Limits of Provable Security From Standard Assumptions

Rafael Pass*
Cornell University
rafael@cs.cornell.edu

April 2, 2011

Abstract

We show that the security of some well-known cryptographic protocols, primitives and assumptions (e.g., the Schnorr identification scheme, commitments secure under adaptive selective-decommitment, the "one-more" discrete logarithm assumption) cannot be based on *any standard assumption* using a Turing (i.e., black-box) reduction. These results follow from a general result showing that Turing reductions cannot be used to prove security of *constant-round sequentially witness-hiding special-sound protocols* for *unique witness* relations, based on standard assumptions; we emphasize that this result holds even if the protocol makes *non-black-box* use of the underlying assumption.

*First version from November 4, 2010. This revision contains new results on blind signatures (Section 9). Pass is supported in part by a Alfred P. Sloan Fellowship, Microsoft New Faculty Fellowship, NSF CAREER Award CCF-0746990, AFOSR Award FA9550-08-1-0197, BSF Grant 2006317.

Contents

1 Introduction **2**
- 1.1 Our Results . . . 3
- 1.2 Other Related Work . . . 5
- 1.3 Proof Techniques . . . 6
- 1.4 On Non-black-box Reductions . . . 8
- 1.5 Overview . . . 8

2 Preliminaries **8**
- 2.1 Notation . . . 8
- 2.2 Basic Definitions . . . 9
- 2.3 Black-box reductions . . . 9
- 2.4 Indistinguishability . . . 10
- 2.5 Witness Relations . . . 10
- 2.6 Interactive Proofs and Arguments . . . 10
- 2.7 Witness Hiding . . . 11

3 New Definitions **11**
- 3.1 Computational Special-soundness . . . 11
- 3.2 Intractability Assumptions . . . 13

4 The Main Theorem **14**

5 Proof of the Main Theorem **16**

6 Security of Identification Schemes with Unique Secret-Keys **24**

7 Security against Adaptive Selective Decommitment **27**
- 7.1 Commitments . . . 27
- 7.2 Defining adaptive selective decommitment security . . . 28
- 7.3 Instantiating GMW . . . 28
- 7.4 Limits of Adaptive-Selective Decommitment Security . . . 29

8 Security of Generalized One-more Inversion Assumptions **31**
- 8.1 Extensions to homomorphic certified permutations . . . 33

9 Security of Unique Blind Signatures **34**

10 Acknowledgements **39**

1 Introduction

Modern Cryptography relies on the principle that cryptographic schemes are proven secure based on mathematically precise assumptions; these can be *general*—such as the existence of one-way functions—or *specific*—such as the hardness of factoring products of large primes. The security proof is a *reduction* that transforms any attacker A of the scheme into a machine that breaks the underlying assumption (e.g., inverts an alleged one-way function). This study has been extremely successful, and during the past four decades many cryptographic tasks have been put under rigorous treatment and numerous constructions realizing these tasks have been proposed under a number of well-studied complexity-theoretic intractability assumptions. But there are some well-known protocols, primitives, and assumptions, that have resisted security reductions under well-studied intractability assumptions.

The Schnorr identification scheme Schnorr's identification [Sch91] scheme is one of the most well-known and commonly-used identification schemes. For instance, recent usage includes the BlackBerry Router protocol. Schnorr showed security of this scheme under a passive (eavesdropper) attack based on the discrete logarithm assumption. But what about active (malicious) attacks? In particular, can security under active—even just sequential—attacks be based on the discrete logarithm assumption, or any other "standard" intractability assumption?[1]

The adaptive selective decommitment problem Assume a polynomial-time adversary receives a number of commitments and may then *adaptively* request openings of, say, half of them. Do the unopened commitments still remain hiding? This problem was first formalized by Dwork, Naor, Reingold and Stockmeyer [DNRS03] but according to them it arouse over 25 years ago in the context of distributed computing.[2] As noted by Dwork et al, random-oracle-based commitment schemes easily satisfy this property. Can we construct non-interactive (or even two-round) commitment schemes that provably satisfy this property based on any standard intractability assumption?[3]

One-more inversion assumptions Can a polynomial-time algorithm A, given a prime-order group G and a generator g for G, find the discrete logarithm (w.r.t to the generator g) to ℓ "target" points $y_1, \ldots, y_\ell \in G$ if it may make $\ell - 1$ queries to a discrete logarithm oracle (for the group G and generator g)? The "one-more" discrete-logarithm assumption states that no such algorithm exists. Assumptions of this "one-more inversion" type were introduced by Bellare, Namprempre, Pointcheval and Semanko [BNPS03] and have been used to prove the security of a number of practical schemes; for instance, Bellare and Palacio [BP02] have shown active security of Schnorr's identification scheme under the one-more discrete logarithm assumption. But can the security of these type of one-more inversion assumptions be based on more standard-type assumptions?

Unique Blind Signatures In 1982, Chaum [Cha82] introduced the concept of a "blind signature"—roughly speaking, a signature scheme where a user may ask a signer S to sign a message m

[1] As we shall discuss shortly, Bellare and Palacio [BP02] have demonstrated that Schnorr's scheme is secure under active attacks under a new type of "one-more inversion" assumption.

[2] We remark that DNRS focused their treatment on a non-adaptive version of this game where the adversary must select all the commitments to be opened up in a single shot; the general adaptive version is considered in Remark 7.1 in [DNRS03].

[3] Interestingly, the related question of constructing *encryption schemes* secure under selective decryption based on "standard-type assumptions" has recently been solved [BHY09].

while keeping the content of m secret from S—and provided its first implementation. His scheme is non-interactive (just as traditional signature schemes) and has the desirable property that for every message there is a *unique* valid signature. Can the security of his scheme, or more generally, any "unique non-interactive blind signature" (i.e., non-interactive blind signature schemes with unique signatures), be based on standard assumptions?

Witness hiding of parallelized versions of classical zero-knowledge protocols It is well-known that parallelized versions of the classic three-round zero-knowledge protocols of [GMR89, GMW91, Blu86] are *witness indistinguishable* [FS90]. As shown by Feige and Shamir [FS90], for certain languages with multiple witnesses, they are also *witness hiding* under sequential (and even concurrent) composition. Can we prove that these protocols also are witness hiding under sequential composition for languages with *unique witnesses* based on any standard assumption?

In this paper, we present *negative* answers to the above questions for a very liberal definition of "standard intractability assumptions", if we restrict to Turing (i.e., black-box) reductions—that is, reductions that use the alleged attacker A as a black-box.

More precisely, following Naor [Nao03] (see also [DOP05, HH09, RV10]), we model an *intractability assumption* as an arbitrary game between a (potentially) unbounded challenger C, and an attacker A.[4] A is said to break the assumption C with respect to the threshold t if it can make C output 1 with probability non-negligibly higher than the threshold t. The only restriction we put on C is that it communicates with A in an *a priori* fixed polynomial number of rounds; we refer to such assumptions as *polynomial-round intractability assumptions*. All traditional cryptographic hardness assumptions (e.g., the hardness of factoring, the hardness of the discrete logarithm problem, the decisional Diffie-Hellman problem etc.) can be modeled as 2-round challengers C with the threshold t being either 0 (in case of the factoring or discrete logarithm problems) or 1/2 (in case of the decisional Deffie-Hellman problem). But also the "one-more discrete logarithm assumption" for a *fixed* polynomial l, or the assumption that a specific protocol (P, V) is witness-hiding under a single, or even an *a priori* bounded number of, interactions can be modeled as polynomial-round challengers C (with the threshold $t = 0$).

1.1 Our Results

Our main result shows that it is impossible to (Turing) reduce any polynomial-round intractability assumption to the *witness hiding under sequential composition* property of certain types of constant-round arguments of knowledge protocols—called "computationally special-sound" protocols—for languages with *unique witnesses*. All our specific lower-bounds follow as corollaries of this result.

Recall that a three-round public-coin interactive proof is said to be *special-sound* [CDS94], if a valid witness to the statement x can be efficiently computed from any two accepting proof-transcripts of x which have the same first message but different second messages. We consider a relaxation of this notion—which we simply call *computational special-soundness*—where a) the number of communication rounds is any constant (instead of just three), b) the extractor may need a polynomial number of accepting transcripts (instead of just two), and c) extraction need only succeed if the transcripts are generated by communicating with a computationally-bounded prover. All traditional constant-round public-coin proofs of knowledge protocols (such as [GMR89,

[4] In contrast to [Nao03], we do not insist that the challenger is efficient; since our aim is to prove lower bounds, this only strengthens our results.

GMW91, Blu86, Sch91], as well as instantiations of [GMW91, Blu86] using statistically-hiding commitments) satisfy this property, and continue to do so also under parallel repetition.

Theorem 1 (Main Theorem - Informally stated)**.** *Let (P, V) be a constant-round computationally special-sound interactive argument with super logarithmic-length verifier messages for the language L with unique witnesses. Assume there exists a polynomial-time Turing reduction R such that R^A breaks the $r(\cdot)$-round assumption C (where r is a polynomial) w.r.t. the threshold t for every A that breaks sequential witness hiding of (P, V). Then C with respect to the threshold t can be broken in polynomial-time.*

Our main theorem is most closely related to the work of Haitner, Rosen and Shaltiel [HRS09]. As we explain in more detail in Section 1.2, [HRS09] also present lower bounds for using Turing reductions to demonstrate witnes hiding for constant-round public-coin protocols. They consider a more general class of protocols than we do, and they present a lower-bound for "stand-alone" (as opposed to sequential) witness hiding. However, their lower bound only applies to *restricted* classes of Turing reductions, whereas our lower bound applies to *arbitrary* Turing reductions.

Our main theorem directly rules out using Turing reductions for demonstrating sequential witness hiding of parallelized versions of classical zero-knowledge protocols for languages with unique witnesses based on polynomial-round intractability assumptions. We next show that all the previously mentioned questions can be restated in the language of "sequential witness hiding for unique witness languages", and we can thus use our main theorem to provide negative answers also to those questions.

Schnorr's identification scheme: To rule out Turing reductions for proving security of the Schnorr identification scheme, note that Schnorr's protocol is a special-sound proof for a unique witness language. Next, if Schnorr's protocol is not sequentially witness hiding, then it cannot be a secure identification scheme: The witness in this proof is the identifier's secret-key, so if an attacker can recover it after hearing polynomially many proofs, we can trivially violate the security of the identification scheme. (We mention that, on the other hand, the related scheme by Okamoto's scheme [Oka92], can be proven secure based on the discrete logarithm assumption. However, the language considered in Okamoto's protocol does not have a unique witnesses.)

Adaptive selective decommitment: To rule out solutions to the selective-decommitment problem, we extend one of the results from [DNRS03] to show that if implementing the commitment scheme in GMW's Graph 3-Coloring protocol with non-interactive (or two-round) commitments that are secure under adaptive selective decommitment, then the resulting protocol is sequentially witness hiding for any efficiently samplable hard language with unique witnesses;[5] so assuming the existence of an efficiently samplable hard language with unique witnesses, Turing reductions cannot be employed to reduce adaptive selective-decommitment security of such commitments to any polynomial-round intractability assumption.

One more inversion assumptions: As mentioned, Bellare and Palacio [BP02] have shown that the security of the Schnorr scheme can be based on the "one-more discrete log" assumption, so by their work, we directly get that polynomial-round intractability assumptions cannot be Turing-reduced to the one-more discrete log assumption. By directly constructing appropriate

[5] Dwork, Naor, Reingold and Stockmeyer [DNRS03] show that the GMW protocol instantiated with "plain" (i.e., non-adaptive) selective-decommitment secure commitment satisfies some weak notions of zero-knowledge which, in particular, imply *single-instance* witness-hiding for unique witness relations.

special-sound protocols, we can generalize this result to rule out even weaker types of "many-more" discrete logarithm assumptions (where we require that it is hard to find ℓ inverses when having access to only ℓ^ϵ inverses, where $\epsilon > 0$.) Using the same approach, we get that polynomial-round intractability assumptions cannot be Turing reduced to many-more variants of the RSA problem either (and more generally, any family of certified permutations that is additive homomorphic).

Unique blind signatures: The notion of unforgeability for blind signature schemes [PS00] requires that no attacker having requested ℓ signatures (where ℓ is an arbitrary polynomial) can come up with $\ell+1$ signatures. We show that the existence of a unique non-interactive blind signatures implies that for every polynomial ℓ, there exists a computationally special-sound protocol for a unique witness language that is witness hiding under ℓ sequential repetitions; this suffices for applying our main theorem to rule out using Turing reductions for basing the security of such blind signature schemes on polynomial-round intractability assumptions.

On the soundness of the Random Oracle and Generic Group models As mentioned, in the Random Oracle model [BR93], commitments secure against adaptive-selective decommitments are easy to construct (see [DNRS03]); thus, by our results, this yields (yet another, see e.g., [CGH04, GK03] example of a Random Oracle based scheme that cannot be provably instantiated using a concrete function, if we restrict to Turing reductions from polynomial-round intractability assumptions. The results of [CGH04, GK03] are stronger in the sense that any instantiation of their scheme with a concrete function can actually be *broken*. In contrast, we just show that the instantiated scheme cannot be *proven secure* using a Turing reduction based on a polynomial-round intractability assumption. On the other hand, the separations of [CGH04, GK03] consider artifical protocols, whereas the protocol we consider arguably is natural. In this respect our separation is similar to that of [DOP05] (which also considers a natural protocol, and rules out proofs of security), but is stronger as [DOP05] only rules out "generic" Turing reductions (see [DOP05] for more details) whereas we rule out arbitrary Turing reductions.

We also mention that Shoup [Sho97] has proven that Schnorr's identification scheme is secure in the *generic group model*; thus, our results yield a natural example where security proofs in the generic group model cannot be extended to security reductions based on any polynomial-round intractability assumption. (As far as we know, such separations had previously only been established for "artificial" protocols.)

1.2 Other Related Work

Fully black-box separations The seminal work of Impagliazzo and Rudich [IR88] provides a framework for proving black-box separations between cryptographic primitives. We highlight that this framework considers so-called "fully-black-box constructions" (see [RTV04] for a taxonomy of various black-box separations); that is, the framework considers black-box *constructions* (i.e., the higher-level primitive only uses the underlying primitive as a black-box), and black-box *reductions*. (In contrast, we consider also non-black-box constructions.) In this regime, Bellare, Hofheinz and Yilek [BHY09] present limitations of (weak notions of) fully-black-box commitment schemes secure against selective decommitment, and Haitner, Rosen and Shaltiel [HRS09] show that certain strong types of fully-black-box constructions of constant-round public-coin proofs of knowledge cannot be witness hiding.[6]

[6] More precisely, they rule out the existence of so-called "transcript-knowledge extractable" public-coin protocols; roughly speaking, these are protocols where a witness to the statement proved can be "straight-line" extracted by

Lower-bounds for general black-box reductions Turning to lower-bounds also for *non-black box* constructions, the seminal work of Goldreich and Krawczyk [GK96] shows that no constant-round public-coin protocols with negligible soundness error can be *black-box zero-knowledge*; (black-box) zero-knowledge is a significantly stronger property than sequential witness hiding, but as we shall see some of the techniques from this work will be useful for our lower bounds.

Following the works of Brassard [Bra83] and Akavia et al [AGGM06], demonstrating limitations of "NP-hard Cryptography", in [Pas06], we relate the question of demonstrating witness hiding of constant-round public-coin proofs for $\mathcal{NP}$ using black-box reductions and the question of whether one-way functions can be based on $\mathcal{NP}$-hardness (again using black-box reductions); as shown in [PTV10], this result can be interpreted as a *conditional* lower bound (under a new assumption) on the possibility of using black-box reductions to demonstrate a notion of witness hiding for constant-round public-coin proofs for $\mathcal{NP}$ based on one-way functions. We also mention the very recent concurrent work of Gentry and Wichs [GW11] which provides conditional lower bounds (assuming the existence of strong pseudorandom functions) on the possibility of using black-box reductions to prove soundness of "succint non-interactive arguments" based on so-called "falsifiable assumptions" [Nao03]. As far as we know, these are the only lower-bounds that consider non-black-box constructions and *general* (i.e., unrestricted) Turing reductions.

Lower-bounds for restricted black-box reductions To obtain stronger lower-bounds for non-black-box constructions, following the works of Feigenbaum and Fortnow [FF93] and Bogdanov and Trevisan [BT03] on the power of random self-reducible and non-adaptive reductions, several recent works prove limitations of *restricted* types of reductions for the above-mentioned problems. In this regime, Bresson, Monnerat and Vergnaud [BMV08] present limitations for basing the one-more discrete logarithm assumption on certain specific assumptions and using some restricted types of Turing reductions; Haitner, Rosen and Shaltiel [HRS09] rule out certain restricted types of reductions for demonstrating witness hiding of constant-round public-coin proof of knowledge protocols; Fischlin and Schroeder [FS10] provide lower-bounds for certain types of non-interactive blind signature schemes based on "non-interactive hardness assumptions" using certain restricted reductions. The above three works all rely on the, so-called, "meta-reduction" paradigm by Boneh and Venkatesan [BV98]; we will also rely on this method. As we shall shortly explain, in the above three works, the reasons for considering restricted reductions (and restricted assumptions) are the same; the main technical contribution of this work is circumventing these problems.

1.3 Proof Techniques

To prove our main theorem, assume there exists a Turing reduction R such that R^A breaks the assumption C whenever A breaks sequential witness hiding of a computationally special-sound argument (P, V) for a language with unique witnesses. We want to use R to directly break C *without the help of* A. So, just as in [BMV08, HRS09, FS10] (following the paradigm of [BV98]), the goal will be to efficiently emulate A for R (i.e., we will construct a "meta-reduction" which uses the underlying reduction R to break C). We will consider a particular oracle A that after hearing an appropriate number of proofs using (P, V) (acting as a verifier) simply outputs a witness to the statement proved. As in the above-mentioned earlier works, the idea is to "extract" out the witness

observing all the oracle calls to the underlying primitive. The GMW protocol satisfies this property if viewing the *commitments* in the GMW protocol as the underlying primitive. But it is unknown if the GMW protocol (when instantiated with one-way function based commitments of [Nao91, HILL99]) satisfies "transcript knowledge-extractability" if viewing the *one-way function* as the underlying primitive.

that A is supposed to provide R by "rewinding" R—after all, since (P, V) is computationally special-sound, R, intuitively, must know a witness for all statements x that it gives proofs of to A. The problem with formalizing this intuition is that the reduction R is not a "stand-alone" prover—it might *rewind and reset* the oracle A, so it is no longer clear that it needs to "know" a witness for x in order to convince A of x. To get around this problem, [BMV08, HRS09, FS10] considered *restricted* reductions which ensure that R only queries A in a "nice" way, facilitating the extraction.[7] When considering general reductions, we run into the following three problems.

1. If R "nests" its oracle calls, then a naive extraction might result in an exponential running-time; the problem is analogous to that of performing simulation in the context of *Concurrent Zero-knowledge* [DNS04].

2. When considering general assumptions C, we need to be careful not to "disturb" the outside interaction with C.

3. Finally, we need to ensure that even if we manage to appropriately rewind R, the witness we extract out actually is a valid witness.

To deal with the first two problems, we leverage the fact that we consider witness hiding under *sequential composition* (as opposed to single-instance witness hiding as in [HRS09]). This means that we only need to provide R with a witness for a statement x after it has provided sufficiently many proofs of x; now, we can use ideas from *positive results* on concurrent zero-knowledge, and, in particular, simulation techniques inspired by those of Richardson and Kilian [RK99] (and their refinements in [PV08, DGS09, CLP10]) to ensure that we can rewind R without "blowing-up" the run-time, and while ensuring that we do not disturb the outside execution with C. We mention that we cannot use these techniques in a "black-box" fashion. For instance, we do not know how to adapt the simulation technique of Kilian-Petrank [KP01] to work in our setting. There are two reasons for this: 1) The reduction R might be rewinding its oracle, so we actually need a "resettable zero-knowledge" simulation [CGGM00]. 2) In contrast to the setting of concurrent and resettable zero-knowledge, we cannot design the protocol to be analyzed—rather the only thing we know is that the protocol consists of sufficiently many repetitions of a computationally special-sound protocol. Handling these issues requires a somewhat different analysis.

To deal with the third problem, we leverage the fact that we consider computationally special-sound protocols (as opposed to general proofs of knowledge, as in [HRS09]). We show (relying on ideas from [GK96, Pas06]) that such protocols intuitively satisfy a notion of "resettably-sound" [BGGL01] proofs of knowledge (when appropriately generating the verifier messages)—that is, they remain proofs of knowledge even under resetting attacks.

On the role of unique witnesses. Let us point out exactly where in the proof the unique witness requirement is used. Recall that we are rewinding R to extract out a witness so that we can emulate the oracle A for R. If the statement x has a unique witness w, we can ensure that the extracted witness will be identical to the witness that the oracle A would have returned. When dealing with statements with multiple witnesses, this might no longer be the case—in particular, although the rewinding procedure will succeed in extracting *some* witnesses, the *distribution* of the extracted witnesses might be different than the distribution of witnesses actually provided by A; thus, we can no longer guarantee that R succeeds in breaking the assumption C. This is not

[7] For instance, [HRS09] consider reductions R, which roughly speaking, never "intertwine" proofs of different statements. This can be viewed as a generalization the "parameter-invariant" reductions of [BMV08].

just an artifact of the proof: As mentioned, for languages with multiple witnesses, constant-round, sequentially witness-hiding special-sound proofs are known [FS90].[8]

1.4 On Non-black-box Reductions

In this work we consider only Turing (i.e., black-box) reductions. As demonstrated by Barak's beautiful work [Bar01], non-black-box reductions can be used to analyze some zero-knowledge arguments. In fact, a variant of the protocol of Barak due to [PR05] is constant-round, public-coin, computationally special-sound and zero-knowledge (thus also witness hiding under sequential composition).

We would like to argue that in the context of security reductions, Turing reductions provide a semantically stronger (and more meaningful) notion of security than non-black-box reductions, and are thus interesting to study in their own right. The existence of a Turing reduction from some problem P to the task of breaking a crypto system implies that any "physical device" with a *reproducible behavior* that breaks our crypto system, can be used—with only a polynomial slow-down—to solve P. With a non-black-box reduction, we can only solve P if we have an *explicit description* of the code of the attack on the crypto system. Such descriptions might be hard to find in practice: Consider, for instance, a "human-aided" computation, where a human is interacting with a computer program in order to break the crypto system;[9] getting an explicit description of the attack requires providing an explicit (and "short") description of our brain.

If the reader does not find the above "philosophical" argument compelling, we note that, to date, non-black reductions have only been successfully used to analyze interactive protocols that are impractical (e.g., they rely on PCPs). Furthermore, such techniques have only been successful in analyzing *computationally-sound* protocols (i.e., arguments); in contrast, many of the protocols we are considering are proof systems (i.e., they are unconditionally sound). So, one conservative way to interpret our results is that "current techniques" cannot be used to prove security of, for instance, Schnorr's identification scheme.

1.5 Overview

We provide some preliminaries and standard definitions in Section 2; we provide some new definitions in Section 3, and present our main theorem in Section 4. In Section 5 we provide the proof of the the main theorem. Section 6 contains lower bounds for identification schemes, Section 7 contains our results about the selective decommitment problem, Section 8 contains lower bounds for generalized versions of one-more inversion assumptions, and in Section 9 we present lower bounds for unique blind signatures.

2 Preliminaries

2.1 Notation

Integer, Strings and Vectors. We denote by N the set of natural numbers: 0, 1, 2, ... Unless otherwise specified, a natural number is presented in its binary expansion (with no *leading* 0s)

[8] We remark that the unique witness requirement can be slightly relaxed. For instance, it suffices to ensure that the special-soundness extractor always recovers a *uniformly* chosen witness if there are many witnesses. This suffices for emulating an oracle that picks a uniform witness. We have not pursued this avenue further.

[9] Practical attacks on crypto-systems often are not fully automatized, but do indeed rely on such interactions; see e.g., [AAG+00].

whenever given as an input to an algorithm. If $n \in N$, we denote by 1^n the unary expansion of n (i.e., the concatenation of n 1's). Given a string x, we let $x|_i$ denote the i'th bit of x. We denote by $\vec{x}$ a finite sequence of elements $x_1, x_2, \ldots x_n$, and we let $|\vec{x}|$ denote the number of elements in the sequence.

Algorithms. We employ the following notation for algorithms.

Probabilistic algorithms. By a probabilistic algorithm we mean a Turing machine that receives an auxiliary random tape as input. If M is a probabilistic algorithm, then for any input x, the notation "$M_r(x)$" denotes the output of the M on input x when receiving r as random tape. We let the notation "$M(x)$" denote the probability distribution over the outputs of M on input x where each bit of the random tape r is selected at random and independently, and then outputting $M_r(x)$.

Interactive Algorithms. We assume familiarity with the basic notions of an *Interactive Turing Machine* [GMR89] (ITM for brevity) and a *protocol*. (Briefly, a protocol is pair of ITMs computing in turns. In each turn, called a round, only one ITM is active. A round ends with the active machine either halting—in which case the protocol halts—or by sending a message m to the other machine, which becomes active with m as a special input. By an interactive algorithm we mean a (probabilistic) interactive Turing Machine.

Given a pair of interactive algorithms (A, B), we let $\langle A(a), B(b)\rangle(x)$ denote the probability distribution over the outputs of $B(b)$ after interacting with $A(a)$ on the common input x.

Oracle algorithms. An oracle algorithm is a machine that gets oracle access to another machine. Given a probabilistic oracle algorithm M and a probabilistic algorithm A, we let $M^A(x)$ denote the probability distribution over the outputs of the oracle algorithm M on input x, when given oracle access to A.

We will also consider oracle algorithms that get access to *deterministic* interactive algorithms. Given a probabilistic oracle algorithm M, and a *deterministic* interactive algorithm A, we let $M^A(x)$ denote the probability distribution over the outputs of the algorithm M on input x, when given oracle access to the "next-messages" function of A (i.e., the function that on input the messages $(m_1, .., m_l)$ outputs the next messages sent by A on common input x and receiving the messages $(m_1, .., m_l)$; note that this is well defined since we only consider deterministic oracles.)

Negligible functions. The term "negligible" is used for denoting functions that are asymptotically smaller than the inverse of any fixed polynomial. More precisely, a function $\nu(\cdot)$ from non-negative integers to reals is called *negligible* if for every constant $c > 0$ and all sufficiently large n, it holds that $\nu(n) < n^{-c}$.

2.2 Basic Definitions

In this section we provide some standard definitions and preliminaries.

2.3 Black-box reductions

We consider probabilistic polynomial time Turing reductions—i.e., *black-box reductions*. A black-box reduction thus refers to a probabilistic polynomial-time oracle algorithm. Roughly speaking,

a black-box reduction for basing the security of a primitive A on the hardness of a primitive B, is a probabilistic polynomial-time oracle machine R such that R^O "breaks" B, whenever the oracle O "breaks" A. As mentioned, when considering interactive oracles, we restrict attention to *deterministic* oracles; in particular, this means that the reduction has the power to restart or "rewind" its oracle; black-box reductions in the context of interactive protocols often take advantage of this feature.

2.4 Indistinguishability

The following definition of (computational) indistinguishability originates in the seminal paper of Goldwasser and Micali [GM84].

Let X be a countable set of strings. A *probability ensemble indexed by* X is a sequence of random variables indexed by X. Namely, any element of $A = \{A_x\}_{x \in X}$ is a random variable indexed by X.

Definition 1 (Indistinguishability)**.** *Let X be a countable set. Two ensembles $\{A_{n,x}\}_{n \in N, x \in X}$ and $\{B_{n,x}\}_{n \in N, x \in X}$ are said to be* computationally indistinguishable*, if for every probabilistic machine D (the "distinguisher") whose running time is polynomial in its first input, there exists a negligible function $\nu(\cdot)$ so that for every $n \in N, x \in X$:*

$$|\Pr[D(n, x, A_{n,x}) = 1] - \Pr[D(n, x, B_{n,x}) = 1]| < \nu(n)$$

In the above expression, D is simply given a sample from $A_{x,y}$ and $B_{x,y}$, respectively. $\{A_{n,x}\}_{n \in N, x \in X}$ and $\{B_{n,x}\}_{n \in N, x \in X}$ are said to be statistically indistinguishable *over X if the above condition holds for all (possibly unbounded) machines D.*

2.5 Witness Relations

We recall the definition of a witness relation for an $\mathcal{NP}$ language [Gol01].

Definition 2 (Witness relation)**.** *A* witness relation *for a language $L \in \mathcal{NP}$ is a binary relation R_L that is polynomially bounded, polynomial time recognizable and characterizes L by $L = \{x : \exists w \, s.t. \, (x, w) \in R_L\}$.*

We say that w is a witness for the membership $x \in L$ if $(x, w) \in R_L$. We will also let $R_L(x)$ denote the set of witnesses for the membership $x \in L$, i.e., $R_L(x) = \{w : (x, w) \in L\}$. If for each $x \in L$, there exists a single $w \in R_L(x)$, we say that R_L is a *unique witness relation.*

We will be interested in probability ensembles over witness relations.

Definition 3 (Probability ensembles over witness relations)**.** *Let R_L be a witness relation. We call $\mathcal{D} = \{D_n\}_{n \in N}$ an* ensemble of distributions over R_L *if D_n is a probability distribution over $R_L \cap (\{0,1\}^{p(n)} \times \{0,1\}^*)$ where p is a polynomial.*

We call $\mathcal{D} = \{D_n\}_{n \in N}$ an ensemble of distributions over R_L with auxiliary information *if D_n is a probability distribution over $R_L \cap (\{0,1\}^{p(n)} \times \{0,1\}^*) \times \{0,1\}^*$ where p is a polynomial.*

2.6 Interactive Proofs and Arguments

We recall the standard definitions of interactive proofs and arguments.[10]

[10] We only consider interactive proofs and arguments with *perfect completeness*. We believe that, at the cost of complicating the proof, our results can be extended also to protocols where completeness only holds with overwhelming probability; we have not pursued this path.

Definition 4 (Interactive Proofs and Arguments [GMR89, BCC88])**.** *A pair of probabilistic interactive algorithms (P, V) is called an* interactive proof system *for a language L with witness relation R_L if V is polynomial-time and the following two conditions hold.*

- Completeness: *For every $x \in L$, and every $y \in R_L(x)$,*

$$\Pr\Big[\langle P(y), V\rangle(x) = 1\Big] = 1$$

- Soundness: *For every $x \notin L$, every $z \in \{0,1\}^*$ and every interactive algorithm P^**

$$\Pr\Big[(\langle P^*(z), V\rangle(x) = 0\Big] \geq \frac{1}{2}$$

In case that the soundness condition holds only with respect to a provers P^* whose running-time is polynomially bounded in the common input, the pair (P, V) is called an interactive argument system*.*

2.7 Witness Hiding

Let us recall the definition of *Witness Hiding* under sequential composition [FS90]. Given an ensemble $\mathcal{D} = \{D_n\}_{n\in N}$ of distributions over R_L with auxiliary information, let $\mathsf{WIN}^{(P,V)}(\mathcal{D}, R_L, A, p, n)$ denote the output of the following experiment: Sample (x, y, z) from D_n, and let $A(x, z)$ communicate with $P(x, y)$ in $p(n)$ sequential executions; output 1 if A finally outputs a witness w s.t. $w \in R_L(x)$.

Definition 5 (Witness Hiding)**.** *Let (P, V) be an interactive proof (argument) for L with witness relation R_L, $\mathcal{D}$ be an ensemble of distributions over R_L with auxiliary input and $p(\cdot)$ be a polynomial. We say that (P, V) is* witness-hiding *w.r.t. $\mathcal{D}, R_L$ under $p(\cdot)$ sequential repetitions if for every adversary A whose running-time is polynomially-bounded in its first input, there exists a negligible function μ such that for all $n \in N$,*

$$\Pr\Big[\mathsf{WIN}^{(P,V)}(\mathcal{D}, R_L, A, p, n) = 1\Big] \leq \mu(n)$$

We say that (P, V) is single-instance witness hiding *(or simply* witness-hiding*) w.r.t $\mathcal{D}, R_L$ if (P, V) is witness-hiding w.r.t $\mathcal{D}, R_L$ under 1 sequential repetition. We say that (P, V) is* witness-hiding *w.r.t $\mathcal{D}, R_L$ under* sequential composition *if (P, V) is witness-hiding w.r.t $\mathcal{D}, R_L$ under $p(\cdot)$ sequential repetitions for all polynomials p.*

3 New Definitions

3.1 Computational Special-soundness

Recall that a three-round public-coin[11] interactive proof is said to be special-sound [CDS94], if a valid witness to the statement x can be efficiently computed from any two accepting proof-transcripts of x which have the same first message but different second messages. For instance, Goldwasser, Micali and Rackoff's Quadratic Residuosity protocol [GMR89], and Blum's Hamiltonian Cycle (Blum-HC) protocol [Blu86] are both all special-sound. One standard relaxation of

[11] Recall that in a public-coin protocol, the verifier's messages are just independent portions of its random tape.

special-soundness only requires that a witness for x can be extracted from $m(|x|)$ accepting proof transcripts with the same first message but different second messages, where $m(\cdot)$ is a polynomial. Goldreich, Micali and Wigderson's Graph 3-Coloring (GMW-G3C) [GMW91] protocol is an example of such a proof system.

Two other relaxations are possible. We may consider protocols with more than three rounds. Furthermore, we may consider protocols that only satisfy a computational notion of special-soundness; that is, we only require that extraction is successful if the transcripts are generated in an interaction with a computationally-bounded prover.

Roughly speaking, we say that a k-round protocol is $m(\cdot)$-*computationally special sound*, if the $k-1$'th round is a verifier round and with overwhelming probability a witness to x can be extracted from any $m(|x|)$ accepting proof transcripts that have been obtained by communicating with a computationally bounded prover, where the first $k-2$'th messages are all the same, but the $k-1$'st messages are all different. We note that, for instance, GMW-G3C and Blum-HC, instantiated with *statistically-hiding* commitments, satisfy this notion of special-soundness. We refer to the $k-1$'th message as the *verifier challenge*.

Definition 6 (Computational Special-soundness)**.** *Let (P, V) be a k-round (where k is a constant) public-coin interactive argument for the language $L \in \mathcal{NP}$ with witness relation R_L. (P, V) is said to be* computationally special-sound *if there exists a polynomial $m(\cdot)$, and a polynomial-time extractor machine X, such that for every polynomial-time machine P^*, and every polynomial $p(\cdot)$, there exists a negligible function μ such that the following holds for every $x \in L$ and every auxiliary input z for P^*. Let $\vec{T} = (T_1, T_2, \ldots T^{p(|x|)})$ denote transcripts in $p(|x|)$ random executions between $P^*(x, z)$ and $V(x)$ where V uses the same randomness for the first $k-2$ rounds (thus, the first $k-2$ rounds are the same in all transcripts). Then, the probability (over the randomness used to generate $\vec{T}$) that:*

1. *$\vec{T}$ contains a set of $m(|x|)$ accepting transcripts with different round $k-1$ messages; and*
2. *$X(\vec{T}')$ does not output a witness $w \in R_L(x)$, where $\vec{T}'$ consist of the first $m(|x|)$ accepting transcripts with different round $k-1$ messages,*

is smaller than $\mu(|x|)$.

We say that a computationally special-sound protocol has a *large challenge space* if the length of the verifier challenge is $\omega(\log n)$ on common inputs of length n.

Note that if a protocol is computationally special-sound with a large challenge space, there is a canonical "extraction" procedure from a prover P^*: Honestly emulate the role of V for P^*; if the proof is accepting, rewind P^* sending it new uniformly chosen verifier challenges until we get $m(|x|)$ accepting proof transcripts; finally apply the special-soundness extractor X on these transcripts—except with negligible probability, all of the accepting proofs have different verifier challenges and extraction thus succeeds.

Remark 1. A slightly more general notion of computational special-soundness: *We note that for our proof it suffices to consider a standard generalization of public-coin protocols. It suffices that the verifier's next message function is a function of the current partial transcript and some fresh random coins: that is, in round i the verifier picks a random string r_i, and computes its next message as a function of the public transcript and r_i (but not as a function of any of the earlier coin tosses r_j, $j < i$). The knowledge extractor, on the other hand, gets access to not only the public transcript, but also all the random strings $r_1, r_3, \ldots r_{k-1}$; that is, the knowledge extractor X now takes as input a sequence of verifier views $\vec{\mathcal{V}}'$. We refer to such protocols as* generalized

computationally special-sound. *For such a generalized notion of computational special-soundness, the notion of "large challenge space" requires that the length of the randomness r_{k-1} used in the verifier challenge is $\omega(\log n)$.*

Remark 2. Why we don't work with *any* proof of knowledge: *In our proof we require the existence of an extractor that works even if the malicious prover gets to* rewind*, or "reset", the extractor. This feature is not* a priori *guaranteed by the traditional notion of a proof of knowledge; indeed, it is easy to come up with good extractors that fail under a reset attack: Consider, for instance, the following public-coin protocol where a) the Prover first commits to a string s, b) the Verifier sends a string s', and c) the Prover finally a provides proof of knowledge that x is true or that $s' = s$. A naive extractor simply sends a random string s, and then extracts a witness from the proof of knowledge protocol in steps c). It is easy to see that this extractor fails under a reset attack: The malicious prover simply finds out what s' is and next commits to $s = s'$ in step a) and uses this as a fake witness in step c). (Of course, for this particular attack, this problem can be circumvented by considering a* specific *extractor that generates the message s' by applying an appropriate hash function to the commitment, but it is unclear how this can be done in general.)*

3.2 Intractability Assumptions

Following Naor [Nao03] (see also [DOP05, HH09, RV10]), we model an intractability assumption as an interaction (or game) between a probabilistic machine C—called the challenger—and an attacker A. Both parties get as input 1^n where n is the security parameter. The only requirement is that this interaction has an *a priori* bounded polynomial number of rounds; that is, there exists some polynomial $r(\cdot)$ such that C on input 1^n communicates with A in at most $r(n)$ rounds and finally outputs either 1 or 0. Any such challenger C, together with a threshold function $t(\cdot)$ intuitively corresponds the assumption:

> *For every polynomial-time adversary A, there exists a negligible function μ such that for all $n \in N$, the probability that C outputs 1 after interacting with A is bounded by $t(n) + \mu(n)$.*

We say that *A breaks C w.r.t t with probability p on common input* 1^n if $\Pr[\langle A, C\rangle(1^n) = 1] \geq t(n) + p$.

Note that we can easily model all traditional cryptographic assumptions as a challenger C and a threshold t. For instance, the assumption that a particular function f is (strongly) one-way corresponds to the threshold $t(n) = 0$ and the 2-round challenger C that on input 1^n pick a random input x of length n, sends $f(x)$ to the attacker, and finally outputs 1 iff the attacker returns an inverse to $f(x)$. Decisional assumptions (such as, e.g., the decisional Diffie-Hellman problem, or the assumption that a particular function g is a pseudorandom generator) can also easily be modelled as 2-round challengers but now we have the threshold $t(n) = 1/2$. Also note that the assumption that a protocol (P, V) is witness hiding for a particular instance distribution can be modeled as a polynomial-round challenger C and the threshold $t(n) = 0$: Simply let C be the machine that samples an instance x according to the distribution and next runs the prover algorithm and finally accepts if A outputs a witness for x. In fact, in the same way we can model the assumption that (P, V) is witness hiding under *a priori bounded* sequential composition. This explains why for our results we need to restrict to C having a bounded number of rounds; otherwise, we could simply consider the assumption that (P, V) is witness hiding under unbounded sequential composition.

Related Definitions of Cryptographic Intractability Assumptions As mentioned, the work of Naor [Nao03] is seminal in formalizing cryptographic assumptions as games. His notion of a *falsifiable assumption* considers 2-round challengers; furthermore, he restricts to polynomial-time challengers. Although this suffices for modeling most natural cryptographic assumptions, we also want to be able to capture assumptions of the kind "it is hard to invert f on two random points, even if we get one arbitrary inversion query"; to model this as a game requires having C perform the inversion, which in general may require super-polynomial time.

The formalization of a falsifiable assumption from Gentry and Wichs [GW11] extends Naor's notion to multi-round challengers. Our notion of polynomial-round intractability assumptions is incomparable to the notion used in [GW11]: we require a fixed (i.e., independent of the running-time of the attcker) polynomial upper-bound on the number of communication round by C, whereas [GW11] allow C to communicate in an arbitrary polynomial number of rounds; on the other hand, we allow C to be unbounded, whereas [GW11] restrict C to be polynomial-time (in the length of the messages received from the attacker). For instance, the assumption that GMW-G3C is sequentialy witness hiding, or that Schnorr's protocol is secure under a sequential attack, can trivially be modeled as falsifiable assumptions (according to the notion of [GW11]), so our lower bounds do not extend to these types of assumptions.[12]

Our modeling of a polynomial-round intractability assumption is also similar to Dodis, Oliveira and Pietrazk's [DOP05] notion of a *hard game*. It differs in the following aspects: First, as [DOP05] consider only fully-black-box constructions, their notion of hard games are defined relative to some oracles. Secondly, just as in [Nao03], they restrict to polynomial-time challengers. Finally, [DOP05] only considers the case when the threshold $t(n) = 0$ (but this threshold restriction is removed in the notion of a δ-hard game of Kiltz and Pietrzak [KP09].)

The notion of a *cryptographic game* from Haitner and Holenstein [HH09] is even more similar; it is more general in that it considers unbounded challengers C that can comunicate with the attacker in an unbounded number of rounds (whereas we restrict to only a polynomial number of rounds), but is less general in that it only considers threshold $t(n) = 0$.

Finally, the notion of a *cryptographic primitive* from Rothblum and Vadhan [RV10] is even more general: it considers challengers (which they refer to as testers) that are given oracle access to (instead of simply communicating with) the alleged attacker.

4 The Main Theorem

We want to formalize the statement that there cannot exists a reduction from breaking any standard assumption C to "blatantly" breaking witness hiding of a computationally special-sound protocol—namely, *always* recovering the witness to the statement x after seeing polynomially many sequential proofs of x, no matter what distribution the instances come from. Let us start by formalizing what it means to break witness hiding in this strong way.

Definition 7 (Strongly Breaking Witness Hiding)**.** *Let (P, V) be an argument for the language L with witness relation R_L. We say that A* strongly breaks *$\ell(\cdot)$-sequential witness hiding of (P, V) with respect to R_L if for every $n \in N$, every $x \in L \cap \{0,1\}^n, w \in R_L(x)$, A wins in the following experiment with probability 1: Let $A(x)$ sequentially communicate with $P(x, w)$, $\ell(n)$ times; A is said to win if it outputs a witness w' such that $w' \in R_L(x)$.*

[12] The one-more discrete logarithm assumption (where the attacker may ask an *a priori* unbounded number of inversion queries), on the other hand, is neither a polynomial-round intractability assumption, nor a multi-round falsifiable assumption.

We say that a protocol (P, V) is *weakly $\ell(\cdot)$-sequentially witness hiding* w.r.t R_L if no polynomial-time algorithm A strongly breaks $\ell(\cdot)$-sequential witness hiding of (P, V) w.r.t R_L. Let us now turn to defining what it means to base weak witness hiding on the hardness of a standard assumption C.

Definition 8 (Basing Weak Witness Hiding on C)**.** *We say that R is a black-box reduction for basing weak $\ell(\cdot)$-sequential witness hiding of (P, V) w.r.t R_L on the hardness of C w.r.t threshold $t(\cdot)$ if R is a probabilistic polynomial-time oracle machine, such that for every deterministic machine A that strongly breaks $\ell(\cdot)$-sequential witness hiding of (P, V) with respect to R_L, there exists a polynomial $p(\cdot)$ such that for infinitely many $n \in N$, R^A breaks C w.r.t t with probability $\frac{1}{p(n)}$ on input 1^n.*

We are now ready to state our main theorem.

Theorem 2. *Let (P, V) be a (generalized) computationally-special-sound argument with large challenge space for the language L with a unique witness relation R_L, and let C be an $r(\cdot)$-round assumption, where r is a polynomial. If for every polynomial $\ell(\cdot)$ there exists a black-box reduction R for basing weak $\ell(\cdot)$-sequential witness hiding of (P, V) w.r.t R_L on the hardness of C w.r.t threshold t, then there exists a probabilistic polynomial-time machine B and a polynomial $p'(\cdot)$ such that for infinitely many $n \in N$, B breaks C w.r.t t with probability $\frac{1}{p'(n)}$ on input 1^n.*

That is, if R_L is a unique witness relation, and (P, V) is computationally special-sound with large challenge space, then if we can base weak sequential witness hiding of (P, V) w.r.t R_L on the hardness of C, then C can already be broken in polynomial time.

Remark 3. *(A quantitative version) As becomes evident in the proof, we, in fact, prove the following quantitative version of Theorem 2: there does not exist a reduction from breaking any $r(\cdot)$-round assumption C that cannot already be broken in polynomial-time, to strongly breaking $\ell(\cdot)$-sequential-witness hiding, where $\ell(n) = \omega(n + 2r(n) + 1)$.*[13]

Remark 4. *(On the length of the verifier challenge) Note that the restriction on the length of the verifier challenge in (P, V) is necessary: There exist zero-knowledge protocols that are (computationally) special-sound with verifier challenges of length $O(\log n)$ based on one-way functions: simply consider a parallelized version of the protocol of Blum [Blu86].*[14]

Remark 5. *(On super-polynomial-time reductions) As becomes evident in the proof, the theorem extends also to super-polynomial-time reductions R with running-time T as long as the computational special-soundness of (P, V) holds with respect to $T' = T^{O(\log_n T)}$ time adversaries, and as long as the length of the verifier challenge is $\omega(\log T')$. Both restrictions are necessary for this result:*

- *As noted in [CGGM00, Pas03], if we repeat Blum-HC $\log^2 n$ times in parallel, we get a special-sound proof that is zero-knowledge with a* quasi-polynomial-time simulator*; this, in particular, means that the protocol is sequentially witness hiding (even for unique witness relations) for all distributions that are "hard for quasi-polynomial time", and this can be proven using a* super-polynomial time *black-box reduction. But this protocol has a "short" verifier challenge.*

[13] In fact, it is easy to see that the our proof can be adapted to even handle the case when $\ell(n) = \omega(n^\epsilon + 2r(n)))$, where $\epsilon > 0$.

[14] This protocol only has inverse polynomial (as opposed to negligible) soundness error. If this bothers the reader, simply consider using Blum-HC for proving knowledge of a witness for the "trivial" language $L = \{0, 1\}^*$. For instance, let f be a one-way permutation, let $y \in R_L(x)$ if and only if $f(y) = x$, and let L be the language characterized by R_L. Blum-HC repeated in parallel $\log n$ times can be used to prove L w.r.t. R_L with 0 soundness error.

- *Additionally, [Pas03] shows (assuming the existence of one-way permutations), the existence of a four-round computationally special-sound argument that is zero-knowledge with a quasi-polynomial-time simulator and has an n-bit long verifier challenge; the protocol, however, is only sound for polynomial-time algorithms.*

5 Proof of the Main Theorem

Let (P, V) be a k-round (generalized) computationally-special-sound argument with large challenge space for the language L with a unique witness relation R_L, let C be an $r(\cdot)$-round assumption, and let $t(\cdot)$ be a threshold function. For ease of presentation, we start by considering the case when C is polynomial-time computable (most common assumptions used in cryptography actually fall into this case); at the end of the proof, we explain how to modify the proof to work also when C is not efficient.

Let $\ell(n) = \omega(n + r(n) + 1)$. Assume that there exists a black-box reduction R for basing weak $\ell(\cdot)$-sequential witness hiding of (P, V) w.r.t R_L on the hardness of C. That is, for every A that strongly breaks $\ell(\cdot)$-sequential witness hiding of (P, V) w.r.t R_L, there exists a polynomial $p(\cdot)$ such that for infinitely many $n \in N$, R^A breaks C w.r.t t on common input 1^n with probability $\frac{1}{p(n)}$. We show the existence of a probabilistic polynomial-time machine B that directly breaks C without the help of A; that is, there exists a polynomial $p'(\cdot)$ such that for infinitely many $n \in N$, B breaks C w.r.t t on common input 1^n with probability $\frac{1}{p'(n)}$.

The machine B will use the reduction R as a black-box, and emulate a *particular* oracle A for R. At first sight this seems impossible: the reduction R is only useful if we run it on an oracle A that strongly breaks witness hiding of (P, V), but doing this (*a priori*) requires super-polynomial time. We here use the fact that (P, V) is computationally special-sound. Recall that A only needs to return a witness to R after R has provided it with (polynomially-many sequentially-ordered) accepting proofs of the same statement x. Now, intuitively, for R to succeed in this task, R must already "know" a witness $w \in R_L(x)$; the idea is to "extract" out this witness from R (this is similar to [HRS09]) by "rewinding" it. The problem is that R might already be rewinding its oracle A, so it is no longer clear how to perform such an extraction. More specifically, we need to deal with the following three issues:[15]

- R might "intertwine" its proof queries to A for many different statements x; in particular, it might "nest" these proofs. This has the effect that the naive way of extracting the witnesses from R leads to an exponential blow-up in running-time (c.f. [DNS04]). We here use the fact that R needs to provide A with *polynomially many* sequential proofs of the statement x, before A needs to return a witness for x; this makes it possible for B to find *one* of these sequential proofs from which extraction can be done, without increasing the running-time by too much. We here rely on techniques from [RK99, PV08, DGS09, CLP10] originally designed for the analysis of concurrent zero-knowledge protocols [DNS04].

- When performing extraction we need to be careful not to affect the external interaction with C. More precisely, when rewinding R, R might send a new message in the external interaction with C, and might only send a query to its oracle after it gets back an answer from C. Again, since R needs to provide polynomially many proofs of x to A before A will return a witness for

[15] The reason that [HRS09] do not have to deal with these problems is that they either consider restricted black-box protocols where extraction can be performed "straight-line" (i.e., without rewinding R), or restricted reductions for which these issues do not arise).

x, we can ensure that there exists at least one proof that can be rewound (without increasing the run-time too much) and at the same time without resending any messages in the external interaction with C.

- Using the above approach, we show how B can obtain any polynomial number of accepting proof transcripts of x where the first $k-2$ messages are the same, but (with high probability) the $k-1$'st messages are all different, before it has to provide R with a witness for x. Thus, intuitively, by computational special-soundness of (P, V), B can run the special-soundness extractor on these accepting transcript and get a valid witness. The problem is that computational special-soundness only requires extraction to work when the first $k-2$ messages have been generated in an interaction with a "standard" (non-rewinding) polynomial-time prover, but in our context they have been generated in a conversation with R (acting as a prover), and R has the possibility of rewinding its oracle. We rely on the fact that (P, V) is public-coin (or that the verifier's next message function only depends on the public transcription) and only has a constant number of rounds (and techniques similar to Goldreich-Krawzyk [GK96] and their use in [Pas06]) to show that despite the fact that R is rewinding its oracle, it can still not generate a transcript for which special-soundness extraction fails.

We proceed to a description of the machine B that breaks C w.r.t t with inverse polynomial probability for infinitely many $n \in N$. For simplicity, we provide the description of a machine B that runs in *expected* polynomial time—by the Markov inequality, we can truncate the execution of this machine while still maintaining an inverse polynomial success probability for infinitely many n. To simplify notation we consider the case when (P, V) is public-coin and (P, V) thus satisfies the standard (as opposed to generalized) notion of computational special-soundness. At the end of the proof we explain how the proof extends to the generalized notion. Let us start by fixing some notation:

- Let $m = m(n)$ denote the number of accepting transcripts required by the computation special-soundness property of (P, V) on inputs of length n;
- Let $r = r(n)$ denote the number of communication rounds by C on input 1^n;
- Let $M = M(n)$ denote the maximum number of queries to its oracle by R on input 1^n;
- Let $\ell = \ell(n) = \omega(n + r + 1)$ denote the number of sequential repetitions used for strongly breaking witness hiding of (P, V).

Towards the construction of B, we also make two (standard) simplifying assumptions about R:

- R never asks the same query twice to its oracle;
- Whenever R sends a query q to its oracle, it has previously queried the oracle on all partial transcripts in q.

Both of these assumptions are without loss of generality; we can always modify R, at the cost of only a polynomial blow-up in running-time, to satisfy these two conditions.

Let f be a function, and let A^f be a machine defined as follows. A^f, on input 1^n, acts as the honest verifier V in ℓ sequential interactions with a prover for (P, V), but with the exception that instead of generating truly random messages (as V would have done), A^f computes its next message by applying f to the current partial transcript (and appropriately truncating the output of f to be of the right length); finally, after the $\ell(n)$ interactions, if all proofs are accepting, A^f

uses brute-force to extract a witness (if one exists) and outputs it (and $\perp$ if no witness exists). We next consider a random oracle: Let RO be a random variable uniformly distributed over functions $\{0,1\}^* \to \{0,1\}^\infty$. By definition we have that for each n, with probability 1 over the choice of RO, A^{RO} strongly breaks $\ell(\cdot)$-sequential witness hiding of (P,V) with respect to R_L.[16] Thus, by the fact that R is a good reduction, we have that there exists some polynomial $p(\cdot)$ such that

$$\Pr\left[\langle R^{A^{RO}}, C\rangle(1^n) = 1\right] \geq t(n) + \frac{1}{p(n)}. \tag{1}$$

Remark 6. *Before continuing, let us briefly explain the reasons for using the random oracle RO. There are two quite different reasons for this. First, it will allow us to "rewind" R while guaranteeing that R does not "notice" that it is being rewound. (Assume instead that we had let A use the honest verifier strategy V. This strategy sends the* same *verifier challenge no matter what $(k-2)$-round message R sends it. Since R might rewind its oracle, it might thus send two different $(k-2)$ round messages for which it expects to hear back the same verifier challenge from A. In such a situation it becomes hard to "rewind" R feeding it new verifier challenges.) Secondly (and similarly to [Pas06]), this will ensure us that R cannot "cheat" in the proofs it provides to its oracle by rewinding it.*

Let us return to the construction of B. Roughly speaking, on input 1^n, the goal of the machine B will be to run R while *efficiently* emulating the role of A^{RO}. As previously mentioned, this will be possible by appropriately rewinding R to extract witnesses w for statements x proved by R to its oracle. More precisely, B will attempt to "rewind" the "verifier-challenge" (i.e., the $k-1$'st message) in some of the proofs. We refer to the pair of a verifier-challenge, and the prover answer (i.e., the $k-1$'st, and the k'th messages) as a slot. We say that a slot "opens" when R receives a verifier challenge from its oracle, and that the same slot slot "closes", when R sends its oracle the answer to the verifier challenge. Note that by our simplifying assumptions on R, a slot can never close without opening. However, also note that R might send a lot of other queries to its oracle between the time when a slot opens and closes. So, when attempting to rewind a slot, we need to be careful not to "redo" too much work. Formally, the opening of a slot is a partial view v of R immediately after which the slot opens; we may now identify a slot s by the view corresponding to its opening. Analogously, the closing of a slot s is a partial view v immediately after which s closes.

Let us turn to providing a high-level description of B. On a high-level, B internally incorporates R, internally emulates A^{RO} for R, but externally sends all communication between R and C (i.e., whenever R wants to send a message to C, B externally forwards it, and whenever B receives an external message from C, it directly forwards it to R as if it came from its external interaction with C). R emulates A^{RO} for R by following exactly the same instruction as A^{RO} until a slot s "closes" and the following two properties hold:

- Between the time when the slot s opened, and the time that it closed, R did not send (and thus not receive) any external messages (to or from C).
- Between the time when the slot s opened, and the time that it closed, the number of other slots that opened is "small", where "small" will be defined shortly.

Whenever such a slot s closes, B rewinds R back until the point where s opened, and instead sends R a new random verifier challenge (by our simplifying assumption that R never asks its oracle the same query twice, this can never create any inconsistencies), and continues the emulation of A^{RO}

[16] We here rely on the fact that (P,V) has perfect completeness, otherwise, we could only claim that A^{RO} breaks sequential witness hiding with overwhelming probability for a random RO.

as before, but with the exception that if R opens too many new slots, the rewinding is cancelled. B continues rewinding R until it gets m accepting closings of slot s; intuitively, this should allow B to use the special-soundness extractor to recover a witness for the statement proved. We remark that in contrast to the simulation technique of [RK99], we do *not* decide what slot to rewind based on the *number of executions* that start within the slot, but rather, following [CLP10], decide what slot to rewind, based on the *number of slots* within the slot.

More precisely, the emulation is defined recursively in the following manner. Given the view τ of R (w.l.o.g., this includes all the messages sent and received by R in the external interaction with C, and all the messages sent and received between R and its oracle), we call a prefix ρ of τ d-good if 1) R makes no external queries in τ after ρ, and 2) the number of slots that open in τ after ρ is at most $\frac{M}{n^d}$ (recall that M is a (polynomial) upper bound on the number of oracle queries made by R). Given a partial view τ after which we have the closing of a slot s, we say that the slot is d-good in τ if s is a d-good prefix of τ.

Now, on recursive level $d \geq 0$, starting from a view $\mathcal{V}$, B emulates A^{RO} for R, until a slot s that opened inside the view $\mathcal{V}$ closes *and* the slot is $d+1$-good for the current view v; whenever this happens, it rewinds R back to the point when s opened, and invokes itself recursively at level $d+1$. It continues rewinding until it gets m accepting closings of slot s, and then applies the special soundness extractor X on these m transcripts; if the extractor outputs a valid witness w (to the statement x currently proved by R), the pair (x, w) is stored. Furthermore, at each recursive level $d \geq 1$ (i.e., on all recursive levels except the first one), if $\mathcal{V}$ is not a d-good prefix of the current view v (i.e., if the number of new openings of slots exceeds $\frac{M}{n^d}$, or if R wants to send an external message) the recursive procedure aborts (returning to the earlier recursive call); this ensures that all rewindings are "cut-off" if R attempts to send external messages, or opens more slots, in the rewinding. Finally, whenever R is expecting to hear back a witness w for a statement x from A^{RO}, B checks whether such a witness w has been extracted; if so, it simply feeds it to R, and otherwise B halts outputting fail.

We proceed to a formal description of the procedure B, and analyze its running-time and success probability. B invokes the procedure EXT, described in Figure 1, on input $(1^n, 0, 1^n)$.

Let us start by showing that the running time of B is bounded in expectation.

Proposition 1. *There exists some polynomial $t(\cdot)$ such that $B(1^n)$ runs in expected time bounded by $t(n)$ for all $n \in N$.*

Proof. To simplify the analysis, let us consider a slight variant of B that never gets "stuck"—instead of ever halting outputting fail, let us assume that B is magically fed a valid witness w if it ever is required to provide a witness for a statement x for which it has not recovered a witness. Clearly this change can only increase B's running-time.

Note that the recursive level is bounded by $c = \log_n M$, which is a constant (since M is polynomial in n). Secondly, at each recursive level d, there are at most M possible points from which we can rewind. As we shall argue, from each of these points (i.e., partial views), the expected number of rewindings is bounded by m. Recall that in the execution of $\mathsf{EXT}(1^n, d, \mathcal{V})$, B only starts "rewinding" a slot s if 1) the slot s opened in the view $\mathcal{V}$, 2) the slot s closes in the current view v, and 3) the slot s is $d+1$-good for v. Furthermore, in each of the rewindings the simulated view of the adversary on the recursive level $d+1$ (i.e., in the execution of $\mathsf{EXT}(1^n, d+1, s)$) is *identically distributed* to its view in the execution on level d; note that we here rely on the unique witness requirement and the assumption that B never gets "stuck". Thus, the probability that the slot s becomes $(d+1)$-good for some view v' in the recursive call on level $d+1$ (i.e., that the rewinding is

PROCEDURE $\mathsf{EXT}(1^n, d, \mathcal{V})$:

On input the recursive level d and the partial view $\mathcal{V}$ of R, proceeds as follows. Let $v = \mathcal{V}$. Repeat the following:

- If $d > 0$ and v is the closing of the slot opened at $\mathcal{V}$, return v.
- If $d > 0$ and the partial view v is not d-good, return $\perp$.
- If $d = 0$ and R attempts to externally forward a message, externally forward it, and feed R the answer received back; let v denote the updated view of R, and continue.
- If v is the closing of a slot s that opened after $\mathcal{V}$ and that is $d+1$-good for v: let $i = 0$; repeat the following until $i = m(n)$:
  - Let $v' = \mathsf{EXT}(1^n, d+1, s)$.
  - If $v' \neq \perp$, let $i = i + 1$, $v_i = v'$.

  Finally, apply the special soundness extractor X on the transcripts of (P, V) corresponding to the $m(n)$ views $v_1, \ldots, v_{m(n)}$. If X succeeds in finding a witness w for the statement x proved, store (x, w).
- If R is expecting to hear back a witness for the statement x, check if a pair (x, w) has been stored. If so, feed w to R and update v accordingly and continue; otherwise halt outputting fail.
- If R is expecting any other message of length $l(n)$; simply feed R a random message of length $l(n)$, update v accordingly and continue.

Figure 1: Pseudo-code for the recursive simulation strategy by B.

successful) is at least the probability that the slot was $d+1$-good on level d.[17] Since B rewinds the slot until it gets m accepting closings, the expected number of rewindings from each partial view is thus m.

So, at each recursive level—i.e., in each invocation of $\mathsf{EXT}(1^n, d, \mathcal{V})$—the expected number of rewindings—i.e., recursive invocations of $\mathsf{EXT}(1^m, d+1, \mathcal{V}')$ for some view $\mathcal{V}'$—is bounded by $O(Mm)$. It follows using a standard induction that for each recursive level $d \leq c$, the total number of messages sent by $\mathsf{EXT}(1^n, d, \mathcal{V})$ (and its recursive sub-routine calls) to R is bounded by $(O(Mm))^{c+1-d}$. □

Let us now turn to analyze the success probability of B. We show that there exists a negligible function $\mu(\cdot)$ such that

$$\Pr\left[\langle B, C\rangle(1^n)\right] \geq \Pr\left[\langle R^{A^{RO}}, C\rangle(1^n)\right] - \mu(n) \tag{2}$$

Towards this, we consider a hybrid machine $\tilde{B}$. $\tilde{B}$ is a variant of B that proceeds exactly as B, but always recovers the witness requested by R using brute force; it thus never halts outputting

[17] The probability might actually be larger, since on level d we might also abort if the current view is no longer d-good.

fail. Note that all recursive calls at level $d > 0$ are irrelevant for the external output of $\tilde{B}$. And, on level $d = 0$, $\tilde{B}$ perfectly emulates the role of A^{RO} for R. Thus, $\tilde{B}$'s success probability in its external interaction with C is identical to $R^{A^{RO}}$'s success probability. That is, we have:

$$\Pr\left[\langle \tilde{B}, C\rangle(1^n) = 1\right] = \Pr\left[\langle R^{A^{RO}}, C\rangle(1^n) = 1\right] \tag{3}$$

Note that, by the unique witness requirement, unless B halts outputting fail, it proceeds identically to $\tilde{B}$. We now show that B outputs fail with negligible probability, which by equation 3 concludes equation 2.

Proposition 2. *There exists a negligible function μ such that for all $n \in N$, the probability that B outputs* fail *in an interaction with C on common input 1^n is bounded by $\mu(n)$.*

Proof. Assume for contradiction that there exists some polynomial $p(\cdot)$ such that B outputs fail with probability $\frac{1}{p(n)}$ for infinitely many n. Since by Proposition 1, the expected running time of B is polynomially bounded, it follows by the Markov inequality that there exists some polynomial $t(\cdot)$ such that the probability that B outputs fail while taking less than $t(n)$ steps is at least $\frac{1}{2p(n)}$. Thus, from this point on, it suffices to consider a truncated version B' of B that runs in strict time $t(\cdot)$, and outputs fail with non-negligible probability. Let us consider the following two events:

- Let E_1 denote the event that B' is required to provide R with the witness for a statement x, without having previously "rewound" at least one slot for a proof of x.
- Let E_2 denote the event that the special soundness extractor X fails to output a valid witness in the execution by B'.

Note that if neither of E_1 of E_2 happens, there always exists some slot that is rewound for which the special-soundness extractor succeeds, which means that B' must have stored an instance-witness pair (x, w) before it is ever required to provide R with a witness for the statement x, and thus B' can never fail.

We show below that the probability that either of these events happens is negligible.

Claim 1. *There exists some negligible function $\mu(\cdot)$, such that the probability that E_1 happens in an execution between B' and C on common input 1^n is bounded by $\mu(n)$.*

Proof. Note that by construction we have that for each instance x, B' must have encountered $\ell(n) = \omega(n + r + 1)$ slots for x before needing to return a witness for x. Since the recursive depth of B' is some constant c, there must thus exist some recursive level d with at least ℓ/c slots for x; for sufficiently big n, $\ell/c \geq n + r + 1$. But since the total number of slots opening on level d is bounded by $\frac{M}{n^d}$ (for $d = 0$, this follows by the definition of M; and for $d > 0$, this follows since by definition of B the simulation at recursive level d is cancelled if more than $\frac{M}{n^d}$ slots open), there exists at least $r + 1$ slots that contains less than $\frac{M}{n^{d+1}}$ slots, and as a consequence at least 1 slot that contains less than $\frac{M}{n^{d+1}}$ slots and also does not contain any messages from the external interaction with C (recall that C has r communication rounds); this slot is thus $d + 1$-good and will consequently be rewound. □

Claim 2. *There exists some negligible function $\mu(\cdot)$, such that the probability that E_2 happens in an execution between B' and C on common input 1^n is bounded by $\mu(n)$.*

Proof. Intuitively, the probability that event E_2 happens should be negligible by the computational special-soundness property of (P, V) (and the fact that the verifier challenge is of length $\omega(\log n)$ which means that except with negligible probability we won't see the same verifier challenge twice). However, as R is rewinding its oracle, we are applying the special-soundness extractor on transcripts that not necessarily are uniformly generated (as required by the definition of computational special-soundness). Nonetheless, we show that with inverse polynomial probability we are in fact applying the special-soundness extractor to transcripts that are uniformly generated, thus reaching a contradiction. Roughly speaking, this follows from the fact that (P, V) only has a constant number of rounds and is public-coin, so (just as in the proof of [GK96, Pas06]) we can with inverse polynomial probability "guess" on which of the "rewindings" special-soundness will be broken, and thus turn the rewinding reduction R into a stand-alone prover P^* that breaks computational special-soundness. We use the assumption that C is efficiently computable to ensure that P^* is efficient as well.

We proceed to a formal proof. Assume for contradiction that there exists some polynomial $g(\cdot)$ such that the special-soundness extraction fails with probability $\frac{1}{g(n)}$ for infinitely many n (in the execution of C and B' on common input 1^n). We construct a polynomial time machine P^* that attempts to break the computational special-soundness property of (P, V). For simplicity, we will allow P^* to probabilistically pick the statement x to prove; using a standard averaging argument, we can then fix an appropriate part of P^*'s random tape to ensure that the proof statement is always the same, while maintaining the same success probability.

P^* will internally emulate an execution between C and B' on input 1^n, but will pick one *random* (P, V) proof provided by R and forward it externally—that is, it externally forwards the messages that R tries to send to its oracle, and uses the external verifier's answers as the answers to those oracle queries). More precisely, P^* randomly picks $i_1, i_2, .., i_q \in [1, .., M]$, where q is the number of prover rounds in (P, V). P^* then internally emulates an execution between C and B' on input 1^n with the following differences:

- If the i_1'st query by R to its oracle is of "length 1"—that is, if it is of the form $x||m_1$ where x is an instance and m_1 is a first round message of (P, V)—proceed as follows. Select the statement x to prove externally, and forward externally the message m_1. Upon receiving an answer c_1 from the external verifier, let B' use c_1 as its answer to R.

- If furthermore the i_2'nd query is of "length 2"—that is, if it consists of a second round query $(x'||m_1', m_2)$ which additionally is consistent with the first round query $x||m_1$ (i.e., $x' = x$ and $m_1 = m_1'$)—externally forward m_2, and let B' use the answer received back as its answer to R.

- Continue in the same manner for all i_j, $j \leq q$.

Now, consider applying the canonical extraction procedure to P^*; that is, emulate the honest verifier V for P^*, next rewind P^* feeding it new uniformly chosen verifier challenges until we get $m(n)$ accepting transcripts, or until $t(n)$ rewindings have been performed, and next apply the special-soundness extractor on these transcripts.

Since B' feeds R messages according to the same distribution as the canonical extraction procedure, the fact that the canonical extractor performs at least as many rewindings as B' (recall that B''s running-time is bounded by $t(n)$), the number of rounds in (P, V) is constant, and R makes at most M oracle calls, it follows that the special-soundness extractor fails with probability at most $\frac{1}{g(n)M^{O(1)}}$ when applying the canonical extraction procedure to P^*. This still doesn't directly contradict the computational special-soundness of (P, V)—it could be the case that we have

got two transcripts with the same verifier challenge. But, since the canonical extraction procedure picks the verifier challenges at random, and the length of the verifier challenge is $\omega(\log n)$, the probability that any given pair of challenges collide is $2^{-\omega(\log n)}$. So, by the union bound (and the fact that we perform at most a polynomial number of rewindings), the probability that any two challenges collide is bounded by $\frac{poly(n)}{2^{-\omega(\log n)}}$, which is negligible. We thus have that computational special-soundness of (P, V) can be broken with probability $\frac{1}{g(n)M^{O(1)}} - \nu(n)$ where $\nu(\cdot)$ is a negligible function; this is a contradiction. □

Remark 7. *For clarity, let us highlight where in the proof of Claim 2 the public-coin and constant-round properties of (P, V) are used: The public-coin property is used only to ensure that we can perfectly generate the verifier's next message given only the public transcript of the interaction—this is needed in order to emulate rewindings of the external execution. (This is why we can generalize the argument also to generalized computationally special-sound protocols).*

The constant-round restriction comes from the fact that the success probability of our reduction degrades exponentially with the number of communication rounds of (P, V).

Dealing with unbounded C The problem with unbounded machines C is that in the proof of Claim 2, P^* can no longer *efficiently* emulate the interaction between C and B'. To handle this problem, we show that by slightly increasing $\ell(n)$, we can ensure that the special-soundness extractor needs to fail more than $r + 1$ times for B' to fail; this means that for one of these invocations of the special-soundness extractor, the transcripts of (P, V) were generated without the help of C. More precisely, let us consider the following two modified events:

- Let E_1 denote the event that B' is required to provide R with the witness for a statement x, without having previously "rewound" $r(n) + 1$ *sequentially ordered slots* for proofs of x.
- Let E_2 denote the event that the special soundness extractor X fails to output a valid witness in the execution by B' for *more than* $r(n)$ *sequentially ordered slots* for some statement x.

Note that if neither of E_1 or E_2 happens, then for every instance x, there always exists some slot that is rewound for which the special-soundness extractor succeeds, and thus as before B' can never fail. We now just need to verify that Claim 1 and Claim 2 holds with respect to these new definitions of the events E_1, E_2. Claim 1 follows identically as before if we let $\ell = \omega(n + 2r + 1)$ (instead of $\omega(n + r + 1)$.) To show Claim 2, we note that if the special soundness extractor fails for $r + 1$ sequentially ordered slots with probability $\frac{1}{g(n)}$, there must exists some partial view τ for B' such that conditioned on τ, with probability $\frac{1}{g(n)}$, the special-soundness extractor fails on some execution of (P, V) that is not yet fixed in τ and without there being any external messages exchanged after τ and the time when the extrator fails (recall that the interaction with C has at most r rounds). We can now continue in exactly the same way as in the proof of Claim 2.

Dealing with generalized computationally special-sound protocols (P, V). In case (P, V) only satisfies the generalized notion of computational special-soundness from Remark 1, we need to slightly adapt the oracle A^f and the algorithm B in the following ways:

- Instead of simply using the function f to compute its next message, A^f now uses the function f to generate a string r of appropriate length, and next uses this string r as "randomness" together with the public transcript to compute the verifier's next message function (just as V would have) in order to generate its next message.

- Instead of simply emulating the verifier messages for R by sending it truly random strings, B emulates them using the same approach as V—it picks a truly random string r and computes the next message as a function of the public transcript and r.
- Instead of feeding the extractor X only the public transcripts, B now feeds X the $m(n)$ accepting views of V.

The rest of the proof remains unchanged: As noted in Remark 7, the fact that we can perfectly generate the verifier's next message function given a partial public transcript suffices for proving Claim 2, and the public-coin property is not used in other parts of the proof. □

6 Security of Identification Schemes with Unique Secret-Keys

In this section we use our main theorem to rule out the possibility of basing the security of certain types of identification schemes with "unique secret keys" (i.e., each public key is associated with a single secret key) on standard assumptions using Turing reductions. As a corollary we establish that the security of Schnorr's identification scheme cannot be based on any standard assumption using a Turing reduction.

Let us start by recalling the definition of an identification scheme [FFS87]. Given an algorithm K, interactive algorithms P, V, A, a polynomial p and $n \in N$, let $\mathsf{WIN}^{(P,V)}(K, A, p, n)$ denote the output of the following experiment: Sample (pk, sk) as the output of $K(1^n)$, let $A(pk)$ communicate with $P(sk, pk)$ in $p(n)$ sequential executions, and next communicate with $V(pk)$ once; output 1 if V accepts.

Definition 9 (Identification Scheme)**.** *An* identification scheme *is a triple (K, P, V) where K is probabilistic polynomial-time algorithm, and P, V are probabilistic polynomial-time interactive algorithms, and for every $n \in N$, every $(pk, sk) \in K(1^n)$,*

$$\Pr\Big[\langle P(pk, sk), V\rangle(pk) = 1\Big] = 1$$

Definition 10 (Secure Identification Scheme)**.** *We say that an identification scheme (K, P, V) is* secure *if for every probabilistic polynomial-time algorithm A, there exists some negligible function μ such that for all polynomial p, and every $n \in N$,*

$$\Pr\Big[\mathsf{WIN}^{(P,V)}(K, A, p, n) = 1\Big] \leq \mu(n)$$

We will consider identification schemes (K, P, V) of the following type:

- (P, V) is a (generalized) computationally special-sound argument with large challenge space for the language L with unique witness relation R_L;
- $(pk, sk) \in R_L$ if $(pk, sk) \in K(1^n)$ for some $n \in N$.

We refer to such identification schemes as *unique identification schemes*.

We can use our main theorem to show that Turing reductions cannot be used to prove the security of any unique identification scheme based on standard assumptions.

Definition 11 (Strongly Breaking Security of Identification Schemes)**.** *Let be (K, P, V) an identification scheme. We say that A* strongly breaks *$\ell(\cdot)$-sequential security of (K, P, V) if for every $n \in N$,* $\Pr\Big[\mathsf{WIN}^{(P,V)}(K, A, \ell, n) = 1\Big] = 1$.

Definition 12 (Basing Weak Security of Identification Schemes on C)**.** *We say that R is a black-box reduction for basing weak $\ell(\cdot)$-sequential security of (K, P, V) on the hardness of C w.r.t. threshold $t(\cdot)$ if R is a probabilistic polynomial-time oracle machine, such that for every deterministic machine A that strongly breaks $\ell(\cdot)$-sequential security of (K, P, V), there exists a polynomial $p(\cdot)$ such that for infinitely many $n \in N$, R^A breaks C w.r.t t with probability $\frac{1}{p(n)}$ on input 1^n.*

Theorem 3. *Let (K, P, V) be a unique identification scheme and let C be an $r(\cdot)$-round assumption, where r is a polynomial. If for every polynomial $\ell(\cdot)$ there exists a black-box reduction R for basing weak $\ell(\cdot)$-sequential security of (K, P, V) on the hardness of C w.r.t. threshold t, then there exists a probabilistic polynomial-time machine B and a polynomial $p'(\cdot)$ such that for infinitely many $n \in N$, B breaks C w.r.t t with probability $\frac{1}{p'(n)}$ on input 1^n.*

Proof. Consider any unique identification scheme (K, P, V) and let R_L be the unique witness relation associated with (P, V). Note that any attacker A that strongly breaks $\ell(\cdot)$-sequential witness hiding of (P, V) w.r.t. R_L can be easily transformed into an attacker A' that strongly breaks $\ell(\cdot)$ security of (K, P, V): Recall that any attacker A that strongly breaks $\ell(\cdot)$-sequential witness hiding of (P, V) recovers a witness $sk \in R_L(pk)$ for any statement pk that it hears $\ell(|pk|)$ sequential accepting proofs of, with probability 1; we can next use the prover algorithm P on input (pk, sk) to convince $V(pk)$ with probability 1 (this follows by the validity of (K, P, V) as an identification scheme). Thus, any black-box reduction R for basing weak $\ell(\cdot)$-sequential security of (K, P, V) on the hardness of C w.r.t. t can be turned into a black-box reduction for basing weak $\ell(\cdot)$-sequential witness hiding of (P, V) w.r.t. R_L on the hardness of C w.r.t. t. The theorem next follows by applying our main theorem, since by definition (P, V) is computationally-special sound with large challenge space and R_L is a unique witness relation. □

Remark 8. *(On super-polynomial-time reductions) We note that both the upper and lower bounds for super-polynomial-time reduction w.r.t witness hiding from Remark 5 directly translate to upper and lower bounds also with respect to unique identification schemes.*

Example 1: Applications to Schnorr's identification scheme: Schnorr's scheme [Sch91] is based on the discrete logarithm problem in prime order groups: A *discrete logarithm parameter generator* $D(1^n)$ outputs $1^n, p, q, g$ where $2^{n-1} \leq p \leq 2^n$ is a prime, q is an $l(n) = \omega(\log n)$ bit prime such that $q|p-1$, and g is a generator of a subgroup G_q of $\mathbb{Z}_p^*$ of order q. A description of the protocol is found in Figure 2.

Consider the unique witness relation $x \in R_L(1^n, p, q, g, X)$ if and only if $2^{n-1} \leq p \leq 2^n$ is a prime, q is a length $l(n)$ prime such that $q|p-1$, g is a generator of a subgroup G_q of $\mathbb{Z}_p^*$ of order q, $X \in G_q$, and $X = g^x \mod p$; let L be the language characterized by R_L. Note that the language L is polynomial-time decidable: Given $(1^n, p, q, g, X)$ we can efficiently check that p, q are appropriate primes, that g is a generator of a subgroup G_q of order q, and that X is an element of G_q; the last two checks amount to checking that both g, X have order q in $\mathbb{Z}_p^*$ which can be done efficiently since q is a prime.[18]

It is easy to see that (P, V) is complete for L; furthermore, it is well-known that for "valid" public-keys $pk \in L$, (P, V) is special-sound (i.e., special-soundness holds for any instance pk that is a valid public key). However, for *invalid* public-keys, special-soundness might no longer hold [Bur90]. But we can easily modify (P, V) into a new scheme (P, V') such that special-soundness holds on all inputs, yet if (P, V) is a secure identification scheme, then so it (P, V'). $V'(pk)$ proceeds just as

[18] To check whether an element y has a prime order q in $\mathbb{Z}_p^*$, simply check whether $y^q = 1 \mod p$; furthermore, since $\mathbb{Z}_p^*$ is cyclic, there is a unique subgroup G_q of order q.

SCHNORR'S IDENTIFICATION SCHEME (K, P, V):

- On input 1^n, the key generation algorithm $K(1^n)$ outputs the pair (pk, sk), where $pk = (1^n, p, q, g, X)$ is the output of a discrete logarithm generator on input 1^n, $X = g^x \mod p$, $x \in \mathbb{Z}_q$, and $sk = x$.
- The protocol (P, V) proceeds as follows.
  - P on inputs $(pk = (1^n, p, q, g, X), sk = x)$, picks a random $u \in \mathbb{Z}_q$ and send $a = g^u \mod p$ to V.
  - V on input $pk = (1^n, p, q, g, X)$, picks a random $c \in \mathbb{Z}_q$ and sends it to the prover.
  - P computes $r = u + cx \mod q$ and sends it to V.
  - V accepts if $g^r = aX^c \mod p$.

Figure 2: Schnorr's identification scheme [Sch91]

$V(pk)$ except that it first checks that $pk \in L$; if the check fails, V' simply aborts, and otherwise it continues just as V.[19] It is easy to see that if (K, P, V) is a secure identification scheme, then so is (K, P, V') (and in particular, any attacker breaking (K, P, V') can be used in a black-box way to break (K, P, V)). Furthermore, we now have that (P, V') is special-sound for L and the witness relation R_L, and has a large challenge space, and so (K, P, V') is a unique identification scheme. It follows from Theorem 3 that the security of (K, P, V'), and thus also (K, P, V), cannot be based on any standard assumption using a Turing reduction.

Example 2: Identification through Parallel GMW Let f be a one-to-one one-way function, and let $K(1^n)$ sample a random $x \in \{0,1\}^n$ and output $(f(x), x)$. Let $x \in R_L(y)$ if and only if $y = f(x)$, and let L be the language characterized by R_L. Let (P, V) be any computationally special-sound argument with large challenge space for L and the unique witness relations R_L—for instance, consider parallelized versions of GMW-G3C, or Blum-HC, implemented with either statistically-binding or statistically-hiding commitments. (K, P, V) is a unique identification scheme, and thus by Theorem 3, its security cannot be based on any standard assumption using Turing reductions.

Example 3: Identification using two secret keys As shown by Feige and Shamir [FS90], if we slightly change the protocol from Example 2, we can get a secure identification scheme: Let $K(1^n)$ sample random $x_1, x_2 \in \{0,1\}^n$ and output $(f(x_1), f(x_2), x_1)$; let $x \in R_L(y_1, y_2)$ if and only if either $y_1 = f(x)$ or $y_2 = f(x)$, and let (P, V) be a parallelized version of GMW-G3C or Blum-HC proving L with respect to the witness relation R_L.

[19] In fact, implementations of Schnorr's protocol often include this check, or require that such a check is performed when users are registering their public-key.

7 Security against Adaptive Selective Decommitment

In this section we investigate commitment schemes secure with respect to *adaptive selective decommitment*. Our definition follows that of Dwork, Naor, Reingold and Stockmeyer (DNRS) [DNRS03], but we allow the adversary to *adaptively* decide what commitments to see openings of; DNRS considered such a notion in Remark 7.1 in [DNRS03]. We next remark that if implementing the commitment scheme in GMW-G3C with commitments secure under adaptive selective decommitment, then the resulting protocol is witness hiding under sequential composition for unique witness relations. Finally, we use our main theorem to prove limitations of basing commitments secure against adaptive selective decommitment on standard assumptions.

Let us first briefly recall the notion of a commitment scheme.

7.1 Commitments

Commitment protocols allow a *sender* to commit itself to a value while keeping it secret from the *receiver*; this property is called *hiding*. At a later time, the commitment can only be opened to a single value as determined during the commitment protocol; this property is called *binding*. Commitment schemes come in two different flavors, statistically binding and statistically hiding; below we sketch the properties of a statistically binding commitment; full definitions can be found in [Gol01]. In statistically binding commitments, the binding property holds against unbounded adversaries, while the hiding property only holds against computationally bounded (non-uniform) adversaries. The statistical-binding property asserts that, with overwhelming probability over the randomness of the receiver, the transcript of the interaction fully determines the value committed to by the sender. The computational-hiding property guarantees that commitments to any two different values are computationally indistinguishable.

Non-interactive statistically-binding commitment schemes can be constructed using any one-to-one one-way function (see Section 4.4.1 of [Gol01]). Allowing some minimal interaction (in which the receiver first sends a single random initialization message), statistically-binding commitment schemes can be obtained from any one-way function [Nao91, HILL99].

As in [DNRS03], we here focus on non-interactive commitments $\mathcal{C}(\cdot,\cdot)$—where the first input is the value v and the second input is the randomness used by the committer—but just as in their treatment, our treatment directly extends to two-round commitments (i.e., "families" of non-interactive commitments). For completeness, let us provide the definition of statsitically-binding and computationally hiding non-interactive commitments.

Definition 13. *We say that $\mathcal{C}(\cdot,\cdot)$ is a* non-interactive commitment scheme *if $\mathcal{C}$ is a deterministic algorithm whose running-time is polynomial in the length of the first input, and the following conditions hold.*

- *(Validity) There exists a polynomial $p(\cdot)$ such that for any $n \in N$, any $v \in \{0,1\}^n, r \in \{0,1\}^{p(n)}$, $\mathcal{C}(v,r) \neq \perp$.*
- *(Binding) For any $v, v', r, r' \in \{0,1\}^*$ such that $v \neq v'$, if $c = \mathcal{C}(v,r) = \mathcal{C}(v',r')$, then $c = \perp$.*
- *(Hiding) The following ensembles are computationally indistinguishable*
  - $\{\mathcal{C}(v)\}_{n\in N, v\in\{0,1\}^n, v'\in\{0,1\}^n}$
  - $\{\mathcal{C}(v')\}_{n\in N, v\in\{0,1\}^n, v'\in\{0,1\}^n}$

  where $\mathcal{C}(v)$ denotes the output of $\mathcal{C}(v,r)$ where r is a uniformly selected random tape of length $p(|v|)$.

7.2 Defining adaptive selective decommitment security

Let $\mathcal{C}(\cdot,\cdot)$ be a non-interactive commitment scheme, and $\mathcal{D} = \{D_n\}_{n\in N}$ be an ensemble of distributions. We compare between a "real" and an "ideal" execution. In the "real" experiment, the adversary gets to see $m(n)$ commitments to values $v_1, \ldots, v_m$ sampled from some distribution D_n, and may adaptively ask for decommitments of any commitment c_i where i is part of the "legal" set I. In the "ideal" experiment, the adversary simply gets to (adaptively) ask for the values v_i for any $i \in I$. Let $\mathsf{real}(\mathcal{C}, \mathcal{D}, f, I, A, n)$ denote the output of the following experiment:

- Sample $(\vec{x}, z)$ from D_n. For each $i \in |\vec{x}|$, let $c_i = \mathcal{C}(x_i; r_i)$ where r is a uniform random string (of the appropriate length). Feed $(1^n, \vec{c}, z)$ to A.
- Iterate the following until A halts: whenever A outputs a message i, if $i \in I$, feed the decommitment (x_i, r_i) to S.
- Finally, output 1 if the final output of A equals $f(\vec{x})$ and 0 otherwise.

Let $\mathsf{ideal}(\mathcal{D}, f, I, S, n)$ denote the output of the following experiment:

- Sample $(\vec{x}, z)$ from D_n. Feed $(1^n, z)$ to S.
- Iterate the following until S halts: whenever A outputs a message i, if $i \in I$, feed x_i to S.
- Finally, output 1 if the final output of A equals $f(\vec{x})$ and 0 otherwise.

Definition 14 (Adaptive Selective Decommitment Security)**.** *Let $\mathcal{C}(\cdot,\cdot)$ be a commitment scheme. We say that $\mathcal{C}(\cdot,\cdot)$ is* secure under adaptive selective decommitment *w.r.t the legal set $I = \{I_n\}_{n\in N}$ where $I_n \subset [m(n)]$ and the ensemble of distributions $\mathcal{D} = \{D_n\}$, where D_n is a distribution over $(\{0,1\}^{\mathrm{poly}(n)})^{m(n)}$, if for every polynomial-time computable function f, every probabilistic polynomial-time adversary A, every polynomial $p(\cdot)$, there exists a polynomial-time algorithm S such that for all $n \in N$,*

$$\Pr\left[\mathsf{ideal}(\mathcal{D}, f, I_n, S, n)\right] \geq \Pr\left[\mathsf{real}(\mathcal{C}, \mathcal{D}, f, I_n, A, n) = 1\right] - \frac{1}{p(n)}$$

Remark 9. *We note that our definition is a slight relaxation of an "adaptive" variant of the notion of selective-decommitment security with respect to functions of [DNRS03]. We remark that [DNRS03] also consider stronger "simulation-based" notions of security; since we are proving a lower bound, we focus on the weaker notion of security with respect to functions.*

7.3 Instantiating GMW

We say that an ensemble $\mathcal{D} = \{D_n\}_{n\in N}$ of distributions over R_L with auxiliary input is *hard* if for every adversary A whose running-time is polynomial in the length of its first input, there exists a negligible function μ such that the probability that $A(x, z)$ outputs $w \in R_L(x)$ is at most $\mu(n)$, where x, y, z are chosen according to D_n.

Let R_L be a unique witness relation and let L be the language characterized by R_L. Let us denote by $\mathsf{Para\text{-}GMW}_L$ the "parallelized" version of GMW's graph 3 coloring protocol for proving the language L—that is, the protocol obtained by sufficiently repeating GMW-G3C in parallel to get a protocol with large challenge space (and thus negligible soundness error). For simplicity (just as in [DNRS03]), we consider a slightly modified version of GMW-G3C where we ensure that the witness used by the prover can be efficiently decoded from the values committed to by the honest prover.

Proposition 3. *Consider* Para-GMW$_L$ *instantiated with a commitment scheme* $\mathcal{C}(\cdot,\cdot)$ *that is secure under adaptive selective decommitment for any legal set* $I = \{I_n\}_{n\in N}$ *where* $I_n \subset [m(n)]$*, and any efficiently computable ensemble of distributions* $\mathcal{D} = \{D_n\}$*, where* D_n *is a distribution over* $(\{0,1\}^{\text{poly}(n)})^{m(n)}$*. Then for every polynomial-time computable hard ensemble* $\mathcal{D}'$ *over* R_L *(with auxiliary information),* Para-GMW$_L$ *is witness hiding under sequential composition w.r.t.* $\mathcal{D}, R_L$*.*

Proof. The proof closely follows the original proof by DNRS showing that Para-GMW$_L$ is "semantically-secure for functions" (see [DNRS03] for more details); this notion implies single-instance witness hiding for unique witness relations. However, since we are considering commitment schemes secure against *adaptive* selective decommitment (instead of non-adaptive selective decommitment as DNRS), we can show that the protocol in fact also is witness hiding under sequential composition.

Assume that there exists a hard ensemble $\mathcal{D}$ and an adversary A that can break witness hiding under $\ell(\cdot)$ sequential repetitions of Para-GMW$_L$ w.r.t. $\mathcal{D}, R_L$. We let $\mathcal{D}' = \{D'_n\}$ be an ensembles where D'_n is defined as follows: sample (x, y, z) from D_n, output $(\vec{v}, (x, z))$ where $\vec{v}$ are the values the honest prover would commit to in $\ell(|x|)$ executions of Para-GMW$_L$ on input (x, y). Let $f_L^{GMW}(\vec{v})$ be the function that "decodes" the values $\vec{v}$ to the witness w (and outputs $\perp$ if the values do not define a valid witness). Let $I_L^{GMW} = \{I_n\}_{n\in I}$, where I_n is the set of indexes denoting legal openings to the $\ell(n)$ executions of the GMW protocol—i.e., for each "chunk" of commitments corresponding to a single execution, we may only open 2 commitments.

Now, consider any attack on witness hiding under $\ell(\cdot)$ sequential execution on Para-GMW$_L$ w.r.t. $\mathcal{D}, R_L$. Any such attack can be viewed as an attack on adaptive selective decommitment on $\mathcal{C}$ w.r.t $\mathcal{D}'$, the function f_L^{GMW} and the legal set I_L^{GMW}. As in [DNRS03], the key point is that in the ideal experiment, the openings to any legal set can be perfectly simulated (by simply picking two random colors for each execution). So, in the ideal experiment, the function f_L^{GMW} cannot be computed except with negligible probability (since by definition $\mathcal{D}'$ is hard); it follows that also in the real experiment f can only be computed with negligible probability, which concludes that A can only recover the (unique) witness with negligible probability.

□

7.4 Limits of Adaptive-Selective Decommitment Security

By combining Proposition 3 with Theorem 2, we get as a corollary that, assuming the existence of an efficiently computable hard ensemble over an $\mathcal{NP}$-relation R_L with unique witnesses (i.e., a one-to-one one-way function), adaptive selective decommitment secure commitments cannot be based on any standard assumption using a Turing reduction.

We proceed to formalize this, while considering a very weak notion of selective decommitment security.

Definition 15 (Strongly Breaking Adaptive Selective-Decommitment)**.** *We say that* A strongly breaks adaptive selective decommitment *of* $\mathcal{C}(\cdot,\dot{)}$ *w.r.t* $\mathcal{D}$*, the legal set* $I = \{I_n\}_{n\in N}$ *and the function* f*, if for every* $n \in N$*,* $\Pr[\mathsf{real}(\mathcal{C}, \mathcal{D}, f, I_n, A, n) = 1] = 1$*.*

Definition 16 (Basing Weak Adaptive Selective-Decommitment Hiding on C)**.** *We say that* R *is a black-box reduction for basing weak adaptive selective-decommitment hiding of* $\mathcal{C}(\cdot,\cdot)$ *w.r.t* $\mathcal{D}$*, the legal set* $I = \{I_n\}_{n\in N}$ *and the function* f*, on the hardness of* C *w.r.t. threshold* t*, if* R *is a probabilistic polynomial-time oracle machine and there exists a polynomial* $p(\cdot)$*, such that for every deterministic machine* A *that strongly breaks adaptive selective decommitment of* $\mathcal{C}$ *w.r.t. to* $\mathcal{D}, I, f$ *we have that for infinitely many* $n \in N$*,* R^A *breaks* C *w.r.t.* t *with probability* $\frac{1}{p(n)}$*.*

We now show that, assuming the existence one-to-one one-way functions, for any polynomial $r(\cdot)$, there exists (efficiently computable) $\mathcal{D}, I, f$ such that we cannot base weak adaptive selective decommitment hiding of $\mathcal{C}$ w.r.t., $\mathcal{D}, I, f$ on any $r(\cdot)$-round assumption C. Note that merely showing the existence of such $\mathcal{D}, I, f$ is trivial—if the legal set I allows a polynomial-time adversary to compute f, even in the "ideal" experiment, then obviously we cannot base hiding on any assumption. The difficulty is to exhibit $\mathcal{D}, I, f$ such that f cannot be computed in the ideal experiment, yet adaptive selective decommitment hiding of $\mathcal{C}$ w.r.t. $\mathcal{D}, I, f$ still cannot be based on any standard assumption.

To formalize this, we say that f *is hard for* $\mathcal{D} = \{D_n\}_{n\in N}$, $I = \{I_n\}_{n\in N}$ if for all polynomial-time A, there exists a negligible function $\mu(\cdot)$ such that for all $n \in N$, $\Pr[\mathsf{ideal}(D_n, f, I_n, A, n)] \leq \mu(n)$.

Theorem 4. *Assume the existence of a one-to-one one-way function. Then for any polynomial $r(\cdot)$, there exists an efficiently computable ensemble $\mathcal{D}$, an efficiently recognizable legal set I, and an efficiently computable function f that is hard for $\mathcal{D}, I$, such that if weak adaptive selective decommitment hiding of $\mathcal{C}$ w.r.t $\mathcal{D}, I, f$ can be based on the hardness of any $r(\cdot)$-round assumption C w.r.t. threshold t, then there exists a machine B and a polynomial $p(\cdot)$, such that for infinitely many $n \in N$, B breaks C w.r.t. t with probability $\frac{1}{p(n)}$.*

Proof. Let g be a one-to-one one-way function let R_L be the unique witness relation where $(x, w) \in R_L$ if and only if $g(w) = x$, and let L be the language characterized by R_L. Let $\mathcal{D} = \{D_n\}_{n\in N}$, where D_n is the distribution obtained by sampling X, Z as follows; let X be the values $\vec{v}$ that the honest prover in $\mathsf{Para\text{-}GMW}_L$ commits to on input (x, w) in $\ell(|x|)$ executions where $\ell(n') = \omega(n' + 2r(n') + 1)$, and where is w is a random n-bit string and $x = g(w)$; let $Z = x$. Let $f = f_L^{GMW}$ and $I = I_L^{GMW}$, defined in the proof of Proposition 3. It easily follows from the same argument as in the proof Proposition 3 that f is hard for $\mathcal{D}, I$. Observe that any adversary A that strongly breaks $\ell(\cdot)$-sequential witness hiding of $\mathsf{Para\text{-}GMW}_L$ w.r.t. R_L, also strongly breaks adaptive selective decommitment security of $\mathcal{C}$ w.r.t. $\mathcal{D}, I, f$. Thus, any reduction R from breaking an $r(\cdot)$-round assumption C to strongly breaking adaptive selective decommitment of $\mathcal{C}$ w.r.t. $\mathcal{D}, I, f$, is also a reduction from breaking C to strongly breaking $\ell(\cdot)$-sequential witness hiding of $\mathsf{Para\text{-}GMW}_L$ w.r.t. R_L. The theorem next follows by applying the quantitative version of Theorem 2 stated in Remark 3. □

Remark 10. *(On super-polynomial-time reductions) We remark that if considering* statistically-binding *commitment schemes, the above argument combined with Remark 5 directly rules out using also $T(\cdot)$-time reductions, where $T(n) = 2^n$: We just need to let $\mathsf{Para\text{-}GMW}_L$ consist of n^2 (where n is the length of the common input) parallel repetitions of GMW-G3C to ensure that the protocol has a sufficiently large challenge space.*

On the other hand, for computationally binding schemes, super-polynomial-time reductions are useful. Consider a (two-round) statistically hiding scheme but with a scaled down security parameter ensuring that correctly distributed openings to any value can be found in quasi-polynomial time. For such commitments, we can easily simulate both commitments and openings to arbitrary values in quasi-polynomial time. Thus, they satify adaptive selective decommitment security with a quasi-polynomial simulator S.

Remark 11. *(On encryption schemes secure against selective decryption) We mention that Bellare, Hofheinz and Yilek [BHY09] have recently shown the existence of* encryption schemes *secure against selective decryption based on standard type assumptions. We refer the reader to [BHY09] for a formal definition of selective decryption security. The reason our lower bounds do not extend to*

encryption schemes is the fact that encryption schemes do not neccesarily have to be commiting *for* all *public keys—that is, there exists some public keys for which the encryption of a message m does not uniquely determine m.*

On the other hand, if we restrict to commiting encryption schemes then our lower bounds directly extend to rule out security against adaptive selective decryption (based on standard assumptions). Consider, for instance, the ElGamal encryption scheme [Gam84]. It is well know that this scheme is committing, so if it is secure against adaptive selective decryption, it yields a non-interactive commitment secure against adaptive selective decommitment.

8 Security of Generalized One-more Inversion Assumptions

In this section we show limitations of basing generalized "one-more" assumptions on standard assumptions using Turing reductions.

Let us start by recalling the one-more discrete logarithm assumption [BNPS03, BP02]. Recall that a discrete logarithm parameter generator $D(1^n)$ outputs $1^n, p, q, g$ where $2^{n-1} \leq p \leq 2^n$ is a prime, q is a length $l(n) = \omega(\log n)$ prime such that $q|p-1$, and g is a generator of a subgroup G_q of $\mathbb{Z}_p^*$ of order q. The *one-more discrete logarithm assumption* states that no polynomial-time algorithm A can, given the output $1^n, p, q, g$ of a discrete logarithm generator, and $\ell(n)$ target values $y_1 = g^{x_1} \mod p, \ldots, y_{\ell(n)} = g^{x_{\ell(n)}} \mod p$, where $x_1, \ldots, x_{\ell(n)}$ are uniformly picked from $\mathbb{Z}_q$ and $\ell(\cdot)$ is a polynomial, recover $x_1, \ldots, x_{\ell(n)}$ with non-negligible probability, even if A has access to $\ell(n) - 1$ oracle queries to a discrete logarithm oracle for G_q, g.

Let us formalize this assumption in the language of witness hiding. Let D be a discrete logarithm generator. Let us define a witness relation R_L, an ensemble of distributions $\mathcal{D}^\ell$ over R_L with auxiliary input, and an interactive proof (P, V).

- Let $(x_1, \ldots, x_{\ell(n)}) \in R_L((1^n, p, q, g), y_1, \ldots, y_{\ell(n)})$ if and only if $(1^n, p, q, g)$ are valid discrete logarithm parameters (as mentioned in Section 6 this can be checked efficiently), and if $y_1, \ldots y_n$ are in a subgroup G_q of order q (again, as mentioned in Section 6 this can be checked efficiently), and for every $i \in [\ell(n)]$, $y_i = g^{x_i} \mod p$. Note that this is a unique witness relation. Let L be the language characterized by R_L; note that this language is polynomial-time decidable.

- Let D_n^ℓ output (X, Y, Z), obtained as follows: let $(1^n, p, q, g)$ be the output of $D(1^n)$; uniformly pick $\ell(n)$ elements $x_1, \ldots, x_{\ell(n)} \in \mathbb{Z}_q$, for $i \in [\ell(n)]$, let $y_i = g^{x_i} \mod p$; finally let $X = ((1^n, p, q, g), y_1, \ldots, y_{\ell(n)}), Y = (x_1, \ldots, x_{\ell(n)})$, and let Z simply be empty (i.e., there is no auxiliary information).

- Now consider the protocol (P, V) defined in Figure 3.

The one-more discrete logarithm assumption with respect to the discrete logarithm parameter generator D can now be stated as the assumption that:

> *For evert polynomial ℓ, (P, V) is $(\ell(\cdot) - 1)$-sequentially witness hiding w.r.t $\mathcal{D}^\ell = \{D_n^\ell\}_{n \in N}$, R_L.*

Let us now show that the assumption that (P, V) is witness hiding under $\ell(n)^\epsilon$ (where $\epsilon > 0$) sequential repetitions w.r.t $\mathcal{D}^\ell, R_L$ cannot be based on any standard assumption using a Turing reduction, for sufficiently large $\ell(\cdot)$. That is, even a "many-more" variant of the discrete logarithm assumption cannot be based on standard assumptions using a Turing reduction.

A DISCRETE LOGARITHM CHALLENGE PROTOCOL (P, V):

On common input $X = ((1^n, p, q, g), y_1, \ldots, y_{\ell(n)})$, the prover P and verifier V proceed as follows.

- V first checks that $X \in L$; if not, it aborts (rejecting). Otherwise, V picks a random $y \in G_q$ and sends it to the prover, where G_q denotes the order q subgroup of $\mathbb{Z}_p^*$ generated by g.
- P checks that $y \in G_q$ and if so, it computes an x such that $g^x = y$ and sends it to the verifier.
- V accepts if and only if $g^x = y$.

Figure 3: A discrete logarithm challenge protocol

Definition 17 (Strongly Breaking Many-More Security of DLOG)**.** *Let (P, V) and R_L be as defined above. We say that A* strongly breaks the (ℓ, ϵ)-many-more DLOG assumption *if A strongly breaks ℓ^ϵ-sequential witness hiding of (P, V) w.r.t. R_L.*

Definition 18 (Basing Weak Many-More Security of DLOG on C)**.** *We say that R is a black-box reduction for basing weak (ℓ, ϵ)-many-more security of DLOG on the hardness of C w.r.t. threshold t if R is a probabilistic polynomial-time oracle machine, such that for every deterministic machine A that strongly breaks the (ℓ, ϵ)-many-more DLOG assumption, there exists a polynomial $p(\cdot)$ such that for infinitely many $n \in N$, R^A break C w.r.t. t with probability $\frac{1}{p(n)}$ on input 1^n.*

Theorem 5. *Let C be an r-round assumption, where r is a polynomial, and let $\epsilon > 0$. If for every sufficiently large polynomial $\ell(\cdot)$ there exists a black-box reduction R for basing weak (ℓ, ϵ)-many-more security of DLOG on the hardness of C w.r.t. threshold t, then there exists a probabilistic polynomial-time machine B and a polynomial $p'(\cdot)$ such that for infinitely many $n \in N$, B breaks C w.r.t. t with probability $\frac{1}{p'(n)}$ on input 1^n.*

Proof. We transform the protocol (P, V) into a protocol (P, V') that satisfies the generalized notion of computational special-soundness. V' proceeds just as V, except that instead of picking the challenge y at random in G_q, it picks a random $i \in [\ell(n)]$, $r \in \mathbb{Z}_p^*$, and computes the challenge y as $y = y_i^r$. The complete description of protocol (P, V') is found in Figure 4.

Clearly, if (P, V) is (sequentially) witness hiding for some distribution, then so is (P, V') as we have only changed the verifier strategy; additionally, it is easy to see that (P, V') still has perfect completeness.

Below we show that (P, V') satisfies the generalized notion of computational special-soundness with large challenge space. Since R_L is a unique witness relation, we can then apply the quantitative version of Theorem 2 stated in Remark 3 to conclude that there is no Turing reduction from recovering $x_1, \ldots, x_{\ell(n)}$ after $\ell(n)^\epsilon$ sequential interactions of (P, V) to any polynomial-round assumption, as long as $\ell(n)$ is sufficiently big.

Proposition 4. *(P, V') is a generalized computationally special-sound argument with large challenge space for L and the witness relation R_L.*

Proof. We show that (P, V') satisfies the definition of generalized special-soundness when the polynomial $m(n) = \ell(n)n$; that is, extraction can be performed from $m(n) = \ell(n)n$ accepting verifier

A MODIFIED DISCRETE LOGARITHM CHALLENGE PROTOCOL (P, V'):

On common input $X = ((1^n, p, q, g), y_1, \ldots, y_{\ell(n)})$, the prover P and verifier V' proceed as follows.

- V' first checks that $X \in L$; if not, it aborts (rejecting). Otherwise, it picks a random $i \in [\ell(n)]$, $r \in \mathbb{Z}_p^*$, and computes the challenge y as $y = y_i^r$ and sends it to the prover.
- P checks that $y \in G_q$ and if so, it computes an x such that $g^x = y$ and sends it to the verifier.
- V' accepts if and only if $g^x = y$.

Figure 4: A modified discrete logarithm challenge protocol

views. Note that for every accepting answer x to a challenge y generated as $y = y_i^r$, we can recover x_i as $x_i = xr^{-1} \mod q$; we say that any such view $\mathcal{V}$ solves index i. Our extractor X, on input $m(n)$ accepting verifier views $\vec{\mathcal{V}}'$, checks if every index $i \in [\ell(n)]$ is solved by some view $\mathcal{V} \in \vec{\mathcal{V}}'$, and if so it applies the above method to recover and output $x_1, \ldots, x_{\ell(n)}$.

Let us now argue that for every polynomial $p(n)$, given $p(n)$ verifier views $\vec{\mathcal{V}}$ generated as in the definition of computational special-soundness, it holds that, except with negligible probability, if $\vec{\mathcal{V}}$ contains $m(n)$ accepting views, then $X(\vec{\mathcal{V}}')$ (where $\vec{\mathcal{V}}'$ are the first $\ell(n)n$ accepting views in $\vec{\mathcal{V}}$) succeeds in recovering $x_1, \ldots, x_{\ell(n)}$. To show this, we just need to argue that, except with negligible probability, for every $i \in [\ell(n)]$, there exists some view $\mathcal{V} \in \vec{\mathcal{V}}'$ that solves index i. Since the distribution of a challenge y generated as $y = y_i^r$ is *independent* of i, we have that for every $j \in [m(n)]$, the probability that the j'th accepting view solves index i is $\frac{1}{\ell(n)}$. So, if we have $m(n) = \ell(n)n$ accepting views, it holds that for every $i \in [\ell(n)]$, the probability that index i is not solved by any of the views in $\vec{\mathcal{V}}'$ is negligible; thus, by the union bound, it holds that, whenever we have $m(n)$ accepting views $\vec{\mathcal{V}}'$, then, except with negligible probability, all indexes are solved by some view $\mathcal{V} \in \vec{\mathcal{V}}'$ and thus $X(\vec{\mathcal{V}}')$ recovers a witness. This concludes that (P, V) satisfied the generalized notion of computational special-soundness. Clearly it also has large challenge space. □

□

Remark 12. *(On super-polynomial-time reductions) By appealing to the super-polynomial version of our main theorem in Remark 5, Theorem 5 directly extends to rule out also using super-polynomial reductions as long as the size of the prime q is sufficiently big (as the protocols described above are special-sound also for unbounded attackers).*

Also note that if q is not sufficiently big, then inversion can be done by brute force, so the inversion oracle can be simulated "for free".

8.1 Extensions to homomorphic certified permutations

There is nothing special about the discrete logarithm problem. The proof of Theorem 5 works as long as we consider any *additive-homomorphic*[20] family of *certified permutations*: that is, 1) there

[20]That is, permutations f for which there exists an efficient algorithm that on input $f(x_1), f(x_2)$ and the description of f can compute $f(x_1 + x_2)$)

exists an efficient algorithm for determining whether the selected function indeed is a permutation, and 2) elements in the domain of the function are efficiently recognizable. We simply need to consider the protocol in Figure 5. It follows using exactly the same proof as in Theorem 5 that

AN INVERSION CHALLENGE PROTOCOL (P, V'):

On common input $X = ((1^n, f), y_1, \ldots, y_{\ell(n)})$, the prover P and verifier V' proceed as follows.

- V' first checks that f is a permutation of the "appropriate length" w.r.t n (i.e., we need to check that the domain of the function is super polynomial); if not, it aborts (rejecting). Otherwise, it picks a random $i \in [\ell(n)]$, and a random r in the domain of f, and homomorphically computes the challenge $y = f(x_i + r)$; since f is homomorphic we only need to know y_i and r to do this.
- P checks that y is in the range (which equals the domain) of f and if so, it computes an x such that $f(x) = y$ and sends it to the verifier.
- V' accepts if and only if $f(x) = y$.

Figure 5: An inversion challenge protocol

the above protocol is computationally special-sound for a unique witness relation—the key points are that 1) given the inverse x to a challenge $y = f(x_i + r)$ and randomness r, we can compute x_i; and, 2) by the permutation property of f, the distribution of $y = f(x_i + r)$ where r is randomly chosen is independent of i. Thus, by the same arguments as in the proof of Theorem 5, "many-more" inversion security cannot be based on any standard assumption using Turing reductions.

Let us give an example of this generalized version. Consider the *one-more RSA assumption* [BNPS03]: It asserts that no polynomial-time algorithm A, given a product N of two random n-bit primes and an exponent e, can find RSA inverses (i.e., inverses to the function $f_{N,e}(m) = m^e \mod N$) to $\ell(n)$ random target values $y_1, \ldots y_{\ell(n)} \in \mathbb{Z}_N^*$, where $\ell(\cdot)$ is an arbitrary polynomial, even if A can access an inversion oracle for $f_{N,e}$, $\ell(n)-1$ times. We here focus on the the case when $e > N$ is a prime; let us call this the "certified" one-more RSA assumption. For typical applications of the one-more RSA assumption (e.g, to prove the security of Chaum's blind signature scheme [Cha82]) this restriction on the exponent e is needed (see e.g., [CNS07]). This restriction ensures that valid exponents e are efficiently recognizable; for every such valid exponent e we are guaranteed that the function $f_{N,e}(x) = x^e \mod N$ is a permutation. Furthermore, given N and an element y, it is easy to determine whether $y \in \mathbb{Z}_N^*$. We can thus directly apply the above proof to conclude that "many-more" variants of the "certified" RSA problem cannot be based on standard assumption using Turing reduction.

9 Security of Unique Blind Signatures

In this section we investigate *blind signature schemes* [Cha82]. Roughly speaking, blind signatures allow a user to ask a signer S to sign a message m, while keeping the message m secret from S. Consider Chaum's classic Blind signature [Cha82]. The scheme is non-interactive: the client on input a message m and a public-key pk sends a single message y to the signer, and the signer (on

input its secret-key sk) replies with a single message x; the client next can compute a signature σ for m. This scheme also has the following desirable feature: There exists a *unique* accepting signature σ for a message m given the public key pk.

Let us call schemes satisfying the above properties *unique non-interactive blind signatures*. We formalize this notion using a weak notion of blinding which requires that no efficient signer S^* can tell what message is being signed, even if it observes whether the receiver got a valid signature or not.[21]

Definition 19. *[Unique Non-interactive Blind Signature] We say that a tuple of probabilistic polynomial-time algorithms* $\pi = (\mathsf{Gen}, \mathsf{Ver}, \mathsf{Sign}, \mathsf{Blind}, \mathsf{Unblind})$ *is a* unique non-interactive blind signature *if the following conditions hold:*

- (Validity) *For every* $n \in N$, *every* $(pk, sk) \in \mathsf{Gen}(1^n)$, *every* $m \in \{0,1\}^n$, *every random tape* $r \in \{0,1\}^\infty$, *every* $\sigma \in \mathsf{Unblind}_r(pk, m, \mathsf{Sign}(sk, \mathsf{Blind}_r(pk, m)))$, $\mathsf{Ver}(pk, m, \sigma) = 1$.

- (Computational Blinding) *For every polynomial time* S^*, *the following ensembles are computationally indistinguishable*

  - $\{pk, \mathsf{view}^\pi_{S^*}(n, z, pk, m_0)\}_{n \in N, z, \in \{0,1\}^*, pk \in \{0,1\}^*, m_0, m_1 \in \{0,1\}^n}$
  - $\{pk, \mathsf{view}^\pi_{S^*}(n, z, pk, m_1)\}_{n \in N, z, \in \{0,1\}^*, pk \in \{0,1\}^*, m_0, m_1 \in \{0,1\}^n}$

  where $\mathsf{view}^\pi_{S^*}(n, z, pk, m)$ *denotes the output of the following experiment: Let* $S^*(n, z, pk)$ *communicate with the "honest" receiver* $R(pk, m)$*:* $R(pk, m)$ *first computes* $y = \mathsf{Blind}_r(pk, m)$, *where* r *is a uniform random string, and sends it to* S^*. S^* *computes an answer* x; R *computes* $\sigma = \mathsf{Unblind}_r(pk, m, x)$ *and outputs* $o = \mathsf{Ver}(pk, m, \sigma)$. *The experiment finally outputs the pair of outputs* (y, o) *(i.e., the blinded message, and whether a successful signature was recovered).*

- (Uniqueness) *For every* $pk \in \{0,1\}^*$, *every* $m \in \{0,1\}^*$, *there exists at most one* $\sigma \in \{0,1\}^*$ *such that* $\mathsf{Ver}(pk, m, \sigma) = 1$.

Unique signatures were first defined by Goldwasser and Ostrovsky [GO92], and first achieved by Micali, Rabin and Vadhan [MRV99].[22] As far as we know, Camenisch, Neven and Shelat [CNS07] were the first to define unique blind signatures, and to point out that e.g., Chaum's blind signature is unique; Camenisch et al also note that a non-interactive blind scheme by Boldyreva [Bol03] is unique.

We mention that many blind signature schemes (including both Chaum's and Boldyreva's schemes) satisfy the stronger notion of *perfect blinding* where the ensembles in Definition of 19 are required to be *identical* for every potentially *unbounded* S^*.

Let us turn to defining unforgeability. The notion of "one-more" unforgeability [PS00] requires that no polynomial-time algorithm can output more signatures than the number of blind signatures it has requested.

Definition 20 (One-more Unforgeability)**.** *We say that a unique non-interactive blind signature is* one-more unforgeable *if for every polynomial-time algorithm* A, *there exists a negligible function* μ *such that for every* $n \in N$, *the probability that* A *wins in the following experiment is bounded*

[21] More traditional definitions require that a notion of "unlinkability" to hold even if the signer gets to see *actual* signatures recovered by the receiver (and not just whether a valid signature was recoverd). As we are proving a lower bound, we content ourselves with providing the simpler weaker definition.

[22] The work of Goldwasser and Ostrovsky only provided a construction of unique signatures in the Common Reference String model.

by $\mu(n)$: Sample (pk, sk) using $\mathsf{Gen}(1^n)$ and let $A(1^n, pk)$ get oracle access to $\mathsf{Sign}(sk, \cdot)$; A is said to win if it manages to output ℓ valid message-signature pairs (m, σ) such that $\mathsf{Ver}(pk, m, \sigma) = 1$, while having made less than ℓ oracle queries.

Let us formalize a weaker notion of unforgeability in the language of witness hiding. We consider a setting where the goal of the attacker is to find signatures on $\ell(n)$ *random* messages that are *a priori* fixed before it gets to talk to the signer.

Let $\pi = (\mathsf{Gen}, \mathsf{Ver}, \mathsf{Sign}, \mathsf{Blind}, \mathsf{Unblind})$ be a unique non-interactive blind signature scheme. We define a witness relation $R_L^{\pi,\ell}$, an ensemble of distributions $\mathcal{D}^{\pi,\ell}$ over $R_L^{\pi,\ell}$ with auxiliary input, and an interactive proof (P^π, V^π).

- Let $(\sigma_1, \ldots, \sigma_{\ell(n)}) \in R_L^{\pi,\ell}((1^n, pk), m_1, \ldots, m_{\ell(n)})$ if and only if for every $i \in [\ell(n)]$, $\mathsf{Ver}(pk, m_i, \sigma_i) = 1$, and let $L^{\pi,\ell}$ be the language characterized by $R_L^{\pi,\ell}$. Note that by the unique signature requirement, $R_L^{\pi,\ell}$ is a unique witness relation.

- Let $D_n^{\pi,\ell}$ output (X, Y, Z), obtained as follows: let (pk, sk) be the output of $Gen(1^n)$; uniformly pick $\ell(n)$ elements $m_1, \ldots, m_{\ell(n)} \in \{0,1\}^n$, for $i \in [\ell(n)]$, let

$$\sigma_i = \mathsf{Unblind}_r(pk, m_i, \mathsf{Sign}(sk, \mathsf{Blind}_r(pk, m_i)))$$

  where r is a sufficiently long uniform random string; finally, let $X = ((1^n, pk), m_1, \ldots, m_{\ell(n)}), Y = \sigma_1, \ldots, \sigma_{\ell(n)}$, and Z simply be empty (i.e., there is no auxiliary information).

- Now consider the protocol (P^π, V^π) defined in Figure 6.

PROTOCOL (P^π, V^π):

On common input $X = ((1^n, pk), m_1, \ldots, m_{\ell(n)})$, the prover P^π and verifier V^π proceed as follows.

- V^π first checks that all m_i have length n; if not, it aborts (rejecting). Otherwise, if it picks a random $i \in [\ell(n)]$, a uniform random string r, and sends $y = \mathsf{Blind}_r(pk, m_i)$ to the prover.
- P^π computes (using brute-force) a secret key sk such that (pk, sk) is in the range of $D(1^n)$ and replies with $x = \mathsf{Sign}(sk, y)$.
- V^π computes $\sigma = \mathsf{Unblind}_r(pk, m_i, x)$ and accepts if and only if $\mathsf{Ver}(pk, m_i, \sigma) = 1$.

Figure 6: Protocol (P^π, V^π): A witness hiding protocol from the blind signature scheme π.

Clearly any attacker on $(\ell(n) - 1)$-sequential witness hiding of (P^π, V^π) w.r.t $\mathcal{D}^{\pi,\ell}, R_L^{\pi,\ell}$ breaks unforgeability of $(\mathsf{Gen}, \mathsf{Ver}, \mathsf{Sign}, \mathsf{Blind}, \mathsf{Unblind})$. Let us now show that the assumption that (P^π, V^π) is witness hiding under $\ell(n)^\epsilon$ sequential repetitions w.r.t $\mathcal{D}^{\pi,\ell}, R_L^{\pi,\ell}$ cannot be based on any standard assumption using a Turing reduction, for sufficiently large $\ell(\cdot)$. That is, even a "many-more" notion of unforgeability for *a priori* fixed random messages, cannot be based on standard assumptions using a Turing reduction.

Definition 21 (Strongly Breaking Unforgeability)**.** *We say A* strongly breaks (ℓ, ϵ)-unforgeability *of the blind signature scheme π if A strongly breaks ℓ^ϵ-sequential witness hiding of (P^π, V^π) w.r.t. $R_L^{\pi,\ell}$.*

Definition 22 (Basing Weak Unforgeability on C)**.** *We say that R is a black-box reduction for basing weak (ℓ, ϵ)-unforgeability of π on the hardness of C w.r.t. threshold t if R is a probabilistic polynomial-time oracle machine, such that for every deterministic machine A that strongly breaks (ℓ, ϵ)-unforgeability of π, there exists a polynomial $p(\cdot)$ such that for infinitely many $n \in N$, R^A breaks C w.r.t. t with probability $\frac{1}{p(n)}$ on input 1^n.*

Theorem 6. *Let π be a unique non-interactive blind signature scheme, let C be an $r(\cdot)$-round assumption, where r is a polynomial, and let $\epsilon > 0$. If for every sufficiently large polynomial $\ell(\cdot)$ there exists a black-box reduction R for basing weak (ℓ, ϵ)-unforgeability of π on the hardness of C w.r.t. threshold t, then there exists a probabilistic polynomial-time machine B and a polynomial $p'(\cdot)$ such that for infinitely many $n \in N$, B breaks C w.r.t. t with probability $\frac{1}{p'(n)}$ on input 1^n.*

Proof. To show the theorem, we show that (P^π, V^π) satisfies the generalized notion of computational special-soundness. Since $R_L^{\pi,\ell}$ is a unique witness relation, we can then apply the quantitative version of Theorem 2 stated in Remark 3 to conclude that there is no Turing reduction from recovering the signatures $\sigma_1, \ldots, \sigma_{\ell(n)}$ after $\ell(n)^\epsilon$ sequential interactions of (P^π, V^π) to any polynomial-round assumption, as long as $\ell(n)$ is sufficiently big.

Proposition 5. *(P^π, V^π) is a generalized computationally special-sound argument with large challenge space for L and the witness relation R_L.*

Proof. First, note that (P^π, V^π) is complete (but does not have an efficient prover strategy); this follows directly by the validity requirement in the definition of a blind signature. The rest of the proof is very similarly to the proof of Proposition 4 but requires some additional care to deal with the fact that blindness is only computational and not perfect.

We show that (P^π, V^π) satisfies the definition of generalized special-soundness when the polynomial $m(n) = \ell(n)n$; that is, extraction can be performed from $m(n) = \ell(n)n$ accepting verifier views. Note that for every accepting answer x to a challenge y generated as $y = \mathsf{Blind}_r(pk, m_i)$, we recover a signature σ_i for m_i; we say that a verifier view $\mathcal{V}$ *solves* index i whenever it contains such a challenge-response pair. Our extractor X, on input $m(n)$ accepting verifier views $\vec{\mathcal{V}}'$, checks if $\vec{\mathcal{V}}'$ contains views that solve every index $i \in [\ell(n)]$; if so, it recovers (using the above procedure) and outputs all the signatures $\sigma_1, \ldots, \sigma_{\ell(n)}$.

Let us now argue that for every polynomial $p(n)$, given $p(n)$ verifier views $\vec{\mathcal{V}}$ generated as in the definition of computational special-soundness, it holds that the probability that $\vec{\mathcal{V}}$ contains $m(n)$ accepting views and $X(\vec{\mathcal{V}}')$ (where $\vec{\mathcal{V}}'$ are the first $\ell(n)n$ accepting views in $\vec{\mathcal{V}}$) fails in recovering signatures on all of $m_1, \ldots, m_{\ell(n)}$, is negligible.

Towards this, let us first consider a simplified case where the blinding property is perfect. In this case we can apply exactly the same proof as in Proposition 4: Since for every prover P^*, every n, z, pk, the distribution of $\mathsf{view}^\pi_{P^*}(n, z, pk, m_i)$ is *independent* of i, we have that for every $j \in [m(n)]$, the probability that the j'th accepting view solves index i is $\frac{1}{\ell(n)}$. So, if we have $m(n) = \ell(n)n$ accepting views, it holds that for every $i \in [\ell(n)]$, the probability that index i is not solved by any view, is negligible; thus, by the union bound, it holds that whenever we have $m(n)$ accepting views $\vec{\mathcal{V}}'$, then except with negligible probability, all indexes are solved by some view $\mathcal{V} \in \vec{\mathcal{V}}'$. This concludes that in the simplified "perfect blinding" case, the probability that $\vec{\mathcal{V}}$ contains $m(n)$ accepting views and $X(\vec{\mathcal{V}}')$ fails, is negligible.

Let us now turn to the more general case where the blinding property is only computational. It follows using a hybrid argument that also in this case the probability that $\vec{\mathcal{V}}$ contains $m(n)$ accepting views and $X(\vec{\mathcal{V}}')$ fails is negligible. Formally, this is shown by considering a mental experiment where the verifier challenges are generated using the following procedure $\tilde{V}$: $\tilde{V}$ picks a random $i \in [\ell(n)]$ (just as V), but lets $x = \mathsf{Blind}(pk, m_0)$ (instead of $\mathsf{Blind}(pk, m_i)$ as V would have computed it). We now say that a view $\mathcal{V}$ solves index i if in the view $\mathcal{V}$, $\tilde{V}$ picked indexed i. It follows using the proof of the perfect blinding case that in this mental experiment, the probability that $\vec{\mathcal{V}}$ contains $m(n)$ accepting views and yet there exists some index that is not solved by any view $\mathcal{V} \in \vec{\mathcal{V}}'$, is negligible. We can now apply the computational blinding property to conclude that this is still the case when we generate verifier challenges using V instead of $\tilde{V}$.

This concludes that (P, V) satisfied the generalized notion of computational special-soundness. Clearly it also has large challenge space. □

□

Example: Chaum's Blind Signature Scheme. For concreteness, we recall Chaum's blind signature scheme in Figure 7. As noted by Camenisch, Shelat and Neven [CNS07], it is easy to

CHAUM'S BLIND SIGNATURE SCHEME ($\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{Blind}, \mathsf{Unblind}$):

- $\mathsf{Gen}(1^n)$ samples two n-bit primes p, q, lets $N = pq$, samples a prime $e > N$, and outputs $(pk = (N, e), sk = (N, d))$ where $d = e^{-1} \mod (p-1)(q-1)$.[a]
- $\mathsf{Blind}_r((N, e), m)$ checks that N is of length $2|m|$ and that $e > N$ is a prime; if not it outputs $\perp$, otherwise it interprets the random coins r as an element in $\mathbb{Z}_N^*$ and outputs $y = h_N(m)r^e$ where h_N is some appropriate "hash-function" mapping elements from $\{0,1\}^n$ to elements in $\mathbb{Z}_N^*$.
- $\mathsf{Sign}((N, d), y)$ outputs $x = y^d \mod N$ if $y \in Z_N^*$ and $\perp$ otherwise.
- $\mathsf{Unblind}_r((N, e), m, x)$ outputs $\sigma = x/r$ (where r is again interpreted as an element in Z_N^*).
- $\mathsf{Ver}((N, e), m, \sigma)$ outputs 1 if and only if $\sigma^e = h_N(m) \mod N$ and $e > N$ is a prime.

[a] Some descriptions of Chaum's blind signature scheme do not insist that $e > N$ is a prime. But it is well-known that the scheme is not blind unless we make this restriction. See e.g., [CNS07].

Figure 7: Chaum's blind signature scheme [Cha82]

see that this scheme is unique (since the function $f_{N,e} = m^e \mod N$ is a permutation whenever $e > N$ is a prime), and furthermore perfectly blinding. Thus, by Theorem 6, weak unforgeability of Chaum's signature scheme cannot be based on any standard assumption using a Turing reduction. As mentioned above, Camenisch, Shelat and Neven also note that a scheme due to Boldyreva [Bol03] is unique, and thus by Theorem 6, unforgeability of this scheme cannot be based on any standard assumption, using a Turing reduction.

Remark 13. *(On super-polynomial-time reductions) Just as our main theorem, Theorem 6 directly extends to rule out also using super-polynomial reductions, as long as the blindness property holds*

for appropriately strong adversaries, and as long as the length of the messages that can be signed is sufficiently long. In particular, this means that even sub-exponential reductions cannot be used to prove Chaum's scheme unforgeable since the scheme is perfectly blinding.

10 Acknowledgements

I am extremely grateful to Huijia Lin, Edward Lui, Mohammad Mahmoody and Wei-lung Dustin Tseng, for many helpful comments on an earlier draft of this paper.

References

[AAG+00] Fredrik Almgren, Gunnar Andersson, Torbjörn Granlund, Lars Ivansson, and Staffan Ulfberg. How we cracked the code book ciphers. Manuscript, 2000. http://codebook.org/codebook_solution.pdf.

[AGGM06] Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. On basing one-way functions on NP-hardness. In *STOC '06*, pages 701–710, 2006.

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS '01*, volume 0, pages 106–115, 2001.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, 1988.

[BGGL01] Boaz Barak, Oded Goldreich, Shafi Goldwasser, and Yehuda Lindell. Resettably-sound zero-knowledge and its applications. In *FOCS '02*, pages 116–125, 2001.

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *EUROCRYPT*, pages 1–35, 2009.

[Blu86] M. Blum. How to prove a theorem so no one else can claim it. *Proc. of the International Congress of Mathematicians*, pages 1444–1451, 1986.

[BMV08] Emmanuel Bresson, Jean Monnerat, and Damien Vergnaud. Separation results on the "one-more" computational problems. In *CT-RSA*, pages 71–87, 2008.

[BNPS03] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-rsa-inversion problems and the security of chaum's blind signature scheme. *J. Cryptology*, 16(3):185–215, 2003.

[Bol03] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *Public Key Cryptography*, pages 31–46, 2003.

[BP02] Mihir Bellare and Adriana Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In *CRYPTO*, pages 162–177, 2002.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[Bra83] Gilles Brassard. Relativized cryptography. *IEEE Transactions on Information Theory*, 29(6):877–893, 1983.

[BT03] Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for np problems. In *FOCS*, pages 308–317, 2003.

[Bur90] Mike Burmester. A remark on the efficiency of identification schemes. In *EUROCRYPT*, pages 493–495, 1990.

[BV98] Dan Boneh and Ramarathnam Venkatesan. Breaking rsa may not be equivalent to factoring. In *EUROCRYPT*, pages 59–71, 1998.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO*, pages 174–187, 1994.

[CGGM00] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In *STOC '00*, pages 235–244, 2000.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.

[Cha82] David Chaum. Blind signatures for untraceable payments. In *CRYPTO*, pages 199–203, 1982.

[CLP10] Ran Canetti, Huijia Lin, and Rafael Pass. Adaptive hardness and composable security in the plain model from standard assumptions. In *FOCS*, pages 541–550, 2010.

[CNS07] Jan Camenisch, Gregory Neven, and Abhi Shelat. Simulatable adaptive oblivious transfer. In *EUROCRYPT*, pages 573–590, 2007.

[DGS09] Yi Deng, Vipul Goyal, and Amit Sahai. Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. In *FOCS*, pages 251–260, 2009.

[DNRS03] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. *J. ACM*, 50(6):852–921, 2003.

[DNS04] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. *J. ACM*, 51(6):851–898, 2004.

[DOP05] Yevgeniy Dodis, Roberto Oliveira, and Krzysztof Pietrzak. On the generic insecurity of the full domain hash. In *CRYPTO*, pages 449–466, 2005.

[FF93] Joan Feigenbaum and Lance Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22(5):994–1005, 1993.

[FFS87] Uriel Feige, Amos Fiat, and Adi Shamir. Zero knowledge proofs of identity. In *STOC*, pages 210–217, 1987.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *STOC '90*, pages 416–426, 1990.

[FS10] Marc Fischlin and Dominique Schröder. On the impossibility of three-move blind signature schemes. In *EUROCRYPT*, pages 197–215, 2010.

[Gam84] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO*, pages 10–18, 1984.

[GK96] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the fiat-shamir paradigm. In *FOCS '03*, pages 102–111, 2003.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):690–728, 1991.

[GO92] Shafi Goldwasser and Rafail Ostrovsky. Invariant signatures and non-interactive zero-knowledge proofs are equivalent (extended abstract). In *CRYPTO*, pages 228–245, 1992.

[Gol01] Oded Goldreich. *Foundations of Cryptography — Basic Tools*. Cambridge University Press, 2001.

[GW11] Craig Gentry and Daniel Wichs. Separating succint non-interactive arguments from all falsifiable assumptions. To appear in *STOC '11*, 2011.

[HH09] Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. In *TCC*, pages 202–219, 2009.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28:12–24, 1999.

[HRS09] Iftach Haitner, Alon Rosen, and Ronen Shaltiel. On the (im)possibility of arthur-merlin witness hiding protocols. In *TCC*, pages 220–237, 2009.

[IR88] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *CRYPTO '88*, pages 8–26, 1988.

[KP01] Joe Kilian and Erez Petrank. Concurrent and resettable zero-knowledge in poly-loalgorithm rounds. In *STOC '01*, pages 560–569, 2001.

[KP09] Eike Kiltz and Krzysztof Pietrzak. On the security of padding-based encryption schemes - or - why we cannot prove oaep secure in the standard model. In *EUROCRYPT*, pages 389–406, 2009.

[MRV99] Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *FOCS '99*, pages 120–130, 1999.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4:151–158, 1991.

[Nao03] Moni Naor. On cryptographic assumptions and challenges. In *CRYPTO*, pages 96–109, 2003.

[Oka92] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *CRYPTO*, pages 31–53, 1992.

[Pas03] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In *EUROCRYPT*, pages 160–176, 2003.

[Pas06] Rafael Pass. Parallel repetition of zero-knowledge proofs and the possibility of basing cryptography on np-hardness. In *IEEE Conference on Computational Complexity*, pages 96–110, 2006.

[PR05] Rafael Pass and Alon Rosen. Concurrent non-malleable commitments. In *FOCS '05*, pages 563–572, 2005.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, 13(3):361–396, 2000.

[PTV10] Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishan Venkitasubramaniam. Towards non-black-box separations in cryptography. To appear in TCC 2011, 2010.

[PV08] Rafael Pass and Muthuramakrishnan Venkitasubramaniam. On constant-round concurrent zero-knowledge. In *TCC '08*, pages 553–570, 2008.

[RK99] Ransom Richardson and Joe Kilian. On the concurrent composition of zero-knowledge proofs. In *Eurocrypt '99*, pages 415–432, 1999.

[RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In *TCC*, pages 1–20, 2004.

[RV10] Guy N. Rothblum and Salil P. Vadhan. Are pcps inherent in efficient arguments? *Computational Complexity*, 19(2):265–304, 2010.

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174, 1991.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *EUROCRYPT*, pages 256–266, 1997.