An Efficient Concurrent Repetition Theorem

Douglas Wikström
KTH Stockholm

July 13, 2009

Abstract

Håstad et al. (2008) prove, using Raz's lemma (STOC '95) the first efficient parallel repetition theorem for protocols with a non-constant number of rounds, for a natural generalization of public-coin protocols. They show that a parallel prover that convinces a fraction $1-\gamma$ of the embedded verifiers of a k-wise repeated m-message verifier can be turned into a prover with error probability $1-\gamma-O(m\sqrt{-\log(\epsilon)/k})$. This improves previous results of Impagliazzo et al. (Crypto 2007) and Pass and Venkitasubramaniam (STOC 2007) that studies the constant round case.

We prove a generalization of Raz's Lemma to random processes that allows us to improve the analysis of the reduction of Håstad et al. in the public-coin case to $1-\gamma-O(\sqrt{-\log(\epsilon)/k})$, i.e., we remove the dependence on the number rounds completely, and thus the restriction to settings where $k > m^2$.

An important implication of the strengthened parallel repetition theorem is the first efficient *concurrent* repetition theorem for protocols with a non-constant number of rounds. In concurrent repetition, the verifiers execute completely independently and only report their final decision, i.e., the prover chooses arbitrarily in which order it interacts with the individual verifiers. This should be contrasted with parallel repetition where the verifiers are synchronized in each round.

1 Introduction

Arthur-Merlin games [1] and interactive proofs [7] were introduced to allow a prover to convince a verifier that a statement x belongs to a language L, without transferring an explicit witness w of this fact. Such a protocol is said to have soundness $1-\delta$ if the probability that the verifier accepts when $x \notin L$ is at most δ, and δ is then called the error probability. A protocol is said to be complete if the verifier accepts a true statement when both parties follow the protocol. In applications, it is often required that the protocol does not leak any knowledge of the secret information, i.e., that the protocol is zero-knowledge [7], but in this paper we focus on the error probability of protocols.

The concept of interactive proofs has been generalized in numerous ways. Proofs of knowledge [2] allow the prover to show that it knows some secret information. Note that this may be interesting even when the existence of the given secret information is obvious. In computationally sound protocols, e.g., interactive arguments [4], the verifier is only safe against computationally bounded cheating prover, i.e., if a prover violates the soundness property, then some computational assumption is violated.

When the error probability of a protocol is not sufficiently low, one may hope that the error probability is decreased exponentially if the protocol is executed repeatedly and the verifier only accepts if all individual executions are accepting. Two forms of repetition are considered in the literature: sequential repetition and parallel repetition. In sequential repetition, the execution of one instance of the basic protocol is completed before another execution is started. This is known to reduce the error probability exponentially for all interesting models.

For parallel repetition the situation is more complicated. On the one hand, parallel repetition of interactive proofs does reduce the error at an optimal rate, i.e., if the basic protocol has error probability δ, its the k-wise repetition has error probability δ^k. On the other hand, Bellare, Impagliazzo and Naor [3] and Pietrzak and Wikström [13] show that parallel repetition may not reduce the error probability of computationally sound protocols. However, Bellare et al. also establish an efficient parallel repetition theorem for three-message protocols, i.e., an efficient black-box reduction that turns any k-wise parallel prover with error probability ϵ into a single instance prover with error probability $1 - O(\sqrt{-\log(\epsilon)/k})$. Canetti, Halevi and Steiner [5] prove a stronger version of this result where the error probability of the single instance prover is $1 - O(-\log(\epsilon)/k)$, which is optimal. Pass and Venkitasubramaniam [12] generalize this result to any constant-round public-coin protocol (with similar parameters). In a generalization in an other direction Impagliazzo, Jaswal and Kabanets [11] prove, for tree-message protocols, that even a parallel prover that only convinces a certain fraction $1-\gamma$ of the individual verifiers with probability ϵ can be used to construct a single instance prover with error probability roughly $1 - \gamma - O(\sqrt{-\log(\epsilon)/k})$.

Håstad, Pass, Pietrzak, and Wikström [9] proves the first parallel repetition theorem for protocols with super-constant number of rounds m. They provide an efficient black-box reduction that turns a k-wise parallel prover that convinces a fraction $1-\gamma$ of the verifiers with probability ϵ into a single instance prover with error probability $1 - \gamma - O(m\sqrt{-\log(\epsilon)/k})$. The running time of the reduction is polynomial in m, n, and k, and linear in $1/\epsilon$.

In work subsequent to [9], Haitner [8] proves that any interactive argument can be modified slightly such that the error probability decreases exponentially with parallel repetition. His single instance prover requires that $k \geq n^8 m^{12}$ and has much worse parameters: the error probability is $1-\gamma-O(mk^{-\frac{1}{10}})$, and the running time is polynomial in m, n, and k, and cubic in $1/\epsilon$.

In the proofs of all the mentioned efficient parallel repetition theorems the constructed single instance prover simulates internally an interaction between the parallel prover given as a black-box and k verifiers $\mathcal{V}_1, \ldots, \mathcal{V}_k$, except that $\mathcal{V}_j$ for some index j in effect is replaced by the external verifier and the messages of the other verifiers are chosen to maximize the probability that the external verifier accepts. In other words, any message handed to $\mathcal{V}_j$ in the simulated interaction is forwarded to the external verifier, and any message from the external verifier is taken as a message output by $\mathcal{V}_j$.

What differs in the proofs of the mentioned results is: (1) how the index j is chosen, (2) how the messages of the internally simulated verifiers $\mathcal{V}_i$, $i \neq j$, are chosen to maximize the accept probability of the external verifier, and (3) how the constructed single instance prover is analyzed.

In some of the cited works it is proved that a random index j is good on average and in other works a good index j is found by sampling. However, from an analytical point of view these two approaches are usually equivalent.

Both Canetti et al. [5] and Pass and Venkitasubramaniam [12] use extensive sampling in each round to find choices of messages of the simulated verifiers $\mathcal{V}_i$, $i \neq j$, that approximate the optimal computationally unbounded single instance prover. In [12] this leads to a super-exponential running time in the number of rounds, which explains the restriction to constant-round protocols. In both cases, the analysis consists of bounding the difference between the unbounded optimal strategy and its approximation.

Impagliazzo et al. [11], Håstad et al. [9], and Haitner [8] all use a lemma of Raz [14]; in the latter two cases in the generalized form given by Holenstein [10]. Raz's lemma states that if $X_1, \ldots, X_k$ are independently distributed random variables and W and event, then $\frac{1}{k}\sum_{i=1}^{k} \left\| \mathsf{P}_{X_i|W} - \mathsf{P}_{X_i} \right\|$ is bounded by $\sqrt{\log\left(1/\Pr\left[W\right]\right)/k}$. We remark that the first paper on parallel repetition of Bellare et al. [3] seems to embed the proof of a related statement, but intertwined with a "trust halving" strategy.

1.1 Our Contribution

We prove a theorem that generalizes Raz's lemma [14] to certain random processes described below. Our result allows us to strengthen the parallel repetition theorem of Håstad et al. in the public-coin case, including the threshold case, in that we remove the restriction to protocols where the number of repetitions is greater than the square of the number of rounds. Then we show that the strengthened parallel repetition theorem gives the first efficient *concurrent* repetition theorem for public-coin protocols with a non-constant number of rounds.

Our results are easily extended to extendable and simulatable [9] verifiers, but only in the case where the verifiers decision to accept or not can be computed publicly from an interaction, i.e., the decision to accept or not does not depend on any private values.

The Analysis of Håstad et al. Is Too Pessimistic. Let us take a closer look at the strategy of Håstad et al. [9] for the case of public-coin protocols. Consider an interaction between a parallel prover $\mathcal{P}^{(k)}$ and the parallel repetition $\mathcal{V}^k$ of an m-message verifier $\mathcal{V}$. Denote the ith verifier by $\mathcal{V}_i$ and denote its lth message by $C_{l,i}$, and denote the list of the lth messages of all verifiers by $C_l = (C_{l,1}, \ldots, C_{l,k})$. We also write $C_{[l]} = (C_1, \ldots, C_l)$ to denote a partial interaction. Let W be the event that all verifiers accept.

If we choose $C_{[m]}$ randomly conditioned on the event W, then clearly all verifiers accept. Note that we may think of the process of sampling this distribution as proceeding round by round, where in the lth round: (1) $\mathcal{V}_j$ samples its message conditioned on the interaction so far and W, and (2) all other verifiers samples their messages jointly conditioned on the interaction so far (including the message of $\mathcal{V}_j$ in Step (1)) and W.

The reduction of Håstad et al. corresponds to sampling, for a random j, a similar distribution with the following two modifications. Firstly, it may not be feasible to sample messages conditioned on W in a given round given some particular partial interactions. They show that such partial interactions occur with a correspondingly low probability. We now consider the more interesting modification, namely that $\mathcal{V}_j$ no longer conditions the choice of its message on W in Step (1) of the sampling of the lth round. To deal with this they apply, for each round l, Raz's lemma (in Holenstein's form) to conclude that the distribution of $C_{l,j}$ for a randomly chosen j remains approximately the same even without conditioning on W. Then using the triangle inequality of statistical distance, the probability that all accept, and in particular $\mathcal{V}_j$,

in the modified process is $1 - O(m\sqrt{-\log(\epsilon)/k})$, where $\epsilon = \Pr[W]$. It is natural to ask if this bound is tight with respect to m, since the bound is useless unless $k > m^2$.

Let us give some intuition why the bound is not tight. Suppose that the repeated protocol itself consists of the $O(m)$-wise sequential repetition of some other constant-round basic protocol. Then we may just as well view the protocol as the $O(m)$-wise sequential repetition of the k-wise parallel repetition of the basic protocol. Consider now what conditioning on W somewhere in the execution of the lth sequential copy of the parallel repetition of the basic protocol means. It simply means conditioning on the event W_l that the lth sequential copy of the k-wise parallel repetition of the basic protocol accepts, i.e., all the basic verifiers in the lth round accept. This event is on average over l much more likely than the event $W = \wedge_{l=1}^{s} W_l$ that all basic verifiers in each sequential copy accepts, where s is the number of sequential copies. This indicates, that at least in some situations, Håstad et al. are too pessimistic when they in each round use the probability of the event W to bound the effect of conditioning on W.

In the general case there exist no events such as W_l that can be used for conditioning instead of W. The main contribution of this work is the observation that we can mimic the above intuition in the general case using the notion of relative entropy, and improve Håstad et al.'s bound to $1 - O(\sqrt{-\log(\epsilon)/k})$.

An Efficient Concurrent Repetition Theorem. A natural generalization of parallel repetition is *concurrent* repetition. The k-wise concurrent repetition of a verifier $\mathcal{V}$, denoted $\mathcal{V}^{\#k}$, executes k independent verifiers and accepts iff all verifiers accept. In contrast to the parallel repetition, the messages of the individual verifiers are *not* synchronized. Thus, a concurrent prover may *adaptively* choose to delay the further interaction with some verifiers of its choosing until it has interacted some more with other verifiers. In other words, the concurrent prover may *arbitrarily schedule* the interactions with the individual verifiers.

We show that any k-wise concurrent repetition of a m-message verifier $\mathcal{V}$ may be viewed as the k-wise parallel repetition of a related mk-message verifier $\mathcal{V}'$. What makes this observation useful is: (1) that a concurrent prover $\mathcal{P}^{\{k\}}$ with error probability ϵ against $\mathcal{V}^{\#k}$ can be turned into a related parallel prover $\mathcal{P}^{(k)}$ against $\mathcal{V}'^{k}$ with the same error probability, and (2) that any prover $\tilde{\mathcal{P}}'$ with error probability δ against $\mathcal{V}'$ can be turned into a prover $\tilde{\mathcal{P}}$ with the same error probability against $\mathcal{V}$.

Thus, the concurrent setting can be reduced to the parallel setting at the cost of increasing the number of rounds to mk. Recall that the parallel repetition theorems of Håstad et al. [9] and Haitner [8] requires that $k > m^2$ and $k > n^8 m^{12}$ respectively. Thus, these theorems can not be used to argue about a parallel interaction derived from a concurrent interaction as explained above.

Our parallel repetition theorem does not suffer from any such restriction. Thus, we get a *concurrent* repetition theorem for public-coin protocols.

1.2 Organization of Paper

In Section 3 we recall the definition of relative entropy (Kullback-Leibler distance) and derive the elementary properties we need to prove our main theorem. Then we state and prove in Section 4 our generalization of Raz's lemma. In Section 5 we then show how the parallel

repetition theorem of Håstad et al. [9] can be strengthened. Finally, in Section 6 we formalize *concurrent* repetition and extend the strengthened parallel repetition theorem to cover also this type of repetition.

2 Notation

We denote the set $\{1, \ldots, m\}$ by $[m]$. We denote the binary logarithm of x by $\log x$. 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 use the convention that $\mathsf{P}_{Y|XW}(\perp|x) = 1$, where $\perp$ is a special symbol, for all $x \in [X]$ such that $\Pr[W|X = x] = 0$. Similarly, we set $\mathsf{P}_{Y|XW}(\perp|\perp) = 1$. This allows us to consider expressions like $\mathsf{P}_X\mathsf{P}_{Y|XW}$ to be distributions, which is otherwise not necessarily the case.

The algorithmic interpretation of the first convention is a process where we first sample x according to P_X and then if possible sample y according to $\mathsf{P}_{Y|XW}$. If the latter is not possible the we set y equal to the failure symbol $\perp$. The second convention simply says that once we have failed to sample some component, then we fail to sample all remaining components as well.

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

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

2.1 Protocols and Parallel Repetition

For simplicity we consider only interactive proofs and arguments and not general computationally sound protocols. Our results are, however, easily generalized along the lines of [9].

We write $\langle \mathcal{P}, \mathcal{V}\rangle(x)$ to denote the output of the verifier $\mathcal{V}$ when a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ interact on common input x. Without loss we assume that x contains any auxiliary input to $\mathcal{P}$, since we can always replace $\mathcal{V}$ by a verifier that ignores a prefix of x.

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, or a given fraction of, the individual verifiers accept. We are also interested in repeated threshold verifiers, denoted by $\mathcal{V}^k_\gamma$, that accept if at least $(1-\gamma)k$ of the individual verifiers accept. We denote the number of accepting individual verifiers in an interaction between a parallel prover $\mathcal{P}^{(k)}$ and $\mathcal{V}^k$ by $\#\langle \mathcal{P}^{(k)}, \mathcal{V}^k\rangle(x)$.

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. Then 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 an instance x

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

where we think of both the prover and verifier as deterministic algorithms and denote the random tape of $\mathcal{V}_i$ by R_i.

Without loss we assume that there is a dedicated symbol $\perp$ such that: $A_l = (\perp, \ldots, \perp)$ when $C_{l,i} = \perp$ for some i, $C_{l+1,i} = \perp$ when $A_{l,i} = \perp$, and $D_i = 0$ when $A_{m,i} = \perp$. In other words, we assume that if the prover outputs the special symbol, then it propagates through the complete process. The prover uses the special symbol to fail explicitly, if it can not compute a good reply of a message of the verifier.

3 Relative Entropy

We use the notion of relative entropy (Kullback-Leibler distance) in the statement and proof of our main theorem. In this section we recall its definition and state a number of elementary results. For completeness we provide proofs of all results in Appendix A. A good source on relative entropy is [6].

Definition 2 (Relative Entropy)**.** The relative entropy (Kullback-Leibler distance), denoted $D\left(\mathsf{P}_X \,\|\mathsf{P}_Y\right)$, of two distributions P_X and P_Y over a set $\mathcal{X}$ is defined by

$$D\left(\mathsf{P}_X \,\|\mathsf{P}_Y\right) = \sum_{x \in \mathcal{X}} \mathsf{P}_X\left(x\right) \log \frac{\mathsf{P}_X\left(x\right)}{\mathsf{P}_Y\left(x\right)} \ ,$$

where we use the conventions that $0 \log \frac{0}{a} = 0 \log \frac{0}{0} = 0$ and $a \log \frac{a}{0} = \infty$ for $0 < a \leq 1$.

We remark that although the relative entropy is always non-negative, it is not a true distance metric, since it is not symmetric and it does not satisfy the triangle inequality. It is, however, related to the statistical distance by the following lemma (Lemma 11.6.1 in [6]).

Lemma 3 (Bound Statistical Distance)**.** *Let P_X and P_Y be distributions. Then*

$$\left\|\mathsf{P}_X - \mathsf{P}_Y\right\|^2 \leq \frac{\ln 2}{2} D\left(\mathsf{P}_X \,\|\mathsf{P}_Y\right) \ .$$

The chain rule of relative entropy given below plays an important part in the proof of our results. We use this rule in both directions.

Lemma 4 (Chain Rule)**.** *Let $\mathsf{P}_{X_1 X_2}$ and $\mathsf{P}_{Y_1 Y_2}$ be distributions over a set $\mathcal{X}_1 \times \mathcal{X}_2$. Then*

$$D\left(\mathsf{P}_{X_1 X_2} \,\|\mathsf{P}_{Y_1 Y_2}\right) = D\left(\mathsf{P}_{X_1} \,\|\mathsf{P}_{Y_1}\right) + \sum_{x \in \mathcal{X}_1} \mathsf{P}_{X_1}\left(x\right) D\left(\mathsf{P}_{X_2|X_1}\left(\,\cdot\,|x\right) \big\| \mathsf{P}_{Y_2|Y_1}\left(\,\cdot\,|x\right)\right) \ .$$

The sum of the right hand side is sometimes called conditional relative entropy, but we do not use this notion explicitly. The next lemma allows bounding the average relative entropy of marginal distributions and their conditioned counterpart in terms of the relative entropy of the joint distribution and its conditioned counterpart.

Lemma 5 (Splitting)**.** *Let* $\mathsf{P}_X = \prod_{i=1}^k \mathsf{P}_{X_i}$ *be a product distribution and* W *an event. Then*

$$D\left(\mathsf{P}_{X|W} \,\middle\|\, \prod_{i=1}^k \mathsf{P}_{X_i}\right) = \sum_{i=1}^k D\left(\mathsf{P}_{X_i|W} \,\middle\|\, \mathsf{P}_{X_i}\right) + D\left(\mathsf{P}_{X|W} \,\middle\|\, \prod_{i=1}^k \mathsf{P}_{X_i|W}\right) \quad .$$

The next lemma allows us, following Holenstein [10], to bound the relative entropy of a distribution and its counterpart conditioned on an event W by the probability of the event W.

Lemma 6 (Bound Relative Entropy)**.** *Let* P_X *be a distribution and* W *an event. Then*

$$D\left(\mathsf{P}_{X|W} \,\middle\|\, \mathsf{P}_X\right) \leq \log\left(\frac{1}{\Pr[W]}\right) \quad .$$

4 Main Theorem

In this section we state and prove our main result.

Theorem 7 (Main Theorem)**.** *Let* $X_{[m]} = (X_1, \ldots, X_m)$*, with* $X_l = (X_{l,1}, \ldots, X_{l,k})$*, be a random process where in step* l *the components of* X_l *are chosen independently conditioned on the previous steps* $X_{[l-1]} = (X_1, \ldots, X_{l-1})$ *of the process, i.e.,*

$$\mathsf{P}_{X_{[l]}} = \mathsf{P}_{X_{[l-1]}} \prod_{i=1}^k \mathsf{P}_{X_{l,i}|X_{[l-1]}} \quad .$$

Let W *be an event and define for* $j = 1, \ldots, k$ *a modified process* $Y_{[m]}^{(j)} = (Y_1^{(j)}, \ldots, Y_m^{(j)})$ *by*[1]

$$\mathsf{P}_{Y_{[l]}^{(j)}} = \mathsf{P}_{Y_{[l-1]}^{(j)}} \mathsf{P}_{X_{l,j}|X_{[l-1]}} \mathsf{P}_{X_{l,\langle j\rangle}|X_{[l-1]},X_{l,j},W} \quad ,$$

where $X_{l,\langle j\rangle} = (X_{l,1}, \ldots, X_{l,j-1}, X_{l,j+1}, \ldots, X_{l,k})$*, i.e., in each step all except the* j*th component are chosen conditioned on all previous steps and on* W*. Then*

$$\sum_{j=1}^k D\left(\mathsf{P}_{X_{[m]}|W} \,\middle\|\, \mathsf{P}_{Y_{[m]}^{(j)}}\right) = D\left(\mathsf{P}_{X_{[m]}|W} \,\middle\|\, \mathsf{P}_{X_{[m]}}\right) - D\left(\mathsf{P}_{X_{[m]}|W} \,\middle\|\, \prod_{l=1}^m \prod_{i=1}^k \mathsf{P}_{X_{l,i}|X_{[l-1]}W}\right) \quad .$$

[1] Note that this is a well defined distribution due to our notational conventions described in Section 2.

4.1 Bounds In Terms of the Probability of the Event

In applications, one is often interested in bounding the statistical distance, and not the relative entropy, between distributions. Furthermore, the goal is usually to bound the distance in terms of the probability of the event W. Here we derive such corollaries.

Corollary 8. *With the hypothesis of Theorem 7,*

$$\sum_{j=1}^{k} D\left(\mathsf{P}_{X_{[m]}|W} \,\middle\|\, \mathsf{P}_{Y_{[m]}^{(j)}}\right) \leq \log\left(\frac{1}{\Pr[W]}\right) \ .$$

Proof. This follows immediately from Lemma 6 and the non-negativity of relative entropy. □

Corollary 9. *With the hypothesis of Theorem 7,*

$$\frac{1}{k}\sum_{j=1}^{k} \left\| \mathsf{P}_{X_{[m]}|W} - \mathsf{P}_{Y_{[m]}^{(j)}} \right\| \leq \sqrt{\frac{\ln 2}{2k}} \sqrt{\log\left(\frac{1}{\Pr[W]}\right)} \ .$$

This follows straightforwardly from Corollary 8 using the Cauchy-Schwarz inequality and Lemma 3 (see Appendix A for a proof).

4.2 Proof of Main Theorem

Repeated application of the chain rule of relative entropy (Lemma 4) gives

$$D\left(\mathsf{P}_{X_{[m]}|W} \,\middle\|\, \mathsf{P}_{Y_{[m]}^{(j)}}\right) = \sum_{l=1}^{m} \sum_{x \in [X_{[l-1]}]} \mathsf{P}_{X_{[l-1]}|W}(x)\, D_{j,l}(x) \ ,$$

where

$$D_{j,l}(x) = D\left(\mathsf{P}_{X_l|X_{[l-1]}W}(\cdot|x) \,\middle\|\, \mathsf{P}_{X_{l,j}|X_{[l-1]}}(\cdot|x)\, \mathsf{P}_{X_{l,\langle j\rangle}|X_{[l-1]},X_{l,j}W}(\cdot|x,\cdot)\right) \ .$$

Another application of the chain rule gives

$$\begin{aligned} D_{j,l}(x) &= D\left(\mathsf{P}_{X_{l,j}|X_{[l-1]}W}(\cdot|x) \,\middle\|\, \mathsf{P}_{X_{l,j}|X_{[l-1]}}(\cdot|x)\right) \\ &\quad + \sum_{y} \mathsf{P}_{X_{l,j}|X_{[l-1]}W}(y|x)\, D\left(\mathsf{P}_{X_{l,\langle j\rangle}|X_{[l-1]},X_{l,j}W}(\cdot|x,y) \,\middle\|\, \mathsf{P}_{X_{l,\langle j\rangle}|X_{[l-1]},X_{l,j}W}(\cdot|x,y)\right) \\ &= D\left(\mathsf{P}_{X_{l,j}|X_{[l-1]}W}(\cdot|x) \,\middle\|\, \mathsf{P}_{X_{l,j}|X_{[l-1]}}(\cdot|x)\right) \ , \end{aligned}$$

since $D(\mathsf{P}_X \| \mathsf{P}_X) = 0$ for any distribution P_X. We write

$$B = \sum_{j=1}^{k} D\left(\mathsf{P}_{X_{[m]}|W} \,\middle\|\, \mathsf{P}_{Y_{[m]}^{(j)}}\right) = \sum_{l=1}^{m} \sum_{x \in [X_{[l-1]}]} \mathsf{P}_{X_{[l-1]}|W}(x) \sum_{j=1}^{k} D_{j,l}(x) \ ,$$

and use the conditional independence of the $X_{l,j}$ and splitting (Lemma 5) to get

$$\sum_{j=1}^{k} D_{j,l}(x) = D\left(\mathsf{P}_{X_l|X_{[l-1]}W}(\cdot|x) \middle\| \prod_{i=1}^{k} \mathsf{P}_{X_{l,i}|X_{[l-1]}}(\cdot|x)\right) - D\left(\mathsf{P}_{X_l|X_{[l-1]}W}(\cdot|x) \middle\| \prod_{i=1}^{k} \mathsf{P}_{X_{l,i}|X_{[l-1]}W}(\cdot|x)\right) ,$$

which implies

$$B = \sum_{l=1}^{m} \sum_{x\in[X_{[l-1]}]} \mathsf{P}_{X_{[l-1]}|W}(x)\, D\left(\mathsf{P}_{X_l|X_{[l-1]}W}(\cdot|x) \middle\| \mathsf{P}_{X_l|X_{[l-1]}}(\cdot|x)\right) - \sum_{l=1}^{m} \sum_{x\in[X_{[l-1]}]} \mathsf{P}_{X_{[l-1]}|W}(x)\, D\left(\mathsf{P}_{X_l|X_{[l-1]}W}(\cdot|x) \middle\| \prod_{i=1}^{k} \mathsf{P}_{X_{l,i}|X_{[l-1]}W}(\cdot|x)\right) .$$

Repeated application of the chain rule simplifies this quantity to

$$D\left(\mathsf{P}_{X_{[m]}|W} \middle\| \mathsf{P}_{X_{[m]}}\right) - D\left(\mathsf{P}_{X_{[m]}|W} \middle\| \prod_{l=1}^{m}\prod_{i=1}^{k} \mathsf{P}_{X_{l,i}|X_{[l-1]}W}\right) ,$$

which concludes the proof. ■

5 A Sharper Parallel Repetition Theorem

We illustrate the use of Theorem 7 by proving a parallel repetition theorem for public-coin protocols, but our result is easily adapted to he slightly more general case of extendable and simulatable [9] verifiers which allow computing the decision to accept or not without using any private values.

Theorem 10. *Let $\mathcal{V} \in \mathsf{P}$ be public-coin and let $\mathcal{P}^{(k)} \in \mathsf{P}$ be a parallel prover. Then there exists a prover $\tilde{\mathcal{P}}$ running in time $\mathsf{Poly}(n, k, m, 1/\epsilon)$, such that for every instance x where $\Pr[\langle \mathcal{P}^{(k)}, \mathcal{V}_\gamma^k\rangle(x) = 1] \geq \epsilon$, for some $0 \leq \gamma < 1$, we have*

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

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 lower bound of [9] is $1 - (m+1)\sqrt{-\log(\epsilon)/k}$. Thus, perhaps surprisingly, we are able to remove the dependence on m in the error term entirely. The running time on the other hand still depends linearly on m. In Section 6 we use the new bound to generalize the theorem to *concurrent* repetition.

5.1 Proof of Theorem 10

There is no need to modify the reduction of [9]; we only perform a tighter analysis.

The Reduction. Recall the reduction of [9]. The constructed single instance prover $\tilde{\mathcal{P}}_u$ simulates an interaction with the parallel prover $\mathcal{P}^{(k)}$ and interacts with a single external verifier by plugging in the external verifier at the jth "slot" of the interaction, where j is chosen randomly in $[k]$. More precisely, for $l = 1, \ldots, m$, $\tilde{\mathcal{P}}_u$ waits for the lth message from the external verifier and takes this to be the message from its internal jth simulated verifier. Then it tries to choose the lth messages of all the other simulated verifiers conditioned on the event that a completion of the current partial interaction makes all the verifiers accept. This is done by sampling a completion of a partial interaction until one is found where at least $(1-\gamma)k$ accept, or until $u = m\sqrt{k}/\epsilon$ attempts have been done. In the latter case, $\tilde{\mathcal{P}}_u$ simply gives up. Below we write $\mathsf{Complete}(c_{[l]}, a_{[l]})$ to denote the output of the completion procedure starting from a partial interaction $(c_{[l]}, a_{[l]})$, i.e., the output of the completion procedure is the list of decisions of the individual verifiers. We write $\#(\cdot)$ to count the number of ones in such a list of decisions. Below we recall the details of this reduction from [9].

Algorithm 11. $\tilde{\mathcal{P}}_{u,\gamma}(x)$

if x is an instance **then**
  $(s_0, a_0) \leftarrow \mathcal{P}^{(k)}(x)$ // *Compute prover's first message*
  $j \leftarrow_R [k]$ // *Choose random index*
  return $\left(\left[j, s_0, \emptyset, a_{[0]}\right], a_{0,j}\right)$ // *Output state and first message*
else
  Interpret x as $\left(\left[j, s_{l-1}, c_{[l-1]}, a_{[l-1]}\right], c_{l,j}\right)$ // *Read state and verifier's message*
  for $v = 1, \ldots, u$ **do**
    $c_{l,\langle j\rangle} \leftarrow_R \{0,1\}^{p(n)\times(k-1)}$ // *Sample verifiers' messages*
    $(s_l, a_l) \leftarrow \mathcal{P}^{(k)}(s_{l-1}, c_l)$ // *Compute prover's reply*
    if $\#(\mathsf{Complete}(c_{[l]}, a_{[l]})) \geq (1-\gamma)k$ **then** // *If messages are good,*
      return $\left(\left[j, s_l, c_{[l]}, a_{[l]}\right], a_{l,j}\right)$ // *then output reply*
    done
  done
  $c_{l,\langle j\rangle} \leftarrow (\perp, \ldots, \perp)$ // *Give up if no good messages are found*
  $(s_l, a_l) \leftarrow \mathcal{P}^{(k)}(s_{l-1}, c_l)$
  return $\left(\left[j, s_l, c_{[l]}, a_{[l]}\right], a_{l,j}\right)$
end

Note that if the for-loop does not return, then the special symbol $\perp$ is propagated through the rest of the process and all verifiers reject. The prover could of course pick any message instead of explicitly admitting failure, but it is convenient in the analysis below that a failure in the for-loop is propagated.

Our Analysis. In the analysis we consider the common input x to be fixed. Since the prover is deterministic, this means that an interaction is completely determined by the messages of the verifier. We consider three processes defined below. In all three processes J denotes the same random variable uniformly distributed in $\{1, \ldots, k\}$.

- Real Process. The real process is generated by an interaction between $\tilde{\mathcal{P}}$ and $\mathcal{V}$. We denote the lth message of $\mathcal{V}_i$ in the real process by $\tilde{C}_{l,i}$. Here $\mathcal{V}_i$ denotes either an internally simulated verifier, or the external real verifier $\mathcal{V}$ depending on if $i = j$ or not.

The list of the lth messages of all verifiers in such an interaction is denoted by $\tilde{C}_l$, and we use $\tilde{C}_{[l]}$ to denote the list $(\tilde{C}_1, \ldots, \tilde{C}_l)$. We use $\tilde{D}_i$ to denote the decision of $\mathcal{V}_i$.

- Idealized Real Process. The idealized real process is identical to the real process except $\tilde{\mathcal{P}}$ is replaced by $\tilde{\mathcal{P}}'$, which is identical to $\tilde{\mathcal{P}}$ except for the following modification. Before entering the for-loop, $\tilde{\mathcal{P}}'$ checks if the probability (over the randomness of the for-loop) that the for-loop returns is positive. If not, then it skips the for-loop and completes the simulation of $\tilde{\mathcal{P}}$. Otherwise, it samples a random execution of the for-loop conditioned on it returning a tuple, i.e., conceptually it sets $u = \infty$ and executes the for-loop. We denote the random variables of this idealized process by adding a prime symbol to the corresponding random variables of the real process, i.e., we write $\tilde{C}'_{l,i}$, $\tilde{C}'_l$, $\tilde{C}'_{[l]}$, and $\tilde{D}'_i$.
- Ideal Process. The ideal process is identical to the idealized real process except that the external verifier in each round chooses its message conditioned on the interaction so far and that at least $(1-\gamma)k$ verifiers accept. We denote the random variables of this process by removing the tilde from the corresponding random variables of the idealized real process, i.e., we write $C'_{l,i}$, C'_l, $C'_{[l]}$, and D'_i.

In the ideal process the oracle can always find good messages, but the ability of the oracle to output a failure symbol is needed in the idealized real process.

Conceptual Modifications. It is convenient to introduce some conceptual modifications. These modifications do change the joint distribution of the three processes considered together, but since we only consider one process at a time this is not a problem. We also define indicator variables for the event that at most u samples are needed in the for-loop of the respective processes.

Let $\mathsf{SampleRound}$ be the probabilistic function which takes as input a tuple $(j, \tilde{c}_{[l-1]}, \tilde{c}_{l,j})$ and simulates $\tilde{\mathcal{P}}'$ of the idealized real process using these verifier messages. When $\tilde{\mathcal{P}}'$ returns, it returns $(\tilde{z}_l, \tilde{c}_{[l]})$, where $\tilde{z}_l$ is one if $\tilde{\mathcal{P}}'$ needed at most u of the for-loop to return and zero otherwise, and $\tilde{c}_{[l]}$ is extracted from the prover state $\left[j, s_l, c_{[l]}, a_{[l]}\right]$ output by $\tilde{\mathcal{P}}'$. Recall that the instance x is fixed and that the verifier messages then determine the prover messages since $\mathcal{P}^{(k)}$ is deterministic, so all the values needed to simulate $\tilde{\mathcal{P}}'$ are determined, and $\mathsf{SampleRound}$ is well-defined. We now use $\mathsf{SampleRound}$ to re-interpret the idealized real process and the real process.

Let $\tilde{Z}'_l$ be the indicator variable of the event that at most u samples are needed in the for-loop of $\tilde{\mathcal{P}}'$ in round l of the idealized real process. Then we may sample $((\tilde{Z}'_1, \ldots, \tilde{Z}'_m), \tilde{C}'_{[m]})$ by first generating an index j and a sample $\tilde{c}'_{[m]}$ of the idealized real process conditioned on this index, and then sample the indicator variables by setting $(\tilde{z}'_l, \cdot) = \mathsf{SampleRound}(j, \tilde{c}'_{[l-1]}, \tilde{c}'_{l,j})$, where we use $\cdot$ to indicate that we ignore the second output. Thus, we may define another probabilistic function $\mathsf{Indicators}$ which calls $\mathsf{SampleRound}$ internally, and simply write

$$(\tilde{Z}'_1, \ldots, \tilde{Z}'_m) = \mathsf{Indicators}(J, \tilde{C}'_{[m]}) \ . \tag{1}$$

Similarly, if we let Z'_l be the indicator variable of the event that at most u samples are needed

in the for-loop of $\tilde{\mathcal{P}}'$ in round l in the ideal process, then we may write

$$(Z'_1, \ldots, Z'_m) = \mathsf{Indicators}(J, C'_{[m]}) \ . \tag{2}$$

We now argue that also the real process can be interpreted in a similar way. Let $\tilde{Z}_l$ be the indicator variable of the event that at most u samples are needed in the for-loop of $\tilde{\mathcal{P}}$ in round l of the real process. Furthermore, let Real denote the deterministic function that takes a tuple $(j, (\tilde{z}'_1, \ldots, \tilde{z}'_m), \tilde{c}'_{[m]})$ as input and returns $((\tilde{z}_1, \ldots, \tilde{z}_m), \tilde{c}_{[m]})$, where

1. $\tilde{z}_l = \tilde{z}'_l$, and
2. $(\tilde{c}_{[l]}, \tilde{c}_{l,j}) = (\tilde{c}'_{[l]}, \tilde{c}'_{l,j})$ and $(\tilde{c}_{l,\langle j \rangle}, \tilde{c}_{l+1}, \ldots, \tilde{c}_m) = (\perp^{k-1}, \perp^{(m-l)\times k})$, where l is the smallest index such that $\tilde{z}'_l = 0$ or $m+1$ if no such index exist.

Then we may write

$$((\tilde{Z}_1, \ldots, \tilde{Z}_m), \tilde{C}_{[m]}) = \mathsf{Real}(J, \mathsf{Indicators}(J, \tilde{C}'_{[m]}), \tilde{C}'_{[m]}) \ .$$

Define $\tilde{Z} = \wedge_{l=1}^m \tilde{Z}_l$ and $\tilde{Z}' = \wedge_{l=1}^m \tilde{Z}'_l$. Then, by definition of $\mathsf{Indicators}$ and Real,

$$\Pr[\tilde{Z} = 1] = \Pr[\tilde{Z}' = 1] \text{ and} \tag{3}$$

$$\mathsf{P}_{\tilde{C}_{[m]}|\tilde{Z}}(\cdot\,|1) = \mathsf{P}_{\tilde{C}'_{[m]}|\tilde{Z}'}(\cdot\,|1) \ . \tag{4}$$

Sampling In the Ideal Process. We let $Z' = \wedge_{l=1}^m Z'_l$ and use the lemma below, taken from [9], to analyze the probability that $Z' = 1$. For completeness, the proof of the lemma is given in Appendix A.

Lemma 12. *Let Y be a random variable and $X_0, X_1, X_2, \ldots$ be identically distributed binary random variables which are only dependent through Y, i.e. $\mathsf{P}_{YX_0\ldots X_j} = \mathsf{P}_Y \prod_{i=0}^j \mathsf{P}_{X_i|Y}$ 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 $\mathrm{E}[J\,|X_0 = 1] \leq \frac{1}{\Pr[X_0=1]}$.*

We apply the lemma to round l in the ideal process with Y equal to $(C'_{[l-1]}, C'_{l,j})$ and X_i equal to one iff the output of $\mathsf{Complete}$ in the for-loop of $\tilde{\mathcal{P}}'$ in the ith sampling in the lth round contains at least $(1-\gamma)k$ ones. Then the lemma and Markov's inequality implies that if we set $u = 2m\sqrt{k}/\epsilon$, then $\tilde{\mathcal{P}}$ fails to find good messages in the lth round with probability at most $1/(2m\sqrt{k})$. The union bound then implies that

$$\Pr[Z' = 0] \leq \frac{1}{2\sqrt{k}} \ . \tag{5}$$

Relating the Idealized Real Process and the Ideal Process. Recall that $C_{l,i}$ is the lth message of $\mathcal{V}_i$ in an interaction between $\mathcal{P}^{(k)}$ and $\mathcal{V}^k$. We apply Theorem 7 with $X_{l,i} = C_{l,i}$ and W equal to the event that at least $(1-\gamma)k$ verifiers accept. Thus, $X_{[m]}$ conditioned on W

is identically distributed to $C'_{[m]}$, and the modified process $Y^{(j)}_{[m]}$ in the theorem is identically distributed to $\tilde{C}'_{[m]}$ conditioned on $J = j$. In other words,

$$\mathsf{P}_{X_{[m]}|W} = \mathsf{P}_{C'_{[m]}} \text{ and } Y^{(j)}_{[m]} = \mathsf{P}_{\tilde{C}'_{[m]}|J}(\cdot|j) \ ,$$

and the theorem then implies that

$$\left\| \mathsf{P}_{C'_{[m]}} - \mathsf{P}_{\tilde{C}'_{[m]}} \right\| \leq \sqrt{-\log(\epsilon)/k} \ , \tag{6}$$

since J is uniformly distributed.

Concluding the Proof. Recall that the decision of $\mathcal{V}_i$ is a deterministic function of its interaction. Thus, we may define a deterministic function $d_{(\cdot)}(\cdot)$ and write $\tilde{D}_J = d_J(\tilde{C}_{[m]})$, $\tilde{D}'_J = d_J(\tilde{C}'_{[m]})$, and $D'_J = d_J(C'_{[m]})$. We have, with $\omega = \sqrt{-\log(\epsilon)/k}$,

$$\begin{aligned}
\Pr[\tilde{D}_J = 1] &= \Pr[\tilde{D}_J = 1 \wedge \tilde{Z} = 1] && \text{(explicit failures using } \perp) \\
&\geq \Pr[\tilde{D}_J = 1 \mid \tilde{Z} = 1] - \Pr[\tilde{Z} = 0] \\
&= \Pr[\tilde{D}'_J = 1 \mid \tilde{Z}' = 1] - \Pr[\tilde{Z}' = 0] && \text{(from (3) and (4))} \\
&= \Pr[\tilde{D}'_J = 1] - \Pr[\tilde{Z}' = 0] && \text{(from (1))} \\
&\geq \Pr[D'_J = 1] - \omega - (\Pr[Z' = 0] + \omega) && \text{(from (2) and (6))} \\
&\geq 1 - \gamma - 2\omega - \frac{1}{\sqrt{k}} \ , && \text{(from (5))}
\end{aligned}$$

where we use the fact that $\Pr[D'_J = 1] \geq 1 - \gamma$. This concludes the proof. ■

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.

More precisely, the k-wise concurrent repetition of a verifier $\mathcal{V}$, denoted $\mathcal{V}^{\#k}$, executes k independent copies of $\mathcal{V}$ and accepts iff each individual verifier accepts. In each round, the concurrent prover sends messages only to a subset of the verifiers, and as a result only these verifiers send a message back to the prover. This generalization fits in our formalization of a parallel execution between $\mathcal{P}^{(k)}$ and $\mathcal{V}^k$ if we use the convention that $T_{l+1,i} = T_{l,i}$ and $C_{l+1,i} = \emptyset$, whenever $A_{l,i} = \emptyset$. In other words, $A_{l,i} = \emptyset$ means that the concurrent prover did not send $\mathcal{V}_i$ anything as part of its lth message, in which case the state of $\mathcal{V}_i$ is unchanged and it does not send anything back.

We provide a general reduction of the concurrent repetition setting to the parallel repetition setting, which allows us to fall back on our parallel repetition theorem.

Theorem 13. *For every instance x, m-message verifier $\mathcal{V} \in \mathsf{P}$, and $k > 0$ there exists an mk-message verifier $\mathcal{V}' \in \mathsf{P}$ with the following properties.*

1. *For every concurrent prover* $\mathcal{P}^{\{k\}} \in \mathsf{P}$ *there exists a parallel prover* $\mathcal{P}^{(k)} \in \mathsf{P}$ *such that* $\#\langle \mathcal{P}^{(k)}, (\mathcal{V}')^k \rangle(x)$ *and* $\#\langle \mathcal{P}^{\{k\}}, \mathcal{V}^{\#k} \rangle(x)$ *are identically distributed.*

2. *For every prover* $\tilde{\mathcal{P}}' \in \mathsf{P}$ *there exists prover* $\tilde{\mathcal{P}} \in \mathsf{P}$ *such that*

$$\Pr\left[\langle \tilde{\mathcal{P}}, \mathcal{V} \rangle(x) = 1\right] = \Pr\left[\langle \tilde{\mathcal{P}}', \mathcal{V}' \rangle(x) = 1\right] \quad .$$

Furthermore, if $\mathcal{V}$ *is public-coin, then* $\mathcal{V}'$ *is public-coin.*

To see how this theorem can be used, consider a concurrent prover $\mathcal{P}^{\{k\}}$ with error probability probability ϵ against the concurrent repetition of a public-coin verifier $\mathcal{V}$. The first claim of the theorem says in particular that there is a related verifier $\mathcal{V}'$ such that $\mathcal{P}^{\{k\}}$ can be turned into a parallel prover $\mathcal{P}^{(k)}$ with error probability ϵ against the parallel repetition of $\mathcal{V}'$. Theorem 10 says that $\mathcal{P}^{(k)}$ can be turned into a single instance prover $\tilde{\mathcal{P}}'$ with error probability $1 - 2\sqrt{-\log(\epsilon)/k} - \sqrt{1/k}$ against $\mathcal{V}'$. Finally, the second claim of the theorem let us convert $\tilde{\mathcal{P}}'$ into a prover $\tilde{\mathcal{P}}$ with the same error probability against $\mathcal{V}$. The threshold case is similar.

Interestingly, the parallel repetition theorem in [9] for protocols with non-constant number of rounds can not be used as illustrated above. The problem is that it only guarantees an error probability of $1 - O(m'\sqrt{-\log(\epsilon)/k})$, where m' is the number of messages sent by $\mathcal{V}'$, and we have $m' = mk$, which would make the resulting statement vacuously true. In [8] the parameters are even worse. We remark that the optimal result of Pass and Venkitasubramaniam [12] for *constant* round public-coin protocols seems to allow a direct generalization to concurrent repetition, despite that it can not be applied as above.

6.1 Proof of Theorem 13

The verifier $\mathcal{V}'$ simulates $\mathcal{V}$ internally. When handed a message *none*, it checks if it has already received mk messages. If not, then it returns *none*. If so, and if $\mathcal{V}$ has previously output a decision, then $\mathcal{V}'$ outputs this decision and otherwise it outputs 0. When handed a message different from *none*, $\mathcal{V}'$ forwards it to its internal copy of $\mathcal{V}$ and forwards the response produced by $\mathcal{V}$ to the external prover. When $\mathcal{V}$ produces an output decision, $\mathcal{V}'$ checks if it has received mk messages in total. If so, then it outputs the decision of $\mathcal{V}$ and otherwise it stores the decision and sends *none* to the external prover. Note that $\mathcal{V}'$ sends exactly mk messages.

The parallel prover $\mathcal{P}^{(k)}$ simulates the concurrent prover $\mathcal{P}^{\{k\}}$ internally. When given the lth messages C_l from the external verifiers, $\mathcal{P}^{(k)}$ first checks if it has received less than mk lists of messages and if all components of C_l equal *none*. If so, then it sets $A_{l+1,i} = \textit{none}$ for $i = 1, \ldots, k$ and returns A_{l+1}. If not, then any component $C_{l,i}$ equal to $\emptyset$ is set to *none* before forwarding C_l to $\mathcal{P}^{\{k\}}$. When the concurrent prover produces its response A_{l+1}, any component $A_{l+1,i}$ equal to $\emptyset$ is set to *none* before A_{l+1} is handed to the external verifiers by $\mathcal{P}^{(k)}$. Note that the constructed parallel prover accepts exactly mk messages.

Consider now a prover $\tilde{\mathcal{P}}'$ expecting to interact with $\mathcal{V}'$. The prover $\tilde{\mathcal{P}}$ simulates $\tilde{\mathcal{P}}'$ and $\mathcal{V}'$ internally except that every call to $\mathcal{V}$ made internally by $\mathcal{V}'$ is forwarded to the external verifier and its response is taken by $\mathcal{V}'$ as the response of $\mathcal{V}$.

Both claims now follow by inspection. ■

7 Acknowledgments

We thank Johan Håstad and Rafael Pass for helpful discussions.

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 and O. Goldreich. On defining proofs of knowledge. In *Advances in Cryptology – Crypto '92*, volume 740 of *Lecture Notes in Computer Science*, pages 390–420. Springer Verlag, 1992.

[3] 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.

[4] G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[5] 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.

[6] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2nd edition, 2005.

[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. Electronic Colloquium on Computational Complexity Report TR09-027, 2009.

[9] J. Håstad, R. Pass, Pietrzak, and D. Wikström. An efficient parallel repetition theorem. in submission.

[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 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.

[13] 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.

[14] R. Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

A Omitted Proofs

We need two additional elementary lemmas to restate the proof of Lemma 3 taken from [6]. Since these lemmas are not needed in the body of the paper we only state them here.

Lemma 14 (Dataprocessing Inequality For Relative Entropy)**.** *Let* P_X *and* P_Y *be distributions over a set* $\mathcal{X}$ *and let* $\phi : \mathcal{X} \to \{0,1\}$ *be a predicate. Then* $D\left(\mathsf{P}_X \| \mathsf{P}_Y\right) \leq D\left(\mathsf{P}_{\phi(X)} \middle\| \mathsf{P}_{\phi(Y)}\right)$.

Proof. The chain rule of relative entropy implies

$$\begin{aligned} D\left(\mathsf{P}_X\mathsf{P}_{\phi(X)} \middle\| \mathsf{P}_Y\mathsf{P}_{\phi(Y)}\right) &= D\left(\mathsf{P}_X \| \mathsf{P}_Y\right) + \sum_x \mathsf{P}_X(x)\, D\left(\mathsf{P}_{\phi(X)|X}\ (\cdot|x) \middle\| \mathsf{P}_{\phi(Y)|Y}\ (\cdot|x)\right) \\ &= D\left(\mathsf{P}_X \| \mathsf{P}_Y\right) \ , \text{ and} \\ D\left(\mathsf{P}_X\mathsf{P}_{\phi(X)} \middle\| \mathsf{P}_Y\mathsf{P}_{\phi(Y)}\right) &= D\left(\mathsf{P}_{\phi(X)} \middle\| \mathsf{P}_{\phi(Y)}\right) \\ &\quad + \sum_x \mathsf{P}_{\phi(X)}(x)\, D\left(\mathsf{P}_{X|\phi(X)}\ (\cdot|x) \middle\| \mathsf{P}_{Y|\phi(Y)}\ (\cdot|x)\right) \ , \end{aligned}$$

which gives the claim, since relative entropy is non-negative. □

Lemma 15. *Let* P_X *and* P_Y *be distributions over a set* $\mathcal{X}$ *and let* $\Phi = \{x : \mathsf{P}_X(x) > \mathsf{P}_Y(x)\}$. *Then* $\|\mathsf{P}_X - \mathsf{P}_Y\| = \mathsf{P}_X(\Phi) - \mathsf{P}_Y(\Phi)$.

Proof. We have

$$\begin{aligned} \|\mathsf{P}_X - \mathsf{P}_Y\| &= \frac{1}{2}\sum_{x\in\Phi}(\mathsf{P}_X(x) - \mathsf{P}_Y(x)) + \frac{1}{2}\sum_{x\notin\Phi}(\mathsf{P}_Y(x) - \mathsf{P}_X(x)) \\ &= \frac{1}{2}(\mathsf{P}_X(\Phi) - \mathsf{P}_Y(\Phi)) + \frac{1}{2}(\mathsf{P}_Y(\mathcal{X}-\Phi) - \mathsf{P}_X(\mathcal{X}-\Phi)) \\ &= \mathsf{P}_X(\Phi) - \mathsf{P}_Y(\Phi) \ . \end{aligned}$$

□

Proof of Lemma 3. This proof taken from the proof of Lemma 11.6.1 in [6]. Suppose first that P_X and P_Y are binary distributions and write $\mathsf{P}_X(1) = p$ and $\mathsf{P}_Y(1) = q$, where $p \geq q$. We show that

$$p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q} \geq \frac{4}{2\ln 2}(p-q)^2 \ .$$

We denote the difference between the left and right sides by $g(p,q)$. Then

$$\begin{aligned} \frac{\partial g(p,q)}{\partial q} &= -\frac{p}{q\ln 2} + \frac{1-p}{(1-q)\ln 2} - \frac{4}{2\ln 2}2(q-p) \\ &= \frac{q-p}{q(1-p)\ln 2} - \frac{4}{\ln 2}(q-p) \leq 0 \ , \end{aligned}$$

since $q(1-q) \leq \frac{1}{4}$ and $q \leq p$. For $q = p$, $g(p,q) = 0$, and hence $g(p,q) \geq 0$ for $q \leq p$, which proves the lemma for the binary case.

To prove the general case we define a set A by

$$\Phi = \{x : \mathsf{P}_X(x) > \mathsf{P}_Y(x)\},$$

and let ϕ be the characteristic function of Φ, i.e., $\phi(x) \Longleftrightarrow x \in \Phi$. Then by Lemma 14, the binary case, and Lemma 15 we have

$$D(\mathsf{P}_X \| \mathsf{P}_Y) \geq D\left(\mathsf{P}_{\phi(X)} \middle\| \mathsf{P}_{\phi(Y)}\right) \geq \frac{4}{2\ln 2}(\mathsf{P}_X(\Phi) - \mathsf{P}_Y(\Phi))^2 = \frac{2}{\ln 2}\|\mathsf{P}_X - \mathsf{P}_Y\|^2 \quad ,$$

which concludes the proof. □

Proof of Lemma 4. We have

$$\begin{aligned}
&D(\mathsf{P}_{X_1X_2} \| \mathsf{P}_{Y_1Y_2}) \\
&= \sum_{x_1,x_2} \mathsf{P}_{X_1X_2}(x_1,x_2) \log\left(\frac{\mathsf{P}_{X_1}(x_1)}{\mathsf{P}_{Y_1}(x_1)} \cdot \frac{\mathsf{P}_{X_2|X_1}(x_2|x_1)}{\mathsf{P}_{Y_2|Y_1}(x_2|x_1)}\right) \\
&= \sum_{x_1,x_2} \mathsf{P}_{X_1X_2}(x_1,x_2) \log\left(\frac{\mathsf{P}_{X_1}(x_1)}{\mathsf{P}_{Y_1}(x_1)}\right) + \sum_{x_1,x_2} \mathsf{P}_{X_1X_2}(x_1,x_2) \log\left(\frac{\mathsf{P}_{X_2|X_1}(x_2|x_1)}{\mathsf{P}_{Y_2|Y_1}(x_2|x_1)}\right) \\
&= \sum_{x_1} \mathsf{P}_{X_1}(x_1) \log\left(\frac{\mathsf{P}_{X_1}(x_1)}{\mathsf{P}_{Y_1}(x_1)}\right) + \sum_{x_1} \mathsf{P}_{X_1}(x_1) \sum_{x_2} \mathsf{P}_{X_2|X_1}(x_2|x_1) \log\left(\frac{\mathsf{P}_{X_2|X_1}(x_2|x_1)}{\mathsf{P}_{Y_2|Y_1}(x_2|x_1)}\right) \\
&= D(\mathsf{P}_{X_1} \| \mathsf{P}_{Y_1}) + \sum_{x_1} \mathsf{P}_{X_1}(x_1)\, D\left(\mathsf{P}_{X_2|X_1}(\cdot|x_1) \middle\| \mathsf{P}_{Y_2|Y_1}(\cdot|x_1)\right) \quad .
\end{aligned}$$

□

Proof of Lemma 5. We have

$$\begin{aligned}
&D\left(\mathsf{P}_{X|W} \middle\| \prod_{i=1}^k \mathsf{P}_{X_i}\right) \\
&= \sum_x \mathsf{P}_{X|W}(x) \log\left(\frac{\mathsf{P}_{X|W}(x)}{\prod_{i=1}^k \mathsf{P}_{X_i}(x_i)}\right) \\
&= \sum_x \mathsf{P}_{X|W}(x) \log\left(\frac{\prod_{i=1}^k \mathsf{P}_{X_i|W}(x_i)}{\prod_{i=1}^k \mathsf{P}_{X_i}(x_i)}\right) + \sum_x \mathsf{P}_{X|W}(x) \log\left(\frac{\mathsf{P}_{X|W}(x)}{\prod_{i=1}^k \mathsf{P}_{X_i|W}(x_i)}\right) \\
&= \sum_{i=1}^k \sum_x \mathsf{P}_{X|W}(x) \log\left(\frac{\mathsf{P}_{X_i|W}(x_i)}{\mathsf{P}_{X_i}(x_i)}\right) + \sum_x \mathsf{P}_{X|W}(x) \log\left(\frac{\mathsf{P}_{X|W}(x)}{\prod_{i=1}^k \mathsf{P}_{X_i|W}(x_i)}\right) \\
&= \sum_{i=1}^k D\left(\mathsf{P}_{X_i|W} \middle\| \mathsf{P}_{X_i}\right) + D\left(\mathsf{P}_{X|W} \middle\| \prod_{i=1}^k \mathsf{P}_{X_i|W}\right) \quad .
\end{aligned}$$

□

Proof of Lemma 6. We have

$$\begin{aligned} D\left(\mathsf{P}_{X|W} \,\middle\|\, \mathsf{P}_X\right) &= \sum_x \mathsf{P}_{X|W}(x) \log\left(\frac{\mathsf{P}_{X|W}(x)}{\mathsf{P}_X(x)}\right) \\ &= \sum_x \mathsf{P}_{X|W}(x) \log\left(\frac{\Pr[W|X=x]}{\Pr[W]}\right) \\ &= \log\left(\frac{1}{\Pr[W]}\right) + \sum_x \mathsf{P}_{X|W}(x) \log\left(\Pr[W|X=x]\right) \ . \end{aligned}$$

□

Proof of Corollary 9. For any distributions $\mathsf{P}_X, \mathsf{P}_{Y_1}, \ldots, \mathsf{P}_{Y_k}$ we have

$$\left(\sum_{i=1}^k \|\mathsf{P}_X - \mathsf{P}_{Y_i}\|\right)^2 \leq k \sum_{i=1}^k \|\mathsf{P}_X - \mathsf{P}_{Y_i}\|^2 \leq \frac{\ln 2}{2} k \sum_{i=1}^k D\left(\mathsf{P}_X \,\|\mathsf{P}_{Y_i}\right) \ ,$$

where the first inequality is an instance of the Cauchy-Schwarz inequality, and the second is an application of Lemma 3. □

Proof of Lemma 12. This proof is taken from [9]. We can consider only values y of 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. □