An Efficient Parallel Repetition Theorem

Johan Håstad[1] and Rafael Pass[2] and Douglas Wikström[3] and Krzysztof Pietrzak[4]

[1] KTH, Stockholm, supported by ERC grant 226-203.
[2] Cornell University, Ithaca, supported in part by a Microsoft New Faculty Fellowship, NSF CAREER Award CCF-0746990, AFOSR Award FA9550-08-1-0197, and BSF Grant 2006317.
[3] KTH, Stockholm
[4] CWI, Amsterdam

Abstract. We present a general parallel-repetition theorem with an efficient reduction. As a corollary of this theorem we establish that parallel repetition reduces the soundness error at an exponential rate in any public-coin argument, and more generally, any argument where the verifier's messages, but not necessarily its decision to accept or reject, can be efficiently simulated with noticeable probability.

1 Introduction

When the soundness error of an interactive proof [7] or interactive argument [3], or more generally computationally-sound interactive proofs, is too large for applications, one might hope to prove a direct-product theorem which applies to the protocol at hand. A direct-product theorem for some class of problems states that if an adversary has some probability of succeeding in a single instance, then his chance in solving many independent instances of the problem drops exponentially. Running several independent instances of a protocol can be done sequentially or in parallel. Sequential repetition means that the $(i+1)$st execution of the protocol is only started after finishing the ith execution. Parallel repetition means that all protocols are run simultaneously. It is well-known that sequential repetition reduces the soundness error at an exponential rate for both proofs and arguments. However, although parallel repetition is known to reduce the soundness error in interactive proofs [1, 6], Bellare, Impagliazzo and Naor [2] demonstrate the existence of argument systems where parallel repetition does not reduce the soundness error, leaving open the following question:

For what computationally-sound proof systems does parallel repetition reduce the soundness error?

There have been several works addressing this question. Yao's [17] work on hardness amplification of one-way functions can be viewed as establishing that parallel repetition reduces the soundness error at an asymptotically optimal rate in every "publicly-verifiable" two-round argument—namely arguments where one can efficiently check if a transcript is accepting without knowing the

verifier's internal randomness. Bellare, Impagliazzo and Naor [2] extended this result to show that parallel repetition reduces the error for general (not necessarily publicly-verifiable) arguments with at most three rounds. For two-round protocols, Canetti, Halevi and Steiner [4] obtain a quantitatively better bound (approaching Yao's original bound for publicly-verifiable arguments), and Impagliazzo, Jaswal and Kabanets [11] show a more fine-grained "Chernoff-type" theorem. Finally, Pass and Venkitasubramaniam [13] show that parallel repetition also reduces the error for any constant-round public-coin protocol.

On the negative side, as shown by Bellare et al [2] and Pietrzak and Wikström [14], parallel repetition does not decrease the error for general (non public-coin) protocols with eight rounds; furthermore, black-box reductions cannot be used to establish such a result even for general four round protocols.

Thus, given the current state of the art, it is unknown whether parallel-repetition reduces the soundness error even in public-coin protocols with a super-constant number of rounds, or any general classes of non public-coin protocols with more than 3 rounds. The former of these questions was stated as an open problem by Bellare et al [2]. In this work we identify a general class of computationally sound protocols for which parallel repetition reduces the soundness error. This class encompasses—and significantly extends—all earlier classes of computationally sounds protocols for which parallel repetition had been established; in particular, it includes *all* public-coin protocols but also natural classes of private-coin protocols.

1.1 Our Results

We say that a verifier is δ-*simulatable* if, roughly speaking, given the prover's view of any partial interaction, with probability δ, the next-message function of the verifier (excluding its verdict to accept or reject) can be simulated for all remaining rounds (with a small statistical error). In other words, it is possible to efficiently simulate a δ-fraction of the verifier's continuations without knowing the verifier's internal randomness.

Note that any public-coin or three-round protocol trivially is 1-simulatable, but this notion captures many other protocols. For instance, public-coin protocols in the public-key model—where the verifier has a secret key and might determine whether to accept or reject based on this key—are also 1-simulatable.

Our main result is an efficient parallel repetition theorem (i.e., a parallel repetition theorem with an efficient reduction) for any $\frac{1}{poly}$-simulatable verifier. More precisely, our main theorem says that for any protocol where the verifier is δ-simulatable, we can turn an arbitrary parallel prover $\mathcal{P}^{(k)}$ for the k-fold repetition of $\mathcal{V}$ with success probability ϵ into a single instance prover $\tilde{\mathcal{P}}$ with success probability $1 - O\big(\frac{m}{\delta}\sqrt{-\log(\epsilon)/k} + \sqrt{m}\log(mk)/\sqrt{k}\big)$ where $2m+1$ is the number of rounds. Note that this implies that the error probability decreases exponentially down to some negligible function when the number of repetitions is sufficiently larger than the number of rounds. Following Impagliazzo et al. [11] we can actually prove a more general "Chernoff-type" theorem, where one only

assumes that the parallel prover convinces a certain fraction (and not all) of the individual verifiers.

As any public-coin protocol or three-round protocol satisfies 1-simulatability, we get as corollaries parallel repetition theorems for three-round protocols [2] and for public-coin protocols [13]. Note that whereas [13] only shows a parallel repetition theorem for *constant-round* protocols, our theorem applies to protocols with an arbitrary polynomial number of rounds. Our parameters are, however, worse that those of [13], which establishes an essentially optimal error reduction for the case of constant-round protocols.

As can be seen from the expression above, the success probability of the single-instance prover decreases linearly with the number of rounds in the protocol. If we restrict our attention to public-coin verifiers, or more generally, 1-simulatable verifiers *with verdict*—i.e., verifiers where the next messages function and its verdict to accept or reject—can be simulated with a small statistical error—then we can show a stronger parallel repetition theorem, where the decrease in error probability is independent of the number of rounds.

Finally, we show using a simple argument that our results hold also for *concurrent* provers, which may schedule their interaction with the individual verifiers arbitrarily.

1.2 Some history and related papers

An earlier version of this paper [9], where we established a parallel repetition only for interactive arguments with $(1 - \frac{1}{poly})$-simulatable verifiers (and some generalizations thereof), dates back to April 2008. Recent works extend it.

Most notably, Haitner [8] gave a modification of any interactive protocol by introducing a "random-termination verifier" where the verifier decides to stop and accept immediately with small but noticeable probability at each round. Haitner proved that any interactive protocol modified in this way, satisfies a parallel repetition theorem.

His construction is the main motivation of our study of δ-simulatable verifiers as it is easy to simulate a verifier that has halted. As a consequence our results gives a new proof of Haitner's theorem which is, in our eyes, simpler and which gives better parameters.

In an even more recent paper Chung and Liu [5] improves the analysis of our reduction. They manage to avoid the use of any lemma of the type obtained by Raz getting optimal reduction of the error rate for the public-coin case and almost optimal result in the case of 1-simulatable verifiers. It does not seem that their result extends to the case of δ-simulatable verifiers.

In a different direction, Pass, Tseng and Wikström [12] rely on our techniques to show that parallel repetition of public-coin protocols also gives a qualitative (rather than quantitative) improvement in soundness: any public-coin argument, when sufficiently repeated in parallel, becomes sound also against a "resetting"-attack if the verifier uses a pseudo-random function to pick its messages. As a corollary of this result, they establish impossibility of public-coin black-box

zero-knowledge protocols (for non-trivial languages) that remain secure under parallel repetition. Interestingly, [12] show that the dependence on m in our security reduction for the main theorem is inherent in their setting; this stands in contrast with our sharper reduction for the case of public-coin protocols.

1.3 Our Techniques

We show how to turn any parallel-prover $\mathcal{P}^{(k)}$ into a single-instance prover $\tilde{\mathcal{P}}$; furthermore, we require that $\tilde{\mathcal{P}}$'s success probability is significantly higher that of $\mathcal{P}^{(k)}$. Traditionally, $\tilde{\mathcal{P}}$ achieves this by internally incorporating $\mathcal{P}^{(k)}$, appropriately feeding it messages, while at the same time picking one of the parallel executions that it feeds to an external verifier. In other words, out of the k instances that $\mathcal{P}^{(k)}$ believes it is participating in, $\tilde{\mathcal{P}}$ controls $k-1$ of them, while one of them is externally forwarded.

The crux of this approach is how to determine the $k-1$ messages sampled in some particular round are good. In the public-coin case, in the work of Pass and Venkitasubramaniam [13], the "goodness" of a message is determined by estimating (using sampling) the probability with which $\tilde{\mathcal{P}}$ would be able to complete the partial interaction if this message was fixed; and $\tilde{\mathcal{P}}$ selects the message which leads to the highest success probability. This procedure requires recursively sampling $\tilde{\mathcal{P}}$ and results in a blow-up of the running-time as a function of the number of rounds and thus only a constant number of rounds can be handled. In the case of private-coin protocols, another problem arises already for the case of three-round protocols: we might not be able to determine if the verifier $\mathcal{V}_i$ accepts in a particular transcript as we do not know its random tape. Bellare et al. [2] overcome this problem by "guessing" that $\mathcal{V}_i$ accepts, if, intuitively, "many" other verifiers accept; as we are internally running all the other verifiers we know their random tapes and thus their decision.

A-priori, it would seem that a combination of these approaches would at least give a parallel-repetition theorem for constant-round private-coin protocols as long as it is possible to appropriately sample the next messages of the verifier. The problem is that when selecting "good" messages, we might be biasing the distribution of continued executions. It is, thus, no longer clear that the procedure of "guessing" that $\mathcal{V}_i$ accepts if many other verifiers accept, yields a good estimate of whether $\mathcal{V}_i$ actually accepts.

The key technique introduced in this paper is a method for selecting "good" messages without biasing the distribution too much. We essentially choose the first continuation that can be seen to lead to a good outcome. The fact that this procedure does not bias the distribution of interactions by too much follows from a powerful lemma of Raz [15] which was used in an essential way in the proof of the parallel repetition theorem for two-prover interactive proofs. Additionally, this approach does not lead to a blow-up in running-time and can be applied to any polynomial number of rounds.

Let us first outline the idea for the case of public-coin protocols. Instead of trying to recursively estimate how good a message is, we use the following simple

procedure to pick messages to forward to $\mathcal{V}_i$. Given a partial interaction, repeatedly sample random completions of this transcripts, until a successful transcript is reached, i.e., one where all verifiers accept. When this happens, select the next message to forward to the external verifier based on what that message was in the sampled accepting transcript. In other words, sample a random message conditioned on it leading to a successful transcript. To analyze why this works, consider the following mental experiment, where messages from $\tilde{\mathcal{P}}$ are determined in the same way, but now also $\mathcal{V}_i$'s messages are selected conditioned on them leading to an accepting execution. Clearly, in this mental experiment $\tilde{\mathcal{P}}$ succeeds in convincing $\mathcal{V}_i$ with probability 1. It is also not hard to see that the expected number of samples required by $\tilde{\mathcal{P}}$ is not too high and that its running-time is still polynomial. The problem is that the real external verifier does not pick its messages conditioned on them leading to an accepting execution; rather, it picks them uniformly at random. However, by relying on Raz' lemma, we can show, provided that i is picked uniformly at random from $[k]$, that the distribution of messages actually sent by the real external verifier are statistically close to those sent in the mental experiment, where we condition on them leading to an accepting execution. By applying the union bound over each round in the protocol, we conclude that also in the real execution, $\tilde{\mathcal{P}}$ succeeds which high probability.

Note that the above argument directly applies also to 1-simulatable verifiers with verdict; we only require it to be possible to 1) emulate continuations of partial interactions with the external verifier, and 2) determine if the external verifier would have accepted in those executions. To extend this analysis to 1-simulatable verifiers without verdict, we augment the argument by first showing that in the mental experiment it is sufficient to guess the decision of $\mathcal{V}_i$ based on the decisions of the other verifiers, in analogy with [2]. Now we can no longer claim that the success probability in the mental experiment is 1, but it will still be sufficiently high; the rest of the proof remains the same, and we conclude that also in the real execution $\tilde{\mathcal{P}}$ succeeds with high probability. We mention that to simplify the analysis, and to generalize it to handle "Chernoff-type" bounds, we generalize the "guessing" procedure of [2].

Finally, consider the case of $\frac{1}{poly}$-simulatable verifiers. Here we can only emulate continuations of the external verifier for a small, but noticeable, fraction of its true continuations. Nevertheless, by another application of Raz's lemma, we can show that the distribution of messages sent to the internal prover does not change by too much even if we condition the ith execution on a noticeable subset of continuations, and thus $\frac{1}{poly}$-simulatability suffices. More precisely, by Raz's lemma, it follows that the probability the external verifier chooses a continuation that we can simulate is not affected much if we condition on getting an accepting interaction; this, in particular means that (on average) the probability that a partial transcript leads to an accepting transcript does not change much even if we condition on only continuations that we can simulate.

Note that in the above proof sketch we lose a factor of m, i.e., the number of rounds in the protocol, by the application of the union bound. For the special

case of 1-simulatable verifiers with verdict, we go back to the underlying tool of *relative entropy* used to prove Raz's lemma, and use it to prove a generalization that considers multiple rounds at once, without losing the factor of m.

1.4 Outline of Paper

We first introduce some basic definitions in Section 2. Then we give a definition of δ-simulatable verifiers in Section 3. In Section 4 we state the parallel repetition theorem. Then in Section 5 we prove the general parallel repetition theorem, leaving the sharper theorem for the full version. Finally, we explain in Section 6 how to generalize our results to *concurrent* provers.

2 Notation and Basic Definitions

We denote the set $\{1, \ldots, m\}$ by $[m]$. We use n to denote the security parameter. All random variables are written in uppercase and usually we use the corresponding lower case for outcomes of the variable. When we say that a random variable X over a set $\mathcal{X}$ is chosen *randomly*, we mean that it is uniformly and independently distributed of all other variables. We use $\log a$ to denote the logarithm of a in base 2. We write $x \leftarrow_R X$ when x is chosen randomly from the set X.

If X is a random variable we write $\mathsf{P}_X(x) = \Pr[X = x]$ to denote the probability that it assumes the value x, and we denote its support by $[X]$. If X and Y are random variables we denote the conditional distributions of Y given X by $\mathsf{P}_{Y|X}$, and when we condition on a fixed value $x \in [X]$ we denote the corresponding probability function by $\mathsf{P}_{Y|X}(\cdot\,|x)$. Thus, $\mathsf{P}_{Y|X}(y\,|x) = \mathsf{P}_{XY}(x,y)/\mathsf{P}_X(x)$. When W is an event, we write $\mathsf{P}_{X|W}(x) = \Pr[X = x\,|W]$.

We often use the chain-rule for distributions and we use dots, when we are interested in a specific conditional distribution, e.g., we write $\mathsf{P}_{XY} = \mathsf{P}_Y \mathsf{P}_{X|Y}$ and $\mathsf{P}_{X|Y}(\cdot\,|y)$.

Definition 1. *The statistical distance between two distributions P_X and P_Y over a set $\mathcal{X}$ is*

$$\|\mathsf{P}_X - \mathsf{P}_Y\| = \frac{1}{2} \sum_{x \in \mathcal{X}} |\mathsf{P}_X(x) - \mathsf{P}_Y(x)| \ .$$

In a computationally sound protocol, soundness only holds against *efficient* (i.e., polynomial-time) provers. In general, a computationally sound protocol accepts a joint parameter λ that may, or may not, contain an instance of a language. We use $\mathcal{P}$ and $\mathcal{V}$ to denote the prover and verifier of a protocol, and we write $\langle \mathcal{P}, \mathcal{V} \rangle(\lambda)$ for the output of $\mathcal{V}$ after an interaction with $\mathcal{P}$ on common input λ. For notational convenience, we consider the security parameter n and any additional advice to the prover as encoded into λ. We denote the k-wise parallel repetition of a verifier $\mathcal{V}$ by $\mathcal{V}^k$. The repeated verifier simulates the individual verifiers independently, except that their message rounds are synchronized. It accepts if all the individual verifiers accept. The ith verifier is denoted by $\mathcal{V}_i$, but

all verifiers run the same program $\mathcal{V}$. We are also interested in repeated threshold verifiers, denoted by $\mathcal{V}_\gamma^k$, that accept if at least $(1-\gamma)k$ of the individual verifiers accept.

The number of exchanges in the protocol is denoted by m, where one exchange consists of two rounds, and the very first message of the prover is considered part of the 0th exchange.

We denote the lth message of the ith verifier $\mathcal{V}_i$ by $C_{l,i}$ and its state after the lth message has been computed by $T_{l,i}$. We denote the lth message sent by the prover to the ith verifier $\mathcal{V}_i$ by $A_{l,i}$, and we denote the state of the prover after it has computed its lth message by S_l. The decision of $\mathcal{V}_i$ is denoted by D_i, i.e., 1 for accept and 0 for reject. We define $C_l = (C_{l,1}, \ldots, C_{l,k})$ and $A_l = (A_{l,1}, \ldots, A_{l,k})$. The variables are then related as follows given a random joint parameter Λ

$$\begin{aligned} T_{0,i} &= \Lambda && (1)\\ (S_0, A_0) &= \mathcal{P}^{(k)}(\Lambda) \\ (T_{l+1,i}, C_{l+1,i}) &= \mathcal{V}_{R_{l,i}}(T_{l,i}, A_{l,i}) && \text{for } 0 \le l < m \\ (S_l, A_l) &= \mathcal{P}^{(k)}(S_{l-1}, C_l) && \text{for } 0 < l \le m \\ D_i &= \mathcal{V}(T_{m,i}, A_{m,i}) \ , \end{aligned}$$

where we think of both the prover and verifier as deterministic algorithms and denote the random tape used by $\mathcal{V}_i$ in round l by $R_{l,i}$. The verifier may of course "store" randomness from one round to be used in later rounds.

To collect random variables belonging to different exchanges we write, e.g., $C_{[l],i} = (C_{1,i}, \ldots, C_{l,i})$ and $C_{[l]} = (C_1, \ldots, C_l)$. Sometimes we wish to exclude only a single index i. Then we write $C_{l,\langle i \rangle} = (C_{l,1}, \ldots, C_{l,i-1}, C_{l,i+1}, \ldots, C_{l,k})$. We mostly view $\mathcal{V}$ and $\mathcal{P}^{(k)}$ as deterministic functions, but when convenient and clear from the context we drop the the random tape from our notation.

3 Simulatable Verifiers

Our parallel repetition theorem is applicable to δ-*simulatable* verifiers. Roughly speaking, we say that a verifier is δ-simulatable if given only the prover's view of any partial interaction (which thus excludes the verifier's internal state), we can efficiently simulate a δ fraction of the verifier's actual continuations.

Recall that given a prover $\mathcal{P}$ and a verifier $\mathcal{V}$, a partial transcript of length l is denoted $(\lambda, a_{[l]}, c_{[l]})$, the lth states of $\mathcal{P}$ and $\mathcal{V}$ are denoted s_l and t_l respectively, and that these values are defined formally by Equation (1) in Section 2. Thus, the prover's view after producing its lth message a_l is given by $(s_{[l]}, \lambda, a_{[l]}, c_{[l]})$.

Definition 2 (δ-Simulatable Verifier). *A verifier $\mathcal{V}$ is said to be* δ-simulatable *if there exists a PPT simulator $\mathcal{S}$ such that for every prover strategy $\mathcal{P}$ and every partial history $(s_{[l]}, t_{[l]}, \lambda, a_{[l]}, c_{[l]})$, there is a subset Δ of $\mathcal{V}$'s random tapes, compatible with the history so far, of density δ such that the output of $\mathcal{S}$ on input $(s_{[l]}, \lambda, a_{[l]}, c_{[l]})$ is statistically close to the prover's view of a continued interaction between $\mathcal{P}$ and $\mathcal{V}$, including $\mathcal{V}$'s verdict, when $\mathcal{V}$'s random tape is chosen*

uniformly from Δ. When the verdict of $\mathcal{V}$ is removed from consideration, we say that $\mathcal{V}$ is δ-simulatable without verdict, or simply δ-simulatable.

Remark 1. Note that the definition requires the simulator to simulate a probability distribution that is allowed to be dependent on the state of the verifier and that this state is unknown. This seems like an impossible task in general unless we minimize the information contained in the state. In the early version of this paper [9] this state was not included in the probability distribution but instead we required that the next message of the internal, fully simulated verifiers could be efficiently generated based on the conversation up to this point. If this is indeed possible then we can instead let the state be given by the messages already sent and then use this generation process to replace the original verifier. With the current definition we need no condition on the internal verifiers and hence it is, in our eyes, preferable.

Remark 2. The property that we only demand the two distributions to be statistically close and not identical is only a technicality. In fact, when using the definition in this abstract we assume that the two distributions are the same, to avoid cumbersome notation to take care of the error terms given by a small statistical distance.

Remark 3. A careful reading of the proof reveals that we can let the probability δ of successful sampling depend on the round but not on the partial history it extends.

Remark 4. We can allow a weaker definition of simulatability where the ability to simulate $\mathcal{V}$ also depends on the $\mathcal{P}$'s messages. This leads to a more complicated proof of Lemma 4 that either loses a factor of m in the error bounds or uses the methods of [16] to get the same bounds. In order to keep this extended abstract self-contained we use the weaker definition here.

Clearly, any public-coin protocol is 1-simulatable with verdict. It is also easy to see that the "random-termination verifiers" of Haitner are $\frac{1}{4m}$-simulatable with verdict: the simulator simply aborts (accepting) with probability $\frac{1}{4m}$. Furthermore, public-coin protocols in a public-key model (where the verifier only sends random messages, but bases is decision on its secret key), as well as three-round protocols, are 1-simulatable without verdict.

4 The Parallel Repetition Theorem

We prove a parallel repetition theorem for any verifier that is δ-simulatable without verdict. The theorem implies that a $(2m+1)$-round protocol when repeated $k = \Omega(\frac{m^2}{\delta^2}t)$ times in parallel reduces the error probability from $1/2$ to $2^{-t} + \mathsf{negl}(n)$ if we require that all parallel verifiers accept. In the general statement we consider also repeated threshold verifiers $\mathcal{V}_\gamma^k$ that accept if at least $(1-\gamma)k$ of the k parallel verifiers accept.

Theorem 1. *Assume $\epsilon \leq 1/2$, let $\mathcal{V} \in \mathsf{PPT}$ be a verifier that is δ-simulatable without verdict, and let $\mathcal{P}^{(k)}$ be a polynomial-time parallel prover. Then there exists a prover $\tilde{\mathcal{P}}$ running in time $\mathsf{Poly}(n, k, m, 1/\epsilon)$ such that for every $\lambda \in \{0,1\}^*$ where $\Pr[\langle \mathcal{P}^{(k)}, \mathcal{V}_\gamma^k \rangle(\lambda) = 1] \geq \epsilon$, for some threshold $0 \leq \gamma < 1$,*

$$\Pr\left[\langle \tilde{\mathcal{P}}, \mathcal{V} \rangle(\lambda) = 1\right] \geq 1 - \gamma - O\left(\frac{m}{\delta}\sqrt{-\log(\epsilon)/k} + \sqrt{m}\log(mk)/\sqrt{k}\right) \ ,$$

where n is the security parameter, m is the number of messages sent by $\mathcal{V}$, and k is the number of verifiers interacting with the parallel prover.

The constants hidden in the $O(\cdot)$-notation in Theorem 1 are small and given explicitly in our proof. It turns out that in the case of 1-simulatable verifiers we can get a stronger theorem.

Theorem 2. *Assume $\epsilon \leq 1/2$, let $\mathcal{V} \in \mathsf{PPT}$ be a verifier, and let $\mathcal{P}^{(k)}$ be a polynomial-time parallel prover. Then there exists a prover $\tilde{\mathcal{P}}$ running in time $\mathsf{Poly}(n, k, m, 1/\epsilon)$ such that for every $\lambda \in \{0,1\}^*$ where $\Pr[\langle \mathcal{P}^{(k)}, \mathcal{V}_\gamma^k \rangle(\lambda) = 1] \geq \epsilon$, for some threshold $0 \leq \gamma < 1$,*

1. ***if $\mathcal{V}$ is 1-simulatable with verdict**, then*

$$\Pr\left[\langle \tilde{\mathcal{P}}, \mathcal{V} \rangle(\lambda) = 1\right] \geq 1 - \gamma - 2\sqrt{-\log(\epsilon)/k} - \sqrt{1/k} \ , \quad \textit{and}$$

2. ***if $\mathcal{V}$ is 1-simulatable without verdict**, then*

$$\Pr\left[\langle \tilde{\mathcal{P}}, \mathcal{V} \rangle(\lambda) = 1\right] \geq 1 - \gamma - O\big(\sqrt{m}\sqrt{-\log(\epsilon)/k} + \sqrt{m}\log(mk)/\sqrt{k}\big) \ ,$$

where n is the security parameter, m is the number of messages sent by $\mathcal{V}$, and k is the number of verifiers interacting with the parallel prover.

Due to the lack of space the proof of Theorem 2 is omitted but can be found in [16]. It relies on the notion of *relative entropy (Kullback-Leibler distance)* and uses a lemma extending Lemma 1 below to treat multiple rounds.

Readers familiar with the recent result of Pass et al. [12], may find Case 1 of Theorem 2 surprising, since superficially it seems the same technique should be applicable to remove the dependence on the number of rounds in [12], which would contradict their results. The reason this is not the case is that in [12], the reduction samples messages in a given round conditioned on *two* events: (1) that "all verifiers accept", and (2) that the "right" message is output by the embedded "resetting" attacker. Thus, in each round a *distinct* event is considered. Another way to say this is that the probability that the "right" messages are output in all rounds in a straight-line execution of the resetting attacker is $\mathsf{Poly}(n)^{-m}$. Thus, we could apply this technique to simplify the proof in [12], but the dependence on m would not disappear.

5 Proof of Theorem 1

We prove Theorem 1 in three steps. First we prove the theorem for public-coin verifiers in the case where $\gamma = 0$. This immediately generalizes to 1-simulatable verifiers with verdict. Then we show how to generalize the proof to verifiers that are only δ-simulatable with verdict. Finally, we prove that the result can be generalized to $\gamma > 0$ and verifiers that are δ-simulatable without verdict.

5.1 Proof of Theorem 1 in the Public-Coin Case

It is quite natural to simulate an interaction between the parallel prover $\mathcal{P}^{(k)}$ and the repeated verifier $\mathcal{V}^k$ and let the external verifier play the role of $\mathcal{V}_i$ for some i. In other words any message to $\mathcal{V}_i$ would instead be forwarded to the external verifier and its reply is taken as the reply of $\mathcal{V}_i$. The question is how to choose the index i and how the other verifiers should be simulated. We solve this in a simple way by picking a uniformly random i, simulating the other verifiers by picking random messages and then taking the first answers that can be seen to lead to making all verifiers accept. Let us discuss the intuition behind this approach.

Consider the tree of all possible interactions between $\mathcal{P}^{(k)}$ and $\mathcal{V}^k$, where each leaf encodes which verifiers accept and the edges on level l are labeled with the random choices of the verifiers in exchange l. If we could sample a random leaf such that all verifiers accept, then clearly $\mathcal{V}_i$ also accepts for a any choice of i. If the success probability of $\mathcal{P}^{(k)}$ is ϵ we can efficiently sample from this distribution in time polynomial in $1/\epsilon$ and the security parameter n as follows. In exchange l we repeatedly choose the messages $c_l = (c_{l,1}, \ldots, c_{l,k})$ of all verifiers randomly and simulate a completion conditioned on the interaction so far and our choice of messages in exchange l. If the completion gives a leaf where all verifiers accept, then we take c_l to be the messages of the verifiers in exchange l. Clearly, if a suitable c_l is found for each l, then all verifiers accept.

Suppose now that we pick a random index i and in exchange l pick the message $c_{l,i}$ of $\mathcal{V}_i$ only once. The messages $c_{l,\langle i \rangle} = (c_{l,1}, \ldots, c_{l,i-1}, c_{l,i+1}, \ldots, c_{l,k})$ of all other verifiers are still repeatedly sampled, but now conditioned on $c_{l,i}$ in addition to the interaction so far. The key observation is that this modified distribution is quite close to the original one, and that we may view $\mathcal{V}_i$ as the external verifier. Thus, we avoid sampling too much to stay close to the original distribution on the leaves where all verifiers accept.

More Details. Denote by $\mathsf{Complete}$ the probabilistic algorithm that given a partial interaction between $\mathcal{P}^{(k)}$ and $\mathcal{V}^k$ returns a random sample from the distribution of the decisions of the verifiers, conditioned on the partial interaction given as input. The detailed reduction is given by Algorithm 1 below. The parameter u denotes the maximal number of samples generated by the prover in each round to find a suitable reply from the parallel prover. For simplicity we assume that the message of the verifier in each exchange is drawn from $\{0,1\}^{p(n)}$ for some polynomial p.

Algorithm 1. $\tilde{\mathcal{P}}_u(x)$

if x is a joint parameter λ **then** // *Read joint parameter*
  $(s_0, a_0) \leftarrow \mathcal{P}^{(k)}(\lambda)$ // *Compute prover's first message*
  $i \leftarrow_R [k]$ // *Choose random index*
  return $\left(\left[i, s_0, \lambda, \emptyset, a_{[0]}\right], a_{0,i}\right)$ // *Output state and first message*
else
  Interpret x as $\left(\left[i, s_{l-1}, \lambda, c_{[l-1]}, a_{[l-1]}\right], c_{l,i}\right)$ // *Read state & verifier's message*
  for $v = 1, \ldots, u$ **do**
    $c_{l,\langle i\rangle} \leftarrow_R \{0,1\}^{p(n)\times(k-1)}$ // *Sample verifiers' messages*
    $(s_l, a_l) \leftarrow \mathcal{P}^{(k)}(s_{l-1}, c_l)$ // *Compute prover's reply*
    if $\mathsf{Complete}(\lambda, c_{[l]}, a_{[l]}) = \overline{1}$ **then** // *If messages are good,*
      return $\left(\left[i, s_l, \lambda, c_{[l]}, a_{[l]}\right], a_{l,i}\right)$ // *then output reply*
    done
  done
  return $(fail, fail)$ // *Give up*
end

Note that the prover keeps as its state the index i corresponding to the external verifier, the state of the simulated parallel prover, and a partial interaction. We now consider the error probability of the constructed prover.

The sampling lemma of Raz [15] says that given independently distributed random variables $U_1, \ldots, U_k$, the distribution of U_i does not change much on average over the index i by conditioning on an event E, provided that the probability of E is not too small. (We mention that the sampling lemma was previously used by Impagliazzo et al. [11] in the context of parallel-repetition of 2-round arguments). We make use of the following variant that appears as Corollary 6 in Holenstein's simplified proof of Raz' theorem [10].

Lemma 1. *[10] Let* $\mathsf{P}_{YU^kV} = \mathsf{P}_Y\left(\prod_{i=1}^k \mathsf{P}_{U_i|Y}\right)\mathsf{P}_{V|YU^k}$ *be a probability distribution and* E *an event. Then*

$$\frac{1}{k}\sum_{i=1}\left\|\mathsf{P}_{YU_iV|E} - \mathsf{P}_{YV|E}\,\mathsf{P}_{U_i|Y}\right\| \leq k^{-1/2}\sqrt{\log|V^*| - \log\Pr[E]}\ .$$

where V^* *is the set of values of* v *that can occur conditioned on* E *occurring.*

In our application, the variable Y represents the interaction so far and U_i are the messages of the verifiers in the current round. We let V be a binary variable such that $\mathsf{P}_{V|YU^k}(1\,|y,u)$ is the probability that all verifiers accept in a random completion, for every partial interaction $(y,u) \in [Y, U^k]$. The lemma then implies for a random Y that most U_i are, even if we condition on extending Y to an accepting interaction, distributed very closely to their unconditional distribution which in this case is the uniform distribution.

Thus, we can conclude that in any single round, if we have chosen Y up to this point with the conditional distribution of a partial interaction leading to an accepting leaf then if we, in this round, pick a random i, the distribution of U_i

is likely to be close to uniform. A problem to be taken care of is that i is chosen once and remains fixed for all rounds.

Let us consider a modified process where the external verifier $\mathcal{V}_i$ instead of choosing $c_{l,i}$ with the uniform distribution does a process similar to that of $\tilde{\mathcal{P}}_u$. It samples complete interactions that extend the current interaction of all verifiers until it finds a complete interaction where all verifiers accept and then chooses the value of $c_{l,i}$ in this interaction as its response. Furthermore, let us remove the restriction that $\tilde{\mathcal{P}}_u$ only makes u attempts to find a complete interaction where all verifiers accept and let it sample until it finds a completion. Let D_{real} be the distribution on interactions produced by $\tilde{\mathcal{P}}_u$ interacting with $\mathcal{V}_i$ and let D_{ideal} be the distribution on interactions in this modified process.

Clearly, D_{ideal} outputs a uniformly selected interaction in which all verifiers accept. Thus, in this modified process $\mathcal{V}_i$ always accepts. Below we estimate the statistical distance between this process and the original process. This statistical distance is an upper bound on the probability that $\mathcal{V}_i$ rejects. Let us first see that it is unlikely that the modified process ever needs to sample a large number of times. This is intuitively not surprising. For the sampling to take a long time we need to choose a partial interaction that is very unlikely to lead to a complete accepting interaction. But as we are choosing partial interactions as part of an accepting interaction we are very unlikely to choose such a partial interaction. This is made formal by the following easy lemma, a proof of which is given below.

Lemma 2. *Let Y be a random variable and let $X_0, X_1, X_2, \ldots$ be identically distributed binary random variables which are only dependent through Y, i.e., $\mathsf{P}_{Y,X_0,\ldots X_j} = \mathsf{P}_Y \prod_{i=0}^{j} \mathsf{P}_{X_i|Y}$ and $\mathsf{P}_{X_i|Y} = \mathsf{P}_{X_j|Y}$ for any i, j. Let J be the random variable denoting the smallest nonzero index such that $X_J = 1$. Then $\mathrm{E}[J \,|X_0 = 1] \leq \frac{1}{\Pr[X_0=1]}$.*

Let us see how this lemma proves that the expected number of samples needed to find an accepting completion is small. We let Y be a random partial interaction which is chosen by picking a complete accepting interaction, i.e., Y is $C_{[l-1]}$ for some l, and we let X_i be one if a particular random completion of Y makes all verifiers accept. Then $\mathrm{E}[J \,|X_0 = 1]$ is exactly the expected number of attempts to complete the interaction Y to make all verifiers accept given that Y was picked by first picking a complete interaction which makes all verifiers accept and then truncating to the appropriate length.

Let $\delta = \sqrt{-\log(\epsilon)/k} + (\epsilon u)^{-1}$. We claim that the statistical difference between D_{real} and D_{ideal} when truncated to t rounds is bounded by $t\delta$. This is clearly true for $t = 0$ and we proceed by induction using the following lemma.

Lemma 3. *Let X_0 and X_1 be two random variables over $\mathcal{X}$, and let Z_x and Z'_x be two families of random variables parameterized by $x \in \mathcal{X}$ such that*

$$\|\mathsf{P}_{X_0} - \mathsf{P}_{X_1}\| = \delta_1 \quad \textit{and} \quad \mathrm{E}_x\left[\left\|\mathsf{P}_{Z_x} - \mathsf{P}_{Z'_x}\right\|\right] = \delta_2 \ ,$$

where x is distributed according to P_{X_0}. Then

$$\left\|\mathsf{P}_{X_0,Z_{X_0}} - \mathsf{P}_{X_1,Z'_{X_1}}\right\| \leq \delta_1 + \delta_2 \ .$$

Before we prove Lemma 3, let us see how it enables us to complete the induction step. We let X_0 be a $(t-1)$-round interaction chosen according to D_{ideal}, X_1 a $(t-1)$ round interaction chosen according to D_{real}, Z_{X_0} the next round message chosen by the verifiers according to D_{ideal} and Z'_{X_0} the next round message chosen from D_{real}. We need to estimate the expected statistical distance between Z'_{X_0} and Z_{X_0} over X_0.

We have two differences between the two distributions, how $\mathcal{V}_i$'s message is chosen and the limited sampling. The latter is, by Lemma 2 and Markov's inequality, bounded by $(\epsilon u)^{-1}$ and we claim that former difference is bounded by $\sqrt{-\log(\epsilon)/k}$. Let us see how this follows from Lemma 1.

As stated before, we let Y be the interaction up to the $(t-1)$st round and U_i the message of $\mathcal{V}_i$ in round t and V a bit which is one with the probability that a random completion of the given interaction accepts. The event E is that "$V = 1$". Then D_{ideal} picks messages with the distribution given by $\mathsf{P}_{U_i|YVE}$ while $\mathcal{V}_i$ picks messages with the uniform distribution which in this case is $\mathsf{P}_{U_i|Y}$. Lemma 1 now tells us exactly that for a random Y and i the statistical distance between these two distributions is at most $\sqrt{-\log(\epsilon)/k}$.

Finally, setting $u = \epsilon^{-1} m\sqrt{k}$ completes the proof of Theorem 1 in the public-coin case as claimed. The missing proofs of Lemma 3 and Lemma 2 are given below.

Proof (Lemma 2). We can consider only values y such that $\Pr[X_0 = 1 \,|Y = y] > 0$ and summing over those we have

$$\begin{aligned}
\mathrm{E}[J\,|X_0 = 1] &= \sum_y \Pr[Y = y\,|X_0 = 1]\,\mathrm{E}[J\,|Y = y \wedge X_0 = 1] \\
&= \sum_y \Pr[Y = y\,|X_0 = 1] \,/\, \Pr[X_1 = 1\,|Y = y \wedge X_0 = 1] \\
&= \sum_y \Pr[Y = y\,|X_0 = 1] \,/\, \Pr[X_1 = 1\,|Y = y] \\
&= \sum_y \frac{\Pr[Y = y \wedge X_1 = 1]}{\Pr[X_0 = 1]} \cdot \frac{\Pr[Y = y]}{\Pr[X_1 = 1 \wedge Y = y]} \leq \frac{1}{\Pr[X_0 = 1]}\ ,
\end{aligned}$$

where the third equality follows from the conditional independence of the X_i's and the fourth equality follows since the X_i's are also identically distributed. □

Proof (Lemma 3). We use the characterization that two distributions are at statistical distance δ if and only if there is a coupled way of choosing elements from the two distributions such that the two samples are equal with probability $1-\delta$. We need to choose coupled pairs (x, z) and (x', z') from the given distributions. First choose a coupled pair (x, x') distributed according to P_{X_0} and P_{X_1}, respectively. If they are unequal, which happens with probability δ_1, we give up. If they are are equal we choose a coupled pair (z, z') according to the distributions P_{Z_x} and $\mathsf{P}_{Z'_x}$. The probability that these are unequal (over the choice of x and the second choice) is upper bounded by δ_2. This completes the proof. □

5.2 Proof of Theorem 1 for δ-Simulatable Verifiers With Verdict

When the verifier is no longer public-coin and only δ-simulatable for some $\delta \geq 1/\mathsf{Poly}(n)$, it may keep its state hidden from the prover inbetween exchanges. To deal with this, we replace each call to $\mathsf{Complete}$ in Algorithm 1 by a call to the δ-simulator on input $(i, s_{[l]}, t_{[l],\langle i\rangle}, \lambda, a_{[l]}, a_{[l]})$.

We consider a fixed round l and all variables below depend on the value of l but, for notational convenience, we omit this dependence. Let us define $X_i = (T_{l-1}, C_{[l-1]}, C_{l,i})$ and $Y_i = (T_{l,\langle i\rangle}, C_{l,\langle i\rangle})$. Recall that $C_{l,\langle i\rangle}$ denotes the array $(C_{l,1}, \ldots, C_{l,i-1}, C_{l,i+1}, \ldots, C_{l,k})$ and similarly for $T_{l,\langle i\rangle}$.

By δ-simulatability, there is a subset, Δ of the external verifiers possible random tapes for which we can simulate $\mathcal{V}_i$.

Let W be an indicator variable of the event $D = \overline{1}$ (that all verifiers accept). Then define $\delta^i_{x_i,y_i}$ as the probability that the prover's view of a random completion of (x_i, y_i), conditioned on the event $W = 1$, is an output from the simulator. Furthermore, let $\delta^i_{x_i}$ be the expected value of $\delta^i_{x_i,y_i}$ over y_i, where y_i is chosen according to the distribution $\mathsf{P}_{Y_i|X_i,W}\ (\,\cdot\,|x_i, 1)$. Due to the conditioning on $W = 1$, δ-simulatability does not immediately say anything about these quantities, but for any fixed x_i the distribution of Y_i conditioned on both $W = 1$ and the event that the output is from simulator is given by the probability function

$$\mathsf{P}_{Y_i|X_i,W}\ (y_i\,|x_i, 1)\,\frac{\delta^i_{x_i,y_i}}{\delta^i_{x_i}}\quad .$$

We want to prove that this, for a uniformly random i, is statistically close to the distribution $\mathsf{P}_{Y_i|X_i,W}\ (\,\cdot\,|x_i, 1)$ and thus we should estimate

$$\frac{1}{k}\sum_{i=1}^{k}\sum_{x_i,y_i}\mathsf{P}_{X_i,Y_i|W}\ (x_i, y_i\,|1)\left|1 - \frac{\delta^i_{x_i,y_i}}{\delta^i_{x_i}}\right|\quad . \tag{2}$$

The following lemma is the key to estimating this distance.

Lemma 4.

$$\frac{1}{k}\sum_{i=1}^{k}\sum_{x_i,y_i}\mathsf{P}_{X_i,Y_i|W}\ (x_i, y_i\,|1)\,\left|\delta^i_{x_i,y_i} - \delta\right| \leq O\big(\sqrt{-\log(\epsilon)/k}\big)\quad . \tag{3}$$

We postpone the proof of the lemma until we have seen how it is used. Fix i and x_i and consider the contribution to the sums in (2) and (3) over a random Y_i conditioned on $W = 1$. Define a random variable Z which takes the value $\delta^i_{x_i,y_i}/\delta^i_{x_i}$ with probability $\mathsf{P}_{Y_i|X_i,W}\ (y_i\,|x_i, 1)$. Then the contribution to Equation (3) is at most $\delta\,\mathrm{E}\,[|1 - sZ|]$ with $s = \delta^i_{x_i}/\delta$ while the contribution to Equation (2) is $\mathrm{E}\,[|1 - Z|]$. Now consider the following lemma.

Lemma 5. *Assume that Z is a positive random variable with $\mathrm{E}\,[Z] = 1$. Then for any $s > 0$ we have $\mathrm{E}\,[|1 - Z|] \leq 2\,\mathrm{E}\,[|1 - sZ|]$.*

Again, we postpone the proof until we have completed the argument. Since $\mathrm{E}[Z] = 1$, we see that Equation (2) is bounded by $O(\delta^{-1}\sqrt{-\log(\epsilon)/k})$. Thus the additional statistical distance between the ideal distribution and that obtained by our parallel prover introduced in round l is bounded by this quantity.

Using coupling and the union bound as in Section 5.1, we conclude that replacing the 1-simulator by a δ-simulator introduces an additional error of at most $O(\frac{m}{\delta}\sqrt{-\log(\epsilon)/k})$.

Finally, let us prove the two lemmas above, completing the proof of Theorem 1 in this case.

Proof (Lemma 4). We apply Lemma 1 with U_i representing $\mathcal{V}_i$'s random tape compatible with the interaction up to this point. We need to analyze the probability that we can simulate $\mathcal{V}_i$ conditioned upon all verifiers accepting. Without conditioning this probability is statistically close to δ by the definition of δ-simulatability (for notational convenience we assume here that this probability equals δ). The deviation from this is bounded by the statistical distance of the conditioned distribution from the uniform distribution. The lemma now follows from Lemma 1. □

Proof (Lemma 5). Note that $\sum_{z \leq 1} \mathsf{P}_Z(z)(1-z) = \frac{1}{2}\mathrm{E}[|1-Z|]$, since $\mathrm{E}[Z] = 1$ and $|1-z|$ is symmetric around 1. If $s \leq 1$, then $|1-z| < |1-sz|$ for every $z \leq 1$ and the claim follows. If $s > 1$, then we instead consider the partial sum for $z > 1$ and apply the corresponding argument. □

5.3 Proof of Theorem 1 For δ-Simulatable Verifiers Without Verdict

First we note that it is easy to generalize the above result to the case with a repeated threshold verifier that accepts if at least $(1-\gamma)k$ verifiers accept. Replace the definition of the indicator variable W such that it is one if and only if $\sum_{i=1}^{k} D_i \geq (1-\gamma)k$. Then in the corresponding "modified process" discussed in Section 5.1 the probability that $\mathcal{V}_i$ accepts is at least $1-\gamma$, since i is chosen uniformly in $[k]$ and independently of the "modified process". A trivial modification of the analysis above then gives the same additional statistical error due to having an external verifier, the use of limited sampling, and a δ-simulatable verifier.

To generalize the theorem to δ-simulatable verifiers *without verdict*, starting from the result established in Section 5.2 for δ-simulatable verifiers *with* verdict, we modify the reduction by redefining W using "soft" decisions as was already done in [2]. Suppose that instead of accepting only samples where at least $(1-\gamma)k$ verifiers accept, we define a binary random variable W that is one with probability $\min(1, 2^{\nu(\gamma k - z)})$, where z is the number of rejecting verifiers, and accept a sample if $W = 1$. Then it turns out that, provided that, we choose ν small enough, this acceptance criteria can be approximated well even if we do not know the verdict of the external verifier $\mathcal{V}_i$. Let us start with the key lemma, of which the proof is postponed to the end of this section.

Lemma 6 (Soft Decision). *Let $D_1, \ldots, D_k$ be binary random variables such that $\Pr[\sum_{i=1}^{k} D_i \geq (1-\gamma)k] \geq \epsilon$, let $Z = k - \sum_{i=1}^{k} D_i$, let $\gamma > 0$, $\nu > 0$, and $m \geq 1$, and let W be a binary random variable such that $\Pr[W = 1 \,|\, Z = z] = \min(1, 2^{\nu(\gamma k - z)})$. Then*

$$\frac{1}{k}\sum_{i=1}^{k} \Pr[D_i = 0 \,|\, W = 1] \leq \gamma + \frac{1}{k\nu}(\log m + \log k - \log \epsilon) + \frac{4}{\nu^2 m k^2} \ .$$

Remark 5. Although setting $\nu = 1$ and $\gamma = 0$ recovers the decision procedure in [2], our analysis differs from theirs. They implicitly use Raz's lemma to argue that the variables W_i and W are close in distribution on *average* over i. We need the stronger statement that these variables are close in distribution for *any* i. This is why we need the additional parameter ν.

Now set $\nu = \frac{1}{\sqrt{m}}\sqrt{-\log(\epsilon)/k}$ and suppose now at first that we did know the verdict of $\mathcal{V}_i$. The old argument carries over and we end up at a random point where $W = 1$. Before we could conclude that $\mathcal{V}_i$ accepted while currently by applying Lemma 6, we see that the probability that $\mathcal{V}_i$ rejects is at most

$$\begin{aligned}
&\gamma + \frac{1}{k\nu}(\log m + \log k - \log \epsilon) + \frac{4}{\nu^2 m k^2} \\
&= \gamma + \frac{\sqrt{m}}{\sqrt{-\log(\epsilon)k}}(\log(mk) - \log(\epsilon)) + \frac{1}{-\log(\epsilon)k} \\
&\leq \gamma + \sqrt{m}\log(mk)/\sqrt{k} + \sqrt{m}\sqrt{-\log(\epsilon)/k} + 1/k \ ,
\end{aligned}$$

and this is enough to prove Theorem 1.

The key to case the case when we do not know the verdict of $\mathcal{V}_i$ is that if ν is small then the decision of an individual verifier is does not affect the behavior very much. In fact, let us simply approximate Z by assuming than $D_i = 1$ and let us run our parallel prover using this approximation. Compare a run of this modified prover and a run of an ideal prover that uses the correct value of Z using the same randomness.

These two provers only behave differently when the the modified prover accepts a history that the ideal prover would have rejected. To be precise, each time the modified prover accepts a history the probability that the ideal prover would have rejected the same history is $1 - 2^{-\nu} \leq \nu$.

As the modified prover only accepts m histories over the course of a run, the statistical difference between the behavior of modified prover and the ideal prover is bounded by νm.

This gives a total additional error from using soft decisions when sampling of $(m+1)\nu \leq (\sqrt{m}+1)\sqrt{-\log(\epsilon)/k}$. Combined with the proof of Lemma 6 below, this concludes the proof of Theorem 1 in its full generality.

Proof (Lemma 6). Let $p_j = \Pr[Z = j]$. We know by assumption that

$$\sum_{j=0}^{k\gamma} p_j \geq \epsilon \ . \tag{4}$$

We know that $\Pr[Z = j \,|W = 1]$ is proportional to $p_j 2^{-\min(0,\nu(j-\gamma k))}$. This implies that the expected number of D_i's equal to zero is

$$\mathrm{E}[Z \,|W = 1] = \frac{\sum_{j=0}^{k} j p_j 2^{-\min(0,\nu(j-\gamma k))}}{\sum_{j=0}^{k} p_j 2^{-\min(0,\nu(j-\gamma k))}} \ . \tag{5}$$

The denominator is lower bounded by $\sum_{j=0}^{\gamma k} p_j 2^{-\min(0,\nu(j-\gamma k))} = \sum_{j=0}^{\gamma k} p_j$ and is thus, by Equation (4), at least ϵ. Let t be a parameter to be determined, then the numerator is bounded by

$$\begin{aligned} &\sum_{j=1}^{k} \max(\gamma k + t, j) p_j 2^{-\min(0,\nu(j-\gamma k))} \\ &\leq (\gamma k + t) \sum_{j=1}^{k} p_j 2^{-\min(0,\nu(j-\gamma k))} + \sum_{j=1}^{k-(\gamma k+t)} j p_{\gamma k+t+j} 2^{-\nu(t+j)} \ . \end{aligned} \tag{6}$$

It is not difficult to see that $\sum_{j=1}^{\infty} j 2^{-\nu j} \leq \frac{4}{\nu^2}$ and thus the upper bound in Equation (6) is at most

$$(\gamma k + t) \sum_{j=1}^{k} p_j 2^{-\min(0,\nu(j-\gamma k))} + \frac{4}{\nu^2} 2^{-\nu t} \ .$$

Setting $t = \frac{1}{\nu}(\log m + \log k - \log \epsilon)$ and using that the denominator of Equation (5) is at least ϵ we see that

$$\mathrm{E}[Z \,|W = 1] \leq \gamma k + \frac{1}{\nu}(\log m + \log k - \log \epsilon) + \frac{4}{\nu^2 m k} \ .$$

The proof is concluded by remembering that i is chosen uniformly at random from $[k]$. □

6 Concurrent Repetition

Although verifiers repeated in parallel perform their computations independently and use independently generated randomness, their communication is *synchronized*. It is natural to consider a more general form of repetition where this restriction is removed, i.e., the prover may *arbitrarily schedule* its interaction with the individual verifiers.

Only minor modifications are needed to generalize Theorem 1 and Theorem 2, with the same parameters, to the setting where a concurrent prover interacting with the k-wise *concurrent* repetition of $\mathcal{V}$ is converted into a prover $\tilde{\mathcal{P}}$ interacting with $\mathcal{V}$. The key observation for this extension is that a concurrent prover only sends $m + 1$ messages to $\mathcal{V}_i$. Thus, $\tilde{\mathcal{P}}$ need only sample completions at m points during an interaction with $\mathcal{V}$, and Lemma 1 is only applied m times. Furthermore, the δ-simulator and soft decisions are only used at each point where $\tilde{\mathcal{P}}$ samples completions, i.e., exactly m times. More details will be given in the full version of this paper.

References

1. L. Babai. Trading group theory for randomness. In *17th ACM Symposium on the Theory of Computing (STOC)*, pages 421–429. ACM Press, 1985.
2. M. Bellare, R. Impagliazzo, and M. Naor. Does parallel repetition lower the error in computationally sound protocols? In *38th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 374–383. IEEE Computer Society Press, 1997.
3. G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.
4. R. Canetti, S. Halevi, and M. Steiner. Hardness amplification of weakly verifiable puzzles. In *2nd Theory of Cryptography Conference (TCC)*, volume 3378 of *Lecture Notes in Computer Science*, pages 17–33, 2005.
5. K-M. Chung and Feng-Hao Liu. Parallel repetition theorems for interactive arguments. these proceedings.
6. O. Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*. Springer-Verlag, Algorithms and Combinatorics, 1998.
7. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.
8. I. Haitner. A parallel repetition theorem for any interactive argument. In *50th IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society Press, 2009.
9. J. Håstad, R. Pass, Pietrzak, and D. Wikström. An efficient parallel repetition theorem. Manuscript, April 2008.
10. T. Holenstein. Parallel repetition: simplifications and the no-signaling case. In *39th ACM Symposium on the Theory of Computing (STOC)*, pages 411–419. ACM, 2007.
11. R. Impagliazzo, R. Jaiswal, and V. Kabanets. Chernoff-type direct product theorems. In *Advances in Cryptology – Crypto 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 500–516. Springer, 2007.
12. R. Pass, D. Tseng, and D. Wikström. On the composition of public-coin zero-knowledge protocols. In *Advances in Cryptology – Crypto 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 160–176. Springer Verlag, 2009.
13. R. Pass and M. Venkitasubramaniam. An efficient parallel repetition theorem for arthur-merlin games. In *39th ACM Symposium on the Theory of Computing (STOC)*, pages 420–429. ACM, 2007.
14. K. Pietrzak and D. Wikström. Parallel repetition of computationally sound protocols revisited. In *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 86–102, 2007.
15. R. Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.
16. D. Wikström. An efficient concurrent repetition theorem. `http://eprint.iacr.org/`, 2009.
17. A. C. Yao. Theory and application of trapdoor functions. In *23rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 80–91. IEEE Computer Society Press, 1982.