On expected polynomial runtime in cryptography

Michael Klooß

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

Abstract. A common definition of black-box zero-knowledge considers *strict polynomial time* (PPT) adversaries but *expected polynomial time* (EPT) simulation. This is necessary for constant round black-box zero-knowledge in the plain model, and the asymmetry between simulator and adversary an accepted consequence. Consideration of EPT adversaries naturally leads to *designated* adversaries, i.e. adversaries which are only required to be efficient in the protocol they are designed to attack. They were first examined in Feige's thesis [9], where obstructions to proving security are shown. Prior work on (designated) EPT adversaries by Katz and Lindell (TCC'05) requires superpolynomial hardness assumptions, whereas the work of Goldreich (TCC'07) postulates "nice" behaviour under rewinding.

In this work, we start from scratch and revisit the definition of *efficient* algorithms. We argue that the standard runtime classes, PPT and EPT, behave "unnatural" from a cryptographic perspective. Namely, algorithms can have indistinguishable runtime distributions, yet one is considered efficient while the other is not. Hence, classical runtime classes are not "closed under indistinguishability", which causes problems. Relaxations of PPT which are "closed" are (well-)known and used.

We propose *computationally* expected polynomial time (CEPT), the class of runtimes which are (computationally) indistinguishable from EPT, which is "closed". We analyze CEPT in the setting of *uniform complexity* (following Goldreich (JC'93)) with *designated adversaries*, and provide easy-to-check criteria for zero-knowledge protocols with black-box simulation in the plain model which show that many (all known?) such protocols handle designated CEPT adversaries in CEPT.

1 Introduction

Interactive proof systems allow a prover $\mathcal{P}$ to convince a verifier $\mathcal{V}$ of the "truth" of a statement x, i.e. that $x \in \mathcal{L}$ for some language $\mathcal{L}$. Soundness of the protocol ensures that if the verifier accepts, then $x \in \mathcal{L}$ with high probability. *Zero-knowledge* proof systems allow $\mathcal{P}$ to convince $\mathcal{V}$ of $x \in \mathcal{L}$ *without revealing anything else*. The definition of zero-knowledge relies on the (more general) simulation paradigm: It stipulates that, for every (malicious) verifier $\mathcal{V}^*$, there is a simulator Sim which, given only the inputs x, *aux* of $\mathcal{V}^*$, can produce a *simulated* output (or *view*[1]) $out = \mathsf{Sim}(x, aux)$, which is indistinguishable from the output

[1] We use view and output synonymously in the introduction.

$\mathsf{out}_{\mathcal{V}^*}\langle \mathcal{P}(x,w), \mathcal{V}^*(x,aux)\rangle$ of a real interaction. Thus, anything $\mathcal{V}^*$ learns in the interaction, it could simulate itself — *if* Sim *and* $\mathcal{V}^*$ *lie in the same complexity class.*

Let us write X/Y (zero-knowledge) for adversary complexity X and simulator complexity Y. The two widespread notions of zero-knowledge are PPT/PPT and PPT/EPT. The former satisfies the "promise of zero-knowledge", but comes at a price. Barak and Lindell [2] show that it is impossible to construct *constant* round proof systems with *black-box* simulation and negligible soundness error in the plain model. Since *constant round black-box* zero-knowledge is attractive for many reasons, the relaxation of PPT/EPT zero-knowledge is common. However, this asymmetry breaks the "promise of zero-knowledge". The adversary *cannot* execute Sim, hence it cannot simulate the interaction. More concretely, this setting does not compose well. If we incorporate an EPT simulator into a (previously PPT) adversary, the new adversary is EPT. This common approach — constructing simulators for more complex systems from simulators of building blocks — therefore fails due to the asymmetry.

To remedy the asymmetry, we need to handle EPT adversaries. There are several sensible definitions of EPT adversaries, but the arguably most natural choice are *designated* EPT adversaries. That is, adversaries which only need to be *EPT when interacting with the protocol they are designed to attack*. Feige [9] first considered this setting, and demonstrates significant technical obstacles against achieving security in the presence of such attacks.

The problems of EPT (and designated adversaries) are not limited to zero-knowledge, and extend to the simulation paradigm, e.g. multi-party computation.

Preliminary conventions. Throughout, κ denotes the security parameter. We generally consider objects which are families (of objects) parameterized by κ, but often leave the dependency implicit. We abbreviate *systems of (interactive) machines (or algorithms)* by *system*. A system is *closed*, if it only expects κ as input, and produces some output. For example, a prover $\mathcal{P}$ does not constitute a closed system, nor does the interaction $\langle \mathcal{P}, \mathcal{V}\rangle$, since it still lacks the inputs to $\mathcal{P}$ and $\mathcal{V}$. Our primary setting is *uniform complexity* [11], where inputs to an (otherwise closed) system are generated efficiently by so-called *input generators*. Interaction of algorithms A, B is denoted $\langle \mathsf{A}, \mathsf{B}\rangle$, the time spent in A is denoted $\mathsf{time}_{\mathsf{A}}(\langle \mathsf{A}, \mathsf{B}\rangle)$, and similarly for time spent in B or $\mathsf{A}+\mathsf{B}$. Oracle access to $\mathcal{O}$ is written $\mathsf{A}^{\mathcal{O}}$. An algorithm A is *a priori* efficient, if the runtime bound is independent from its environment, e.g. classical "a priori PPT". The term *a posteriori* emphasizes an absence of a priori efficiency, i.e. bounds which depend on the environment, e.g. in the case of designated adversaries.

1.1 Obstacles

We first recall some obstacles regarding expected runtime and designated adversaries which we have to keep in mind. For more discussions and details, we refer to the excellent introductions of [19, 14] and to [9, Section 3].

Runtime squaring. Consider (a family of) random variables T_κ over $\mathbb{N}$, where $\mathbb{P}(T_\kappa = 2^\kappa) = 2^{-\kappa}$ and T_κ is 0 otherwise. Then T_κ has polynomially bounded expectation $\mathbb{E}(T_\kappa) = 1$, but $\mathbb{E}(T_\kappa^2) = 2^\kappa$. That is $S_\kappa = T_\kappa^2$ is *not* expected polynomial time anymore. This behaviour not only prevents machine model independence of EPT as an efficiency notion, but also the non-black-box simulation technique of Barak [1] (which suffers from a quadratic growth in runtime).

Composition and rewinding. Consider an oracle algorithm $\mathsf{A}^{\mathcal{O}}$ with access to a PPT oracle $\mathcal{O}$. Then to check if the total time $\mathsf{time}_{\mathsf{A}+\mathcal{O}}(\mathsf{A}^{\mathcal{O}})$ is PPT, we can count an oracle call as a single step. Moreover, it makes no difference if A has "straightline" or "rewinding" access to $\mathcal{O}$. For EPT, even a standalone definition of "$\mathcal{O}$ is EPT" is non-trivial and possibly fragile. For example, there are oracles, where any PPT A with "straightline" access to $\mathcal{O}$ results in an EPT interaction, yet access "with rewinding" to $\mathcal{O}$ allows an explosion of expected runtime. See [19] for a concrete example.

Designated EPT adversaries. For a **designated adversary** $\mathcal{A}$ against zero-knowledge of a proof system $(\mathcal{P}, \mathcal{V})$, we require (only) that $\mathcal{A}$ is *efficient when interacting with that protocol*. Since a zero-knowledge simulator *deviates* from the real protocol, the runtime guarantees of $\mathcal{A}$ are void.

1.2 Motivation: Reproving zero-knowledge of graph 3-colouring

The constant-round black-box zero-knowledge proof of Goldreich and Kahan [15] is our running example for demonstrating problems and developing our approach.

Recall that (non-interactive) commitment schemes allow a committer to commit to a value in a way which is *hiding* and *binding*, i.e. the commitment does not reveal the value to the receiver, yet it can be unveiled to at most one value. A commitment scheme consists of algorithms $(\mathsf{Gen}, \mathsf{Com}, \mathsf{VfyOpen})$. The *commitment key* is generated via $\mathsf{ck} \leftarrow \mathsf{Gen}(\kappa)$.

The constant round protocol of Goldreich–Kahan The protocol of [15] uses two different commitments, $\mathsf{Com}^{(\mathrm{H})}$ is perfectly hiding, $\mathsf{Com}^{(\mathrm{B})}$ is perfectly binding. The idea of protocol $\mathrm{G3C}_{\mathrm{GK}}$ is a parallel, N-fold, repetition of the standard zero-knowledge proof for G3C, with the twist that the verifier commits to all of its challenges beforehand. Let $G = (V, E)$ be the graph and let ψ be a 3-colouring of G. The prover is given (G, ψ) and the verifier G.

(P0) $\mathcal{P}$ sends $\mathsf{ck}_{\mathrm{hide}} \leftarrow \mathsf{Gen}^{(\mathrm{H})}(\kappa)$. ($\mathsf{ck}_{\mathrm{bind}} \leftarrow \mathsf{Gen}^{(\mathrm{B})}(\kappa)$ is deterministic.)
(V0) $\mathcal{V}$ picks $N = \kappa \cdot \mathsf{card}(E)$ challenge edges $e_i \leftarrow E$, and commits to them using $\mathsf{Com}^{(\mathrm{H})}$.
(P1) $\mathcal{P}$ picks randomized colourings for each of the N parallel repetitions of the standard graph 3-colouring proof system, and sends the $\mathsf{Com}^{(\mathrm{B})}$-committed randomized node colours to $\mathcal{V}$.
(V1) $\mathcal{V}$ opens all commitments (to e_i).

(P2) $\mathcal{P}$ aborts if any opening is invalid. Otherwise, $\mathcal{P}$ proceeds in the parallel repetition using these challenges, i.e. in the i-th repetition $\mathcal{P}$ opens the committed colours for the nodes of edge e_i.

(V2) $\mathcal{V}$ aborts iff any edge not correctly coloured, or if $\mathrm{ck}_{\mathrm{hide}}$ is "bad". Else $\mathcal{V}$ accepts.

The soundness of this protocol follows from $\mathsf{Com}^{(\mathrm{H})}$ being perfectly hiding. Therefore, each of the N parallel repetitions is essentially an independent repetition of the usual graph 3-colouring proof. For $N = \kappa \cdot \mathsf{card}(E)$ parallel rounds, the probability to successfully cheat is negligible (in κ), see [15].

Proving zero-knowledge: A (failed?) attempt Now, we prove black-box zero-knowledge for *designated adversaries*. That is, we describe a simulator which uses the adversary $\mathcal{V}^*$ only as a black-box, which can be queried and rewound to a (previous) state. We proceed in three game hops, gradually replacing the view of a real interaction with a simulated view. Successive games are constructed so that their change in output (which is a purported view) is indistinguishable.

G_0 This is the real G3C protocol. The output is the real view.

G_1 The prover rewinds a verifier which completes (V1) successfully (i.e. sends *valid* openings on the first try) to (V0) and repeats (P1) until a second run where $\mathcal{V}$ validly opens all commitments. The output is the view of this second succesful run. The prover uses fresh randomness in each reiteration of (P1) (whereas the black-box has fixed randomness).

G_2 If the two openings in (V1) differ, return `ambig`, indicating ambiguity of the commitment. Otherwise, proceed unchanged.

G_3 The initial commitments (in (P1)) to a 3-colouring are replaced with commitments to 0. These commitments are never opened. In successive iterations, the commitments to a 3-colouring are replaced by commitments to pseudo-colourings ψ_i (for e_i), i.e. for edge $e_i = (u_i, v_i)$, ψ_i colours u_i and v_i differently (and uniformly), whereas ψ_i colours all $v \neq u_i, v_i$ with 0. Hence the opened commitments simulate a valid 3-colouring at the challenge edges e_i.

Evidently, Game G_3 outputs a purported view independent of the witness. Thus, the simulator is defined as in G_3: In a first try, it commits (using $\mathsf{Com}^{(\mathrm{B})}$) to all zeroes instead of a 3-colouring in (P1), and uses this "garbage" commitment to learn the verifier's challenge (in (V1)). If the verifier does not successfully open the commitments (in (V1)), Sim aborts (as an honest prover would) and outputs the respective view. Otherwise, Sim rewinds the verifier to Step 2 and sends a pseudo-colouring (w.r.t. the previously revealed challenge) instead. Sim retries until the verifier succesfully unveils (in (V1)) again. (If the verifier opens to a different challenge, return $\mathit{view} = \mathtt{ambig}$.)

Now, we sketch a security proof for Sim. We argue by game hopping.

G_0 to G_1. The expected number of rewinds is at most 1. Namely, if $\mathcal{V}^*$ opens in (V1) with probability ε, then an expected number of $\frac{1}{\varepsilon}$ rewinds are required. Consequently, the expected runtime is polynomial (and G_1 is EPT). The output distribution of the games is identical.

G_1 to G_2. It is easy to obtain an adversary against the binding property of $\mathsf{Com}^{(\mathrm{H})}$ which succeeds with the same probability that G_2 outputs `ambig`. Thus, this probability is negligible.

G_2 to G_3. Embedding a (multi-)hiding game for $\mathsf{Com}^{(\mathrm{B})}$ in this step is straightforward. Namely, using the left-or-right indistinguishability formulation, where the commitment oracle either commits the first or second challenge message. Thus, by security of the commitment scheme, G_2 and G_3 are indistinguishable.[2]

A closer look. The above proof is clear and simple. But the described simulator is not EPT! While G_2 and G_3 are (computationally) indistinguishable, the transition *does not necessarily preserve expected polynomial runtime* [9, 19]. Feige [9] points out a simple attack, where $\mathcal{V}^*$ brute-forces the commitments with some tiny probability p, and runs for a very long time if the contents are not valid 3-colourings. This is EPT in the real protocol, but our simulator as well as the simulator in [15] do not handle $\mathcal{V}^*$ in EPT. The problem lies with *designated* adversaries as following example shows.

Example 1. Let $\mathcal{V}^*$ sample in step (V0) a "garbage" commitment c' to zeroes, using $\mathsf{Com}^{(\mathrm{B})}$ just like Sim in its first step, trying to predict Sim's choice. (c' is a "proof of simulation".) Now $\mathcal{V}^*$ unveils e in (V1) if and only if it receives c'. The honest prover always aborts in (P2) because $\mathcal{V}^*$ will never unveil. However, if Sim happens to chose $c = c'$ as its "garbage" commitment, the simulation runs forever, because $\mathcal{V}^*$ unveils only for this c', which is not a pseudo-colouring.

As described, $\mathcal{V}^*$ is a priori PPT, and indeed, the simulator in [15] uses a "normalization technique" which prevents this attack. However, exploiting *designated* PPT, $\mathcal{V}^*$ may instead run for a very long time, when it receives c'.

Obstructions to simple fixes. Let us recall a few simple, but insufficient fixes. A first idea is to *truncate* the execution of $\mathcal{A}$ at some point. For PPT adversaries, this may seem viable.[3] However, there are EPT adversaries, or more concretely runtime distributions, where *any strict polynomial truncation* affects the output in the real protocol *noticeably*. So we cannot expect that such a truncation works well for Sim. See [9, Section 3] for a more convincing argument against truncation.

Being unable to truncate, we could enforce better behaviour on the adversary. Intuitively, it seems enough to require that $\mathcal{V}^*$ runs in expected polynomial time *in any interaction* [19, 14]. However, even this is not enough. Katz and Lindell [19] exploit the soundness error of the proof system to construct an adversary which runs in expected polynomial time in any interaction, but still makes the

[2] We rely on security of binding and hiding against *expected time* adversaries, which follows from PPT-security by runtime truncation arguments, e.g. by Lemma 1.

[3] Even there, the situation is far from easy. In a UC setting with an *a posteriori* efficiency notion (and designated adversaries), Hofheinz, Unruh, and Müller-Quade show in [18, Section 9] that (pathological) functionalities can make simulation in PPT impossible (if one wants security under composition for just a single instance).

expected runtime of the simulator superpolynomial. The problem is that these runtime guarantees are void in the presence of *rewinding*.

Modifications of these fixes work, but at a price: Katz and Lindell [19] use *superpolynomial* truncation and need to assume *superpolynomial* hardness. Goldreich [14] *restricts* to algorithms (hence adversaries) which behave well under rewinding. We discuss these in Section 1.5. Our price will be proof techniques, which become more technical and, perhaps, more limited.

Our fix: There is no problem. Our starting point is *the conviction* that the given "proof" should *evidently* establish the security of the scheme for any *cryptographically sensible* notion of runtime. If one could *distinguish* the runtime of G_2 and G_3, then this would break the hiding property of the commitment scheme. Thus, the *runtimes are indistinguishable*. Following, in computational spirit, Leibniz' "identity of indiscernibles", we declare runtimes which are *indistinguishable from efficient* by efficient distinguishers as *efficient per definition*. With this, the proof works and the simulator, while not expected polynomial time, is *computationally expected polynomial time* (CEPT), which means its runtime distribution is indistinguishable from EPT.

We glossed over a crucial detail: We solved the problem with the very strategy we claim to fix — different runtime classes for Sim and $\mathcal{V}^*$! Fortunately, Sim also handles CEPT adversaries in CEPT.

1.3 Contribution

Our main contribution is the reexamination of the notion of runtime in cryptography. We offer a novel, and arguably natural, alternative solution for a problem that was never fully resolved. Our contribution is therefore primarily of explorational and definitional nature. More concretely:

- We define CEPT, a small relaxation of EPT with a simple characterization.
- To the best of our knowledge, this is the first work which embraces *uniform*[4] complexity, *expected* time, and *designated* adversaries.
- We develop general tools for this setting, most importantly, a hybrid lemma.
- Easy-to-check criteria show that many (all known?) black-box zero-knowledge arguments from standard assumptions in the plain model[5] have CEPT simulators which handle designated CEPT adversaries. Consequently, security against designated adversaries is natural. For example, the proof systems [16, 15, 24, 29, 20, 28] satisfy our criteria.
- We impose no (non-essential) restrictions on the adversary, nor do we need additional (hardness) assumptions.
- We sketch the application of our techniques to secure function evaluation.

[4] Our results are applicable to a minor generalization of the non-uniform setting as well, namely non-uniformly generated input *distributions*.

[5] Unfortunately, problems might arise with superpolynomial hardness assumptions.

All of this comes at a price. Our notions and proofs are not complicated, yet somewhat technical. This is, in part, because of a posteriori runtime and uniform complexity. Still, we argue that we have demonstrated the viability of our new notion of efficiency, at least for zero-knowledge.

A complexity theoretic perspective. This work is only concerned with the complexity class of feasible *attacks*, and does not assume or impose complexity requirements on protocols. Due to designated adversaries, the complexity class of adversaries is (implicitly) defined per protocol, similar to [19]. We bootstrap feasibility from complexity classes for (standalone) sampling algorithms, i.e. algorithms with no inputs except κ. Hence a (designated) adversary is feasible if the *completed system* of protocol and adversary (including input generation) is CEPT (or more generally, in some complexity class of feasible sampling algorithms).

The complexity class of simulators is relative to the adversary, and thus depends both on the protocol and the ideal functionality. Namely, feasibility of a simulator Sim means that if an adversary $\mathcal{A}$ is feasible (w.r.t. the protocol), then "$\mathsf{Sim}(\mathcal{A})$" is feasible (w.r.t. the ideal functionality).

Comments on our approach. The uniform complexity setting drives complexity, yet is necessary, since a notion of time that depends on non-uniformity is rather pathological. Losing the power of non-uniformity (and strictness of PPT) requires many small adjustments to definitions. Moreover, annoying technical problems with efficiency arise inadvertently, depending on formalizations of games and models. As in prior work, we mostly ignore them, but do point them out and propose solutions. They are easily fixed by adding "laziness", "indirection", or "caching".

An important point raised by a reviewer of TCC'20 is the "danger of zero-knowledge being trivialized" by "expanding the class of attacks", and a case for "moving towards knowledge tightness" (with which we fully agree). Many variations of zero-knowledge, from weak distributional [8, 6] to precise [25, 7], exist. We argue that our notion is very close to the "standard" notion with EPT simulation, but allows designated (C)EPT adversaries. Indeed, it seems to gravitate towards "knowledge tightness" [14], as seen by runtime explosion examples due to expectation.

1.4 Technical overview and results

We give an overview of our techniques, definitions, and results. Recall that we only consider runtimes for closed systems (which receive only κ as input and produce some output). W.r.t. *uniform complexity* and *designated adversaries*, i.e. adversaries which only need to be efficient in the real protocol [9], closed systems are the default situation anyway. A **runtime class** $\mathcal{T}$ is a set of runtime distributions. A **runtime (distribution)** is a family $(T_\kappa)_\kappa$ of distributions T_κ over $\mathbb{N}_0$. We use *runtime* and *runtime distribution* synonymously. Computational $\mathcal{T}$-time indistinguishability of oracles and distributions is defined in the

obvious way (c.f. Definition 5). For statistical $\mathcal{T}$-query indistinguishability, we count *only* queries as steps, and require $\mathcal{T}$-time w.r.t. this. (In our setting, unbounded queries often imply perfect indistinguishability, which is too strong.)

The basic tools.

Statistical vs. computational indistinguishability. The (folklore) *equivalence* of statistical and computational indistinguishability for distributions with *"small" support* is a simple, but central, tool. For polynomial runtime, "small" support means polynomial support, say $\{0, \dots, \mathsf{poly}_1(\kappa)\}$. Assuming non-uniform advice, the advice is large enough to encode the optimal decisions, achieving statistical distance as distinguishing advantage. This extends to "polynomially-tailed" runtime distributions T. There, by assumption, for any poly_0 there is a poly_1 such that $\mathbb{P}(T_\kappa > \mathsf{poly}_1(\kappa)) \leq \frac{1}{\mathsf{poly}_0(\kappa)}$, Hence, we can reduce to strict polynomial support by truncating at poly_1, sacrificing $1/\mathsf{poly}_0$ in statistical distance. The Markov bound shows that expected polynomial time is polynomially tailed. Removing non-uniformity is possible with repeated sampling, e.g. by approximating the distribution.

Standard reduction. Another simple, yet central, tool is the *standard cutoff argument*. It is the core tool to obtain *efficiency from indistinguishability*.

Lemma 1 (Standard reduction to PPT). *Let $\mathcal{D}$ be a distinguisher for two oracles $\mathcal{O}_0$, $\mathcal{O}_1$. Suppose $\mathcal{D}$ has advantage at least $\varepsilon \geq \frac{1}{\mathsf{poly}_{\mathrm{adv}}}$ (infinitely often). Suppose furthermore that $\mathcal{D}^{\mathcal{O}_0}$ is EPT (even CEPT) with expected time poly_0. Then there is an a priori PPT distinguisher $\mathcal{A}$ with advantage at least $\frac{\varepsilon}{4}$ (infinitely often). (Here, $\varepsilon, \mathsf{poly}_{\mathrm{adv}}, \mathsf{poly}_0$ are functions in κ.)*

We stress that we require *no runtime guarantees* for $\mathcal{D}^{\mathcal{O}_1}$ — it may never halt. For a proof sketch, define $N = 4\mathsf{poly}_0 \cdot \mathsf{poly}_{\mathrm{adv}}$ and let $\mathcal{A}$ be the runtime cutoff of $\mathcal{D}$ at N. The outputs of $\mathcal{A}^{\mathcal{O}_0}$ and $\mathcal{D}^{\mathcal{O}_0}$ are $\frac{\varepsilon}{4}$ close. For $\mathcal{A}^{\mathcal{O}_1}$ and $\mathcal{D}^{\mathcal{O}_1}$ this may be false. However, if $\mathcal{D}^{\mathcal{O}_1}$ exceeds N steps with probability higher than $\frac{2\varepsilon}{4}$, then the runtime is a distinguishing statistic with advantage $\frac{\varepsilon}{4}$. Thus, we can assume the outputs of $\mathcal{A}^{\mathcal{O}_1}$ and $\mathcal{D}^{\mathcal{O}_1}$ are $\frac{2\varepsilon}{4}$ close. Now, a short calculation shows that $\mathcal{A}$ has advantage at least $\frac{\varepsilon}{4}$. Namely, $\Delta(\mathcal{A}^{\mathcal{O}_1}, \mathcal{A}^{\mathcal{O}_0}) \geq \Delta(\mathcal{D}^{\mathcal{O}_1}, \mathcal{D}^{\mathcal{O}_0}) - \Delta(\mathcal{A}^{\mathcal{O}_1}, \mathcal{D}^{\mathcal{O}_1}) - \Delta(\mathcal{D}^{\mathcal{O}_0}, \mathcal{A}^{\mathcal{O}_0})$.

Computationally expected polynomial time. We define the runtime classes $\mathcal{PPT}$ (resp. $\mathcal{EPT}$), as usual, i.e. $(T_\kappa)_\kappa \in \mathcal{PPT} \iff \exists \mathsf{poly}\colon \mathbb{P}(T_\kappa \leq \mathsf{poly}(\kappa)) = 1$ (resp. $(T_\kappa)_\kappa \in \mathcal{EPT} \iff \exists \mathsf{poly}\colon \mathbb{E}(T_\kappa) \leq \mathsf{poly}(\kappa)$).

Definition 1 (Simplified[6] Definition 8). *A runtime S, i.e. a family of random variables S_κ with values in $\mathbb{N}_0$, is* ***computationally expected polynomial***

[6] Formally, "triple-oracle" instead of "standard" indistinguishability is used. Assuming non-uniform advice, or runtimes T, S which are induced by algorithms, the simplified definition is equivalent to the actual one.

***time* (CEPT)**, *if there exists a runtime T which is (perfectly) expected polynomial time (i.e. EPT), such that any a priori PPT distinguisher has negligible distinguishing advantage for the distributions T and S. The class of CEPT runtime distributions is denoted $\mathcal{CEPT}$.* ***Computationally strict polynomial time* (CPPT)** *is defined analogously.*

Characterizing CEPT. At a first glimpse, CEPT looks hard to handle. Fortunately, this is a mirage. We have following characterization of CEPT.

Proposition 1 (Simplified[6] Corollary 1). *Let T be a runtime. The following are equivalent:*

(0) T is in $\mathcal{CEPT}$.
(1) $\exists S \in \mathcal{EPT}$ which is computationally *PPT-indistinguishable from T.*
(2) $\exists S \in \mathcal{EPT}$ s.t. T and S are statistically indistinguishable (given polynomially many samples).
(3) There is a set of good events $\mathcal{G}_\kappa$ with $\mathbb{P}(\mathcal{G}_\kappa) \geq 1-\varepsilon(\kappa)$ such that $\mathbb{E}(T_\kappa \,|\, \mathcal{G}_\kappa) \leq t_\kappa$ (for the conditional expectation), where ε is negligible and t is polynomial.

Let T be a runtime. Item (3) defines **virtually expected time** (t, ε) with *virtual expectation* (bounded by) t and *virtuality* ε. Thus, the characterization says that computational, statistical and virtual EPT coincide.

Proposition 1 follows essentially from the statistical-to-computational reduction and a variant of Lemma 1. Thanks to this characterization, working with CEPT is feasible. One uses item (1) to justify that indistinguishability transitions preserve CEPT. And one relies on item (3) to simplify to the case of EPT, usually in unconditional transitions, such as efficiency of rewinding.

An intrinsic characterization. The full Corollary 1 not only reveals that CEPT is "well-behaved". It also shows that the runtime class $\mathcal{CEPT}$ is "closed under indistinguishability": Any runtime S which is CEPT-indistinguishable from some $T \in \mathcal{CEPT}$ lies in $\mathcal{CEPT}$. This intrinsic property sets it apart from EPT. (Indeed, $\mathcal{CEPT}$ is the closure of $\mathcal{EPT}$.) PPT and CPPT behave analogously.

Example 2. Let A be an algorithm which outputs 42 in exactly 10^{10} steps, and let A' act identical to A, except with probability $2^{-\kappa}$, in which case it runs $2^{2\kappa}$ steps. Then A' is neither PPT nor EPT. Yet, A and A' are indistinguishable even given *timed* black-box access. That is, observing both output and runtime of the black-box, it is not possible to tell A and A' apart. Thus, it is rather unexpected that A' is considered inefficient. For many properties, e.g. correctness or soundness, statistical relaxations from "perfect" exist. CPPT and CEPT should be viewed as such relaxations for efficiency.

Working with CEPT. Applying the characterization of CEPT to a whole system $\langle \mathcal{P}, \mathcal{V}^* \rangle$, the good event $\mathcal{G}$ may induce arbitrary stochastic dependencies on (internal) random coins of the parties. This is inconvenient. We are interested only in one party, namely $\mathcal{V}^*$. Moreover, in a simulation, there is no $\mathcal{P}$ anymore

and the probability space changed, hence there is no event $\mathcal{G}$. To account for this, we observe that only the messages $\mathcal{V}^*$ receives from $\mathcal{P}$ are relevant for $\mathcal{V}^*$'s behaviour, not $\mathcal{P}$'s internal randomness. In the full version [21], we formulate a convenience lemma for handling this, whereas in this extended abstract, we deal with it directly.

Definitions and tools for zero-knowledge. Here, we state our definition of uniform complexity zero-knowledge, demonstrate how to prove zero-knowledge for $\mathrm{G3C}_{\mathrm{GK}}$, and then abstract the approach to cover a large class of protocols.

Definition of zero-knowledge. For uniform auxiliary input zero-knowledge, the input $(x, w, aux, state) \leftarrow \mathcal{I}(\kappa)$ is efficiently generated by an *input generator* $\mathcal{I}$. A designated adversary $(\mathcal{I}, \mathcal{V}^*)$ consists of input generation, malicious verifier, and distinguisher, but we leave $\mathcal{I}$ often implicit. The distinguisher receives *out* and *state*, the latter is needed for modular sequential composition.[7] Here, $out = \mathsf{out}_{\mathcal{V}^*}\langle \mathcal{P}(x,w), \mathcal{V}^*(x, aux)\rangle$ or $out = \mathsf{out}_{\mathsf{Sim}}\mathsf{Sim}(\mathsf{code}(\mathcal{V}^*), x, aux)$, where $(x, w, aux, state)$ is sampled by $\mathcal{I}(\kappa)$. As a shorthand, for the system which lets $\mathcal{I}$ sample inputs and passes them as above, we write $\langle \mathcal{P}, \mathcal{V}\rangle_{\mathcal{I}}$. From designated CEPT adversaries, we require that $\mathsf{time}_{\mathcal{I}+\mathcal{P}+\mathcal{V}^*+\mathcal{D}}((state, \mathsf{out}_{\mathcal{V}^*}\mathcal{P}(x,w), \mathcal{V}^*(x, aux)))$ is CEPT.

Concrete example. Recall that in Section 1.2, we showed zero-knowledge of the graph 3-colouring protocol $\mathrm{G3C}_{\mathrm{GK}}$ of Goldreich and Kahan [15] as follows:

Step 1: Introduce all rewinding steps as in G_1. Here, virtually expected runtime and virtuality at most doubles. Roughly, rewinding at most doubles the probability that a query *query* is asked. Since this, in particular, applies to long running "bad" queries, virtuality at most doubles.

Step 2: Apply indistinguishability transitions, which reduce to hiding resp. binding properties of the commitment. From this, we obtain both good output quality and efficiency of Sim. Concretely, indistinguishability and efficiency follow by an application of the standard reduction (to PPT).

We abstract this strategy to cover a large class of zero-knowledge proofs.[8] Intuitively, we apply the ideas of [14] ("normality") and [19] ("query indistinguishability"), but separate the unconditional part (namely, that rewinding preserves efficiency), and the computational part (namely, that simulated queries preserve efficiency).[9]

Abstracting Step 1 (Rewinding strategies). A **rewinding strategy** RWS has black-box rewinding ($\mathsf{bb\text{-}rw}$) access to a malicious verifier $\mathcal{V}^*$, and abstracts a simulator's rewinding behaviour. Unlike the simulator, RWS has access to the witness. For RWS to be **normal**, we impose three requirements.

[7] While [11] passes no extra *state*, only sequential *repetition* is proven there.

[8] Strictly speaking, we concentrate on zero-knowledge *arguments*, since we need efficient provers.

[9] We significantly deviate from [19] to obtain simpler reductions.

Firstly, a normal rewinding strategy outputs an adversarial view which is *distributed (almost) as in the real execution*. Secondly, there is some poly so that

$$\mathbb{E}(\mathsf{time}_{\mathsf{RWS}+\mathcal{V}^*}(\mathsf{RWS}^{\mathcal{V}^*})) \leq \mathsf{poly}(\kappa) \cdot \mathbb{E}(\mathsf{time}_{\mathcal{P}+\mathcal{V}^*}(\langle \mathcal{P}, \mathcal{V}^* \rangle))$$

for any adversary $\mathcal{V}^*$. We call this (polynomial) **runtime tightness** of RWS.[10] Thirdly, RWS has (polynomial) **probability tightness**, which is defined as follows: Let $\mathsf{pr}_{\mathrm{rws}}(\mathit{query})$ be the probability that RWS asks $\mathcal{V}^*$ a query *query*. Let $\mathsf{pr}_{\mathrm{real}}(\mathit{query})$ be the probability that the prover $\mathcal{P}$ asks *query*. Then RWS has probability tightness poly if for all queries *query*

$$\mathsf{pr}_{\mathrm{rws}}(\mathit{query}) \leq \mathsf{poly}(\kappa) \cdot \mathsf{pr}_{\mathrm{real}}(\mathit{query}).$$

Intuitively, runtime tightness ensures that RWS preserves EPT, whereas probability tightness bounds the growth of virtuality. Indeed, the virtuality δ in $\langle \mathcal{P}, \mathcal{V}^* \rangle$ increases to at most $\mathsf{poly} \cdot \delta$ in $\mathsf{RWS}^{\mathcal{V}^*}$. This follows because the probability for a "bad" query in $\mathsf{RWS}^{\mathcal{V}^*}$ is at most poly-fold higher than in $\langle \mathcal{P}, \mathcal{V}^* \rangle$.

Lemma 2 (Informal). *Let* RWS *be a normal rewinding strategy for* $(\mathcal{P}, \mathcal{V})$ *with runtime and probability tightness* poly*. Let* $(\mathcal{I}, \mathcal{V}^*)$ *be an adversary. If* $\langle \mathcal{P}, \mathcal{V}^* \rangle_{\mathcal{I}}$ *is CEPT with virtually expected time* (t, ε)*, then* $\mathsf{RWS}(\mathcal{V}^*)$ *composed with* $\mathcal{I}$ *is CEPT with virtually expected time* $(\mathsf{poly} \cdot t, \mathsf{poly} \cdot \varepsilon)$*.*

(Weak) relative efficiency. We generalize the guarantees of rewinding strategies to *relative efficiency* of (oracle) algorithms. An oracle algorithm B is **efficient relative to** A with **runtime tightness** $(\mathsf{poly}_{\mathsf{time}}, \mathsf{poly}_{\mathsf{virt}})$ if for all oracles $\mathcal{O}$: If $\mathsf{time}_{\mathsf{A}+\mathcal{O}}(\mathsf{A}^{\mathcal{O}})$ is virtually expected (t, ε)-time, then $\mathsf{time}_{\mathsf{B}+\mathcal{O}}(\mathsf{B}^{\mathcal{O}})$ is virtually expected $(\mathsf{poly}_{\mathsf{time}} \cdot t, \mathsf{poly}_{\mathsf{virt}} \cdot \varepsilon)$-time.

We call B **weakly efficient relative** to A, if whenever $\mathsf{time}_{\mathsf{A}+\mathcal{O}}(\mathsf{A}^{\mathcal{O}})$ is efficient (e.g. CEPT), then $\mathsf{time}_{\mathsf{B}+\mathcal{O}}(\mathsf{B}^{\mathcal{O}})$ is efficient (e.g. CEPT).

Abstracting Step 2 (Simple assumptions). A **"simple" assumption** is a pair of efficiently computable oracles $\mathcal{C}_0$ and $\mathcal{C}_1$, and the assumption that $\mathcal{C}_0 \overset{c}{\approx} \mathcal{C}_1$, i.e. $\mathcal{C}_0$ and $\mathcal{C}_1$ cannot be distinguished in PPT.[11] For example, hiding resp. binding for commitment schemes are simple assumptions.

In Step 2, we reduce the indistinguishability of $\mathsf{RWS}^{\mathcal{V}^*}$ and $\mathsf{Sim}^{\mathcal{V}^*}$ to a simple assumption. That is, there is some algorithm R such that $\mathsf{RWS}^{\mathcal{V}^*} \equiv \mathrm{R}^{\mathcal{C}_0}(\mathcal{V}^*)$, and $\mathrm{R}^{\mathcal{C}_1}(\mathcal{V}^*) \equiv \mathsf{Sim}^{\mathcal{V}^*}$. Moreover, we assume that $\mathrm{R}^{\mathcal{C}_0}(\mathcal{V}^*)$ is efficient relative to $\mathsf{RWS}^{\mathcal{V}^*}$, and $\mathsf{Sim}^{\mathcal{V}^*}$ is efficient relative to $\mathrm{R}^{\mathcal{C}_1}(\mathcal{V}^*)$.

[10] Up to minor technical details, polynomial runtime tightness of RWS coincides with "normality" of Sim in [14, Def. 6].

[11] Technically, our definition of "simple assumption" corresponds to falsifiable assumptions [26] in the sense of [10]. We deliberately do not call them falsifiable, since our proof techniques should extend to a larger class of assumptions, which includes non-falsifiable assumptions.

Putting it together (Benign simulators). Black-box simulators whose security proof follows the above outline are called **benign**. See Fig. 1 for an overview of properties and their relation.

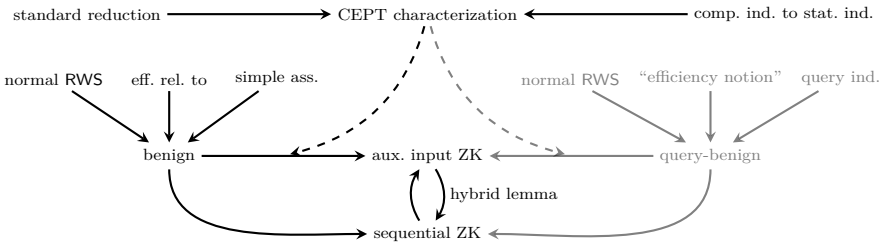

Fig. 1: A rough overview of dependencies of core results and definitions. The greyed out approach follows [19] more closely, see the full version [21]. The top line is used everywhere implicitly.

Lemma 3 (Informal). *Argument systems with* benign *simulators are* auxiliary-input zero-knowledge *against CEPT adversaries.*

Proof (Summary). The proof strategy above can be summarized symbolically:

$$\mathsf{out}_{\mathcal{V}^*}\langle \mathcal{P}, \mathcal{V}^* \rangle \equiv \mathsf{RWS}(\mathcal{V}^*) \equiv \mathrm{R}^{\mathcal{C}_0}(\mathcal{V}^*) \overset{c}{\approx} \mathrm{R}^{\mathcal{C}_1}(\mathcal{V}^*) \equiv \mathsf{Sim}(\mathcal{V}^*).$$

More precisely, consider a CEPT adversary $(\mathcal{I}, \mathcal{V}^*)$. By normality of RWS, $\mathsf{out}_{\mathcal{V}^*}\langle \mathcal{P}, \mathcal{V}^* \rangle$ and $\mathsf{RWS}(\mathcal{V}^*)$ have (almost) identical output distributions, and $\mathsf{RWS}(\mathcal{V}^*)$ is CEPT. By relative efficiency, $\mathrm{R}^{\mathcal{C}_0}(\mathcal{V}^*)$ is CEPT if $\mathsf{RWS}^{\mathcal{V}^*}$ is CEPT. Since $\mathcal{C}_0 \overset{c}{\approx} \mathcal{C}_1$, by a standard reduction, if $\mathrm{R}^{\mathcal{C}_0}(\mathcal{V}^*)$ is CEPT, so is $\mathrm{R}^{\mathcal{C}_1}(\mathcal{V}^*)$, and their outputs are indistinguishable. Finally, since $\mathsf{Sim}^{\mathcal{V}^*}$ is efficient relative to $\mathrm{R}^{\mathcal{C}_1}(\mathcal{V}^*)$, also $\mathsf{Sim}^{\mathcal{V}^*}$ is CEPT. All in all, $\mathsf{Sim}^{\mathcal{V}^*}$ is efficient and produces indistinguishable outputs.

Benign simulators are common, e.g. the classic, constant round, and concurrent zero-knowledge protocols in [16, 15, 24, 29, 20, 28] satisfy this property.

Sequential composition and hybrid arguments. It turns out that hybrid arguments are non-trivial in the setting of a posteriori efficiency. Here, we outline the challenges in proving the hybrid lemma, how to overcome them, and how to obtain security of sequential composition from our abstract hybrid lemma.

Intermezzo: Tightness bounds. The use of relative efficiency with polynomial tightnesss bounds is not strictly necessary. Nevertheless, it offers "more quantifiable" security and is easier to handle. For example, benign simulators are easily

seen to "compose sequentially" because, (1) normal RWS and relative efficiency compose sequentially, and (2) "simple" assumptions satisfy indistinguishability under "repeated trials". Together, this translates to sequential composition of benign simulation. Hence, argument systems with *benign* simulators are *sequential zero-knowledge* against CEPT adversaries. Unfortunately, the general case is much more involved.

The hybrid lemma. To keep things tidy, we consider an abstract hybrid argument, which applies to zero-knowledge simulation and much more. Due to a posteriori efficiency, the lemma is both non-trivial to prove and non-trivial to state.

Lemma 4 (Hybrid lemma). *Let $\mathcal{O}_0$ and $\mathcal{O}_1$ be two oracles and suppose that $\mathcal{O}_1$ is weakly efficient relative to $\mathcal{O}_0$ and $\mathcal{O}_0 \overset{c}{\approx} \mathcal{O}_1$. Denote by $\mathsf{rep}(\mathcal{O}_0)$ and $\mathsf{rep}(\mathcal{O}_1)$ oracles which give repeated access to independent instances of $\mathcal{O}_b$. Then $\mathsf{rep}(\mathcal{O}_1)$ is weakly efficient relative to $\mathsf{rep}(\mathcal{O}_0)$ and $\mathsf{rep}(\mathcal{O}_0) \overset{c}{\approx} \mathsf{rep}(\mathcal{O}_1)$.*

Lemma 4 hides much of the complexity caused by a posteriori efficiency, and is often a suitable black-box drop-in for the hybrid argument. We sketch how to adapt the usual hybrid reduction. In our setting, $\mathsf{rep}(\mathcal{O}_b)$ gives access to arbitrarily many independent instances of $\mathcal{O}_b$. The usual hybrids H_i use $\mathcal{O}_1$ for the first i instances, and switch to $\mathcal{O}_0$ for all other instances. W.l.o.g., only $q = \mathsf{poly}(\kappa)$ many $\mathcal{O}$-instances are accessed by the distinguisher $\mathcal{D}$. The hybrid distinguisher $\mathcal{D}'$ guesses an index $i^* \leftarrow \{0, \dots, q-1\}$, and simulates a hybrid H_{i+b} embedding its challenge oracle $\mathcal{O}_b^*$.

If $\mathcal{D}$ has advantage ε, then the hybrid distinguisher $\mathcal{D}'$ has advantage ε/q. In the classic PPT setting, we assume that $\mathcal{O}_0$ and $\mathcal{O}_1$ are classical PPT, and hence find that $\mathcal{D}'$ is PPT and therefore efficient. In an a posteriori setting, the efficiency of $\mathcal{D}'$ is a bigger hurdle. We make the minimal assumptions, that $\mathsf{time}_{\mathcal{D}+\mathsf{rep}(\mathcal{O}_0)}(\mathcal{D}^{\mathsf{rep}(\mathcal{O}_0)})$ is efficient and that $\mathcal{O}_1$ is weakly efficient relative to $\mathcal{O}_0$.[12] Hence, we do not trivially know whether $\mathsf{time}_{\mathcal{D}+\mathsf{rep}(\mathcal{O}_1)}(\mathcal{D}^{\mathsf{rep}(\mathcal{O}_1)})$ or the hybrid distinguisher $\mathcal{D}'$, which has to emulate many oracle instances, is efficient. Indeed, a naive argument would invoke weak relative efficiency q times. In the case of PPT, this would mean q-many polynomial bounds. But, for all we know, these could have the form $2^i\mathsf{poly}(\kappa)$ in the i-th invocation, leading to an inefficient simulation.

The core problem is therefore to avoid a superconstant application of weak relative efficiency.[13] Essentially this problem was encountered by Hofheinz, Unruh, and Müller-Quade [18] in the setting of universal composability and a posteriori PPT. They provide a nifty solution, namely to *randomize the oracle indexing*. This ensures that, in each hybrid, every emulation of $\mathcal{O}_0$ (resp. $\mathcal{O}_1$) has identical runtime distribution T_0 (resp. T_1). This gives a uniform bound on runtime

[12] The hybrid proof technique requires the hybrid distinguisher to emulate all but one oracle instance, and for this we need weak relative efficiency.

[13] For reference, even for a priori PPT sequential composition for zero-knowledge, one must avoid a superconstant invocation of the existence of simulators. There, the solution is to consider a "universal" adversary and its "universal" simulator.

changes. Now, we show how to extend the proof of [18], which is limited to CPPT.

We prove the hybrid argument in game hops, starting from the real protocol G_1. In G_2, we replace *one* oracle instance of $\mathcal{O}_0$ by $\mathcal{O}_1$ (at a random point). In G_3, every instance of $\mathcal{O}_0$ but one is replaced by $\mathcal{O}_1$. In G_4, only $\mathcal{O}_1$ is used. Since $\mathcal{O}_1$ is weakly efficient relative to $\mathcal{O}_0$ and $\mathcal{O}_0 \overset{c}{\approx} \mathcal{O}_1$, the transitions from G_1 to G_2 (resp. G_3 to G_4) preserve efficiency and are indistinguishable. The step from G_2 to G_3 is the crux. Note that we have at least one $\mathcal{O}_0$ (resp. $\mathcal{O}_1$) instance in either game. Take any one and denote the time spent in that instance by T_0 (resp. T_1). Since we randomized the instances, the distribution of T_0 (resp. T_1) does not depend on the concrete instance. Importantly, even in the hybrid *reduction*, there is an instance which can be used to compute T_0 (resp. T_1). Moreover, the total time spent in computing instances of $\mathcal{O}_0$ and $\mathcal{O}_1$ is "dominated"[14] by $q \cdot T_0 + q \cdot T_1$. Thus, it suffices to prove that $S = T' + T_0 + T_1$ is CEPT, where T' is the time spent outside emulation of instances of $\mathcal{O}_0$ and $\mathcal{O}_1$. (Note that S, T', T_0, T_1 depend on the hybrid H_ℓ, where $\ell \in \{1, \dots, q-1\}$; we suppressed this dependency.) Now, we have two properties:

- S_ℓ is CEPT if and only if $\mathsf{time}(\mathsf{H}_\ell)$ is CEPT for the ℓ-th hybrid H_ℓ.
- The reduction can compute and output S_ℓ.

Thus, it suffices that S_1 and S_{q-1} are indistinguishable, since we know that S_1 is CEPT. Curiously, we now reduced efficiency to indistinguishability.[15] To prove indistinguishability, we can truncate the reduction (or rather, the hybrids) to strict PPT as in the standard reduction. Thus, we obtain $S_1 \overset{c}{\approx} S_q$. The hybrid lemma follows. The actual reasoning of this last step is a bit lengthier, but follows [18] quite closely: We truncate each oracle separately to maintain symmetry of `timeout` probabilities. Unfortunately, the reduction does not give the usual telescoping sum, since the challenge oracle cannot be truncated. Due to symmetry, the error is "dominated" by observed `timeout`s. Hence, it suffices to find a (uniform) bound for the `timeout` probabilities over all H_ℓ. Our reasoning for this is mildly more complex than [18], since we do not have negligible bounds for `timeout`s, but only polynomial tail bounds, and we make a weaker assumption on efficiency of $\mathcal{O}_0$ and $\mathcal{O}_1$.

Modular sequential composition. With Lemma 4 at hand, it is straightforward to prove that auxiliary input zero-knowledge composes sequentially. In fact, the well-known proof works almost without modifications by using the hybrid lemma (Lemma 4), which absorbs the bulk of the complexity. Indeed, it is possible to prove a modular sequential composition theorem for secure function evaluation, similar to [19]. Interestingly, in [19], subprotocols must have simulators which are EPT *in any interaction*, whereas in our setting, there is no such restriction.

[14] To be exact, dominated with slack q: $\mathbb{P}(\mathsf{time}_{\mathcal{O}_0+\mathcal{O}_1}(\mathsf{H}_\ell) > t) \leq q \cdot \mathbb{P}(q(T_{\ell,0}+T_{\ell,1}) > t)$.

[15] The CEPT characterization does not strictly apply here, but a simple variation does.

1.5 Related work

We are aware of three (lines of) related works w.r.t. EPT: The results by Katz and Lindell [19] and those of Goldreich [14], both focused on cryptography. And the relaxation of EPT for average-case complexity by Levin [23]. A general difference of our approach is, that we treat the security parameter separate from input sizes, whereas [19, 14] assume $\kappa = |x|$. With respect to a posteriori runtime, [18] is a close analogue, although for PPT and in the UC setting.

Comparison with [19]. Katz and Lindell [19] tackle the problem of expected polynomial time by using a *superpolynomial runtime cutoff*. They show that this cutoff guarantees a (strict) EPT adversary. However, for the superpolynomial cutoff, they need to *fix* one superpolynomial function α and have to assume security of primitives w.r.t. (strict) α-time adversaries. Squinting hard enough, their approach is dual to ours. Instead of assuming superpolynomial security and doing a cutoff, we "ignore negligible events" in runtime statistics, thus doing a "cutoff in the probability space". Moreover, we require no fixed bound.

Interestingly, their first result [19, Theorem 5] holds for "adversaries which are EPT w.r.t. the real protocol". Their notion is minimally weaker than ours, as it requires efficiency of the adversary *for all inputs* instead of a sequence of input distributions.[16] [19, Section 3.5] claims that other scenarios, e.g. sequential composition, fall within [19, Theorem 5]. Their *modular* sequential composition theorem, [19, Theorem 12], however, requires that subprotocol simulators are "expected polynomial time *in any interaction*", which neither Theorem 5 nor Theorem 12 assert for the resulting simulators.

Comparison with [14]. Goldreich [14] strengthens the notion of expected polynomial time to obtain a complexity class which is stand-alone and suitable for rewinding based proofs. He requires *expected polynomial time w.r.t. any reset attack*, hence restricts to "nice" adversaries. With this, normal (in the sense of [14]) black-box simulators run in expected polynomial time, essentially by assumption. This way of dealing with designated adversaries is far from the spirit of our work.

Comparison with [23]. The relaxation of expected polynomial time adopted by Levin [23] and variations [13, 14, 3] are very strong. Let T be a runtime distribution. One definition requires that for some poly and $\gamma > 0$, $\mathbb{P}(T_\kappa > C) \leq \frac{\mathsf{poly}(\kappa)}{C^\gamma}$ for large enough κ and $C \geq 0$. Equivalently, $\mathbb{E}(T_\kappa^\gamma)$ is polynomially bounded (in κ) for some $\gamma > 0$. Allowing negligible "errors" relaxes the notion further. This definition fixes the composition problems of expected polynomial time. But arguably, it stretches what is considered efficient far beyond what one may be willing to accept. Indeed, runtimes whose expectation is "very infinite" are considered efficient.[17] The goals of average case complexity theory and cryptography

[16] Their definitions are a consequence of their non-uniform security definition and complexity setting. The proof of [19, Theorem 5] never changes adversarial inputs, so there is no obstruction to handling designated adversaries in our sense.

[17] Setting $c = 2$ and $\gamma = 3$ in Remark 1 yields a runtime T with $\mathbb{E}(T) = \sum_{n=1}^{\infty} n$, which is still considered efficient.

do not align here. We stress that our approach, while relaxing expected polynomial time, is far from being so generous, see Section 1.6.

Related work on CPPT. The notion of CPPT is (in different forms) used and well-known. For example, Boneh and Shoup [4] rely on such a notion. This sidesteps technical problems, such as sampling uniformly from $\{0, 1, 2\}$ with binary coins. With a focus on complexity theory, Goldreich [12] defines *typical efficiency* similar to CPPT. As the relaxations for strict bounds is very straightforward, we suspect more works using CPPT variations for a variety of reasons.

Comparison with [18]. Hofheinz, Unruh, and Müller-Quade [18] define *PPT with overwhelming probability (w.o.p.)*, i.e. CPPT, and consider a posteriori efficiency. They work in the setting of universal composability (UC), and their main focus is an overall sensible notion of runtime, which does not artificially restrict evidently efficient *functionalities*, such as databases or bulletin boards. Their notion of efficiency is similar to our setting with CPPT. In fact, we use their techniques for the hybrid argument. Since [18] defines and assumes *protocol efficiency*, which we deliberately neglect, there are some differences. Reinterpreting [18], their approach is based on: "If *for all* (stand-alone) efficient $\mathcal{D}$ the machine $\mathcal{D}^{\mathcal{O}_0}$ is efficient, then *for all* (stand-alone) efficient $\mathcal{D}$ the machine $\mathcal{D}^{\mathcal{O}_1}$ is efficient."[18] Our approach is based on: "*For all* $\mathcal{D}$, if the machine $\mathcal{D}^{\mathcal{O}_0}$ is efficient, then the machine $\mathcal{D}^{\mathcal{O}_1}$ is efficient." The stronger (protocol) efficiency requirements are harder to justify in our setting. (Even classical PPT $\mathcal{O}_0$ can be "inefficient" for *expected* poly-size inputs. E.g., disallowing quadratic time protocols seems harsh.)

More related work. Halevi and Micali [17] define a notion of efficiency for proofs of knowledge, which closely resembles our notion of normal rewinding strategies. Precise zero-knowledge [25, 27] requires that simulation and real execution time are closely related. Due to Feige's "attack" (or Example 1), this does not seem to help with designated EPT adversaries.

1.6 Separations

We briefly provide separations between some runtime notions. Here, we focus only on efficiency of adversaries, and *ignore* requirements imposed on protocol efficiency, since we deliberately neglected those. We consider *basic runtime classes* (i.e. runtimes of sampling algorithms) and how they are *lifted to interactive algorithms*.

Both [19, Definition 1] and [18, Definitions 1 and 2] use an "a posteriori" lifting. The former lifts EPT, the latter lifts CPPT; both allow designated adversaries and are similar to our setting. "A priori" liftings, such as [14, Definitions 1–4] are far more restrictive (on adversaries), effectively disallowing designated adversaries.

[18] Think of $\mathcal{D}$ as the environment, $\mathcal{O}_0$ as the protocol, and $\mathcal{O}_1$ as the simulator.

Regarding the underlying runtime classes, the works [19, 14] deal with (perfect) EPT, negligible deviations are not allowed. The notion of PPT w.o.p. from [18] and CPPT coincide. To separate PPT, EPT, CPPT, CEPT, and Levin's relaxations, we first recall fat-tailed distributions.

Remark 1 (Fat-tailed distributions). The sum $\sum_n n^{-c}$ is finite if and only if $c > 1$. Thus, we obtain a random variable X with $\mathbb{P}(X = n) \propto n^{-c}$. For $\gamma > 0$ we have $\mathbb{E}(X^\gamma) \propto \sum_n n^{-c+\gamma}$. If $c - \gamma \leq 1$, then $\mathbb{E}(X^\gamma) = \infty$. Moreover, $\mathbb{P}(X \geq k) \geq k^{-c}$, i.e. X has **fat tails**. In particular, for $c = 3$, $\mathbb{E}(X) < \infty$ but $\mathbb{E}(X^2) \propto \sum_n n^{-1} = \infty$, and $\mathbb{P}(X \geq \mathsf{poly}) \geq \frac{1}{\mathsf{poly}^3}$ for any poly.

Allowing a negligible deviation clearly separates perfect runtime distributions from their computational counterparts. Clearly, PPT is strictly contained in EPT. The separation of CPPT and CEPT follows from fat-tailed distributions. In Section 1.6 below, we separate CEPT from Levin's relaxations of EPT, denoted $\mathcal{LT}$, and Vadhan's relaxation [14] of $\mathcal{LT}$, denoted $\mathcal{VT}$, which allows negligible deviation. In the following diagram, *strict* inclusions are denoted by arrows.

$$\begin{array}{ccccc} \mathcal{PPT} & \longrightarrow & \mathcal{EPT} & \longrightarrow & \mathcal{LT} \\ \downarrow & & \downarrow & & \downarrow \\ \mathcal{CPPT} & \longrightarrow & \mathcal{CEPT} & \longrightarrow & \mathcal{VT} \end{array}$$

Levin's relaxation and CEPT. We noted in Remark 1, that $\sum_{n=1}^{\infty} n^{-c} = \alpha_c < \infty$ for $c > 1$ gives rise to a distribution Z_c over $\mathbb{N}$ via normalizing the sum. Let $X = Z_2^3$. Then $\mathbb{E}(X) = \frac{1}{\alpha_c} \sum_{n=1}^{\infty} n = \infty$. Since Z_2 is fat-tailed, so is X. Let $Y_k = X|_{(\cdot \geq k^3) \mapsto 0}$. It follows immediately that $\mathbb{E}(Y_k) = \mathbb{E}(X|_{(\cdot \geq k^3) \mapsto 0}) \geq \frac{1}{\alpha_c} k^2$ for any $k \in \mathbb{N}$. Thus, for any superpolynomial cutoff K, we find $\mathbb{E}(Y_K) \geq \frac{1}{2\alpha_c} K^2$ is superpolynomial, and as a consequence, there is no superpolynomial cutoff which makes X EPT. (We interpret X as a constant family of runtimes.)

Formally, CEPT uses ν-quantile cutoffs (i.e. we may condition on an event $\mathcal{G}$ of overwhelming probability $1-\nu$ that minimizes $\mathbb{E}(T|\mathcal{G})$). For X, any ν-quantile cutoff for negligible ν induces some bound k which maximizes $\mathbb{P}(T \leq k) \geq \nu$. If k were polynomial, then (due to "fat tails") ν must also be polynomial. Hence, k must be superpolynomial, and consequently there is no negligible quantile cutoff which makes X EPT. All in all, the runtime distribution X is allowed by Levin's relaxation, but is not CEPT.

1.7 Structure of the paper

In the introduction, we discuss motivation, contribution and related work, and sketch our definitions and techniques.. In Section 2, we clarify notation, recall and adapt standard definitions, and give basic requirements for runtime. In Section 3, we define virtually expected time, the "triple-oracle distinguishing" notion, and

CEPT. We also state the characterization of CEPT and provide a proof sketch. In Section 4, we define zero-knowledge protocols and designated adversaries. We then prove, in full detail, that the naive simulator for $\mathrm{G3C}_{\mathrm{GK}}$ works, and show by example how benign simulators look like. Lastly, in Section 5, we discuss the hybrid lemma and sequential composition.

Due to limited space, many of the definitions, tools, and results in the introduction are only sketched or missing. For these, we refer to the full version [21].

2 Preliminaries

In this section, we state some basic definitions and (non-)standard conventions. Since machine models more influence in an EPT setting than in a strict PPT setting, we fix some suitable RAM model for the rest of this work.

Notation and basic definitions. We denote the security parameter by κ; it is often suppressed. Similarly, we often speak of an object X, instead of a family of objects $(X_\kappa)_\kappa$ parameterized by κ. We always assume binary encoding of data, unless explicitly specified otherwise. We write $X \sim Y$ if a random variable X is distributed as Y. For random variables X, Y over a (partially) ordered set $(A, \leq)$ we write $X \overset{d}{\leq} Y$ if $\mathbb{P}(X > a) \leq \mathbb{P}(Y < a)$ for all $a \in A$ and say Y *dominates* X (or is greater than X in distribution). We use the same notation for families of random variables, i.e. we write $X \leq Y$ and mean $X_\kappa \overset{d}{\leq} Y_\kappa$ for all κ. We write $X|_{a \mapsto b}$ (resp. $X|_{S \mapsto b}$, resp. $X|_{\mathsf{pred} \mapsto b}$) for the random variable where a (resp. any a satisfying $a \in S$ resp. $\mathsf{pred}(a) = 1$) is mapped to b, and everything else unchanged, e.g. $X|_{\perp \mapsto 0}$ or $X|_{S \mapsto 0}$ or $X|_{\geq N \mapsto N}$. The **statistical distance** $\Delta(\rho, \sigma)$ of distributions (i.e. measures) ρ, σ over a countable set Ω is $\frac{1}{2} \sum_{\omega \in \Omega} |\rho(\omega) - \sigma(\omega)|$. With poly, $\mathsf{polylog}$, and negl we denote polynomial, polylogarithmic and negligible functions (in κ) respectively. Usually, we (implicitly) assume that poly, $\mathsf{polylog}$, and negl are *monontone*. A function negl is negligible if $\lim_{\kappa \to \infty} \mathsf{poly}(\kappa)\mathsf{negl}(\kappa) = 0$ for every polynomial poly. In many definitions, we assume the existence of a negligible bound negl on some advantage $\varepsilon = \varepsilon(\kappa)$. We use "strict pointwise $\leq$" for bounds, i.e. $\varepsilon \leq \mathsf{negl}$ denotes $\forall \kappa \colon \varepsilon(\kappa) \leq \mathsf{negl}(\kappa)$.

Systems, algorithms, interaction and machine models. We always consider (induced) systems, which offer **interfaces** for (message-based) communication.[19] Input and output are modelled as interfaces as well. The security parameter κ is an implicit input interface of (almost) every system. A system is **closed** if its only open interfaces are input for κ and output, i.e. if it is a "sampling algorithm" which on inputs κ samples some output. Besides (black-box) oracle-access, an algorithm A can have timed access to an oracle $\mathcal{O}$, which means $\mathsf{A}^{\mathcal{O}}$ can limit the allotted time s of a call to $\mathcal{O}$ and is informed of the elapsed runtime t when $\mathcal{O}$ returns. We let $t = \texttt{timeout}$ if $\mathcal{O}$ did not return in the allotted timespan s, i.e. if $t > s$. We write $\mathsf{bbrw}(\mathsf{A})$ for black-box rewinding access to A.

[19] We use an ad-hoc definition of system. A compatible, precise notion was recently (concurrently) introduced in [22].

Preliminary remarks on runtime. For an oracle algorithm A, we write $\mathsf{time}_{\mathsf{A}}(\mathsf{A}^{\mathcal{O}})$ for the time spent in A, $\mathsf{time}_{\mathcal{O}}(\mathsf{A}^{\mathcal{O}})$ for the time spent in $\mathcal{O}$, and $\mathsf{time}_{\mathsf{A}+\mathcal{O}}(\mathsf{A}^{\mathcal{O}})$ for the time spent in both. This notation extends naturally to systems built from interacting machines. Note that $T = \mathsf{time}_{\mathsf{A}}(\mathsf{A}^{\mathcal{O}})$ is a *random variable*. We assume that an oracle call is a single step and that runtimes sum up, i.e. $\mathsf{time}_{\mathsf{A}}(\mathsf{A}^{\mathcal{O}}) + \mathsf{time}_{\mathcal{O}}(\mathsf{A}^{\mathcal{O}}) = \mathsf{time}_{\mathsf{A}+\mathcal{O}}(\mathsf{A}^{\mathcal{O}})$, *as dependent random variables.*

Definition 2. *A **runtime (distribution)** T is a family of random variables (resp. distributions) over $\mathbb{N}_0$ parameterized by the security parameter κ. We (only) view a runtime as a random variable $T_\kappa \colon \Omega_\kappa \to \mathbb{N}_0$, when stochastic dependency is relevant. A **runtime class** $\mathcal{T}$ is a set of runtime distributions. A (sampling) algorithm A is $\mathcal{T}$**-time** if $\mathsf{time}_{\mathsf{A}}(\mathsf{A}) \in \mathcal{T}$, more explicitly, $T_\kappa = \mathsf{time}_{\mathsf{A}}(\mathsf{A}(\kappa))$ is in $\mathcal{T}$.*

Example 3. The runtime classes $\mathcal{PPT}$ and $\mathcal{EPT}$ of strict polynomial time (PPT) and expected polynomial time (EPT) are defined in the obvious way, i.e.: $T \in \mathcal{PPT}$ (resp. $T \in \mathcal{EPT}$) if there exists a polynomial poly such that $\mathbb{P}(T_\kappa > \mathsf{poly}(\kappa)) = 0$ (resp. $\mathbb{E}(T_\kappa) \leq \mathsf{poly}(\kappa)$).

In any closed system, every component has an associated random variable, describing the time spent in it. We only consider such runtimes (most often, the total runtime). Hence, efficiency depends only on κ, since closed systems have no (other) input. In particular, we do not assign a stand-alone runtime notion to a non-closed system, e.g. an algorithm A which needs inputs (besides κ), resp. oracle access, resp. communication partners. The exception to the rule are *a priori PPT resp. EPT* algorithms A, for which there is a bound poly such that $\mathsf{time}_{\mathsf{A}}(\dots) \leq \mathsf{poly}$ resp. $\mathbb{E}(\mathsf{time}_{\mathsf{A}}(\dots)) \leq \mathsf{poly}$ for any choice of inputs, resp. oracles, resp. parties.

Our central tool for dealing with expected time is truncation.

Definition 3 (Runtime truncation). *Let A be an algorithm. We define $\mathsf{A}^{\leq N}$ as the algorithm which executes A up to N steps, and then returns A's output. If A did not finish in time, $\mathsf{A}^{\leq N}$ returns* `timeout`.

Probability theory. The underlying probability space is usually denoted by Ω. We allow product extension of Ω to suit our needs, say extending to $\Omega' = \Omega \times \Sigma$ with Bernoulli distribution $\mathsf{Ber}(\frac{1}{3})$ on $\Sigma = \{0, 1\}$. Random variables over Ω are lifted implicitly and we again write Ω instead of Ω'. Let $\mathbb{N}_0 \cup \{\infty, \mathtt{timeout}\}$ be totally ordered via $n < \infty < \mathtt{timeout}$ for all $n \in \mathbb{N}_0$.

Definition 4 (ν-quantile cutoff). *Let T be a distribution on $\mathbb{N}_0 \cup \{\infty\}$ and $\nu > 0$. Suppose that $\mathbb{P}(T = \infty) \leq \nu$. The (exact) ν**-quantile (cutoff)** T^ν is following distribution on $\mathbb{N}_0 \cup \mathtt{timeout}$. Let $\mathsf{CDF}_T(\,\cdot\,) \colon \mathbb{N}_0 \cup \{\infty\} \to [0, 1]$ be the CDF of T. Then $\mathsf{CDF}_{T^\nu}(\,\cdot\,) \colon \mathbb{N}_0 \cup \mathtt{timeout} \to [0, 1]$ is defined by $\mathsf{CDF}_{T^\nu}(n) = \min\{1-\nu, \mathsf{CDF}_T(n)\}$ for $n \in \mathbb{N}$, and $\mathsf{CDF}_{T^\nu}(\infty) = \lim_{n\to\infty} \min\{1-\nu, \mathsf{CDF}_T(n)\}$, hence $\mathbb{P}(T^\nu = \infty) = 0$, and $\mathsf{CDF}_{T^\nu}(\mathtt{timeout}) = 1$,*

It is easy to see that, perhaps after extending Ω, there always is an event $A \subseteq \Omega$ such that $T^\nu \coloneqq T|_{A \mapsto \mathtt{timeout}}$ is a ν-quantile cutoff of T.

Remark 2 (Equal-unless). If $X, Y : \Omega \to \mathcal{S}$ are random variables over Ω and coincide (as functions), except for an event $\mathcal{E} \subseteq \Omega$, then X and Y are **(pointwise) equal unless** $\mathcal{E}$. We extend this to oracles (and algorithms) in the natural way.

Definition 5 ((Oracle-)Indistinguishability). *Let $\mathcal{O}_0$ and $\mathcal{O}_1$ be (not necessarily computable) oracles with identical interfaces. A distinguisher $\mathcal{D}$ is a system which connects to all interfaces or $\mathcal{O}_0$, $\mathcal{O}_1$, resulting in a closed system $\mathcal{D}^{\mathcal{O}_b}$. The **(standard) distinguishing advantage** of $\mathcal{D}$ is defined by*

$$\mathrm{Adv}^{dist}_{\mathcal{D},\mathcal{O}_0,\mathcal{O}_1}(\kappa) = |\mathbb{P}(\mathcal{D}^{\mathcal{O}_1(\kappa)}(\kappa) = 1) - \mathbb{P}(\mathcal{D}^{\mathcal{O}_0(\kappa)}(\kappa) = 1)|.$$

By abuse of notation, we sometimes abbreviate $\mathrm{Adv}^{dist}_{\mathcal{D},\mathcal{O}_0,\mathcal{O}_1}$ by $\mathrm{Adv}^{dist}_{\mathcal{D},\mathcal{O}}$.

*Let $\mathcal{T}$ be a runtime class. Then $\mathcal{O}_0$ and $\mathcal{O}_1$ are **computationally (standard) indistinguishable in $\mathcal{T}$-time**, written $\mathcal{O}_0 \overset{c}{\approx}_{\mathcal{T}} \mathcal{O}_1$ if for any $\mathcal{T}$-time distinguisher $\mathcal{D}$, i.e. $\mathsf{time}_{\mathcal{D}}(\mathcal{D}^{\mathcal{O}_b(\kappa)}(\kappa)) \in \mathcal{T}$ (for $b = 0, 1$), there is some negligible negl such that $\mathrm{Adv}^{dist}_{\mathcal{D},\mathcal{O}}(\kappa) \leq \mathsf{negl}$. We define* statistical $\mathcal{T}$*-query indistinguishability by counting only oracle-queries as runtime. If all (unbounded) distinguishers have advantage* 0*, we speak of* perfect *indistinguishability and write $\mathcal{O}_0 \equiv \mathcal{O}_1$.*

Indistinguishability of distributions X and Y [under repeated samples] is defined in the natural compatible way, namely via oracles $\mathcal{O}_X$ and $\mathcal{O}_Y$ which outputs a single [a fresh] sample of X resp. Y [for each query]. By truncation arguments, if a statistical $\mathcal{EPT}$-query distinguisher exists, so does a statistical $\mathcal{PPT}$-query distinguisher, i.e. a strict polynomial number of queries suffice.

3 Computationally expected polynomial time

In this section, we define computationally expected polynomial time (CEPT).

Virtually expected time. We are interested in properties, which need only hold with overwhelming probability. We formalize this for the expectation of non-negative random variables as follows.

Definition 6 (Virtual expectation). *Let $X : \Omega \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ Let $\varepsilon > 0$. We say X has ε**-virtual expectation (bounded by)** t if*

$$\exists \mathcal{G} \subseteq \Omega\colon\ \mathbb{P}(\mathcal{G}) \geq 1 - \varepsilon \ \wedge\ \mathbb{E}(X \mid \mathcal{G}) \leq t$$

*We extend this to families by requiring it to hold component-wise. Moreover, we say a runtime T is ε**-virtually** t**-time** if T has ε-virtual expectation bounded by t. We abbreviate this as **virtually expected** (t, ε)**-time** and call ε the **virtuality** of time (t, ε). If we do not specify ε, it is a negligible function.*

Virtual expectation has a "probably approximately" flavour. It is closely related to "ε-smooth properties", such as ε-smooth min-entropy. Virtual expectation behaves well under restriction (up to a certain extent):

Let $X : \Omega \to \mathbb{R}_{\geq 0}$ be a random variable and $\mathbb{E}(X) = t$. Then any restriction of X to an event $\mathcal{G}$ of measure $1 - \varepsilon$ implies $\mathbb{E}(X \mid \mathcal{G}) \leq (1 - \varepsilon)^{-1} t$. The upshot is that, as long as we condition on *overwhelming* (in fact, noticeable) events $\mathcal{G}$, polynomially bounded expectation is preserved.

Triple-oracle indistinguishability. To prevent technical artefacts in definitions of runtime classes and distinguishing-closedness, we use triple-oracle indistinguishability. Triple-oracle distinguishing should be interpreted as distinguishing with repeated samples, plus sampling access to the distributions X_0, X_1. Recall that we always use *binary* encodings, and this includes runtime oracles (even though unary encodings work there without change).

Definition 7. *A **triple-oracle distinguisher** $\mathcal{D}$ for distributions X_0, X_1, receives access to three oracles $\mathcal{O}_0$, $\mathcal{O}_1$ and $\mathcal{O}_b^*$, which sample according to some distributions X_0, X_1, and X_b. The distinguishing advantage is* $\mathrm{Adv}^{\textit{3-dist}}_{\mathcal{D},\mathcal{O}_0,\mathcal{O}_1} = |\mathbb{P}(\mathcal{D}^{\mathcal{O}_0,\mathcal{O}_1,\mathcal{O}_1^*}(\kappa)=1) - \mathbb{P}(\mathcal{D}^{\mathcal{O}_0,\mathcal{O}_1,\mathcal{O}_0^*}(\kappa)=1)|$.

*Two runtime distributions T, S are **computationally $\mathcal{T}$-time triple-oracle indistinguishable**, if any $\mathcal{T}$-time distinguisher has advantage $o(1)$. If $\mathcal{T}$ contains $\mathcal{PPT}$, then (by amplification) any distinguisher has negligible advantage. For statistical triple-oracle indistinguishability, we only count oracle queries as a step (and often explicitly speak of statistical $\mathcal{T}$-query distinguishers).*[20]

*A runtime class $\mathcal{T}$ is computationally **closed** if for any runtime S: If some $T \in \mathcal{T}$ is triple-oracle indistinguishable from S, then $S \in \mathcal{T}$.*

In the definition, we sketched our approach for general runtime classes (namely requiring $o(1)$ advantage bound). This definition applies to runtime classes from other algebras, such as polylog or quasi-polynomial time, and implicitly uses the notion of negligible function for these algebras. From now on, we specialize to the polynomial setting, where amplification enforces (poly-)negligible advantage.

Characterizing CEPT. We begin with the fundamental definitions.

Definition 8 (CEPT and CPPT). *The runtime class $\mathcal{CEPT}$ of **computationally expected polynomial time** contains all runtimes which are (triple-oracle) $\mathcal{PPT}$-time indistinguishable from expected polynomial time: In other words: A runtime T is CEPT if there is an EPT $\widetilde{T}$, such that T and $\widetilde{T}$ are triple-oracle PPT-indistinguishable.*

*The runtime class $\mathcal{CPPT}$ of **computationally (strict) probabilistic polynomial time** is defined analogously.*

Now, we turn towards the characterization of CEPT. We start with a few simple lemmata. Their central technique is to approximate probability distributions with suitable precision, and then use this information for distinguishing.

Lemma 5. *Suppose S and T are runtimes and $T \in \mathcal{CEPT}$. Then statistical $\mathcal{CEPT}$-query and computational $\mathcal{CEPT}$-time triple-oracle indistinguishability coincide. Moreover, a priori PPT distinguishers are sufficient.*

[20] We *never* consider unbounded queries for statistical triple-oracle distinguishing, as this trivially coincides with perfect indistinguishability.

Proof (Proof sketch). It is clear that statistical indistinguishability implies computational indistinguishability. The proof is quite simple, and based on standard truncation arguments. For $T \in \mathcal{CEPT}$ there exists, by definition, some $\widetilde{T} \in \mathcal{EPT}$ such that T and $\widetilde{T}$ are computationally triple-oracle indistinguishable. Hence, for any efficiently computable $N = N(\kappa)$, we have $|\mathbb{P}(T > N) - \mathbb{P}(\widetilde{T} > N)| \leq \mathsf{negl}$.

To show that T and $\widetilde{T}$ are *statistically* triple-oracle indistinguishable, we argue by contraposition and assume the statistical distance $\Delta(T, \widetilde{T})$ is at least $\delta = \frac{1}{\mathsf{poly}_0}$ infinitely often. Note that $\mathbb{P}(\widetilde{T} > N) \leq \frac{\mathsf{poly}_1}{N}$, where $\mathbb{E}(\widetilde{T}) \leq \mathsf{poly}_1$. Thus, by truncating T, $\widetilde{T}$ after, say $N = 4\mathsf{poly}_0\mathsf{poly}_1$, we know that $T^{\leq N}$ and $\widetilde{T}^{\leq N}$ are distributions with *polynomial support in* $\{0, \ldots, N\}$ and *non-negligible statistical distance* $\frac{\delta}{4}$ infinitely often. Since we have (repeated) sample access to T, $\widetilde{T}$ and the challenge runtime, we can approximate the probability distributions up to any $\frac{1}{\mathsf{poly}}$ precision in polynomial time. Consequently, we can construct a (computational) PPT distinguisher if T and $\widetilde{T}$ are not statistically $\mathcal{PPT}$-query indistinguishable. A similar line of reasoning show that T and S are computationally distinguishable if they are statistically far.

Lemma 6. *Let T and S be runtimes induced by algorithms A, B, and suppose $T \in \mathcal{CEPT}$. Then triple-oracle and standard $\mathcal{PPT}$-time indistinguishability coincide.*

Proof (Proof sketch). As in Lemma 6, after suitable truncation we can assume that T and S, and hence A and B, are actually PPT. By sampling T resp. S via emulation of A resp. B (instead of oracle queries), and a hybrid argument (over challenge queries) the claim follows.

We stress that to efficiently distinguish two induced runtimes, it is sufficient that *one* of the two algorithms is efficient.[21] Putting things together yields following characterization of CEPT and CPPT:

Corollary 1 (Characterization of CEPT). *Let T be a runtime. The following conditions are equivalent:*

(0) T is in $\mathcal{CEPT}$.
(1) T is $\mathcal{PPT}$-time triple-oracle comp. indist. from some $\widetilde{T} \in \mathcal{EPT}$.
(2) T is $\mathcal{PPT}$-query triple-oracle stat. indist. from some $\widetilde{T} \in \mathcal{EPT}$.
(3) T is virtually expected polynomial time. Explicitly: There is a negligible function negl, an event $\mathcal{G}$ with $\mathbb{P}(\mathcal{G}) \geq 1 - \mathsf{negl}$, and a polynomial poly, such that $\mathbb{E}(T_\kappa \mid \mathcal{G}) \leq \mathsf{poly}(\kappa)$.

Furthermore, $T \in \mathcal{CEPT}$ satisfies the tail bound $\mathbb{P}(T_\kappa > N) \leq \frac{\mathsf{poly}(\kappa)}{N} + \mathsf{negl}(\kappa)$ for poly and negl as in (3). Consequently, $\mathcal{CEPT}$ is a closed runtime class. For induced runtimes $T = \mathsf{time}_{\mathsf{A}}(\mathsf{A})$, $S = \mathsf{time}_{\mathsf{B}}(\mathsf{B})$, where $T \in \mathcal{CEPT}$, and

[21] If neither runtime is efficient, we are in a setting where the truncation argument does not work. Indeed, strings can be encoded as numbers, hence runtimes. Thus, this is indistinguishability of general distributions.

S is arbitrary, triple-oracle indistinguishability and standard indistinguishability coincide. The analogous characterization and properties hold for CPPT.

Proof (Proof sketch of Corollary 1). Equivalence of items (1) and (2) follows from Lemma 5. Now, we show (2) implies (3). As in Lemma 5, we see that triple-oracle indistinguishability implies statistical closeness. By assumption, there is some $\widetilde{T} \in \mathcal{EPT}$ with $\Delta(T, \widetilde{T}) \leq \nu$ negligible. Let $S = T^{\nu}$ be the ν-quantile of T, note that $S \leq \mathbb{E}(\widetilde{T})$, and let $\mathcal{G}$ be an event associated with the quantile, (which exists, perhaps after extension of Ω) that is, $S = T|_{\Omega \setminus \mathcal{G} \mapsto \texttt{timeout}}$. Then we have $\mathbb{E}(S) \leq \mathbb{E}(\widetilde{T}) \leq \mathsf{poly}$, and item (3) easily follows. The converse is trivial.

To see the tail bound, note that for $T \in \mathcal{CEPT}$ there is a "good" runtime $\widetilde{T} \in \mathcal{EPT}$ with $\Delta(T, \widetilde{T}) \leq \mathsf{negl}$. Thus, the tail bound follows immediately from Markov's bound applied to $\widetilde{T}$ and statistical distance of negl. Hence, CEPT distinguishers are as powerful as PPT distinguishers. Thus, $\mathcal{CEPT}$ is closed

Finally, for induced runtimes, Lemma 6 demonstrates the equivalence of triple-oracle and standard distinguishing.

Proof (Proof sketch of Corollary 1). Equivalence of items (1) and (2) follows from Lemma 5. For (2) $\implies$ (3) note that since T is statistically triple-oracle indistinguishable from $\widetilde{T}$ and $\widetilde{T} \in \mathcal{EPT}$, we have that $\Delta(()T, \widetilde{T}) = \nu$ is negligible. The converse is trivial.

To see the tail-bound, note that for $T \in \mathcal{CEPT}$ there is a "good" runtime $\widetilde{T} \in \mathcal{EPT}$ with $\Delta(T, \widetilde{T}) \leq \mathsf{negl}$. Thus, the tail bound follows immediately from Markov's bound applied to $\widetilde{T}$ and statistical distance of negl. Hence, CEPT distinguishers are as powerful as PPT distinguisher. Thus $\mathcal{CEPT}$ is closed.

Lemma 6 shows the equivalence of triple-oracle and standard distinguishing.

Remark 3 (Non-uniformity). As noted in the introduction, non-uniform advice can replace sampling access. For non-uniform distinguishers, triple-oracle and standard indistinguishability coincide. This simplifies Corollary 1.

4 Application to zero-knowledge arguments

Our flavour of zero-knowledge follows Goldreich's treatment of uniform complexity [11], combined with Feige's designated adversaries [9]. We only define efficient proof systems for NP-languages.

Definition 9 (Interactive arguments). *Let $\mathcal{R}$ be an NP-relation with corresponding language $\mathcal{L}$. An **argument (system) for** $\mathcal{L}$ consists of two interactive algorithms $(\mathcal{P}, \mathcal{V})$ such that:*

Efficiency: *There is a polynomial* poly *so that for all (κ, x, w) the runtime* $\mathsf{time}_{\mathcal{P}+\mathcal{V}}(\langle \mathcal{P}(x, w), \mathcal{V}(x) \rangle)$ *is bounded by* $\mathsf{poly}(\kappa, |x|)$.
Completeness: $\forall (x, w) \in \mathcal{R} : \mathsf{out}_{\mathcal{V}} \langle \mathcal{P}(x, w), \mathcal{V}(x) \rangle = 1$.

Definition 9 essentially assumes "classic" PPT algorithms, but it will be evident that our techniques do not require this. We do not define soundness, but note that it is easily handled via truncation to a PPT adversary. The terms proof and argument systems are often used interchangeably (and we also do this). Strictly speaking, *proof* systems require unconditional soundness and allow unbounded provers. Argument systems allow computational soundness and require efficient provers. All our exemplary proof systems [16, 15, 24, 29, 20, 28] have efficient provers, hence are also argument systems.

4.1 Zero-knowledge

Definition 10. *Let $\mathcal{T}, \mathcal{S} \in \{\mathcal{PPT}, \mathcal{CPPT}, \mathcal{EPT}, \mathcal{CEPT}\}$. Let $(\mathcal{P}, \mathcal{V})$ be an argument system. A* ***universal simulator*** Sim *takes as input $(\mathsf{code}(\mathcal{V}^*), x, aux)$ and simulates $\mathcal{V}^*$'s output. Let $(\mathcal{I}, \mathcal{V}^*, \mathcal{D})$ be an adversary. We define the real and ideal executions as*

$$\mathsf{Real}_{\mathcal{I},\mathcal{V}^*}(\kappa) \coloneqq (state, \mathsf{out}_{\mathcal{V}^*}\langle \mathcal{P}(x,w), \mathcal{V}^*(x, aux)\rangle)$$
$$\text{and} \quad \mathsf{Ideal}_{\mathcal{I},\mathsf{Sim}(\mathsf{code}(\mathcal{V}^*))}(\kappa) \coloneqq (state, \mathsf{Sim}(\mathsf{code}(\mathcal{V}^*), x, aux))$$

where $(x, w, aux, state) \leftarrow \mathcal{I}$ and $(x, w) \in \mathcal{R}$, else Real *and* Ideal *return a failure symbol, say $\perp$. We omit the input $\mathsf{code}(\mathcal{V}^*)$ to* Sim*, if it is clear from the context. The advantage of $(\mathcal{I}, \mathcal{V}^*, \mathcal{D})$ is*

$$\mathrm{Adv}^{zk}_{\mathcal{I},\mathcal{V}^*,\mathcal{D}}(\kappa) \coloneqq |\mathbb{P}(\mathcal{D}(\mathsf{Real}_{\mathcal{I},\mathcal{V}^*}(\kappa)) = 1) - \mathbb{P}(\mathcal{D}(\mathsf{Ideal}_{\mathcal{I},\mathsf{Sim}}(\kappa)) = 1)|.$$

An adversary $(\mathcal{I}, \mathcal{V}^*, \mathcal{D})$ is $\mathcal{T}$-time if $\mathsf{time}_{\mathcal{I}+\mathcal{P}+\mathcal{V}^*+\mathcal{D}}(\mathcal{D}(\mathsf{Real}_{\mathcal{I},\mathcal{V}^*})) \in \mathcal{T}$.

The argument is ***(uniform) (auxiliary input) zero-knowledge*** *against $\mathcal{T}$-time adversaries w.r.t. $\mathcal{S}$-time* Sim*, if for any $\mathcal{T}$-time adversary $(\mathcal{I}, \mathcal{V}^*, \mathcal{D})$:*

- $\mathsf{time}_{\mathcal{I}+\mathsf{Sim}+\mathcal{D}}(\mathcal{D}(\mathsf{Ideal}_{\mathcal{I},\mathsf{Sim}})) \in \mathcal{S}$*. The runtime of* Sim *includes whatever time is spent to emulate $\mathcal{V}^*$. In a (generalized) sense,* Sim *is weakly $(\mathcal{T}, \mathcal{S})$-efficient relative to $\mathcal{P}$.*
- $\mathrm{Adv}^{zk}_{\mathcal{I},\mathcal{V}^*,\mathcal{D}}(\kappa)$ *is negligible*

Some more remarks are in order.

Remark 4. In our setting, *existential* and *universal* simulation are equivalent. The adversary $\mathcal{V}_{\mathrm{univ}}$, which executes *aux* as its code, is universal.

Remark 5 (Reductions to PPT). By a standard reduction to PPT, we may w.l.o.g. assume that $\mathcal{D}$ is a priori PPT. Perhaps surprisingly, this is false for $\mathcal{I}$. Intuitively, verifying the *quality* of the output of Sim requires only PPT $\mathcal{D}$ (and $\mathcal{I}$). Verifying the *efficiency*, however, does not. The cause is that $\mathcal{I}$ may generate *expected* poly-size inputs.

Remark 6 (Efficiency of the simulation). Definition 10 only ensures that Sim is *weakly* efficient relative to $\mathcal{P}$, i.e. we have no tightness bounds. Relative efficiency

with tightness bounds is an unconditional property, and hence not possible if zero-knowledge holds only computationally.

In the definition, it is possible to replace $\mathsf{time}_{\mathcal{I}+\mathsf{Sim}+\mathcal{D}}(\mathcal{D}(\mathsf{Ideal}_{\mathcal{I},\mathsf{Sim}})) \in \mathcal{S}$ with $\mathsf{time}_{\mathsf{Sim}}(\mathcal{D}(\mathsf{Ideal}_{\mathcal{I},\mathsf{Sim}})) \in \mathcal{S}$, since $\mathcal{I}$ is unaffected, and $\mathcal{D}$ is w.l.o.g. a priori PPT.

Remark 7 ("Environmental" distinguishing: Why $\mathcal{I}$ outputs state). In Definition 10, we allow $\mathcal{I}$ to output *state*, effectively making $(\mathcal{I}, \mathcal{D})$ into a stateful distinguishing "environment". Viewing Sim and $\mathcal{P}$ as oracles, this corresponds to oracle indistinguishability. Without this, the security does not obviously help when used as a subprotocol, since a protocol is effectively a (stateful) distinguishing environment. Definition 10 is discussed in-depth in [21]. Here, we only note that in the *non-uniform* classical PPT setting, it coincides with the standard definition.

Remark 8. We seldom mention non-uniform zero-knowledge formulations in the rest of this work. Our definitions, constructions and proofs make timed $\mathsf{bb\text{-}rw}$ use of the adversary, and therefore apply in the non-uniform setting without change.

4.2 Application to graph 3-colouring

To exemplify the setting, the technical challenges, and our techniques, we use the constant-round zero-knowledge proof of Goldreich and Kahan [15] as a worked example. We only prove zero-knowledge, as completeness and soundness are unconditional. Formal definitions of commitment schemes are in the full version [21]. We assume *left-or-right (LR) oracles* in the hiding experiment for commitment schemes. (Security against CEPT adversaries follows from security against PPT adversaries by a simple truncation argument.)

The protocol. We recall $\mathrm{G3C}_{\mathrm{GK}}$ from Section 1.2. It requires two non-interactive commitments schemes; $\mathsf{Com}^{(\mathrm{H})}$ is perfectly hiding, $\mathsf{Com}^{(\mathrm{B})}$ is perfectly binding. The common input is $G = (V, E)$ and the prover's witness is a 3-colouring ψ.

(P0) $\mathcal{P}$ sends $\mathrm{ck}_{\mathrm{hide}} \leftarrow \mathsf{Gen}^{(\mathrm{H})}(\kappa)$. ($\mathrm{ck}_{\mathrm{bind}} \leftarrow \mathsf{Gen}^{(\mathrm{B})}(\kappa)$ is deterministic.)

(V0) $\mathcal{V}$ randomly picks challenge edges $e_i \leftarrow E$ for $i = 1, \ldots, N = \kappa \cdot \mathsf{card}(E)$, commits to them as $c_i^e = \mathsf{Com}^{(\mathrm{H})}(\mathrm{ck}_{\mathrm{hide}}, e_i)$, and sends all $\{c_i^e\}_{i=1,\ldots,N}$.

(P1) $\mathcal{P}$ picks randomized colourings ψ_i for all $i = 1, \ldots, N$ and commits to all node colours for all graphs in (sets of) commitments $\{\{c_{i,j}^{\psi}\}_{j \in V}\}_{i=1,\ldots,N}$ using $\mathsf{Com}^{(\mathrm{B})}$. $\mathcal{P}$ sends all $c_{i,j}^{\psi}$ to $\mathcal{V}$.

(V1) $\mathcal{V}$ opens the commitments c_i^e to e_i for all $i = 1, \ldots, N$.

(P2) $\mathcal{P}$ aborts if any opening is invalid or $e_i \notin E$ for some i. Otherwise, for all iterations $i = 1, \ldots, N$, $\mathcal{P}$ opens the commitments $c_{i,a}^{\psi}$, $c_{i,b}^{\psi}$ for the colours of the nodes of edge $e_i = (a, b)$ in repetition i.

(V2) $\mathcal{V}$ aborts iff any opening is invalid, any edge not correctly coloured, or if $\mathrm{ck}_{\mathrm{hide}}$ is bad. Otherwise, $\mathcal{V}$ accepts.

In [15], delaying the check of $\mathrm{ck}_{\mathrm{hide}}$ to the end of the protocol weakens the requirements on VfyCK, as the verifier may learn setup randomness of $\mathrm{ck}_{\mathrm{hide}}$ at that point. But this is irrelevant for zero-knowledge.

Proof of zero-knowledge. Our goal is to show the following lemma.

Lemma 7. *Suppose* $\mathsf{Com}^{(\mathrm{H})}$ *and* $\mathsf{Com}^{(\mathrm{B})}$ *are a priori PPT algorithms. Then protocol* $\mathrm{G3C}_{\mathrm{GK}}$ *in Section 4.2 is zero-knowledge against CEPT adversaries with a* bb-rw *CEPT simulator. Let* $(\mathcal{I}, \mathcal{V}^*)$ *be a CEPT adversary and suppose* $T := \mathsf{time}_{\mathcal{P}+\mathcal{V}^*}(\mathsf{Real}_{\mathcal{I},\mathcal{V}^*})$ *is* (t, ε)*-time. Then* Sim *handles* $(\mathcal{I}, \mathcal{V}^*)$ *in virtually expected time* $(t', 2\varepsilon + \varepsilon')$*. Here* ε' *stems from an advantage against the hiding property of* $\mathsf{Com}^{(\mathrm{B})}$*, hence* ε' *negligible. If the time to compute a commitment depends only on the message length, then* t' *is roughly* $2t$*.*

Our proof differs from that in [15] on two accounts: First, we do not use the runtime normalization procedure in [15]. This is because a negligible deviation from EPT is absorbed into the CEPT virtuality, namely ε'. Second, we handle designated CEPT adversaries. In particular, the runtime classes of simulator and adversary coincide. We first prove the result for *perfect* EPT adversaries.

Lemma 8. *The claims in Lemma 7 hold if* $T \in \mathcal{EPT}$*, i.e.* $\varepsilon = 0$*.*

Proof (Proof sketch). We proceed in game hops. The initial game being $\mathsf{Real}_{\mathcal{I},\mathcal{V}^*}$ and the final game being $\mathsf{Ideal}_{\mathcal{I},\mathsf{Sim}}$. We consider (timed) bb-rw simulation.

Game G_0 is the real protocol. The output is the verifier's output and *state* (from $\mathcal{I}$). From now on, we ignore the *state* output, since no game hop affects it.

Game G_1**:** If the verifier opens the commitments in (V1) correctly, the game repeatedly rewinds it to (P1) using fresh prover randomness, until it obtains a second run where $\mathcal{V}^*$ unveils the commitments correctly (in (V1)). The output is $\mathcal{V}^*$'s output at the end of this second successful run. If the verifier failed in the first run, the protocol proceeds as usual. The outputs of G_1 and G_0 are identically distributed. It can be shown that this modification preserves (perfect) EPT of the overall game, i.e. G_1 is perfect EPT. More precisely, the (virtually) expected time is about $2t$ (plus emulation overhead). To see this, use that each iteration executes $\mathcal{P}$'s code with fresh randomness. For the analysis, condition to fix the randomness of everything but $\mathcal{P}$; averaging over the randomness of $\mathcal{I}$, $\mathcal{V}^*$ (and $\mathcal{D}$), then extends the reasoning again. Since bb-rw-access fixes the randomness of $\mathcal{V}^*$ between rewinds, the probability that $\mathcal{V}^*$ opens the commitment in step (V1) is p in each (independent) try. Hence, the number of rewinds is distributed geometrically, and $1 + p\sum_{i=1}^{\infty} i \cdot p(1-p)^{i-1} = 2$ is the expected number of overall iterations (including the first try). Consequently, the expected runtime doubles at most.[22]

Game G_2**:** Test if both (valid) openings of $\mathcal{V}^*$'s commitments in (P1) open to the same value. Else, G_2 outputs `ambig`, indicating equivocation of the commitment. This modification hardly affects the runtime, so it is still bounded roughly by $2t$. The probability for `ambig` is negligible, since one can (trivially) reduce to

[22] Formally arguing that the expected time is bounded by $2t$ is a bit more technical than for strict time bounds. But it follows easily from the independence of the iterations (due to fresh prover randomness), and the fact that t, in particular, upper bounds the expected time per iteration.

an adversary against the binding property of $\mathsf{Com}^{(\mathrm{B})}$. That is, there is an adversary $\mathcal{B}$ such that $|\mathbb{P}(\mathcal{D}(\mathsf{out}(\mathsf{G}_2)=1)) - \mathbb{P}(\mathcal{D}(\mathsf{out}(\mathsf{G}_1)=1))| = \mathrm{Adv}^{\mathrm{bind}}_{\mathsf{Com}^{(\mathrm{B})}}(\mathcal{B})$.

In **Game** G_3, the initial commitments (in (P1)) to 3-colourings are replaced with commitments to 0. These commitments are never opened. Thus, we can reduce distinguishing Games 2 and 3 to breaking the hiding property of $\mathsf{Com}^{(\mathrm{B})}$ modelled as left-or-right indistinguishability. More precisely, the reduction constructs real resp. all-zero colourings, and uses the *LR-challenge commitment oracle* $\mathcal{O}_b$ which receives two messages (m_0, m_1) and commits to m_b. Use m_0 to commit to the real colouring *(left)*, whereas m_1 is the all-zero colouring *(right)*. The modification of G_2 to "oracle committing" yields an EPT Game $\mathsf{G}_{2'}$ (instantiated with $\mathcal{O}_0$). The modification of G_3 to $\mathsf{G}_{3'}$ (with $\mathcal{O}_1$) is CEPT. This follows immediately from the standard reduction, because Games $\mathsf{G}_{2'}$ and $\mathsf{G}_{3'}$ differ only in their oracle, and the case of $\mathcal{O}_0$ is EPT. More precisely, the standard reduction applied to $\mathcal{O}_0$ and $\mathcal{O}_1$ yields an adversary $\mathcal{B}$ such that $|\mathbb{P}(\mathcal{D}(\mathsf{out}(\mathsf{G}_2)) = 1) - \mathbb{P}(\mathcal{D}(\mathsf{out}(\mathsf{G}_1)) = 1)| \geq \frac{1}{4}\,\mathrm{Adv}^{\mathrm{hide}}_{\mathsf{Com}^{(\mathrm{B})}}(\mathcal{B})$ infinitely often, assuming $\mathcal{B}$ has non-negligible advantage.

Consequently, Game $\mathsf{G}_{3'}$ is efficient with (oracle) runtime $T_{3'} \overset{c}{\approx} T_{2'}$, and the output distributions of Games $\mathsf{G}_{2'}$ and $\mathsf{G}_{3'}$ are indistinguishable. Finally, note that Game G_3 and $\mathsf{G}_{3'}$ only differ by (not) using oracle calls. Incorporating these oracles does not affect CEPT (as $\mathcal{O}_1$ is an *a priori* PPT oracle). Thus, G_3 is efficient (i.e. CEPT) as well. Assuming the time to compute a commitment depends only on the message length, a precise analysis shows, that the (virtually) expected time is affected negligibly (up to machine model artefacts).

In **Game** G_4, the commitments in the reiterations of (P1) are replaced by commitments to pseudo-colourings for each e_i, that is, at the challenge edge e_i, two random different colours are picked, and all other colours are set to 0. If $\mathcal{V}^*$ equivocates, the game outputs `ambig`. The argument for efficiency and indistinguishability of outputs is analogous to the step from Game G_2 to Game G_3. It reduces all replacements to the hiding property in a single step. This is possible since our definition of hiding is left-or-right oracle indistinguishability with an arbitrary number of challenge commitments. As before, a precise analysis shows that the (virtually) expected time is affected negligibly.

All in all, if G_0 runs in (virtually) expected time t, then G_4 runs in expected time about $2t$, ignoring the overhead introduced by $\mathsf{bb\text{-}rw}$ emulation, etc. Moreover, the output is indistinguishable, i.e. $|\mathbb{P}(\mathcal{D}(\mathsf{out}(\mathsf{G}_4)) = 1) - \mathbb{P}(\mathcal{D}(\mathsf{out}(\mathsf{G}_0))) = 1| \leq \mathsf{negl}$.

The simulator is defined as in G_4: It makes a first test-run with an all-zeroes colouring. If the verifier does not open its challenge commitment in (V1), Sim aborts (like the real prover in (P2)). Otherwise, it rewinds $\mathcal{V}^*$ (and uses pseudo-colourings) until $\mathcal{V}^*$ opens the challenge commitment again, and outputs the verifier's final output of this run (or `ambig`). (To prevent non-halting executions, we may abort after, e.g., $2^{2\kappa}$ steps. But this is not necessary for our results.)

We point out some important parts of the proof: First, in Game G_1, rewinding and its preservation of EPT is unconditional. That is, rewinding is separated from the computational steps happening after it. Second, since the simulator's

time per iteration is roughly that of prover and verifier, the total simulation time is CEPT (and roughly virtually expected $2t$).

There is only one obstacle to extend our result to CEPT adversaries. It is not obvious, whether the introduction of rewinding in G_1 preserves CEPT. Fortunately, this is quite simple to see: The probability that a certain commitment is sent in (P1) increases, since the verifier is rewound and many commitments may be tried. However, the probability only increases by a factor of 2. Thus, "bad" queries are only twice as likely as before.

Proof (Proof sketch of Lemma 7). G_0 **to** G_1**:** Fix the first message $\mathsf{ck}_{\mathrm{hide}}$ of $\mathscr{P}$ to $\mathsf{bbrw}(\mathcal{V}^*)$ and the randomness of $\mathcal{V}^*$ (which is fixed since we consider a $\mathsf{bb\text{-}rw}$ oracle). Let $p_b(c)$ be the probability, that in protocol step (P1) G_b sends $c = \{\{c_{i,j}^{\psi}\}_{j\in V}\}_{i=1,\dots,N}$ to $\mathsf{bbrw}(\mathcal{V}^*)$ at least once. (For G_0, also at most once. But rewinding in G_1 increases the chances.) Let γ_i denote the i-th query sent in step (P1) (or $\perp$ if none was sent), let the random variable I denote the total number of queries. Then

$$\begin{aligned} p_1(c) \;&=\; \mathbb{P}(\exists j \le i\colon \gamma_j = c \wedge I \le j) \;\le\; \sum_{i=1}^{\infty} \mathbb{P}(I \ge i \wedge \gamma_i = c) \\ &=\; \sum_{i=1}^{\infty} \mathbb{P}(I \ge i)\,\mathbb{P}(\gamma_i = c \mid I \ge i) \;\le\; \sum_{i=1}^{\infty} \mathbb{P}(I \ge i)\cdot p_0(c) \;=\; \mathbb{E}(I)\cdot p_0(c). \end{aligned}$$

In the penultimate equality, we use that, for any fixed i, γ_i is a fresh random commitment (or never sampled, if $I < i$). As argued before, $\mathbb{E}(I) = 2$, hence $p_1(c) \le 2p_0(c)$. Thus, the probability $p_1(c)$ for G_1 to issue query c is at most twice that of G_0. By averaging over first messages c (according to prover randomness), the derivation extends to our setting of interest, where c is chosen randomly by $\mathscr{P}$. Next, we conclude from this, that the virtuality at most doubles.

We argue similar to the "good set/runtime" from Corollary 1, but with interactive machines. That is, we define an oracle $\mathcal{V}'$, which will be EPT, as follows: Consider the "behavioural decision" tree for $\mathcal{V}^*$ where the root has edges which are labelled by a choice of random tape for $\mathcal{V}^*$. The edges in lower layers are labelled by messages which $\mathcal{V}^*$ can receive. The nodes are labelled with the runtime spent to answer the string of message on the path. We can now construct a runtime cutoff at t as follows: Replace every node which is labelled with $\ell > t$ by a `timeout` node. Let $\mathcal{V}'$ be a oracle which acts according to this decision tree. That is, $\mathcal{V}'$ acts exactly like $\mathcal{V}^*$, except if a `timeout` node is chosen. In that case $\mathcal{V}'$ outputs `timeout` and shuts down. Thus, $\mathcal{V}^*$ and $\mathcal{V}'$ are equal until `timeout`. Suppose for simplicity, that there is a superpolynomial t, such that truncation at t yields an EPT $\mathcal{V}'$, i.e. $\mathsf{time}_{\mathcal{V}'}(\mathsf{G}_0)$ is EPT.[23]

[23] There are two small problems, which are dealt with in detail in the full version [21]. First, we argued using runtime cutoffs instead of quantile cutoffs (without justification). Assuming there exists a unique t such that $\mathbb{P}(T \le t) = 1 - \nu$, then the cutoff at t is the ν-quantile. Otherwise, defining $\mathcal{V}'$ is a bit more technical. The second problem is that $\mathcal{V}'$ is not an algorithm, it is a "timeful system", i.e. a system which

Denote by G'_0 the modification of G_0 which uses $\mathcal{V}'$ instead of $\mathcal{V}^*$, and let G'_0 immediately output `timeout` if $\mathcal{V}'$ does. Then $\mathsf{time}_{\mathcal{V}'}(\mathsf{G}'_0)$ is EPT by construction, and essentially equals the virtual expected time of $\mathsf{time}_{\mathcal{V}^*}(\mathsf{G}_0)$. The statistical distance $\Delta(\mathsf{G}_0, \mathsf{G}'_0)$ is exactly the probability that $\mathcal{V}'$ outputs `timeout`. Let G'_1 be defined analogously to G'_0.

Let $\texttt{timeout}(\mathit{query})$ be 1 if *query* causes a `timeout` and 0 otherwise. Then

$$\begin{aligned}\mathbb{P}_{\mathsf{G}'_1}(\texttt{timeout}) &= \sum_{\mathit{query}} \texttt{timeout}(\mathit{query}) \cdot p_1(\mathit{query}) \\ &\leq 2 \sum_{\mathit{query}} \texttt{timeout}(\mathit{query}) \cdot p_0(\mathit{query}) = 2 \cdot \mathbb{P}_{\mathsf{G}'_0}(\texttt{timeout}).\end{aligned}$$

Since the probability for `timeout` bounds the virtuality if we use $\mathcal{V}^*$ instead of $\mathcal{V}'$, this shows that G_1 is CEPT, with virtuality 2ε. If G_0 always halts, the outputs of G_1 and G_0 are identically distributed. In general, the statistical distance is (at most) $2 \cdot \mathbb{P}(\mathsf{G}_0 = \texttt{nohalt})$; this follows as for virtuality, which must encompass the probability of non-halting executions. Conditioned on halting executions, the distributions G_0 and G_1 are identical. The transition to G_2 now relies on the standard reduction, all other steps of Lemma 8 apply literally.

In the full version [21], we abstract the above proof strategy, by defining rewinding strategies, reductions to "simple assumptions", and benign simulators, as sketched in the introduction.

Remark 9. With an analogous proof, one finds that the simulator in [15] is also a CEPT simulator. Its advantage is, that it handles adversaries which are *a priori PPT*, as well as *EPT w.r.t. any reset attack* [14], without introducing any "virtuality", i.e. the simulation is EPT. On the other hand, it increases virtuality of CEPT adversaries by a larger factor.

5 Hybrid argument and sequential composition

We formally state the hybrid lemma, and sketch its application to composition of zero-knowledge proofs.

5.1 Hybrid lemma

Definition 11 (Relative efficiency). *Let* A *and* B *be two (interactive) algorithms with identical interfaces. We say that* B *is* ***weakly*** $(\mathcal{T}, \mathcal{S})$***-efficient relative to*** A w.r.t. (implicit) runtime classes $\mathcal{T}$, $\mathcal{S}$, *if for all distinguishing environments* $\mathcal{E}$ *(which yield closed systems* $\langle \mathcal{E}, \mathsf{A} \rangle$, $\langle \mathcal{E}, \mathsf{B} \rangle$*) we have*

$$\mathsf{time}_{\mathcal{E}+\mathsf{A}}(\langle \mathcal{E}, \mathsf{A} \rangle) \in \mathcal{T} \implies \mathsf{time}_{\mathcal{E}+\mathsf{A}}(\langle \mathcal{E}, \mathsf{B} \rangle) \in \mathcal{S}.$$

has an associated notion of runtime. One way around this is to note that we do not need $\mathcal{V}'$ to be an algorithm. It is merely a formalism to track the change of virtuality.

In the following, we assume $\mathcal{T} = \mathcal{S} = \mathcal{CEPT}$ unless specified otherwise. Note that, except for notation, Definition 11 considers oracle indistinguishability.

Example 4. Viewing the real (resp. simulated) interaction in Definition 10 as oracles $\mathcal{O}_{\mathscr{P}}$ (resp. $\mathcal{O}_{\mathsf{Sim}}$), security implies that $\mathcal{O}_{\mathscr{P}} \overset{c}{\approx} \mathcal{O}_{\mathsf{Sim}}$ and $\mathcal{O}_{\mathsf{Sim}}$ is weakly efficient relative to $\mathcal{O}_{\mathscr{P}}$

Now, we define a natural generalization of distinguishing with repeated samples, but for general oracle indistinguishability.

Definition 12 (Repeated oracle access). *Let $\mathcal{O}$ be an oracle. We denote by $\mathsf{rep}(\mathcal{O})$ an oracle which offers repeated access to* independent instances *of $\mathcal{O}$. We denote by $\mathsf{rep}_q(\mathcal{O})$ an oracle which limits access to a total of q instances of $\mathcal{O}$.*

Note that $\mathsf{rep}(\mathcal{O}) = \mathsf{rep}_\infty(\mathcal{O})$. We can now state the hybrid argument.

Lemma 9 (Hybrid-Lemma for CEPT). *Suppose $\mathcal{O}_1$ is weakly efficient relative to $\mathcal{O}_0$ and suppose $\mathcal{O}_0 \overset{c}{\approx} \mathcal{O}_1$. Let $\mathcal{D}$ be an algorithm with oracle-access to $\mathsf{rep}(\mathcal{O}_b)$, and suppose that $\mathsf{time}_{\mathcal{D}+\mathsf{rep}(\mathcal{O}_0)}(\mathcal{D}^{\mathsf{rep}(\mathcal{O}_0)}) \in \mathcal{CEPT}$. Then $\mathsf{time}_{\mathcal{D}+\mathsf{rep}(\mathcal{O}_1)}(\mathcal{D}^{\mathsf{rep}(\mathcal{O}_1)}) \in \mathcal{CEPT}$ and the distinguishing advantage is*

$$|\mathbb{P}(\mathcal{D}^{\mathsf{rep}(\mathcal{O}_0)} = 1) - \mathbb{P}(\mathcal{D}^{\mathsf{rep}(\mathcal{O}_1)} = 1)| \leq \mathsf{negl}.$$

That is, $\mathsf{rep}(\mathcal{O}_1)$ is weakly efficient relative to $\mathsf{rep}(\mathcal{O}_0)$, and $\mathsf{rep}(\mathcal{O}_0) \overset{c}{\approx} \mathsf{rep}(\mathcal{O}_1)$.

Due to limited pages, we refer to the introduction (Section 1.4) for a proof sketch. A full proof is given in [21].

5.2 Sequential zero-knowledge

Definition 13 (Sequential zero-knowledge). *Let $(\mathscr{P}, \mathscr{V})$ be a zero-knowledge argument. Suppose Sim is a universal simulator. Modify Definition 10 as follows: Replace $\mathcal{I}$ by $\mathcal{E}^{\mathcal{O}}$ which can repeatedly call its oracle $\mathcal{O}$ (either $\mathcal{O}_{\mathscr{P}}$ or $\mathcal{O}_{\mathsf{Sim}}$). Define real and ideal executions and the advantage accordingly. For security, require the analogue of the efficiency for Sim and negligible advantage.*

Some remarks are in order. First, one $\mathscr{V}^*$ is fixed for all calls to $\mathcal{O}$, but one may assume a universal $\mathscr{V}^*$ anyway. Second, $\mathcal{E}$ can adaptively choose (x, w, aux) in its call to $\mathcal{O}$. Third, effectively, Definition 13 stipulates that $\mathsf{rep}(\mathcal{O}_{\mathsf{Sim}})$ is weakly efficient relative to $\mathsf{rep}(\mathcal{O}_{\mathscr{P}})$, and $\mathsf{rep}(\mathcal{O}_{\mathsf{Sim}}) \overset{c}{\approx} \mathsf{rep}(\mathcal{O}_{\mathscr{P}})$.

We are now ready to state and prove the sequential composition lemma.

Lemma 10 (Sequential composition lemma). *Let $(\mathscr{P}, \mathscr{V})$ be an argument system. Suppose Sim is a simulator for auxiliary input zero-knowledge (which handles CEPT adversaries in CEPT). Then $(\mathscr{P}, \mathscr{V})$ is sequential zero-knowledge (with the same simulator Sim).*

The proof is an almost trivial consequence of the hybrid lemma.

Proof. Let $(\mathcal{E}, \mathcal{V}^*, \mathcal{D})$ be a CEPT adversary against sequential zero-knowledge. Let $\mathcal{O}_{\mathcal{P}}(x, w, aux)$ and $\mathcal{O}_{\mathsf{Sim}}(x, w, aux)$ be as in Definition 13. By definition,

$$\mathsf{Real}_{\mathcal{E},\mathcal{V}^*}(\kappa) = \mathsf{out}_{\mathcal{E}}\langle \mathcal{E}, \mathsf{rep}(\mathcal{O}_{\mathcal{P}})\rangle \qquad \text{and} \qquad \mathsf{Ideal}_{\mathcal{G},\mathsf{Sim}}(\kappa) = \mathsf{out}_{\mathcal{E}}\langle \mathcal{E}, \mathsf{rep}(\mathcal{O}_{\mathsf{Sim}})\rangle$$

Define a distinguisher $\mathcal{A}$ for $\mathsf{rep}(\mathcal{O}_{\mathcal{P}})$ and $\mathsf{rep}(\mathcal{O}_{\mathsf{Sim}})$ as $\mathcal{D}(\mathsf{out}_{\mathcal{E}}\langle \mathcal{E}, \mathsf{rep}(\mathcal{O})\rangle)$. Now, we are in the usual setting of oracle (in)distinguishability. Since Sim is an auxiliary input zero-knowledge simulator for $(\mathcal{P}, \mathcal{V})$, we have that $\mathcal{O}_{\mathsf{Sim}}$ is weakly efficient relative to $\mathcal{O}_{\mathcal{P}}$ and that $\mathcal{O}_{\mathcal{P}} \overset{c}{\approx} \mathcal{O}_{\mathsf{Sim}}$. Thus, the hybrid lemma (Lemma 9) is applicable. Hence $\mathsf{rep}(\mathcal{O}_{\mathcal{P}})$ is weakly relative efficient to $\mathsf{rep}(\mathcal{O}_{\mathsf{Sim}})$ and $\mathsf{rep}(\mathcal{O}_{\mathcal{P}}) \overset{c}{\approx} \mathsf{rep}(\mathcal{O}_{\mathsf{Sim}})$. This concludes the proof.

Remark 10. In the real-ideal setting for secure function evaluation (SFE), our definition of auxiliary input and sequential security are analogous to zero-knowledge. The modular sequential composition theorem [5, 19] stipulates that if a protocol π is secure in F-hybrid model (and uses F-calls sequentially), then it π^ρ is secure, where F-calls are replaced by subprotocol calls to ρ, and ρ securely realizes F. Even in our a posteriori setting, the proof is straightforward, since it is basically an application of the hybrid lemma, similar to Lemma 10.

Acknowledgements. I am grateful to Alexander Koch and Jörn Müller-Quade for feedback on an entirely different approach on EPT, and to Dennis Hofheinz for essentially breaking said approach. I also extend my gratitude to the reviewers of CRYPTO'20/21 and TCC'21, and to Akin Ünal and Marcel Tiepelt, whose suggestions helped to improve the overall presentation. Special thanks go to the reviewers of TCC'20 and Dakshita Khurana for great feedback, which eventually resulted in the addition of the hybrid lemma. This work was supported by KASTEL Security Research Labs.

References

[1] Boaz Barak. "How to Go Beyond the Black-Box Simulation Barrier". In: *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*. IEEE Computer Society, 2001.

[2] Boaz Barak and Yehuda Lindell. "Strict Polynomial-Time in Simulation and Extraction". In: *SIAM J. Comput.* 33.4 (2004).

[3] Andrej Bogdanov and Luca Trevisan. "Average-Case Complexity". In: *Foundations and Trends in Theoretical Computer Science* 2.1 (2006).

[4] Dan Boneh and Victor Shoup. *A Graduate Course in Applied Cryptography*. Version 0.5. 2020.

[5] Ran Canetti. "Security and Composition of Multiparty Cryptographic Protocols". In: *J. Cryptol.* 13.1 (2000).

[6] Kai-Min Chung, Edward Lui, and Rafael Pass. "From Weak to Strong Zero-Knowledge and Applications". In: *TCC (1)*. Vol. 9014. Lecture Notes in Computer Science. Springer, 2015.

[7] Ning Ding and Dawu Gu. "On Constant-Round Precise Zero-Knowledge". In: *Information and Communications Security - 14th International Conference, ICICS 2012, Hong Kong, China, October 29-31, 2012. Proceedings*. Ed. by Tat Wing Chim and Tsz Hon Yuen. Vol. 7618. Lecture Notes in Computer Science. Springer, 2012.

[8] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. "Magic Functions". In: *J. ACM* 50.6 (2003).

[9] Uriel Feige. "Alternative models for zero-knowledge interactive proofs". PhD thesis. Weizmann Institute of Science, 1990.

[10] Craig Gentry and Daniel Wichs. "Separating succinct non-interactive arguments from all falsifiable assumptions". In: *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*. Ed. by Lance Fortnow and Salil P. Vadhan. ACM, 2011.

[11] Oded Goldreich. "A Uniform-Complexity Treatment of Encryption and Zero-Knowledge". In: *J. Cryptology* 6.1 (1993).

[12] Oded Goldreich. "Average Case Complexity, Revisited". In: *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation - In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*. Ed. by Oded Goldreich. Vol. 6650. Lecture Notes in Computer Science. Springer, 2011.

[13] Oded Goldreich. "Notes on Levin's Theory of Average-Case Complexity". In: *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation - In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*. Ed. by Oded Goldreich. Vol. 6650. Lecture Notes in Computer Science. Springer, 2011.

[14] Oded Goldreich. "On Expected Probabilistic Polynomial-Time Adversaries: A Suggestion for Restricted Definitions and Their Benefits". In: *J. Cryptol.* 23.1 (2010).

[15] Oded Goldreich and Ariel Kahan. "How to Construct Constant-Round Zero-Knowledge Proof Systems for NP". In: *J. Cryptology* 9.3 (1996).

[16] Oded Goldreich, Silvio Micali, and Avi Wigderson. "Proofs that Yield Nothing But their Validity and a Methodology of Cryptographic Protocol Design (Extended Abstract)". In: *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*. IEEE Computer Society, 1986.

[17] Shai Halevi and Silvio Micali. "More on Proofs of Knowledge". In: *IACR Cryptol. ePrint Arch.* 1998 (1998).

[18] Dennis Hofheinz, Dominique Unruh, and Jörn Müller-Quade. "Polynomial Runtime and Composability". In: *J. Cryptology* 26.3 (2013).

[19] Jonathan Katz and Yehuda Lindell. "Handling Expected Polynomial-Time Strategies in Simulation-Based Security Proofs". In: *J. Cryptol.* 21.3 (2008).

[20] Joe Kilian and Erez Petrank. "Concurrent and resettable zero-knowledge in poly-loalgorithm rounds". In: *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*. Ed. by Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis. ACM, 2001.

[21] Michael Klooß. "On (expected polynomial) runtime in cryptography". In: *IACR Cryptol. ePrint Arch.* (2020).

[22] David Lanzenberger and Ueli Maurer. "Coupling of Random Systems". In: *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part III*. Ed. by Rafael Pass and Krzysztof Pietrzak. Vol. 12552. Lecture Notes in Computer Science. Springer, 2020.

[23] Leonid A. Levin. "Average Case Complete Problems". In: *SIAM J. Comput.* 15.1 (1986).

[24] Yehuda Lindell. "A Note on Constant-Round Zero-Knowledge Proofs of Knowledge". In: *J. Cryptol.* 26.4 (2013).

[25] Silvio Micali and Rafael Pass. "Local zero knowledge". In: *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*. Ed. by Jon M. Kleinberg. ACM, 2006.

[26] Moni Naor. "On Cryptographic Assumptions and Challenges". In: *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*. Ed. by Dan Boneh. Vol. 2729. Lecture Notes in Computer Science. Springer, 2003.

[27] Rafael Pass. "A precise computational approach to knowledge". PhD thesis. Massachusetts Institute of Technology, Cambridge, MA, USA, 2006.

[28] Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. "Concurrent Zero Knowledge, Revisited". In: *J. Cryptol.* 27.1 (2014).

[29] Alon Rosen. "A Note on Constant-Round Zero-Knowledge Proofs for NP". In: *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*. Ed. by Moni Naor. Vol. 2951. Lecture Notes in Computer Science. Springer, 2004.