strict Polynomial-Time in Simulation and Extraction". In: *SIAM Journal on Computing* 33.4 (2004), pp. 783–818. DOI: 10.1137/S0097539703427975.

[CDDGS25] Alessandro Chiesa, Marcel Dall'Agnol, Zijing Di, Ziyi Guan, and Nicholas Spooner. "Quantum Rewinding for IOP-Based Succinct Arguments". In: *Proceedings of the 23rd Theory of Cryptography Conference*. TCC '25. 2025, pp. 460–479. DOI: 10.1007/978-3-032-12296-4_16.

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

[CDGSY24] Alessandro Chiesa, Marcel Dall'Agnol, Ziyi Guan, Nicholas Spooner, and Eylon Yogev. "Untangling the Security of Kilian's Protocol: Upper and Lower Bounds". In: *Proceedings of the 22nd Theory of Cryptography Conference*. TCC '24. 2024, pp. 158–188. DOI: 10.1007/978-3-031-78011-0_6.

[CFFQR21] Matteo Campanelli, Antonio Faonio, Dario Fiore, Anaïs Querol, and Hadrián Rodríguez. "Lunar: A Toolbox for More Efficient Universal and Updatable zkSNARKs and Commit-and-Prove Extensions". In: *Proceedings of the 27th International Conference on the Theory and Application of Cryptology and Information Security*. ASIACRYPT '21. 2021, pp. 3–33. DOI: 10.1007/978-3-030-92078-4_1.

[CGKY26] Alessandro Chiesa, Ziyi Guan, Christian Knabenhans, and Zihan Yu. "On the Fiat–Shamir Security of Succinct Arguments from Functional Commitments". In: *Proceedings of the 46th Annual International Cryptology Conference*. CRYPTO '26. 2026, pp. 189–213. DOI: 10.1007/978-3-032-35424-2_7.

[CHMMVW20] Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Noah Vesely, and Nicholas Ward. "Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS". In: *Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '20. 2020, pp. 738–768. DOI: 10.1007/978-3-030-45721-1_26.

[CMSZ21] Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. "Post-Quantum Succinct Arguments: Breaking the Quantum Rewinding Barrier". In: *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science*. FOCS '21. IEEE, 2021, pp. 49–58. DOI: 10.1109/FOCS52979.2021.00014.

[CY24] Alessandro Chiesa and Eylon Yogev. *Building Cryptographic Proofs from Hash Functions*. 2024. URL: https://github.com/hash-based-snargs-book/hash-based-snargs-book.

[Cla26] Claude (Anthropic). *A $2^{o(n)}$ Time Algorithm for 3-SAT*. Preprint, October 2026. 2026. URL: https://github.com/anthropics/formal-math/tree/main/eth.

[Din07] Irit Dinur. "The PCP Theorem by Gap Amplification". In: *Journal of the ACM* 54.3, 12 (2007). DOI: 10.1145/1236457.1236459.

[GK96] Oded Goldreich and Hugo Krawczyk. "On the Composition of Zero-Knowledge Proof Systems". In: *SIAM Journal on Computing* 25.1 (1996). Preliminary version appeared in ICALP '90., pp. 169–192. DOI: 10.1137/S0097539791220688.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Volume 1 – Basic Tools*. Cambridge University Press, 2001. ISBN: 978-0-521-79172-4. DOI: 10.1017/CBO9780511546891.

[HM26] Iftach Haitner and Nikolaos Makriyannis. "Tight Bounds on Uniform-Challenge Reductions from Σ Protocols". In: *Proceedings of the 45th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '26. 2026, pp. 125–153. DOI: 10.1007/978-3-032-25330-9_5.

[IJK09] Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. "Approximate List-Decoding of Direct Product Codes and Uniform Hardness Amplification". In: *SIAM Journal on Computing* 39.2 (2009), pp. 564–605. DOI: 10.1137/070683994.

[JT20] Joseph Jaeger and Stefano Tessaro. "Expected-Time Cryptography: Generic Techniques and Applications to Concrete Soundness". In: *Proceedings of the 18th Theory of Cryptography Conference*. TCC '20. 2020, pp. 414–443. DOI: 10.1007/978-3-030-64381-2_15.

[Kil92] Joe Kilian. "A note on efficient zero-knowledge proofs and arguments (extended abstract)". In: *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*. STOC '92. ACM, 1992, pp. 723–732. DOI: 10.1145/129712.129782.

[Klo21] Michael Klooß. “On Expected Polynomial Runtime in Cryptography”. In: *Proceedings of the 19th Theory of Cryptography Conference*. TCC ’21. 2021, pp. 558–590. DOI: 10.1007/978-3-030-90459-3_19.

[LM19] Russell W. F. Lai and Giulio Malavolta. “Subvector Commitments with Application to Succinct Arguments”. In: *Proceedings of the 39th Annual International Cryptology Conference*. CRYPTO ’19. 2019, pp. 530–560. DOI: 10.1007/978-3-030-26948-7_19.

[LPS24] Helger Lipmaa, Roberto Parisella, and Janno Siim. “Constant-Size zk-SNARKs in ROM from Falsifiable Assumptions”. In: *Proceedings of the 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT ’24. 2024, pp. 34–64. DOI: 10.1007/978-3-031-58751-1_2.

[LPS25] Helger Lipmaa, Roberto Parisella, and Janno Siim. “On Knowledge-Soundness of Plonk in ROM from Falsifiable Assumptions”. In: *Proceedings of the 45th Annual International Cryptology Conference*. CRYPTO ’25. 2025, pp. 362–395. DOI: 10.1007/978-3-032-01907-3_12.

[Ope26] OpenAI. *The Exponential Time Hypothesis is False*. Preprint, October 2026. 2026. URL: https://github.com/openai/math.

[Wik24] Douglas Wikström. “Special Soundness Revisited”. In: *IACR Communications in Cryptology* 1.3 (2024). DOI: 10.62056/aep2c3w9p.