An Efficient Parallel Repetition Theorem for Arthur-Merlin Games

Rafael Pass
Cornell University
rafael@cs.cornell.edu

Muthuramakrishnan Venkitasubramaniam
Cornell University
vmuthu@cs.cornell.edu

ABSTRACT

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**]: Interactive and reactive computation

General Terms

Theory

Keywords

parallel repetition, computationally-sound arguments, public-coin protocols, Arthur-Merlin games, proofs of knowledge

1. INTRODUCTION

Interactive proof system were introduced independently by Goldwasser, Micali and Rackoff [17] and Babai and Moran [1]. 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 e.g. [1]) 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 systems with soundness error ϵ^k. In other words, the probability that a cheating prover succeeds in convincing k independent parallel executions of V, of a false statement, is bounded by ϵ^k.

STOC'07, June 11–13, 2007, San Diego, California, USA.

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 [5] 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 [9, 12, 23, 3].

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 [2] 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 recently obtained in [6]). On the other hand, [2] also shows that there exists *4-round* interactive arguments where parallel repetition does not reduce the soundness error.[1] More

[1]Pedantically, the lower-bounds of [2], as well as subsequent results, 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.

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. Very recently, the result of Pietrzak and Wikstrom extends the results of [2] to show the existence of a constant-round protocol for which arbitrary parallel-repetition does not reduce the soundness error ([20]).

Nevertheless, although severe negative results were shown in [2] regarding parallel-repetition of interactive arguments, Bellare et al note that their 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 [3] (see also [13]) 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 *efficient* reductions. 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 [2]). 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.

We mention that most "classical" argument, or proof of knowledge, protocols (e.g. [19, 4]) 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 for proofs of knowledge) in these particular protocols, as far as we know, our results present the first *general* theorems for showing this.

Our techniques To prove our results we show a black-box reduction S which 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 becomes hard 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 [21]). 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 (quite non-standard) hybrid experiments (as in [15])– this allows us to, step-by-step, reduce the problem to an information-theoretic parallel-repetition question (which becomes simple to analyze).

Open questions 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). This leaves open the question of whether parallel repetition reduces the soundness error in public-coin arguments (or proofs of knowledge) with a *non-constant* number of communication 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. [24, 7]). The question investigated is also related to the large body of work on parallel repetition of *multi-prover games* (e.g., [8, 10, 11]), and in particular Raz' repetition theorem [22]. However, we mention that both our techniques and desiderata (i.e., efficiency of the reduction) are very different.

We finally mention that parallel repetition in the context of zero-knowledge proofs [17] has also received a lot attention. For instance, Goldreich and Krawczyk [14] 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 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 provide an overview of the proof of our main theorem. Details of the proof are found in Section 5.

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* [17] (ITM for brevity) and a *protocol* (in essence a pair of ITMs[3]).

[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.

[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.

Probabilistic notation The following probabilistic notation follows almost verbatim [18]. 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 $x_1 \stackrel{R}{\leftarrow} S_1;\ x_2 \stackrel{R}{\leftarrow} S_1;\ \ldots$ If S, T, $\ldots$ 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 [17] and arguments [5]. 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 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:*

- Completeness: *For every $x \in L$ there exists a (witness) string y such that*
$$\Pr[\langle P(y), V\rangle(x) = 1] = 1$$
- Soundness: *For every $x \notin L$, every interactive machine B and every $z \in \{0,1\}^*$*
$$\Pr[\langle B(z), V\rangle(x) = 1] \leq s(|x|)$$

In case that the soundness condition is required to hold only with respect to all 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 compute a valid witness to the statement proved. We proceed to a formal definition, following Bellare and Goldreich [3].

Definition 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 $q(|x|, \frac{1}{\delta})$, where $\delta = \Pr[\langle P'_{x,y,r}, V\rangle(x)] - \kappa(|x|)$ 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) We point out that our definition is slightly weaker than that of [3], in that we only require that the extractors running-time is bounded by $q(|x|, \frac{1}{\delta})$, whereas [3] requires it to be bounded by $q(|x|)\frac{1}{\delta}$. (In other words, [3] require an extraction with a "smaller" overhead). We mention that our results do not seem applicable to the more stringent definition of [3]. In any case, our definition implies previous definitions of proofs of knowledge (e.g., [9, 23]) and is sufficient for traditional applications of proofs of knowledge.

(2) 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).

(3) 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 execution 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 $M_j = M_{j,1} \ldots M_{j,k}$. The reply $M_{j+1} = V_k(x, M_1, \ldots, M_j; R)$ of V^k to some message M_j is defined via $M_{j+1,i} = V(x, M_{1,i} \ldots M_{j,i}; R_j)$ for $j = 1, \ldots, m$. 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 1. *Let (P,V) be a $2m$-round public-coin interactive argument for L. Then for every k, there exists a probabilistic oracle machine S, such that for every $x \notin L$, $\epsilon' > 0$, integer c, every interactive machine B, such that*

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

it holds that

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

Furthermore, the running-time of S is poly$\Big(\frac{1}{\epsilon'^m}, |x|, n^c, n^m\Big)$.

The proof of Theorem 1 is found in section 4. We point out that since we are only interested in *public-coin* argument systems, the verifiers next message is always independent of the history; in fact it will always be a truly random string. The proof of Theorem 1 crucially relies on this fact.

By applying Theorem 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.

THEOREM 2. *Let (P,V) be a constant-round public-coin interactive argument for the language L with soundness error $\epsilon(\cdot)$. Then (P_k, V_k) has soundness error $\epsilon(\cdot)^k + \nu(\cdot)$, where ν is a negligible function.*

THEOREM 3. *Let (P,V) be a constant-round public-coin proof of knowledge for the witness relation R with knowledge error $\epsilon(\cdot)$. Then (P_k, V_k) has knowledge error $\epsilon(\cdot)^k + \frac{1}{n^d}$ for any d.*

The proofs of Theorem 2 and Theorem 3 have been omitted and can be found in the full version. We mention, however, that the constant-round restriction is due to the fact that the running-time of the reduction, guaranteed by Theorem 1, is exponential in the number of rounds in the protocol.

4. PROOF OVERVIEW

We assume, without loss of generality, that in (P,V), P sends both the first and the last message. Furthermore, for simplicity, we assume that in each round the verifier sends messages of length n. We show how to construct an oracle machine S that satisfies the conditions of Theorem 1. As already mentioned, the construction of S is very generic (and relatively straight-forward); the main challenge, however, is to bound its success probability.

4.1 Description of S

On a high-level, S^B on input a statement x and given oracle access to a prover B, proceeds as follows. In a first stage, S^B will decide on a good "coordinate" $i' \in [k]$. In the second stage, S makes use of B in order to convince an outside verifier: S will interact with B while emulating the honest verifier strategy V for B in all executions except execution i (recall that B is supposed to convince k parallel executions of V); on the other hand, messages in execution i will be forwarded externally.

Now we proceed to describe S. It proceeds in the following two phases:

Pre-processing Phase: For each $i \in [k]$, S estimates the probability that its execution phase (described below) succeeds in convincing V, when coordinate i is selected. This estimation is performed by sampling an appropriate number of executions and computing the average success probability. Finally, the coordinate i' with the highest estimate is selected.

Execution Phase: Given a coordinate i, S will perform an "almost" straight-line invocation of B, externally forwarding all messages that are part of B's i'th executions (i.e., all incoming messages 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 the messages that are part of the other executions of B. The invocation of B will, however, not be entirely straight-line: in order to select the messages that are internally generated, S will run "look-aheads" of B in order to evaluate whether these generated messages are "good". More precisely (but still informally), given a history of messages $\overline{r}^q$ sent to B, S proceeds as follows:

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

We proceed to a formal description of S. Towards this goal, we first introduce some notation.

Histories and Extensions 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 history: *verifier*-step histories are of the form $h = \overline{r}^q$, where $\overline{r}^q = (r_1^1, .., r_k^1), .., (r_1^q, .., r_k^q)$ is a sequence of k-tuples of n-bit strings, whereas *prover*-step histories are of the form $h = \overline{r}^q; r$. We say that a history $\overline{r}^q$ is of length q if it consists of a sequence of q k-tuples. We also let $\overline{r}^q; (s_1, \ldots, s_k)$ denote the $q+1$-length history obtained by concatenating the q-length history $\overline{r}^q$ and the k-tuple $(s_1, \ldots, s_k)$,

Let $Ext_i(\overline{r}^q; r)$ denote the set of all length $q+1$ histories of the form $\overline{r}^q; (s_1, \ldots, s_k)$ where $s_i = r$ and $s_1, \ldots, s_k \in \{0,1\}^n$. Any extension in $Ext_i(\overline{r}^q; r)$ will be called an *i-extension* of the history $\overline{r}^q; r$. We will refer to as it as just an *extension* whenever i is obvious from context.

Formal Description of S A formal description of S can be found in Figure 1. For convenience of analysis, we let the algorithm S_i (described in Figure 2) denote the algorithm that performs the execution phase described above, given that coordinate i has been selected. S is then simply a machine that performs a preprocessing stage to find an "good" i and then calls S_i.

Algorithm 1 : S^B

1: **for** $i = 1$ to M **do**
2: Let $N = n^3$. Run $S_i^B(\cdot, true)$, N times. Let `count` denote the number of times S^B outputs 1.
3: Compute an estimate, X_j, of the success probability of S_i^B given the samples, i.e., $X_j \leftarrow \frac{\texttt{count}}{N}$.
4: Let i' be the index of the coordinate with the largest estimate, i.e. $i' \leftarrow \text{argmax}_i\{X_i\}$.
5: Run $S_i^B(\cdot, false)$.

Figure 1: Description of the machine S^B

Algorithm 2 : $S_i^B(\overline{r}^q, \texttt{internal})$

1: Let $(c_1, \ldots, c_k)$ be the output of B when fed the messages in $\overline{r}^q$, i.e. $(c_1, \ldots, c_k) \leftarrow B(\overline{r}^q)$.
2: **if** $\overline{r}^q$ is a complete history **then**
3: **if** $\texttt{internal} = false$ **then**
4: Externally forward c_i and halt.
5: **else**
6: Output whether B succeeds in convincing V in coordinate i, when fed the messages in $\overline{r}^q$ and halt.
7: **else**
8: **if** $\texttt{internal} = false$ **then**
9: Externally forward c_i and let r be the message externally received.
10: **else**
11: Pick a random message r, i.e. $r \leftarrow \{0,1\}^n$.
12: Let $M = \frac{2n^{c+1}}{\epsilon'}$ and $M_q = \lceil 4^{2q} n^{2c} \log(4^q n^c M) \rceil$.
13: Pick M extensions $\overline{s_1}^{q+1}, \ldots, \overline{s_M}^{q+1}$ of $\overline{r}^q; r$.
14: **for** $j = 1$ to M **do**
15: Run $S_i^B(\overline{s_j}^{q+1}, true)$, M_{q+1} times. Let `count` denote the number of times S_i^B outputs 1.
16: Compute an estimate X_j, of the success probability of S_i^B on $\overline{s_j}^{q+1}$. More precisely, let $X_j \leftarrow \frac{\texttt{count}}{M_{q+1}}$.
17: Let j' be the index of the sample with the largest estimate, i.e. $j' \leftarrow \text{argmax}_j\{X_j\}$.
18: Run $S_i^B(\overline{s_{j'}}^{q+1}, \texttt{internal})$.

Figure 2: Description of the machine S_i^B

4.2 Analyzing the success probability of S

We proceed to analyzing the success probability of S on input a machine B. Our analysis proceeds in several steps.

Step 1: Defining $\widehat{S}_i$ In the first step we define a particular *unbounded* algorithm $\widehat{S}_i$. Whereas S_i "samples" to find "good" messages to send to B, $\widehat{S}_i$ always picks the best messages to send to B (this is possible since $\widehat{S}_i^B$ is computationally unbounded). A formal description of $\widehat{S}$ can be found in Figure 4. In contrast to S_i, it will however be simple to analyze the success probability of $\widehat{S}_i$. In particular, we show:

Algorithm 3 : $\widetilde{S}_i^B(\overline{r}^q)$

1: Let $(c_1, \ldots, c_k)$ be the output of B when fed the messages in $\overline{r}^q$, i.e. $(c_1, \ldots, c_k) \leftarrow B(\overline{r}^q)$.
2: **if** $\overline{r}^q$ is a complete history **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 M extensions $\overline{s_1}^{q+1}, \ldots, \overline{s_M}^{q+1}$ of $\overline{r}^q; r$.
7: Let j' be the sample on which $\widetilde{S}_i^B$ on input $\overline{s_{j'}}^{q+1}$ wins with highest probability.
8: Run $\widetilde{S}_i^B(\overline{s_{j'}}^{q+1})$.

Figure 3: Description of the machine $\widetilde{S}_i^B$

CLAIM 1. *For any interactive machine B, it holds that*

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

Intuitively, the claim holds by the following argument: Consider a prover Q for the protocol (P_k, V_k) that forwards messages in its execution j (for $j \in [k]$) to and from $\widehat{S}_j^B$. It can be seen that such a Q does better than B (since $\widehat{S}_j^B$ maximizes its success probability), and furthermore Q acts independently in each of the executions; thus the claim follows.

By Claim 1 we conclude that there exists some i, such that $\widehat{S}_i^B$ wins with high probability (in fact, with probability at least $\Pr[\langle B, V_k\rangle(x) = 1]^{\frac{1}{k}}$).

Furthermore, as we will show in the following steps, the success probability of S_i^B will not be too far from the success probability of $\widehat{S}_i^B$.

Step 2: Defining $\widetilde{S}$ Towards the goal of showing that the success probability of S_i^B is close to the success probability of $\widehat{S}_i^B$, we define an intermediary computationally unbounded machine $\widetilde{S}_i$. $\widetilde{S}_i$ will proceed 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 3. We show that the success probability of S_i^B is close to the success probability of $\widetilde{S}_i^B$. Intuitively (on a very high-level) this follows since by a Chernoff bound, the estimates computed by S_i^B will be "close enough". More formally, we show:

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

$$\Pr\left[\langle S_i^{B(x,y)}(x, \epsilon', c), V\rangle(x) = 1\right] \geq \Pr\left[\langle \widetilde{S}_i^{B(x,y)}(x, \epsilon', c), V\rangle(x) = 1\right] - \frac{1}{n^c}$$

We proceed to show that the success probability of $\widetilde{S}_i^B$ is not too far from the success probability of $\widehat{S}_i^B$.

Algorithm 4 : $\widehat{S}_i^B(\overline{r}^q)$

1: Let $(c_1, \ldots, c_k)$ be the output of B when fed the messages in $\overline{r}^q$, i.e. $(c_1, \ldots, c_k) \leftarrow B(\overline{r}^q)$.
2: **if** $\overline{r}^q$ is a complete history **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 the extension $\overline{s}^{q+1}$ of $\overline{r}^q; r$ on which $\widehat{S}_i^B$ on input $\overline{s}^{q+1}$ wins with highest probability.
7: Run $\widehat{S}_i^B(\overline{s}^{q+1})$.

Figure 4: Description of the machine $\widehat{S}_i$

Step 3: Defining $\widetilde{B}$ Note that 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). To make the game more fair, we instead compare the success probability of $\widetilde{S}_i^B$, with the success probability of $\widehat{S}_i$ when executed on a "hybrid" cheating prover $\widetilde{B}$. $\widetilde{B}$ will proceed exactly as B, except that it will *abort* 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 $\widehat{S}_i^{\widetilde{B}}$.

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 $h = \overline{r}^q$, we denote by $H_i(\overline{r}^p; r)$ a subset of the valid i-extensions of $\overline{r}^q; r$ that contains the $\frac{\epsilon'}{n^c}$ fraction of extensions, on which $\widetilde{S}_i$ has the highest probability of winning, where ϵ' is the success probability of B. The extensions in $H_i(\overline{r}^p; r)$ are called *strong* (as these are the "best" extensions for $\widetilde{S}_i$) and the remaining ones *weak*. We define $H(\overline{r}^{q-1}) = \bigcup_{i \in [k], r \in \{0,1\}^n} H_i(\overline{r}^{q-1}; r)$.

We next proceed to defining $\widetilde{B}$. At a high level, all that $\widetilde{B}$ does, is to abort whenever it receives a *strong* view. Formally, at any stage if $\overline{r}^q \in H(\overline{r}^{q-1})$, then $\widetilde{B}$ returns $\perp$ and terminates the protocol (we here assume without loss of generality that the honest verifier V always rejects whenever it receives the $\perp$ symbol).

We have the following claims:

CLAIM 3. *Let B be an interactive machine such that*

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

Then

$$\Pr\Big[\Big\langle \widetilde{B}(y), V_k\Big\rangle(x) = 1\Big] \geq \epsilon'\Big(1 - \frac{km}{n^c}\Big)$$

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

$$\Pr\Big[\langle \widetilde{S}_i^{B(x,y)}(x, \epsilon', c), V\rangle(x) = 1\Big] \geq \Pr\Big[\langle \widehat{S}_i^{\widetilde{B}(x,y)}(x, \epsilon', c), V\rangle(x) = 1\Big] - e^{-n}$$

The first of these claims follows from a simple counting argument. The proof of the second claim is more complex; however, on a high-level it will follow from the fact that $\widetilde{S}^B$ with high probability will "hit" a *strong* extension, and thus will perform better than $\widehat{S}^{\widetilde{B}}$.

Step 4: Putting it all together In the above steps we have thus provided a lower bound on the success probability of S_i^B for the best coordinate i. It only remains to lower bound the success probability of S^B. Recall that S estimates the success probability of S_i for each i, and then executes the S_i which obtained the highest estimate. We finally show (again applying the Chernoff bound) that the success probability of S, will not be far from the success probability of S_i for the actual best i.

5. PROOF OF MAIN THEOREM

In this section, we provide the details of the proof of the main theorem. We start by proving the claims stated in Section 4. Next, we conclude the proof given these claims.

In the remainder of the proof we focus on inputs x, ϵ', c for S. To simplify notation, we omit these inputs to S (e.g., we assume that they are hardcoded in S). We also assume that the argument system (P, V) has $2m$ rounds, for some fixed constant m. We start by providing some additional notation that will by used throughout the proof.

Additional Notation Let $\mathsf{win}_i[M^B, h]$ denote the probability that machine M with oracle access to B wins in its external execution with V, if 1) starting from history h, and 2) using coordinate i for forwarding messages to the external V. It follows immediately from the definition that

$$\mathsf{win}_i[M^B, \overline{r}^q] = E\Big[r \leftarrow \{0,1\}^n : \mathsf{win}_i[M^B, \overline{r}^q; r]\Big]$$

since the verifier picks a random string r independent of the history $\overline{r}^q$, and the probability of M^B winning from history $\overline{r}^q; r$ is $\mathsf{win}_i[M^B, \overline{r}^q; r]$. Thus,

$$\mathsf{win}_i[M^B, \overline{r}^q] = \frac{1}{2^n} \sum_{r \in \{0,1\}^n} \mathsf{win}_i[M^B, \overline{r}^q; r]$$

We will also let $\mathsf{win}[B, \overline{r}^p]$ denote the probability that prover B wins in (B, V^k) on a given history $\overline{r}^p$. Before proving the theorem, we show the following a simple observation which will be useful in the proof of Claim 2 and the conclusion of the theorem (in Section 5.2).

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,

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

■

5.1 Proof of Claims

We first restate the claims with the notation described in the beginning of this section, and next proceed to proving them.

$$\begin{aligned}
\prod_{i=1}^{k} \mathsf{win}_i[\widehat{S}_i^B, \cdot] &\geq \mathsf{win}[B, \cdot] && \text{(Claim 1)} \\
\mathsf{win}_i[S_i^B, \cdot] &\geq \mathsf{win}_i[\widetilde{S}_i^B, \cdot] - \frac{1}{n^c} && \text{(Claim 2)} \\
\mathsf{win}_i[\widetilde{S}_i^B, \cdot] &\geq \mathsf{win}_i[\widehat{S}_i^{\widetilde{B}}, \cdot] - e^{-n} && \text{(Claim 4)}
\end{aligned}$$

5.1.1 Proof of Claim 1

We start by giving some intuition to this claim. 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 $\prod_{i=1}^k \mathsf{win}_i[\widehat{S}_i^B, \cdot]$. 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 1. Given any history $\overline{r}^q$, we show that $\prod_{i=1}^k \mathsf{win}_i[\widehat{S}_i^B, \overline{r}^q] \geq \mathsf{win}[B, \overline{r}^q]$.

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

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 $\overline{r}^{q-1}$ be a $q-1$ length history. It holds that

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

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

$$\begin{aligned}
&\prod_{i=1}^{k} \left(\frac{1}{2^n} \sum_{s_i \in \{0,1\}^n} \mathsf{win}_i[\widehat{S}_i^B, \overline{r}^{q-1}; s_i] \right) \\
&\quad = \frac{1}{2^{nk}} \sum_{s_1,\ldots,s_k \in \{0,1\}^n} \left(\prod_{i=1}^{k} \mathsf{win}_i[\widehat{S}_i^B, \overline{r}^{q-1}; s_i] \right)
\end{aligned}$$

Since $\widehat{S}_i^{\widetilde{B}}$ always finds the *best* extension to any history $\overline{r}^{q-1}; s_i$, it holds that for all $s_1, \ldots, s_k \in \{0,1\}^n$,

$$\mathsf{win}_i[\widehat{S}_i^B, \overline{r}^{q-1}; s_i] \geq \mathsf{win}_i[\widehat{S}_i^B, \overline{r}^{q-1}; (s_1, \ldots, s_k)]$$

Therefore,

$$\begin{aligned}
&\frac{1}{2^{nk}} \sum_{s_1,\ldots,s_k \in \{0,1\}^n} \left(\prod_{i=1}^{k} \mathsf{win}_i[\widehat{S}_i^B, \overline{r}^{q-1}; s_i] \right) \\
&\quad \geq \frac{1}{2^{nk}} \sum_{s_1,\ldots,s_k \in \{0,1\}^n} \left(\prod_{i=1}^{k} \mathsf{win}_i[\widehat{S}_i^B, \overline{r}^{q-1}; (s_1, \ldots, s_k)] \right)
\end{aligned}$$

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

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

it follows that

$$\begin{aligned}
&\prod_{i=1}^{k} \mathsf{win}_i[\widehat{S}_i^B, \overline{r}^{q-1}] \\
&\qquad \geq \frac{1}{2^{nk}} \sum_{s_1,\ldots,s_k \in \{0,1\}^n} \mathsf{win}[B, \overline{r}^{q-1}; (s_1, \ldots, s_k)] \\
&\qquad = \mathsf{win}[B, \overline{r}^{q-1}]
\end{aligned}$$

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

5.1.2 Proof of Claim 2

We will show by induction that for all q, $\overline{r}^q$ and $\overline{r}^{q-1}; r$,

$$\begin{aligned}
\mathsf{win}_i[S_i^B, \overline{r}^q] &\geq \mathsf{win}_i[\widetilde{S}_i^B, \overline{r}^q] - \frac{1}{4^q n^c} \\
\mathsf{win}_i[S_i^B, \overline{r}^{q-1}; r] &\geq \mathsf{win}_i[\widetilde{S}_i^B, \overline{r}^{q-1}; r] - \frac{1}{4^{q-1} n^c}
\end{aligned}$$

Base case: $q = m$ Given a complete view $\overline{r}^m$, both S_i^B and $\widetilde{S}_i^B$ win exactly when B wins on $\overline{r}^m$ 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 $\overline{r}^q$, or of the form $\overline{r}^{q-1}; r$. First, consider the case when $h = \overline{r}^q$ is the history at a *verifier*-step. By the induction hypothesis, for every r we have that

$$\mathsf{win}_i[S_i^B, \overline{r}^q; r] \geq \mathsf{win}_i[\widetilde{S}_i^B, \overline{r}^q; r] - \frac{1}{4^q n^c}$$

From the definition we know that,

$$\mathsf{win}_i[S_i^B, \overline{r}^q] = \frac{1}{2^n} \sum_{r \in \{0,1\}^n} \mathsf{win}_i[S_i^B, \overline{r}^q; r]$$

Thus,

$$\mathsf{win}_i[S_i^B, \overline{r}^q] \geq \mathsf{win}_i[\widetilde{S}_i^B, \overline{r}^q] - \frac{1}{4^q n^c}$$

Next, consider the case when $h = \overline{r}^{q-1}; r$ is the history at a *prover*-step. We, here, need to compare the probabilities $\mathsf{win}_i[S_i^B, \overline{r}^{q-1}; r]$ and $\mathsf{win}_i[\widetilde{S}_i^B, \overline{r}^{q-1}; r]$. Recall that both S_i^B and $\widetilde{S}_i^B$ pick M samples that are extensions of $\overline{r}^{q-1}; r$. Given these M 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 $\overline{s}^q$ of $\overline{r}^{q-1};r$, let the random variable $\mathsf{est}[S_i^B](\overline{s}^q)$ denote the estimate computed by S_i^B on $\overline{s}^q$. We have,

$$\mathsf{win}_i[S_i^B,\overline{r}^{q-1};r] = \frac{1}{2^{(k-1)nm}}\times \sum_{\substack{\overline{s_z}^q\in Ext_i(\overline{r}^{q-1};r)\\ 1\le z\le M}} E\Big[j'\leftarrow \operatorname*{argmax}_j\{\mathsf{est}[S_i^B](\overline{s_j}^q)\}:\mathsf{win}_i[S_i^B,\overline{s_{j'}}^q]\Big]$$

$$\mathsf{win}_i[\widetilde{S}_i^B,\overline{r}^{q-1};r] = \frac{1}{2^{(k-1)nm}}\times \sum_{\substack{\overline{s_z}^q\in Ext_i(\overline{r}^{q-1};r)\\ 1\le z\le M}} \max_j\{\mathsf{win}_i[\widetilde{S}_i^B,\overline{s_j}^q]\}$$

To show the induction step, it thus suffices to show that given any set of samples $(\overline{s_1}^q,\ldots,\overline{s_M}^q)$,

$$E\Big[j'\leftarrow \operatorname*{argmax}_j\{\mathsf{est}[S_i^B](\overline{s_j}^q)\}:\mathsf{win}_i[S_i^B,\overline{s_{j'}}^q]\Big] \ge \max_j\{\mathsf{win}_i[\widetilde{S}_i^B,\overline{s_j}^q]\} - \frac{1}{4^{q-1}n^c} \quad (1)$$

By the induction hypothesis it follows that for every j,

$$\mathsf{win}_i[S_i^B,\overline{s_j}^q] \ge \mathsf{win}_i[\widetilde{S}_i^B,\overline{s_j}^q] - \frac{1}{4^q n^c}$$

Therefore,

$$\max_j\{\mathsf{win}_i[S_i^B,\overline{s_j}^q]\} \ge \max_j\{\mathsf{win}_i[\widetilde{S}_i^B,\overline{s_j}^q]\} - \frac{1}{4^q n^c} \quad (2)$$

Below we will show that,

$$E\Big[j'\leftarrow \operatorname*{argmax}_j\{\mathsf{est}[S_i^B](\overline{s_j}^q)\}:\mathsf{win}_i[S_i^B,\overline{s_{j'}}^q]\Big] \ge \max_j\{\mathsf{win}_i[S_i^B,\overline{s_j}^q]\} - \frac{3}{4^q n^c} \quad (3)$$

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 $\overline{s_j}^q$, S_i^B picks M_q samples to compute $\mathsf{est}[S_i^B](\overline{s_j}^q)$. By construction,

$$E[\mathsf{est}[S_i^B](\overline{s_j}^q)] = \mathsf{win}_i[S_i^B,\overline{s_j}^q]$$

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

$$\Big|X - \mathsf{win}_i[S_i^B,\overline{s_j}^q]\Big| \le \frac{1}{4^q n^c}$$

Otherwise we call the estimate X "bad" for $\overline{s_j}^q$. We now analyze the expectation when all the estimates are "good". It follows by Observation 1 that

$$\begin{aligned} E\quad & \Big[j'\leftarrow \operatorname*{argmax}_j\{\mathsf{est}[S_i^B](\overline{s_j}^q)\}:\mathsf{win}_i[S_i^B,\overline{s_{j'}}^q]\Big| \\ & \qquad \forall j\ \ \mathsf{est}[S_i^B](\overline{s_j}^q) \text{ is "good" for } \overline{s_j}^q\Big] \\ \ge\quad & E\Big[\max_j\Big\{\mathsf{win}_i[S_i^B,\overline{s_j}^q] - \frac{2}{4^q n^c}\Big\}\Big| \\ & \qquad \forall j\ \ \mathsf{est}[S_i^B](\overline{s_j}^q) \text{ is "good" for } \overline{s_j}^q\Big] \\ =\quad & \max_j\{\mathsf{win}_i[S_i^B,\overline{s_j}^q]\} - \frac{2}{4^q n^c} \end{aligned} \quad (4)$$

We show that except with "small" probability, for any j, the estimate $\mathsf{est}[S_i^B](\overline{s_j}^q)$ is "good" for $\overline{s_j}^q$. In fact, it directly follows by the Chernoff bound that

$$Pr[\mathsf{est}[S_i^B](\overline{s_j}^q) \text{ is "bad" for } \overline{s_j}^q] \le 2^{-\frac{M_q}{n^{2c}4^{2q}}}.$$

Using the union bound,

$$\begin{aligned} & Pr[\forall j\ \ \mathsf{est}[S_i^B](\overline{s_j}^q) \text{ is "good" for } \overline{s_j}^q] \\ & \qquad \ge 1 - M\cdot 2^{-\frac{M_q}{n^{2c}4^{2q}}} \ge 1 - \frac{1}{4^q n^c} \end{aligned} \quad (5)$$

where the last inequality follows since $M_q = \lceil 4^{2q}n^{2c}\log(4^q n^c M)\rceil$. Therefore, by expanding the left hand side of equation (3) we get

$$\begin{aligned} E\quad & \Big[j'\leftarrow \operatorname*{argmax}_j\{\mathsf{est}[S_i^B](\overline{s_j}^q)\}:\mathsf{win}_i[S_i^B,\overline{s_{j'}}^q]\Big] \\ \ge\quad & E\Big[j'\leftarrow \operatorname*{argmax}_j\{\mathsf{est}[S_i^B](\overline{s_j}^q)\}:\mathsf{win}_i[S_i^B,\overline{s_{j'}}^q]\Big| \\ & \qquad \forall j\ \ \mathsf{est}[S_i^B](\overline{s_j}^q) \text{ is "good" for } \overline{s_j}^q\Big] \\ & \times \Pr[\forall j\ \ \mathsf{est}[S_i^B](\overline{s_j}^q) \text{ is "good" for } \overline{s_j}^q] \\ \ge\quad & \Big[\max_j\{\mathsf{est}[S_i^B](\overline{s_j}^q)\} - \frac{2}{4^q n^c}\Big]\times\Big(1-\frac{1}{4^q n^c}\Big) \\ & \qquad \text{using (4) and (5)} \\ \ge\quad & \max_j\{\mathsf{est}[S_i^B](\overline{s_j}^q)\} - \frac{3}{4^q n^c} \end{aligned}$$

This concludes the proof of equation (3) and the induction step. Returning to the original claim, we have that

$$\begin{aligned} \Pr\quad & \Big[\langle S_i^{B(x,y)}(x,\epsilon',c),V\rangle(x)=1\Big] = \mathsf{win}_i[S_i^B,\cdot] \\ \ge\quad & \mathsf{win}_i[\widetilde{S}_i^B,\cdot] - \frac{1}{4^0 n^c} \\ =\quad & \Pr\Big[\langle \widetilde{S}_i^{B(x,y)}(x,\epsilon',c),V\rangle(x)=1\Big] - \frac{1}{n^c} \end{aligned}$$

■

5.1.3 Proof of Claim 3

Recall that prover $\widetilde{B}$ acts exactly as prover B on all histories that are *weak*. On the other hand, on *strong* 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 *strong*.

First, we determine the fraction of *strong* histories (or correspondingly *weak* histories) for a coordinate i. For any *weak* history, the fraction of extensions that are *weak* is $1-\frac{\epsilon'}{n^c} = 1-\alpha$ (where $\alpha = \frac{\epsilon'}{n^c}$). The overall fraction of histories that are *weak* is thus, $(1-\alpha)^m$. Therefore, the fraction of histories that are *strong* for coordinate i is $1-(1-\alpha)^m \le m\alpha$.

Using union bound, the total fraction that are *strong* for all coordinates is at most $km\alpha$. Thus,

$$\begin{aligned} \Pr\Big[\Big\langle \widetilde{B}(y),V_k\Big\rangle(x)=1\Big] \quad &\ge\quad \Pr\Big[\langle B(y),V_k\rangle(x)=1\Big] - km\alpha \\ &=\quad \epsilon'\Big(1-\frac{km}{n^c}\Big) \end{aligned}$$

■

5.1.4 Proof of Claim 4

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, $\overline{r}^q$ and $\overline{r}^{q-1}; r$,

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

Base case: $q = m$ At the final *verifier*-step, either the verifier accepts or rejects. Given a complete history $\overline{r}^m$, we know that $\widetilde{S}_i^B$ wins whenever B wins in coordinate i on $\overline{r}^m$ and $\widehat{S}_i^{\widetilde{B}}$ wins whenever $\widetilde{B}$ wins in coordinate i on $\overline{r}^m$. 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 if of the form $\overline{r}^q$, or of the form $\overline{r}^{q-1}; r$. We first consider the case when $h = \overline{r}^q$ 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 $h = \overline{r}^{q-1}; r$, is history at a *prover*-step. Towards the goal of proving the induction step, we show that for any *weak* extension $\overline{s}^q$ of $\overline{r}^{q-1}; r$, the following holds.

$$\mathsf{win}_i[\widetilde{S}_i^B, \overline{r}^{q-1}; r] \geq \mathsf{win}_i[\widetilde{S}_i^B, \overline{s}^q] - (1-\alpha)^M$$

Intuitively, this follows since except with small probability $\widetilde{S}_i^B$ on input $\overline{r}^{q-1}; r$ finds a *strong* extension. We proceed to a formal proof.

Recall that in a *prover*-step, $\widetilde{S}_i^B$ samples M extensions of $(\overline{r}^{q-1}; r)$. Among the M samples, $\widetilde{S}_i^B$ picks the extension that maximizes its probability of winning. Since, $\overline{s}^q$ is a *weak* extension, $\widetilde{S}_i^B$ on $\overline{r}^{q-1}; r$ will succeed with probability at least $\mathsf{win}_i[\widetilde{S}_i^B, \overline{s}^q]$, if any of the M samples of $\widetilde{S}_i^B$ hits a *strong* extension (since by definition of *strong* extensions, the success probability of $\widetilde{S}_i^B$ on *strong* extensions is at least as high as on any *weak* extension). We call the set of M samples "good" if at least one of the samples hits a *strong* extension, and "bad" otherwise.

We proceed to show that the probability of a random sample-set being "bad" is small. Recall that, the fraction of extensions that are *strong* is α. Hence, the probability that none of the samples hit a *strong* extension is $\leq (1-\alpha)^M$. By the union bound,

$$\begin{aligned}&\Pr \quad [\widetilde{S}_i^B \text{ loses on } \overline{r}^{q-1}; r]\\ \leq\ &\Pr[\widetilde{S}_i^B \text{ loses } | \widetilde{S}_i^B \text{ picks a "good" sample-set}]\\ &+\Pr[\widetilde{S}_i^B \text{ picks a "bad" sample-set}]\\ \leq\ &1 - \mathsf{win}_i[\widetilde{S}_i^B, \overline{s}^q] + (1-\alpha)^M\end{aligned}$$

Thus,

$$\begin{aligned}\mathsf{win}_i[\widetilde{S}_i^B, \overline{r}^{q-1}; r] &= 1 - \Pr[\widetilde{S}_i^B \text{ loses on } \overline{r}^{q-1}; r]\\ &\geq \mathsf{win}_i[\widetilde{S}_i^B, \overline{s}^q] - (1-\alpha)^M\\ &\geq \mathsf{win}_i[\widehat{S}_i^{\widetilde{B}}, \overline{s}^q] - (m-q)(1-\alpha)^M - (1-\alpha)^M\\ &= \mathsf{win}_i[\widehat{S}_i^{\widetilde{B}}, \overline{s}^q] - (m-q+1)(1-\alpha)^M\end{aligned}$$

where the second inequality follows by the induction hypothesis. Since, this is true for any *weak* extension $\overline{s}^q$ of $\overline{r}^{q-1}; r$, and in particular it holds for the *weak* extension that maximizes the probability of $\widehat{S}_i^{\widetilde{B}}$ winning, we have that

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

This concludes the induction step. Recall that, $M = \frac{2n^{c+1}}{\epsilon'} = 2n\frac{1}{\alpha} > (n + \log m)\frac{1}{\alpha}$. Therefore, we have

$$\begin{aligned}&\Pr[\langle \widetilde{S}_i^{B(x,y)}(x, \epsilon', c), V\rangle(x) = 1]\\ &\quad \geq \Pr[\langle \widehat{S}_i^{\widetilde{B}^*(x,y)}(x, \epsilon', c), V\rangle(x) = 1] - m(1-\alpha)^M\\ &\quad \geq [\langle \widehat{S}_i^{\widetilde{B}^*(x,y)}(x, \epsilon', c), V\rangle(x) = 1] - e^{-n}\end{aligned}$$

■

5.2 Proof of Theorem 1

Using Claims 1 and 3 we get that

$$\prod_{i=1}^{k} \Pr\Big[\langle \widehat{S}_i^{\widetilde{B}(x,y)}(x, \epsilon', c), V\rangle(x) = 1\Big] \geq \epsilon'\Big(1 - \frac{km}{n^c}\Big)$$

Since each of the factors in the product is positive (as they are all probabilities), it holds that at least one of factors must be $\geq \Big[\epsilon'\Big(1 - \frac{km}{n^c}\Big)\Big]^{\frac{1}{k}}$. Thus, there exists an i^* such that

$$\Pr\Big[\langle \widehat{S}_{i^*}^{\widetilde{B}(x,y)}(x, \epsilon', c), V\rangle(x) = 1\Big] \geq \Big[\epsilon'\Big(1 - \frac{km}{n^c}\Big)\Big]^{\frac{1}{k}}$$

It follows from Claims 2 and 4 that

$$\Pr\Big[\langle S_{i^*}^{B(x,y)}(x, \epsilon', c), V\rangle(x) = 1\Big] \geq \Big[\epsilon'\Big(1 - \frac{km}{n^c}\Big)\Big]^{\frac{1}{k}} - e^{-n} - \frac{1}{n^c}$$

i.e.,

$$\mathsf{win}_{i^*}[S_{i^*}^B, \cdot] \geq \Big[\epsilon'\Big(1 - \frac{km}{n^c}\Big)\Big]^{\frac{1}{k}} - \frac{1}{n^c} - e^{-n} \tag{6}$$

Recall that in the preprocessing phase, S^B estimates the success probability of S_i^B for all i, and chooses the one with the highest estimate. The estimate for i is denoted by the random variable $\mathsf{est}[S_i^B]$. An estimate X_i is called "good", if

$$\Big|X_i - \mathsf{win}_i[S_i^B, \cdot]\Big| \leq \frac{1}{n^c}$$

and "bad" otherwise. When all the estimates are "good", it follows by Observation 1 that

$$\begin{aligned}E \quad &\Big[i' \leftarrow \operatorname*{argmax}_i X_i : \mathsf{win}_{i'}[S_{i'}^B, \cdot] \Big| \forall i \quad X_i \text{ is "good"}\Big]\\ \geq \quad &E\Big[\max_i \Big\{\mathsf{win}_{i'}[S_{i'}^B, \cdot] - \frac{2}{n^c}\Big\} \Big| \forall i \quad X_i \text{ is "good"}\Big]\\ = \quad &\max_i\{\mathsf{win}_{i'}[S_{i'}^B, \cdot]\} - \frac{2}{n^c}\end{aligned} \tag{7}$$

We now show that except with "small" probability for all i estimate X_i is "good". It follows from the Chernoff bound that,

$$\Pr[X_i \text{ is "bad"}] \leq 2^{-\frac{N}{n^2}}$$

By applying the union bound and recalling that $N = n^3 \geq n^2 \log(kn^c)$, we get

$$\Pr[\forall i \quad X_i \text{ is "good"}] \geq 1 - k \cdot 2^{-\frac{N}{n^2}} \geq 1 - \frac{1}{n^c} \tag{8}$$

We conclude that

$$\begin{aligned}
&\Pr \quad \Big[\langle S^{B(x,y)}(x,\epsilon',c),V\rangle(x)=1\Big] \\
&= \quad E\Big[i' \leftarrow \operatorname*{argmax}_i\{\mathsf{est}[S_i^B]\} : \mathsf{win}_{i'}[S_{i'}^B,\cdot]\Big] \\
&\geq \quad E\Big[i' \leftarrow \operatorname*{argmax}_i\{\mathsf{est}[S_i^B]\} : \mathsf{win}_{i'}[S_{i'}^B,\cdot] \\
&\qquad\qquad \Big|\forall i \;\; \mathsf{est}[S_i^B] \text{ is "good"}\Big] \\
&\qquad \times \Pr[\forall i \;\; \mathsf{est}[S_i^B] \text{ is "good"}] \\
&\geq \quad \Big[\max_i\{\mathsf{win}_{i'}[S_{i'}^B,\cdot]\} - \frac{2}{n^c}\Big]\Big(1-\frac{1}{n^c}\Big) \\
&\qquad\qquad\qquad\qquad \text{using (7) and (8)} \\
&\geq \quad \max_i\{\mathsf{win}_{i'}[S_{i'}^B,\cdot]\} - \frac{3}{n^c} \\
&\geq \quad \mathsf{win}_{i^*}[S_{i^*}^B,\cdot] - \frac{3}{n^c} \\
&\geq \quad \Big[\epsilon'\Big(1-\frac{km}{n^c}\Big)\Big]^{\frac{1}{k}} - e^{-n} - \frac{4}{n^c} \qquad \text{using (6)}
\end{aligned}$$

Running time analysis: We first determine the number of oracle queries made by S_i^B. Let $T(q)$ be the number of oracle queries made by S_i^B on any history $\overline{r}^q$ of length q. Recall that, S_i^B on $\overline{r}^q$, first makes an oracle query (see step (1) in Figure 2). Then, it samples M extensions of $\overline{r}^q$. For each sample $\overline{s_j}^{q+1}$ (all of which are of length $q+1$), it runs S_i^B recursively on $\overline{s_j}^{q+1}$, M_{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 $MM_{q+1}+1$. 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,

$$T(q) = 1 + (MM_{q+1}+1)T(q+1)$$

Since, $M > 2$ it follows that

$$T(q) \leq 2MM_{q+1}T(q+1) \leq (2M)^{m-q}\prod_{j=q+1}^{m} M_j$$

We now analyze the number of oracle queries made by $\widehat{S}^{\widetilde{B}}$. There are two phases: the preprocessing and the execution phase. In the preprocessing phase, S^B runs S_i^B (on the empty history) N times for every i. Hence the number of oracle queries in this phase is $kN\cdot T(0) \leq kN(2M)^m M_1\cdots M_m$. In the execution phase, S^B runs $S_i^{\widetilde{B}}$ (again on the empty history) for a particular i. Therefore, the number of oracle queries in this phase is $T(0) \leq (2M)^m M_1\cdots M_m$. We conclude that the total number of oracle queries made by S^B is upper bounded by

$$(1+kN)(2M)^m M_1\cdots M_m$$

Recall that $M_q = \lceil 4^{2q}n^{2c}\log(4^q n^c M)\rceil$, $N = n^3$ and $M = \frac{2n^{c+1}}{\epsilon'}$. Since, the oracle B runs in time polynomial in $|x|$ and n, we have that the total running time of S^B is polynomial in $|x|, \frac{1}{\epsilon'^m}, n^c$ and n^m. ∎

Acknowledgements

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

6. REFERENCES

[1] L. Babai and S. Moran. Arthur-Merlin games: A randomized proof system, and a hierarchy of complexity classes. *JCSS*, Vol. 36, pages 254–276, 1988.

[2] M. Bellare, R. Impagliazzo and M. Naor. Does Parallel Repetition Lower the Error in Computationally Sound Protocols? In *38th FOCS*, pages 374–383, 1997.

[3] M. Bellare and O. Goldreich. On Defining Proofs of Knowledge. In *C*RYPTO '92, pages 390-420,1993.

[4] M. Blum. How to prove a Theorem So No One Else Can Claim It. *Proc. of the International Congress of Mathematicians*, Berkeley, California, USA, pages 1444-1451, 1986.

[5] G. Brassard, D. Chaum and C. Crépeau. Minimum Disclosure Proofs of Knowledge. *JCSS*, Vol. 37, No. 2, pages 156–189, 1988. Preliminary version by Brassard and Crépeau in *27th FOCS*, 1986.

[6] R. Canetti and S. Halevi and M. Steiner. Hardness Amplification of Weakly Verifiable Puzzles. In *2nd TCC*, Springer LNCS 3876, pages 17-33, 2005.

[7] O. Goldreich and R. Impagliazzo and L. A. Levin and R. Venkatesan and D. Zuckerman. Security Preserving Amplification of Hardness. In *31th FOCS*, pages 318-326, 1990.

[8] M. Ben-or and S. Goldwasser and J. Kilian and A. Wigderson. Multi Prover Interactive Proofs: How to Remove Intractability. In *20th STOC*, pages 113-131, 1988.

[9] U. Feige and A. Fiat and A. Shamir. Zero-Knowledge Proofs of Identity. In *J. Cryptology*, 1(2), pages 77-94, 1988.

[10] U. Feige and J. Kilian. Two prover protocols: low error at affordable rates. In *26th STOC*, pages 172-183, 1994.

[11] L. Fortnow and J. Rompel and M. Sipser. On the power of multi-prover interactive protocols. In *Theor. Comput. Sci.*, 134(2), pages 545-557, 1994.

[12] U. Feige and A. Shamir. Zero Knowledge Proofs of Knowledge in Two Rounds. In *Crypto89*, Springer LNCS 435, pages. 526–544, 1989.

[13] O. Goldreich. *Foundations of Cryptography – Basic Tools.* Cambridge University Press, 2001.

[14] O. Goldreich and H. Krawczyk. On the composition of Zero-Knowledge Proof Systems. *SIAM Journal on Computing*, Vol. 25(1), pages 169–192, 1996.

[15] S. Goldwasser and S. Micali. Probabilistic Encryption. *JCSS*, Vol. 28, No 2, pages 270-299, 1984.

[16] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *STOC 85*, pages 291–304, 1985.

[17] S. Goldwasser, S. Micali and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Jour. on Computing*, Vol. 18(1), pp. 186–208, 1989.

[18] S. Goldwasser, S. Micali and R.L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen Message Attacks. *SIAM Jour. on Computing*, Vol. 17, No. 2, pp. 281–308, 1988.

[19] S. Goldwasser, S. Micali and A. Wigderson. Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems. *JACM*, Vol. 38(1), pp. 691-729, 1991.

[20] K. Pietrzak and D. Wikström. Parallel Repetition of Computationally Sound Protocols Revisited. To appear in TCC 2007.

[21] R. Richardson and J. Kilian. On the Concurrent Composition of Zero-Knowledge Proofs. *Eurocrypt 99*, Springer LNCS 1592, pages 415-431, 1999.

[22] R. Raz. A parallel repetition theorem. In *27th STOC*, pages 447-456, 1995.

[23] M. Tompa, H. Woll. Random Self-Reducibility and Zero Knowledge Interactive Proofs of Possession of Information. In *28th FOCS*, pages 472–482, 1987.

[24] A. Yao. Theory and Applications of Trapdoor Functions. In *23th FOCS*, pages 80-91, 1982.