A Parallel Repetition Theorem for Constant-Round Arthur-Merlin Games

RAFAEL PASS, Cornell University
MUTHURAMAKRISHNAN VENKITASUBRAMANIAM, University of Rochester

We show a parallel-repetition theorem for constant-round Arthur-Merlin Games, using an *efficient* reduction. As a consequence, we show that parallel repetition reduces the soundness-error at an optimal rate (up to a negligible factor) in constant-round public-coin *argument* systems, and constant-round public-coin *proofs of knowledge*. The former of these results resolves an open question posed by Bellare, Impagliazzo and Naor (FOCS '97).

Categories and Subject Descriptors: F.1.2 [**Theory of Computation**]: Modes of Computation

General Terms: Security, Theory

Additional Key Words and Phrases: Interactive proofs, parallel-repetition, soundness error, public-coin protocols

ACM Reference Format:
Pass, R., Venkitasubramaniam M. 2012. A Parallel Repetition Theorem for Constant-Round Arthur-Merlin Games. ACM Trans. Embedd. Comput. Syst. V, N, Article A (January YYYY), 22 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

Interactive proof system were introduced independently by Goldwasser, Micali and Rackoff [Goldwasser et al. 1989] and Babai and Moran [Babai and Moran 1988]. Roughly speaking, interactive proofs are protocols that allow one party P, called the *Prover* (or Merlin), to convince a computationally-bounded party V, called the *Verifier* (or Arthur), of the validity of some statement $x \in L$. In contrast to traditional "written" proofs (e.g., NP witnesses), in an interactive proof a cheating proof can convince the verifier of a false statement $x \notin L$ with some small probability ϵ; this ϵ is called the *soundness-error* of the interactive proof system. It is well-known (see [Babai and Moran 1988; Goldreich 1998]) that both sequential and parallel repetition can be used to reduce the soundness error at an exponentially decaying rate: k parallel repetitions of an interactive proof system (P, V) with soundness error ϵ results in an interactive proof system with soundness error ϵ^k. In other words, the probability that a cheat-

A preliminary version of this paper appeared in *STOC 2007* under the name "An Efficient Parallel-Repetition Theorem for Arhur-Merlin Games". Pass is supported in part by a Alfred P. Sloan Fellowship, Microsoft New Faculty Fellowship, AFOSR Award FA9550-08-1-0197, BSF Grant 2006317 and I3P grant 2006CS-001-0000001-02, AFOSR YIP Award FA9550-10-1-0093, and DARPA and AFRL under contract GG11413-137380. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US government.
Author's address: Rafael Pass, Department of Computer Science, Cornell University; Muthuramakrishnan Venkitasubramaniam, Department of Computer Science, University of Rochester.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

ing prover succeeds in convincing k independent parallel executions of V, of a false statement, is bounded by ϵ^k.

1.1. Variants Of Interactive Proofs

Since their original introduction, several variants of interactive proofs have been proposed in the literature.

Public v.s. Private Coins.. For one, the notion of interactive proofs introduced by Goldwasser, Micali and Rackoff considers arbitrary probabilistic polynomial-time verifiers, whereas the notion introduced by Babai and Moran, called *Arthur-Merlin Games* considers verifiers that only send truly random messages; such proof systems are also called *public coin*.

Interactive Arguments.. Subsequently, Brassard, Chaum and Crepeau [Brassard et al. 1988] introduced the notion of an interactive argument. In contrast to interactive proofs, where soundness is required to hold with respect to all (potentially unbounded) cheating provers, in an interactive argument, soundness need only to hold for *computationally bounded* cheating provers.

Proofs of Knowledge.. A proof of knowledge protocol is an interactive proof where the prover not only convinces the verifier of the validity of some statement x, but also that it "knows" a (short) certificate for the validity of x. Proof of knowledge protocols were intuitively introduced by Goldwasser, Micali and Rackoff, and were later formalized in [Feige et al. 1988; Feige and Shamir 1989; Tompa and Woll 1987; Bellare and Goldreich 1992].

1.2. Parallel Repetition Of Interactive Proofs

As mentioned, it is known that parallel-repetition reduces the soundness error in all interactive proofs (and thus also for public-coin interactive proofs). However, the situation is less clear when considering the other above-mentioned variants of interactive proofs.

Bellare, Impagliazzo and Naor [Bellare et al. 1997] obtain the first positive results regarding parallel repetition of interactive arguments: they show that parallel repetition reduces the soundness error at an exponentially fast rate in *3-round* interactive arguments (an essentially optimal error reduction rate was more recently obtained in [Canetti et al. 2005]). On the other hand, [Bellare et al. 1997] also shows that there exists *4-round* interactive arguments where parallel repetition does not reduce the soundness error.[1] More precisely, they show that for every k, there exists a protocol (P, V) with communication and computation polynomial in k, such that k-parallel repetitions of (P, V) do not reduce the soundness error below some fixed constant. Additionally, they show the existence of a 4-round protocol for which even *arbitrary* many parallel repetitions cannot be used to reduce the soundness error by a *black-box* reductions. The result of Pietrzak and Wikstrom extends the results of [Bellare et al. 1997] to show the existence of a constant-round protocol for which arbitrary parallel-repetition does not reduce the soundness error [Pietrzak and Wikström 2007].

Nevertheless, although severe negative results were shown in [Bellare et al. 1997] regarding parallel-repetition of interactive arguments, Bellare et al note that their

[1] Pedantically, the lower-bounds of [Bellare et al. 1997] as well as the subsequent result by [Pietrzak and Wikström 2007], only apply to a more general type of computationally-sound interactive protocols where some public-parameter—honestly chosen according to some distribution—is available to both the prover and the verifier. However, as noted in [Pandey et al. 2008], this assumption can be removed by assuming the existence of non-interactive concurrent non-malleable commitments, which may be based on the existence of adaptively-secure one-way permutations.

impossibility results seemingly only apply to interactive protocols where the verifier uses private coins. Consequently, they leave as an open question whether parallel-repetition reduces the soundness error in *public-coin* (i.e., Arthur-Merlin) arguments.

Regarding proofs of knowledge even less is known. Bellare and Goldreich [Bellare and Goldreich 1992] (see also [Goldreich 2001]) point out difficulties with parallel-repetition of proofs of knowledge, but as far as we know, no general results have been shown in the literature.

1.3. Our Results

In this paper, we focus on establishing parallel repetition theorems for interactive proofs with an *efficient* (i.e., polynomial-time) security reduction. Our main theorem shows the existence of such a repetition theorem for constant-round public-coin protocols. As a first corollary of this main theorem, we show that parallel repetition reduces the soundness-error at an essentially optimal rate in constant-round public-coin *argument* systems (resolving one of the open questions in [Bellare et al. 1997]). As a second corollary, we show that parallel repetition reduces the soundness-error at an essentially optimal rate also in constant-round public-coin *proof of knowledge* protocols. As far as we know, these result constitute the first parallel repetition theorems for any general class of protocols with more than three rounds.

We mention that most "classical" arguments, or proof of knowledge protocols (e.g. [Goldreich et al. 1991; Blum 1986]) are public-coin and require 4 rounds of communication.[2] Although it is a well-known fact that parallel repetition decreases both the soundness error, and the so called "knowledge-error" (in the context of proofs of knowledge) in these particular protocols, as far as we know, no general theorems for establishing this were previously known.

Our techniques. To prove our results we show a black-box reduction S that transforms any prover B that convinces k "parallel" verifiers with probability ϵ, into a "stand-alone" prover $B' = S^B$ that only talks to a single verifier. Furthermore (and most importantly), the success probability of the stand-alone prover B' is "roughly" $\epsilon^{1/k}$. The idea behind the reduction is quite straight-forward: S selects one of the k executions that B expects to participate in; this execution will be forwarded externally, while all other executions are "emulated" internally. The complication that arises with this approach is that once a message has been sent externally it is impossible to "rewind" B behind the point where B sent this message. To circumvent this problem, we thus make sure to only externally send messages that allow us to maintain the desired success probability. On a high-level, this is obtained by *recursively* sampling the success probability of our reduction S, while only selecting executions paths that maximize the "recursive" success probability (in a sense, our reduction is somewhat reminiscent of the concurrent zero-knowledge simulation techniques of [Richardson and Kilian 1999]). The main contribution of this paper is to show that this approach indeed works: the principal technique we employ to analyze the success probability of the reduction is a carefully defined sequence of hybrid experiments (as in [Goldwasser and Micali 1984])—this allows us to, step-by-step, reduce the problem to an information-theoretic parallel-repetition question (which becomes simple to analyze). We mention that, due to the recursive nature of our reduction, its running-time grows exponentially with the number of communication rounds in the protocol (and is thus only applicable to constant-round protocols).

[2]Recall that these protocols rely on the existence of a commitment scheme. When using a 2-round commitment scheme (which is needed in the case of statistically-hiding commitments, but also required by most constructions of even computationally-binding commitments) the resulting protocol becomes 4 rounds.

1.4. Related Work

We mention that the questions investigated in this paper, as well as the techniques employed, are very related to *hardness amplification* (see e.g. [Yao 1982; Goldreich et al. 1990]). The question investigated is also related to the large body of work on parallel repetition of *multi-prover games* (e.g., [Ben-Or et al. 1988; Feige and Kilian 2000; Fortnow et al. 1994]), and in particular Raz' repetition theorem [Raz 1998]. However, we mention that both our techniques and desiderata (i.e., efficiency of the reduction) are quite different.

We finally mention that parallel repetition in the context of zero-knowledge proofs [Goldwasser et al. 1989] has also received a lot attention. For instance, Goldreich and Krawczyk [Goldreich and Krawczyk 1996] show that the notion of zero-knowledge is not closed under parallel repetition; furthermore, they also show that the notion of *black-box* zero-knowledge is not closed under parallel repetition, even when considering only constant-round public-coin protocols. In this paper we, however, will not consider "secrecy" properties of interactive proofs, such as the notion of zero-knowledge.

1.5. Subsequent Work

Following the conference publication of this work [Pass and Venkitasubramaniam 2007], several recent works have extended our work in various different directions. Let us provide a brief overview of some of these new developments.

Håstad, Pass, Wikström and Pietrzak (HPWP) [Håstad et al. 2010] remove the constant-round restriction in our result, showing that parallel repetition reduces the soundness error in *any* public-coin argument. Although the high-level idea behind their reduction is similar to ours, there is a critical difference: instead of establishing whether a continuation is "good" by recursive sampling (as we do), HPWP simply pick the first continuation that leads to a successful execution. They next rely on a probabilistic lemma due to Raz (used in Raz's parallel repetition theorem for two-prover games [Raz 1998]) to show that doing this does not significantly reduce the success probability of the reduction. The analysis of HPWP is not tight (even for the case of constant-round protocols), but the recent elegant work of Chung and Liu (CL) [Chung and Liu 2010] provides a better analysis of the HPWP reduction achieving an essentially optimal error reduction.

In an orthogonal vein, Haitner's beautiful work [Haitner 2009] shows how to modify any interactive argument to obtain another interactive argument (for the same language) but whose soundness error reduces with parallel repetition. Another approach due to CL achieves a better error reduction rate, but instead assumes the existence of a fully homomorphic encryption scheme.

HPWP also establishes that parallel repetition reduces the soundness error for a general class of protocols—referred to as "simulatable" protocols—where the verifier's messages (but not necessarily its decision whether to accept or not) can be simulated with noticeable probability.

Impagliazzo, Jaiswal and Kabanets [Impagliazzo et al. 2007] prove "Chernoff-type" parallel-repetition theorems for 2-round private-coin protocols; HPWP extends these results also to simulatable interactive protocols (any 2-round protocol is "trivially" simulatable) and CL provides a tighter security proof. We mention that for arbitrary threshold verifiers, tight bounds for interactive protocols are currently only known for constant-round AM protocols while relying on the reduction (and analysis) presented in this paper. Parallel repetition theorems involving more general classes of parallel verifiers are presented for the case of three-round protocols by Jutla [Jutla 2010], Chung, Lu, Yan [Chung et al. 2010] and Holenstein and Schoenbeck [Holenstein and Schoenebeck 2011]

Finally, Pass, Tseng and Wikström [Pass et al. 2009] present a connection between parallel repetition theorems for interactive arguments, and *lower-bounds* for black-box zero-knowledge protocols; in particular, by relying on the techniques from HPWP (which extend ours) they show that public-coin interactive protocols cannot remain black-box zero-knowledge under parallel repetition. Xiao [Xiao 2011] uses similar ideas in the context of commitments.

1.6. Overview

Section 2 contains basic notation and definitions of interactive proofs, arguments and proofs of knowledge. Section 3 contains formal statements of our results. In Section 4, we start by proving an information-theoretic analogue of our main result. The proof of the main theorem is found in Sections 5 and 6. The remaining theorems are proved in Section 7 and 8.

2. DEFINITIONS

2.1. General Notation

Throughout our paper we use the terms "Turing machine" and "algorithm" interchangeably. We assume familiarity with the basic notions of an *Interactive Turing Machine* [Goldwasser et al. 1989] (ITM for brevity) and a *protocol* (in essence a pair of ITMs[3]).

Probabilistic notation. The following probabilistic notation follows almost verbatim [Goldwasser et al. 1988]. If S is a probability space and p a predicate, then "$x \stackrel{R}{\leftarrow} S$" denotes the elementary probabilistic algorithm consisting of choosing an element x at random according to S and returning x, and "$x \stackrel{R}{\leftarrow} S \mid p(x)$" that of choosing x according to S until $p(x)$ is true and then returning x. Let p be a predicate and $S_1, S_2, \ldots$ probability spaces, then the notation $\Pr[x_1 \stackrel{R}{\leftarrow} S_1;\ x_2 \stackrel{R}{\leftarrow} S_2;\ \ldots\ :\ p(x_1, x_2, \ldots)]$ denotes the probability that $p(x_1, x_2, \ldots)$ will be true after the ordered execution of the probabilistic assignments $x_1 \stackrel{R}{\leftarrow} S_1;\ x_2 \stackrel{R}{\leftarrow} S_2;\ \ldots$ If S, T, ... are probability spaces, the notation $\{x \stackrel{R}{\leftarrow} S; y \stackrel{R}{\leftarrow} T; \cdots : (x, y, \cdots)\}$ denotes the new probability space over $\{(x, y, \cdots)\}$ generated by the ordered execution of the probabilistic assignments $x \stackrel{R}{\leftarrow} S$, $y \stackrel{R}{\leftarrow} T, \cdots$.

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. Interactive Proofs And Arguments

We state the standard definitions of interactive proofs [Goldwasser et al. 1989] and arguments [Brassard et al. 1988]. Given a pair of interactive Turing machines, P and V, we denote by $\langle P(y), V\rangle(x)$ the random variable representing the (local) output of V when interacting with machine P on common input x, when the random input to each machine is uniformly and independently chosen, and P has auxiliary input y.

Definition 2.1 (*Interactive Proof System*). A pair of interactive machines (P, V) is called an interactive proof system for a language L with soundness error $s(\cdot)$ if machine V is probabilistic polynomial-time and the following two conditions hold:

[3]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.

— *Completeness:* For every $x \in L$ it holds that

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

— *Soundness:* For every $x \notin L$, every interactive machine B and every $z \in \{0,1\}^*$

$$\Pr[\langle B(z), V\rangle(x) = 1] \leq s(|x|)$$

If additionally, P receives as auxiliary input a witness y corresponding to input statement x and runs in polynomial time, and the soundness condition is required to hold only with respect to all non-uniform probabilistic polynomial-time provers B, the pair (P, V) is called an interactive argument. Furthermore, if V only sends the prover consecutive and disjoint subsets of its random tape (starting from the beginning of the tape), (P, V) is called public-coin or Arthur-Merlin.

2.3. Proofs Of Knowledge

Loosely speaking, an interactive proof is a proof of knowledge if the prover convinces the verifier that it *possesses*, or can *feasibly compute*, a witness for the statement proved. This notion is formalized by requiring the existence of a *probabilistic polynomial-time* "extractor"-machine that can, given the description of any (malicious) prover that succeeds in convincing the honest verifier, readily computes a valid witness to the statement proved. We proceed to a formal definition, following Bellare and Goldreich [Bellare and Goldreich 1992].

Definition 2.2 (*Proofs of Knowledge*). An interactive proof (argument) (P, V) is a said to be a proof of knowledge for the witness relation R with knowledge error κ if the following two conditions hold:

— *Non-triviality:* For all $x \in L$ and $y_x \in R(x)$:

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

— *Validity:* There exists a polynomial q and a probabilistic oracle machine K, called the *knowledge extractor*, such that for every interactive machine P', every $x \in L$ and every $y, r \in \{0,1\}^*$ the machine K with oracle access to $P'_{x,y,r}$ outputs a solution $s \in R_L(x)$ within an expected number of steps bounded by $\frac{q(|x|)}{\delta}$ as long as $\delta > 0$ where $\delta = \Pr[\langle P'_{x,y,r}, V\rangle(x) = 1] - \kappa(|x|)$ and $P'_{x,y,r}$ denotes the machine P with common input fixed to x, auxiliary input fixed to y and random tape fixed to r.

Remarks

(1) Note that the definition of proofs of knowledge considers *unbounded* prover machines P'. At first sight, one might therefore think that soundness-error reduction would follow by information-theoretic arguments. Note, however, that the definition of a proof of knowledge requires the existence of an *efficient* extractor machine (which thus complicates the analysis).
(2) We finally mention that the definition of a proof of knowledge can be relaxed to consider only efficient prover machines P'; this leads to the definition of an *argument of knowledge*. Our results remain valid also for such a relaxed definition.

3. MAIN THEOREM AND APPLICATIONS

Let (P, V) be a $2m$-round interactive argument for a language L. k-parallel repetition of a (P, V) denotes k *independent* parallel executions of (P, V): We denote by V_k the verifier that runs k independent executions of V and accepts if and only if V accepts in all the executions. More formally, the random tape R of V_k has the form $R_1, \ldots, R_k$ where each R_i is a random tape for V. The j'th message in the protocol, M_j, is parsed

as $M_j = M_{j,1} \dots M_{j,k}$. The reply $M_{j+1} = V_k(x, M_1, \dots, M_j; R)$ of V^k to some message M_j is defined via $M_{j+1,i} = V(x, M_{1,i} \dots M_{j,i}; R_j)$ for $j = 1, \dots, m$. The verifier V_k finally accepts if and only if all of the executions are accepted by V.

Note, however, that a *malicious* prover may not act independently in the k executions. In particular, such a prover might compute a reply M_{j+1} to M_j by choosing $M_{j+1,i}$ to depend on all previous information, including values $M_{t,l}$ where $l \neq i$ (and $t \leq j$).

Our main theorem is presented below.

THEOREM 3.1. *Let (P, V) be a $2m$-round public-coin interactive argument for L, where $m > 0$ is an integer. Then for any interger $d > 0$, there exists a probabilistic oracle machine S, such that for every $x \notin L$, $\epsilon' > 0$, $0 < \delta < \epsilon'$, integer $k > 0$, every interactive machine B, if*

$$\Pr\Big[\langle B, V_k\rangle(x) = 1\Big] \geq \epsilon'$$

then

$$\Pr\Big[\langle S^B(x, \delta, d), V\rangle(x) = 1\Big] \geq (\epsilon')^{\frac{1}{k}} - \frac{1}{|x|^d}$$

Furthermore, the running-time of S is bounded by a polynomial in $\frac{1}{\delta}, |x|$ and k.

The proof of Theorem 3.1 is found in section 5. We point out that since we are only interested in *public-coin* argument systems, the verifier's next message is always independent of the history; in fact it will always be a truly random string. The proof of Theorem 3.1 crucially relies on this fact.

As an application of Theorem 3.1, we show that parallel repetition reduces the soundness error at an essentially optimal rate for both constant-round public-coin interactive arguments and proofs of knowledge. If $k(\cdot)$ is a function, let (P_k, V_k) denote the interactive protocol that on common input x runs $(P_{k(|x|)}, V_{k(|x|)})$.

THEOREM 3.2. *Let (P, V) be a constant-round public-coin interactive argument for the language L with soundness error $\epsilon(\cdot)$. Then for every polynomial $k(\cdot)$, (P_k, V_k) has soundness error $\epsilon'(\cdot)$ where $\epsilon'(n) = \epsilon(n)^{k(n)} + \nu(n)$ and $\nu(\cdot)$ is a negligible function.*

THEOREM 3.3. *Let (P, V) be a constant-round public-coin proof of knowledge for the witness relation R with knowledge error $\epsilon(\cdot)$. Then for every integer d, and every polynomial $k(\cdot)$, (P_k, V_k) has knowledge error $\epsilon'(\cdot)$ where $\epsilon'(n) = \epsilon(n)^{k(n)} + \frac{1}{n^d}$.*

The proofs of Theorem 2 and Theorem 3 are found in sections 7 and 8 respectively. We mention that the constant-round restriction is due to the fact that the running-time of the reduction, guaranteed by Theorem 3.1, is exponential in the number of rounds in the protocol.

4. A WARM UP: THE INFORMATION-THEORETIC CASE

We begin by proving an information-theoretic analogue of Theorem 3.1; that is, we construct an unbounded machine $\widehat{S}$ that satisfies the conditions of Theorem 3.1.

LEMMA 4.1 (INFORMATION-THEORETIC CASE). *Let (P, V) be any $2m$-round Arthur-Merlin argument/proof system. Then, there exists a probabilistic oracle machine $\widehat{S}$ such that for every $x \notin L$, $\epsilon' > 0$, integer k and every deterministic interactive machine B, if*

$$\Pr\Big[\langle B, V_k\rangle(x) = 1\Big] \geq \epsilon'$$

then

$$\Pr\left[\langle \widehat{S}^B(x), V\rangle(x) = 1\right] \geq \epsilon'^{\frac{1}{k}}$$

On input a statement x and given oracle access to B, $\widehat{S}$ proceeds as follows. First, $\widehat{S}$ picks a random coordinate $i \in [k]$. Then, $\widehat{S}$ interacts with B while emulating the honest verifier strategy V for B in all k-parallel executions, except execution i; messages in execution i, on the other hand, will be forwarded to an external verifier. More precisely, $\widehat{S}$ performs a straight-line invocation of B, externally forwarding all messages that are part of B's i'th executions (i.e., all incoming messages from the external verifier are directly forwarded to B as part of its i'th execution, and all messages sent by B in its i'th execution are externally forwarded), and internally generating messages that are part of the other executions of B. In order to select the internally generated messages, $\widehat{S}$ chooses the "best" possible messages; that is, the messages that maximizes $\widehat{S}$'s success probability with the external verifier. Since $\widehat{S}$ is an unbounded machine it can find the best messages by enumerating every possible message and computing the success probability $\widehat{S}$ conditional on the current history.

We let $\widehat{S}_i$ denote the machine that uses the i^{th} coordinate to forward the messages externally and chooses the "best" possible messages for the other coordinates. $\widehat{S}$ is simply the machine that randomly chooses an index i from $[k]$ and runs $\widehat{S}_i$. A formal description of $\widehat{S}_i$ is provided in Figure 1. The following notations is useful.

4.1. Conventions And Notations

In the remainder of the proof we fix the inputs x for $\widehat{S}$ (and $\widehat{S}_i$). To simplify notation, we omit these inputs (e.g., we assume that they are hardcoded). Let $n = |x|$. We assume (without loss of generality) that the argument system (P, V) has $2m$ rounds, for some fixed constant m, and that V sends the first message and P sends the last message. For simplicity of notation, we assume that the length of the verifier's messages in every round is n; our proof does not depend on this assumption and works as long as the messages are of polynomial length (in n). We start by providing some additional notation that will be used throughout the proof.

A history is a sequence of tuples of messages. Since we are interested in the "view" of B, we will only consider histories of verifier messages. We distinguish between two types of histories: *prover*-step histories and *verifier*-step histories: *verifier*-step histories are of the form $h^q = \left((r_1^1, .., r_k^1), .., (r_1^q, .., r_k^q)\right)$ is a sequence of k-tuples of n-bit strings, whereas *prover*-step histories are of the form $(h^q; i, r)$ where h^q is a *verifier*-step history and r is the verifier's next message in the i^{th} coordinate. We say that a history h^q is of length q if it consists of a sequence of q k-tuples. We denote by $(h^q; (s_1, \ldots, s_k))$ the $q+1$-length history obtained by concatenating the q-length history h^q and the k-tuple $(s_1, \ldots, s_k)$. We denote by $(h^q; i, r, U_n^{k-1})$ denotes the set containing all histories $h^q; (s_1, \ldots, s_k)$ such that $s_i = r$ and $s_1, \ldots, s_k \in \{0,1\}^n$. This will be called an *i-extension* of the history $(h^q; r)$. We will refer to as it as just an *extension* whenever i is obvious from context. Finally, we let λ denote the empty history.

Let $\text{WIN}[A^B, h]$ denote the probability the oracle machine A with oracle access to B wins in its external execution with V, if starting from history h. It follows immediately from the definition that

$$\text{WIN}[A^B, h^q] = E\left[r \leftarrow \{0,1\}^n : \text{WIN}[A^B, (h^q; i, r)]\right]$$

since the verifier picks a random string r independent of the history h^q, and the probability of A^B winning from history $h^q; r$ is $\text{WIN}[A^B, (h^q; i, r)]$. Thus,

$$\text{WIN}[M^B, h^q] = \frac{1}{2^n} \sum_{r \in \{0,1\}^n} \text{WIN}[A^B, (h^q; i, r)]$$

We will also let $\text{WIN}[B, h^q]$ denote the probability that prover B wins in (B, V^k) on a given history h^q.

Algorithm 1 : $\widehat{S}_i^B(h^q)$

1: Let $(c_1, \ldots, c_k)$ be the output of B when fed the messages in h^q, i.e., $(c_1, \ldots, c_k) \leftarrow B(h^q)$.
2: **if** $q = m$ **then**
3: Externally forward c_i and halt.
4: **else**
5: Externally forward c_i and let r be the message externally received.
6: Sample extension h^{q+1} uniformly from $(h^q; i, r, U_n^{k-1})$ that maximizes $\text{WIN}[\widehat{S}_i^B, h^{q+1}]$.
7: Run $\widehat{S}_i^B(h^{q+1})$.

Fig. 1. Description of the machine $\widehat{S}_i$

4.2. Proof Of Lemma 4.1

Using the above notation, Lemma 4.1 can be restated as

$$\text{WIN}[\widehat{S}^B, \lambda] \geq \epsilon'^{\frac{1}{k}}$$

Recall that, $\widehat{S}$ chooses i uniformly at random and runs $\widehat{S}_i$. Hence,

$$\text{WIN}[\widehat{S}^B, \lambda] = \frac{1}{k} \sum_{i=1}^{k} \text{WIN}[\widehat{S}_i^B, \lambda]$$

Below, we show the following claim.

CLAIM 1. $\prod_{i=1}^{k} \text{WIN}[\widehat{S}_i^B, \lambda] \geq \text{WIN}[B, \lambda]$

The proof of Lemma 4.1 is next concluded using the Arithmetic Mean-Geometric Mean inequality (AM-GM).[4]

$$\text{WIN}[\widehat{S}^B, \lambda] = \frac{1}{k} \sum_{i=1}^{k} \text{WIN}[\widehat{S}_i^B, \lambda] \geq \Big(\prod_{i=1}^{k} \text{WIN}[\widehat{S}_i^B, \lambda] \Big)^{\frac{1}{k}} \geq \Big(\text{WIN}[B, \lambda] \Big)^{\frac{1}{k}} = \epsilon'^{\frac{1}{k}}$$

We turn to prove Claim 1.

[4]The AM-GM inequality states that for any positive reals $a_1, \ldots, a_n$, it holds that $\frac{a_1 + \cdots + a_n}{n} \geq \sqrt[n]{a_1 \ldots a_n}$.

Proof of Claim 1: We start by providing some intuition. Consider a prover Q that for each i, behaves like $\widehat{S}_i^B$ in coordinate i. This new prover Q behaves independently in each coordinate, and hence the probability it wins in all coordinates is $\prod_{i=1}^k \textsc{win}[\widehat{S}_i^B, \lambda]$. The claim says that the success probability of Q is at least the success probability of B. Informally, this holds since $\widehat{S}_i^B$ does the best it can do in coordinate i with unbounded computation using complete information about B.

We proceed to a formal treatment by induction on the length of histories, starting at length m and ending at length 0. Given any history h^q, we show that $\prod_{i=1}^k \textsc{win}[\widehat{S}_i^B, h^q] \geq \textsc{win}[B, h^q]$.

Base case: $q = m$. Given a complete view h^m, $\widehat{S}_i^B$ wins in coordinate i, if B wins in all coordinates, i.e., $\textsc{win}[\widehat{S}_i^B, h^m]$ is 1, if $\textsc{win}[B, h^m] = 1$. Therefore, $\prod_{i=1}^k \textsc{win}[\widehat{S}_i^B, h^m] \geq \textsc{win}[B, h^m]$.

Induction step. Assume that the induction hypothesis holds for all length q histories. We show that it also holds for $q-1$ length histories. Let h^{q-1} be a $q-1$ length history. It holds that

$$\prod_{i=1}^k \textsc{win}[\widehat{S}_i^B, h^{q-1}] = \prod_{i=1}^k \left(\frac{1}{2^n} \sum_{r_i \in \{0,1\}^n} \textsc{win}[\widehat{S}_i^B, (h^{q-1}; i, r_i)] \right)$$

We expand this as a sum of product terms. Since every term can be expressed as $\prod_{i=1}^k \textsc{win}[\widehat{S}_i^B, (h^{q-1}; i, r_i)]$, where for each i, $r_i \in \{0,1\}^n$, we get

$$\prod_{i=1}^k \left(\frac{1}{2^n} \sum_{r_i \in \{0,1\}^n} \textsc{win}[\widehat{S}_i^B, (h^{q-1}; i, r_i)] \right) = \frac{1}{2^{nk}} \sum_{r_1,\ldots,r_k \in \{0,1\}^n} \left(\prod_{i=1}^k \textsc{win}[\widehat{S}_i^B, (h^{q-1}; i, r_i)] \right)$$

Since $\widehat{S}_i^{\widetilde{B}}$ always finds the *best* extension to any history $h^{q-1}; r_i$, it holds that for all $r_1, \ldots, r_k \in \{0,1\}^n$,

$$\textsc{win}[\widehat{S}_i^B, (h^{q-1}; i, r_i)] \geq \textsc{win}[\widehat{S}_i^B, (h^{q-1}; i, (r_1, \ldots, r_k))]$$

Therefore,

$$\frac{1}{2^{nk}} \sum_{r_1,\ldots,r_k \in \{0,1\}^n} \left(\prod_{i=1}^k \textsc{win}[\widehat{S}_i^B, (h^{q-1}; i, r_i)] \right) \geq \frac{1}{2^{nk}} \sum_{r_1,\ldots,r_k \in \{0,1\}^n} \left(\prod_{i=1}^k \textsc{win}[\widehat{S}_i^B, (h^{q-1}; i, (r_1, \ldots, r_k))] \right)$$

Since, by the induction hypothesis, the following holds for every q-length history h^q

$$\prod_{i=1}^k \textsc{win}[\widehat{S}_i^B, h^q] \geq \textsc{win}[B, h^q]$$

it follows that

$$\prod_{i=1}^k \textsc{win}[\widehat{S}_i^B, h^{q-1}] \geq \frac{1}{2^{nk}} \sum_{r_1,\ldots,r_k \in \{0,1\}^n} \textsc{win}[B, h^{q-1}; (r_1, \ldots, r_k)] = \textsc{win}[B, h^{q-1}]$$

This concludes the induction step and the proof of the claim.

5. PROOF OF MAIN THEOREM

In this section, we prove our main theorem. As in the information-theoretic case, the construction of S is generic (and relatively straight-forward); the main challenge is to bound its success probability.

THEOREM 5.1 (THEOREM 3.1 RESTATED). *Let (P, V) be a $2m$-round public-coin interactive argument for L. Then for any constants m and d, there exists a probabilistic oracle machine S, such that for every $x \notin L$, $\epsilon' > 0$, $0 < \delta < \epsilon'$, integer k, every deterministic interactive machine B, if*

$$\Pr\Big[\langle B, V_k\rangle(x) = 1\Big] \geq \epsilon'$$

then

$$\Pr\Big[\langle S^B(x, \delta, d), V\rangle(x) = 1\Big] \geq (\epsilon')^{\frac{1}{k}} - \frac{1}{|x|^d}$$

Furthermore, the running-time of S is bounded by a polynomial in $\frac{1}{\delta}, |x|$ and k.

5.1. Description Of S

On a high-level, S^B proceeds similarly to $\widehat{S}$ (described in the previous section). On input a statement x and given oracle access to a prover B, S picks a random coordinate $i \in [k]$ and invokes the machine S_i: S_i interacts with B while emulating the honest verifier strategy V for B in all executions except execution i; on the other hand, messages in execution i will be forwarded externally. Recall that, to generate the verifier messages for other executions, $\widehat{S}_i$ chooses messages that maximizes its success probability. Since S_i needs to run in polynomial-time, it will instead uniformly generate several sets of messages and evaluate whether these generated messages are "good" by running "look-aheads" of B. More precisely, given a history of messages h^q sent to B, S_i proceeds as follows:

(1) S_i starts by externally forwarding the output of B.
(2) Upon receiving back an answer r from the outside, S_i feeds r to B as part of its i'th execution (i.e. as part of its i'th coordinate). In order to do this, S_i must also feed to B messages for all $k-1$ other executions. S_i, thus carefully chooses a "good" extension h^{q+1} of the history $(h^q; r)$ where the new verifier message r is placed in the i'th coordinate. S_i will choose a good extension as follows:
   (a) It samples an appropriate number of random extensions of $(h^q; r)$ (by picking random messages as the $(q+1)^{st}$ verifier message in all $k-1$ executions).
   (b) For each such sample h^{q+1}, it then estimates the probability that S_i succeeds in convincing V given the history h^q. Again, this estimation is performed by sampling an appropriate number of executions and computing the average success probability of S_i.
   (c) Finally, the sample h^{q+1} with the highest estimate is selected.
(3) Next, given a "good" extension h^{q+1}, S_i now feeds the history h^{q+1} to B, and continues as in step 1 until the external execution has finished.

A formal description of S_i can be found in Figure 2.

5.2. Analyzing Success Probability Of S

We proceed to analyzing the success probability of S on input a machine B.

On a high-level, our analysis proceeds by constructing a sequence of "hybrid" simulators. The first intermediate simulator $\widetilde{S}_i$ proceeds just like S_i except that it computes true success probabilities, as opposed to estimating them through sampling as S_i would

Algorithm 2 : $S_i^B(h^q, \texttt{mode})$
Parameters : $c = d + \log(4km)/\log n$, $\eta = 2n^{c+1}/\epsilon'$ and $\zeta_q = \lceil 4^{2q} n^{2c} \log(4^q n^c \eta) \rceil$

1: Let $(c_1, \ldots, c_k)$ be the output of B when fed the messages in h^q
2: **if** $q = k$ **then**
3: **if** $\texttt{mode} = \texttt{mainthread}$ **then** externally forward c_i and halt.
4: **else** output 1 if B convinces V in coordinate i in history when fed with messages in h^q
5: **else**
6: **if** $\texttt{mode} = \texttt{mainthread}$ **then** externally forward c_i and let r be the message externally received **else** pick a random message r, i.e., $r \leftarrow \{0,1\}^n$
7: Pick η extensions ${h_1}^{q+1}, \ldots, {h_\eta}^{q+1}$ from $(h^q; i, r, U_n^{k-1})$.
8: **for** $j = 1$ to η **do**
9: Run $S_i^B({h_j}^{q+1}, \texttt{lookahead})$, ζ_{q+1} times. Let $\texttt{count}$ denote the number of times S_i^B outputs 1. Let $X_j \leftarrow \frac{\texttt{count}}{\zeta_{q+1}}$.
10: $j' \leftarrow \text{argmax}_{j \in [\eta]}\{X_j\}$.
11: Run $S_i^B({h_{j'}}^{q+1}, \texttt{mode})$.

Fig. 2. Description of the machine S_i^B

Symbol	Description
S	The actual simulator.
$\widehat{S}$	The computationally-unbounded simulator that chooses the best possible response.
$\widetilde{S}$	The computationally-unbounded simulator that computes exact success probabilities on polynomially many samples and picks the sample with highest probability.
B	The cheating prover that interacts with k parallel verifiers.
$\widehat{B}$	The cheating prover that follows B's strategy only on *heavy* histories.

Fig. 3. List of Hybrid Simulators and Provers

have done (see Step 2(b) in informal description of S_i). Using a Chernoff bound, we can show that the success probability of S_i^B and $\widetilde{S}_i^B$ are close. Next, we consider an unbounded simulator $\widehat{S}_i$ which proceeds in exactly the same way as the simulator used in the information-theoretic case. To make the game more fair, we compare the success probability of $\widetilde{S}_i^B$ with the success probability of $\widehat{S}_i^{\widetilde{B}}$ where $\widetilde{B}$ is a "weaker" version of B that proceeds just as B except that it *aborts* when fed views on which its success probability is "too" high. We show that $\widetilde{B}$ can be defined in such a way that 1) the success probability of $\widetilde{B}$ is still "high enough", yet 2) the success probability of $\widetilde{S}_i^B$ is "close" to the success probability of $\widehat{S}_i^{\widetilde{B}}$. Finally, we conclude the proof using the proof of the information-theoretic case (i.e., Lemma 4.1) to show that the success probability $\widehat{S}_i^{\widetilde{B}}$ is "high".

More formally, our analysis proceeds in three steps.

Step 1:. First, we consider an intermediate machine $\widetilde{S}_i$ that proceeds exactly as S_i, except that instead of estimating its own success probability when evaluating if a sample extension is "good", $\widetilde{S}_i$, computes its *actual* success probability. A formal

description of $\widetilde{S}_i$ can be found in Figure 4. We show that the success probability of S_i^B is close to the success probability of $\widetilde{S}_i^B$. Intuitively, this follows since by a Chernoff bound, the estimates computed by S_i^B will be "close enough" with high probability and when this happens S will select a sufficiently good continuation. More formally, we prove the following claim in Section 6.

CLAIM 2. *For any interactive machine B and any $i \in [k]$, it holds that*

$$\text{WIN}[S_i^B, \lambda] \geq \text{WIN}[\widetilde{S}_i^B, \lambda] - \frac{1}{n^c}$$

Step 2:. Next, we compare the success probability of $\widetilde{S}_i$ with the unbounded machine $\widehat{S}_i$ defined in the previous section (Figure 1). Since $\widetilde{S}_i$ (just as S_i) only uses a polynomial number of samples to determine what extension to pick, we can never expect it to find the *actual* best extension (as $\widehat{S}_i$ does).

We proceed to formally defining the (computationally unbounded) prover $\widetilde{B}$. Towards this goal we start by introducing some additional notation.

Given a length q history h^q, we denote by $Hvy_i(h^q; i, r)$ the subset of the valid i-extensions of $(h^q; i, r)$ containing the α fraction of extensions on which $\widetilde{S}_i$ has the highest probability of winning, where $\alpha = \frac{\epsilon'}{n^c}$ and ϵ' is the success probability of B. The extensions in $Hvy_i(h^q; i, r)$ are called *heavy* for i (as these are the "best" extensions for $\widetilde{S}_i$) and the remaining ones *light* for i. Let

$$Hvy(h^{q-1}) = \bigcup_{i \in [k], r \in \{0,1\}^n} Hvy_i(h^{q-1}; i, r)$$

We call histories in $Hvy(h^{q-1})$ *heavy* and the remaining ones *light*.

Algorithm 3 : $\widetilde{S}_i^B(h^q)$
`Parameters` : $c = d + \log(4mk)/\log n$, $\eta = 2n^{c+1}/\epsilon'$

1: Let $(c_1, \ldots, c_k)$ be the output of B when fed the messages in h^q
2: **if** $q = k$ **then**
3: Externally forward c_i and halt.
4: **else**
5: Externally forward c_i and let r be the message externally received.
6: Pick η extensions ${h_1}^{q+1}, \ldots, {h_\eta}^{q+1}$ from $(h^q; i, r, U_n^{k-1})$.
7: Let j' be the sample on which $\widetilde{S}_i^B$ on input ${h_{j'}}^{q+1}$ wins with highest probability.
8: Run $\widetilde{S}_i^B({h_{j'}}^{q+1})$.

Fig. 4. Description of the machine $\widetilde{S}_i^B$

We next proceed to defining $\widetilde{B}$. At a high level, all that $\widetilde{B}$ does is to abort whenever it receives a *heavy* view. Formally, at any stage if $h^q \in Hvy(h^{q-1})$, then $\widetilde{B}$ returns $\perp$ and terminates the protocol[5]. We have the following claims:

[5]We here assume without loss of generality that the honest verifier V always rejects whenever it receives the $\perp$ symbol.

CLAIM 3. *Let B be an interactive machine such that* $\text{WIN}[B, \lambda] \geq \epsilon'$. *Then*

$$\text{WIN}[\widetilde{B}, \lambda] \geq \epsilon' \left(1 - \frac{km}{n^c}\right)$$

PROOF. Recall that prover $\widetilde{B}$ acts exactly as prover B on all histories that are *light*. On the other hand, on *heavy* histories, $\widetilde{B}$ returns $\perp$ and thus by our assumption on the proof system (P, V), $\widetilde{B}$ always loses. Consequently, to compare the success probabilities of B and $\widetilde{B}$, we upper bound the fraction of histories that are *heavy*. Given a length q history h^q, coordinate i and string r, it follows by definition that the fraction of tuples in $(h^q; i, r, U_n^{k-1})$ that are *heavy* for i is α, i.e., $\frac{|Hvy_i(h^{q-1};i,r)|}{|(h^{q-1};i,r,U_n^{k-1})|} = \alpha$. This means that the total fraction of extensions among all strings r that are *heavy* for i is α since

$$\frac{\sum_{r\in\{0,1\}^n} |Hvy_i(h^{q-1}; i, r)|}{|(h^{q-1}; i, r, U_n^{k-1})|} = \frac{\sum_{r\in\{0,1\}^n} |Hvy_i(h^{q-1}; i, r)|}{\sum_{r\in\{0,1\}^n} |(h^{q-1}; i, r, U_n^{k-1})|} = \alpha$$

where the last equality follows since $|Ext_i(h^{q-1}; r)|$ is the same for all r. Since $Hvy(h^{q-1}) = \cup_{i\in[k]} \left(\cup_{r\in\{0,1\}^n} Hvy_i(h^{q-1}; i, r)\right)$, it follows by the union bound that the fraction of extensions of h^q that are *heavy* (for any coordinate i) is bounded by $k\alpha$; thus the probability that $\widetilde{B}$ abort in any given round is bounded by $k\alpha$; another application of the union bound concludes that the probability that $\widetilde{B}$ aborts at all is bounded by $km\alpha$. Therefore,

$$\text{WIN}[\widetilde{B}, \lambda] \geq \text{WIN}[B, \lambda] - km\alpha = \epsilon'\Big(1 - \frac{km}{n^c}\Big)$$

■

CLAIM 4. *For any interactive machine B and any $i \in [k]$, it holds that*

$$\text{WIN}[\widetilde{S}_i^B, \lambda] \geq \text{WIN}[\widehat{S}_i^{\widetilde{B}}, \lambda] - e^{-n}$$

The proof of this claim is found in Section 6; on a high-level, the claim follows from the fact that with high probability $\widetilde{S}^B$ will "hit" a *heavy* extension, and will thus perform better than $\widehat{S}^{\widetilde{B}}$.

Step 3:. Combining Claims 2-4 and Lemma 4.1 (from the analysis in the information theoretic-case), we have that the success probability of S is

$$\begin{aligned}
\textsc{win}[S^B, \lambda] &= \frac{1}{k}\sum_{i=1}^{k} \textsc{win}[S_i^B, \lambda] \\
&\geq \frac{1}{k}\sum_{i=1}^{k} \textsc{win}[\widetilde{S}_i^B, \lambda] - \frac{1}{n^c} && \text{(Using Claim 2)} \\
&\geq \frac{1}{k}\sum_{i=1}^{k} \textsc{win}[\widehat{S}_i^{\widetilde{B}}, \lambda] - \frac{1}{n^c} - e^{-n} && \text{(Using Claim 4)} \\
&= \textsc{win}[\widehat{S}^{\widetilde{B}}, \lambda] - \frac{1}{n^c} - e^{-n} \\
&\geq \left(\textsc{win}[\widetilde{B}, \lambda]\right)^{\frac{1}{k}} - \frac{1}{n^c} - e^{-n} && \text{(Using Lemma 4.1)} \\
&\geq \left(\epsilon'\left(1 - \frac{km}{n^c}\right)\right)^{\frac{1}{k}} - \frac{1}{n^c} - e^{-n} && \text{(Using Claim 3)} \\
&\geq (\epsilon')^{\frac{1}{k}} - \frac{2m}{n^c} - \frac{1}{n^c} - e^{-n} \\
&\geq (\epsilon')^{\frac{1}{k}} - \frac{1}{n^d}
\end{aligned}$$

The last two steps are obtained by first using the inequality[6] $(1-x)^b > 1 - 2bx$ (this holds whenever $b \in [0,1]$ and $x \in [0, \frac{1}{2}]$) and then using the fact that $c = d + \log(4mk)/\log n$.

5.3. Analyzing Running Time Of S

CLAIM 5. *The running time of S is polynomial in $\frac{1}{\epsilon'}, |x|$ and k.*

PROOF. Recall that S^B runs S_i^B (on the empty history) for a particular i. If T is the number of oracle queries made by S, then running time of S (assuming unit time for oracle queries) is bounded by $T \times \mathrm{poly}(|x|, k)$ since simulating verifier messages for B is polynomial in k and $|x|$. It thus suffices to determine the number of oracle queries made by S_i^B. Let $n = |x|$ and $T(q)$ be the number of oracle queries made by S_i^B on any history h^q of length q. We show inductively that

$$T(q) \leq (2\eta)^{m-q} \prod_{j=q+1}^{m} \zeta_j$$

A bound on the number of queries is obtained by evaluating $T = T(0)$ and setting

$$c = d + \frac{\log(4mk)}{\log n},\ \zeta_q = \lceil 4^{2q} n^{2c} \log\left(4^q n^c \eta\right)\rceil \text{ and } \eta = \frac{2n^{c+1}}{\epsilon'}$$

(as in the definition of S_i). The proof of the claim then follows by observing that m and d are constants. We now proceed bound $T(q)$.

Recall that, S_i^B on input h^q, first makes an oracle query (see step (1) in Figure 2). Then, it samples η extensions of h^q. For each sample ${h_j}^{q+1}$ (all of which are of length

[6]Let $f(x) = (1-x)^b$ and $g(x) = 1 - 2bx$. We have that for any $0 < b < 1$, $f(0) = g(0) = 1$ and $g'(x) < f'(x) < 0$ in the interval $(0, \frac{1}{2})$. Therefore $g(x) < f(x)$ in the same interval.

$q+1$), it recursively runs S_i^B on input ${h_j}^{q+1}$, ζ_{q+1} times. Finally, in the last step, it runs S_i^B on a history of length $q+1$. Thus, the total number of recursive calls made on histories of length $q+1$ is $\eta\zeta_{q+1}+1$. Therefore, including the first oracle query the total number of queries made by S_i^B on a history of length q is given by,

$$T(q) = 1 + (\eta\zeta_{q+1} + 1)T(q+1)$$

Since $\eta > 2$, it follows that

$$T(q) \leq 2\eta\zeta_{q+1}T(q+1) \leq (2\eta)^{m-q} \prod_{j=q+1}^{m} \zeta_j$$

■

6. PROOF OF CLAIMS

We here provide the formal proofs of Claim 2 and 4. First, we show the following simple observation which is useful in the proof of Claim 2.

An observation on sampling. Suppose we have n events with probabilities $y_1, \ldots, y_n$ respectively. Given n "good" estimates $x_1, \ldots, x_n$ of the true probabilities $y_1, \ldots, y_n$, we wish to select the event i^* with the highest true probability. The following simple observation states that if simply selecting the event i' that has the highest estimate $y_{i'}$, it holds that the true success probability $x_{i'}$ of i' is "close" to the success probability x_{i^*} of the optimal event i^*.

OBSERVATION 1. *Consider any two sequences,* $X = (x_1 \ldots, x_n)$ *and* $Y = (y_1, \ldots, y_n)$ *such that for all* $i \in [n]$, $|x_i - y_i| < \delta$. *Then, for* $i' = \text{argmax}_i\{x_i\}$ *it holds that* $y_{i'} \geq \max_i\{y_i\} - 2\delta$.

Proof: We are given that $|x_i - y_i| < \delta$ for all i. Hence, for all i we have that

$$x_i \geq y_i - \delta \text{ and } y_i \geq x_i - \delta$$

Thus,

$$y_{i'} \geq x_{i'} - \delta = \max_i\{x_i\} - \delta \geq \max_i\{y_i - \delta\} - \delta = \max_i\{y_i\} - 2\delta$$

■

6.1. Proof Of Claim 2

Recall that we are required to prove that $\text{WIN}[S_i^B, \lambda] \geq \text{WIN}[\widetilde{S}_i^B, \lambda] - \frac{1}{n^c}$. We will show by induction that for all q, h^q and $(h^{q-1}; r)$,

$$\text{WIN}[S_i^B, h^q] \;\geq\; \text{WIN}[\widetilde{S}_i^B, h^q] - \frac{1}{4^q n^c}$$

$$\text{WIN}[S_i^B, (h^{q-1}; i, r)] \;\geq\; \text{WIN}[\widetilde{S}_i^B, (h^{q-1}; i, r)] - \frac{1}{4^{q-1} n^c}$$

We then conclude Claim 2 by setting $q = 0$.

Base case: $q = m$. Given a complete view h^m, both S_i^B and $\widetilde{S}_i^B$ win exactly when B wins on h^m in coordinate i. Therefore, the base case is true.

Induction step. There are two parts to proving the induction step: either the history is of the form h^q, or of the form $(h^{q-1}; i, r)$. First, consider the case when h^q is the history at a *verifier*-step. By the induction hypothesis, for every r we have that

$$\text{WIN}[S_i^B, (h^q; i, r)] \geq \text{WIN}[\widetilde{S}_i^B, (h^q; i, r)] - \frac{1}{4^q n^c}$$

From the definition of $\text{WIN}[,]$, we know that,

$$\text{WIN}[S_i^B, h^q] = \frac{1}{2^n} \sum_{r \in \{0,1\}^n} \text{WIN}[S_i^B, (h^q; i, r)]$$

Thus,

$$\text{WIN}[S_i^B, h^q] \geq \text{WIN}[\widetilde{S}_i^B, h^q] - \frac{1}{4^q n^c}$$

Next, consider the case when $(h^{q-1}; i, r)$ is the history at a *prover*-step. We here need to compare the probabilities $\text{WIN}[S_i^B, (h^{q-1}; i, r)]$ and $\text{WIN}[\widetilde{S}_i^B, (h^{q-1}; i, r)]$. Recall that both S_i^B and $\widetilde{S}_i^B$ pick η samples that are extensions of $h^{q-1}; r$. Given these η samples, $\widetilde{S}_i^B$ chooses the sample which maximizes its probability of winning. On the other hand, since S_i^B is computationally bounded, it cannot efficiently find the one that maximizes its success probability; instead it estimates the success probability in each of these samples by sampling and then chooses the one that has the maximum estimate. Given an extension h^q of $(h^{q-1}; i, r)$, let the random variable $\text{EST}[S_i^B, h^q]$ denote the estimate computed by S_i^B for h^q. We have,

$$\begin{aligned}
&\text{WIN}[S_i^B, (h^{q-1}; i, r)] \\
&\quad = \frac{1}{2^{(k-1)n\eta}} \times \sum_{\substack{h_z{}^q \in Ext_i(h^{q-1}; r) \\ 1 \leq z \leq \eta}} E\Big[j' \leftarrow \operatorname*{argmax}_{j \in [\eta]} \{\text{EST}[S_i^B, h_j{}^q]\} : \text{WIN}[S_i^B, h_{j'}{}^q]\Big]
\end{aligned}$$

$$\text{WIN}[\widetilde{S}_i^B, (h^{q-1}; i, r)] = \frac{1}{2^{(k-1)n\eta}} \times \sum_{\substack{h_z{}^q \in Ext_i(h^{q-1}; r) \\ 1 \leq z \leq \eta}} \max_{j \in [\eta]} \{\text{WIN}[\widetilde{S}_i^B, h_j{}^q]\}$$

To show the induction step, it thus suffices to show that given any set of samples $(h_1{}^q, \ldots, h_\eta{}^q)$,

$$E\Big[j' \leftarrow \operatorname*{argmax}_{j \in [\eta]} \{\text{EST}[S_i^B, h_j{}^q]\} : \text{WIN}[S_i^B, h_{j'}{}^q]\Big] \geq \max_{j \in [\eta]} \{\text{WIN}[\widetilde{S}_i^B, h_j{}^q]\} - \frac{1}{4^{q-1} n^c} \tag{1}$$

By the induction hypothesis it follows that for every j,

$$\text{WIN}[S_i^B, h_j{}^q] \geq \text{WIN}[\widetilde{S}_i^B, h_j{}^q] - \frac{1}{4^q n^c}$$

Therefore,

$$\max_{j \in [\eta]} \{\text{WIN}[S_i^B, h_j{}^q]\} \geq \max_{j \in [\eta]} \{\text{WIN}[\widetilde{S}_i^B, h_j{}^q]\} - \frac{1}{4^q n^c} \tag{2}$$

Below we will show that,

$$E\Big[j' \leftarrow \operatorname*{argmax}_{j \in [\eta]} \{\text{EST}[S_i^B, h_j{}^q]\} : \text{WIN}[S_i^B, h_{j'}{}^q]\Big] \geq \max_{j \in [\eta]} \{\text{WIN}[S_i^B, h_j{}^q]\} - \frac{3}{4^q n^c} \tag{3}$$

Thus, combining equations (2) and (3) we prove equation (1) and the induction step follows. We proceed to showing equation (3).

Recall that for each sample $h_j{}^q$, S_i^B picks ζ_q samples to compute $\text{EST}[S_i^B, h_j{}^q]$. By construction,

$$E[\text{EST}[S_i^B, h_j{}^q]] = \text{WIN}[S_i^B, h_j{}^q]$$

We say that an estimate X for ${h_j}^q$ is "good", if

$$\left|X - \textsc{win}[S_i^B, {h_j}^q]\right| \leq \frac{1}{4^q n^c}$$

Otherwise we call the estimate X "bad" for ${h_j}^q$. From Observation 1 it follows that, conditioned on all estimates being "good", the expectation of $\left\{j' \leftarrow \operatorname{argmax}_{j\in[\eta]}\{\textsc{est}[S_i^B, {h_j}^q]\} : \textsc{win}[S_i^B, {h_{j'}}^q]\right\}$ is at least

$$\textsc{win}[S_i^B, {h_j}^q] - \frac{2}{4^q n^c}$$

We show below that the probability of some estimate being "bad" is at most $\frac{1}{4^q n^c}$. Therefore

$$\begin{aligned} E\Big[j' \leftarrow \operatorname*{argmax}_{j\in[\eta]}\{\textsc{est}[S_i^B, {h_j}^q]\} : \textsc{win}[S_i^B, {h_{j'}}^q]\Big] & \\ \geq \left(1 - \frac{1}{4^q n^c}\right)\left(\max_{j\in[\eta]}\{\textsc{win}[S_i^B, {h_j}^q]\} - \frac{2}{4^q n^c}\right) & \\ \geq \max_{j\in[\eta]}\{\textsc{win}[S_i^B, {h_j}^q]\} - \frac{3}{4^q n^c} & \end{aligned}$$

and this concludes the proof of Equation (3) and the induction step. It only remains to bound the probability of some estimate being "bad".

Using Chernoff bound, we have that each sample $\textsc{est}[S_i^B, {h_j}^q]$ is bad with probability at most $2^{-\frac{\zeta_q}{n^{2c}4^{2q}}}$. Therefore, using the union bound, the probability that at least one of the samples is "bad" is at most

$$\eta \cdot 2^{-\frac{\zeta_q}{n^{2c}4^{2q}}} \leq \frac{1}{4^q n^c}$$

6.1.1. Proof Of Claim 4. Recall that we are required to prove that $\textsc{win}[\widetilde{S}_i^B, \lambda] \geq \textsc{win}[\widehat{S}_i^{\widetilde{B}}, \lambda] - e^{-n}$. We will show by induction that $\widetilde{S}$ loses at most by $(1-\alpha)^M$ in each *prover*-step, where $\alpha = \frac{\epsilon'}{n^c}$, i.e., for all q, h^q and $(h^{q-1}; i, r)$,

$$\begin{aligned} \textsc{win}[\widetilde{S}_i^B, h^q] &\geq \textsc{win}[\widehat{S}_i^{\widetilde{B}}, h^q] - (m-q)(1-\alpha)^\eta \\ \textsc{win}[\widetilde{S}_i^B, (h^{q-1}; i, r)] &\geq \textsc{win}[\widehat{S}_i^{\widetilde{B}}, (h^{q-1}; i, r)] - (m-q+1)(1-\alpha)^M \end{aligned}$$

Since $M = \frac{2n^{c+1}}{\epsilon'} = 2n\frac{1}{\alpha} > (n + \log m)\frac{1}{\alpha}$, setting $q = 0$, the proof of Claim 4 follows.

Base case: $q = m$. At the final *verifier*-step, either the verifier accepts or rejects. Given a complete history h^m, we know that $\widetilde{S}_i^B$ wins whenever B wins in coordinate i on h^m and $\widehat{S}_i^{\widetilde{B}}$ wins whenever $\widetilde{B}$ wins in coordinate i on h^m. Since $\widetilde{B}$ is strictly a weaker prover than B, the base case follows.

Induction step. As in Claim 2, there are two parts to proving the induction step: either the history is of the form h^q, or of the form $(h^{q-1}; i, r)$. We first consider the case when h^q is the history at a *verifier*-step. The induction step in this case follows exactly as in Claim 2. Next, we consider the case when $h^{q-1}; r$, is a history at a *prover*-step. Towards the goal of proving the induction step, let us first note that for any *light* extension h^q of $(h^{q-1}; i, r)$,

$$\textsc{win}[\widetilde{S}_i^B, (h^{q-1}; i, r)] \geq \textsc{win}[\widetilde{S}_i^B, h^q] - (1-\alpha)^\eta \tag{4}$$

Intuitively, this follows since except with probability $(1-\alpha)^\eta$, $\widetilde{S}_i^B$, on input $(h^{q-1};i,r)$ finds a *heavy* extension in which case its success probability is higher than $\text{WIN}[\widetilde{S}_i^B, h^q]$. More formally, in a *prover*-step, $\widetilde{S}_i^B$ samples η extensions of $(h^{q-1};r)$. Among the η samples, $\widetilde{S}_i^B$ picks the extension that maximizes its probability of winning. Since h^q is a *light* extension, $\widetilde{S}_i^B$ on $h^{q-1};r$ will succeed with probability at least $\text{WIN}[\widetilde{S}_i^B, h^q]$ if any of the η samples of $\widetilde{S}_i^B$ hits a *heavy* extension (since by definition of *heavy* extensions, the success probability of $\widetilde{S}_i^B$ on *heavy* extensions is at least as high as on any *light* extension). Since a sample is *heavy* with probability α, the probability that all M samples are bad is $(1-\alpha)^\eta$ and therefore using the union bound, we have that

$$\begin{aligned}\text{WIN}[\widetilde{S}_i^B, (h^{q-1};i,r)] = &\geq \text{WIN}[\widetilde{S}_i^B, h^q] - (1-\alpha)^\eta \\ &\geq \text{WIN}[\widehat{S}_i^{\widetilde{B}}, h^q] - (m-q)(1-\alpha)^\eta - (1-\alpha)^\eta \\ &= \text{WIN}[\widehat{S}_i^{\widetilde{B}}, h^q] - (m-q+1)(1-\alpha)^\eta\end{aligned}$$

where the second inequality follows by the induction hypothesis. Since Equation 4 holds for any *light* extension h^q of $h^{q-1};r$, and in particular holds for the *light* extension that maximizes the probability of $\widehat{S}_i^{\widetilde{B}}$ winning, we have that

$$\text{WIN}[\widetilde{S}_i^B, (h^{q-1};i,r)] \geq \text{WIN}[\widehat{S}_i^{\widetilde{B}}, (h^{q-1};i,r)] - (m-q+1)(1-\alpha)^\eta$$

This concludes the induction step. ∎

7. PROOF OF THEOREM 2

THEOREM 7.1 (THEOREM 3.2 RESTATED). *Let (P,V) be a constant-round public-coin interactive argument for the language L with soundness error $\epsilon(\cdot)$. Then for every polynomial $k(\cdot)$, (P_k,V_k) has soundness error $\epsilon'(\cdot)$ where $\epsilon(n) = \epsilon(n)^{k(n)} + \nu(n)$ and ν is a negligible function.*

PROOF. Assume for contradiction that there is a polynomial $p(\cdot)$ and a non-uniform probabilistic polynomial-time machine B, such that for infinitely many $x \notin L$, it holds that

$$\text{WIN}[B,\lambda] \geq \epsilon(|x|)^k + \frac{1}{p(|x|)} \tag{5}$$

We may assume without loss of generality that B is deterministic as it may always get its "best" random coins as non-uniform advice. By Theorem 3.1, for every constant d, there is a probabilistic oracle machine S such that for every x that satisfies (5), it holds that

$$\text{WIN}[S^B,\lambda] \geq \left(\epsilon(|x|)^k + \frac{1}{p(|x|)}\right)^{\frac{1}{k}} - \frac{1}{n^d}$$

By picking d sufficiently large, we have the right hand side of the equation is bigger than $\epsilon(|x|)$, which contradicts the soundness of (P,V).

∎

8. PROOF OF THEOREM 3

THEOREM 8.1 (THEOREM 3.3 RESTATED). *Let (P,V) be a constant-round public-coin proof of knowledge for the witness relation R with knowledge error $\epsilon(\cdot)$. Then*

for every integer d, and every polynomial $k(\cdot)$, (P_k, V_k) has knowledge error $\epsilon'(\cdot)$ where $\epsilon'(n) = \epsilon(n)^{k(n)} + \frac{1}{n^d}$.

PROOF. Let $k(\cdot)$ be a polynomial and d an integer. We construct a knowledge extractor K' for (P_k, V_k) that has knowledge error $\epsilon'(\cdot)$. Let K be the knowledge extractor for (P, V) and S be the oracle machine corresponding to the public-coin interactive argument (P, V) guaranteed by Theorem 3.1.

The knowledge-extractor K' proceeds as follows: On input x and oracle access to a deterministic prover B, K' runs K with oracle access to $S^B(x, \frac{1}{n^d}, c)$ where $n = |x|$ and c is set to $d + \log(k(n))/\log n$, and outputs whatever K outputs. We will show that the knowledge error of K' is $\epsilon^{k(n)} + \frac{1}{n^d}$ where $\epsilon = \epsilon(n)$. Let $\mu = \Pr\Big[\langle B, V_k\rangle(x) = 1]$.

We need to show that whenever $\mu > \epsilon^{k(n)} + \frac{1}{n^d}$, the expected running time of K' is bounded by a $\frac{q'(n)}{\delta'}$ for some polynomial $q'(\cdot)$

$$\delta' = \Pr[\langle B, V_k\rangle(x) = 1] - \left(\epsilon^{k(n)} + \frac{1}{n^d}\right) = \mu - \epsilon^{k(n)} - \frac{1}{n^d} \tag{6}$$

Furthermore K' always outputs a valid witness in this case. For the remainder of the proof, we restrict to the case that $\delta' > 0$, i.e. $\mu > \epsilon^{k(n)} + \frac{1}{n^d}$.

Let us first note that $\delta' \leq \delta poly(n)$ where $\delta = \Pr[\langle S^B(x, \frac{1}{n^d}, c), V\rangle(x) = 1] - \epsilon$ and therefore $\delta > 0$. Since, $0 < \frac{1}{n^d} < \mu$, by Theorem 3.1, we have that $\Pr[\langle S^B(x, \frac{1}{n^d}, c), V\rangle(x) = 1] \geq \mu^{\frac{1}{k(n)}} - \frac{1}{n^c}$. It follows that

$$\delta \geq \mu^{\frac{1}{k(n)}} - \frac{1}{n^c} - \epsilon \geq \frac{\mu - \epsilon^{k(n)}}{k(n)} - \frac{1}{n^c} = \frac{\delta'}{k(n)}$$

where the second inequality follows from the identity $(\alpha^k - \beta^k) = (\alpha - \beta)(\alpha^{k-1} + \alpha^{k-2}\beta + \cdots + \beta^{k-1})$, by setting $\alpha = \mu^{\frac{1}{k(n)}}$ and $\beta = \epsilon$ and observing that $0 < \epsilon^{k(n)} < \mu < 1$. The last equality follows from our choice of c.

Since $\delta' > 0$ it directly follows that K' always outputs a valid witness (since K will always do so when $\delta > 0$). Additionally, from the definition of K', it follows that the running time of K' with prover B is equal to the time taken for K' to simulate K with oracle access to S^B. Therefore, the expected running time of K' can be bound by the product of the expected running time of K and the running time of S^B. Since K is a knowledge extractor for (P, V), there exists a polynomial $q(\cdot)$ such that the expected running time of K (assuming unit time for oracle queries) with oracle access to S^B is bounded by $\frac{q(n)}{\delta}$ and therefore is also bounded by $\frac{q(n)k(n)}{\delta'}$. Additionally, since $0 < \frac{1}{n^d} < \mu$, by Theorem 3.1, the running-time of S (assuming unit time for oracle queries to B) is bounded by a polynomial in n, n^d and $k(n)$, which all are polynomial in n. Hence, there exists a polynomial q' such that the running time of K' is bounded by $\frac{q'(n)}{\delta'}$. This concludes the proof. ∎

Remark. We mention that the proof of Theorem 3 relies on Theorem 1 only for the existence of an efficient parallel-repetition theorem with a black-box proof of security. In particular, our construction does not require optimal reduction of soundness error or the protocol to have a constant number of rounds, and is thus generally applicable also to other parallel-repetition theorems, such as [Håstad et al. 2010; Chung and Liu 2010].

ACKNOWLEDGMENTS

We wish to thank Moni Naor for suggesting this problem to us (a few years back), and Douglas Wikström, for a conversation that reminded us of it.

REFERENCES

BABAI, L. AND MORAN, S. 1988. Arthur-merlin games: a randomized proof system, and a hierarchy of complexity class. *Journal of Computer and System Sciences 36,* 2, 254–276.

BELLARE, M. AND GOLDREICH, O. 1992. On defining proofs of knowledge. In *CRYPTO*. 390–420.

BELLARE, M., IMPAGLIAZZO, R., AND NAOR, M. 1997. Does parallel repetition lower the error in computationally sound protocols? In *Foundations of Computer Science*. 374–383.

BEN-OR, M., GOLDWASSER, S., KILIAN, J., AND WIGDERSON, A. 1988. Multi-prover interactive proofs: How to remove intractability assumptions. In *Symposium on Theory of Computing*. 113–131.

BLUM, M. 1986. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*.

BRASSARD, G., CHAUM, D., AND CRÉPEAU, C. 1988. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences 37,* 2, 156–189.

CANETTI, R., HALEVI, S., AND STEINER, M. 2005. Hardness amplification of weakly verifiable puzzles. In *Theory of Cryptography Conference*. 17–33.

CHUNG, K.-M. AND LIU, F.-H. 2010. Parallel repetition theorems for interactive arguments. In *Theory of Cryptography Conference*. 19–36.

CHUNG, K.-M., LIU, F.-H., LU, C.-J., AND YANG, B.-Y. 2010. Efficient string-commitment from weak bit-commitment. In *ASIACRYPT*. 268–282.

FEIGE, U., FIAT, A., AND SHAMIR, A. 1988. Zero-knowledge proofs of identity. *Journal of Cryptology 1,* 2, 77–94.

FEIGE, U. AND KILIAN, J. 2000. Two-prover protocols - low error at affordable rates. *SIAM Journal on Comput. 30,* 1, 324–346.

FEIGE, U. AND SHAMIR, A. 1989. Zero knowledge proofs of knowledge in two rounds. In *CRYPTO*. 526–544.

FORTNOW, L., ROMPEL, J., AND SIPSER, M. 1994. On the power of multi-prover interactive protocols. *Theoretical Computer Science 134,* 2, 545–557.

GOLDREICH, O. 1998. *Modern Cryptography, Probabilistic Proofs, and Pseudorandomness*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

GOLDREICH, O. 2001. *Foundations of Cryptography — Basic Tools*. Cambridge University Press.

GOLDREICH, O., IMPAGLIAZZO, R., LEVIN, L. A., VENKATESAN, R., AND ZUCKERMAN, D. 1990. Security preserving amplification of hardness. In *Foundations of Computer Science*. 318–326.

GOLDREICH, O. AND KRAWCZYK, H. 1996. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing 25,* 1, 169–192.

GOLDREICH, O., MICALI, S., AND WIGDERSON, A. 1991. Proofs that yield nothing but their validity for all languages in np have zero-knowledge proof systems. *Journal of the ACM 38,* 3, 691–729.

GOLDWASSER, S. AND MICALI, S. 1984. Probabilistic encryption. *Journal of Computer and System Sciences 28,* 2, 270–299.

GOLDWASSER, S., MICALI, S., AND RACKOFF, C. 1989. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing 18,* 1, 186–208.

GOLDWASSER, S., MICALI, S., AND RIVEST, R. L. 1988. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing 17,* 2, 281–308.

HAITNER, I. 2009. A parallel repetition theorem for any interactive argument. In *Foundations of Computer Science*. 241–250.

HÅSTAD, J., PASS, R., WIKSTRÖM, D., AND PIETRZAK, K. 2010. An efficient parallel repetition theorem. In *Theory of Cryptography Conference*. 1–18.

HOLENSTEIN, T. AND SCHOENEBECK, G. 2011. General hardness amplification of predicates and puzzles - (extended abstract). In *TCC*. 19–36.

IMPAGLIAZZO, R., JAISWAL, R., AND KABANETS, V. 2007. Chernoff-type direct product theorems. In *CRYPTO*. 500–516.

JUTLA, C. S. 2010. Almost optimal bounds for direct product threshold theorem. In *TCC*. 37–51.

PANDEY, O., PASS, R., AND VAIKUNTANATHAN, V. 2008. Adaptive one-way functions and applications. In *CRYPTO*. 57–74.

PASS, R., TSENG, W.-L. D., AND WIKSTRÖM, D. 2009. On the composition of public-coin zero-knowledge protocols. In *CRYPTO*. 160–176.

PASS, R. AND VENKITASUBRAMANIAM, M. 2007. An efficient parallel repetition theorem for arthur-merlin games. In *Symposium on Theory of Computing*. 420–429.

PIETRZAK, K. AND WIKSTRÖM, D. 2007. Parallel repetition of computationally sound protocols revisited. In *Theory of Cryptography Conference*. 86–102.

RAZ, R. 1998. A parallel repetition theorem. *SIAM Journal on Computing 27,* 3, 763–803.

RICHARDSON, R. AND KILIAN, J. 1999. On the concurrent composition of zero-knowledge proofs. In *Eurocrypt*. 415–432.

TOMPA, M. AND WOLL, H. 1987. Random self-reducibility and zero knowledge interactive proofs of possession of information. In *Foundations of Computer Science*. 472–482.

XIAO, D. 2011. (nearly) round-optimal black-box constructions of commitments secure against selective opening attacks. In *Proc. 8th TCC*. 541–558.

YAO, A. C.-C. 1982. Theory and applications of trapdoor functions (extended abstract). In *Foundations of Computer Science*. 80–91.