Efficient Parallel Repetition Theorems with Applications to Security Amplification

A dissertation presented

by

Kai-Min Chung

to

The School of Engineering and Applied Sciences

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Computer Science

Harvard University

Cambridge, Massachusetts

March 2011

Thesis advisor
Salil P. Vadhan

Author
Kai-Min Chung

Efficient Parallel Repetition Theorems with Applications to Security Amplification

Abstract

This thesis focuses on establishing efficient parallel repetition theorems for computationally sound protocols, which assert that under parallel repetition, the computational soundness error of interactive protocols decreases at an exponential rate, and ideally, behaves as if the repetitions are completely independent. For example, suppose a protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ, then its n-fold parallel repetition $\langle \mathsf{P}^n, \mathsf{V}^n \rangle$, where V^n (called direct-product verifier) accepts iff all n subverifiers accept, should have soundness error δ^n.

The soundness error captures the probability of breaking a cryptographic protocol and/or the probability of convincing a party of a false assertion. Parallel repetition is a simple and desirable way to amplify soundness since it preserves the round complexity. However, existing negative examples show that this does not hold for all interactive protocols. Therefore, the question is, for what classes of protocols do parallel repetition theorems hold?

We prove new parallel repetition theorems for several classes of protocols such as public-coin protocols, three-message protocols, and a more general class of "simulatable" protocols. For some settings such as public-coin protocols with direct product verifiers, we obtain tight results that match information-theoretic bounds. In addition, we will discuss strength and limitations of different reduction ideas. We hope that the discussion can make the current progress more transparent, and lead to better understanding of parallel repetition.

The reductions used for proving parallel repetition theorems have several applications, in particular, to security amplification. We will also present our work on improving the efficient of security amplification for cryptographic primitives such as commitment schemes, signature schemes, message authentication codes, CAPTCHAs, etc.

Contents

Citations to Previously Published Work

The research presented in the thesis is extended from two conference papers [3, 4].

The paper "Parallel Repetition Theorems for Interactive Arguments" [3] published in TCC 2010 consists of parallel repetition theorems for public-coin protocols and computationally simulatable protocols presented in Chapter 3 and 4, except for Section 3.3 and 4.2.

The paper "Efficient String-Commitment from Weak Bit-Commitment" published in Asiacrypt 2010 consists of parallel repetition theorems for three-message protocols presented in Section 3.3 and 4.2 and security amplification applications presented in Chapter 5.

Acknowledgments

First of all, I would like to deeply thank my advisor, Salil Vadhan, for his invaluable guidance throughout the past five years that leads me toward being an independent researcher. He always encourages me to pursue my own research interests and directions. The interaction with Salil are vivid demonstration of his way of doing research, presenting research, thinking about research, as well as his passionate on research, which greatly fostered me in the past, and will continue to guide me in the future — this thesis could not have been done without his insights throughout every stage of the work.

For several years, I have had the good fortune to work with Feng-Hao Liu, who collaborates with me on all work in the thesis and many other projects in cryptography and shares firm friendship with me. I also owe special thanks to Yael Tauman Kalai, who is not only my committee member, but also my collaborator for research on delegating computation. I have benefited a great amount from her creativity and passionate in the fascinating field of cryptography. I would also like to thanks to all my other collaborators: Sherman S.M. Chow, Zhenming Liu, Chi-Jen Lu, Michael Mitzenmacher, Omer Reingold, Bo-Yin Yang, Ching-Hua Yu, and Colin Jia Zheng. Research is always more fun when discussing with people, and it has been a pleasure to work with them in during my Ph.D. period. In particular, the numerous interesting discussions with Zhenming Liu and Colin Jia Zheng in our office are precious memories to me. I want to say yet another special thanks to Chuna-Heng Hsiao, who helped me presenting my research by listening my practice talks with endless patience.

I would also like to express my gratitude to Chih-Jen Lin and Hsueh-I Lu, both of who were my undergraduate advisors at National Taiwan University and guided my very first step to do research.

Of course, there are many more people who have contributed to the development and education of my Ph.D., including, and certainly not limited to, my thesis committee Salil Vadhan, Yael Tauman Kalai, Michael Mitzenmacher, and Les Valiant, many teachers and academic staffs at Harvard, my officemates Yan-Cheng Chang, Eleni Drinea, Alex Healy Stephan Holzer, Shaili Jain, Adam Kirsch, Varun Kanade, Zhenming Liu, Loizos Michael, Shien Jin Ong, Jonathan Pines, Justin Thaler, Jonathan Ullman, and Colin Jia Zheng in Maxwell Dworkin, Brendan Juba at MIT, Wei-Chun Kao, Grant Schoenebeck, Alexandre Stauffer, and Madhur Tulsiani during the visit at Berkeley, Huijia Lin, Rafael Pass,and Wei-lung Dustin Tseng during the time at Cornell, and all my Taiwanese friends at Harvard and other places (too many to list). Thank you to all!

The final thank is reserved to my parents Yao-Ting Chung and Pao-Tsai Liao, brother Po-Han Chung, and girl friend Yun-Ru Chen, whose unreserved support continues to flow more generously than I deserve.

This research was supported by US-Israel BSF grant 2006060 and NSF grant CNS-0831289.

Chapter 1

Introduction

Soundness is a fundamental property required by many (two-party) interactive protocols studied in complexity theory and cryptography, such as interactive proofs, interactive arguments, proofs of knowledge, puzzle systems, and many two-party cryptographic primitives. In an *interactive protocol*, two parties receive some common inputs and perhaps some private inputs, toss some random coins, and interact with each other following some prescribed protocol to exchange a certain number of messages. At the end of the interaction, both parties may or may not generate some outputs. In most models, either explicitly or implicitly, a party (referred to as a prover P) wants to convince the other party (referred to as a verifier V) to accept, and V will either to accept or reject at the end of the interaction. In this case, we always require certain types of *soundness property*, which asserts that when the verifier V is supposed to reject, V will only accept with bounded (error) probability, even when he interacts with a certain class of adversarial cheating provers P^*. Such an upper bound on the error probability of V is called the *soundness error* of the protocol.

For example, in an interactive proof or interactive argument, the prover P proves to the verifier V the membership of an input x in a certain language L, and we require the following completeness and soundness properties to be satisfied. The *completeness* property says that if both parties follows the prescribed protocol honestly, then for every $x \in L$, the verifier V will accept with high probability, whereas the *soundness* property, as mentioned, guarantees that when $x \notin L$, the verifier V, even when interacting with a certain class of adversarial cheating provers P^*, will reject with high probability.

As a second example, in (bit-)commitment schemes, a sender S interacts with R to commits to a bit $b \in \{0, 1\}$ in the commit stage, and later in the decommit stage, reveals the bit b to the receiver R. The security of commitment schemes consists of the *hiding* property, which says that the receiver R cannot learn any information about the committed bit b from the commit stage, and the *binding* property, which says that the sender S cannot decommit to both 0 and 1 in the decommit stage. Both properties can be viewed as the soundness of certain interactive protocols. For the

hiding property, the receiver R plays the role of the prover, who sends to the sender S his guess b' to the committed bit b after the commit stage, and the sender S plays the role of the verifier, who accepts if the guess b' is correct. For the binding property, the sender S is the prover, who runs the commit stage first and then runs the decommit stage twice to decommit to both 0 and 1. The receiver R, who plays the role of the verifier, accepts iff he accepts both decommitments.

As usual in complexity theory and cryptography, two versions of the soundness property, *statistical soundness* and *computational soundness*, are considered. *Statistical soundness* requires the upper bound on V's error probability (to accept incorrectly) to hold against computationally unbounded adversarial provers, whereas *computational soundness* only requires the soundness to hold against *efficient* adversarial provers (e.g., ones that run in probabilistic polynomial time). Computational soundness is a weaker requirement than statistical soundness. However, in many settings, requiring only computational soundness allows us to improve the efficiency (e.g., in round complexity or communication complexity), or obtain additional properties (e.g., soundness against "reset attacks") that are impossible to be achieved together with statistical soundness. We focus on computational soundness in this thesis.

Ideally, we would like the soundness error to be as small as possible. However, in many settings, our starting point is a protocol with somewhat large soundness error. For example, to design an interactive proof for a language L, it may be easier to first design a protocol with soundness error $1/2$. This leads to the question of *soundness amplification*: Is there a way to decrease the soundness error of a given protocol?

A natural approach to soundness amplification is by *repetition*, which can be done either *sequentially* (i.e., each repetition is executed one by one) or *in parallel* (i.e., all repetitions are executed in parallel). Parallel repetition is more desirable, since it preserves the round complexity of the protocol. At the end of all repetitions, the verifier may decide to accept if all subverifiers accept (called a *direct product verifier*), if more than a certain threshold number of subverifiers accept (called a *threshold verifier*), or according to some other monotone combining functions (called a *monotone verifier*).

Intuitively, if no prover strategy can convince a verifier with probability greater than $1/2$, and we repeat the protocol n times, then it should be the case that no prover strategy can convince all n subverifiers with probability greater than $1/2^n$. Similarly, there should also be no prover strategy that can convince at least $0.6n$ subverifiers with probability greater than $e^{-\Omega(n)}$. This is indeed the case when we amplify soundness for non-interactive randomized algorithms by repetition, since the n executions are independent. However, for interactive protocols, although the n subverifiers are independent, an adversarial prover has the chance to correlate his messages among different repetitions, especially when the repetitions are done in parallel. Therefore, for interactive protocols, whether parallel repetition decreases soundness error, and more generally whether the soundness error behaves the same as for independent events under parallel repetition, becomes a subtle non-trivial question. When we

can prove that it is the case, such statements are referred to as *parallel repetition theorems*.

Soundness amplification is very related to *security amplification* and *hardness amplification* in cryptography and complexity theory. In fact, the three terms are sometimes interchangeable. Soundness is a kind of security property, so soundness amplification can be viewed as an instance of security amplification. On the other hand, the security of many cryptographic primitives can be cast in terms of a security game played between two parties, which then can be viewed as the soundness property of certain corresponding interactive protocols. Thus, results on soundness amplification for computationally sound protocols have several applications to security amplification for cryptographic primitives. Also, computational soundness amounts to hardness for any computationally efficient prover P^* to make the verifier accept. Hence, amplifying computationally soundness can be viewed as amplifying an interactive version of computational hardness.

Parallel repetition is a natural and simple approach to these amplification tasks and can be implemented in many settings. Indeed, parallel repetition has been studied for interactive proofs, interactive arguments, probabilistic checkable proofs (PCPs), amplifying hard functions, security amplification for puzzle systems and other cryptographic primitives, and more.

In this thesis, we will focus on parallel repetition for *computationally sound protocols*. It is known that for statistical soundness, under parallel repetition, the statistical soundness error behaves as if the repetitions are completely independent. However, for computational soundness, as we will discuss in next section, parallel repetition does not decrease the soundness error for general computationally sound protocols, so we need to consider restricted classes of protocols. It turns out that how parallel repetition affects computational soundness for interactive protocols is a subtle question. The answers are sensitive to the settings (classes of protocols and types of parallel verifiers) and involve constructing and analyzing subtly different reduction algorithms with different ideas.

We prove new parallel repetition theorems for several classes of computationally sound protocols such as public-coin protocols, three-message protocols, and a more general class of "computationally simulatable" protocols. For some settings, such as public-coin protocols with direct product verifiers, we obtain tight results that match information-theoretic bounds. In addition, we will discuss the strength and limitations of different reduction ideas. We hope that the discussion can make the current progress more transparent, and lead to a better understanding of parallel repetition.

Parallel repetition theorems for computationally sound protocols, as well as the reduction algorithms used in the proof, have applications to several other questions, and in particular, to security amplification for cryptographic primitives. Later in the introduction, we will discuss more about the applications and our work on improving the efficiency of security amplification for cryptographic primitives, such as commit-

ment schemes, signature schemes, message authentication codes, and CAPTCHAs.

1.1 Parallel Repetition for Computationally Sound Protocols

In this section, we discuss the question of how parallel repetition affects computational soundness of interactive protocols. We introduce some notation to facilitate the discussion.

Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be an interactive protocol. We use $\langle \mathsf{P}^n, \mathsf{V}^{n,k} \rangle$ to denote the n-fold parallel repetition of $\langle \mathsf{P}, \mathsf{V} \rangle$, where $\mathsf{V}^{n,k}$ accepts iff at least k out n subverifiers accept. The parallel verifier $\mathsf{V}^{n,k}$ is called the *threshold verifier*, and the special case $\mathsf{V}^{n,n}$ is called the *direct product verifier*.

Our original motivation is soundness amplification, for which is natural to consider threshold and direct product verifiers. However, it is also interesting to see how soundness error behaves under parallel repetition with more general types of combining functions. For a boolean function $g : \{0,1\}^n, \rightarrow \{0,1\}$, let $\langle \mathsf{P}^n, \mathsf{V}^{n,g} \rangle$ denote the n-fold parallel repetition of $\langle \mathsf{P}, \mathsf{V} \rangle$ such that $\mathsf{V}^{n,g}$ accepts iff $g(d_1, \ldots, d_n) = 1$, where d_i's are the decisions bits of the n subverifiers. We call $\mathsf{V}^{n,g}$ the *(parallel) verifier with combining function* g.

The general question to ask is, how does the computational soundness error of $\langle \mathsf{P}^n, \mathsf{V}^{n,g} \rangle$ relate to that of $\langle \mathsf{P}, \mathsf{V} \rangle$? Intuitively, we expect the soundness error to behave as if the decisions are independent events. Namely, if $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ, then $\langle \mathsf{P}^n, \mathsf{V}^{n,g} \rangle$ should have soundness error $\Pr[g(X_1, \ldots, X_n) = 1]$, where the X_i's are independent random bits with $\Pr[X_i = 1] = \delta$. Note that such upper bounds on the soundness error are optimal, since if there exists a single instance prover strategy P^* that can make V accept with probability δ, then a parallel prover strategy P^{n*}, who runs independent copies of P^* for each repetition, can make $\mathsf{V}^{n,g}$ accept with probability exactly $\Pr[g(X_1, \ldots, X_n) = 1]$. Also note that since soundness error refers to an *upper bound* on the acceptance probability, it only makes sense to consider monotone function g.

As mentioned, the subtlety is that an adversarial prover has a chance of correlating his answers among repetitions, and hence, the soundness error may not behave like independent events. It is known that the behavior of *statistical* soundness under parallel repetition matches the case of independent events [12], and so does that of computational soundness under *sequential* repetition, at least for direct product verifiers [6]. However, the situation for computational soundness under parallel repetition is much more complicated. To see why the problem is non-trivial, we first discuss the negative results of Bellare, Impagliazzo, and Naor [1], and Pietrzak and Wikström [31], which state that parallel repetition (with a direct product verifier) does not decrease the computational soundness error for some protocols.

1.1.1 Parallel Repetition May not Decrease Computational Soundness Error

In this section, we informally present the negative examples of Bellare, Impagliazzo, and Naor [1], and Pietrzak and Wikström [31] to illustrate why the computational soundness error may not behave the same as independent events under parallel repetition. The discussion in this section is meant to be informal and uses physical analogues to avoid technical details. For a formal treatment, we refer the reader to [31].

Let us consider the interactive protocol described in Figure 1.1: In the first round, both parties write down a random bit on a sheet of paper, put the paper in a box, lock the box, and send the locked box to each other. In the second round, both parties send their keys to each other, and the verifier opens the prover's box and accepts iff the two bits are distinct. Assuming that one cannot break the box B without the key k, when a prover selects his bit b', he does not know V's bit b. Hence, any prover strategy P^* can only guess a bit $b' \neq b$ with proability $1/2$, and the protocol has soundness error $1/2$.

However, when we run the protocol twice in parallel, the simple prover strategy P^{2*} described in Figure 1.1 can make both subverifiers accept simultaneously with probability $1/2$. Indeed, noting that P^{2*} simply swaps the boxes and keys of the two subverifiers, it is not hard to see that both subverifiers accept when $b_1 \neq b_2$, which happens with probability $1/2$. Therefore, the 2-fold parallel repetition $\langle \mathsf{P}^2, \mathsf{V}^{2,2} \rangle$ has soundness error at least $1/2$, as opposed to $1/4$.

Figure 1.1 gives an example protocol where two-fold parallel repetition does not decrease the soundness error at all. Note that in the example, P^{2*} correlates the two repetitions by letting the two subverifiers V_1 and V_2 play against each other, so the two repetitions are not independent in the interaction $\langle \mathsf{P}^{2*}, \mathsf{V}^{2,2} \rangle$. The physical boxes and keys used in the protocol can be implemented by using "non-malleable" commitment schemes.

The protocol in Figure 1.1 is a vanilla version of the negative examples in [1, 31], but already contains the main idea in these examples. Bellare, Impagliazzo, and Naor [1] extends the idea to show that for every $n \in \mathbb{N}$, there exists a four-message protocol $\langle \mathsf{P}_n, \mathsf{V}_n \rangle$ such that its n-fold parallel repetition $\langle \mathsf{P}_n^n, \mathsf{V}_n^{n,n} \rangle$ has essentially the same soundness error as $\langle \mathsf{P}_n, \mathsf{V}_n \rangle$. Pietrzak and Wikström [31] strengthened the result of Bellare et al. by showing that there exists a single eight-message protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ with constant soundness error such that the n-fold parallel repetition $\langle \mathsf{P}^n, \mathsf{V}^{n,n} \rangle$ has soundness error $\Omega(1)$ for every $n \in \mathbb{N}$. These negative results hold under cryptographic assumptions such as the existence of non-interactive non-malleable commitment schemes.

The negative results indicate that parallel repetition theorems are unlikely to hold for general protocols or even the class of protocols consisting of four-message protocols. However, it does not rule out the possibility that parallel repetition theorems

Don't Do It Twice in Parallel $\langle \mathsf{P}, \mathsf{V} \rangle$:

1. V writes down a random bit $b \in_R \{0,1\}$ on a sheet of paper, puts the paper in a box B, locks the box B using a key k, and send the locked box B to P.
2. P writes down another random bit $b' \in_R \{0,1\}$ on a sheet of paper, puts the paper in his box B', locks the box B' using his key k', and send the locked box B' to V.
3. V sends his key k to P.
4. P sends his key k' to V.
5. V opens the box B' using key k' and accepts if the bit $b \neq b'$.

Prover Strategy P^{2*} for $\mathsf{V}^{2,2} = (\mathsf{V}_1, \mathsf{V}_2)$:

1. P^{2*} receives boxes B_1 from V_1 and B_2 from V_2.
2. P^{2*} sends B_2 to V_1 and B_1 to V_2.
3. P^{2*} receives keys k_1 from V_1 and k_2 from V_2.
4. P^{2*} sends k_2 to V_1 and k_1 to V_2.

/* Both V_1 and V_2 accept when $b_1 \neq b_2$. */

Figure 1.1: A vanilla version of the negative examples in [1, 31].

hold for other restricted classes of protocols, such as three-message protocols and public-coin protocols. Indeed, it turns out that parallel repetition theorems hold for both three-message protocols and public-coin protocols, and some more general classes of protocols. We proceed to discuss positive results in the next section.

1.1.2 Efficient Parallel Repetition Theorems for Computationally Sound Protocols

In this section, we present our new parallel repetition theorems for different classes of computationally sound protocols, as well as other known positive results.

Recall that the general question is whether the soundness error, under parallel repetition with a certain combining function g, behaves as if the decisions are independent events. Namely, if a protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ and $g : \{0,1\}^n \to \{0,1\}$ is a boolean function, does its n-fold parallel repetition $\langle \mathsf{P}^n, \mathsf{V}^{n,g} \rangle$ have soundness error $\Pr[g(X_1, \ldots, X_n) = 1]$, where X_i's are independent bits with $\Pr[X_i = 1] = \delta$?

As discussed, soundness error $\Pr[g(X_1, \ldots, X_n) = 1]$ for $\langle \mathsf{P}^n, \mathsf{V}^{n,g} \rangle$ is the best we can hope for and it only makes sense to consider monotone combining functions. Furthermore, it is unlikely to hold for all protocols, and we need to restrict the class of protocols to obtain positive results.

For the purpose of soundness amplification, it is natural to consider the direct product verifier $\mathsf{V}^{n,n}$ or the threshold verifier $\mathsf{V}^{n,k}$ with sufficiently large threshold k (e.g., $k = (1 + \gamma) \cdot \delta n$, where δ is the soundness error of the original protocol and γ is a constant). Parallel repetition theorems for these cases are called *direct product theorems* and *Chernoff-type theorems*, respectively. On the other hand, parallel repetition with more general classes of combining functions are also interesting in their own right and have useful applications (e.g., for security amplification of commitment schemes [22]).

Before describing our contributions, we first discuss how parallel repetition theorems are proved. As usual in dealing with computational hardness, parallel repetition theorems are proved via *efficient reductions*. Namely, we construct a reduction procedure that converts a parallel prover strategy P^{n*} for $\mathsf{V}^{n,g}$ to a single-instance prover strategy P^* for V. The reduction needs to preserve the efficiency, and when P^{n*} succeeds with good probability (say, δ^n for the direct product verifier $\mathsf{V}^{n,n}$), the reduced P^* needs to succeed with large enough probability (say, δ) to obtain a contradiction. Finally, the reductions usually have an additional *black-box* property, which means that the only way P^* uses P^{n*} is to run P^{n*} with many different inputs.

All known parallel repetition theorems for computationally sound protocols are proved by efficient black-box reductions described above, and we refer to them as *efficient* parallel repetition theorems. However, both the reduction algorithms and their analyses for different settings are subtly different and require different ideas.

To get a sense about why the reductions have to be different for different settings, let us consider three-message protocols (protocols that consists of only three messages exchange) and public-coin protocols (protocols where the verifier's messages are just independent uniformly random strings). In both classes of protocols, the prover is able to simulate the verifier's messages without knowing the verifier's coin tosses. This contrasts with the negative example in Figure 1.1, where the verifier's second message is very hard to generate for the prover (unless he can generate a key just from the box). For public-coin protocols, an additional advantage is that the verifier's decision is publicly verifiable by the prover, but a challenge is that public-coin protocols may have many rounds of interactions. In contrast, three-message protocols are less interactive, but the prover cannot predict the verifier's decision from the transcript due to the lack of information about the verifier's coins. Therefore, the reduction algorithms in the two settings have to exploit different advantages and the analyses are very different.

We will discuss in more detail the differences among the different settings in Section 3.1.3 after presenting a proof of parallel repetition theorem for a basic setting.

Our contributions are as follows.[1]

A *Tight* Direct Product Theorem for Public-coin Protocols. We prove that parallel repetition with direct product verifiers decreases the soundness error of public-coin protocols from δ to δ^n, matching the information-theoretic bound. This is somewhat surprising since all previous reductions and analyses pay a price in the number m of rounds of the protocol in some ways. We prove the theorem by giving a tight analysis of the reduction prover strategy of Haståd, Pass, Pietrzak, and Wikström [19], who gave a suboptimal analysis showing that the soundness error decreases from $(1-\alpha)$ to $e^{-\Omega\left(\alpha^2 n/m\right)}$. Independently, Wikström [36] also improved the analysis of Haståd et al. [19] and showed that the soundness error decreases from $(1-\alpha)$ to $e^{-\Omega\left(\alpha^2 n\right)}$, which is not tight in comparison to $(1-\alpha)^n = e^{-\Omega(\alpha n)}$.

A Chernoff-type Theorem for Public-coin Protocols. We prove a Chernoff-type theorem for public-coin protocols with an almost matching bound to the standard Chernoff bound. More precisely, if a public-coin protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ, then the parallel protocol $\langle \mathsf{P}^n, \mathsf{V}^{n,k} \rangle$ with $k = (1+\gamma)\delta n$ has soundness error $e^{-\gamma^2 \delta n/3}$, for constants $\delta, \gamma \in (0,1)$. As in the direct product case, the previous bound of Haståd et al. [19] has undesirable dependency on the number m of rounds. Independently, Wikström [36] proved a slightly worse bound of $e^{-\gamma^2 \delta^2 n/4}$. We prove our bound by a generic reduction showing that for any class of protocols, a good enough direct product theorem implies a Chernoff-type theorem.

A *Tight* "Monotone" Repetition Theorem for Constant-round Public-coin Protocols. We show that for the special case of *constant-round* public-coin protocols, tight parallel repetition theorems hold for the most general class of monotone combining functions (referred to as a *monotone* repetition theorem). Namely, under parallel repetition with any monotone combining function, the soundness error behaves as if the repetition are completely independent. More precisely, if a constant-round public-coin protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ, then the parallel protocol $\langle \mathsf{P}^n, \mathsf{V}^{n,g} \rangle$ has soundness error $\Pr[g(X_1, \ldots, X_n) = 1]$, where $X_i's$ are i.i.d. binary random variable with $\Pr[X_i = 1] = \delta$. This generalizes the previous tight direct product theorem of Pass and Venkitasubramaniam [30].

A "Threshold" Repetition Theorem for Three-message Protocols. We prove a parallel repetition theorem for three-message (private-coin) protocols with threshold verifiers (referred to as a *threshold* repetition theorem) that almost matches the information-theoretic bound. More precisely, if a three-message protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ has

[1] In the discussion below, we omit the necessary negligible slackness in the bounds for the sake of clarity.

soundness error δ, then the parallel protocol $\langle \mathsf{P}^n, \mathsf{V}^{n,k} \rangle$ has soundness error $\Pr[\sum_i X_i \geq k]$, where $X_i's$ are i.i.d. binary random variable with $\Pr[X_i = 1] = (\delta + \alpha)$, where α is an arbitrarily small constant. Our bound for the Chernoff-type case (i.e., $k \geq (1+\gamma)\delta n$) is tight, i.e., the constant slackness α can be omitted. This generalizes and improves a previous Chernoff-type theorem [24] and (tight) direct-product theorem [1, 2]. Independent of our work, Holenstein and Schoenebeck [22] proved a stronger result of tight monotone repetition theorem for three-message protocols. Both our work and the work of Holenstein and Schoenebeck [22] generalize the reduction algorithm of Canetti, Halevi, and Steiner [2] in the same way. Holenstein and Schoenebeck obtain a stronger result by a better analysis of the same reduction algorithm.

A Direct Product Theorem for "Computationally Simulatable" Protocols. We prove a direct product theorem for a more general class of "computationally simulatable" protocols, which contains both three-message protocols and public-coin protocols as special cases. We show that for computationally simulatable protocols, parallel repetition with direct product verifiers decreases the soundness error from δ to $\delta^{n/2}$, almost matching the information-theoretic bound.

Informally, a protocol is *simulatable* if the verifier's messages (but not necessarily his decision) can be simulated with a certain quality by the prover, who does not have the verifier's coins. Computational simulatability means that the verifier's messages can be simulated in a computationally indistinguishable way against a certain class of efficient distinguishers. This property was first considered by Haståd, Pass, Pietrzak, and Wikström [19]. We generalize their definition to contain a larger class of protocols and improve their bound (which depends on the number m of rounds). Our generalization of the definition is important for an application that we discuss later.

A Chernoff-type Theorem for Computationally Simulatable Protocols. We prove a Chernoff-type theorem for computationally simulatable protocols, which gives similar bounds to the standard Chernoff bound but requires a higher threshold. More precisely, if a computationally simulatable protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ, then the parallel protocol $\langle \mathsf{P}^n, \mathsf{V}^{n,k} \rangle$ with $k = (1+\gamma)\sqrt{\delta}n$ has soundness error $e^{-\Omega(\gamma^2\sqrt{\delta}n)}$. Note that we require the threshold to be greater than $\sqrt{\delta}n$ instead of δn. Our bound is incomparable to the bounds of Haståd, Pass, Pietrzak, and Wikström [19] and their later improvement in [20]. Their bound says that for $k = (1+\gamma)\delta n$, the soundness error of $\langle \mathsf{P}^n, \mathsf{V}^{n,k} \rangle$ is $e^{-\Omega(\gamma^2\delta^2 n/m)}$, which has undesirable dependency on the number m of rounds, but only requires the threshold to be $k > \delta n$. Our bound is obtained by the generic reduction we mentioned earlier together with our direct product theorem.

Recall that the negative results of Bellare et al. [1] and Pietrzak and Wikström [31]

say that parallel repetition does not decrease the soundness error for all computationally sound protocols. Haitner [16] suggested an approach to get around the negative results and amplify the soundness for *any* protocol in a round-preserving way. His idea is to slightly modify the protocol so that parallel repetition decreases the soundness error of the modified protocol. Of course, for this to be useful, it is desirable to maintain the structure, in particular, the soundness and round complexity, of the original protocol. Following Haitner [16], we propose such a modification.

A Transformation to Make *Any* Protocols Computationally Simulatable. We propose a way to modify any protocol slightly to make it computationally simulatable. The idea, inspired by Gennaro, Gentry, and Parno [10], is to carry out the interaction under encryption. Specifically, we let both parties run the original protocol under a *fully homomorphic encryption* with the verifier's key. Fully homomorphic encryption schemes were recently constructed by Gentry [11], and these encryption schemes allow a prover to homomorphically compute an encrypted response to an encrypted verifier's message without knowing the underlying message. This clearly preserves the round complexity, and we show that running a protocol under encryption (if done properly) preserves the soundness of the protocol and makes it computationally simulatable. It follows that parallel repetition (with direct product or Chernoff-type verifiers) decreases the soundness error of the modified protocol at an exponential rate.

For completeness, we survey related work on parallel repetition theorems for computationally sound protocols below.

Related Work. Haitner [16] proposed a different way to modify interactive protocols to make parallel repetition work. In his modification, the verifier terminates and accepts the interaction with a certain probability in every round. Haitner showed that the modified "random-termination" protocol has comparable soundness to the original protocol, and parallel repetition (with direct product verifiers) decreases the soundness error at an exponential rate. Later on, Haståd, Pass, Wikström, and Pietrzak [20] observed that Haitner's modification makes protocols "weakly simulatable" in the sense that conditioned on a noticeable probability event, the verifier's messages can be simulated perfectly. Haståd et al. [20] also proved Chernoff-type theorem for the general class of weakly simulatable protocols, which generalizes the direct product theorem of Haitner [16] with quantitatively better bounds.

We remark that both results are incomparable to our results on computationally simulatable protocols and running protocols under encryption. The definitions of the computationally simulatable and weakly simulatable properties are incomparable. Weak simulatability only requires simulating the interaction conditioned on some noticeable event, but requires the simulation to be statistically close. In contrast,

	Direct Product $\mathsf{V}^{n,n}$	Chernoff-type $\mathsf{V}^{n,k}$, large k	Threshold $\mathsf{V}^{n,k}$, any k	Monotone $\mathsf{V}^{n,g}$
$O(1)$-round Public-coin	—	—	—	Thm 4.12 *I.T.*
General Public-coin	Thm 3.2 *I.T.*: $\varepsilon \leq \delta^n$	Thm 4.2 ($k = (1+\gamma)\delta n$) $\varepsilon \leq e^{-\gamma^2 \delta n/3}$	open	open
Three-message	—	—	Thm 4.7	[22] *I.T.*
Comp. Simulatable	Thm 3.16 $\varepsilon \leq \delta^{n/2}$	Thm 4.3 ($k = (1+\gamma)\sqrt{\delta} n$) $\varepsilon \leq e^{-\Omega(\gamma^2\sqrt{\delta} n)}$ [20] ($k = (1+\gamma)\delta n$) $\varepsilon \leq e^{-\Omega(\gamma^2\delta^2 n/m)}$	open	open
β-weakly Simulatable	[20] ($\delta = 1-\alpha$) $\varepsilon \leq e^{-\Omega(\alpha^2\beta^2 n/m^2)}$	[20] ($k = (1+\gamma)\delta n$) $\varepsilon \leq e^{-\Omega(\beta^2\gamma^2\delta^2 n/m^2)}$	open	open

Table 1.1: Summary of the best known results on parallel repetition theorems for computationally sound protocols. In the table, "I.T." means the bounds match the information-theoretic bounds, and "—" means it is covered by more general settings. The soundness error of the original protocol and its parallel repetition are denoted by δ and ε, respectively.

computational simulatability requires simulating the whole interaction in a computationally indistinguishable way. Our modification gives better bounds ($\delta \mapsto \delta^{n/2}$ for the direct product case) but requires the existence of fully homomorphic encryption schemes. In contrast, the modification of Haitner is unconditional, but the (improved) bound of [20] ($(1-\alpha) \mapsto e^{-\Omega(\alpha^2 n/m^4)}$ for the direct product case) is worse and depends on the number m of rounds.

A summary of parallel repetition theorems for computationally sound protocol can be found in Table 1.1. As indicated in the table, the settings of threshold verifiers and monotone verifiers remain open for *super-constant* round protocols. It seems harder to prove parallel repetition theorems when the soundness error is actually degraded as opposed to amplified, for example in the case of threshold verifier with small threshold. The currently known black-box reduction techniques seem to not apply to these settings. There are also questions of improving bounds for parallel repetition theorems for protocols with simulatable verifiers. For example, can the dependency on the number m of rounds for the setting of weakly simulatable protocols be removed?

1.2 Applications to Security Amplification

In this section, we discuss applications of efficient parallel repetition theorems for computationally sound protocols, as well as our contributions to improving the efficiency of security amplification for cryptographic primitives.

As mentioned, computational soundness of interactive protocols can be used to capture the security property of many cryptographic primitives. For the primitives whose security are captured by three-message/public-coin/simulatable protocols, parallel repetition theorems immediately give a way to amplify the security of the primitives. For example, the security of one-way functions can be viewed as a two-message protocol where the verifier V samples a random input x and sends $f(x)$ to P, and to make V accept, P needs to return some pre-image $x' \in f^{-1}(f(x))$. Similarly, weakly verifiable puzzle systems of [2], where a puzzle generator generates a puzzle for a solver to solve, are essentially two-message interactive protocols. Parallel repetition theorems for three-message protocols imply security amplification for these primitives.

Another straightforward application is to error reduction of interactive arguments. Interactive arguments are simply interactive proofs with computational soundness, where the prover proves to the verifier that an input x is in a certain language L. Given an interactive argument $\langle \mathsf{P}, \mathsf{V} \rangle$ with constant completeness and soundness, we can apply parallel repetition with a threshold verifier of proper threshold to amplify both the completeness and soundness properties, provided that a Chernoff-type theorem holds for the given protocol $\langle \mathsf{P}, \mathsf{V} \rangle$.

However, in many other cases, the security properties of primitives is more interactive and is not captured by the class of protocols where parallel repetition theorems are available. Nevertheless, the security property may have additional structure so that the black-box reduction algorithms used to prove parallel repetition theorems for computationally sound protocols can be implemented in the corresponding settings. We present two applications of this type below.

Security Amplification for Commitment Schemes. Commitment schemes are interactive protocols that are digital analogue of safes, where Alice can put a value inside the safe and send it to Bob without leaking any information about the value (hiding property), and later on, Alice can only open the safe in one way to reveal a unique value to Bob (binding property). The goal of security amplification is to turn a weak bit-commitment scheme Com_0, where both properties can be broken with bounded but, say, constant probability, to a fully secure one, where both properties can be broken with only a negligible probability. Security amplification for commitment schemes require more complicated construction than simple parallel repetition, but understanding the hardness of, say, breaking the binding property of at least k out of n calls to Com_0, is useful to analyze the constructions.

We construct a black-box transformation that amplifies a weak commitment scheme Com_0 with constant security to a fully secure one, using only $\omega(\log s)$ black-box calls

to Com_0, where s is a security parameter. Furthermore, our resulting scheme is a string-commitment scheme that commits to a $\Omega(\log s)$-bit string. This improves the efficiency over the previous work of Halevi and Rabin [17], which requires $\omega(\log^2 s)$ black-box calls to securely commit a single bit. The key of our improvement is to use error-correcting codes and randomness extractors to amplify both the hiding and binding property *simultaneously* as opposed to separately in [17].

To analyze our transformation, we prove a Chernoff-type theorem for repetition of weak commitment schemes. Noting that the commit stage of Com_0 may consist of multiple rounds, it can be shown that to amplify the security, the calls to Com_0 needs to be done sequentially as opposed to in parallel (by generalizing the negative examples of Bellare et al. [1]). However, even if the calls to Com_0 are sequential in the commit stage, all calls to Com_0 are decommitted in parallel in the reveal stage. This can be viewed as a special type of "two-phase repetition," where the first phase is sequential and the second phase is parallel. It turns out that the black-box reduction for proving parallel repetition theorems for three-message protocols can be implemented in this setting to prove corresponding parallel repetition theorems.

Security Amplification for Dynamic Weakly Verifiable Puzzle Systems and Related Primitives. Dodis, Impagliazzo, Jaiswal, and Kabanets [7] defined "dynamic weakly verifiable puzzle systems" to capture the security properties of several cryptographic primitives such as message authentication codes (MACs), signature schemes (SIGs), and pseudorandom functions (PRFs). They proved a Chernoff-type theorem for the puzzle systems and used it to prove security amplification for the corresponding primitives.

We improve the bound of the Chernoff-type theorem of Dodis et al. [7] to almost match the corresponding information-theoretic bound, and hence improve the efficiency of security amplification for the related primitives. Our improvement is obtained by observing that the reduction for proving parallel repetition theorems for three-message protocols can be implemented and used to improve the main step of the analysis of Dodis et al. [7].

We remark that the security of these primitives can also be captured as the soundness of certain interactive protocols. For example, consider the chosen message attack (CMA) security for MACs. The security can be viewed as an interaction between an adversary and a user, who has the secret key. The interaction has a unspecified polynomial number of rounds where the user tags the messages sent by the adversary, and at the end of interaction, the user accepts if the adversary sends a fresh message with a valid tag in his last message. However, this protocol is not simulatable so parallel repetition theorems are not directly applicable. Nevertheless, when we model the security more carefully as dynamic weakly verifiable puzzle systems, the additional structure allows us to prove parallel repetition theorems for this model.

In addition to security amplification, we present one more simple application below.

Sequential Repetition for Computationally Sound Protocols. While it is believed that computational soundness behaves well under sequential repetition, it seems that only a direct product theorem is found in literature [6]. We observe that the black-box reduction for proving parallel repetition theorems for three-message protocols can be implemented for sequential repetition of any interactive protocols. It follows that the proof also gives a tight sequential repetition theorem for computationally sound protocols with any monotone combining functions.

1.3 Roadmap

We outline the remaining of the thesis in this section. After definitions and preliminaries in Chapter 2, we will present a series of black-box reductions to prove parallel repetition theorems for different settings in Chapter 3—4. We will start with direct product theorems for different classes of protocols, and then generalize to the setting of Chernoff-type and threshold/monotone verifiers. We will compare all known reductions and discuss their strengths and limitations. In Chapter 5, we present applications to security amplification of cryptographic primitives mentioned above. A more detailed outline is as follows.

Definitions and Preliminaries. In this chapter, we present necessary definitions of computationally sound protocols and parallel repetitions, as well as some preliminary on black-box reductions.

Efficient Direct Product Theorems. In this chapter, we focus on proving direct product theorems for different classes of protocols. We start with a basic setting of three-message public-coin protocols, where the reduction is essentially the same as that of Yao's security amplification for one-way functions [38]. We present this to illustrate the general framework of black-box reductions. We then prove the direct product theorems for public-coin protocols, three-message protocols, and computationally simulatable protocols. We will also present our generic transformation of running a protocol under encryption that makes any protocol computationally simulatable.

Efficient Chrenoff-type and Threshold/Monotone Repetition Theorems. In this chapter, we prove parallel repetition theorems for other combining functions. We first present a generic reduction showing that a good enough direct product theorem implies Chernoff-type theorems, and use it to obtain Chernoff-type theorem for public-coin protocols and computationally simulatable protocols. We then generalize

the reduction for three-message protocols from the case of direct product verifiers to the case of threshold verifiers. We present the better analysis of Holenstein and Schoenebeck [22] since it gives better parameters. We present it for threshold verifiers as opposed to the more general case of any monotone combining function since we feel that the threshold case is more intuitive. Finally, we prove a monotone repetition theorem for constant-round public-coin protocols.

Applications to Security Amplification. In this chapter, we present applications of parallel repetition theorems to security amplification for cryptographic primitives as discussed in Section 1.2 above. As mentioned, the security property of commitment schemes, message authentication codes, digital signatures, and pseudorandom functions can be captured by variants of "puzzle systems." We will show that the reductions for proving parallel repetition theorems for interactive protocols can be used to prove corresponding repetition theorems for puzzle systems, which provide useful tools to analyze the security amplification constructions for corresponding primitives. Specifically, we propose new constructions to improve the efficiency of security amplification for commitment schemes, and improve the Chernoff-type theorem of Dodis et al. [7] for dynamic weakly verifiable puzzle systems.

Chapter 2

Definitions and Preliminaries

In this chapter, we present the necessary definitions of computationally sound protocols for studying parallel repetition theorems in later sections. Along the way, we also introduce some notations, conventions, as well as preliminaries.

2.1 Computationally Sound Protocols

We consider the following general setting of two-party protocols $\langle \mathsf{P}, \mathsf{V} \rangle$. We refer to the two parties the prover P and the verifier V, which are both PPT (probabilistic polynomial time) algorithms. Before the interaction, both parties receive a *common input* x from some *domain* $\Lambda \subset \{0,1\}^*$, which can be, for example, the input for which V wants to decide the membership to some language L, the messages from previous interaction with perhaps some other parties, or some public keys. We assume without loss of generality that a *security parameter* 1^s is encoded in the input x with $|x| = s^{O(1)}$ and the complexity of both parties are measured by this security parameter s. In the interaction, P and V exchange a fixed number of messages (which may depend on the security parameter), each of which has a predefined length. At the end of the interaction, V decides either to accept (output 1) or reject (output 0) the interaction based on his *view*, which consists of the common input, the transcript of the interaction, and his random coins. The interaction is denoted by $\langle \mathsf{P}, \mathsf{V} \rangle(x)$, and $\Pr[\langle \mathsf{P}, \mathsf{V} \rangle(x) = 1]$ is the probability that the verifier accepts at the end of the interaction, where the probability is over the randomness of both the prover strategy and the verifier strategy.

The *soundness* of a protocol is an upper bound on the probability that the verifier V accepts an input x that he is supposed not to accept when V interacts with a certain class of prover strategies. For example, when the protocol is used to decide membership to some language L (i.e., the setting of interactive proofs/arguments), the soundness refers to the (error) probability that the verifier accepts an input $x \notin L$ when interacts with a certain class of cheating prover strategies P^*. When the

soundness holds against all cheating prover strategies, it is called *statistical soundness*. We will focus on *computational soundness*, where the soundness only holds against any *efficient* PPT cheating prover strategy P^*. Note that soundness is a property of the verifier V. A formal definition of computational soundness is as follows.

Definition 2.1 (Computational Soundness) *Let* V *be a* PPT *verifier for an interactive protocol* $\langle \mathsf{P}, \mathsf{V} \rangle$ *with domain* Λ*, and* $\varepsilon : \Lambda \to [0, 1]$ *an efficiently computable function. The protocol has* **computational soundness** ε *if for every* PPT *prover strategy* P^**, for sufficiently large* s*, and for every common input* $x \in \Lambda$ *with security parameter* s*, we have*

$$\Pr[\langle \mathsf{P}^*, \mathsf{V} \rangle(x) = 1] \leq \varepsilon(x),$$

where the probability is over the randomness of both strategies P^* *and* V*.*

The ε in the above definition is called the *soundness error* of the protocol, and we call $\Pr[\langle \mathsf{P}^*, \mathsf{V} \rangle(x) = 1]$ the success probability of P^* in convincing V. We remark that the choice of PPT as the notion of efficiency is not essential, and our results of parallel repetition theorems have analogues to other time bounds as well as for concrete security.

The above definition of computational soundness can be used to capture the soundness/security property of cryptographic primitives. When we set the domain $\Lambda = \bar{L}$, the complement of a language L, the definition captures the standard soundness property of interactive arguments. On the other hand, as explained in the introduction, both the hiding and binding properties of commitment schemes can be captured as the soundness of certain interactive protocols.

A protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ is *public-coin* if V's messages are independent random coins. We say that $\langle \mathsf{P}, \mathsf{V} \rangle$ is a *c-message* protocol if the total number of messages sent by P and V is c. For three-message protocols, P's first message is denoted by w, V's first message is denoted by v, and P's second message is denoted by p. Note that the prover always sends the first message, since otherwise, the verifier would send the last message, which cannot affect his decision. On the other hand, one *round* means two messages exchanged. When we say that $\langle \mathsf{P}, \mathsf{V} \rangle$ is a m*-round* protocol, we mean $\langle \mathsf{P}, \mathsf{V} \rangle$ consists of $2m$ messages and we assume (w.l.o.g.) that the verifier V sends the first message. The vierifer V's (resp., the prover P's) messages are denoted by $v_1, \ldots, v_m$ (resp., $p_1, \ldots, p_m$).

2.2 Parallel Repetition of Protocols

We proceed to consider parallel repetition of a protocol $\langle \mathsf{P}, \mathsf{V} \rangle$. Informally, in a n-fold parallel repetition $\langle \mathsf{P}^n, \mathsf{V}^n \rangle$, both parties run n copies of the original protocol in parallel. For example, suppose $\langle \mathsf{P}, \mathsf{V} \rangle$ is a three-message protocol, then in the parallel protocol $\langle \mathsf{P}^n, \mathsf{V}^n \rangle$, P^n first sends n messages $(w_1, \ldots, w_n)$ to V^n, then V^n

sends $(v_1, \ldots, v_n)$ to P^n, and finally P^n sends $(p_1, \ldots, p_n)$ to V^n. At the end of the interaction, each subverifier of V^n makes a decision based on the corresponding copy of the interaction, and the parallel verifier V^n can decide to accept/reject based on the n subverifiers' decisions in different ways. For example, the parallel verifier may accept only when all subverifiers accept, or when at least k out of n subverifiers accept. Since a verifier strategy is sufficient to specify a protocol, we define parallel repetition of protocols by defining parallel verifiers.

Definition 2.2 (Parallel Verifiers) *Let V be a* PPT *verifier for an interactive protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ with domain Λ. Let $n : \mathbb{N} \to \mathbb{N}$ and $g : \{0,1\}^n \to \{0,1\}$ be efficiently computable. We define an n-***fold parallel verifier with combining function** *g, denoted by $\mathsf{V}^{n,g}$, to be the following* PPT *verifier for an interactive protocol $\langle \mathsf{P}^n, \mathsf{V}^{n,g} \rangle$ with the same domain Λ.*

$\mathsf{V}^{n,g} = (\mathsf{V}_1, \ldots, \mathsf{V}_n; g)$ consists of n copies V_i of the original verifier V, each of which has its own independent random tape. Upon receiving a common input $x \in \Lambda$, $\mathsf{V}^{n,g}$ interacts with a prover P^n by running the n subverifiers V_i in parallel. At the end of the interaction, each subverifier V_i outputs a decision bit d_i, and the decision of the parallel verifier $\mathsf{V}^{n,g}$ is $g(d_1, \ldots, d_n)$.

When the combining function g is a monotone function, we also refer to $\mathsf{V}^{n,g}$ as a *monotone* verifier. We also define the following three special cases of parallel verifiers.

- **Threshold verifiers.** A threshold verifier $\mathsf{V}^{n,k}$ accepts iff k out of n subverifiers accept. This corresponds to the case where g is a threshold function with threshold k.

- **Direct product verifiers.** This is a special case of the threshold verifiers with $k = n$. Namely, a direct product verifier $\mathsf{V}^{n,n}$ accepts iff all the subverifiers V_i's accept. The corresponding g is the AND function.

- **Chernoff-type verifiers.** This is a special case of the threshold verifiers with sufficiently large threshold k. If the original protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ and the threshold $k \geq (1+\gamma) \cdot \delta n$ for some constant $\gamma > 0$, then the corresponding threshold verifier $\mathsf{V}^{n,k}$ is also called a Chernoff-type verifier.

We are interested in how the soundness property behaves under parallel repetition with different types of parallel verifiers. Ideally, we would like to show that the behavior of the soundness error matches the information-theoretic analogue. Namely, if a protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ, then the parallel protocol $\langle \mathsf{P}^n, \mathsf{V}^{n,g} \rangle$ has soundness error $\Pr[g(d_1, \ldots, d_n) = 1]$, where the d_i's are independent random bits with $\Pr[d_i = 1] = \delta$. In particular, for the direct product and Chernoff-type verifiers, we hope that the soundness error decreases in an exponential rate in the number of repetition n, and ideally to δ^n and $e^{-\Omega(\gamma^2 \delta n)}$, respectively.

We refer to such upper bounds on the soundness error of parallel protocols $\langle \mathsf{P}^n, \mathsf{V}^{n,g} \rangle$ as *parallel repetition theorems* in general, and *direct product theorems*, *Chernoff-type theorems*, *threshold repetition theorems*, and *monotone repetition theorems* for protocols with corresponding type of parallel verifiers.

As discussed in the introduction, the above stated bounds are optimal and can only hold for monotone verifiers. Also, assuming standard cryptographic assumptions, it is known that parallel repetition (with direct product verifiers) does not decrease the soundness error for general interactive protocols [1, 31]. The focus of this thesis is on the positive side, where we study parallel repetition theorems for natural classes of protocols with different types of parallel verifiers. All parallel repetition theorems studied in this thesis are proved by *black-box reductions*, which give stronger results and may be useful for other settings. We discuss black-box reductions in the next section.

2.3 Preliminaries on Black-Box Reductions

In this section, we present some preliminaries on proving parallel repetition theorems by black-box reductions. Let us take a direct product theorem for public-coin protocols as an example. In this example, our goal is to show:

"*If a public-coin protocol* $\langle \mathsf{P}, \mathsf{V} \rangle$ *has soundness error* δ*, then its* n*-fold parallel repetition* $\langle \mathsf{P}^n, \mathsf{V}^{n,n} \rangle$ *has soundness error* δ^n*.*"

As usual in cryptography, we prove such a statement by proving its contrapositive:

"*If there exists a* PPT *parallel strategy* P^{n*} *that can convince* $\mathsf{V}^{n,n}$ *with probability at least* δ^n*, then there exists a* PPT *single instance prover strategy* P^* *that can succeed with probability at least* δ *in convincing* V*.*"

Specifically, we give a reduction procedure that exploits the given parallel strategy P^{n*} to construct a single instance prover strategy P^* with good success probability. Furthermore, as it is often the case in cryptography, the reduction prover strategy P^* we constructed has the additional property that P^* only uses the parallel strategy P^{n*} in a black-box way. Namely, the only way that P^* uses P^{n*} is to run P^{n*} with many different inputs (and coins) specified by P^*. Such a reduction is called a *black-box reduction*.

As we discussed in the introduction, black-box reductions are desirable as they give a stronger and more general result. Also, sometimes the same reduction algorithm can be implemented in different models and give unified proof for results in different settings.

On the other hand, at least on the intuitive level, it seems that P^* needs to run in time at least $\Omega(1/\delta^n)$. This is because that it is possible for P^{n*} to convince $\mathsf{V}^{n,n}$ with probability δ^n, but fail to convince any subverifiers with the remaining probability $1-\delta^n$ (e.g., P^{n*} aborts with probability $1-\delta^n$). Since P^* does not know the structure of P^{n*}, it seems that P^* needs to sample at least $\Omega(1/\delta^n)$ times to obtain useful

information from P^{n*}. Indeed, all known reduction prover strategies P^* for parallel repetition theorems use various sampling techniques to exploit the parallel prover P^{n*}, and have runtime polynomial in the inverse of the success probability of P^{n*}. Since the reduction prover strategy P^* needs to be efficient for obtaining contradiction, black-box reductions can only prove that the soundness error decreases to $1/\mathrm{poly}(s)$ for any polynomial $\mathrm{poly}(s)$ (when efficiency is interpreted as PPT).

As an example, we state the following theorem, which is proved in Section 3.2 by a black-box reduction. The theorem says that given a parallel prover P^{n*} with success probability ε, we can obtain a single instance prover strategy P^* with success probability at least roughly $\varepsilon^{1/n}$.

Theorem 2.3 (same as Theorem 3.2) *Let $\mathsf{V} \in \mathrm{PPT}$ be a public-coin verifier. There exists a prover strategy P^* such that for every common input $x \in \{0,1\}^*$, every $n \in \mathbb{N}$, every $\varepsilon, \xi \in (0,1)$, and every parallel prover strategy P^{n*},*

1. $\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1] \geq \varepsilon \Rightarrow$
$$\Pr[\langle \mathsf{P}^{*(\mathsf{P}^{n*})}(n, \varepsilon, \xi), \mathsf{V} \rangle(x) = 1] \geq \varepsilon^{1/n} \cdot (1 - \xi).$$
2. $\mathsf{P}^{*(\cdot)}(x, n, \varepsilon, \xi)$ *runs in time* $\mathrm{poly}(|x|, n, \varepsilon^{-1}, \xi^{-1})$ *given oracle access to* $\mathsf{P}^{n*}(x)$.

Note that in the above theorem, there is a slackness parameter ξ that trades off the closeness to the ideal success probability $\varepsilon^{1/n}$ with the runtime of P^*. Also note that the runtime of P^* is allowed to be polynomial in both ε^{-1} and ξ^{-1}. This is sufficient to give the desired upper bound on the soundness error of the parallel protocol $\langle \mathsf{P}^n, \mathsf{V}^{n,n} \rangle$ up to a negligible additive term, as we show in the following corollary. We remark that similar corollaries of parallel repetition theorems (may be with slightly different form of slackness parameters) in later chapters can be proved in very similar ways. We omit the proofs of similar corollaries to avoid repetitive arguments.

Corollary 2.4 *Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a public-coin protocol with input domain Λ, $\delta : \Lambda \to [0,1]$ and $n : \mathbb{N} \to \mathbb{N}$ efficiently computable functions with $n \leq \mathrm{poly}(s)$. If $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ, then its n-fold parallel repetition with direct product verifier $\langle \mathsf{P}^n, \mathsf{V}^{n,n} \rangle$ has soundness error $\delta^n + \mathsf{ngl}$, where ngl denotes a negligible function in the security parameter s.*

Proof. We prove it by contradiciton. Suppose the conclusion is not true, then there exists a PPT parallel prover P^{n*} and a noticeable η such that for infinitely many $s \in \mathbb{N}$, there exists some x with security parametmer s such that

$$\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1] > \delta^n(x) + \eta(s).$$

Consider the reduction prover strategy P^* defined in Theorem 2.3 with parameters $\varepsilon = \delta^n + \eta$ and $\xi = \eta/n$. By Theorem 2.3, P^* runs in time $\mathrm{poly}(|x|, n, \delta^n + \eta, \eta/n) = \mathrm{poly}(s)$ and for every x that satisfies the above inequality, we have

$$\Pr[\langle \mathsf{P}^*, \mathsf{V} \rangle(x) = 1] \geq (\delta^n + \eta)^{1/n} \cdot (1 - \xi) > \delta,$$

which contradicts to the fact that $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ. ■

In Section 3.2, we actually prove the above Theorem 2.3 assuming that the parallel prover strategy P^{n*} is *deterministic*. However, this assumption can be made without loss of generality as we argue below. Intuitively, if a randomized parallel prover P^{n*} strategy has success probability ε, then by an averaging argument, there must exist some coins r such that $\mathsf{P}^{n*}(r)$ (i.e., P^{n*} with the fixed coins r) can also succeed with probability at least ε. Furthermore, we can use sampling to find some coins r such that $\mathsf{P}^{n*}(r)$ can succeed with probability at least roughly ε, and this is sufficient since we can exploit the slackness parameter ξ. We formalize the above discussion in the following two lemmas. We emphasize that the proofs of the following two lemmas are general and can be applied to *any* interactive protocol.

Lemma 2.5 *There exists an efficient transformation* Derand *such that for every* PPT *verifier* V*, (randomized)* PPT *prover strategy* P^**, common input* $x \in \{0,1\}^*$*, parameters* $\varepsilon, \xi, \alpha \in (0,1)$*,* Derand *on the above input outputs a* deterministic *prover strategy* $\tilde{\mathsf{P}}^*$ *such that if* $\Pr[\langle \mathsf{P}^*, \mathsf{V} \rangle(x) = 1] \geq \varepsilon$*, then with probability* $(1-\alpha)$ *over the randomness of* Derand*,*

$$\Pr[\langle \tilde{\mathsf{P}}^*, \mathsf{V} \rangle(x) = 1] \geq \varepsilon \cdot (1 - \xi).$$

Furthermore, both Derand *and the output* $\tilde{\mathsf{P}}^*$ *can be implemented with oracle access to* P^* *with runtime* $\mathrm{poly}(|x|, \varepsilon^{-1}, \xi^{-1}, \log(1/\alpha))$ *and* $\mathrm{poly}(|x|)$*, respectively.*

Proof. Let r denote the coins used by P^*. By a Markov argument, $\Pr_r[\langle \mathsf{P}^*(r), \mathsf{V} \rangle(x) = 1] \geq \varepsilon$ implies that with probability at least $\varepsilon\xi$ over the coins r,

$$\Pr[\langle \mathsf{P}^*(r), \mathsf{V} \rangle(x) = 1] \geq \varepsilon \cdot (1 - \xi).$$

Hence, P^* with fixed such coins r is a desired deterministic prover strategy. Intuitively, we can find such coins r by sampling, and checking candidate coins r by estimating the success probability $\Pr[\langle \mathsf{P}^*(r), \mathsf{V} \rangle(x) = 1]$ also using sampling. Formally, a description of Derand can be found in Figure 2.1. The parameters are adjusted to accommodate the sampling errors.

We proceed to analyze Derand. Define a set of *good* coins r by defining

$$\mathsf{Good} = \{r : \Pr[\langle \mathsf{P}^*(r), \mathsf{V} \rangle(x) = 1] \geq \varepsilon \cdot (1 - \xi/4)\}.$$

By a Markov argument, we have $\Pr[r \in \mathsf{Good}] \geq (\varepsilon\xi/4)$. Now, observe that the constants in M_1, M_2 can be chosen so that the following holds.

- With probability at least $(1 - \alpha/2)$ over the M_1 random samples of coins r, at least one sample of r is good. We say that the sampling fails if no good r is found.
- In estimating $\hat{p}(r)$ for coins r, with probability at least $(1 - \alpha/(2M_1))$ over the M_2 random samples of simulation $\langle \mathsf{P}^*(r), \mathsf{V} \rangle(x)$, the estimator $\hat{p}(r)$ satisfies $|\hat{p}(r) - p(r)| \leq (\varepsilon\xi/4)$. We say that the sampling fails if $|\hat{p}(r) - p(r)| > (\varepsilon\xi/4)$.

$\mathsf{Derand}(\mathsf{P}^{n*}, n, \varepsilon, \xi, \alpha)$
/* Implicitly, there is a PPT verifier V and an input x as part of the input. */
Repeat the following at most $M_1 = O\left(\frac{1}{\varepsilon\xi} \cdot \log \frac{1}{\alpha}\right)$ times.

- Sample uniformly random coins r and estimate the success probability of $\mathsf{P}^*(r)$ (i.e., P^* with the fixed coins r), denoted by $p(r) \stackrel{\text{def}}{=} \Pr[\langle \mathsf{P}^*(r), \mathsf{V}\rangle(x) = 1]$, as follows: Simulate the interaction $\langle \mathsf{P}^*(r), \mathsf{V}\rangle(x)$ for $M_2 = O\left(\frac{1}{\varepsilon^2\xi^2} \cdot \log \frac{1}{\varepsilon\xi\alpha}\right)$ times (with fresh randomness for V), and compute an estimator $\hat{p}(r) = (\text{number of accept interactions})/M_2$.
- If $\hat{p}(r) \geq \varepsilon \cdot (1 - \xi/2)$, then output $\tilde{\mathsf{P}}^* \stackrel{\text{def}}{=} \mathsf{P}^*(r)$.

Return $\tilde{\mathsf{P}}^* \stackrel{\text{def}}{=} \mathsf{P}^*(r)$ for some arbitrary coins r (or simply aborts).

Figure 2.1: Correlation reduction for direction product verifiers.

Note that Derand performs the first type sampling once, and the second type sampling at most M_1 times. By a union bound, with probability at least $(1 - \alpha)$ over the randomness of Derand, no sampling fails. Now, we argue that when no sampling fails, Derand outputs a deterministic $\tilde{\mathsf{P}}^*$ with

$$\Pr[\langle \tilde{\mathsf{P}}^*, \mathsf{V}\rangle(x) = 1] \geq \varepsilon \cdot (1 - \xi).$$

We first observe that Derand can find a sample of coins r in M_1 samples such that Derand outputs $\tilde{\mathsf{P}}^* = \mathsf{P}^*(r)$ for the coins r. This is because there exists a sample of good coins $r \in \mathsf{Good}$, and the corresponding estimator satisfies

$$\hat{p}(r) \geq p(r) - (\varepsilon\xi/4) \geq \varepsilon \cdot (1 - \xi/4) - (\varepsilon\xi/4) = \varepsilon \cdot (1 - \xi/2).$$

Also, when Derand outputs $\tilde{\mathsf{P}}^* = \mathsf{P}^*(r)$ for one of M_1 samples of r, we have

$$\Pr[\langle \tilde{\mathsf{P}}^*, \mathsf{V}\rangle(x) = 1] = p(r) \geq \hat{p}(r) - (\varepsilon\xi/4) \geq \varepsilon \cdot (1 - \xi/2) - (\varepsilon\xi/4) > \varepsilon \cdot (1 - \xi).$$

Finally, it is easy to see by inspection that both Derand and the output $\tilde{\mathsf{P}}^*$ can be implemented with oracle access to P^* with runtime $\mathrm{poly}(|x|, \varepsilon^{-1}, \xi^{-1}, \log(1/\alpha))$ and $\mathrm{poly}(|x|)$, respectively, provided that the oracle access to P^* is allowed to specify the random coins r used by P^*. ■

The next lemma says that by Lemma 2.5, we can assume without loss of generality that in proving Theorem 2.3, the given parallel prover strategy P^{n*} is deterministic. We remark that it is not hard to modify the proof of the following lemma to show that in proving parallel repetition theorems in later chapters, the parallel prover P^{n*} can also be assumed to be deterministic without loss of generality. We omit the proofs of these similar statements to avoid repetitive arguments.

Lemma 2.6 *If Theorem 2.3 holds for the special case where the parallel prover P^{n*} is deterministic, then Theorem 2.3 also holds for the general case with randomized P^{n*}.*

Proof. Let P^*_{det} be the reduction prover strategy of the deterministic-version of Theorem 2.3 assumed in the lemma. The desired reduction prover strategy P^* simply compose P^*_{det} with the Derand defined in Lemma 2.5 with proper chosen slackness ξ. Specifically, $\mathsf{P}^{*(\mathsf{P}^{n*})}(x, n, \varepsilon, \xi)$ is defined as follows.

- Apply Derand to P^{n*} with parameters ξ set to $\xi/3$ and α set to $(\varepsilon\xi/3)$ to obtain a deterministic prover strategy $\tilde{\mathsf{P}}^{n*}$.
- Run P^*_{det} with oracle $\tilde{\mathsf{P}}^{n*}$ and parameters ε set to $\varepsilon \cdot (1 - \xi/3)$ and ξ set to $\xi/3$.

It is easy to check by inspection that $\mathsf{P}^{*(\cdot)}(x, n, \varepsilon, \xi)$ runs in time $\mathrm{poly}(|x|, n, \varepsilon^{-1}, \xi^{-1})$ given oracle access to $\mathsf{P}^{n*}(x)$. Now, suppose for given parameters x, n, ε, ξ and a randomized prover strategy P^{n*},

$$\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1] \geq \varepsilon.$$

Then with probability at least $(1 - (\varepsilon\xi/3))$, the $\tilde{\mathsf{P}}^{n*}$ returned by Derand satisfies

$$\Pr[\langle \tilde{\mathsf{P}}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1] \geq \varepsilon \cdot (1 - \xi/3).$$

Let us call this event Good. When the event Good happens, we have

$$\Pr[\langle \mathsf{P}^{*(\tilde{\mathsf{P}}^{n*})}_{det}(n, \varepsilon\cdot(1-\xi/3), \xi/3), \mathsf{V} \rangle(x) = 1] \geq (\varepsilon \cdot (1 - \xi/3))^{1/n} \cdot (1-\xi/3) \geq \varepsilon\cdot(1-(2\xi/3)).$$

Finally, we have

$$\begin{aligned} \Pr[\langle \mathsf{P}^*(n, \varepsilon, \xi), \mathsf{V} \rangle(x) = 1] &\geq \Pr[\langle \mathsf{P}^*(n, \varepsilon, \xi), \mathsf{V} \rangle(x) = 1 | \mathsf{Good}] - \Pr[\neg \mathsf{Good}] \\ &\geq \varepsilon \cdot (1 - (2\xi/3)) - (\varepsilon\xi/3) \\ &= \varepsilon \cdot (1 - \xi), \end{aligned}$$

as desired. ■

2.4 Additional Notation and Conventions

In this section. we introduce some more notations and conventions that is used throughout this thesis.

For an n-tuple of messages $\vec{v} = (v_1, \ldots, v_n)$, we will often single out a coordinate $i \in [n]$ and refer to the remaining $n - 1$ coordinates "$-i$." Namely, we write $\vec{v} = (v_i, \vec{v}_{-i})$. Similarly, we write a parallel verifier $\mathsf{V}^{n,n} = (\mathsf{V}_i, \mathsf{V}_{-i})$.

Recall that the verifier's messages of a m-round protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ are denoted by $v_1, \ldots, v_m$. We will use $v_{[\ell]} = (v_1, \ldots, v_\ell)$ to denote the first ℓ messages. Similarly, for its parallel repetition $\mathsf{V}^{n,n}$, we write $\vec{v}_{[\ell]} = (\vec{v}_1, \ldots, \vec{v}_\ell)$, $v_{[\ell],i} = (v_{1,i}, \ldots, v_{\ell,i})$, and $\vec{v}_{[\ell],-i} = (\vec{v}_{1,-i}, \ldots, \vec{v}_{\ell,-i})$.

Chapter 3

Efficient Direct Product Theorems

3.1 Efficient Direct Product Theorem for Three-Message Public-Coin Protocols

In this section, as a warm up, we first present an efficient direct product theorem for the simplest case of three-message public-coin protocols, which says that n-fold parallel repetition reduces soundness error from δ to $\delta^n + \mathsf{ngl}$ for three-message public-coin protocols. We prove it by constructing a black-box reduction that converts a parallel prover strategy P^{n*} with success probability ε to a single instance prover strategy P^* with success probability close to $\varepsilon^{1/n}$. The reduction is essentially a classical reduction of Yao [38] for proving security amplification for one-way functions. The reductions for the more general classes of protocols we study later can be viewed as different generalizations of this classical reduction. We present the classical reduction in the setting of three-message public-coin protocols to illustrate the common intuition and framework of these reductions. Formally, we prove the following theorem.

Theorem 3.1 *Let* $\mathsf{V} \in \mathrm{PPT}$ *be a three-message public-coin verifier. There exists a prover strategy* P^* *such that for every common input* $x \in \{0,1\}^*$*, every* $n \in \mathbb{N}$*, every* $\varepsilon, \xi \in (0,1)$*, and every parallel prover strategy* P^{n*}*,*

1. $\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1] \geq \varepsilon \Rightarrow$

$$\Pr[\langle \mathsf{P}^{*(\mathsf{P}^{n*})}(n, \varepsilon, \xi), \mathsf{V} \rangle(x) = 1] \geq \varepsilon^{1/n} \cdot (1 - \xi).$$

2. $\mathsf{P}^{*(\cdot)}(x, n, \varepsilon, \xi)$ *runs in time* $\mathrm{poly}(|x|, n, \varepsilon^{-1}, \xi^{-1})$ *given oracle access to* $\mathsf{P}^{n*}(x)$.

Recall that for three-message protocols, we use w, v, p to denote the three messages of the protocol, and the messages of the n-fold parallel protocol are denoted by $\vec{w} = (w_1, \ldots, w_n), \vec{v} = (v_1, \ldots, v_n)$, and $\vec{p} = (p_1, \ldots, p_n)$. As mentioned, by Lemma

2.5, 2.6, we can assume without loss of generality that our starting point is a *deterministic* parallel prover strategy P^{n*} with success probability at least ε. Since P^{n*} is deterministic, $\vec{w}$ is fixed and the outcome of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x)$ is determined by $\mathsf{V}^{n,n}$'s message $\vec{v}$. For convenience, we write $\vec{p} = \mathsf{P}^{n*}(\vec{v})$, and we say "$\mathsf{P}^{n*}(\vec{v})$ convinces V_i" if the i-th subverifier V_i accepts in $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x)$ when $\mathsf{V}^{n,n}$'s message is $\vec{v}$.

Our goal is to construct a single instance prover P^* with success probability arbitrarily close to $\varepsilon^{1/n}$. To exploit P^{n*} to convince V, a natural idea is to let P^* interact with V by simulating the interaction between P^{n*} and $\mathsf{V}^{n,n}$ internally. For this, P^* selects a coordinate $i \in [n]$ of $\mathsf{V}^{n,n}$ for the external verifier V to play, and P^* simulates the remaining $n-1$ subverifiers of $\mathsf{V}^{n,n}$ and the parallel prover P^{n*} by itself. Since the protocol is three-message and public-coin, simulating a subverifier V_j is amount to select a random message v_j for V_j.

In more detail, P^* first runs P^{n*} to obtain $\vec{w}$, and sends w_i to V. Upon receiving the external verifier's message v, P^* views v as v_i, and selects (in a way to be determined below) the remaining $n-1$ messages $\vec{v}_{-i} = (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$ for the internal verifier $\mathsf{V}^{n,n}$. Then P^* runs P^{n*} to generate the response $\vec{p} = \mathsf{P}^{n*}(\vec{v})$, and sends p_i to V. To make V accept, the task of P^* is to find messages $\vec{v}_{-i}$ such that $\mathsf{P}^{n*}(v_i, \vec{v}_{-i})$ convinces V_i.

We can think of the above framework as a game played between P^* and V, where P^* takes the first move to select a coordinate $i \in [n]$, then V plays a random move v, and then P^* takes the last move to select messages $\vec{v}_{-i}$. In sum, the game consists of three moves $(i, v_i, \vec{v}_{-i})$.

For this simple setting of three-message public-coin protocols, the following naive reduction strategy works: P^* first selects the coordinate i uniformly at random, and upon receiving v from V, P^* randomly samples as many copies of $\vec{v}_{-i}$ as he can to find a good $\vec{v}_{-i}$ such that $\mathsf{P}^{n*}(v_i, \vec{v}_{-i})$ convinces V_i. If any such $\vec{v}_{-i}$ is found, P^* sends the corresponding p_i to V and succeeds. Otherwise, P^* simply gives up and fails. In other words, P^* selects a random first move i, and finds a good second move $\vec{v}_{-i}$ by sampling. A formal description of the strategy P^* can be found in Figure 3.1.

We need to show that P^* is a good strategy. Namely, we need to show that if P^{n*} succeeds with probability at least ε, then P^* can succeed with probability at least $\varepsilon^{1/n} \cdot (1 - \xi)$.

Recall that P^* uses sampling to find $\vec{v}_{-i}$. Sampling is a natural approach for P^* to exploit P^{n*} via block-box access. Indeed, the reduction prover strategies for the more general settings discussed later all consist of more involved sampling processes to find good prover messages from P^{n*}. However, since P^* needs to be efficient, P^* can only make a polynomially bounded number of samples and there are inevitable sampling errors.

It is instructive to consider an ideal version P^*_{ideal} of P^* where P^*_{ideal} can make unbounded number of samples (alternatively, do exhaustive search), and hence has no sampling errors. In fact, the analysis of the ideal version is usually a lot simpler and consists of clearer intuition. The hard part of the analysis usually lies in showing

Prover Strategy $\mathsf{P}^*(x, n, \varepsilon, \xi)$
/* P^* interacts with V and is given oracle access to P^{n*}. */

- P^* selects a coordinate $i \in_R [n]$ uniformly at random, runs P^{n*} to generate $\vec{w}$, and sends w_i to V.
- Upon receiving $v_i = v$ from V, P^* repeats the following at most $M = O\left(\frac{n}{\varepsilon\xi} \cdot \log \frac{1}{\varepsilon\xi}\right)$times.
  - P^* generates random messages $\vec{v}_{-i} = (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$. Then P^* checks if $\mathsf{P}^{n*}(v_i, \vec{v}_{-i})$ convinces V_i. Namely, P^* computes $\vec{p} = \mathsf{P}^{n*}(v_i, \vec{v}_{-i})$ and checks if $\mathsf{V}_i(w_i, v_i, p_i) = 1$ (i.e., the i-th subverifier accepts). If so, P^* sends p_i to V and terminates.
- If all the M attempts fail, P^* sends $\perp$ to V (or simply aborts).

Figure 3.1: Reduction prover strategy P^* for three-message public-coin protocols.

that the sampling error does not change the success probability too much. A formal description of P^*_{ideal} can be found in Figure 3.2.

3.1.1 Analysis of the Ideal Strategy P^*_{ideal}

We proceed to analyze the ideal strategy P^*_{ideal}. We shall show that if P^{n*} has success probability at least ε, then P^*_{ideal} has success probability at least $\varepsilon^{1/n}$. Note that no slackness ξ is needed for the ideal strategy.

Fix a coordinate $i \in [n]$ that P^*_{ideal} selects in his first move and consider V's message $v = v_i$. Observe that P^*_{ideal} can succeed on exactly the v_i's such that there exist messages $\vec{v}_{-i}$ such that $\mathsf{P}^{n*}(v_i, \vec{v}_{-i})$ convinces V_i. Let

$$\mathsf{Good}_i = \{v_i : \exists \vec{v}_{-i} \text{ s.t. } \mathsf{P}^{n*}(v_i, \vec{v}_{-i}) \text{ convinces } \mathsf{V}_i\}$$

We have

$$\Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V} \rangle(x) = 1] = \frac{1}{n} \sum_{i=1}^{n} \Pr[v_i \in \mathsf{Good}_i].$$

Also note that for P^{n*} to convince $\mathsf{V}^{n,n}$ on $\vec{v} = (v_1, \ldots, v_n)$, P^{n*} needs to convince every subverifier V_i. It follows by definition that for such $\vec{v}$, $v_i \in \mathsf{Good}_i$ for every $i \in [n]$. Hence, we have

$$\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1] \leq \Pr_{\vec{v}}[\forall i \in [n]\ v_i \in \mathsf{Good}_i] = \prod_{i=1}^{n} \Pr[v_i \in \mathsf{Good}_i].$$

Prover Strategy $\mathsf{P}^*_{ideal}(x, n, \varepsilon, \xi)$
/* P^*_{ideal} may *not* be efficient. */

- P^*_{ideal} selects a coordinate $i \in_R [n]$ uniformly at random, runs P^{n*} to generate $\vec{w}$, and sends w_i to V.
- Upon receiving $v_i = v$ from V, P^*_{ideal} repeats the following until seeing all possible $\vec{v}_{-i}$'s.
  - P^*_{ideal} generates random messages $\vec{v}_{-i} = (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$. Then P^*_{ideal} checks if $\mathsf{P}^{n*}(v_i, \vec{v}_{-i})$ convinces V_i. Namely, P^*_{ideal} computes $\vec{p} = \mathsf{P}^{n*}(v_i, \vec{v}_{-i})$ and checks if $\mathsf{V}_i(w_i, v_i, p_i) = 1$ (i.e., the i-th subverifier accepts). If so, P^*_{ideal} sends p_i to V and terminates.
- If no such $\vec{v}_{-i}$ exists, P^*_{ideal} sends $\perp$ to V (or simply aborts).

Figure 3.2: Ideal version P^*_{ideal} of P^* for three-message public-coin protocols.

Finally, putting it together and applying the Arithmetic-Mean-Geometric-Mean Inequality complete the analysis of the ideal strategy.

$$\begin{aligned}
\Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V} \rangle(x) = 1] &= \frac{1}{n} \sum_{i=1}^{n} \Pr[v_i \in \mathsf{Good}_i] \\
&\geq \left(\prod_{i=1}^{n} \Pr[v_i \in \mathsf{Good}_i] \right)^{1/n} \\
&\geq (\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1])^{1/n} \\
&\geq \varepsilon^{1/n}.
\end{aligned}$$

3.1.2 Analysis of the Prover Strategy P^*

We continue to analyze the actual (non-ideal) prover strategy P^*. We shall show that if P^{n*} has success probability at least ε, then P^* has success probability at least $\varepsilon^{1/n} \cdot (1 - \xi)$.

Again, fix a coordinate $i \in [n]$ that P^*_{ideal} selects in his first move and consider V's message $v = v_i$. In comparison to P^*_{ideal}, the issue for P^* is that if there are too few $\vec{v}_{-i}$ such that $\mathsf{P}^{n*}(v_i, \vec{v}_{-i})$ convinces V_i, P^* cannot find such $\vec{v}_{-i}$ by sampling. We need to show that this sampling error cannot decrease the success probability too much. We remark that we cannot show that the success probability of P^* is close to that of P^*_{ideal}. Indeed, it is possible that the success probability of P^*_{ideal} is 1, but the success probability of P^* remains small. Instead, we generalize the previous analysis for P^*_{ideal} to show a lower bound on the success probability of P^*.

As before, we define a good set Good_i of v_i such that P^* can convince V (with high probability) when P^* selects coordinate $i \in [n]$ and V's message is v_i. In Figure 3.1, the parameter M is set so that if $\Pr_{\vec{v}_{-i}}[\mathsf{P}^{n*}(v_i, \vec{v}_i) \text{ convinces } \mathsf{V}_i] \geq (\varepsilon\xi/2n)$, then P^* can find such $\vec{v}_{-i}$ with probability at least $1-(\varepsilon\xi/2)$. Indeed, the probability that P^* fails to find such a $\vec{v}_{-i}$ when $\Pr_{\vec{v}-i}[\mathsf{P}^{n*}(v_i, \vec{v}_i) \text{ convinces } \mathsf{V}_i] \geq (\varepsilon\xi/2n)$ is at most $(1-(\varepsilon\xi/2n))^M \leq (\varepsilon\xi/2)$ for properly chosen constant in $M = O\left(\frac{n}{\varepsilon\xi} \cdot \log \frac{1}{\varepsilon\xi}\right)$. Hence, for every $i \in [n]$, we define the good set Good_i as follows.

$$\mathsf{Good}_i = \left\{ v_i : \Pr_{\vec{v}_{-i}}[\mathsf{P}^{n*}(v_i, \vec{v}_i) \text{ convinces } \mathsf{V}_i] \geq (\varepsilon\xi/2n) \right\}.$$

It follows that

$$\Pr[\langle \mathsf{P}^*, \mathsf{V} \rangle(x) = 1] \geq \frac{1}{n} \sum_{i=1}^{n} \Pr[v_i \in \mathsf{Good}_i] \cdot \left(1 - \frac{\varepsilon\xi}{2}\right) \geq \frac{1}{n} \left(\sum_{i=1}^{n} \Pr[v_i \in \mathsf{Good}_i] \right) - \frac{\varepsilon\xi}{2}.$$

On the other hand, we can upper bound the success probability of P^{n*} as follows.

$$\begin{aligned}
&\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1] \\
&\quad \leq \Pr_{\vec{v}}[\forall i \; v_i \in \mathsf{Good}_i] + \Pr_{\vec{v}}[(\exists i \; v_i \notin \mathsf{Good}_i) \wedge (\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1)] \\
&\quad \leq \prod_{i=1}^{n} \Pr[v_i \in \mathsf{Good}_i] + \sum_{i=1}^{n} \Pr[(v_i \notin \mathsf{Good}_i) \wedge (\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1)] \\
&\quad \leq \prod_{i=1}^{n} \Pr[v_i \in \mathsf{Good}_i] + n \cdot \frac{\varepsilon\xi}{2n} \\
&\quad = \prod_{i=1}^{n} \Pr[v_i \in \mathsf{Good}_i] + \frac{\varepsilon\xi}{2},
\end{aligned}$$

where the third inequality follows by the fact that if $v_i \notin \mathsf{Good}_i$, then we have $\Pr_{\vec{v}_{-i}}[\mathsf{P}^{n*}(v_i, \vec{v}_i) \text{ convinces } \mathsf{V}_i] < (\varepsilon\xi/2n)$.

Again, putting it together and applying the Arithmetic-Mean-Geometric-Mean

Inequality complete the analysis of the prover strategy P^*.

$$\begin{aligned}
&\Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V}\rangle(x) = 1] \\
&\geq \frac{1}{n}\left(\sum_{i=1}^{n} \Pr[v_i \in \mathsf{Good}_i]\right) - \frac{\varepsilon\xi}{2} \\
&\geq \left(\prod_{i=1}^{n} \Pr[v_i \in \mathsf{Good}_i]\right)^{1/n} - \frac{\varepsilon\xi}{2} \\
&\geq \left(\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1] - \frac{\varepsilon\xi}{2}\right)^{1/n} - \frac{\varepsilon\xi}{2} \\
&\geq \left(\varepsilon \cdot \left(1 - \frac{\xi}{2}\right)\right)^{1/n} - \frac{\varepsilon\xi}{2} \\
&\geq \varepsilon^{1/n} \cdot (1 - \xi).
\end{aligned}$$

Comments on the reduction prover strategy. We remark that a few natural variants of the reduction prover strategy P^* also work for the simple three-message public-coin setting. First, in the above reduction, P^* selects a uniformly random coordinate $i \in [n]$ in which to embed V. It is natural to consider a strengthened $\tilde{\mathsf{P}}^*$ who tries find the best coordinate i. For example, $\tilde{\mathsf{P}}^*$ can use sampling to estimate the success probability of P^* when P^* selects coordinate i, and always select the best coordinate i in which to embed V. Such a $\tilde{\mathsf{P}}^*$ clearly does no worse than P^*. Although in this setting, we do not need the strengthening as P^* is already optimal, carefully selecting the coordinate $i \in [n]$ can be useful in other settings. We will discuss more on this in the next section.

On the other hand, it is very natural for P^* to find a $\vec{v}_{-i}$ such that $\mathsf{P}^{n*}(v_i, \vec{v}_{-i})$ convinces (only) V_i, as we presented above. However, it also works for P^* to find a $\vec{v}_{-i}$ such that $\mathsf{P}^{n*}(v_i, \vec{v}_{-i})$ convinces $\mathsf{V}^{n,n}$, i.e., *every* subverifier. It is not hard to check that exactly the same analysis (with "$\mathsf{P}^{n*}(v_i, \vec{v}_{-i})$ convinces V_i" replaced by "$\mathsf{P}^{n*}(v_i, \vec{v}_{-i})$ convinces $\mathsf{V}^{n,n}$") goes through. Both strategies can be viewed as "rejection sampling" strategies, where P^* keeps sampling and rejecting until certain desired outcomes ($\mathsf{P}^{n*}(v_i, \vec{v}_{-i})$ convinces $\mathsf{V}_i/\mathsf{V}^{n,n}$) happen. Again, this observation is useful in other settings we study later.

3.1.3 Discussion

As mentioned, the above proof demonstrates the common framework for proving efficient parallel repetition theorems for more general settings. In all settings, the starting point is a deterministic parallel prover strategy P^{n*}. All the reduction prover strategies P^* interact with V by simulating the parallel interaction $\langle \mathsf{P}^{n*}, \mathsf{V}^n\rangle$ internally with V playing one coordinate of V^n, which can be viewed as a game played

between P^* and V. The challenge is to design strategies to win the game with good probability, which requires designing clever sampling processes to exploit P^{n*} to find good responses. To analyze such sampling strategies, it is helpful to first consider the ideal version where there are no sampling errors, and then show that sampling errors do not lower the success probability too much.

The setting of direct product theorem for three-message public-coin protocols is simpler for a few reasons:

- Protocols consisting of only three messages are less interactive. The prover only needs to response to one verifier's challenge as opposed to committing an answer to the first challenge without knowing the future challenges. Furthermore, there is no issue for generating the parallel verifier's messages in simulating the parallel interaction. In contrast, the second message of the parallel verifier may be hard to generate for general private-coin protocols with more than three messages. For example, consider a protocol where the first message of V is an encryption $c = \mathsf{Enc}_k(m)$ of some message m under V's secret key k, and the second message is the underlying message m. Generating the second message of V based on his first message amounts to decrypt the ciphertext without the secret key k.

  In fact, the ability to simulate the external verfier's message by a prover is the key property for parallel repetition to decrease the soundness of a protocol. As mentioned in the introduction, we can prove parallel repetition theorems for protocols with "simulatable" verifiers [3, 20], and in contrast, under standard cryptographic assumptions, we know couterexamples of four-message protocols, where the prover cannot generate the verifier's second message, such that parallel repetition does not decrease the soundness error at all [1, 31].

- The public-coin verifier consists of no secrets, which allows the prover to compute the verifier's decision from the transcript. In contrast, for three-message private-coin protocols, the verifier's decision may depend on the private coins of the verifier, and so the decision is not computable by the prover. In this case, P^* needs to decide whether to forward P^{n*}'s message to V base on the decision of the $n-1$ internal subverifiers (where P^* knows the coins). In fact, the reduction prover strategy P^* for proving optimal parallel repetition theorem for three-message protocols is significantly different from the naive reduction presented in this section. In particular, P^* carefully finds a coordinate $i \in [n]$ in which to embed V, as opposed to a random coordinate used here.

  The public-coin property also allows the prover the simulate the verifier's messages easily even if the protocol has more than three rounds.

- The direct product verifier $\mathsf{V}^{n,n}$ accepts only when all subverifiers accept, which makes all coordinates symmetric. Note that even for the case of a threshold

verifier $\mathsf{V}^{n,k}$, the parallel prover P^{n*} may always convince the first k subverifiers, which breaks the symmetry among coordinates.

In particular, note that in the above reduction, it suffices for P^* to select a *random* coordinate $i \in [n]$ in which to embed V. To obtain an optimal parallel repetition theorem for the case of threshold verifiers $\mathsf{V}^{n,k}$, P^* cannot let V play a uniformly random coordinate of $\mathsf{V}^{n,k}$. For example, when $k = 1$, a parallel prover P^{n*} may always convince the first subverifier but fail on the remaining subverifiers. In this case, $\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,1} \rangle = 1] = 1$, but if a reduction prover strategy P^* selects a random coordinate $i \in [n]$ in which to embed V, then P^* can succeed with probability at most $1/n$. Nevertheless, when k is sufficiently large (i.e., the Chernoff-type case), letting V play a random coordinate is sufficient to obtain an asymptotically tight Chernoff-type theorem, although it is not exactly optimal even for $k = n - 1$.

Overall, proving parallel repetition theorems for different settings require subtly different sampling strategies. We know optimal strategies for several settings, but there are settings where the known strategies are known to be suboptimal. It would be interesting to prove tighter results for these settings by designing better prover strategies. We will discuss the limitations of the suboptimal strategies when we come to the corresponding settings in subsequent sections.

3.2 Efficient Direct Product Theorem for Public-Coin Protocols

In this section, we present a tight efficient direct product theorem for public-coin protocols (with an arbitrary number of rounds), which says that n-fold parallel repetition reduces soundness error from δ to $\delta^n + \mathsf{ngl}$ for any public-coin protocol. Again, this is proved by using a black-box reduction.

Our contribution is a tight analysis for a reduction strategy of Haståd, Pass, Pietrzak, and Wikström [19] showing that if a parallel prover strategy P^{n*} has success probability ε, then the reduction strategy can success with probability at least roughly $\varepsilon^{1/n}$. The original analysis of Haståd et al. only showed that n-fold parallel repetition reduces soundness error from $(1 - \alpha)$ to $e^{-\Omega\left((\alpha^2 n)/m\right)} + \mathsf{ngl}$, which has undesirable dependency on the number of rounds m. Independent of our work, Wikström also improved the bound of Haståd et al. [19] by generalizing their analysis. Wikström removed dependency on m and proved soundness error decreases from $(1 - \alpha)$ to $e^{-\Omega(\alpha^2 n)}$, which is still not tight in comparison to $(1 - \alpha)^n = e^{-\Omega(\alpha n)}$.[1]

We first state the theorem and introduce some notations, and discuss known reduction prover strategies and our contributions in the below section.

[1] We mention that [19] and [36] have been merged to a single paper [20].

Theorem 3.2 *Let* $\mathsf{V} \in \mathrm{PPT}$ *be a public-coin verifier. There exists a prover strategy* P^* *such that for every common input* $x \in \{0,1\}^*$*, every* $n \in \mathbb{N}$*, every* $\varepsilon, \xi \in (0,1)$*, and every parallel prover strategy* P^{n*}*,*

1. $\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1] \geq \varepsilon$

$$\Rightarrow \Pr[\langle \mathsf{P}^{*(\mathsf{P}^{n*})}(n, \varepsilon, \xi), \mathsf{V} \rangle(x) = 1] \geq \varepsilon^{1/n} \cdot (1 - \xi).$$

2. $\mathsf{P}^{*(\cdot)}(x, n, \varepsilon, \xi)$ *runs in time* $\mathrm{poly}(|x|, n, \varepsilon^{-1}, \xi^{-1})$ *given oracle access to* $\mathsf{P}^{n*}(x)$*.*

We assume without loss of generality that the verifier speaks first, and we denote the verifier V's (resp., the prover P's) messages by $v_1, \ldots, v_m$ (resp., $p_1, \ldots, p_m$), where m is the number of rounds of the protocol. The messages of the n-fold parallel repetition $\langle \mathsf{P}^n, \mathsf{V}^{n,n} \rangle$ of $\langle \mathsf{P}, \mathsf{V} \rangle$ are denoted by $\vec{v}_1 = (v_{1,1}, \ldots, v_{1,n}), \vec{v}_2, \ldots, \vec{v}_m$, and $\vec{p}_1, \ldots, \vec{p}_m$, respectively. Also, as mentioned, by Lemma 2.5, 2.6, we can assume without loss of generality that our starting point is a *deterministic* parallel prover strategy P^{n*} with success probability at least ε. Hence the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x)$ is determined solely by $\mathsf{V}^{n,n}$'s messages $(\vec{v}_1, \ldots, \vec{v}_m)$. For convenience, we ignore P^{n*}'s messages and say $(\vec{v}_1, \ldots, \vec{v}_m)$ is the transcript of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x)$, and refer to $(\vec{v}_1, \ldots, \vec{v}_j)$ and $(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})$ as a partial transcript of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x)$. We also write $\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1 | \vec{v}_1, \ldots, \vec{v}_j]$ as the success probability of P^{n*} conditioned on the partial transcript being $(\vec{v}_1, \ldots, \vec{v}_j)$. Similarly, $\Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_i | \vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i}]$ is the probability that P^{n*} convinces the first subverifier V_1 conditioning on partial transcript $(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})$.

3.2.1 Reduction Prover Strategies

We proceed to discuss the reduction prover strategies for public-coin protocols. Recall the common framework that the interaction of $\langle \mathsf{P}^*, \mathsf{V} \rangle$ simulates the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ as follows. At beginning, P^* first selects a coordinate $i \in [n]$ in which to embed V. Then in each round j, V randomly generate a message $v_j = v_{j,i}$ and the task of P^* is to select messages $\vec{v}_{j,-i}$ of V_{-i}. It is convenient to think of $\langle \mathsf{P}^*, \mathsf{V} \rangle$ as a game played between P^* and V, where P^*'s moves are $(i, \vec{v}_{1,-i}, \vec{v}_{2,-i}, \ldots, \vec{v}_{m,-i})$ and V's moves are $(v_{1,i}, v_{2,i}, \ldots, v_{m,i})$. In the game, V plays uniformly random strategy, and our goal is to find an optimal strategy for P^* such that P^* can succeed with probability at least roughly $\varepsilon^{1/n}$ whenever $\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1] \geq \varepsilon$.

Following the discussion in the end of the previous section, the challenge here is that the protocol has multiple rounds, so P^* needs to decide his move $\vec{v}_{j,-i}$ before seeing V's future moves $v_{j+1,i}, \ldots, v_{m,i}$. One natural idea is to let P^* to evaluate the game tree and play the optimal strategy. Namely, at each step, P^* selects the move that maximizes his success probability. It can be shown that the optimal strategy can indeed succeed with probability at least $\varepsilon^{1/n}$, but the problem is that both finding

the optimal move and evaluating the success probability of each node of the game tree may not efficient.

Pass and Venkitasubramaniam [30] observed that both tasks can be approximated by recursive sampling, which gives a somewhat efficient strategy with success probability at least roughly $\varepsilon^{1/n}$. However, due to the recursion, this strategy has complexity depending exponentially on the number of rounds m. Therefore, the strategy is only efficient for constant-round protocols. For protocols with a super-constant number of rounds, a different reduction strategy is required to prove the direct product theorem for public-coin protocols.

Haståd, Pass, Pietrzak, and Wikström [19] proved that the following surprisingly simple *rejection sampling* strategy P^*_{rej} works fairly well: At beginning, P^*_{rej} selects the coordinate $i \in [n]$ uniformly at random. Upon receiving V's message $v_{j,i}$ (i.e., after V plays move $v_{j,i}$), P^*_{rej} uses *rejection sampling* to find a *successful continuation* $(\vec{v}_{j,-i}, \vec{v}_{j+1}, \ldots, \vec{v}_m)$ of the current partial interaction $(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})$ such that P^{n*} convinces $\mathsf{V}^{n,n}$ on the corresponding interaction $(\vec{v}_1, \ldots, \vec{v}_m)$. Namely, P^*_{rej} keeps sampling, as many times as he can, uniform random continuation $(\vec{v}_{j,-i}, \vec{v}_{j+1}, \ldots, \vec{v}_m)$ and rejecting until one successful continuation is found. If such a successful continuation is found, P^*_{rej} selects the corresponding $\vec{v}_{j,-i}$ as his move; otherwise, P^*_{rej} simply gives up and aborts. A formal description of P^*_{rej} can be found in Figure 3.3.

Prover Strategy $\mathsf{P}^*_{rej}(x, n, \varepsilon, \xi)$
/* P^*_{rej} interacts with V and is given oracle access to P^{n*}. */
/* The interaction of $\langle \mathsf{P}^*_{rej}, \mathsf{V} \rangle$ simulates the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$. */

- P^*_{rej} selects a coordinate $i \in_R [n]$ uniformly at random.
- For each round $j \in [m]$, upon receiving $v_{j,i} = v_j$ from V, P^*_{rej} repeats the following at most $M = O\left(\frac{mn}{\varepsilon\xi}\right)$ times.
  - P^*_{rej} generates a random *continuation* $(\vec{v}_{j,-i}, \vec{v}_{j+1}, \ldots, \vec{v}_m)$ of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ and checks if it is a *successful continuation* in the sense that P^{n*} convinces $\mathsf{V}^{n,n}$ at the end of interaction.
  - If so, P^*_{rej} selects the corresponding $\vec{v}_{j,-i}$ as his move in this round, and runs P^{n*} on $(v_{j,i}, \vec{v}_{j,-i})$ to generate $\vec{p}_j$ and sends $p_{j,i}$ to V.
- If all the M attempts fail, P^* aborts.

Figure 3.3: Rejection sampling strategy P^*_{rej} for public-coin protocols.

Note that this strategy is a natural generalization of a variant of the reduction strategy for three-message public-coin protocols in Figure 3.1 where we conditioning on P^{n*} convincing $\mathsf{V}^{n,n}$ instead of only V_i. As discussed at the end of Section 3.1, both versions are good for the setting of three-message public-coin protocols. However, in

this setting of public-coin protocols, it seems to be important for P^*_{rej} to conditioning P^{n*} convincing $\mathsf{V}^{n,n}$ as opposed to only V_i, since in the analysis, we crucially use the fact that P^* conditions on *the same* event for different coordinate $i \in [n]$.

To illustrate why the rejection sampling strategy P^*_{rej} works, we give the following intuition from Haståd et al [19]. We first observe that P^*_{rej} playing the rejection sampling strategy is equivalent to P^*_{rej} sampling a uniformly random successful continuation $(\vec{v}_{j,-i}, \vec{v}_{j+1}, \ldots, \vec{v}_m)$ and then selecting the corresponding $\vec{v}_{j,-i}$ (Here, we ignore the sampling errors where P^*_{rej} fails to find a successful $(\vec{v}_{j,-i}, \vec{v}_{j+1}, \ldots, \vec{v}_m)$.) Let us consider a *mental experiment* $\langle \mathsf{P}^*_{rej}, \mathsf{V}^*_{rej} \rangle$ where the verifier V^*_{rej}, instead of playing a random strategy, also plays the rejection sampling strategy. Namely, V^*_{rej} also selects his move $v_{j,i}$ by first sampling a uniformly random successful continuation $(v_{j,i}, \vec{v}_{j,-i}, \vec{v}_{j+1}, \ldots, \vec{v}_m)$, and then selecting the corresponding $v_{j,i}$. It is not hard to see that the interaction of $\langle \mathsf{P}^*_{rej}, \mathsf{V}^*_{rej} \rangle$ amounts to P^*_{rej} and V^*_{rej} select a uniformly random successful transcript $(\vec{v}_1, \ldots, \vec{v}_m)$ jointly. Therefore, P^{n*} always convince $\mathsf{V}^{n,n}$ on the resulting transcript $(\vec{v}_1, \ldots, \vec{v}_m)$, which means P^*_{rej} succeeds with probability 1 in convincing V^*_{rej}. If the distribution of the actual interaction $\langle \mathsf{P}^*_{rej}, \mathsf{V} \rangle$ and that of the mental interaction $\langle \mathsf{P}^*_{rej}, \mathsf{V}^*_{rej} \rangle$ are statistically close, P^*_{rej} must also success with good probability in convincing the actual verifier V. This statistical closeness turns out to be the case, as we explain below.

Consider a fixed round $j \in [m]$. Note that in the mental interaction $\langle \mathsf{P}^*_{rej}, \mathsf{V}^*_{rej} \rangle$, both messages $v_{j,i}$ and $\vec{v}_{j,-i}$ are selected conditioning on successful interaction, while in the actual interaction, the messages $\vec{v}_{j,-i}$ are selected conditioning on successful interaction but the message $v_{j,i}$ is selected uniformly at random. The following lemma of Raz [32] says that for a product distribution $(v_{j,1}, \ldots, v_{j,n})$, conditioning on an event with noticeable probability cannot change the marginal distributions too much on average over coordinates $i \in [n]$.

Lemma 3.3 (Raz [32]) *Let $X_1, \ldots, X_n$ be* independent *random variables on a finite domain U. Let W an event defined on $\vec{X} = (X_1, \ldots, X_n)$. We have*

$$\frac{1}{n} \cdot \sum_{i=1}^{n} \Delta(X_i, X_i|_W) \leq \sqrt{\frac{1}{n} \cdot \log \frac{1}{\Pr[W]}}.$$

As an example, by Raz's Lemma, averaging over the coordinates $i \in [n]$, the statistical distance between the verifier's message $v_{j,i}$ played by V^*_{rej} and the uniform $v_{j,i}$ played by V is upper bounded by

$$\sqrt{\frac{1}{n} \cdot \log \frac{1}{\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1 | \vec{v}_1, \ldots, \vec{v}_{j-1}]}}.$$

The averaging makes sense since P^*_{rej} selects the coordinate i uniformly at random. Applying Raz's Lemma round by round with a hybrid argument, Haståd et al. showed

that if P^{n*} has success probability at least ε, then the statistical distance between the two interactions is at most $m \cdot \sqrt{(1/n) \cdot \log(1/\varepsilon)}$ and the success probability of P^*_{rej} is at least

$$1 - (m+1)\sqrt{\frac{1}{n} \cdot \log \frac{1}{\varepsilon}},$$

where m is the number of rounds. However. this bound is not tight in comparison to the optimal bound

$$\varepsilon^{1/n} \approx 1 - O\left(\frac{1}{n} \cdot \log \frac{1}{\varepsilon}\right).$$

Unfortunately, although the above analysis is quite intuitive and gives a lower bound on the success probability of P^*_{rej}, it cannot give a tight direct product theorem for public-coin protocols. The problem is that, in lower bounding the success probability of P^*_{rej}, the above analysis loses an additive factor of the statistical distance between the two interactions. However, we cannot afford even a single application of the Raz's Lemma. Suppose that the success probability of P^{n*} is ε, then the bound given by the Raz's Lemma is

$$\sqrt{\frac{1}{n} \cdot \log \frac{1}{\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1]}} = \sqrt{\frac{1}{n} \cdot \log \frac{1}{\varepsilon}}.$$

This loss is already too large since the optimal bound only allows us to loss a factor of $O((1/n) \cdot \log(1/\varepsilon))$.

Furthermore, the Raz's Lemma itself is tight and we can embed a tight example of Raz's Lemma below into the interaction of P^{n*} and $\mathsf{V}^{n,n}$. This means that the statistical distance between the actual interaction and the mental interaction can indeed be

$$\Omega\left(\sqrt{\frac{1}{n} \cdot \log \frac{1}{\varepsilon}}\right) > O\left(\frac{1}{n} \cdot \log \frac{1}{\varepsilon}\right).$$

Therefore, to obtain a tight direct product theorem, we cannot afford to move from the actual interaction to the mental interaction.

A Tight Example. Let $X_1, \ldots, X_n$ be i.i.d. binary random variables with $\Pr[X_i = 1] = 1/2$, and W be the event that the average of the X_i's is large:

$$\frac{1}{n} \cdot \sum_i X_i \geq \frac{1}{2} + \alpha.$$

By standard Chernoff bounds, we know that $\Pr[W] \leq e^{-\Omega(\alpha^2 n)}$. Also, by symmetry among the coordinates, we know that for every $i \in [n]$,

$$\Pr[X_i = 1 | W] \geq \frac{1}{2} + \alpha.$$

In this case, the Raz's Lemma is tight:

$$\frac{1}{n} \cdot \sum_{i=1}^{n} \Delta(X_i, X_i|_W) \geq \alpha = \Omega\left(\sqrt{\frac{1}{n} \cdot \log \frac{1}{\Pr[W]}}\right).$$

We now consider a two-message public-coin protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ and its n-fold parallel repetition $\langle \mathsf{P}^n, \mathsf{V}^{n,n} \rangle$. Let P^{n*} be an (artificial) parallel prover such that P^{n*} convinces $\mathsf{V}^{n,n}$ iff the sum of the first bits of $v_{1,1}, \ldots, v_{1,n}$ is greater than $(1/2 + \alpha) \cdot n$. It is not hard to see that the average (over $i \in [n]$) statistical distance between the $v_{1,i}$ played by V^*_{rej} and the uniform $v_{1,i}$ played by V is exactly the average statistical distance of $X_i|_W$ and X_i, which is

$$\Omega\left(\sqrt{\frac{1}{n} \cdot \log \frac{1}{\varepsilon}}\right) > O\left(\frac{1}{n} \cdot \log \frac{1}{\varepsilon}\right).$$

■

3.2.2 Our Tight Analysis to the Rejection Sampling Strategy

Our contribution is a tight analysis to the rejection sampling strategy P^*_{rej}. To obtain a tight analysis, we analyze the success probability of $\langle \mathsf{P}^*_{rej}, \mathsf{V} \rangle$ directly and avoid using any form of Raz's Lemma. We lower bound the success probability of P^*_{rej} by a careful induction on the number of rounds regarding the *geometric mean* of P^*_{rej}'s success probabilities over the choice of coordinate i. Again, we start with analyzing an ideal version P^*_{ideal} of P^*_{rej}, where there are no sampling errors, and then generalize the analysis to analyze the actual strategy P^*_{rej}. As in the warm-up setting of three-message public-coin protocols in Section 3.1, P^*_{ideal} is the same as P^*_{rej} except that P^*_{ideal} can sample a unbounded number of (instead of M) random continuations $(\vec{v}_{j,-i}, \vec{v}_{j+1}, \ldots, \vec{v}_m)$ until seeing all possible continuations, and hence P^*_{ideal} can find a random successful continuation as long as there exists one. A formal description of P^*_{ideal} can be found in Figure 3.4.

Analysis of the Ideal Strategy P^*_{ideal}

We shall show that if P^{n*} has success probability at least ε, then P^*_{ideal} can succeed with probability at least $\varepsilon^{1/n}$. Note that no slackness ξ is needed for the ideal strategy.

As mentioned, we will lower bound the success probability of P^*_{ideal} by an induction on the number of rounds regarding the *geometric mean* of P^*_{ideal}'s success probabilities over the choice of coordinate i. Recall that we use $\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1 | \vec{v}_1, \ldots, \vec{v}_j]$ to denote the success probability of P^{n*} conditioning on the partial interaction $(\vec{v}_1, \ldots, \vec{v}_j)$. Similarly, we use $\Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V} \rangle(x) | i, \vec{v}_1, \ldots, \vec{v}_j]$ to denote the conditional success probability of P^*_{ideal} when the partial moves of P^*_{ideal} and V are

Prover Strategy $\mathsf{P}^*_{ideal}(x, n, \varepsilon, \xi)$
/* P^*_{ideal} may *not* be efficient. */
/* The interaction of $\langle \mathsf{P}^*_{ideal}, \mathsf{V} \rangle$ simulates the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$. */

- P^*_{ideal} selects a coordinate $i \in_R [n]$ uniformly at random.
- For each round $j \in [m]$, upon receiving $v_{j,i} = v_j$ from V, P^*_{ideal} repeats the following until seeing all possible continuations $(\vec{v}_{j,-i}, \vec{v}_{j+1}, \ldots, \vec{v}_m)$.
  - P^*_{ideal} generates a random continuation $(\vec{v}_{j,-i}, \vec{v}_{j+1}, \ldots, \vec{v}_m)$ and checks if it is a successful continuation. Namely P^*_{ideal} checks if P^{n*} convinces $\mathsf{V}^{n,n}$ on interaction $(\vec{v}_1, \ldots, \vec{v}_m)$. If so, P^*_{ideal} selects the corresponding $\vec{v}_{j,-i}$ as his move in this round, and runs P^{n*} on $(v_{j,i}, \vec{v}_{j,-i})$ to generate $\vec{p}_j$ and sends $p_{j,i}$ to V.
- If no successful continuation exists, P^*_{ideal} aborts.

Figure 3.4: Ideal version P^*_{ideal} of P^*_{rej} for public-coin protocols.

$(i; \vec{v}_1, \ldots, \vec{v}_j)$. Namely, P^*_{ideal} selects the coordinate i in which to embed V, V's messages are $v_{1,i}, \ldots, v_{j,i}$, and P^*_{ideal}'s corresponding moves are $\vec{v}_{1,-i}, \ldots, \vec{v}_{j,-i}$. With these notations, our induction hypothesis can be stated as follows.

Induction Hypothesis for P^*_{ideal}. For every $j \in [m]$ and every partial interaction $(\vec{v}_1, \ldots, \vec{v}_j)$, the following inequality holds.

$$\prod_{i=1}^{n} \Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V} \rangle(x) = 1 | i, \vec{v}_1, \ldots, \vec{v}_j] \geq \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1 | \vec{v}_1, \ldots, \vec{v}_j].$$

In words, it says that for every partial interaction $(\vec{v}_1, \ldots, \vec{v}_j)$, the product of the success probability of P^*_{ideal} conditioning on V playing i and partial interaction $(\vec{v}_1, \ldots, \vec{v}_j)$ is at least the success probability of P^{n*} conditioning on partial interaction $(\vec{v}_1, \ldots, \vec{v}_j)$. We remark that the same induction hypothesis is already used in Pass and Venkitasubramaniam [30] to prove their tight direct production theorem for constant-round public-coin protocols.

We shall prove the induction backward on $j = m, m-1, \ldots, 1, 0$ and apply the induction hypothesis with $j = 0$ (i.e., without conditioning) to complete the proof. Let us first see how the induction hypothesis is applied.

Induction Hypothesis for $j = 0 \Rightarrow \Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V} \rangle(x) = 1] \geq \varepsilon^{1/n}$. When $j = 0$, the induction hypothesis says that

$$\prod_{i=1}^{n} \Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V} \rangle(x) = 1 | i] \geq \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1].$$

Recalling that P^*_{ideal} selects the coordinate i uniformly at random, we have

$$\Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V}\rangle(x) = 1] = \frac{1}{n}\sum_{i=1}^{n} \Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V}\rangle(x) = 1|i].$$

Similar to the setting of three-message public-coin protocols in the previous section, putting them together by applying the Arithmetic-Mean-Geometric-Mean Inequality gives the desired lower bound on the success probability of P^*_{ideal}.

$$\begin{aligned}
\Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V}\rangle(x) = 1] &= \frac{1}{n}\sum_{i=1}^{n} \Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V}\rangle(x) = 1|i] \\
&\geq \left(\prod_{i=1}^{n} \Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V}\rangle(x) = 1|i]\right)^{1/n} \\
&\geq \left(\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1]\right)^{1/n} \\
&\geq \varepsilon^{1/n}.
\end{aligned}$$

■

It is also easy to verify the base case $j = m$ of the induction, where we conditioning on a *complete* transcript $(\vec{v}_1, \ldots, \vec{v}_m)$. Indeed, when we conditioning on a complete transcript, there is no randomness and the probabilities are simply 0 or 1 and the inequality is trivial to verify.

Base Case of the Induction. For every complete transcript $(\vec{v}_1, \ldots, \vec{v}_m)$, we have

$$\prod_{i=1}^{n} \Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V}\rangle(x) = 1|i, \vec{v}_1, \ldots, \vec{v}_m] \geq \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1|\vec{v}_1, \ldots, \vec{v}_m].$$

By inspection, the LHS is 1 iff P^{n*} convinces V_i for every $i \in [n]$ on interaction $(\vec{v}_1, \ldots, \vec{v}_m)$, which is equivalent to P^{n*} convinces $\mathsf{V}^{n,n}$ on $(\vec{v}_1, \ldots, \vec{v}_m)$, which is the case iff RHS is 1. ■

Finally, the remaining and challenging part is to prove the following induction step. We first introduce some shorthand notations below to simplify the expressions later in the analysis, and then use the new notations to state and prove the induction step. We will use $\bar{h} = (\vec{v}_1, \ldots, \vec{v}_j)$ to denote a partial interaction and define

$$\begin{aligned}
\gamma(\bar{h}) = \gamma(\vec{v}_1, \ldots, \vec{v}_j) &\stackrel{\text{def}}{=} \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1|\vec{v}_1, \ldots, \vec{v}_j], \text{ and} \\
\eta_i(\bar{h}) = \eta_i(\vec{v}_1, \ldots, \vec{v}_j) &\stackrel{\text{def}}{=} \Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V}\rangle(x) = 1|i, \vec{v}_1, \ldots, \vec{v}_j] \text{ for every } i \in [n].
\end{aligned}$$

Induction Step. For every $j \in [m]$ and every partial interaction $\bar{h} = (\vec{v}_1, \ldots, \vec{v}_{j-1})$, the following holds. Suppose that for every message $\vec{v}_j$, it is true that

$$\prod_{i=1}^{n} \eta_i(\bar{h}, \vec{v}_j) \geq \gamma(\bar{h}, \vec{v}_j).$$

Then we have

$$\prod_{i=1}^{n} \eta_i(\bar{h}) \geq \gamma(\bar{h}).$$

It should be clear that the above induction step together with the base case prove the induction. We proceed to prove the induction step. The first step is to express the probabilities $\gamma(\bar{h})$ and $\eta_i(\bar{h})$ in terms of $\gamma(\bar{h}, \vec{v}_j)$ and $\eta_i(\bar{h}, \vec{v}_j)$. It is easy to see by definition that

$$\gamma(\bar{h}) = \mathop{\mathrm{E}}_{\vec{v}_j}\left[\gamma(\bar{h}, \vec{v}_j)\right] \text{ and } \gamma(\bar{h}, v_{j,i}) = \mathop{\mathrm{E}}_{\vec{v}_{j,-i}}\left[\gamma(\bar{h}, \vec{v}_j)\right].$$

For $\eta_i(\bar{h})$, we state the following claim.

Claim 3.4 *For every $i \in [n]$, $j \in [m]$, and partial interaction $\bar{h} = (\vec{v}_1, \ldots, \vec{v}_{j-1})$,*[2]

$$\eta_i(\bar{h}) \;=\; \mathop{\mathrm{E}}_{\vec{v}_j}\left[\frac{\gamma(\bar{h}, \vec{v}_j) \cdot \eta_i(\bar{h}, \vec{v}_j)}{\gamma(\bar{h}, v_{j,i})}\right].$$

Proof of claim: Recall that V plays the random strategy and P^*_{ideal} plays the rejection sampling strategy. Let $\Pr[v_{j,i}], \Pr[\vec{v}_{j,-i}]$ denote the uniform probability on $v_{j,i}, \vec{v}_{j,-i}$, respectively. For a given $\vec{v}_j = (v_{j,i}, \vec{v}_{j,-i})$, observe that V plays $v_{j,i}$ with probability $\Pr[v_{j,i}]$. By Bayes Rule, P^*_{ideal} plays $\vec{v}_{j,-i}$ with probability

$$\begin{aligned}
&\Pr[\vec{v}_{j,-i} | \bar{h}, v_{j,i}, \langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1] \\
&= \frac{\Pr[\vec{v}_{j,-i}] \cdot \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1 | \bar{h}, \vec{v}_j]}{\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1 | \bar{h}, v_{j,i}]} \\
&= \Pr[\vec{v}_{j,-i}] \cdot \frac{\gamma(\bar{h}, \vec{v}_j)}{\gamma(\bar{h}, v_{j,i})},
\end{aligned}$$

and by definition, P^*_{ideal} can succeed with probability $\eta_i(\bar{h}, \vec{v}_j)$. Hence, we have

$$\begin{aligned}
\eta_i(\bar{h}) &= \sum_{\vec{v}j} \Pr[v_{j,i}] \cdot \Pr[\vec{v}_{j,-i}] \cdot \frac{\gamma(\bar{h}, \vec{v}_j)}{\gamma(\bar{h}, v_{j,i})} \cdot \eta_i(\bar{h}, \vec{v}_j) \\
&= \mathop{\mathrm{E}}_{\vec{v}_j}\left[\frac{\gamma(\bar{h}, \vec{v}_j) \cdot \eta_i(\bar{h}, \vec{v}_j)}{\gamma(\bar{h}, v_{j,i})}\right]
\end{aligned}$$

□

[2] We use a convention that $0/0 = 0$.

Using the above formulas, our goal is to show that

$$\prod_{i=1}^{n} \eta_i(\bar{h}) = \prod_{i=1}^{n} \mathop{\mathrm{E}}_{\vec{v}_j} \left[\frac{\gamma(\bar{h}, \vec{v}_j) \cdot \eta_i(\bar{h}, \vec{v}_j)}{\gamma(\bar{h}, v_{j,i})} \right] \geq \gamma(\bar{h}).$$

This can be done by applying Hölder's Inequality twice. We first recall Hölder's Inequality, and then prove the induction step by a lemma below.

Lemma 3.5 (Hölder's Inequality[8]) *Let $F_1, \ldots, F_n$ be non-negative functions from a finite domain Ω to $\mathbb{R}$, and $a_1, \ldots a_n > 0$ satisfying $1/a_1 + \ldots 1/a_n = 1$. Let q be a uniform random variable over Ω. We have*

$$\mathop{\mathrm{E}}_{q}[F_1(q) \cdots F_n(q)] \leq \mathop{\mathrm{E}}_{q}[F_1(q)^{a_1}]^{1/a_1} \cdots \mathop{\mathrm{E}}_{q}[F_n(q)^{a_n}]^{1/a_n}.$$

Lemma 3.6 (Induction Step) *Let $\gamma, \eta_1, \ldots, \eta_n : \Omega^n \to [0,1]$ be $[0,1]$-valued functions over a product space Ω^n such that $\prod_i \eta_i(\vec{q}) \geq \gamma(\vec{q})$ for every $\vec{q} = (q_1, \ldots, q_n) \in \Omega^n$. Let $\gamma = \mathrm{E}_{\vec{q}}[\gamma(\vec{q})]$. For every $i \in [n]$ and $q_i \in \Omega$, let*

$$\gamma(q_i) = \mathop{\mathrm{E}}_{\vec{q}_{-i}}[\gamma(\vec{q})] \quad \text{and} \quad \eta_i = \mathop{\mathrm{E}}_{\vec{q}} \left[\frac{\gamma(\vec{q}) \cdot \eta_i(\vec{q})}{\gamma(q_i)} \right],$$

where the above expectation is over uniform distribution over Ω^n. We have

$$\prod_{i=1}^{n} \eta_i = \prod_{i=1}^{n} \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{\gamma(\vec{q}) \cdot \eta_i(\vec{q})}{\gamma(q_i)} \right) \right] \geq \gamma.$$

Proof. We apply Hölder's Inequality twice to prove the lemma, as presented in the calculation below. Informally, the first application of Hölder's Inequality moves the product operator to inside the expectation so that we can apply the induction hypothesis, and the second application of Hölder's Inequality moves the expectation operator inside again so that we can simplify the terms. We present the whole computation first, and then explain how Hölder's Inequality is applied in each step.

$$\begin{aligned}
&\prod_{i=1}^{n} \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{\gamma(\vec{q}) \cdot \eta_i(\vec{q})}{\gamma(q_i)} \right) \right] \\
&\quad \geq \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{\gamma(\vec{q})^n \cdot \prod_{i=1}^{n} \eta_i(\vec{q})}{\prod_{i=1}^{n} \gamma(q_i)} \right)^{1/n} \right]^n \qquad \text{(by Hölder's Inequality)} \\
&\quad \geq \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{\gamma(\vec{q})^{n+1}}{\prod_{i=1}^{n} \gamma(q_i)} \right)^{1/n} \right]^n \qquad \text{(by induction hypothesis)} \\
&\quad \geq \left[\left(\frac{\mathrm{E}_{\vec{q}}[\gamma(\vec{q})]^{n+1}}{\mathrm{E}_{\vec{q}}[\prod_{i=1}^{n} \gamma(q_i)]} \right)^{1/n} \right]^n \qquad \text{(by Hölder's Inequality)} \\
&\quad = (\gamma^{n+1}/\gamma^n) = \gamma.
\end{aligned}$$

We now explain the application of Hölder's Inequalities.

- The first inequality uses $\mathrm{E}[X_1^n]^{1/n} \cdot \cdots \cdot \mathrm{E}[X_n^n]^{1/n} \geq \mathrm{E}[X_1 \cdot \cdots \cdot X_n]$ with

$$X_i = \left(\frac{\gamma(\vec{q}) \cdot \eta_i(\vec{q})}{\gamma(q_i)}\right)^{1/n}.$$

- The third inequality uses $\mathrm{E}\left[B^{n+1}\right]^{1/(n+1)} \cdot \mathrm{E}\left[(A/B)^{(n+1)/n}\right]^{n/(n+1)} \geq \mathrm{E}[A]$, or equivalently,

$$\mathrm{E}\left[\left(\frac{A^{n+1}}{B^{n+1}}\right)^{1/n}\right] \geq \left(\frac{\mathrm{E}[A]^{n+1}}{\mathrm{E}[B^{n+1}]}\right)^{1/n}$$

with

$$\begin{cases} A = \gamma(\vec{q}), \\ B^{n+1} = \prod_{i=1}^{n} \gamma(q_i). \end{cases}$$

■

Remark 3.7 One might worry about the legitimacy of the manipulation when the denominators are zeros. One way to justify it is by adding some $\mu > 0$ in the denominators before the manipulation. Formally, we have

$$\prod_{i=1}^{n} \mathop{\mathrm{E}}_{\vec{q}}\left[\left(\frac{\gamma(\vec{q}) \cdot \eta_i(\vec{q})}{\gamma(q_i)}\right)\right] \geq \prod_{i=1}^{n} \mathop{\mathrm{E}}_{\vec{q}}\left[\left(\frac{\gamma(\vec{q}) \cdot \eta_i(\vec{q})}{\gamma(q_i)+\mu}\right)\right] \geq \cdots \geq (\gamma^{n+1}/(\gamma+\mu)^n),$$

which is valid for arbitrary $\mu > 0$. Taking $\mu \to 0$, we obtain the desired result.

It is easy to verify that the probabilities $\gamma(\bar{h}, \cdot)$ and $\eta_i(\bar{h}, \cdot)$ satisfy the premise of the above lemma. Therefore, a straightforward application of the above lemma completes the analysis of the induction step and hence completes the analysis of P^*_{ideal}.

Analysis of the Rejection Sampling Strategy P^*_{rej}

We proceed to analyze to actual (non-ideal) rejection sampling strategy P^*_{rej} where there are sampling errors. Namely, P^*_{rej} may abort due to the failure of finding a successful continuation in M trials when there are too few successful continuations. The challenge is to show that the sampling errors do not lower the success probability to much. We remark again (as in Section 3.1.2) that we cannot show that the success probability of P^* is close to that of P^*_{ideal}, since it is possible that the success probability of P^*_{ideal} is 1, but the success probability of P^* remains small. We have to lower bound the success probability of P^*_{rej} directly.

We will generalize our inductive hypothesis for P^*_{ideal} to accommodate the sampling errors. Before we proceed, we first give some intuition about why a polynomially many number of samples of random continuations is sufficient for P^*_{rej} to find a successful continuation $(\vec{v}_{j,-i}, \vec{v}_{j+1}, \ldots, \vec{v}_m)$ with high probability.

For intuition, let us investigate the mental interaction $\langle \mathsf{P}^*_{rej}, \mathsf{V}^*_{rej} \rangle$ instead, and consider the first step of both the rejection sampling of V^*_{rej} and P^*_{rej}. In the first step, V^*_{rej} uses rejection sampling to find a successful interaction $(\vec{v}_1, \ldots, \vec{v}_m)$ and selects the corresponding $v_{1,i}$. Here, since the success probability of P^{n*} is at least ε, in expectation, only $1/\varepsilon$ samples is needed to find a successful interaction. Then, given $v_{1,i}$ selected by V^*_{rej}, P^*_{rej} samples random continuations $(\vec{v}_{1,-i}, \vec{v}_2, dots, \vec{v}_m)$ to find a successful one. Although for some $v_{1,i}$'s, the success probability $\gamma(v_{1,i}) = \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1 | v_{1,i}]$ can be very small so that a successful continuation is hard to find. Since V^*_{rej} selects $v_{1,i}$ with probability proportional to $\gamma(v_{1,i})$, such a $v_{1,i}$ is selected by V^*_{rej} with very small probability as well. Intuitively, for most $v_{1,i}$ selected by V^*_{rej}, P^*_{rej} can find a successful continuation by a reasonably small number of samples.

Indeed, in the mental interaction, one can show (as proved in Haståd et al. [19]) that in every round j, the expected number of samples for P^*_{rej} to find a successful continuation $(\vec{v}_{j,-i}, \vec{v}_{j+1}, \ldots, \vec{v}_m)$ is $1/\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1]$, where the expectation is over the random partial interaction $(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})$ selected so far and the randomness of rejection sampling process. It follows by a Markov type argument that a sufficiently larger than $1/\varepsilon$ number of samples is enough for P^*_{rej} to find a successful continuation with high probability (in each step) over the interaction of $\langle \mathsf{P}^*_{rej}, \mathsf{V}^*_{rej} \rangle$. Since the two interactions $\langle \mathsf{P}^*_{rej}, \mathsf{V} \rangle$ and $\langle \mathsf{P}^*_{rej}, \mathsf{V}^*_{rej} \rangle$ are statistically close, the same number of samples are enough for P^*_{rej} to interact with V as well.

However, again, we cannot afford to move to the mental interaction since the statistical distance between the two interactions is not small enough to prove a tight direct product theorem. We proceed to generalize the inductive analysis in the previous subsection to analyze P^*_{rej}. We shall show that if P^{n*} has success probability at least ε, then P^* can succeed with probability at least $\varepsilon^{1/n} \cdot (1 - \xi)$. We start by recapping the key steps of the analysis of P^*_{ideal}.

Recall that we analyzed the ideal strategy P^*_{ideal} by induction with the following induction hypothesis: For every $j \in [m]$ and every partial interaction $\bar{h} = (\vec{v}_1, \ldots, \vec{v}_j)$, we have

$$\prod_{i=1}^{n} \eta_i(\bar{h}) \geq \gamma(\bar{h}),$$

where $\gamma(\bar{h}) = \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1 | \vec{v}_1, \ldots, \vec{v}_j]$, and for every $i \in [n]$, $\eta_i(\bar{h}) = \Pr[\langle \mathsf{P}^*_{ideal}, \mathsf{V} \rangle(x) = 1 | i, \vec{v}_1, \ldots, \vec{v}_j]$. Then, in the induction step, we show that for every partial interaction $\bar{h} = (\vec{v}_1, \ldots, \vec{v}_{j-1})$,

$$\prod_{i=1}^{n} \eta_i(\bar{h}) = \prod_{i=1}^{n} \mathop{\mathrm{E}}_{\vec{v}_j} \left[\frac{\gamma(\bar{h}, \vec{v}_j) \cdot \eta_i(\bar{h}, \vec{v}_j)}{\gamma(\bar{h}, v_{j,i})} \right] \geq \gamma(\bar{h}),$$

provided that $\prod_{i=1}^{n} \eta_i(\bar{h}, \vec{v}_j) \geq \gamma(\bar{h}, \vec{v}_j)$ for every $\vec{v}_j$.

To analyze P^*_{rej}, we redefine

$$\eta_i(\bar{h}) = \eta_i(\vec{v}_1, \ldots, \vec{v}_j) \stackrel{\text{def}}{=} \Pr[\langle \mathsf{P}^*_{rej}, \mathsf{V} \rangle(x) = 1 | i, \vec{v}_1, \ldots, \vec{v}_j] \text{ for every } i \in [n].$$

The issue with P^*_{rej} is that due to the sampling errors,

$$\eta_i(\bar{h}) \neq \mathop{\mathrm{E}}_{\vec{v}_j}\left[\frac{\gamma(\bar{h}, \vec{v}_j) \cdot \eta_i(\bar{h}, \vec{v}_j)}{\gamma(\bar{h}, v_{j,i})}\right].$$

Instead, it becomes a more complicated formula below.

Claim 3.8 *For every $i \in [n]$, $j \in [m]$, and partial transcript $\bar{h} = (\vec{v}_1, \ldots, \vec{v}_{j-1})$, we have*

$$\eta_i(\bar{h}) = \mathop{\mathrm{E}}_{\vec{v}_j}\left[\frac{\gamma(\bar{h}, \vec{v}_j) \cdot \eta_i(\bar{h}, \vec{v}_j)}{\gamma(\bar{h}, v_{j,i})} \cdot f(\gamma(\bar{h}, v_{j,i}))\right],$$

where $f(\alpha) = (1 - (1 - \alpha)^M)$, and $M = O\left(\frac{mn}{\varepsilon\xi}\right)$ is the number of samples specified in the strategy of P^*_{rej} in Figure 3.3.

> **Proof of claim:** Observing that P^*_{rej} can find a successful continuation with probability exactly $f(\gamma(\bar{h}, v_{j,i}))$, and that conditioning on a successful continuation is found, P^*_{rej} plays $\vec{v}_{j,-i}$ with the same probability as P^*_{ideal}, we obtain the above formula for η_i. □

Fortunately, we can still prove an induction with a variant induction hypothesis to accommodate the "error term" $f(\gamma(\bar{h}, v_{j,i}))$.

Induction Hypothesis for P^*_{ref}. For every $j \in [m]$ and every partial interaction $(\vec{v}_1, \ldots, \vec{v}_j)$, the following inequality holds.

$$\prod_{i=1}^{n} \eta_i(\bar{h}) \geq \left(\frac{(\gamma(\bar{h}) - (m-j) \cdot \nu)_+^{n+1}}{(\gamma(\bar{h}) + \nu)^n}\right),$$

where $(\alpha)_+ \stackrel{\text{def}}{=} \max\{\alpha, 0\}$ and $\nu = 1/M$.

Observe that as $M \to 0$ (i.e., $\nu \to 0$), our induction hypothesis is the same as before. Compared to the induction hypothesis for P^*_{ideal}, we add a copy of $\gamma(\bar{h})$ in numerator and denominator, and further add certain slackness in $\nu = 1/M$ in both the numerator and the denominator, where the slackness in the numerator grows round by round backwardly to accommodate the sampling errors.

As before, we show how the induction hypothesis implies a lower bound on the success probability of P^*_{rej} first.

Induction Hypothesis for $j = 0 \to \Pr[\langle \mathsf{P}^*_{rej}, \mathsf{V}\rangle(x) = 1] \geq \varepsilon^{1/n} \cdot (1 - \xi)$. When $j = 0$, the induction hypothesis says that

$$\prod_{i=1}^{n} \Pr[\langle \mathsf{P}^*_{rej}, \mathsf{V}\rangle(x) = 1|i] \geq \frac{(\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1] - m \cdot \nu)_+^{n+1}}{(\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1] + \nu)^n}.$$

Applying the Arithmetic-Mean-Geometric-Mean Inequality in the same way as before with a bit more complicated calculation gives the desired lower bound on the success probability of P^*_{rej}.

$$\begin{aligned}
&\Pr[\langle \mathsf{P}^*_{rej}, \mathsf{V}\rangle(x) = 1] \\
&= \frac{1}{n}\sum_{i=1}^{n} \Pr[\langle \mathsf{P}^*_{rej}, \mathsf{V}\rangle(x) = 1|i] \\
&\geq \left(\prod_{i=1}^{n} \Pr[\langle \mathsf{P}^*_{rej}, \mathsf{V}\rangle(x) = 1|i]\right)^{1/n} \\
&\geq \left(\frac{(\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1] - m \cdot \nu)_+^{n+1}}{(\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1] + \nu)^n}\right)^{1/n} \\
&\geq (\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1])^{1/n} \cdot \left(1 - O\left(\frac{mn\nu}{\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1]}\right)\right) \\
&\geq \varepsilon^{1/n} \cdot (1 - \xi).
\end{aligned}$$

■

We proceed to prove the induction. Again, the base case $j = m$ is trivial to check. Conditioning on a complete transcript $\bar{h} = (\vec{v}_1, \ldots, \vec{v}_m)$, the probabilities $\gamma(\bar{h})$ and $\eta_i(\bar{h})$ are simply 0 or 1, and $\gamma(\bar{h}) = 1$ iff $\eta_i(\bar{h}) = 1$ for every $i \in [n]$, which implies that the above inequality holds for the base case. The complicated part is the induction step.

Induction Step. For every $j \in [m]$ and every partial interaction $\bar{h} = (\vec{v}_1, \ldots, \vec{v}_{j-1})$, the following holds. Suppose for all messages $\vec{v}_j$, it is true that

$$\prod_{i=1}^{n} \eta_i(\bar{h}, \vec{v}_j) \geq \left(\frac{(\gamma(\bar{h}, \vec{v}_j) - (m - j) \cdot \nu)_+^{n+1}}{(\gamma(\bar{h}, \vec{v}_j) + \nu)^n}\right),$$

Then we have

$$\prod_{i=1}^{n} \eta_i(\bar{h}) \geq \left(\frac{(\gamma(\bar{h}) - (m - (j-1)) \cdot \nu)_+^{n+1}}{(\gamma(\bar{h}) + \nu)^n}\right).$$

As before, the induction step can be proved by applying Hölder's Inequality twice in the same way, but with a more careful analysis on the error terms. We prove the induction step by the following lemma.

Lemma 3.9 *Let $\nu \in (0,1)$ and $t, M \geq 0$ such that $M \cdot \nu \geq 1$. Let $\gamma, \eta_1, \ldots, \eta_n : \Omega^n \to [0,1]$ be functions over Ω^n such that*

$$\prod_i \eta_i(\vec{q}) \geq \left(\frac{(\gamma(\vec{q}) - t \cdot \nu)_+^{n+1}}{(\gamma(\vec{q}) + \nu)^n} \right)$$

for every $\vec{q} = (q_1, \ldots, q_n) \in \Omega^n$. Let $\gamma = \mathrm{E}_{\vec{q}}[\gamma(\vec{q})]$. For every $i \in [n]$, let

$$\gamma(q_i) = \mathop{\mathrm{E}}_{\vec{q}_{-i}}[\gamma(\vec{q})] \quad \text{and} \quad \eta_i = \mathop{\mathrm{E}}_{\vec{q}} \left[\frac{\gamma(\vec{q}) \cdot \eta_i(\vec{q})}{\gamma(q_i)} \cdot f(\gamma(q_i)) \right],$$

where $f(\alpha) = (1 - (1-\alpha)^M)$, and the above expectation is over uniform distribution over Ω^n. We have

$$\prod_{i=1}^n \eta_i = \prod_{i=1}^n \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{\gamma(\vec{q}) \cdot \eta_i(\vec{q})}{\gamma(q_i)} \right) \cdot f(\gamma(q_i)) \right] \geq \left(\frac{(\gamma - (t+1) \cdot \nu)_+^{n+1}}{(\gamma + \nu)^n} \right).$$

Proof. The proof is similar to that of Lemma 3.6 but a bit more technical. Again, we first write down the whole computation, and then justify the inequalities.

$$\begin{aligned}
&\prod_{i=1}^n \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{\gamma(\vec{q}) \cdot \eta_i(\vec{q})}{\gamma(q_i)} \cdot f(\gamma(q_i)) \right) \right] \\
&= \prod_{i=1}^n \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{\gamma(\vec{q}) \cdot \eta_i(\vec{q})}{\gamma(q_i)/f(\gamma(q_i))} \right) \right] \\
&\geq \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{\gamma(\vec{q})^n \cdot \prod_{i=1}^n \eta_i(\vec{q})}{\prod_{i=1}^n (\gamma(q_i)/f(\gamma(q_i)))} \right)^{1/n} \right]^n \qquad \text{(by Hölder's Inequality)} \\
&\geq \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{\gamma(\vec{q})^n \cdot (\gamma(\vec{q}) - t \cdot \nu)_+^{n+1} / (\gamma(\vec{q}) + \nu)^n}{\prod_{i=1}^n (\gamma(q_i)/f(\gamma(q_i)))} \right)^{1/n} \right]^n \qquad \text{(by induction hypothesis)} \\
&\geq \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{(\gamma(\vec{q}) - (t+1) \cdot \nu)_+^{n+1}}{\prod_{i=1}^n (\gamma(q_i)/f(\gamma(q_i)))} \right)^{1/n} \right]^n \qquad \text{(justified below)} \\
&\geq \left[\left(\frac{\mathrm{E}_{\vec{q}}[(\gamma(\vec{q}) - (t+1) \cdot \nu)_+]^{n+1}}{E_{\vec{q}}[\prod_{i=1}^n (\gamma(q_i)/f(\gamma(q_i)))]} \right)^{1/n} \right]^n \qquad \text{(by Hölder's Inequality)} \\
&\geq \left(\frac{(\gamma - (t+1) \cdot \nu)_+^{n+1}}{(\gamma + \nu)^n} \right) \qquad \text{(justified below)}
\end{aligned}$$

In the first equality, observing that $\alpha / f(\alpha) \to 1/M$ as $\alpha \to 0$, we can take the convention that $0/f(0) = 1/M$. This gives us a correct formula for η_i's, and gets around the zero denominator issue. The application of Hölder's Inequalities are the same as the proof in Lemma 3.6. We check the third and last inequality below.

- Third inequality: we check that the inequality holds pointwise for every $\vec{q}$. The denominator is the same. For the numerator, we need to check that

$$\left(\frac{\gamma(\vec{q})\cdot(\gamma(\vec{q})-t\cdot\nu)_+^{n+1}}{(\gamma(\vec{q})+\nu)^n}\right)\geq(\gamma(\vec{q})-(t+1)\cdot\nu)_+^{n+1},$$

  which follows by inequality $\alpha^n\cdot(\alpha-t\nu)_+^{n+1}\geq(\alpha+\nu)^n\cdot(\alpha-(t+1)\nu)_+^{n+1}$ for every $t,\alpha,\nu\geq 0$ (Claim 3.10 below).

- Last inequality: We have $\mathrm{E}_{\vec{q}}[(\gamma(\vec{q})-(t+1)\cdot\nu)_+]\geq(\gamma-(t+1)\cdot\nu)_+$ for the numerator by Jensen's inequality. For the denominator, we check that for every $i\in[n]$,

$$\mathop{\mathrm{E}}_{q_i}\left[\frac{\gamma(q_i)}{1-(1-\gamma(q_i))^M}\right]\leq\gamma+\nu.$$

  This holds since if $M\nu\geq 1$, then $\alpha/(1-(1-\alpha)^M)\leq\alpha+\nu$ for every $\alpha\in[0,1]$ (Claim 3.11 below).

■

Claim 3.10 *The inequality* $\alpha^n\cdot(\alpha-t\nu)_+^{n+1}\geq(\alpha+\nu)^n\cdot(\alpha-(t+1)\nu)_+^{n+1}$ *holds for every* $t,\alpha,\nu\geq 0$.

> **Proof of claim:** Fix arbitrary $t,\nu\geq 0$, the inequality is trivial for $\alpha\leq(t+1)\nu$. For $\alpha\geq(t+1)\nu$, let us consider $h(x)\stackrel{\text{def}}{=}(\alpha+x)^n\cdot(\alpha-t\nu-x)^{n+1}$. Clearly, we have $h(0)=\alpha^n\cdot(\alpha-t\nu)_+^{n+1}$, and $h(\nu)=(\alpha+\nu)^n\cdot(\alpha-(t+1)\nu)_+^{n+1}$. Furthermore, it is easy to verify that $h'(x)\leq 0$ for every $x\in[0,\nu]$. Therefore, we have $h(0)\geq h(\nu)$, which proves the claim. □

Claim 3.11 *Let* $M\in\mathbb{R}$, $\nu\in(0,1]$ *be two numbers with* $M\nu\geq 1$. *Let*

$$g(\alpha)=\begin{cases}\frac{\alpha}{1-(1-\alpha)^M} & \alpha\in(0,1]\\ 1/M & \alpha=0\end{cases}$$

Then $g(\alpha)\leq\alpha+\nu$ *for* $\alpha\in[0,1]$.

> **Proof of claim:** If $\alpha=0$, then the inequality holds trivially. For the case $\alpha\in(0,1]$, first we consider the function $h(\alpha)=(1-(1-\alpha)^M)(\alpha+\nu)-\alpha$, and prove that $h(\alpha)\geq 0$. By the fact $h'(\alpha)=(1-\alpha)^M(M\alpha+M\nu-1)\geq 0$ for $\alpha\in[0,1]$, which tells that h is a non-decreasing function in $[0,1]$, we have $h(0)=0$ implies $h(\alpha)\geq h(0)\geq 0$ for $\alpha\in[0,1]$. Then we observe that for $\alpha\in(0,1]$, $h(\alpha)\geq 0$ implies $g(\alpha)\leq\alpha+\nu$ (since $(1-(1-\alpha)^M)>0$). Thus the claim holds for $\alpha\in[0,1]$. □

Applying Lemma 3.9 directly completes the proof of induction and the analysis of P^*_{rej}.

3.2.3 Discussion

As we have hinted, the nice properties of public-coin protocols that allow us to prove parallel repetition theorems are that (1) the verifier V's next messages can be efficiently simulated without knowing V's coin, and that (2) V's decision can be efficiently computed from the transcript, also without knowing V's coin. Indeed, it is not hard to see that when both properties are satisfied, the rejection sampling strategy can be implemented efficiently, and one can check that the analysis presented in this section goes through as well. Hence, a tight direct product theorem actually holds for any protocol satisfying both properties. Furthermore, we will show in later sections that the reduction can be generalized to work for more general class of protocols with "simulatable" verifiers, where only the first property holds but not the second one, which proves direct product theorem for these protocols.

On the other hand, as mentioned in the introduction, optimal monotone repetition theorems are known for three-message (private-coin) protocols [22] and constant-round public-coin protocols [3]. However, for protocols with super-constant rounds, we can only prove Chernoff-type theorems (not exactly match the information-theoretic bounds), but the more general threshold/monotone repetition theorems remain open. Here, we briefly discuss the limitations of known reduction strategies when applied to super-constant-round protocols with more general type of parallel verifiers.

As mentioned in this section, Pass and Venkitasubramaniam [30] proved the first tight direct product theorem for *constant-round* public-coin protocols by an efficient reduction strategy that approximate the optimal strategy using recursive sampling. We will show in Section 4.3 that the recursive sampling strategy can be applied to the setting of monotone verifiers and gives *tight* monotone repetition theorem for constant-round public-coin protocols. Unfortunately, the reduction is only efficient for constant-round protocols.

At the first glance, given the simplicity of the rejection sampling strategy and the clean induction hypothesis in the inductive analysis, one may expect that we can extend the result to a tight Chernoff-type theorem by some natural generalization of both the rejection sampling strategy and the inductive analysis.

Unfortunately, this seems to be *not* the case. The issue is that, the rejection sampling strategy P^*_{rej} selects a coordinate $i \in [n]$ uniformly at random. As discussed in Section 3.1.3, this works for the direct product verifier since all coordinates are symmetric, but even for the threshold verifier with threshold $k = n - 1$, randomly selecting a coordinate i cannot achieve the information-theoretic bound. For Chernoff-type verifiers, Wikström [36] showed that the rejection sampling strategy P^*_{rej} still gives fairly good bounds. However, when the threshold is small, selecting a random coordinate $i \in [n]$ becomes a bad idea. Consider the extreme case where the threshold $k = 1$, where it is possible that the given parallel prover P^{n*} always convince only the first subverifier. In this case, a reduction prover strategy P^* that embeds V in a random coordinate $i \in [n]$ can only succeed with probability at most $1/n$. It is

unclear how to modify the way to select the coordinate i for the rejection sampling strategy.

3.3 Efficient Direct Product Theorem for Three-Message Protocols

In thise section, we present a tight efficient direct product theorem for general three message (private-coin) protocols, which says that n-fold parallel repetition reduces soundness error from δ to $\delta^n + \mathsf{ngl}$ for any three-message protocols. Again, this is proved by a black-box reduction converting a parallel prover strategy P^{n*} with success probability $\delta^n + \xi$ to a single instance prover strategy P^* with success probability at least δ.[3]

The reduction presented here is due to Canetti, Halevi, and Steiner [2], who presented it in the context of weakly verifiable puzzles, which are essentially two-message protocols. This reduction can be generalized naturally to prove tight parallel repetition theorems for three-message protocols with threshold verifiers [4] and more general parallel verifiers with monotone combining functions [22], but with more involved analysis. Hence, we first present the direct product theorem of Canetti et al. [2], and then generalize it in later sections. Formally, we prove the following theorem in this section.

Theorem 3.12 *Let* $\mathsf{V} \in \mathrm{PPT}$ *be a three-message verifier. There exists a prover strategy* P^* *such that for every common input* $x \in \{0,1\}^*$*, every* $n \in \mathbb{N}$*, every* $\delta, \xi \in (0,1)$*, and every parallel prover strategy* P^{n*}*,*

1. $\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1] \geq \delta^n + \xi \Rightarrow$

$$\Pr[\langle \mathsf{P}^{*(\mathsf{P}^{n*})}(n, \delta, \xi), \mathsf{V} \rangle(x) = 1] \geq \delta + \frac{\xi}{10n}.$$

2. $\mathsf{P}^{*(\cdot)}(x, n, \delta, \xi)$ *runs in time* $\mathrm{poly}(|x|, n, \xi^{-1})$ *given oracle access to* $\mathsf{P}^{n*}(x)$*.*

Recall that the three messages of $\langle \mathsf{P}^*, \mathsf{V} \rangle$ and $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ are denoted by w, v, p and $\vec{w}, \vec{v}, \vec{p}$ respectively. We use c to denote V's private coins, and write $v = \mathsf{V}(w, c)$ or simply $v = \mathsf{V}(c)$ when the prover's first message w is clear from the context. Again, we assume without loss of generality that P^{n*} is *deterministic*, and hence $\vec{w}$ is fixed and the outcome of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x)$ is determined by $\mathsf{V}^{n,n}$'s private coins $\vec{c} = (c_1, \ldots, c_n)$. With a slight abuse of notation, we write "$\mathsf{P}^{n*}(\vec{c})$ convinces V_{-i},"

[3]The choice of slackness parameters is different from the previous section, but it can be shown that they are equivalent. The reason for the different choice of parameters is for the convenience of presenting the proof.

"$\mathsf{P}^{n*}(v_i, \vec{c}_{-i})$ convinces V_{-i}," or "P^{n*} convinces V_{-i} on $\vec{c}$ (or $(v_i, \vec{c}_{-i})$)" to denote the subverifiers $\mathsf{V}_{-i} = (\mathsf{V}_1, \ldots, \mathsf{V}_{i-1}, \mathsf{V}_{i+1}, \ldots, \mathsf{V}_n)$ accept in the corresponding interaction $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x)$. (Note that $\vec{c}_{-i}$ determines $\vec{v}_{-i}$, which together with v_i determines P^{n*}'s messages $\vec{p}$, which together with $\vec{c}_{-i}$ determines the decisions of V_{-i}.)

We proceed to discuss the reduction prover strategy P^* for three-message protocols. Recall the common framework that the interaction of $\langle \mathsf{P}^*, \mathsf{V} \rangle$ simulates the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$, where (1) P^* first selects coordinate i, (2) V generates random coins $c = c_i$, which determine the message $v_i = \mathsf{V}_i(c_i)$, and then (3) P^* selects the remaining $n-1$ sequences of coins $\vec{c}_{-i}$, which determine messages $\vec{v}_{-i}$. P^* succeeds iff P^{n*} convinces V_i in the corresponding interaction (i.e., V_i with coins c_i accepts the interaction (w_i, v_i, p_i).).

As discussed in Section 3.1.3, in comparison to the public-coin setting, the challenge is that P^* cannot predict V's decision from the transcript, since P^* does not know V's private coin c_i. On the other hand, P^* can compute the decision of the $n-1$ internal subverifiers V_{-i} since P^* knows the coins $\vec{c}_{-i}$. A natural attempt is for P^* to select $\vec{c}_{-i}$ such that P^{n*} convinces all V_{-i} on interaction $(c_i, \vec{c}_{-i})$, as convincing all internal subverifiers may give confidence on convincing the external verifier as well. However, such naive P^* cannot succeed with good probability if the "success pattern" of this coordinate and the remaining coordinates have certain *bad correlations*, as illustrated by the following example. In the discussion below, the slackness parameter ξ is omitted for clarity.

Consider a deterministic parallel prover P^{n*} such that when interacting with $\mathsf{V}^{n,n}$, (i) P^{n*} can convince the parallel verifier $\mathsf{V}^{n,n}$ with probability δ^n, and (ii) for every $i \in [n]$, P^{n*} can convince all except the i-th subverifier with probability $(1 - \delta^n)/n$. Observe that for this P^{n*}, the above naive P^* convinces V iff the simulated interaction $(c_i, \vec{c}_{-i})$ falls in case (i). Intuitively, one can expect the simulated interaction $(c_i, \vec{c}_{-i})$ to fall in each case with probability proportional to $p = \delta^n$ and $q = (1 - \delta^n)/n$, respectively, and hence P^* may succeed with probability only $p/(p+q) \approx n\delta^n \ll \delta$.

Nevertheless, the key observation is that one can exploit such bad correlations to reduce the problem size: one can convert a parallel prover P^{n*} (interacting with $\mathsf{V}^{n,n}$) with such bad correlations to a parallel prover $\mathsf{P}^{(n-1)*}$ (interacting with $\mathsf{V}^{n-1,n-1}$) that has success probability higher than δ^{n-1}. To illustrate the idea using the above example, a such $\mathsf{P}^{(n-1)*}$ can simply interact with $\mathsf{V}^{n-1,n-1}$ by simulating the interaction of P^{n*} and $\mathsf{V}^{n,n}$, where $\mathsf{P}^{(n-1)*}$ simulates P^{n*} and the first coordinate V_1 honestly, and $\mathsf{V}^{n-1,n-1}$ plays the remaining coordinates V_{-1} of $\mathsf{V}^{n,n}$. Hence, $\Pr[\langle \mathsf{P}^{(n-1)*}, \mathsf{V}^{n-1,n-1} \rangle(x) = 1] = \Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-1}]$. It is not hard to see that such $\mathsf{P}^{(n-1)*}$ can succeed with probability $(\delta^n + (1 - \delta^n)/n) \gg \delta^{n-1}$.

More generally, we can consider the above $\mathsf{P}^{(n-1)*}$ with the internal verifier V_1's coin fixed to some c_1^* and estimate the success probability of $\langle \mathsf{P}^{(n-1)^*}, \mathsf{V}^{n-1,n-1} \rangle(x)$. If the success probability is higher than δ^{n-1}, we can reduce the problem size to $n-1$. We can iteratively apply this idea until either (1) $n = 1$ or (2) we cannot find coins c_1^* such that the corresponding $\mathsf{P}^{(n-1)^*}$ has success probability at least δ^{n-1}

(which implies no bad correlations). In case (1), we trivially obtain a P^* with success probability at least δ, while in case (2), it turns out that the above naive strategy P^* can succeed with probability at least δ.

An informal intuition of why the naive strategy works is that since P^{n*} can only convince V_{-1} with probability at most δ^{n-1}, P^{n*} has to convince V_1 with probability at least δ when P^{n*} convinces V_{-1}, so that P^{n*} can convince $\mathsf{V}^{n,n}$ with probability at least δ^n. Although this intuition is not quite accurate, the (stronger) fact that we cannot find any coins c_1^* such that the corresponding $\mathsf{P}^{(n-1)^*}$ has success probability at least δ^{n-1} allows the naive P^* to succeed with probabilty at least δ.

We formalize the above correlation reduction idea and state its property in the following section, and then present the reduction prover strategy P^* for three-message protocols in Section 4.2.2 and its analysis in Section 3.3.3

3.3.1 Correlation Reduction for Direct Product Verifiers

In this section, we formalize the correlation reduction idea discussed above, which iteratively exploits the (bad) correlations (if exist) in the success pattern of a parallel prover P^{n*} (for $\mathsf{V}^{n,n}$) to construct a parallel prover $\mathsf{P}^{(n-1)*}$ (for $\mathsf{V}^{(n-1),(n-1)}$) with good success probability, and hence, no bad correlations exist after the reduction. We emphasize that the reduction is general and can be applied to *any* interactive protocols (not necessarily three-message).

We recap the idea with more details. Again, we omit the slackness parameter ξ for clarity. The starting point is a (deterministic) parallel prover P^{n*} for $\mathsf{V}^{n,n}$ with success probability at least δ^n, and we perform the following process to exploit the correlations in the success pattern of P^{n*}: If $n = 1$, we output P^{n*}. Otherwise, we randomly sample several copies of coins c_1^*, and consider the following (deterministic) parallel prover $\mathsf{P}^{(n-1)*}(c_1^*)$ for $\mathsf{V}^{n-1,n-1}$: $\mathsf{P}^{(n-1)*}(c_1^*)$ interacts with $\mathsf{V}^{n-1,n-1}$ by simulating the interaction of P^{n*} and $\mathsf{V}^{n,n}$, where $\mathsf{P}^{(n-1)*}(c_1^*)$ simulates P^{n*} and the first coordinate V_1 with coins c_1^* honestly, and $\mathsf{V}^{n-1,n-1}$ plays the remaining coordinates V_{-1} of $\mathsf{V}^{n,n}$. We estimate the success probability of $\mathsf{P}^{(n-1)*}(c_1^*)$ by sampling. If there exists a coin c_1^* such that $\mathsf{P}^{(n-1)*}(c_1^*)$ has success probability at least δ^{n-1}, then we repeat the above process with P^{n*} replaced by $\mathsf{P}^{(n-1)*}(c_1^*)$. Otherwise, we output P^{n*}. A formal description of the above process can be found in Figure 3.5.

We shall show that the above process outputs $\mathsf{P}^{n'*}$ such that either (1) $n' = 1$ and $\mathsf{P}^{n'*}$ succeeds with probability δ, or (2) $\mathsf{P}^{n'*}$ has success probability $\delta^{n'}$, and for most coins c_1^*, $\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}_{-1} | c_1 = c_1^*] \leq \delta^{n'-1}$. We formally state the property of the transformation CR as the following lemma.

Let a PPT verifier V (not necessarily three-message), an input $x \in \{0,1\}^*$, parameters $n \in N$, $\delta, \xi \in (0,1)$, and a deterministic parallel prover P^{n*} for $\mathsf{V}^{n,n}$ be given as above.

Sub-Routine $\mathsf{FindC}(\mathsf{P}^{n*}, n, i, \delta, \xi)$
/* Find correlations in the success pattern of P^{n*} on coordinate $i \in [n]$. */
/* Return $\mathsf{P}^{(n-1)*}$ if such correlation is found. Otherwise, return $\perp$ */
Repeat the following at most $M_1 = O\left(\frac{n}{\xi} \cdot \log \frac{n}{\xi}\right)$ times:

- Sample random coins c_i^* and estimate $p(c_i^*) \stackrel{\text{def}}{=} \Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i} | c_i = c_i^*]$ by sampling. Namely, randomly sample $M_2 = O\left(\frac{n^2}{\xi^2} \cdot \log \frac{n}{\xi}\right)$ independent copies of $\vec{c}_{-i}$'s, check if P^{n*} convinces V_{-i} on $(c_i^*, \vec{c}_{-i})$, and compute an estimator $\hat{p}(c_i^*) = |\{\vec{c}_{-i} : \mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i} \text{ on } (c_i^*, \vec{c}_{-i})\}|/M_2$.
- If $\hat{p}(c_i^*) \geq \delta^{n-1} + (1 - (1/n)) \cdot \xi$, then return a parallel prover $\mathsf{P}^{(n-1)*}(c_i^*)$ for $\mathsf{V}^{n-1,n-1}$ defined as follows: $\mathsf{P}^{(n-1)*}(c_i^*)$ interacts with $\mathsf{V}^{n-1,n-1}$ by simulating the interaction of P^{n*} and $\mathsf{V}^{n,n}$, where $\mathsf{P}^{(n-1)*}(c_i^*)$ simulates P^{n*} and the i-th coordinate V_i with coin c_i^* honestly, and $\mathsf{V}^{n-1,n-1}$ plays the remaining coordinates V_{-i} of $\mathsf{V}^{n,n}$.

Return $\perp$ after M_1 (failure) attempts.

$\mathsf{CR}(\mathsf{P}^{n*}, n, \delta, \xi)$
/* Implicitly, there are a PPT verifier V and an input x as part of the input. */
/* Iteratively exploit correlation in the success pattern of P^{n*} to obtain $\mathsf{P}^{(n-1)*}$.*/
Iteratively apply FindC until $n = 1$ or FindC returns $\perp$, namely

- Call $\mathsf{FindC}(\mathsf{P}^{n*}, n, 1, \delta, \xi)$. If FindC returns $\mathsf{P}^{(n-1)*}$, then set $\xi \leftarrow \left(1 - \frac{1}{n}\right) \cdot \xi$ and $n \leftarrow n - 1$ (so that P^{n*} refers to the prover strategy returned by FindC).

Return the final P^{n*}.

Figure 3.5: Correlation reduction for direction product verifiers.

Lemma 3.13 *If P^{n*} has success probability at least $(\delta^n + \xi)$ on input x, then with probability at least $(1 - (\xi/10n))$ over the randomness of CR, $\mathsf{CR}(\mathsf{P}^{n*}, n, \delta, \xi)$ outputs a deterministic prover strategy $\mathsf{P}^{n'*}$ satisfying the following properties.*

- $\Pr[\langle \mathsf{P}^{n'*}, \mathsf{V}^{n',n'} \rangle(x) = 1] \geq \delta^{n'} + ((10n' - 1)/10n) \cdot \xi$.
- *Either $n' = 1$, or with probability at least $(1 - (\xi/10n))$ over V_1's coin c_1^*,*

$$\Pr[\mathsf{P}^{n'*} \textit{ convinces } \mathsf{V}_{-1} | c_1 = c_1^*] \leq \delta^{n'-1} + \frac{10(n'-1)+1}{10n} \cdot \xi.$$

Furthermore, $\mathsf{CR}(\mathsf{P}^{n*}, n, \delta, \xi)$ can be implemented with oracle access to P^{n*} with runtime $\mathrm{poly}(|x|, n, \xi^{-1})$, and the output $\mathsf{P}^{n'*}$ can be implemented in time $\mathrm{poly}(|x|, n)$ given oracle access to P^{n*}.

Proof. We first verify the "furthermore" part of the lemma. We observe that CR calls FindC at most n times, and FindC simulates the interaction $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x)$ at most $M_1 \cdot M_2$ times. Since V is PPT, CR runs in time $\mathrm{poly}(|x|, n, \xi^{-1})$ given oracle access to P^{n*}. Also, a careful inspection shows that the output $\mathsf{P}^{n'*}$ of CR is simply the following: $\mathsf{P}^{n'*}$ interacts with $\mathsf{V}^{n',n'}$ by simulating the interaction of P^{n*} and $\mathsf{V}^{n,n}$, where $\mathsf{P}^{n'*}$ simulates P^{n*} and the first $n - n'$ coordinates $\mathsf{V}_1, \ldots, \mathsf{V}_{n-n'}$ of $\mathsf{V}^{n,n}$ with some fixed coins $c_1^*, \ldots, c_{n-n'}^*$ (found by FindC iteratively) honestly, and $\mathsf{V}^{n',n'}$ plays the remaining coordinates of $\mathsf{V}^{n,n}$. Hence, the output $\mathsf{P}^{n'*}$ can be implemented in time $\mathrm{poly}(|x|, n)$ given oracle access to P^{n*}.

We proceed to prove the main statement of the lemma. Observe that the subroutine FindC uses sampling to (1) find a coin c_i^* such that the probability $p(c_i^*) \stackrel{\text{def}}{=} \Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i} | c_i = c_i^*]$ is high, and (2) estimate the probability $p(c_i^*)$. For intuition, we first prove the lemma assuming that there were no sampling errors. Namely, FindC could find such a c_i^* if there exists one and compute $p(c_i^*)$ exactly. In this case, the output prover $\mathsf{P}^{n'}$ can succeed with probability at least

$$\delta^{n'} + \left(\prod_{k=n'+1}^{n} \left(1 - \frac{1}{k}\right) \right) \cdot \xi = \delta^{n'} + \frac{n'}{n} \cdot \xi > \delta^{n'} + \frac{10n' - 1}{10n} \cdot \xi.$$

Also, if $n' > 1$, then the fact that $\mathsf{FindC}(\mathsf{P}^{n'}, n', 1, \delta, (n'/n) \cdot \xi)$ returns $\perp$ implies that for every c_1^*,

$$\Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i} | c_i = c_i^*] < \delta^{n'-1} + \left(1 - \frac{1}{n'}\right) \cdot \frac{n'}{n} \cdot \xi < \delta^{n'-1} + \frac{10(n'-1)+1}{10n} \cdot \xi.$$

We continue to analyze the actual transformation. Note that the parameters M_1 and M_2 in FindC are chosen so that

- If
$$\Pr_{c_i^*} \left[p(c_i^*) \geq \delta^{n-1} + \frac{10(n-1)+1}{10n} \cdot \xi \right] \geq \frac{\xi}{10n}, \tag{3.1}$$
then with probability at least $(1 - (\xi/20n^2))$ over the M_1 random samples of c_i^*, at least one of c_i^* satisfies the above event, i.e., $p(c_i^*) \geq \delta^{n-1} + ((10(n-1)+1)/10n) \cdot \xi$. Indeed, the probability that FindC fails to find such a c_i^* is at most $(1-(\xi/10n))^{M_1} \leq (\xi/20n^2)$ for a properly chosen constant in $M_1 = O\left(\frac{n}{\xi} \log \frac{n}{\xi}\right)$. We say that the sampling is *failed* if (3.1) holds but no such a c_i^* is found.

- With probability at least $(1 - (\xi/20M_1n^2))$ over the M_2 random samples of $\vec{c}_{-i}$, the estimator $\hat{p}(c_i^*)$ computed via these $\vec{c}_{-i}$'s satisfies $|\hat{p}(c_i^*) - p(c_i^*)| \leq (\xi/10n)$. This follows by a standard Chernoff bound with a properly chosen constant in $M_2 = O\left(\frac{n^2}{\xi^2} \log \frac{n}{\xi}\right)$. We say that the sampling is *failed* if $|\hat{p}(c_i^*) - p(c_i^*)| > (\xi/10n)$.

Also observe that CR makes at most n calls to FindC, and each FindC consists of one sampling of the first type and at most M_1 sampling of the second type. By an union bound, we know that with probability at least $(1 - (\xi/10n))$ over the randomness of CR, no sampling is failed during the execution of CR. In this case, the output prover $\mathsf{P}^{n'}$ (constructed by $\mathsf{FindC}(\mathsf{P}^{n'+1}, n'+1, i, \delta, ((n'+1)/n) \cdot \xi)$) can succeed with probability at least

$$\delta^{n'} + \frac{n'}{n} \cdot \xi - \frac{\xi}{10n} = \delta^{n'} + \frac{10n' - 1}{10n} \cdot \xi,$$

since the estimator has error less than $\xi/10n$. Also, when $n' > 1$, the facts that $\mathsf{FindC}(\mathsf{P}^{n'}, n', 1, \delta, (n'/n) \cdot \xi)$ returns $\perp$ and that no sampling is failed imply that with probability at least $(1 - (\xi/10n))$ over c_1^*,

$$\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}_{-1} | c_1 = c_1^*] < \delta^{n'-1} + \frac{n'-1}{n} \cdot \xi + \frac{\xi}{10n} = \delta^{n'-1} + \frac{10(n'-1)+1}{10n} \cdot \xi,$$

since otherwise, FindC can find a c_1^* with $p(c_1^*) \geq \delta^{n'-1} + (10(n'-1)+1)/(10n) \cdot \xi$, and so the corresponding estimator $\hat{p}(c_1^*) \geq \delta^{n'-1} + ((n'-1)/n) \cdot \xi$. ■

3.3.2 Reduction Prover Strategy P^*

In this section, we present the reduction prover strategy P^* of Canetti et al. [2] for proving a tight direct product theorem for three message protocols. As discussed, the idea is to first apply the correlation reduction CR to the given (deterministic) parallel prover P^{n*} to get rid of bad correlations in the success pattern of P^{n*}, and then apply the aforementioned "naive" strategy to the resulting prover strategy $\mathsf{P}^{n'*}$ from CR.

We recap with a bit more detail on the naive strategy as follows. The interaction of $\langle \mathsf{P}^*, \mathsf{V} \rangle$ simulates the interaction of $\langle \mathsf{P}^{n'*}, \mathsf{V}^{n',n'} \rangle$, and P^* needs to select coordinate i and $n'-1$ internal subverifiers' coins $\vec{c}_{-i}$. P^* always selects the first coordinate as correlation reduction guarantees no bad correlations. Then P^* uses random sampling to find a $\vec{c}_{-1}$ such that $\mathsf{P}^{n'*}$ convinces V_{-1} on the corresponding interaction. Intuitively (omitting the slackness parameter ξ), since $\mathsf{P}^{n'}$, as Lemma 3.13 promised, has the additional property that with high probability over V_1's coin c_1^*, $\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}_{-1}] \leq \delta^{n'-1}$, $\mathsf{P}^{n'*}$ needs to convince V_1 with probability at least δ when $\mathsf{P}^{n'*}$ convinces V_{-1}, so that $\mathsf{P}^{n'}$ can convince $\mathsf{V}^{n',n'}$ with probability at least $\delta^{n'}$. A formal description of P^* can be found in Figure 3.6.

3.3.3 Analysis of the Prover Strategy P^*

We shall show that P^* can succeed with good probability. Again, it is instructive to consider the ideal case where there are no sampling errors for intuition. Specifically, we consider an ideal scenario (where there are no sampling errors in both the CR transformation and sampling $\vec{c}_{-1}$) that satisfies the following properties:

Prover Strategy $\mathsf{P}^*(x, n, \delta, \xi)$
/* P^* interacts with V and is given oracle access to P^{n*}. */

- P^* applies CR to P^{n*} to obtain $\mathsf{P}^{n'*} = \mathsf{CR}(\mathsf{P}^{n*}, n, \delta, \xi)$.
- If $n' = 1$, P^* interacts with V by running $\mathsf{P}^{n'*}$; otherwise, P^* does the following.
- P^* runs $\mathsf{P}^{n'*}$ to generate $\vec{w}$, and sends w_1 to V.
- Upon receiving $v_1 = v$ from V, P^* repeats the following at most $M = O\left(\frac{n}{\xi} \cdot \log \frac{n}{\xi}\right)$ times.
  - P^* generates random coins $\vec{c}_{-1} = (c_2, \ldots, c_{n'})$. Then P^* checks if $\mathsf{P}^{n'*}(v_1, \vec{c}_{-1})$ convinces V_{-1}. Namely, P^* computes $\vec{v}_{-1} = \mathsf{V}_{-1}(\vec{c}_{-1})$ and $\vec{p} = \mathsf{P}^{n*}(v_1, \vec{v}_{-1})$, and checks if V_j with coin c_j accepts transcript (w_j, v_j, p_j) for every $j = 2, \ldots, n$. If so, P^* sends p_1 to V and terminates.
- If all the M attempts fail, P^* sends $\perp$ to V (or simply abort).

Figure 3.6: Reduction prover strategy P^* for three-message protocols.

- The $\mathsf{P}^{n'}$ returned by CR (assuming that $n' > 1$ as the $n' = 1$ case is trivial) always satisfies the following properties:
  - $\Pr[\langle \mathsf{P}^{n'*}, \mathsf{V}^{n',n'} \rangle(x) = 1] \geq \delta^{n'} + (n'/n) \cdot \xi$.
  - For every c_1^*, $\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}_{-1} | c_1 = c_1^*] \leq \delta^{n'-1} + ((n'-1)/n) \cdot \xi$.
- If there exist $\vec{c}_{-1}$ such that $\mathsf{P}^{n'*}$ convinces V^{-1} on $(v_1, \vec{c}_{-1})$, then P^* can sample a uniformly random such $\vec{c}_{-1}$. (This is achieved by randomly sample a unbounded number of $\vec{c}_{-1}$ as opposed to at most M times in Figure 3.6.)

We argue that (in this ideal scenario) the success probability of P^* can be expressed in the following formula:

$$\Pr[\langle \mathsf{P}^*, \mathsf{V} \rangle(x) = 1] = \mathop{\mathrm{E}}_{c_1^*} \left[\frac{\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}^{n',n'} | c_1 = c_1^*]}{\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}_{-1} | c_1 = c_1^*]} \right].^4$$

First, the expectation operator corresponds to that V uses uniformly random coin c_1^*. Second, note that P^* samples a random $\vec{c}_{-1}$ conditioning on $\mathsf{P}^{n'*}$ convinces V_{-1} on $(c_1^*, \vec{c}_{-1})$, and P^* succeeds iff $\mathsf{P}^{n'*}$ also convinces V_1 on $(c_1^*, \vec{c}_{-1})$. Conditioning on V's coin being c_1^*, the success probability of P^* is precisely

$$\frac{\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}^{n',n'} | c_1 = c_1^*]}{\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}_{-1} | c_1 = c_1^*]}.$$

[4] Here, we take a convention that $0/0 = 0$.

Recall that for *every* c_1^*, $\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}_{-1}|c_1 = c_1^*] \leq \delta^{n'-1} + ((n'-1)/n) \cdot \xi$ (again, in the ideal scenario). We can bound the expectation as follows.

$$\begin{aligned}
&\mathop{\mathrm{E}}_{c_1^*}\left[\frac{\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}^{n',n'}|c_1 = c_1^*]}{\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}_{-1}|c_1 = c_1^*]}\right] \\
&\geq \mathop{\mathrm{E}}_{c_1^*}\left[\frac{\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}^{n',n'}|c_1 = c_1^*]}{\delta^{n'-1} + ((n'-1)/n) \cdot \xi}\right] \\
&= \frac{\mathrm{E}_{c_1^*}\left[\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}^{n',n'}|c_1 = c_1^*]\right]}{\delta^{n'-1} + ((n'-1)/n) \cdot \xi} \\
&= \frac{\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}^{n',n'}]}{\delta^{n'-1} + ((n'-1)/n) \cdot \xi} \\
&\geq \frac{\delta^{n'} + (n'/n) \cdot \xi}{\delta^{n'-1} + ((n'-1)/n) \cdot \xi} \\
&\geq \delta + \frac{\xi}{n}.
\end{aligned}$$

This completes the analysis in the ideal scenario.

We proceed to analyze the actual (non-ideal) prover strategy P^*. Again, the challenge is to show that the sampling errors does not lower the success probability too much. We shall show that if P^{n*} has success probability at least $\delta^n + \xi$, then P^* can succeed with probability at least $\delta + (\xi/10n)$.

Recall that by Lemma 3.13, with probability at least $(1 - (\xi/10n))$ over the randomness of CR, the $\mathsf{P}^{n'*}$ returned by CR satisfies two good properties. Let us call a $\mathsf{P}^{n'*}$ returned by CR *good* if $\mathsf{P}^{n'}$ satisfies the two properties stated in the lemma. Informally, Lemma 3.13 allows us to focus on good $\mathsf{P}^{n'*}$'s with a loss of of at most $(\xi/10n)$ on the success probability, since $\mathsf{P}^{n'*}$ is not good with probability at most $(\xi/10n)$. Also observe that by definition, when $\mathsf{P}^{n'*}$ is good and $n' = 1$, P^* can success with probability at least $\delta + (9/10n) \cdot \xi$. Therefore, it remains to analyze the success probability of P^* for the case that the $\mathsf{P}^{n'*}$ returned by CR is good and $n' > 1$.

Fix a good $\mathsf{P}^{n'*}$ returned by CR with $n' > 1$. We introduce two shorthand notations below to simplify the expressions later in the analysis, and recall the two properties stated in Lemma 3.13 using the new notations. We define

$$\begin{aligned}
\alpha(c_1^*) &\stackrel{\text{def}}{=} \Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}^{n',n'}|c_1 = c_1^*], \text{ and} \\
\beta(c_1^*) &\stackrel{\text{def}}{=} \Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}_{-1}|c_1 = c_1^*].
\end{aligned}$$

With these notations, Lemma 3.13 say that a good $\mathsf{P}^{n'*}$ with $n' \geq 1$ satisfies

- $\mathrm{E}_{c_1^*}[\alpha(c_1^*)] \geq \delta^{n'} + ((10n'-1)/10n) \cdot \xi$.
- With probability at least $(1 - (\xi/10n))$ over c_1^*,

$$\beta(c_1^*) \leq \delta^{n'-1} + \frac{10(n'-1)+1}{10n} \cdot \xi.$$

Let us call c_1^* is good and denote it by $c_1^* \in \mathsf{Good}$ if the above inequality holds. In other words, we define

$$\mathsf{Good} = \left\{ c_1^* : \beta(c_1^*) \leq \delta^{n'-1} + \frac{10(n'-1)+1}{10n} \cdot \xi \right\}.$$

By a similar argument as that in the analysis of the ideal scenario, we observe that the success probability of P^* can be expressed as

$$\Pr[\langle \mathsf{P}*, \mathsf{V} \rangle(x) = 1 | \mathsf{P}^{n'}] = \mathop{\mathrm{E}}_{c_1^*} \left[\frac{\alpha(c_1^*)}{\beta(c_1^*)} \cdot \left(1 - (1 - \beta(c_1^*))^M\right) \right],$$

where $\left(1 - (1 - \beta(c_1^*))^M\right)$ is the probability that P^* can find a $\vec{c}_{-1}$ such that $\mathsf{P}^{n'*}$ convinces V_{-1} on $(c_1^*, \vec{c}_{-1})$ from at most M random samples of $\vec{c}_{-1}$. Our goal is to lower bound the expectation.

Observing that by definition, $\alpha(c_1^*) \leq \beta(c_1^*)$, and when $\beta(c_1^*)$ is not too small, $(1 - (1 - \beta(c_1^*)^M)$ is very close to 1, we can simplify the quantity in the expectation by the following simple claim.

Claim 3.14 *Let $\gamma \in (0,1)$. If $M \in \mathbb{N}$ satisfies $(1-\gamma)^M \leq \gamma$, then for every α, β with $0 \leq \alpha \leq \beta \leq 1$, we have*

$$\frac{\alpha}{\beta} \cdot \left(1 - (1-\beta)^M\right) \geq \frac{(\alpha - \gamma)_+}{\beta},$$

where $(x)_+ \stackrel{\text{def}}{=} \max\{x, 0\}$.[5]

> **Proof of claim:** Note that the LHS is non-negative. If $\beta \leq \gamma$, then RHS $= 0$ and the inequality holds. If $\beta > \gamma$, then $(1 - (1-\beta)^M) \geq 1 - \gamma$. It follows that
>
> $$\frac{\alpha}{\beta} \cdot (1 - (1-\beta)^M) \geq \frac{\alpha \cdot (1-\gamma)}{\beta} \geq \frac{(\alpha - \gamma)_+}{\beta}.$$
>
> □

Choosing the constant in $M = O\left(\frac{n}{\xi} \log \frac{n}{\xi}\right)$ properly so that $(1 - (\xi/10n))^M \leq (\xi/10n)$, and using the above claim, we have

$$\mathop{\mathrm{E}}_{c_1^*} \left[\frac{\alpha(c_1^*)}{\beta(c_1^*)} \cdot \left(1 - (1 - \beta(c_1^*))^M\right) \right] \geq \mathop{\mathrm{E}}_{c_1^*} \left[\frac{(\alpha(c_1^*) - (\xi/10n))_+}{\beta(c_1^*)} \right].$$

[5] Again, we use a convention that $0/0 \stackrel{\text{def}}{=} 0$.

Also, we can get rid of bad c_1^* by

$$\mathop{\mathrm{E}}_{c_1^*}\left[\frac{(\alpha(c_1^*)-(\xi/10n))_+}{\beta(c_1^*)}\right] \geq \mathop{\mathrm{E}}_{c_1^*}\left[\frac{(\alpha(c_1^*)-(\xi/10n))_+ \cdot \mathbf{1}[c_1^* \in \mathsf{Good}]}{\beta(c_1^*)}\right],$$

where $\mathbf{1}[\mathcal{E}] = 1$ if the event $\mathcal{E}$ is true, and 0 otherwise. We can then perform similar calculation as in the ideal scenario:

$$\begin{aligned}
&\mathop{\mathrm{E}}_{c_1^*}\left[\frac{(\alpha(c_1^*)-(\xi/10n))_+ \cdot \mathbf{1}[c_1^* \in \mathsf{Good}]}{\beta(c_1^*)}\right] \\
&\quad\geq \mathop{\mathrm{E}}_{c_1^*}\left[\frac{(\alpha(c_1^*)-(\xi/10n))_+ \cdot \mathbf{1}[c_1^* \in \mathsf{Good}]}{\delta^{n'-1}+((10(n'-1)+1)/10n)\cdot\xi}\right] \\
&\quad= \frac{\mathrm{E}_{c_1^*}\left[(\alpha(c_1^*)-(\xi/10n))_+ \cdot \mathbf{1}[c_1^* \in \mathsf{Good}]\right]}{\delta^{n'-1}+((10(n'-1)+1)/10n)\cdot\xi},
\end{aligned}$$

where we can bound the numerator by

$$\begin{aligned}
&\mathop{\mathrm{E}}_{c_1^*}\left[(\alpha(c_1^*)-(\xi/10n))_+ \cdot \mathbf{1}[c_1^* \in \mathsf{Good}]\right] \\
&\quad\geq \mathop{\mathrm{E}}_{c_1^*}\left[\alpha(c_1^*)\cdot\mathbf{1}[c_1^* \in \mathsf{Good}]\right] - (\xi/10n) \\
&\quad\geq \mathop{\mathrm{E}}_{c_1^*}\left[\alpha(c_1^*)\right] - \Pr[c_1^* \notin \mathsf{Good}] - (\xi/10n) \\
&\quad\geq (\delta^{n'}+((10n'-1)/10n)\cdot\xi) - (\xi/10n) - (\xi/10n) \\
&\quad= \delta^{n'}+((10n'-3)/10n)\cdot\xi.
\end{aligned}$$

Putting things together, for every good $\mathsf{P}^{n'*}$ returned by CR with $n' > 1$, we have

$$\Pr[\langle \mathsf{P}^*, \mathsf{V}\rangle(x) = 1 | \mathsf{P}^{n'*}] \geq \frac{\delta^{n'}+((10n'-3)/10n)}{\delta^{n'-1}+((10(n'-1)+1)/10n)\cdot\xi} \geq \delta + \frac{2\xi}{10n}.$$

It follows that

$$\begin{aligned}
&\Pr[\langle \mathsf{P}^*, \mathsf{V}\rangle(x) = 1] \\
&\quad\geq \Pr[\langle \mathsf{P}^*, \mathsf{V}\rangle(x) = 1 | \mathsf{P}^{n'*} \text{ is good }] - \Pr[\mathsf{P}^{n'*} \text{ is not good }] \\
&\quad\geq \delta + \frac{2\xi}{10n} - \frac{\xi}{10n} \\
&\quad\geq \delta + \frac{\xi}{10n},
\end{aligned}$$

which completes the analysis.

3.3.4 Historical Notes and Discussion

The first direct product theorem for three-message protocols is proved by Bellare, Impagliazzo, and Naor [1], who showed that n-fold parallel repetition decreases soundness error from $(1-\varepsilon)$ to $e^{-\Omega(\varepsilon^2 n)} + \mathsf{ngl}$ for three-message protocols using a *different* reduction strategy. Later on, Canetti, Halevi, and Steiner [2] improved the bound to tight using the reduction strategy presented in this section.

We discuss the two different approaches below. Recall that the main challenge for the setting of three-message protocols is that P^* cannot predict the verifier V's decision due to the lack of V's coins.

- As presented in this section, Canetti et al. use *correlation reduction* to deal with the challenge. Correlation reduction allows us to get rid of bad correlations, and hence P^* can succeed with good probability by sampling a random $\vec{v}_{-i}$ such that P^{n*} convinces *all* internal subverifiers V_{-i} on $(v_i, \vec{v}_{-i})$.

- In contrast, Bellare et al. use a "*soft-decision*" method to deal with the challenge. There is no preprocessing and P^* selects a random coordinate $i \in [n]$ in which to embed V.[6] The term soft-decision refers to how P^* selects $\vec{v}_{-i}$: P^* randomly sample a $\vec{v}_{-i}$ and decides whether to accept this $\vec{v}_{-i}$ *probabilistically* with probability depending on the number of internal subverifiers V_{-i} that P^{n*} can convince on $(v_i, \vec{v}_{-i})$. Specifically, P^* accepts the $\vec{v}_{-i}$ with probability 2^{-t} if t out of the $n-1$ subverifiers V_{-i} reject on $(v_i, \vec{v}_{-i})$.

To get some intuition on why the soft-decision method works, let us see how it works on the adversarial example presented in this section, which is recalled below: Consider a deterministic parallel prover P^{n*} such that when interacting with $\mathsf{V}^{n,n}$, (i) P^{n*} can convince the parallel verifier $\mathsf{V}^{n,n}$ with probability δ^n, and (ii) for every $i \in [n]$, P^{n*} can convince all except the i-th subverifier with probability $(1-\delta^n)/n$.

Informally, in this example, with probability $\delta^n + (1-\delta^n)/n$, P^* samples a $\vec{v}_{-i}$ such that P^{n*} convinces all V_{-i}, and P^* accepts this $\vec{v}_{-i}$ with probability 1. Conditioning on this case, P^* is "cheated" and can only succeed with small probability, roughly $n\delta^n$. However, with the remaining $(n-1)\cdot(1-\delta^n)/n$ probability, P^* samples a $\vec{v}_{-i}$ such that only one subverifiers of V_{-i} rejects, and P^* accepts this $\vec{v}_{-1}$ with probability $1/2$ according to the soft-decision. Note that conditioning on this case, P^* succeeds with probability 1, which compensate the first case where P^* is cheated.

As indicated in the above example, soft-decision works since it smooths out the "error" that it makes. Very informally, whenever it gets "cheated", it gets some compensation (with smaller "weight") from other coordinates. However, this seems to be also the reason that soft-decision does not achieve optimal information-theoretic

[6] Actually, Bellare et al. used sampling to select a best coordinate $i \in [n]$ in which to embed V. However, it can be shown that a random coordinate also works, which is also the choice of later generalizations of Bellare et al. in other settings.

bounds, since the smoothing process incurs some inherent loss. Nevertheless, both techniques are useful and can be generalized to different settings, which we briefly discuss below.

- For three-message protocols, the correlation reduction approach can be generalized naturally to handle any threshold verifiers [4] and even more general monotone verifiers [22], which give *tight* bounds for these settings. Furthermore, for more general protocols with *simulatable* verifiers[7] studied in the next section, the correlation reduction approach can be combined with the rejection sampling strategy in Section 3.2 to give a nearly tight direct product theorem for protocols with simulatable verifiers. However, it is unclear how to use this approach to handle threshold or even Chernoff-type (parallel) verifiers[8] for protocols with simulatable verifiers.

- In contrast, the soft-decision method seems to only be able to be generalized to handle Chernoff-type verifiers and cannot achieve optimal information-theoretic bounds. However, it can be applied to all classes of protocols where parallel repetition is known to decrease soundness error, and gives Chernoff-type theorems for all these settings.

Finally, we remark that while the reduction presented in this section is discovered by Canetti et al. [2], the analysis presented here is in spirit close to the analysis of Holenstein and Schoenebeck [22]. The analysis of Holenstein and Schoenebeck handles the sampling errors in a better way, which allows the numbers of samples M_1, M_2, M that P^* needs to generate depend *only* on the (additive) slackness parameter ξ, but independent of the success probability δ^n. This is important to obtain an efficient reduction for the more general case of verifier with arbitrary monotone combining functions in [22].

On the other hand, we [4] considered the same generalization of the reduction presented in this section as Holenstein and Schoenebeck [22]. However, the better analysis is missed in [4], and hence we are only able to get an efficient reduction when the number of repetition n is small. To obtain an efficient reduction for threshold parallel verifiers with large n, we apply a generic reduction first to reduce n to a small enough n' before applying the generalized reduction.

---

[7] Informally, a verifier V is *simulatable* if one can simulate V's next messages without knowing V's coin.

[8] Recall that a Chernoff-type verifier is a threshold verifier with sufficiently large threshold.

3.4 Efficient Direct Product Theorem for Computationally Simulatable Protocols

In this section, we define a more general class of *computationally simulatable protocols*, and prove a direct product theorem for computationally simulatable protocols. The class of computationally simulatable protocols contains both public-coin protocols and three-message protocols. More interestingly, we will show in the next section that *any* interactive protocols can be made computationally simulatable by running it under a fully homomorphic encryption scheme, such as the one recently proposed by Gentry [11]. This gives a way to get around the negative results of Bellare et al. [1] and Pietrzak and Wikström [31] and amplify soundness for any interactive protocol in a round preserving way. Namely, we first make the protocol computationally simulatable, and then do parallel repetition.

3.4.1 Definition of Simulatability and Theorem Statement

The study of "simulatable protocols" was initiated by Haståd, Pass, Pietrzak, and Wikström [19]. They observed that parallel repetition theorems hold for protocols where the rejection sampling strategy P^*_{rej} can be implemented efficiently. In other words, it suffices that P^*_{rej} can efficiently sample random continuations of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ and identify successful continuations. Furthermore, the rejection sampling strategy can be modified so that the definition of successful continuation does not depend on the external verifier's decision. Hence, only the verifier's next messages (but not necessary his decision) need to be simulated efficiently. Roughly speaking, "simulatable protocols" are protocols where a certain version of rejection sampling strategy can be implemented efficiently, so that parallel repetition theorem holds for these protocols.

However, the simulatability property is tricky to define. The first definition of Haståd et al. [19] defined not only a simulatability property, but also an "extendability" property. A later version of Haståd et al. [20] considered a simpler (but incomparable) formulation and defined a *weak simulatablility* property. Roughly speaking, weak simulatability only requires simulating continuation of the interaction conditioned on some noticeable event, but the quality of the simulation is required to be statistically close. In contrast, we define a *computational simulatability* property, which requires simulating the whole continuation of the interaction, but a computationally indistinguishable simulation is sufficient. The above three formulations are subtly different and incomparable.

Before further discussions, let us consider some examples informally. Public-coin protocols are clearly simulatable, since the verifier's messages are simply independent random strings. Three-message protocols are also simulatable, since the first message of a verifier V is easy to simulate by just running V with fresh random coins. Note that for three-message protocols, a prover P^* cannot compute V's decision, but this is

OK for simulatability. On the other hand, the negative example discussed in Section 1.1.1 is *not* simulatable, since the second message of V is hard to simulate unless the prover can generate the key from only the box.

We proceed to discuss subtle issues in defining the computational simulatability and introduce our definition formally.

Issue on Random Continuation. Let us first look closely at the rejection sampling strategy P^*_{rej} for public-coin protocols. Consider the interaction of $\langle \mathsf{P}^*, \mathsf{V} \rangle$ at the ℓ-th round for some $\ell \in [m]$. Before the ℓ-th round, P^* and V jointly select the parallel verifier $\mathsf{V}^{n,n}$'s messages $\vec{v}_{[\ell-1]} = (\vec{v}_1, \ldots, \vec{v}_{\ell-1})$. At the ℓ-th round, P^*_{rej} receives a message $v_{\ell,i}$ from the external verifier $\mathsf{V} = \mathsf{V}_i$, and P^*_{rej} selects $\vec{v}_{\ell,-i}$ by rejection sampling. Namely, P^*_{rej} keeps sampling random continuation of the parallel interaction $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ from partial interaction $(\vec{v}_1, \ldots, \vec{v}_{\ell-1}, v_{\ell,i})$, rejecting until a successful continuation is found, and selects the corresponding $\vec{v}_{\ell,-i}$. For the case of public-coin protocols, sampling a random continuation amounts to sampling uniformly random messages $(\vec{v}_{\ell,-i}, \vec{v}_{\ell+1}, \ldots, \vec{v}_m)$.

However, when the protocol is not public-coin, the definition of a random continuation is less clear. Does a random continuation refer to a random interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ conditioned on the *transcript* $(\vec{v}_1, \ldots, \vec{v}_{\ell-1}, v_{\ell,i})$, or conditioned on the *random coins* tossed so far? It turns out that both definitions *work* in the sense that the rejection sampling strategy can succeed with good probability under both definitions of random continuations. However, the issue is whether P^*_{rej} can sample a random continuation efficiently.

Suppose random continuations are defined by conditioning on the transcript. Then even the $n-1$ internal verifiers V_{-i}'s messages may also be hard to generate. For example, a verifier's first two messages may be $(h, h(x))$ and x, where h is a shrinking collision-resistant hash function and x is a random input to h. Sampling a random second message x' conditioned on the first message $(h, h(x))$ is equivalent to sample a random preimage of $h(x)$, which breaks the collision-resistant property. The original formulation of Haståd et al. [19] used this definition, and so they required not only simulatability but also "extendability," which roughly says that a prover can sample the external verifiers' and the internal verifiers' next messages, respectively.

In contrast, if random continuations are defined by conditioning on the random coins tossed so far, then the issue of simulating the internal verifiers' messages goes away. In the above example, conditioned on the coins h and x, the second message is deterministic, namely, just x, which is trivial to sample given the coins h and x. In general, for the $n-1$ internal subverifiers V_{-i}, P^*_{rej} has the coins tossed in the previous $\ell-1$ rounds, so P^*_{rej} can sample the ℓ-th round message of V_{-i} conditioned on the previous coins easily.

On the other hand, the task of simulating the external verifier $\mathsf{V} = \mathsf{V}_i$'s next message conditioned on the previous coins is non-trivial, since P^*_{rej} does not have V_i's

coins. Both the weak simulatablility property and the computational simulatablility property refer to the ability to efficiently perform this task in certain respects.

We remark that one needs to carefully specify the coins tossed by V in each round so that such task is possible. As an extreme example, consider a public-coin verifier V who tosses all coins in the first round. Conditioned on the coins tossed in the first round, the second message is fixed. However, there is no way for a prover to guess the fixed message correctly without knowing the verifier's coins.

Issue on Computational Indistinguishability. We mentioned that the rejection sampling strategy can succeed with good probability if P^*_{rej} can simulate a random continuation perfectly, i.e., P^*_{rej} can perfectly sample a random interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ conditioned on the coins tossed in the first ℓ-rounds of interaction. Intuitively, since both P^*_{rej} and V are PPT, computationally indistinguishable samples of random continuations should be as good as samples from the actual distribution. This is indeed the case when computational indistinguishability is formulated properly. There are a few possible formulations. The question is what is the most general formulation of computational indistinguishability for defining computational simulatablility property, so that the class of computationally simulatable protocols contains more interesting protocols, and the proof of parallel repetition goes through.

Our definition of computational simulatability generalizes the definition in the original version of Haståd et al. [19] in some aspects. The generalization is necessary for us to prove that running a protocol under fully homomorphic encryption schemes makes the protocol computationally simulatable. The key issue is that, Haståd et al. [19] requires computationally indistinguishability to hold even against distinguishers with verifier's coins. This is problematic for us since the verifier's coins would leak the secret keys of fully homomorphic encryption schemes completely.

Our Definition. We proceed to introduce our first definition of computational simulatability. We require that a simulator can simulate a random continuation from any partial interaction in a computationally indistinguishable way to efficient, non-uniform distinguishers who get only the prover's view but *not* the verifier's coins. The partial interaction includes not only the transcripts, but also both parties' coins tossed so far. Therefore, to define the computational simulatable property, we need to specify both parties' coins in each round. Recall that for a m-round protocol $\langle \mathsf{P}, \mathsf{V} \rangle$, the verifier's (resp., the prover's) messages are denoted by $v_1, \ldots, v_m$ (resp., $p_1, \ldots, p_m$). We use $c_1, \ldots, c_m$ and $t_1, \ldots, t_m$ to denote the verifier and the prover's coins in each round, respectively. Hence, a ℓ-round partial interaction of $\langle \mathsf{P}, \mathsf{V} \rangle(x)$ can be described by $(t_{[\ell]}, c_{[\ell]}, x, p_{[\ell]}, v_{[\ell]})$, and P's and V's views are $(t_{[\ell]}, x, p_{[\ell]}, v_{[\ell]})$ and $(c_{[\ell]}, x, p_{[\ell]}, v_{[\ell]})$, respectively. Recall that $p_{[\ell]}$ denotes $(p_1, \ldots, p_\ell)$.

Informally, we would like to say that given a prover strategy P^*, there is a simulator S such that starting from any partial interaction $(t_{[\ell]}, c_{[\ell]}, x, p_{[\ell]}, v_{[\ell]})$, when P^* continues

the interaction with either the actual verifier V or the simulator S, the resulting views of the prover (without V's coins and verdict) are computationally indistinguishable.

Definition 3.15 (Computational Simulatable Verifier) *A verifier* V *is said to be* **computationally simulatable** *if for every non-uniform* PPT *prover strategy* P^* *there exists a* PPT *simulator* S *such that for every* PPT *distinguisher* D*, the following holds.*

There exists a negligible function $\mathsf{ngl} : \mathbb{N} \to (0,1)$ *such that for sufficiently large* s*, for every common input* x *with security parameter* s*, for every partial interaction* $(t_{[\ell]}, c_{[\ell]}, x, p_{[\ell]}, v_{[\ell]})$*,* D *cannot distinguish the following two distributions with probability greater than* $\mathsf{ngl}(s)$*.*

- *The prover* P^**'s view* $(t_{[m]}, x, p_{[m]}, v_{[m]})$ *after continuing the interaction with* V *from* $(t_{[\ell]}, c_{[\ell]}, x, p_{[\ell]}, v_{[\ell]})$*.*
- *The simulator* S*'s output* $(t'_{[m]}, x, p'_{[m]}, v'_{[m]})$*, when we run* S *on the prover's view* $(t_{[\ell]}, x, p_{[\ell]}, v_{[\ell]})$ *of the partial interaction.*

Namely, we have

$$|\Pr[\mathsf{D}(t_{[m]}, c_{[\ell]}, x, p_{[m]}, v_{[m]}) = 1] - \Pr[\mathsf{D}(t'_{[m]}, c_{[\ell]}, x, p'_{[m]}, v'_{[m]}) = 1]| \leq \mathsf{ngl}(s).^9$$

Note that the verifier V*'s decision bit as well as* V*'s coins* $c_{[m]}$ *are* not *included in the distributions. When the above holds for a specific* PPT *prover* P^**, we say* V *is* **computationally simulatable w.r.t.** P^**.*

Note that in the above definition, we require that the indistinguishability holds for *any* partial interaction of $\langle \mathsf{P}^*, \mathsf{V} \rangle$ and holds against non-uniform distinguishers, which seem to be strong requirements. Nevertheless, we will show that, running a protocol under a fully homomorphic encryption scheme, when it is done properly, can satisfy the above definition.

Theorem Statement. We proceed to state our direct product theorem for computationally simulatable protocols below. The theorem says that n-fold parallel repetition with direct product verifiers decreases the soundness error of computationally simulatable protocols from δ to $\delta^{n/2}$, which almost matches the information theoretical bound of δ^n.

Theorem 3.16 *Let* $\langle \mathsf{P}, \mathsf{V} \rangle$ *be a computationally simulatable protocol with input domain* Λ*,* $\delta : \Lambda \to [0,1]$ *and* $n : \mathbb{N} \to \mathbb{N}$ *efficiently computable functions with*

[9] Note that since the verifier's coins $c_{[\ell]}$ in the partial interaction is fixed and the distinguisher D is non-uniform, D has access to $c_{[\ell]}$ through non-uniform advice. Hence, we include in the distribution explicitly for clarity. The point is that V's coins after ℓ-th round are not given to D.

$n \leq \text{poly}(s)$. *If* $\langle \mathsf{P}, \mathsf{V} \rangle$ *has soundness error* δ*, then its* n*-fold parallel repetition with direct product verifier* $\langle \mathsf{P}^n, \mathsf{V}^{n,n} \rangle$ *has soundness error* $\delta^{n/2} + \mathsf{ngl}$*, where* ngl *denotes a negligible function in the security parameter* s*.*

We remark that the bound $\delta^{n/2}$ is tight for our reduction prover strategy. It is an interesting open question to see if $\delta^{n/2}$ is optimal, since improving the bound to δ^n would imply an essentially tight Chernoff-type theorem for computationally simulatable protocols. In the following section, we prove Theorem 3.16 by defining a black-box reduction prover strategy, which exploits ideas from the reductions for public-coin protocols and three-message protocols in previous sections.

3.4.2 Reduction Prover Strategy

In this section, we define an efficient reduction prover strategy P^* for proving Theorem 3.16. Our goal is to show that if there exists a parallel prover strategy P^{n*} with at least non-negligible success probability $\delta^n + \xi$, then our reduction prover strategy P^* can convince a single instance verifier V with probability at least δ^2.[10]

As discussed, our reduction prover strategy is a variant of the rejection sampling strategy P^*_{rej} for public-coin protocols, so let us first recall the rejection sampling strategy P^*_{rej} from Figure 3.3: P^*_{rej} interacts with a public-coin verifier V by simulating the interaction of a *deterministic* parallel prover strategy P^{n*} and a parallel verifier $\mathsf{V}^{n,n}$, where V plays a random subverifier V_i of $\mathsf{V}^{n,n}$, and P^*_{rej} plays P^{n*} and the remaining $n-1$ subverifiers V_{-i} of $\mathsf{V}^{n,n}$. Since P^{n*} is deterministic, the interaction of $\langle \mathsf{P}^*_{rej}, \mathsf{V} \rangle$ amounts to determining $\mathsf{V}^{n,n}$'s messages $\vec{v}_1, \ldots, \vec{v}_m$, where V selects $v_{1,i}, \ldots, v_{m,i}$ uniformly at random, and P^*_{rej} selects $\vec{v}_{1,-i}, \ldots, \vec{v}_{m,-i}$ by rejection sampling. Namely, at each round j, P^*_{rej} repeatedly samples random continuations $(\vec{v}_{j,-i}, \vec{v}_{j+1}, \ldots, \vec{v}_m)$ of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ until a successful continuation is found, where a successful continuation means $\mathsf{V}^{n,n}$ accepts at the end of the interaction. Then P^*_{rej} selects the corresponding $\vec{v}_{j,-i}$ in the successful continuation.

Let us look at closely whether the above rejection sampling strategy P^*_{rej} can be implemented when V is not public-coin. First, P^*_{rej} can still interact with V by simulating the interaction of a deterministic P^{n*} and a $\mathsf{V}^{n,n}$, but now P^*_{rej} and V jointly select the random coins $\vec{c}_1, \ldots, \vec{c}_m$ of $\mathsf{V}^{n,n}$, instead of the messages $\vec{v}_1, \ldots, \vec{v}_m$. In the interaction, V selects V_i's coins $c_{1,i}, \ldots, c_{m,i}$ uniformly at random, and the issue is whether P^*_{rej} can efficiently select $\vec{c}_{1,-i}, \ldots, \vec{c}_{m,-i}$ by rejection sampling. This requires to sample random continuations of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ and identify successful continuations efficiently. As discussed in Section 3.4.1, for private-coin protocols, random continuations means continuations of interaction $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ conditioned on the coins $(\vec{c}_1, \ldots, \vec{c}_\ell)$ tosses in previous rounds, as opposed to conditioned on the transcript $(\vec{v}_1, \ldots, \vec{v}_\ell)$.

[10] As we will see later, it is more convenient to use parameters δ^2 and δ^n than δ and $\delta^{n/2}$.

Consider the first round of the interaction, where V sends $v_{1,i} = \mathsf{V}(c_{1,i})$ to P^*_{rej}, and P^*_{rej} needs to select $\vec{c}_{1,-i}$. It is easy for P^*_{rej} to sample coins $(\vec{c}_{1,-i}, \vec{c}_2, \ldots, \vec{c}_m)$ and simulate P^{n*} and $n-1$ internal subverifiers V_{-i}. However, in general, P^* cannot simulate the external subverifier V_i due to the lack of information about the external subverifier V_i's coins $c_{1,i}$.

Intuitively, when the verifier is computationally simulatable, P^*_{rej} can use a simulator S to simulate the external verifier $\mathsf{V} = \mathsf{V}_i$. Indeed, consider the "naive" prover strategy P^*_{naive} defined in Figure 3.7, who selects a random coordinate i and interacts with V by playing P^{n*} and V_{-i} honestly. Note that the interaction of P^*_{naive} with V simulates the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ honestly, and a partial interaction of the first ℓ rounds of $\langle \mathsf{P}^*_{naive}, \mathsf{V} \rangle$ can be described by $(i; \vec{c}_1, \ldots, \vec{c}_\ell)$. By the computational simulatability, there exists an efficient simulator S that can simulate random continuations of $\langle \mathsf{P}^*_{naive}, \mathsf{V} \rangle$ from a partial interaction $(i; \vec{c}_1, \ldots, \vec{c}_\ell)$, which corresponds to random continuations of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ from the corresponding partial interaction $(\vec{c}_1, \ldots, \vec{c}_\ell)$ with V_i being the external subverifier.

Prover Strategy $\mathsf{P}^*_{naive}(x, n)$
/* P^*_{naive} interacts with V and is given oracle access to P^{n*}. */
/* The interaction of $\langle \mathsf{P}^*_{naive}, \mathsf{V} \rangle$ simulates the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ honestly. */

- P^*_{naive} selects a coordinate $i \in_R [n]$ uniformly at random.
- For each round $j \in [m]$, upon receiving $v_{j,i} = v_j$ from V, P^*_{naive} samples coins $\vec{c}_{j,-i}$ uniformly at random, runs V_{-i} to generate $\vec{v}_{j,-i}$, runs P^{n*} to generate $\vec{p}_j$, and sends $p_{j,i}$ to V.

Figure 3.7: A "naive" prover strategy P^*_{naive} such that the interaction of $\langle \mathsf{P}^*_{naive}, \mathsf{V} \rangle$ simulates the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ honestly.

Note that the simulator S only generates a (computationally indistinguishable) view of P^*_{naive}, which consists of the internal subverifiers V_{-i}'s view, but not the external verifier V_i's view. Hence, P^*_{rej} can only compute the verdict bits of the internal subverifiers, but not the verdict bit of the external verifier. However, to check if a continuation is successful, P^*_{rej} needs to check the verdict bits of all subverifiers. Hence, the computational simulatablility property is not sufficient to imply that the rejection sampling strategy of Figure 3.3 is efficiently implementable. Nevertheless, if the definition of successful continuation depends only on the verdict bits of the internal subverifiers but not on the external subverifier's verdict, then the corresponding rejection sampling strategy becomes efficiently implementable for computationally simulatable protocols.

Note that this situation is similar to the situation of three-message protocols in Section 3.3, where a reduction prover strategy P^* can only base his decision on the internal subverifiers' verdict bits, as the external subverifier's verdict bit is unknown.

In Section 3.3, we considered a naive strategy, where P^* only checks whether all internal subverifiers accept. We first observed that this naive strategy does not work when there are"bad correlations" in the "success pattern" of P^{n*}. Then, we presented a correlation reduction procedure in Figure 3.5, which exploits such bad correlations to convert a parallel prover strategy P^{n*} to another parallel prover strategy $\mathsf{P}^{n'*}$ with smaller $n' < n$. We proved that correlation reduction eliminates bad correlations, and the naive strategy works when applied to the $\mathsf{P}^{n'*}$ returned by the correlation reduction procedure.

We observe that the same idea can be applied to computationally simulatable protocols as well, although the analysis is very different and much more involved. First, we consider an analogous naive modification of the rejection sampling strategy, where the definition of successful continuation is changed to only require that all internal subverifiers accept. For notationally simplicity, let us still use P^*_{rej} to denote the modified rejection sampling strategy. Such a P^*_{rej} is efficeintly implementable for computationally simulatable protocols, but does not work because of the same issue of "bad correlations", as illustrated by the following example copied from Section 3.3.

Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a two-message protocol (which is clearly computationally simulatable). Consider a deterministic parallel prover P^{n*} such that when interacting with $\mathsf{V}^{n,n}$, (i) P^{n*} can convince the parallel verifier $\mathsf{V}^{n,n}$ with probability δ^n, and (ii) for every $i \in [n]$, P^{n*} can convince all except the i-th subverifier with probability $(1-\delta^n)/n$. In the reduction, the modified P^*_{rej} selects a random coordinate $i \in [n]$, V selects V_i's coin c_i, and P^*_{rej} samples random continuations $(c_i, \vec{c}_{-i})$. Observe that for this P^{n*}, P^*_{rej} convinces V iff the random continuation $(c_i, \vec{c}_{-i})$ falls in case (i). Intuitively, one can expect the random continuation $(c_i, \vec{c}_{-i})$ to fall in each case with probability proportional to $p = \delta^n$ and $q = (1-\delta^n)/n$, respectively, and hence P^*_{rej} may succeed with probability only $p/(p+q) \approx n\delta^n \ll \delta$. Fortunately, as before, a natural generalization of the correlation reduction procedure in Figure 3.5 can be applied to eliminate bad correlations, and the modified rejection sampling strategy works when applies to the $\mathsf{P}^{n'*}$ returned by the generalized correlation reduction.

3.4.3 Correlation Reduction

Again, we briefly recall the key idea of correlation reduction: one can convert a parallel prover P^{n*} (interacting with $\mathsf{V}^{n,n}$) with such bad correlations to a parallel prover $\mathsf{P}^{(n-1)*}$ (interacting with $\mathsf{V}^{n-1,n-1}$) that has success probability higher than δ^{n-1}. To illustrate, in the above example, a such $\mathsf{P}^{(n-1)*}$ can simply interact with $\mathsf{V}^{n-1,n-1}$ by simulating the interaction of P^{n*} and $\mathsf{V}^{n,n}$, where $\mathsf{P}^{(n-1)*}$ simulates P^{n*} and the first coordinate V_1 honestly, and $\mathsf{V}^{n-1,n-1}$ plays the remaining coordinates V_{-1} of $\mathsf{V}^{n,n}$. Hence, $\Pr[\langle \mathsf{P}^{(n-1)*}, \mathsf{V}^{n-1,n-1} \rangle(x) = 1] = \Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-1}] = (1-\delta^n)/n + \delta^n$.

Formal descriptions of the generalized correlation reduction CR and our final reduction prover strategy P^* can be found in Figure 3.8 and 3.9, respectively. At a high level, our P^* first applies CR to obtain a prover strategy $\mathsf{P}^{n'*}$, and then applies the

modified rejection sampling strategy P^*_{rej} to $\mathsf{P}^{n'*}$. (Again, we use the same name CR and P^*_{rej} for notational simplicity.)

Sub-Routine $\mathsf{FindC}(\mathsf{P}^{n*}, n, i, \delta, \xi)$
/* Find correlations in the success pattern of P^{n*} on coordinate $i \in [n]$. */
/* Return $\mathsf{P}^{(n-1)*}$ if such correlation is found. Otherwise, return $\perp$ */

Repeat the following at most $M_1 = O\left(\frac{n}{\xi} \cdot \log \frac{n}{\xi}\right)$ times:

- Sample random $c^*_{[m],i}$ and estimate $p(c^*_{[m],i}) \stackrel{\text{def}}{=} \Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i} | c_{[m],i} = c^*_{[m],i}]$ by sampling. Namely, randomly sample $M_2 = O\left(\frac{n^2}{\xi^2} \cdot \log \frac{n}{\xi}\right)$ independent copies of $\vec{c}_{[m],-i}$'s, check if P^{n*} convinces V_{-i} on $(c^*_{[m],i}, \vec{c}_{[m],-i})$, and compute an estimator $\hat{p}(c^*_{[m],i}) = |\{\vec{c}_{[m],-i} : \mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i} \text{ on } (c^*_{[m],i}, \vec{c}_{[m],-i})\}|/M_2$.

- If $\hat{p}(c^*_{[m],i}) \geq \delta^{n-1} + (1 - (1/n)) \cdot \xi$, then return a parallel prover $\mathsf{P}^{(n-1)*}(c^*_i)$ for $\mathsf{V}^{n-1,n-1}$ defined as follows: $\mathsf{P}^{(n-1)*}(c^*_{[m],i})$ interacts with $\mathsf{V}^{n-1,n-1}$ by simulating the interaction of P^{n*} and $\mathsf{V}^{n,n}$, where $\mathsf{P}^{(n-1)*}(c^*_{[m],i})$ simulates P^{n*} and the i-th coordinate V_i with random coins $c^*_{[m],i}$ honestly, and $\mathsf{V}^{n-1,n-1}$ plays the remaining coordinates V_{-i} of $\mathsf{V}^{n,n}$.

Return $\perp$ after M_1 (failure) attempts.

$\mathsf{CR}(\mathsf{P}^{n*}, n, \delta, \xi)$
/* Implicitly, there are a PPT verifier V and an input x as part of the input. */
/* Iteratively exploit correlation in the success pattern of P^{n*} to obtain $\mathsf{P}^{(n-1)*}$.*/

Iteratively, apply FindC to every coordinate $i \in [n]$, until $n = 1$ or FindC returns $\perp$ for all coordinates, namely

- Call $\mathsf{FindC}(\mathsf{P}^{n*}, n, i, \delta, \xi)$ for every $i \in [n]$. If there exists a coordinate i such that $\mathsf{FindC}(\mathsf{P}^{n*}, n, i, \delta, \xi)$ returns $\mathsf{P}^{(n-1)*}$, then set $\xi \leftarrow \left(1 - \frac{1}{n}\right) \cdot \xi$ and $n \leftarrow n - 1$ (so that P^{n*} refers to the prover strategy returned by FindC).

Return the final P^{n*}.

Figure 3.8: Generalized correlation reduction for direction product verifiers.

Note that the subroutine FindC defined here and in Figure 3.5 are exactly the same, although the notation is slightly different. Here, the coins of $\mathsf{V}^{n,n}$ are denoted by $\vec{c}_{[m]} = (\vec{c}_1, \ldots, \vec{c}_m)$, and $c_{[m],i} = (c_{1,i}, \ldots, c_{m,i})$ denotes V_i's whole coin tosses. On the other hand, the difference between the correlation reduction CR defined here and in Figure 3.5 is that, here, CR tries to find correlation for every coordinate $i \in [n]$, as opposed to only the first coordinate in Figure 3.5. This simple generalization is needed

for an intuitive reason: the rejection sampling strategy P^*_{rej} embeds V in a random coordinate $i \in [n]$, and in contrast, the reduction prover strategy for three-message protocols always embeds V in the first coordinate.

As in Section 3.3.1, we formalize the property of CR in the following lemma, which is a straightforward generalization of Lemma 3.13. The lemma says that when we apply CR to a prover strategy P^{n*} with success probability at least roughly δ^n, with high probability, CR outputs a "*good*" prover strategy $\mathsf{P}^{n'*}$ with success probability at least roughly $\delta^{n'}$ such that either (1) $n' = 1$, in which case we obtain a good single instance prover strategy, or (2) for every coordinate $i \in [n']$, we have $\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}_{-i}] \lesssim \delta^{n'-1}$.[11] The latter case informally means that there are no bad correlations in the success pattern of $\mathsf{P}^{n'*}$.

Let a PPT verifier V (not necessarily computationally simulatable), an input $x \in \{0,1\}^*$, parameters $n \in N$, $\delta, \xi \in (0,1)$, and a deterministic parallel prover P^{n*} for $\mathsf{V}^{n,n}$ be given as in Figure 3.8.

Lemma 3.17 *If P^{n*} has success probability at least $(\delta^n + \xi)$ on input x, then with probability at least $(1 - (\xi/10n))$ over the randomness of CR, $\mathsf{CR}(\mathsf{P}^{n*}, n, \delta, \xi)$ outputs a* good *deterministic prover strategy $\mathsf{P}^{n'*}$ satisfying the following properties.*

- $\Pr[\langle \mathsf{P}^{n'*}, \mathsf{V}^{n',n'} \rangle(x) = 1] \geq \delta^{n'} + ((10n' - 1)/10n) \cdot \xi$.
- *Either $n' = 1$, or for every $i \in [n']$,*

$$\Pr[\mathsf{P}^{n'*} \textit{ convinces } \mathsf{V}_{-i}] \leq \delta^{n'-1} + \frac{10(n'-1)+2}{10n} \cdot \xi.$$

Furthermore, $\mathsf{CR}(\mathsf{P}^{n*}, n, \delta, \xi)$ can be implemented with oracle access to P^{n*} with runtime $\mathrm{poly}(|x|, n, \xi^{-1})$, and the output $\mathsf{P}^{n'*}$ can be implemented in time $\mathrm{poly}(|x|, n)$ given oracle access to P^{n*}.

The above lemma can be proved by a straightforward generalization of the proof of Lemma 3.13. We omit the proof to avoid repetitive arguments. The lemma allows us to analyze the rejection sampling strategy assuming that it is given a good parallel prover P^{n*} that satisfies the conclusion of Lemma 3.17.

3.4.4 Rejection Sampling

Our reduction prover strategy P^* for proving Theorem 3.16 is given in Figure 3.9. We remark that when we restrict to the case of three-message protocols, P^* is very similar to the reduction prover strategy for three-message protocols in Section 4.2.2,

[11] In Lemma 3.13, we have upper bounds on the probability of $\mathsf{P}^{n'*}$ convinces V_{-1} even conditioned on every $c_1 = c_1^*$. The same statement holds, but here, we only need the stated weaker assertion.

Prover Strategy $\mathsf{P}^*(x, n, \delta, \xi)$
/* P^* interacts with V and is given oracle access to P^{n*}. */

- Let $\alpha = \delta^n + \xi$. If $\delta^n \leq \xi$, update the value of δ and ξ: set $\delta \leftarrow (\alpha/2)^{1/n}, \xi \leftarrow \alpha/2$.
- P^* applies CR to P^{n*} to obtain $\mathsf{P}^{n'*} = \mathsf{CR}(\mathsf{P}^{n*}, n, \delta, \xi)$.
- If $n' = 1$, P^* interacts with V by running $\mathsf{P}^{n'*}$. Otherwise, P^* performs the following rejection sampling strategy with parameter $\mathsf{P}_{rej}^{*(\mathsf{P}^{n'*})}(x, n', \delta, (n'/n) \cdot \xi)$.

Prover Strategy $\mathsf{P}^*_{rej}(x, n, \delta, \xi)$
/* P^*_{rej} interacts with V and is given oracle access to P^{n*}. */
/* Let $\mathsf{P}^{*(\mathsf{P}^{n*})}_{naive}$ be the naive prover strategy defined in Figure 3.7, and S the corresponding simulator for P^*_{naive}. */

- P^*_{rej} selects a coordinate $i \in_R [n]$ uniformly at random.
- For each round $j \in [m]$, upon receiving $v_{j,i} = v_j$ from V, P^*_{rej} repeats the following at most $M = O\left(\frac{mn}{\delta^n \xi}\right)$ times.
  - P^*_{rej} first samples coins $\vec{c}_{j,-i}$ uniformly at random.
  - P^*_{rej} uses S to generate a random continuation from current partial interaction $(i; \vec{c}_1, \ldots, \vec{c}_j)$ of $\langle \mathsf{P}^*_{naive}, \mathsf{V} \rangle$ (which emulates $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ honestly), and checks if it is a *successful continuation* in the sense that all internal subverifiers accept at the end of the interaction.
  - If the continuation is successful, then P^*_{rej} selects the coins $\vec{c}_{j,-i}$, runs V_{-i} to generate $\vec{v}_{j,-i}$, runs P^{n*} to generate $\vec{p}_j$, and sends the corresponding $p_{j,i}$ to V.
- If all the M attempts fail, P^*_{rej} aborts.

Figure 3.9: Rejection sampling strategy P^*_{rej} for computationally simulatable protocols.

and one can slightly modify the analysis in Section 3.3.3 to show that P^* can succeed with probability at least δ as well. However, as a black-box reduction for general computationally simulatable protocols, the bound δ^2 is tight for P^*.

We will show that if a deterministic parallel prover strategy P^{n*} can succeed with probability $\delta^n + \xi$, then the reduction prover strategy P^*, given oracle access to P^{n*}, can succeed with probability at least δ^2, where ξ is a (non-negligible) slackness parameter. The core part of our analysis is to lower bound the success probability of the rejection sampling strategy P^*_{rej} when it is applied to a "good" parallel prover strategy P^{n*} returned by CR. The analysis of P^*_{rej} consists of the following two steps.

Step 1. We first consider a "perfectified" version of the rejection sampling strategy, where P^*_{rej} can get perfect samples of random continuations of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$, as opposed to computationally indistinguishable samples obtained from a simulator.

One way to formalize this perfectified strategy is to consider an "omniscient" third party O_{V}, who can generate perfect samples of random continuations for P^*_{rej}. More precisely, during the interaction of P^*_{rej} and V, O_{V} keeps track the view V (including V's coins), and P^*_{rej} is allowed to query O_{V} as follows. P^*_{rej} can send O_{V} the prover strategy P^*_{naive} together with a partial view of P^*_{naive} (that is consistent with the current view of V), and O_{V} returns a sample of random continuation of $\langle \mathsf{P}^*_{naive}, \mathsf{V} \rangle$ from the partial interaction defined by the views of P^*_{naive} and V.

For notationally convenience, we use $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$ to denote the perfectified rejection sampling strategy. More generally, for a prover strategy P^* who uses some simulator S to generate random continuations, we use $\mathsf{P}^{*(O_{\mathsf{V}})}$ to denote the perfectified version of P^*, who queries O_{V} to get perfect samples of random continuations, instead of using S. We note that O_{V} is defined for the purpose of analysis and is *not* an oracle in the usual sense.

A formal description of the perfectified rejection sampling strategy $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$ can be found in Figure 3.10. We will analyze the success probability of $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$ when it is given a good parallel prover P^{n*} that satisfies the conclusion of Lemma 3.17. The analysis follows closely the inductive analysis of the rejection sampling strategy for public-coin protocols in Section 3.2.2, but using a different induction hypothesis.

Step 2. We then show that the success probability of the actual P^*_{rej}, who uses a simulator to generate random continuations, is close to the success probability of the perfectified $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$. More generally, we will prove by a hybrid argument that for any PPT prover strategy P^*, who uses a simulator S to generate samples of random continuations, the success probability of P^* is close to that of a corresponding perfectified prover strategy $\mathsf{P}^{*(O_{\mathsf{V}})}$, who uses the oracle O_{V} to generate such samples with correct distribution.

3.4.5 Analysis of Perfectified Rejection Sampling Strategy $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$

Recall that the main difference between the rejection sampling strategies P^*_{rej} for public-coin protocols and computationally simulatable protocols is in the definition of *successful* continuations. Previously, a continuation is successful if P^{n*} convinces $\mathsf{V}^{n,n}$. Here, we only require that P^{n*} convinces all internal subverifiers V_{-i}. To analyze $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$, we use the same inductive analysis as in Section 3.2.2 with a different induction hypothesis.

As before, it is instructive to first analyze an ideal version $\mathsf{P}^{*(O_{\mathsf{V}})}_{ideal}$ of $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$, where there are no sampling errors in the sense that $\mathsf{P}^{*(O_{\mathsf{V}})}_{ideal}$ can always find a random suc-

Prover Strategy $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}(x, n, \delta, \xi)$
/* $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$ interacts with V and is given oracle access to P^{n*}. */

- $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$ selects a coordinate $i \in_R [n]$ uniformly at random.
- For each round $j \in [m]$, upon receiving $v_{j,i} = v_j$ from V, $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$ repeats the following at most $M = O\left(\frac{mn}{\delta^n \xi}\right)$ times.
  - $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$ first samples coins $\vec{c}_{j,-i}$ uniformly at random.
  - $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$ uses the oracle O_{V} to generate a random continuation of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle$ (by querying O_{V} with P^{*}_{naive}), and checks if it is a *successful continuation* in the sense that all *internal* subverifiers accept at the end of the interaction.
  - If the continuation is successful, then $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$ selects the coins $\vec{c}_{j,-i}$, runs V_{-i} to generate $\vec{v}_{j,-i}$, runs P^{n*} to generate $\vec{p}_j$, and sends the corresponding $p_{j,i}$ to V.
- If all the M attempts fail, $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$ aborts.

Figure 3.10: perfectified rejection sampling strategy $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$ for computationally simulatable protocols.

cessful continuation if there exists one.

Analysis of the Ideal Version $\mathsf{P}^{*(O_{\mathsf{V}})}_{ideal}$ of Perfectified $\mathsf{P}^{*(O_{\mathsf{V}})}_{rej}$

We shall show that if a deterministic P^{n*} satisfies (1) $\Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}^{n,n}] \geq \delta^n$, and (2) for every coordinate $i \in [n]$, $\Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i}] \leq \delta^{n-1}$, then $\mathsf{P}^{*(O_{\mathsf{V}})}_{ideal}$, given oracle access to P^{n*}, can convince V with probability at least δ^2. We use the following induction hypothesis for $\mathsf{P}^{*(O_{\mathsf{V}})}_{ideal}$.

Induction Hypothesis for $\mathsf{P}^{*(O_{\mathsf{V}})}_{ideal}$. For every $j \in [m]$ and every partial interaction $(\vec{c}_1, \ldots, \vec{c}_j)$, the following inequality holds.[12]

$$\begin{aligned}
&\prod_{i=1}^{n} \Pr[\langle \mathsf{P}^{*(O_{\mathsf{V}})}_{ideal}, \mathsf{V}\rangle(x) = 1 | i, \vec{c}_1, \ldots, \vec{c}_j] \\
&\quad \geq \quad \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1 | \vec{c}_1, \ldots, \vec{c}_j] \cdot \\
&\qquad \left(\prod_{i=1}^{n} \Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_i | (\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i}), \vec{c}_1, \ldots, \vec{c}_j] \right) \\
&\quad = \quad \frac{\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1 | \vec{c}_1, \ldots, \vec{c}_j]^{n+1}}{\prod_{i=1}^{n} \Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i} | \vec{c}_1, \ldots, \vec{c}_j]}.
\end{aligned}$$

Recall that the induction hypothesis used in Section 3.2.2 is of the form

$$\prod_{i=1}^{n} \Pr[\langle \mathsf{P}^{*(O_{\mathsf{V}})}_{ideal}, \mathsf{V}\rangle(x) = 1 | i, \vec{c}_1, \ldots, \vec{c}_j] \geq \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1 | \vec{c}_1, \ldots, \vec{c}_j].$$

In comparison, here we pay extra factors of

$$\prod_{i=1}^{n} \Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_i | (\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i}), \vec{c}_1, \ldots, \vec{c}_j],$$

intuitively, since in the rejection sampling, $\mathsf{P}^{*(O_{\mathsf{V}})}_{ideal}$ makes his decision based only on whether P^{n*} convinces V_{-i}.

We introduce some shorthand notations below to simplify the expressions. We will use $\bar{h} = (\vec{c}_1, \ldots, \vec{c}_j)$ to denote a partial interaction and define

$$\begin{aligned}
\gamma(\bar{h}) = \gamma(\vec{c}_1, \ldots, \vec{c}_j) &\stackrel{\text{def}}{=} \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1 | \vec{c}_1, \ldots, \vec{c}_j], \\
\gamma_i(\bar{h}) = \gamma(\vec{c}_1, \ldots, \vec{c}_j) &\stackrel{\text{def}}{=} \Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i} | \vec{c}_1, \ldots, \vec{c}_j], \text{ for every } i \in [n], \text{ and} \\
\eta_i(\bar{h}) = \eta_i(\vec{c}_1, \ldots, \vec{c}_j) &\stackrel{\text{def}}{=} \Pr[\langle \mathsf{P}^{*(O_{\mathsf{V}})}_{ideal}, \mathsf{V}\rangle(x) = 1 | i, \vec{c}_1, \ldots, \vec{c}_j] \text{ for every } i \in [n].
\end{aligned}$$

With the above shorthand notation, the inductive hypothesis becomes

$$\prod_{i=1}^{n} \eta_i(\bar{h}) \geq \frac{\gamma^{n+1}(\bar{h})}{\prod_{i=1}^{n} \gamma_i(\bar{h})}.$$

We will prove the induction backward on $j = m, m-1, \ldots, 1, 0$ and apply the induction hypothesis with $j = 0$ to complete the analysis. We present the latter step of applying the induction hypothesis first.

[12] We use a convention that $0/0 = 0$.

Induction Hypothesis for $j = 0 \Rightarrow \Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}_{ideal}, \mathsf{V}\rangle(x) = 1] \geq \delta^2$. When $j = 0$, the induction hypothesis says that

$$\prod_{i=1}^{n} \Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}_{ideal}, \mathsf{V}\rangle(x) = 1|i] \geq \frac{\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1]^{n+1}}{\prod_{i=1}^{n} \Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i}]} \geq \frac{\delta^{n\cdot(n+1)}}{\delta^{(n-1)\cdot n}} = \delta^{2n}.$$

Recalling that $\mathsf{P}^{*(O_\mathsf{V})}_{ideal}$ selects the coordinate i uniformly at random, we have

$$\Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}_{ideal}, \mathsf{V}\rangle(x) = 1] = \frac{1}{n}\sum_{i=1}^{n} \Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}_{ideal}, \mathsf{V}\rangle(x) = 1|i].$$

Putting them together by applying the Arithmetic-Mean-Geometric-Mean Inequality gives the desired lower bound on the success probability of $\mathsf{P}^{*(O_\mathsf{V})}_{ideal}$.

$$\begin{aligned}
\Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}_{ideal}, \mathsf{V}\rangle(x) = 1] &= \frac{1}{n}\sum_{i=1}^{n} \Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}_{ideal}, \mathsf{V}\rangle(x) = 1|i] \\
&\geq \left(\prod_{i=1}^{n} \Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}_{ideal}, \mathsf{V}\rangle(x) = 1|i]\right)^{1/n} \\
&\geq \delta^2.
\end{aligned}$$

■

We proceed to verify the base case $j = m$ of the induction, where we condition on a *complete* interaction $(\vec{c}_1, \ldots, \vec{c}_m)$ and there is no randomness.

Base Case of the Induction. For every complete interaction $\bar{h} = (\vec{c}_1, \ldots, \vec{c}_m)$, we have

$$\prod_{i=1}^{n} \Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}_{ideal}, \mathsf{V}\rangle(x) = 1|i, \bar{h}] \geq \frac{\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1|\bar{h}]^{n+1}}{\prod_{i=1}^{n} \Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i}|\bar{h}]}.$$

By inspection, the left-hand-side is 1 iff P^{n*} convinces V_i for every $i \in [n]$ on interaction $\bar{h}$, which is equivalent to P^{n*} convinces $\mathsf{V}^{n,n}$ on $\bar{h}$, which is the case iff the right-hand-side is 1. ■

Finally, we prove the following induction step.

Induction Step. For every $j \in [m]$ and every partial interaction $\bar{h} = (\vec{c}_1, \ldots, \vec{c}_{j-1})$, the following holds. Suppose that for all coins $\vec{c}_j$, it is true that

$$\prod_{i=1}^{n} \eta_i(\bar{h}, \vec{c}_j) \geq \frac{\gamma^{n+1}(\bar{h}, \vec{c}_j)}{\prod_{i=1}^{n} \gamma_i(\bar{h}, \vec{c}_j)}.$$

Then we have

$$\prod_{i=1}^{n} \eta_i(\bar{h}) \geq \frac{\gamma^{n+1}(\bar{h})}{\prod_{i=1}^{n} \gamma_i(\bar{h})}.$$

The above induction step together with the base case prove the induction. As in the previous analysis of P^*_{ideal} in Section 3.2.2, the first step to prove the induction step is to express the probabilities $\gamma(\bar{h}), \gamma_i(\bar{h})$, and $\eta_i(\bar{h})$ in terms of $\gamma(\bar{h}, \vec{c}_j), \gamma_i(\bar{h}, \vec{c}_j)$ and $\eta_i(\bar{h}, \vec{c}_j)$. By definition, we have

$$\gamma(\bar{h}) = \mathop{\mathrm{E}}_{\vec{c}_j}\left[\gamma(\bar{h}, \vec{c}_j)\right] \text{ and } \gamma_i(\bar{h}) = \mathop{\mathrm{E}}_{\vec{c}_j}\left[\gamma_i(\bar{h}, \vec{c}_j)\right].$$

Now, since for $\mathsf{P}^{*(O_\mathsf{V})}_{ideal}$, a successful continuation only requires the acceptance of the internal subverifiers, the formula of $\eta_i(\bar{h})$'s is different from that of P^*_{ideal}.

Claim 3.18 *For every $i \in [n]$, $j \in [m]$, and partial interaction $\bar{h} = (\vec{c}_1, \ldots, \vec{c}_{j-1})$,*

$$\eta_i(\bar{h}) \;=\; \mathop{\mathrm{E}}_{\vec{c}_j}\left[\frac{\gamma_i(\bar{h}, \vec{c}_j) \cdot \eta_i(\bar{h}, \vec{c}_j)}{\gamma_i(\bar{h}, c_{j,i})}\right].$$

Proof of claim: Recall that V plays the random strategy and $\mathsf{P}^{*(O_\mathsf{V})}_{ideal}$ plays the rejection sampling strategy. Let $\Pr[c_{j,i}], \Pr[\vec{c}_{j,-i}]$ denote the uniform probability on $c_{j,i}, \vec{c}_{j,-i}$, respectively. For a given $\vec{c}_j = (c_{j,i}, \vec{c}_{j,-i})$, observe that V plays $c_{j,i}$ with probability $\Pr[c_{j,i}]$. By Bayes Rule, $\mathsf{P}^{*(O_\mathsf{V})}_{ideal}$ plays $\vec{c}_{j,-i}$ with probability

$$\begin{aligned}
&\Pr[\vec{c}_{j,-i} | \bar{h}, c_{j,i}, \mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i}] \\
&= \frac{\Pr[\vec{c}_{j,-i}] \cdot \Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i} | \bar{h}, \vec{c}_j]}{\Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-i} | \bar{h}, c_{j,i}]} \\
&= \Pr[\vec{c}_{j,-i}] \cdot \frac{\gamma_i(\bar{h}, \vec{c}_j)}{\gamma_i(\bar{h}, c_{j,i})},
\end{aligned}$$

and by definition, $\mathsf{P}^{*(O_\mathsf{V})}_{ideal}$ can succeed with probability $\eta_i(\bar{h}, \vec{c}_j)$. Hence, we have

$$\begin{aligned}
\eta_i(\bar{h}) &= \sum_{\vec{c}j} \Pr[c_{j,i}] \cdot \Pr[\vec{c}_{j,-i}] \cdot \frac{\gamma_i(\bar{h}, \vec{c}_j)}{\gamma_i(\bar{h}, c_{j,i})} \cdot \eta_i(\bar{h}, \vec{c}_j) \\
&= \mathop{\mathrm{E}}_{\vec{c}_j}\left[\frac{\gamma_i(\bar{h}, \vec{c}_j) \cdot \eta_i(\bar{h}, \vec{c}_j)}{\gamma_i(\bar{h}, c_{j,i})}\right]
\end{aligned}$$

□

Using the above formulas, our goal is to show that

$$\prod_{i=1}^{n} \eta_i(\bar{h}) = \prod_{i=1}^{n} \mathop{\mathrm{E}}_{\vec{c}_j}\left[\frac{\gamma_i(\bar{h},\vec{c}_j)\cdot\eta_i(\bar{h},\vec{c}_j)}{\gamma_i(\bar{h},c_{j,i})}\right] \geq \frac{\gamma^{n+1}(\bar{h})}{\prod_{i=1}^{n}\gamma_i(\bar{h})}.$$

As in Section 3.2.2, we formalize the induction step as the following lemma, and prove it by applying Hölder's Inequality twice.

Lemma 3.19 (Induction Step) *Let $\gamma, \gamma_1, \ldots, \gamma_n, \eta_1, \ldots, \eta_n : \Omega^n \to [0,1]$ be $[0,1]$-valued functions over a product space Ω^n such that*

$$\prod_i \eta_i(\vec{q}) \geq \frac{\gamma^{n+1}(\vec{q})}{\gamma_i(\vec{q})} \quad \text{for every } \vec{q} = (q_1,\ldots,q_n) \in \Omega^n.$$

Let $\gamma = \mathrm{E}_{\vec{q}}[\gamma(\vec{q})]$. For every $i \in [n]$ and $q_i \in \Omega$, let

$$\gamma_i = \mathop{\mathrm{E}}_{\vec{q}}[\gamma_i(\vec{q})], \quad \gamma_i(q_i) = \mathop{\mathrm{E}}_{\vec{q}_{-i}}[\gamma_i(\vec{q})] \text{ and } \eta_i = \mathop{\mathrm{E}}_{\vec{q}}\left[\frac{\gamma_i(\vec{q})\cdot\eta_i(\vec{q})}{\gamma_i(q_i)}\right],$$

where the above expectation is over uniform distribution over Ω^n. We have

$$\prod_{i=1}^{n}\eta_i = \prod_{i=1}^{n}\mathop{\mathrm{E}}_{\vec{q}}\left[\left(\frac{\gamma_i(\vec{q})\cdot\eta_i(\vec{q})}{\gamma_i(q_i)}\right)\right] \geq \frac{\gamma^{n+1}}{\prod_{i=1}^{n}\gamma_i}.$$

Proof. We apply Hölder's Inequality twice to prove the lemma, as presented in the calculation below. The calculation is similar to that in the proof of Lemma 3.6, and the applications of Hölder's Inequality are the same.

$$\begin{aligned}
&\prod_{i=1}^{n}\mathop{\mathrm{E}}_{\vec{q}}\left[\left(\frac{\gamma_i(\vec{q})\cdot\eta_i(\vec{q})}{\gamma_i(q_i)}\right)\right] \\
&\quad\geq \mathop{\mathrm{E}}_{\vec{q}}\left[\left(\frac{\prod_{i=1}^{n}\gamma_i(\vec{q})^n\cdot\prod_{i=1}^{n}\eta_i(\vec{q})}{\prod_{i=1}^{n}\gamma_i(q_i)}\right)^{1/n}\right]^n && \text{(by Hölder's Inequality)} \\
&\quad\geq \mathop{\mathrm{E}}_{\vec{q}}\left[\left(\frac{\prod_{i=1}^{n}\gamma_i(\vec{q})\cdot(\gamma(\vec{q})^{n+1}/\prod_{i=1}^{n}\gamma_i(\vec{q})}{\prod_{i=1}^{n}\gamma_i(q_i)}\right)^{1/n}\right]^n && \text{(by induction hypothesis)} \\
&\quad\geq \mathop{\mathrm{E}}_{\vec{q}}\left[\left(\frac{\gamma(\vec{q})^{n+1}}{\prod_{i=1}^{n}\gamma_i(q_i)}\right)^{1/n}\right]^n \\
&\quad= \left[\left(\frac{\mathrm{E}_{\vec{q}}[\gamma(\vec{q})]^{n+1}}{\mathrm{E}_{\vec{q}}[\prod_{i=1}^{n}\gamma_i(q_i)]}\right)^{1/n}\right]^n && \text{(by Hölder's Inequality)} \\
&\quad= \frac{\gamma^{n+1}}{\prod_{i=1}^{n}\gamma_i}.
\end{aligned}$$

■

It is easy to verify that the probabilities $\gamma(\bar{h},\cdot)$, $\gamma_i(\bar{h},\cdot)$ and $\eta_i(\bar{h},\cdot)$ satisfy the premise of the above lemma. A straightforward application of the above lemma completes the analysis of the induction step and hence completes the analysis of $\mathsf{P}^{*(O_\mathsf{V})}_{ideal}$.

Analysis of the Perfectified Rejection Sampling Strategy $\mathsf{P}^{*(O_\mathsf{V})}_{rej}$

Following the proof in Section 3.2.2, we proceed to analyze the actual (non-ideal version of) perfectified rejection sampling strategy $\mathsf{P}^{*(O_\mathsf{V})}_{rej}$, where we modify the induction hypothesis to accommodate the sampling errors, i.e., $\mathsf{P}^{*(O_\mathsf{V})}_{rej}$ may abort due to the failure of finding a successful continuation in M trials when there are too few successful continuations.

Let a m-round PPT verifier V, input x, parameters $n, M \in \mathbb{N}$, $\delta, \xi \in (0,1)$, a deterministic parallel prover P^{n*} be given as in Figure 3.10. We prove the following technical lemma regarding the success probability of $\mathsf{P}^{*(O_\mathsf{V})}_{rej}$.

Lemma 3.20 *Let* $\gamma \stackrel{\text{def}}{=} \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1]$ *and* $\gamma_i \stackrel{\text{def}}{=} \Pr[\mathsf{P}^{n*}$ *convinces* $\mathsf{V}_{-i}]$ *for every* $i \in [n]$*. We have*

$$\Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}_{rej}, \mathsf{V}\rangle(x) = 1] \geq \left(\frac{(\gamma - m\cdot\nu)^{n+1}_+}{\prod_{i=1}^n(\gamma_i+\nu)}\right)^{1/n},$$

where $(\alpha)_+ \stackrel{\text{def}}{=} \max\{\alpha, 0\}$ *and* $\nu = 1/M$*.*

Proof. The proof follows closely the analysis of $\mathsf{P}^{*(O_\mathsf{V})}_{ideal}$. Recall that we use $\bar{h} = (\vec{c}_1, \ldots, \vec{c}_j)$ to denote a partial interaction, and use $\gamma(\bar{h})$ and $\gamma_i(\bar{h})$ to denote the success probability of P^{n*} in convincing $\mathsf{V}^{n,n}$ and V_{-i}, respectively, starting from partial interaction $\bar{h}$. We redefine $\eta_i(\bar{h})$ to denote the success probability of $\mathsf{P}^{*(O_\mathsf{V})}_{rej}$:

$$\eta_i(\bar{h}) = \eta_i(\vec{c}_1, \ldots, \vec{c}_j) \stackrel{\text{def}}{=} \Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}_{rej}, \mathsf{V}\rangle(x) = 1 | i, \vec{c}_1, \ldots, \vec{c}_j] \text{ for every } i \in [n].$$

We use the following induction hypothesis.

Induction Hypothesis for $\mathsf{P}^{*(O_\mathsf{V})}_{rej}$. For every $j \in [m]$ and every partial interaction $(\vec{c}_1, \ldots, \vec{c}_j)$, the following inequality holds.

$$\prod_{i=1}^n \eta_i(\bar{h}) \geq \left(\frac{(\gamma(\bar{h}) - (m-j)\cdot\nu)^{n+1}_+}{\prod_{i=1}^n(\gamma_i(\bar{h})+\nu)}\right).$$

Observe that as $M \to \infty$ (i.e., $\nu \to 0$), our induction hypothesis is the same as before. Compared to the induction hypothesis for $\mathsf{P}^{*(O_\mathsf{V})}_{ideal}$, we add certain slackness in $\nu = 1/M$ in both the numerator and the denominator, where the slackness in the numerator grows round by round backwardly to accommodate the sampling errors. We show how to prove the lemma using the induction hypothesis first.

Induction Hypothesis for $j = 0 \Rightarrow$ Lemma 3.20. When $j = 0$, the induction hypothesis says that

$$\prod_{i=1}^{n} \Pr[\langle \mathsf{P}_{rej}^{*(O_\mathsf{V})}, \mathsf{V} \rangle(x) = 1 | i] \geq \frac{(\gamma - m \cdot \nu)_+^{n+1}}{\prod_{i=1}^{n}(\gamma_i + \nu)}.$$

Applying the Arithmetic-Mean-Geometric-Mean Inequality in the same way as before gives the desired lower bound on the success probability of $\mathsf{P}_{ideal}^{*(O_\mathsf{V})}$.

$$\begin{aligned}
&\Pr[\langle \mathsf{P}_{rej}^{*(O_\mathsf{V})}, \mathsf{V} \rangle(x) = 1] \\
&= \frac{1}{n} \sum_{i=1}^{n} \Pr[\langle \mathsf{P}_{rej}^{*(O_\mathsf{V})}, \mathsf{V} \rangle(x) = 1 | i] \\
&\geq \left(\prod_{i=1}^{n} \Pr[\langle \mathsf{P}_{rej}^{*(O_\mathsf{V})}, \mathsf{V} \rangle(x) = 1 | i] \right)^{1/n} \\
&\geq \left(\frac{(\gamma - m \cdot \nu)_+^{n+1}}{\prod_{i=1}^{n}(\gamma_i + \nu)} \right)^{1/n}
\end{aligned}$$

■

We proceed to prove the induction. Again, the base case $j = m$ is trivial to check, as the probabilities $\gamma(\bar{h}), \gamma_i(\bar{h})$ and $\eta_i(\bar{h})$ are simply 0 or 1 when conditioned on a complete interaction $\bar{h} = (\vec{c}_1, \ldots, \vec{c}_m)$. We prove the induction step below.

Induction Step. For every $j \in [m]$ and every partial interaction $\bar{h} = (\vec{c}_1, \ldots, \vec{c}_{j-1})$, the following holds. Suppose for all coins $\vec{c}_j$, it is true that

$$\prod_{i=1}^{n} \eta_i(\bar{h}, \vec{c}_j) \geq \left(\frac{(\gamma(\bar{h}, \vec{c}_j) - (m - j) \cdot \nu)_+^{n+1}}{\prod_{i=1}^{n}(\gamma_i(\bar{h}, \vec{c}_j) + \nu)} \right),$$

Then we have

$$\prod_{i=1}^{n} \eta_i(\bar{h}) \geq \left(\frac{(\gamma(\bar{h}) - (m - (j-1)) \cdot \nu)_+^{n+1}}{\prod_{i=1}^{n}(\gamma(\bar{h}) + \nu)} \right).$$

The induction step can be proved by applying Hölder's Inequality twice in the same way as before, but with a more careful analysis on the error terms. We first express the success probabilities $\eta_i(\bar{h})$ in terms of $\gamma_i(\bar{h}, \vec{c}_j)$ and $\eta_i(\bar{h}, \vec{c}_j)$.

Claim 3.21 *For every $i \in [n]$, $j \in [m]$, and partial interaction $\bar{h} = (\vec{c}_1, \ldots, \vec{c}_{j-1})$, we have*

$$\eta_i(\bar{h}) = \mathop{\mathrm{E}}_{\vec{c}_j} \left[\frac{\gamma_i(\bar{h}, \vec{c}_j) \cdot \eta_i(\bar{h}, \vec{c}_j)}{\gamma_i(\bar{h}, c_{j,i})} \cdot f(\gamma_i(\bar{h}, c_{j,i})) \right],$$

where $f(\alpha) = (1 - (1 - \alpha)^M)$, and M is the number of samples specified in the strategy of $\mathsf{P}_{rej}^{*(O_\mathsf{V})}$ in Figure 3.10.

Proof of claim: Observing that $\mathsf{P}_{rej}^{*(O_{\mathsf{V}})}$ can find a successful continuation with probability exactly $f(\gamma_i(\bar{h}, c_{j,i}))$, and that conditioning on a successful continuation is found, $\mathsf{P}_{rej}^{*(O_{\mathsf{V}})}$ plays $\vec{c}_{j,-i}$ with the same probability as $\mathsf{P}_{ideal}^{*(O_{\mathsf{V}})}$, we obtain the above formula for η_i. □

We prove the induction step by the following lemma.

Lemma 3.22 *Let $\nu \in (0,1)$ and $t, M \geq 0$ such that $M \cdot \nu \geq 1$. Let $\gamma, \gamma_1, \ldots, \gamma_n, \eta_1, \ldots, \eta_n : \Omega^n \to [0,1]$ be functions over Ω^n such that*

$$\prod_i \eta_i(\vec{q}) \geq \left(\frac{(\gamma(\vec{q}) - t \cdot \nu)_+^{n+1}}{\prod_{i=1}^n (\gamma_i(\vec{q}) + \nu)} \right)$$

for every $\vec{q} = (q_1, \ldots, q_n) \in \Omega^n$. Let $\gamma = \mathrm{E}_{\vec{q}}[\gamma(\vec{q})]$. For every $i \in [n]$, let $\gamma_i = \mathrm{E}_{\vec{q}}[\gamma_i(\vec{q})]$, $\gamma_i(q_i) = \mathrm{E}_{\vec{q}_{-i}}[\gamma_i(\vec{q})]$, and

$$\eta_i = \mathop{\mathrm{E}}_{\vec{q}} \left[\frac{\gamma_i(\vec{q}) \cdot \eta_i(\vec{q})}{\gamma_i(q_i)} \cdot f(\gamma_i(q_i)) \right],$$

where $f(\alpha) = (1 - (1-\alpha)^M)$, and the above expectation is over uniform distribution over Ω^n. We have

$$\prod_{i=1}^n \eta_i = \prod_{i=1}^n \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{\gamma_i(\vec{q}) \cdot \eta_i(\vec{q})}{\gamma_i(q_i)} \right) \cdot f(\gamma_i(q_i)) \right] \geq \left(\frac{(\gamma - (t+1) \cdot \nu)_+^{n+1}}{\prod_{i=1}^n (\gamma_i + \nu)} \right).$$

Proof. The proof is very similar to that of Lemma 3.9 in Section 3.2.2. In the calculation below, Hölder's Inequalities are applied in the same way as before, and the third and last inequality can be justified in the same way as the corresponding

inequalities in the proof of Lemma 3.9.

$$
\begin{aligned}
&\prod_{i=1}^{n} \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{\gamma_i(\vec{q}) \cdot \eta_i(\vec{q})}{\gamma_i(q_i)} \cdot f(\gamma_i(q_i)) \right) \right] \\
&= \prod_{i=1}^{n} \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{\gamma_i(\vec{q}) \cdot \eta_i(\vec{q})}{\gamma_i(q_i)/f(\gamma_i(q_i))} \right) \right] \\
&\geq \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{\prod_{i=1}^{n} \gamma_i(\vec{q}) \cdot \prod_{i=1}^{n} \eta_i(\vec{q})}{\prod_{i=1}^{n} (\gamma_i(q_i)/f(\gamma_i(q_i)))} \right)^{1/n} \right]^n \qquad \text{(by Hölder's Inequality)} \\
&\geq \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{(\prod_{i=1}^{n} \gamma_i(\vec{q})) \cdot ((\gamma(\vec{q}) - t \cdot \nu)_+^{n+1} / \prod_{i=1}^{n} (\gamma_i(\vec{q}) + \nu))}{\prod_{i=1}^{n} (\gamma_i(q_i)/f(\gamma_i(q_i)))} \right)^{1/n} \right]^n \text{(by induction hypothesis)} \\
&\geq \mathop{\mathrm{E}}_{\vec{q}} \left[\left(\frac{(\gamma(\vec{q}) - (t+1) \cdot \nu)_+^{n+1}}{\prod_{i=1}^{n} (\gamma_i(q_i)/f(\gamma_i(q_i)))} \right)^{1/n} \right]^n \\
&\geq \left[\left(\frac{\mathrm{E}_{\vec{q}}[(\gamma(\vec{q}) - (t+1) \cdot \nu)_+]^{n+1}}{\mathrm{E}_{\vec{q}}[\prod_{i=1}^{n} (\gamma_i(q_i)/f(\gamma_i(q_i)))]} \right)^{1/n} \right]^n \qquad \text{(by Hölder's Inequality)} \\
&\geq \left(\frac{(\gamma - (t+1) \cdot \nu)_+^{n+1}}{\prod_{i=1}^{n} (\gamma_i + \nu)} \right)
\end{aligned}
$$

■

Applying the above lemma directly completes the proof of induction and Lemma 3.20. ■

3.4.6 Relating the Success Probability of P^*_{rej} and $\mathsf{P}^{*(O_\mathsf{V})}_{rej}$

In this section, we argue that the success probability of the actual rejection sampling strategy P^*_{rej} is close to that of the perfectified rejection sampling strategy $\mathsf{P}^{*(O_\mathsf{V})}_{rej}$, provided that they run in polynomial time. More generally, we prove by a hybrid argument that for any PPT prover strategy that uses a simulator S to generate samples of random continuations, the success probability of P^* is close the that of a corresponding perfectified prover strategy $\mathsf{P}^{*(O_\mathsf{V})}$, who uses oracle O_V to generate such samples with correct distribution.

Lemma 3.23 *Let* $\mathsf{V} \in$ PPT *be a* PPT *verifier that is computationally simulatable with respect to a* PPT *prover strategy* P^*_0*. Let* S *be the corresponding simulator. Let* P^* *be a* PPT *prover strategy, who uses* S *to generate (computationally indistinguishable) samples of random continuations of* $\langle \mathsf{P}^*_0, \mathsf{V} \rangle$*. Let* $\mathsf{P}^{*(O_\mathsf{V})}$ *be a corresponding perfectified prover strategy of* P^**, who instead of using* S*, uses oracle* O_V *to generate such samples with correct distribution. It holds that for sufficiently large* s*, and every input* $x \in \{0,1\}^*$ *with security parameter* s*,*

$$|\Pr[\langle \mathsf{P}^*, \mathsf{V} \rangle(x) = 1] - \Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}, \mathsf{V} \rangle(x) = 1]| \leq \mathsf{ngl}(s),$$

where ngl *is a negligible function in the security parameter* s*.*

Proof. We prove it by contradiction. Suppose that for infinitely many $s \in \mathbb{N}$, there exists an input $x \in \{0,1\}^*$ with security parameter s such that

$$|\Pr[\langle \mathsf{P}^*, \mathsf{V}\rangle(x) = 1] - \Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}, \mathsf{V}\rangle(x) = 1]| \geq \varepsilon(s), \tag{3.2}$$

where $\varepsilon(s)$ is a non-negligible function. We will show that S is not a simulator for P_0^*. Namely, there exists a non-uniform PPT distinguisher D such that for infinitely many s, there exists a partial interaction $(t_{[\ell]}, c_{[\ell]}, x, p_{[\ell]}, v_{[\ell]})$ of $\langle \mathsf{P}_0^*, \mathsf{V}\rangle$ such that D distinguishes the actual random continuation $(t_{[m]}, x, p_{[m]}, v_{[m]})$ from the one $(t'_{[m]}, x, p'_{[m]}, v'_{[m]})$ generated by S with non-negligible probability.

Given an input x such that Inequality (3.2) holds, and let $T \in \mathrm{poly}(s)$ be an upper bound on the number of calls to S that P^* makes. We define the following hybrid experiments.

- Experiment H_k: the interaction between P^* and V on input x, where P^* uses oracle O_V to generate the first k samples of random continuations, and uses the simulator S to generate the remaining samples of random continuations.

It is easy to see that the interactions $\langle \mathsf{P}^*, \mathsf{V}\rangle(x)$ and $\langle \mathsf{P}^{*(O_\mathsf{V})}, \mathsf{V}\rangle(x)$ correspond to hybrids H_0 and H_T, respectively. Also note that the whole experiment H_k can be generated efficiently since all parties $\mathsf{V}, \mathsf{P}^*, \mathsf{P}_0^*$, and S are PPT, and O_V is just V. By a standard hybrid argument, there exists an index $k \in [T]$ such that

$$|\Pr[H_{k-1} \text{ outputs } 1] - \Pr[H_k \text{ outputs } 1]| \geq \varepsilon/T.$$

Note that the only difference between H_{k-1} and H_k is that the k-th sample of random continuation is generated by S in H_{k-1} and generated from the correct distribution in H_k. By an averaging argument, we can fix the randomness of the remaining part of the experiments while maintaining the non-negligible probability gap. More precisely, recall that $c_{[m]}$ denotes the coins of V. Let $R_1, \ldots, R_T$ denote the random coins for generating the T random continuations (either from S or from the correct distribution), and let R be the coins used by P^*. By an averaging argument, there exists a set of coins $R, \vec{R}_{-k} = (R_1, \ldots, R_{k-1}, R_{k+1}, \ldots, R_T), c_{[m]}$, such that

$$|\Pr[H_{k-1} \text{ outputs } 1 | R, \vec{R}_{-k}, c_{[m]}] - \Pr[H_k \text{ outputs } 1 | R, \vec{R}_{-k}, c_{[m]}]| \geq \varepsilon/T.$$

Note that when the coins $R, \vec{R}_{-k}, c_{[m]}$ are fixed, the partial interaction $(t_{[\ell]}, c_{[\ell]}, x, p_{[\ell]}, v_{[\ell]})$ of the k-th random continuation is also determined. The above non-negligible probability gap of the two experiments gives the desired distinguisher.

Specifically, our distinguisher D has coins $R, \vec{R}_{-k}, c_{[m]}$ hard-wired in. On input $(\tilde{t}_{[m]}, x, \tilde{p}_{[m]}, \tilde{v}_{[m]})$, D simulates the interaction of P^* and V with the hard-wired coins R and $c_{[m]}$, respectively, and outputs the verdict of V, where (1) the first $k-1$ random

continuations are generated by actual distribution using coins $R_1, \ldots, R_{i-1}$, (2) the k-th random continuation is $(\tilde{t}_{[m]}, x, \tilde{p}_{[m]}, \tilde{v}_{[m]})$, and (3) the remaining continuations are generated using S with coins $R_{k+1}, \ldots, R_T$. Note that D can be implemented in non-uniform PPT.

By construction, if the input is a random continuation $(t'_{[m]}, x, p'_{[m]}, v'_{[m]})$ generated by S from the partial interaction $(t_{[\ell]}, c_{[\ell]}, x, p_{[\ell]}, v_{[\ell]})$, then D simulates H_{k-1} with coins $R, \vec{R}_{-k}, c_{[m]}$, and if the input is a random continuation $(t_{[m]}, x, p_{[m]}, v_{[m]})$ with correct distribution, then D simulates H_k with coins $R, \vec{R}_{-k}, c_{[m]}$. Hence, for this partial interaction, we have

$$|\Pr[D((t_{[m]}, x, p_{[m]}, v_{[m]}) = 1] - \Pr[D((t'_{[m]}, x, p'_{[m]}, v'_{[m]}) = 1]| \geq \varepsilon/T.$$

Note that this holds for infinitely many $s \in \mathbb{N}$, which implies that S is not a simulator for P_0^*, a contradiction. ■

3.4.7 Proof of Theorem 3.16

Putting things together, we prove Theorem 3.16 restated as follows. (Again, we use parameters δ^2, δ^n instead of $\delta, \delta^{n/2}$ for convenience.)

Theorem 3.24 *(Theorem 3.16 restated) Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a computationally simulatable protocol with input domain Λ, $\delta : \Lambda \to [0, 1]$ and $n : \mathbb{N} \to \mathbb{N}$ efficiently computable functions with $n \leq \text{poly}(s)$. If $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ^2, then its n-fold parallel repetition with direct product verifier $\langle \mathsf{P}^n, \mathsf{V}^{n,n} \rangle$ has soundness error $\delta^n + \mathsf{ngl}$, where ngl denotes a negligible function in the security parameter s.*

Proof. We prove it by contradiction. Suppose the conclusion is not true, then there exists a PPT parallel prover P^{n*} and a noticeable η such that for infinitely many $s \in \mathbb{N}$, there exists some x with security parameter s such that

$$\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1] > \delta^n(x) + \eta(s). \tag{3.3}$$

Consider the reduction prover strategy P^* defined in Figure 3.9 with parameters n, δ and $\xi = \eta$ and oracle access to P^{n*}. We claim that

1. P^* runs in time $\text{poly}(|x|, n, \xi) = \text{poly}(s)$, which is efficient.

2. For sufficiently large $s \in \mathbb{N}$, for every x with security parameter s such that the above Inequality 3.3 holds, we have

$$\Pr[\langle \mathsf{P}^{*(\mathsf{P}^{n*})}(n, \delta, \xi), \mathsf{V} \rangle(x) = 1] > \delta.$$

This contradicts to the fact that $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ and completes the proof.

It remains to prove the two claims. We first argue that P^* runs in time $\mathrm{poly}(|x|, n, \xi^{-1})$. It is not hard to see by inspection that both CR and P^*_{rej} run in time $\mathrm{poly}(|x|, n, \delta^{-n}, \xi^{-1})$. Note that in the first step of P^*, we update the value of δ, ξ so that the updated δ^n, ξ are at least a half of the original ξ. Hence, P^* runs in time $\mathrm{poly}(|x|, n, \xi^{-1}) = \mathrm{poly}(s)$.

We proceed to prove the second claim. In the following analysis, we refer to the updated value of δ, ξ after the first step of P^*. We first lower bound the success probability of the corresponding perfectified $\mathsf{P}^{*(O_\mathsf{V})}$, who calls the perfectified $\mathsf{P}^{*(O_\mathsf{V})}_{rej}$ instead of P^*_{rej}. By Lemma 3.17, with probability at least $1 - (\xi/10n)$, CR outputs a good $\mathsf{P}^{n'*}$ with $1 \leq n' \leq n$ such that

- $\Pr[\langle \mathsf{P}^{n'*}, \mathsf{V}^{n',n'} \rangle(x) = 1] \geq \delta^{n'} + \frac{10n'-1}{10n} \cdot \xi$.
- Either $n' = 1$ or for every $i \in [n']$, $\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}_{-i}] \geq \delta^{n'-1} + \frac{10(n'-1)+2}{10n} \cdot \xi$.

By Lemma 3.20, when apply to such a good $\mathsf{P}^{n'*}$, $\mathsf{P}^{*(O_\mathsf{V})}_{rej}$ can succeed with good probability. In particular, for $1 < n' \leq n$, we have

$$\left(\frac{\left(\delta^{n'} + \frac{10n'-1}{10n} \cdot \xi\right)^{n'+1}}{\left(\delta^{n'-1} + \frac{10(n'-1)+2}{10n} \cdot \xi\right)^{n'}} \right)^{1/n'} \geq \delta^2 + \frac{2\xi}{10n}.$$

Also, when $n' = 1$, we have $\delta + (9\xi/10n) \geq \delta^2 + (2\xi/10n)$. It follows that

$$\begin{aligned}
&\Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}, \mathsf{V} \rangle(x) = 1] \\
&\geq \Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}, \mathsf{V} \rangle(x) = 1 | \mathsf{P}^{n'*} \text{ is good }] - \Pr[\mathsf{P}^{n'*} \text{ is not good }] \\
&\geq \delta^2 + \frac{2\xi}{10n} - \frac{\xi}{10n} \\
&\geq \delta^2 + \frac{\xi}{10n}.
\end{aligned}$$

Finally, by Lemma 3.23, when s is sufficiently large, we have

$$\Pr[\langle \mathsf{P}^*, \mathsf{V} \rangle(x) = 1] \geq \Pr[\langle \mathsf{P}^{*(O_\mathsf{V})}, \mathsf{V} \rangle(x) = 1] - \mathsf{ngl}(s) \geq \delta^2 + \frac{\xi}{10n} - \mathsf{ngl}(s) > \delta^2.$$

Noting the δ can only increase in the first step of P^*, this finishes the proof of our second claim. ■

3.4.8 Discussion

In this section, we briefly discuss the reduction prover strategy for simulatable protocols (both computationally simulatable and "weakly simulatable") and the tightness of our analysis.

Recall that in terms of the reduction, a main difference of simulatable protocols from public-coin protocols is that the external verifier's verdict bit is unknown when we sample a random continuation, and this situation is similar to the difference between three-message protocols and three-message public-coin protocols. Also recall that there are two methods to handle the issue of unknown verdict bit. In fact, both methods can be generalized to the setting of simulatable protocols. We give a high level comparison of the two methods as follows.

- Haståd, Pass, Pietrzak, and Wikström [19, 20] generalize the "soft-decision method" of Bellare, Impagliazzo, and Naor [1] to prove parallel repetition theorems for simulatable protocols. This method can be applied to both computationally simulatable and weakly simulatable protocols, and gives Chernoff-type theorems directly. However, this method gives worse bounds in the direct product case.

- In this section, we used the correlation reduction method of Canetti, Halevi, and Steiner [2] to prove a direct product theorem for computationally simulatable protocols. Our bound is better than the bound of Haståd et al. [20], and almost matches the information-theoretic bounds. However, unlike the case of three-message protocols, where we can obtain tight parallel repetition theorems for general monotone verifiers, here, it is unclear how to prove Chernoff-type theorems for simulatable protocols using the correlation reduction method.

We proceed to discuss the tightness of our analysis for our reduction prover strategy P^* in Figure 3.9. We will show that as a black-box reduction, the bound $\delta^{n/2}$ is actually tight for P^*. Therefore, to improve the bound to δ^n (if it is possible), we need a different reduction prover strategy. We will construct an (artificial) example to prove the tightness of our analysis. The tight example consists of four messages. In contrast, as we remarked before, when we restrict to three-message protocols, one can improve the bound for P^* to the optimal δ^n by slightly modifying the analysis in Section 3.3. For simplicity, in the following example, we omit the common input x, and ignore the sampling errors and slackness parameters.

A Tight Example (informal). Let us consider a four-message public-coin protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ and its two-fold repetition $\langle \mathsf{P}^2, \mathsf{V}^{2,2} \rangle$. Let the first and second message of $\mathsf{V}^{2,2}$ be denoted by $\vec{v}_1 = (v_{1,1}.v_{1,2}) \in \{0,1\}^{2\times k}$ and $\vec{v}_2 = (v_{2,1}, v_{2,2}) \in \{0,1\}^{2\times k}$, respectively. Our goal is to construct a (deterministic) parallel prover strategy P^{2*} for $\mathsf{V}^{2,2}$ such that P^{2*} can succeed with probability at least δ^2, but the reduction prover strategy P^* in Figure 3.9, given oracle access to P^{2*}, can only success with probability at most δ^2.

Note that for our purpose, the details of the protocol does not matter. The important data to design is the "success pattern" of P^{2*}, namely, a mapping from $\mathsf{V}^{2,2}$'s

messages $\vec{v}_1, \vec{v}_2$ to $\{0,1\}^2$ that specifies whether P^{2*} convinces the two subverifiers of $\mathsf{V}^{2,2}$ when $\mathsf{V}^{2,2}$'s messages are $\vec{v}_1, \vec{v}_2$.

We will construct a P^{2*} such that correlation reduction CR does not have any effect, i.e., always returns P^{2*}. For this to hold, we need to make sure that

- For all V_1 messages $v_{[2],1} = (v_{1,1}, v_{2,1})$, $\Pr[\mathsf{P}^{2*} \text{ convinces } \mathsf{V}_2 | v_{[2],1}] \leq \delta$.
- For all V_2 messages $v_{[2],2} = (v_{1,2}, v_{2,2})$, $\Pr[\mathsf{P}^{2*} \text{ convinces } \mathsf{V}_1 | v_{[2],2}] \leq \delta$.

Of course, the main properties we want are that $\Pr[\mathsf{P}^{2*} \text{ convinces } \mathsf{V}^{2,2}] \geq \delta^2$, and that the rejection sampling strategy P^*_{rej} in Figure 3.9 can only succeed with probability at most δ^2, given oracle access to P^{2*}.

The construction is not too complicated, so we present it directly as follows. We view each message $v_{i,j} \in \{0,1\}^k$ as an element in $\mathbb{Z}_K$, where $K = 2^k$. Let $S \subset \mathbb{Z}_K$ be an arbitrary subset of $\mathbb{Z}_K$ of size $|S| = \delta \cdot K$. We construct P^{2*} such that

- P^{2*} convinces V_1 iff $v_{1,1} + v_{1,2} + v_{2,2} \in S$, where the operation $+$ is over $\mathbb{Z}_K$.
- P^{2*} convinces V_2 iff $v_{1,1} + v_{1,2} + v_{2,1} \in S$, where the operation $+$ is over $\mathbb{Z}_K$.

We comment that this example is very artificial in the sense that whether P^{2*} convinces V_1 depends on V_2's message $v_{2,2}$ but not on V_1's message $v_{2,1}$.

By construction, it is not hard to see that the success probability of P^{2*} is δ^2.

$$\Pr[\mathsf{P}^{2*} \text{ convinces } \mathsf{V}^{2,2}] = \Pr[(v_{1,1} + v_{1,2} + v_{2,2} \in S) \wedge (v_{1,1} + v_{1,2} + v_{2,1} \in S)] = \delta^2$$

Let us proceed to analyze P^*_{rej}'s success probability given oracle access to P^{2*}. We claim that conditioning on every coordinate $i \in [2]$ and first message $\vec{v}_1 = (v_{1,1}, v_{1,2})$, P^*_{rej} can only succeed with probability δ^2, and hence, the success probability of P^*_{rej} is δ^2.

To see this, consider say, $i = 1$, and an arbitrary first message $\vec{v}_1 = (v_{1,1}, v_{1,2})$. In the second round, V selects a uniformly random $v_{2,1}$, and P^*_{rej} uses rejection sampling to select a random successful continuation $v_{2,2}$ such that P^{2*} convinces V_2. Note that $v_{1,1}, v_{1,2}$, and $v_{2,1}$ determines whether P^{2*} convinces V_2, so either

1. every $v_{2,2}$ is a successful continuation, and P^*_{rej} selects a uniformly random $v_{2,2}$, or
2. there exists no successful continuation, and P^*_{rej} aborts and fails.

Hence, P^*_{rej} succeeds only when both $v_{1,1} + v_{1,2} + v_{2,1} \in S$ and $v_{1,1} + v_{1,2} + v_{2,2} \in S$, where both $v_{2,1}$ and $v_{2,2}$ are uniformly random. It follows that P^*_{rej} can succeed with probability δ^2, as claimed. A similar argument applies to the case $i = 2$ as well.

It may look unsatisfactory that the reason that P^*_{rej} fails in the second case is the non-existence of successful continuation. However, it is not hard to modify the

example slightly so that in the second case, there exist a few successful continuations and P^*_{rej} fails when using any of these successful continuations. We omit the details since it is not very enlightening.

Finally, we check that the correlation reduction CR has no effect. For every V_1's messge $v_{[2],1} = (v_{1,1}, v_{2,1})$, we have

$$\Pr[\mathsf{P}^{2*} \text{ convinces } \mathsf{V}_2 | v_{1,1}, v_{2,1}] = \Pr[v_{1,1} + v_{1,2} + v_{2,1} \in S | v_{1,1}, v_{2,1}] = \delta.$$

Similarly, for every V_2's messge $v_{[2],2} = (v_{1,2}, v_{2,2})$, we have

$$\Pr[\mathsf{P}^{2*} \text{ convinces } \mathsf{V}_1 | v_{1,2}, v_{2,2}] = \Pr[v_{1,1} + v_{1,2} + v_{2,2} \in S | v_{1,2}, v_{2,2}] = \delta.$$

Therefore, given P^{2*}, CR always returns P^{2*}.

Putting things together, we showed that given oracle access to the above P^{2*}, who has success probability δ^2, the reduction prover strategy P^* in Figure 3.9 can only success with probability δ^2, which implies that our analysis to P^* is tight. We remark that in the above presentation, we omit some implementation details such as sampling errors, which can be handled by small changes to the above example. We skip the details to make the idea clear.

3.5 Making Any Protocol Computationally Simulatable

In this section, we present a simple transformation that converts any interactive protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ to one $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ that is computationally simulatable. Roughly speaking, the idea, inspired by Gennaro, Gentry, and Parno [10] is for the transformed protocol $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ to run the original protocol under encryption using a fully homomorphic encryption scheme. This gives a way to get around the negative results of Bellare, Impagliazzo, and Naor [1] and Pietrzak and Wikström [31]. Namely, we can amplify the soundness of *any* computationally sound protocols by first applying the transformation and then doing parallel repetition.

In such a transformation, it is desirable to preserve the structure of the original protocol, such as the round complexity and completeness and soundness property. Our transformation preserves these properties, and blows up the communication complexity by a factor that depends only on the security parameter of the fully homomorphic encryption scheme.

The idea of getting around the negative results of Bellare et al. [1] and Pietrzak and Wikström [31] by slightly modifying the protocol was first proposed by Haitner [16]. He suggested a "random-termination" transformation, where in every round, the new verifier $\tilde{\mathsf{V}}$ terminates and accepts with a small probability. The transformation also preserves most of the structure of the original protocol, and it makes any protocol

weakly simulatable, so that parallel repetition decreases the soundness error of the transformed protocol.

In comparison, our transformation requires a strong assumption of the existence of fully homomorphic encryption schemes, but gives better bounds on the soundness error under parallel repetition. In contrast, the transformation of Haitner [16] is unconditional, but the bound of soundness error is worse, with an undesirable dependency on the round complexity.

We start by introducing fully homomorphic encryption schemes in the following section.

3.5.1 Fully Homomorphic Encryption Schemes

A *fully homomorphic encryption scheme* is a public-key encryption scheme $\mathrm{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ associated with an additional polynomial-time algorithm Eval, that takes as input a public key pk, a ciphertext $\hat{m} = \mathsf{Enc}(m)$ and a circuit C, and outputs, a new ciphertext $c = \mathsf{Eval}_{\mathsf{pk}}(C, \hat{m})$, such that $\mathsf{Dec}_{\mathsf{sk}}(c) = C(m)$, where sk is the secret key corresponding to the public key pk. It is required that the size of $c = \mathsf{Eval}_{\mathsf{pk}}(\mathsf{Enc}_{\mathsf{pk}}(m), C)$ depends polynomially on the security parameter and the length of $C(m)$, but is otherwise independent of the size of the circuit C.

In a recent breakthrough, Gentry [11] proposed a fully homomorphic encryption scheme based on ideal lattices. Following Gentry, an alternative construction was proposed by van Dijk, Gentry, Halevi, and Vaikuntanathan [35] based on the hardness of an "Approximate-GCD" problem. In both schemes, the complexity of the algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ depend linearly on the *depth* of the circuits C that are allowed as inputs to Eval. However, under the additional assumption that the encryption schemes are circular secure (i.e., it remains secure even given an encryption of the secret key), the complexity of these algorithms are independent of C.

For the sake of completeness, we present a formal definition of fully homomorphic encryption schemes as follows. We separate the security parameter k for fully homomorphic encryption schemes from the security parameter s for computationally sound protocols .

Definition 3.25 (Fully Homomorphic Encryption Schemes) *A fully homomorphic encryption scheme is a four-tuple of* PPT *algorithms* $\mathrm{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *defined as follows.*

- KeyGen*: on input a security parameter* 1^k*,* $\mathsf{KeyGen}(1^k)$ *outputs a pair of public and secret keys* $(\mathsf{pk}, \mathsf{sk})$*.*
- Enc*: on input a public key* pk *and a message* $m \in \{0,1\}^*$*,* $\mathsf{Enc}_{pk}(m)$ *outputs a ciphertext* $\hat{m}$*.*[13]

[13] We assume without loss of generality that both pk and sk encodes the security parameter k.

- Dec*: on input a secret key* sk *and a ciphertext* $c \in \{0,1\}^*$*,* $\mathsf{Dec}_{sk}(c)$ *outputs a message* m*.*
- Eval*: on input a public key* pk*, a ciphertext* c*, and a description of a circuit* C*,* $\mathsf{Eval}_{pk}(C, c)$ *outputs a new ciphertext* c'*.*

Furthermore, for every security parameter 1^k*,* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^k)$*, every message* $m \in \{0,1\}^*$*, there exists a set* $\mathsf{Val}_{(\mathsf{pk},\mathsf{sk})}(m) \subset \{0,1\}^*$ *of* valid ciphertexts *of* m *such that*

- $\Pr[\mathsf{Enc}_{\mathsf{pk}}(m) \in \mathsf{Val}_{(\mathsf{pk},\mathsf{sk})}(m)] = 1$.
- *if* $c \in \mathsf{Val}_{(\mathsf{pk},\mathsf{sk})}(m)$*, then* $\mathsf{Eval}_{\mathsf{pk}}(C, c) \in \mathsf{Val}_{(\mathsf{pk},\mathsf{sk})}(C(m))$ *for every circuit* C*.*
- *if* $c \in \mathsf{Val}_{(\mathsf{pk},\mathsf{sk})}(m)$*, then* $\mathsf{Dec}_{\mathsf{sk}}(c) = m$*.*
- *if* $c \in \mathsf{Val}_{(\mathsf{pk},\mathsf{sk})}(m)$*, then* $|c| \leq \mathrm{poly}(|m|, k)$*.*

For the security, we consider the usual message indistinguishability against non-uniform efficient adversaries, which says that the distributions $(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(m_1))$ and $(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(m_2))$ are computationally indistinguishable for every pair of messages m_1 and m_2.

Definition 3.26 ((non-adaptive) IND-security) *A fully homomorphic encryption scheme* $\mathrm{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *is* IND*-secure if for every non-uniform* PPT *distinguisher* D*, the following holds.*

There exists a negligible function $\mathsf{ngl} : \mathbb{N} \to (0,1)$ *such that for sufficiently large security parameter* k*, for every two messages* $m_1, m_2 \in \{0,1\}^{\mathrm{poly}(k)}$ *of equal length, let* $(\mathsf{pk}, \mathsf{sk}) = \mathsf{KeyGen}(1^k)$*, we have*

$$|\Pr[\mathsf{D}(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(m_1)) = 1] - \Pr[\mathsf{D}(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(m_2)) = 1]| \leq \mathsf{ngl}(k).$$

We remark that in the usual definition of security, we also allow an adversary to pick the messages m_1 and m_2 depending on the public key pk, but we do not need it for our task. We also remark that the above definition does not imply "circuit privacy". In other words, it does not guarantee that $\mathsf{Eval}_{\mathsf{pk}}(C, \mathsf{Enc}_{\mathsf{pk}}(x))$ and $\mathsf{Enc}_{\mathsf{pk}}(C(x))$ are indistinguishable, and it is possible that the output of $\mathsf{Eval}_{\mathsf{pk}}(C, \mathsf{Enc}_{\mathsf{pk}}(x))$ contains certain partial information about the circuit C. Circuit privacy is a desirable property and can be achieved by both schemes of Gentry [11] and van Dijk et al. [35]. We do not require circuit privacy here.

Finally, we remark that for our purpose, security against non-uniform adversaries is necessary, since the soundness property is a worst-case definition on every input x in a certain domain Λ, which becomes a non-uniform advice in the security reduction.

3.5.2 The Transformation

In this section, we formally present our transformation of running a protocol under a fully homomorphic encryption scheme. We will show in the following section that our transformation preserves the round complexity and completeness and soundness property of the original protocol, and make the resulting protocol computationally simulatable.

As a warm up, let us consider the following naive implementation of the idea: In the transformed protocol $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ of $\langle \mathsf{P}, \mathsf{V} \rangle$, $\tilde{\mathsf{V}}$ generates a pair of keys $(\mathsf{pk}, \mathsf{sk})$ of a fully homomorphic encryption scheme E, publishes pk, and then $\tilde{\mathsf{P}}$ and $\tilde{\mathsf{V}}$ run the original protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ under encryption with key pk. By the fully homomorphic property, although the messages p_j, v_j are encrypted (denoted by ciphertexts $\hat{p}_j, \hat{v}_j$, respectively), $\tilde{\mathsf{V}}$ and $\tilde{\mathsf{P}}$ can still compute their encrypted responses $\hat{v}_{j+1}, \hat{p}_j$ under encryption. More precisely, $\tilde{\mathsf{V}}$ can compute $\hat{v}_{j+1}$ by applying $\mathsf{Eval}_{\mathsf{pk}}(C_{j+1}, (\hat{p}_1, \ldots, \hat{p}_j))$, where C_{j+1} is the circuit of the corresponding next-message function of V. At the end of the interaction, $\tilde{\mathsf{V}}$, who has the secret key sk, can decrypt and output the verdict bit of V.

Clearly, this preserves the round complexity as well as the completeness, since $\tilde{\mathsf{P}}$ performs exactly the same strategy as P under encryption. Intuitively, we would like to say that the soundness is preserved too, since $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ is essentially the same as $\langle \mathsf{P}, \mathsf{V} \rangle$, just executed under encryption. We would also like to say that $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ is computationally simulatable, since $\tilde{\mathsf{V}}$'s messages are ciphertexts, which are computationally indistinguishable to encryption of zero strings $\bar{0}$ and hence are easy to simulate. However, the above $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ may not preserve the soundness and may not be computationally simulatable due to the issues discussed below.

Issue with Soundness. Note that the prover strategy $\tilde{\mathsf{P}}$ receives from $\tilde{\mathsf{V}}$ ciphertexts $\hat{v}_j = \mathsf{Eval}_{\mathsf{pk}}(C_j, (\hat{p}_1, \ldots, \hat{p}_{j-1}))$ outputted by the Eval function. As mentioned, the IND-security does not imply circuit privacy. Hence, $\tilde{\mathsf{P}}$ may learn additional information from $\hat{v}_j$, such as information about the coins used to compute v_j. Such additional information may help an adversarial $\tilde{\mathsf{P}}^*$ to convince $\tilde{\mathsf{V}}$. This is also an issue for computational simulatability, since computational simulatability requires that V's future messages reveals no information about previous rounds. Furthermore, $\tilde{\mathsf{P}}^*$ may send invalid ciphertexts to $\tilde{\mathsf{V}}$. In this case, it is not even guaranteed that the interaction of $\langle \tilde{\mathsf{P}}^*, \tilde{\mathsf{V}} \rangle$ simulates the interaction of some $\langle \mathsf{P}^*, \mathsf{V} \rangle$.

To resolve this issue, we modify $\tilde{\mathsf{V}}$ to, upon receiving $\hat{p}_{j-1}$, decrypt $\hat{p}_{j-1}$ to obtain the underlying message p_{j-1}, compute the next message v_j of V, and send a fresh encryption $\hat{v}_j = \mathsf{Enc}_{\mathsf{pk}}(v_j)$ to $\tilde{\mathsf{P}}$. Here, we assume without loss of generality that $\tilde{\mathsf{V}}$ always get a certain message p_{j-1} from decryption, even if the ciphertext $\hat{p}_{j-1}$ received from $\tilde{\mathsf{P}}$ is invalid. This is easy to achieve by slightly modifying the decryption function Dec.

Note that by doing so, we guarantee that the interaction of $\langle \tilde{\mathsf{P}}^*, \tilde{\mathsf{V}} \rangle$ indeed simulates

the interaction of some $\langle \mathsf{P}^*, \mathsf{V} \rangle$. Furthermore, it can be shown that any prover strategy $\tilde{\mathsf{P}}^*$ for $\tilde{\mathsf{V}}$ can be simulated by a prover strategy P^* for V, and vice versa, which implies that $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ and $\langle \mathsf{P}, \mathsf{V} \rangle$ have exactly the same soundness.

Issue with Computational Simulatability. Recall that in our definition of the computational simulatability property in Section 3.4.1, we require a simulator to simulate random continuations of $\langle \mathsf{P}, \mathsf{V} \rangle$ from any partial interaction. In the above naive implementation, $\tilde{\mathsf{V}}$ uses a single pair of keys generated at beginning throughout the interaction. This protocol cannot be computationally simulatable, since the keys are fixed when conditioned on partial interactions.

This issue can be resolved by letting $\tilde{\mathsf{V}}$ use different fresh keys pk_j to encrypt messages in every round. By doing so, conditioning on any partial interaction does not fix the keys for encrypting the next messages, and we can simulate $\tilde{\mathsf{V}}$ using encryptions of zero $\mathsf{Enc}_{\mathsf{pk}}(\bar{0})$.

Note that to compute P's next message p_j, $\tilde{\mathsf{P}}$ needs V's previous messages $v_1, \ldots, v_j$ under encryption of the same key pk_j. One way to do is to let $\tilde{\mathsf{V}}$ send all encrypted $\hat{v}_1, \ldots, \hat{v}_j$ under the key pk_j. However, this blows up the communication complexity. A simple alternative, pointed out in [11], is to let $\tilde{\mathsf{V}}$ send $\mathsf{Enc}_{\mathsf{pk}_j}(\mathsf{sk}_{j-1})$ to $\tilde{\mathsf{P}}$. This allows $\tilde{\mathsf{P}}$ to convert ciphertexts under key pk_{j-1} to ciphertexts under key pk_j and compute P's next message under encryption of pk_j without blowing up the communication complexity of the resulting protocol.

A formal description of our transformation can be found in Figure 3.11. We analyze the transformation in the next section.

3.5.3 Analysis of Our Transformation

In this section, we analyze our transformation by proving the following simple lemmas.

Lemma 3.27 *Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a m-round interactive protocol, and $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ the corresponding transformed protocol defined in Figure 3.11. Both protocols $\langle \mathsf{P}, \mathsf{V} \rangle$ and $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ have the same completeness and soundness.*

Proof. The completeness part follows by inspection. Observe that for every input x, the interaction of $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle(x)$ perfectly simulates the interaction of $\langle \mathsf{P}, \mathsf{V} \rangle(x)$ under encryption. $\tilde{\mathsf{P}}$ convinces $\tilde{\mathsf{V}}$ with exactly the same probability as P convinces V. Hence, the completeness of the two protocols are the same.

For the soundness property, we show that for every prover strategy $\tilde{\mathsf{P}}^*$ for $\tilde{\mathsf{V}}$, there exists a corresponding prover strategy P^* for V that simulates $\tilde{\mathsf{P}}^*$ and achieves the same success probability as $\tilde{\mathsf{P}}^*$, and vice versa. This implies that the soundness of the two protocols are the same.

Transformed Protocol $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle(x)$
/* $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ runs a given m-round protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ under encryption. */
/* Let $\mathrm{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a fully homomorphic encryption scheme. */

Set the security parameter k of E to be $k = s^c$ for any fixed constant $c > 0$. For each round $j \in [m]$, upon receiving $\hat{p}_{j-1}$ from $\tilde{\mathsf{P}}$ (except for $j = 1$), $\tilde{\mathsf{V}}$ does the following.

- If $j = 1$, $\tilde{\mathsf{V}}$ generates a whole random tape R for simulating V; otherwise, $\tilde{\mathsf{V}}$ decrypts the received $\hat{p}_{j-1}$ from $\tilde{\mathsf{P}}$ to obtain $p_{j-1} = \mathsf{Dec}_{\mathsf{sk}_{j-1}}(\hat{p}_{j-1})$.
- $\tilde{\mathsf{V}}$ computes V's next message v_j using randomness R and $\tilde{\mathsf{P}}$'s messages $p_1, \ldots, p_{j-1}$.
- $\tilde{\mathsf{V}}$ generates a pair of keys $(\mathsf{pk}_j, \mathsf{sk}_j) \leftarrow \mathsf{KeyGen}(1^k)$.
- $\tilde{\mathsf{V}}$ encrypts v_j to obtain $\hat{v}_j = \mathsf{Enc}_{\mathsf{pk}_j}(v_j)$, and computes $\hat{\mathsf{sk}}_{j-1} = \mathsf{Enc}_{pk_j}(\mathsf{sk}_{j-1})$.
- $\tilde{\mathsf{V}}$ sends $\mathsf{pk}_j, \hat{v}_j$, and $\hat{\mathsf{sk}}_{j-1}$ to $\tilde{\mathsf{P}}$ (except for $j = 1$, there is no $\hat{\mathsf{sk}}_{j-1}$).

At the end, $\tilde{\mathsf{V}}$ receives $\hat{p}_m$ from $\tilde{\mathsf{P}}$, decrypts it to obtain p_m, and then computes and outputs V's verdict bit on interaction $(v_1, p_1, \ldots, v_m, p_m)$ and randomness R.

For each round $j \in [m]$, upon receiving $\hat{v}_j$ from $\tilde{\mathsf{V}}$, $\tilde{\mathsf{P}}$ does the following.

- $\tilde{\mathsf{P}}$ uses pk_j and $\hat{\mathsf{sk}}_{j-1}$ to convert encryptions $\hat{v}_1, \ldots, \hat{v}_{j-1}$ under key pk_{j-1} to that under key pk_j. This can be done by computing $\mathsf{Eval}_{\mathsf{pk}_j}(\mathsf{Dec}(\cdot;\cdot), (\hat{\mathsf{sk}}_{j-1}, \hat{v}_\ell))$, where $\hat{v}_\ell$ is an encryption under key pk_{j-1}.
- $\tilde{\mathsf{P}}$ homomorphically computes P's response p_j under encryption. I.e., $\tilde{\mathsf{P}}$ computes $\hat{p}_j = \mathsf{Eval}_{\mathsf{pk}_j}(C_j, (\hat{v}_1, \ldots, \hat{v}_j))$, where C_j denotes the circuit of the next-message function of P, and $\hat{v}_1, \ldots, \hat{v}_j$ are V's messages encrypted under key pk_j.
- $\tilde{\mathsf{P}}$ sends $\hat{p}_j$ to $\tilde{\mathsf{V}}$.

Figure 3.11: A generic transformation that makes a protocol computationally simulatable.

One direction is simple. Given any prover strategy P^* for V, we can define a corresponding prover strategy $\tilde{\mathsf{P}}^*$, which simulates P^* in the same way as $\tilde{\mathsf{P}}$ simulates P. Such a $\tilde{\mathsf{P}}^*$ has exactly the same success probability as P^* on every input x.

For the other direction, given any prover strategy $\tilde{\mathsf{P}}^*$ for $\tilde{\mathsf{V}}$, it is also not hard to define a corresponding prover strategy P^*, who interacts with V by simulating the interaction of $\tilde{\mathsf{P}}^*$ and $\tilde{\mathsf{V}}$, where P^* plays $\tilde{\mathsf{P}}^*$ and parts of $\tilde{\mathsf{V}}$. Namely, P^* generates the public keys and secret keys on his own, so he can decrypt $\tilde{\mathsf{P}}^*$'s messages and send them to V (like $\tilde{\mathsf{V}}$ does). Formally, such a reduction is given in Figure 3.12.

Prover Strategy $\mathsf{P}^*(x)$
/* P^* is given oracle access to a prover strategy $\tilde{\mathsf{P}}^*$ for $\tilde{\mathsf{V}}$. */
/* P^* interacts with V by simulating the interaction of $\langle \tilde{\mathsf{P}}^*, \tilde{\mathsf{V}} \rangle$. */

Let $\mathrm{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a fully homomorphic encryption scheme used by $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$. Set the security parameter k of E to be $k = s^c$ for any fixed constant $c > 0$. For each round $j \in [m]$, upon receiving v_j from V, P^* does the following.

- P^* generates a pair of keys $(\mathsf{pk}_j, \mathsf{sk}_j) \leftarrow \mathsf{KeyGen}(1^k)$.
- P^* encrypts v_j to obtain $\hat{v}_j = \mathsf{Enc}_{\mathsf{pk}_j}(v_j)$, and computes $\hat{\mathsf{sk}}_{j-1} = \mathsf{Enc}_{pk_j}(\mathsf{sk}_{j-1})$.
- P^* simulates $\tilde{\mathsf{P}}^*$, who receives $\mathsf{pk}_j, \hat{v}_j$, and $\hat{\mathsf{sk}}_{j-1}$, and sends back his response $\hat{p}_j$.
- P^* decrypts $\hat{p}_j$ to obtain $p_j = \mathsf{Dec}_{\mathsf{sk}_j}(\hat{p}_j)$, and sends p_j to V.

Figure 3.12: A reduction prover strategy P^*.

By inspection, for every input x, the interaction of $\langle \mathsf{P}^*, \mathsf{V} \rangle(x)$ perfectly simulates the interaction of $\langle \tilde{\mathsf{P}}^*, \tilde{\mathsf{V}} \rangle(x)$, and so P^* has the same success probability as $\tilde{\mathsf{P}}^*$. ■

Lemma 3.28 *Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a m-round interactive protocol, and $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ the corresponding transformed protocol defined in Figure 3.11. If the fully homomorphic encryption scheme E used in $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ is IND-secure, then $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ is computationally simulatable.*

Proof. We prove it by contradiction. We will show by a hybrid argument that if $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ is not computationally simulatable, then E is not IND-secure. The proof is simple, but the notation is a bit messy. We start by reviewing the notation.

Let $\tilde{\mathsf{P}}^*$ be a prover strategy. We denote a partial interaction of $\langle \tilde{\mathsf{P}}^*, \tilde{\mathsf{V}} \rangle$ by a five-tuple $(t_{[\ell]}, c_{[\ell]}, x, \tilde{p}_{[\ell]}, \tilde{v}_{[\ell]})$, where $t_{[\ell]} = (t_1, \ldots, t_\ell)$ are coins tossed by $\tilde{\mathsf{P}}^*$ in the first ℓ rounds, $c_{[\ell]}$ are coins tossed by $\tilde{\mathsf{V}}$, $\tilde{p}_{[\ell]}$ are $\tilde{\mathsf{P}}^*$'s messages, and $\tilde{v}_\ell$ are $\tilde{\mathsf{V}}$'s messages. Recall that $\tilde{\mathsf{V}}$'s messages $\tilde{v}_j$ are of the form $(\mathsf{pk}_j, \hat{v}_j, \hat{\mathsf{sk}}_{j-1})$ (except for $j = 1$, there is no $\hat{\mathsf{sk}}_{j-1}$). As mentioned, we need to be explicit about $\tilde{\mathsf{V}}$'s coins in every round. $\tilde{\mathsf{V}}$'s coins c_j tossed in round j consists of randomness for generating keys $(\mathsf{pk}_j, \mathsf{sk}_j)$ and

encryptions $\hat{v}_j, \mathsf{Enc}_{\mathsf{pk}_j}(\mathsf{sk}_{j-1})$. In addition, in the first round, c_1 also consists of R, the whole random tape used by V.

$\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ is not computationally simulatable means that there exists a PPT prover strategy $\tilde{\mathsf{P}}^*$ such that there is no PPT simulator S that can generate computationally indistinguishable random continuations of $\langle \tilde{\mathsf{P}}^*, \tilde{\mathsf{V}} \rangle$. We consider a naive simulator S that generates a random continuation by simulating $\tilde{\mathsf{P}}^*$ honestly, but using encryption of zero strings as $\tilde{\mathsf{V}}$'s messages. A formal description of S can be found in Figure 3.13. We will show that if S is not a legitimate simulator, then we can break the IND-security of E.

Simulator $\mathsf{S}(t_{[\ell]}, x, \tilde{p}_{[\ell]}, \tilde{v}_{[\ell]})$
/* S outputs a (computationally indinstinguishable) $\tilde{\mathsf{P}}^*$'s view of a random continuation of a partial interaction $(t_{[\ell]}, c_{[\ell]}, x, \tilde{p}_{[\ell]}, \tilde{v}_{[\ell]})$ of $\langle \tilde{\mathsf{P}}^*, \tilde{\mathsf{V}} \rangle$. */
S continues the interaction by simulating $\tilde{\mathsf{P}}^*$ and $\tilde{\mathsf{V}}$ alternatively as follows.

- $\tilde{\mathsf{P}}^*$: S gets $\tilde{\mathsf{P}}^*$'s view, so S can simulate $\tilde{\mathsf{P}}^*$ honestly. Namely, S generates $\tilde{\mathsf{P}}^*$'s coins $t'_{\ell+1}, \ldots, t'_m$ and applies $\tilde{\mathsf{P}}^*$'s next message function to generate $\tilde{p}'_{\ell+1}, \ldots, \tilde{p}'_m$.
- $\tilde{\mathsf{V}}$: Recall that $\tilde{\mathsf{V}}$'s messages are of the form $\tilde{v}_j = (\mathsf{pk}_j, \hat{v}_j, \hat{\mathsf{sk}}_{j-1})$. S generates the key pk_j honestly, and simulates $\tilde{\mathsf{V}}$'s message by $\hat{v}'_j = (\mathsf{pk}_j, \mathsf{Enc}_{\mathsf{pk}_j}(\bar{0}), \mathsf{Enc}_{\mathsf{pk}_j}(\bar{0}))$, where the two $\mathsf{Enc}_{\mathsf{pk}_j}(\bar{0})$'s are independent encryptions of zero strings with proper lengths.

S outputs the generated $\tilde{\mathsf{P}}^*$'s view $(t'_{[m]}, x, \tilde{p}'_{[m]}, \tilde{v}'_{[m]})$ of a complete interaction.

Figure 3.13: A candidate simulator S.

S is not a legitimate simulator of $\tilde{\mathsf{P}}^*$ means that there exists a PPT distinguisher D such that for infinitely many s, there exists a partial interaction $(t_{[\ell]}, c_{[\ell]}, x, \tilde{p}_{[\ell]}, \tilde{v}_{[\ell]})$ of $\langle \tilde{\mathsf{P}}^*, \tilde{\mathsf{V}} \rangle$ such that

$$|\Pr[\mathsf{D}(t_{[m]}, x, \tilde{p}_{[m]}, \tilde{v}_{[m]}) = 1] - \Pr[\mathsf{D}(t'_{[m]}, x, \tilde{p}'_{[m]}, \tilde{v}'_{[m]}) = 1]| > \varepsilon(s),$$

where $(t_{[m]}, x, \tilde{p}_{[m]}, \tilde{v}_{[m]}) = \langle \tilde{\mathsf{P}}^*, \tilde{\mathsf{V}} \rangle (t_{[\ell]}, c_{[\ell]}, x, \tilde{p}_{[\ell]}, \tilde{v}_{[\ell]})$, $(t'_{[m]}, x, \tilde{p}'_{[m]}, \tilde{v}'_{[m]}) = \mathsf{S}(t_{[\ell]}, x, \tilde{p}_{[\ell]}, \tilde{v}_{[\ell]})$, and ε is a non-negligible function.

We consider hybrid experiments H_i for $\ell \leq i \leq m$ defined as follows.

- **Experiment** H_i: H_i outputs a $\tilde{\mathsf{P}}^*$'s view of continuation of $(t_{[\ell]}, c_{[\ell]}, x, \tilde{p}_{[\ell]}, \tilde{v}_{[\ell]})$ as follows. For rounds $\ell+1, \ldots, i$, the view is generated by $\langle \tilde{\mathsf{P}}^*, \tilde{\mathsf{V}} \rangle$. For the remaining rounds $i+1, \ldots, m$, the view is generated by S. Namely, $\tilde{\mathsf{P}}^*$'s coins and messages are generated by $\tilde{\mathsf{P}}^*$, and $\tilde{\mathsf{V}}$'s messages of the first $i-\ell$ rounds are generated honestly and the remaining $m-i$ rounds are $(\mathsf{pk}_j, \mathsf{Enc}_{\mathsf{pk}_j}(\bar{0}), \mathsf{Enc}_{\mathsf{pk}_j}(\bar{0}))$.

By definition, experiment H_m outputs $\langle \tilde{\mathsf{P}}^*, \tilde{\mathsf{V}} \rangle(t_{[\ell]}, c_{[\ell]}, x, \tilde{p}_{[\ell]}, \tilde{v}_{[\ell]})$, and H_ℓ outputs $\mathsf{S}(t_{[\ell]}, x, \tilde{p}_{[\ell]}, \tilde{v}_{[\ell]})$. A standard hybrid argument implies that there exists an index $\ell \leq j \leq m$ such that

$$|\Pr[\mathsf{D}(H_{j-1}) = 1] - \Pr[\mathsf{D}(H_j) = 1]| > \varepsilon/(m - \ell + 1).$$

Note that the only difference between experiments H_{j-1} and H_j is that in the j-th round, $\tilde{\mathsf{V}}$'s message is $(\mathsf{pk}_j, \hat{v}_j, \mathsf{Enc}_{\mathsf{pk}_j}(\mathsf{sk}_{j-1}))$ in H_j and $(\mathsf{pk}_j, \mathsf{Enc}_{\mathsf{pk}_j}(\bar{0}), \mathsf{Enc}_{\mathsf{pk}_j}(\bar{0}))$ in H_{j-1}. By an averaging argument, we can fix randomness in both experiments except for the randomness used to generate pk_j and to encrypt (v_j, sk_{j-1}) or $(\bar{0}, \bar{0})$ while preserving the probability gap. Note that this also fixed the value of (v_j, sk_{j-1}). By hardwiring the fixed randomness, we obtain a (non-uniform) distinguisher D' such that

$$|\Pr[\mathsf{D}'(\mathsf{pk}_j, \mathsf{Enc}_{\mathsf{pk}_j}(\bar{0}), \mathsf{Enc}_{\mathsf{pk}_j}(\bar{0})) = 1] - \Pr[\mathsf{D}'(\mathsf{pk}_j, \hat{v}_j, \mathsf{Enc}_{\mathsf{pk}_j}(\mathsf{sk}_{j-1})) = 1]| > \frac{\varepsilon}{m - \ell + 1},$$

which breaks the IND-security of E. ■

We summarize the property of our transformation in the following theorem.

Theorem 3.29 *Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a m-round interactive protocol, and $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ the corresponding transformed protocol defined in Figure 3.11. $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ has the following properties.*

- *$\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ is computationally simulatable.*
- *$\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ preserves the round complexity and completeness and soundness of $\langle \mathsf{P}, \mathsf{V} \rangle$.*
- *$\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$ blows up the communication complexity of $\langle \mathsf{P}, \mathsf{V} \rangle$ by a factor of $\mathrm{poly}(k)$, where k is the security parameter of the fully homomorphic encryption scheme used in $\langle \tilde{\mathsf{P}}, \tilde{\mathsf{V}} \rangle$.*

Chapter 4

Efficient Chernoff-type and Threshold/Monotone Repetition Theorems

In this chapter, we present our efficient parallel repetition theorems for protocols with more general threshold and monotone verifiers. We first present a generic reduction showing that a good enough direct product theorem implies Chernoff-type theorems. This gives new Chernoff-type theorem for public-coin protocols and computationally simulatable protocols. We then generalize the reduction for three-message protocols from the case of direct product verifiers to the case of threshold verifiers. We present the analysis of Holenstein and Schoenebeck [22] since it gives better parameters than our original analysis [4]. Finally, we prove a parallel repetition theorem for constant-round public-coin protocols with any monotone combining function.

4.1 Chernoff-type Theorem from Direct Product Theorem

In this section, we present a simple generic transformation that converts a parallel prover strategy P^{n*} for a threshold verifier $\mathsf{V}^{n,k}$ with good success probability to a parallel prover strategy P^{t*} for a direct product verifier $\mathsf{V}^{t,t}$ with good success probability for some $t \leq n$. Composing this transformation with the direct production theorems in the previous chapter proves Chernoff-type theorems for corresponding protocols. In particular, this gives new Chernoff-type theorems for public-coin protocols and computationally simulatable protocols.

The transformation is a very naive one – P^{t*} simply embeds $\mathsf{V}^{t,t}$ in random t coordinates of $\mathsf{V}^{n,k}$, and simulate the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle$. More precisely, the reduced prover strategy P^{t*} interacts with $\mathsf{V}^{t,t}$ by simulating the interaction between P^{n*} and $\mathsf{V}^{n,k}$, where P^{t*} selects a random coordinate set $S \subset [n]$ of size t, let $\mathsf{V}^{t,t}$

play the coordinates S of $\mathsf{V}^{n,k}$ (denoted by V_S), and P^{t*} plays P^{n*} and the remaining $n-t$ coordinates of $\mathsf{V}^{n,k}$ (denoted by V_{-S}) honestly. Clearly, P^{t*} convinces $\mathsf{V}^{t,t}$ iff P^{n*} convinces subverifiers V_i for every $i \in S$, which allows us to lower bound the success probability of P^{t*} in terms of the success probability of P^{n*}.

A formal description of the above transformation can be found in Figure 4.1. We analyze the transformation in the following simple lemma.

Prover Strategy $\mathsf{P}^{t*}(x, n)$
/* P^{t*} interacts with $\mathsf{V}^{t,t}$ and is given oracle access to P^{n*}. */
/* The interaction of $\langle \mathsf{P}^{t*}, \mathsf{V}^{t,t} \rangle$ simulates the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle$ honestly. */

- P^{t*} selects a coordinate set $S = \{i_1, \ldots, i_t\} \subset [n]$ of size t uniformly at random.
- For each round $j \in [m]$, upon receiving messages $\vec{v}_{j,S} = (v_{j,i_1}, \ldots, v_{j,i_t})$ from $\mathsf{V}^{t,t}$, P^{t*} samples coins $\vec{c}_{j,-S}$ uniformly at random (the fresh coins tossed by V_{-S} at round j), simulates V_{-S} to generate $\vec{v}_{j,-S}$, simulates P^{n*} to generate $\vec{p}_j$, and sends $p_{j,S} = (p_{j,i_1}, \ldots, p_{j,i_t})$ to $\mathsf{V}^{t,t}$.

Figure 4.1: Reduction prover strategy P^{t*}.

Lemma 4.1 *Let* V *be a* PPT *verifier, and* $t, k, n \in \mathbb{N}$ *such that* $1 \leq t \leq k \leq n$. *Let* P^{n*} *be a parallel prover strategy for* $\mathsf{V}^{n,k}$, *and* P^{t*} *the reduction prover strategy defined in Figure 4.1. For every common input* $x \in \{0,1\}^*$, *we have*

1. *The success probability of* P^{t*} *is at least*

$$\Pr[\langle \mathsf{P}^{t*(\mathsf{P}^{n*})}, \mathsf{V}^{t,t} \rangle(x) = 1] \geq \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle(x) = 1] \cdot \frac{\binom{k}{t}}{\binom{n}{t}}.$$

2. $\mathsf{P}^{t*(\cdot)}(x, n)$ *runs in time* $\mathrm{poly}(|x|, n)$ *given oracle access to* $\mathsf{P}^{n*}(x)$.

Proof. The assertion about the runtime of P^{t*} follows by inspection. For the success probability, by construction, we have

$$\begin{aligned}
&\Pr[\langle \mathsf{P}^{t*}, \mathsf{V}^{t,t} \rangle(x) = 1] \\
&\quad \geq \Pr[(\mathsf{V}_i \text{ accepts } \forall i \in S \text{ in } \langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle(x)) \wedge (\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle(x) = 1)] \\
&\quad = \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle(x) = 1] \cdot \Pr\left[\mathsf{V}_i \text{ accepts } \forall i \in S \text{ in } \langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle(x) \,\middle|\, \langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle(x) = 1\right] \\
&\quad \geq \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle(x) = 1] \cdot \frac{\binom{k}{t}}{\binom{n}{t}}.
\end{aligned}$$

■

As mentioned, composing the above transformation with direct product theorems yields Chernoff-type theorems. In particular, if the direct product theorem is tight, then the resulting Chernoff-type theorem matches the information-theoretic Chernoff bounds up to a constant in the exponent. More precisely, suppose $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ, and consider threshold $k = (1+\gamma) \cdot \delta n$ for some constant $\gamma \in (0,1)$. The resulting Chernoff-type theorem (proved below) says that $\langle \mathsf{P}^n, \mathsf{V}^{n,k} \rangle$ has soundness error $e^{-\gamma^2 \delta n/3}$ (c.f. the standard Chernoff bound gives bound $e^{-\gamma^2 \delta n/2}$).

For some settings such as three-message protocols and constant-round public-coin protocols, we can prove better Chernoff-type theorems by constructing direct reductions. However, for public-coin protocols and computationally simulatable protocols, we obtain interesting Chernoff-type theorems by the above simple argument. One limitation of this argument is that it gives interesting result only when $\delta = \Omega(1)$. Nevertheless, this is sufficient for most applications.

Formally, we prove the following two Chernoff-type theorems. The first theorem says that for public-coin protocols with threshold $k = (1+\gamma) \cdot \delta n$, the soundness errors decreases from δ to $e^{-\gamma^2 \delta n/3}$. For computationally simulatable protocols, our theorem requires the threshold $k > \sqrt{\delta} \cdot n$ to guarantee the decrease of soundness error. The second theorem says that for computationally simulatable protocols with threshold $k = (1+\gamma) \cdot \sqrt{\delta} \cdot n$, the soundness errors decreases from δ to $e^{-\gamma^2 \sqrt{\delta} \cdot n/3}$. Note that the bound obtained for computationally simulatable protocols is significantly worse than that of public-coin protocols. Indeed, the resulting bounds for Chernoff-type verifiers obtained by our reduction are very sensitive to the corresponding bounds for direct product verifiers. Our reduction does not give interesting results for weakly simulatable protocols.

Theorem 4.2 *Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a public-coin protocol with input domain Λ, and let $\delta, \gamma \in (0,1)$ be constants, and $n : \mathbb{N} \to \mathbb{N}$ efficiently computable functions with $n \leq \mathrm{poly}(s)$. Let $k = (1+\gamma)\delta n$. If $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ, then its n-fold parallel repetition with threshold verifier $\langle \mathsf{P}^n, \mathsf{V}^{n,k} \rangle$ has soundness error $e^{-\lfloor \gamma^2 \delta n/3 \rfloor} + \mathsf{ngl}$, where ngl denotes a negligible function in the security parameter s.*

Theorem 4.3 *Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a computationally simulatable protocol with input domain Λ, and let $\delta, \gamma \in (0,1)$ be constants, and $n : \mathbb{N} \to \mathbb{N}$ efficiently computable functions with $n \leq \mathrm{poly}(s)$. Let $k = (1+\gamma)\sqrt{\delta} \cdot n$. If $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ, then its n-fold parallel repetition with threshold verifier $\langle \mathsf{P}^n, \mathsf{V}^{n,k} \rangle$ has soundness error $e^{-\lfloor \gamma^2 \delta n/3 \rfloor} + \mathsf{ngl}$, where ngl denotes a negligible function in the security parameter s.*

Both theorems can be proved by applying Lemma 4.1 and the corresponding direct-product theorem straightforwardly. However, the choice of t and the proof requires some tedious calculations. For the sake of intuition, before proving the theorems formally, we give some informal discussions.

Let us consider the public-coin case and try to prove that $\langle \mathsf{P}^n, \mathsf{V}^{n,k} \rangle$ has soundness error ε. For the sake of contradiction, we assume there exists a P^{n*} with success

probability ε. By Lemma 4.1, we have a P^{t*} with success probability $\varepsilon \cdot \binom{k}{t}/\binom{n}{t}$, for some t to be determined later. Then by the direct product theorem for public-coin protocols, we have a P^* with success probability

$$\left(\varepsilon \cdot \frac{\binom{k}{t}}{\binom{n}{t}}\right)^{1/t}.$$

To obtain a contradiction, we need the above quantity to be greater than δ. Rearranging the term, it suffices for

$$\varepsilon > \delta^t \cdot \frac{\binom{n}{t}}{\binom{k}{t}}.$$

We should choose t that minimizes the RHS to obtain the best bound. Let us try to see how small the RHS can be. Observe that when $t = 0$, the RHS $= 1$, and when t is increased by 1, the RHS is multiplied by a factor $\delta \cdot (n-t)/(k-t)$. The factor is equal to $1/(1+\gamma) < 1$ when $t = 0$, and increases as t increasing. The factor becomes 1 when $t = \gamma\delta n/(1-\delta)$, so we should set t to be this quantity. Noting that $1/(1+\gamma) = e^{-\Omega(\gamma)}$, and there are t factors multiplied together, the RHS should be roughly $e^{-\Omega(t\cdot\gamma)} = e^{-\Omega(\gamma^2\delta n)}$, which is the bound in Theorem 4.2. We proceed to prove Theorem 4.2 and 4.3 below.

Proof. (of Theorem 4.2) We prove it by contradiction. Suppose the conclusion is not true, then there exists a PPT parallel prover P^{n*} and a noticeable η such that for infinitely many $s \in \mathbb{N}$, there exists some x with security parameter s such that

$$\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k}\rangle(x) = 1] > e^{-\lfloor \gamma^2\delta n/3\rfloor} + \eta(s) \stackrel{\text{def}}{=} \varepsilon_1.$$

Applying transformation in Figure 4.1 with

$$t = \min\left\{\left\lfloor \frac{\gamma\delta n}{1-\delta}\right\rfloor, \left\lceil \frac{3}{\gamma}\ln\frac{2}{\eta}\right\rceil\right\},$$

we obtain an efficient parallel prover strategy P^{t*} for direct product verifier $\mathsf{V}^{t,t}$. By Lemma 4.1, we have

$$\Pr[\langle \mathsf{P}^{t*}, \mathsf{V}^{t,t}\rangle(x) = 1] \geq \left(\varepsilon_1 \cdot \frac{\binom{k}{t}}{\binom{n}{t}}\right) \stackrel{\text{def}}{=} \varepsilon_2.$$

Now, applying Theorem 3.2 with parameters t, ε_2, and $\xi = \eta/4$, we obtain a single-instance prover strategy P^* with

$$\Pr[\langle \mathsf{P}^*, \mathsf{V}\rangle(x) = 1] \geq \varepsilon_2^{1/t} \cdot (1 - \eta/4).$$

By the choice of t and recall that δ is a constant, we have

$$\varepsilon_2 \geq \varepsilon_1 \cdot \left(\frac{k-t}{n-t}\right)^t \geq \varepsilon_1 \cdot \delta^t \geq 1/\mathrm{poly}(s).$$

Hence, P^* has runtime $\mathrm{poly}(|x|, t, \varepsilon_2^{-1}, \xi^{-1}) = \mathrm{poly}(s)$, which is efficient. To obtain a contradiction, it remains to show that

$$\varepsilon_2^{1/t} \cdot \left(1 - \frac{\eta}{4}\right) = \left(\left(e^{-\lfloor \gamma^2 \delta n/3 \rfloor} + \eta\right) \cdot \frac{\binom{k}{t}}{\binom{n}{t}}\right)^{1/t} \cdot \left(1 - \frac{\eta}{4}\right) > \delta.$$

To simplify the expression a bit, we note that

$$\left(e^{-\lfloor \gamma^2 \delta n/3 \rfloor} + \eta\right) \geq \left(e^{-\lfloor \gamma^2 \delta n/3 \rfloor} + \frac{\eta}{2}\right) \cdot \left(1 + \frac{\eta}{2}\right),$$

and hence

$$\left(\left(e^{-\lfloor \gamma^2 \delta n/3 \rfloor} + \eta\right) \cdot \frac{\binom{k}{t}}{\binom{n}{t}}\right)^{1/t} \cdot \left(1 - \frac{\eta}{4}\right) \geq \left(\left(e^{-\lfloor \gamma^2 \delta n/3 \rfloor} + \frac{\eta}{2}\right) \cdot \frac{\binom{k}{t}}{\binom{n}{t}}\right)^{1/t}.$$

Now, we lower bound the term $\binom{k}{t}/\binom{n}{t}$ by (proved in Lemma 4.4 below)

$$\frac{\binom{k}{t}}{\binom{n}{t}} \geq e^{\gamma t/3} \cdot \delta^t.$$

Hence, we are reduced to show that

$$\left(\left(e^{-\lfloor \gamma^2 \delta n/3 \rfloor} + \frac{\eta}{2}\right) \cdot e^{t \cdot \gamma/3} \cdot \delta^t\right)^{1/t} > \delta.$$

which follows by observing that

$$e^{t \cdot \gamma/3} \geq \min\left\{e^{\lfloor \gamma^2 \delta n/3 \rfloor}, \frac{2}{\eta}\right\}.$$

In sum, the prover strategy P^* obtained above is efficient and

$$\Pr[\langle \mathsf{P}^*, \mathsf{V} \rangle(x) = 1] > \delta.$$

This contradicts to the fact that $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ and completes the proof. ■

Lemma 4.4 *Let $n, k, t \in \mathbb{N}$, and $\gamma, \delta \in (0, 1)$ be numbers satisfying (i) $1 \leq t \leq k \leq n$, (ii) $k = (1 + \gamma)\delta n$, and (iii) $t \leq (\gamma \delta n)/(1 - \delta)$. We have*

$$\frac{\binom{k}{t}}{\binom{n}{t}} \geq e^{\gamma t/3} \cdot \delta^t.$$

Proof. We prove it by a brute-force calculation as follows. Observe that

$$\frac{\binom{k}{t}}{\binom{n}{t}} = \frac{k \cdot (k-1) \ldots (k-t+1)}{n \cdot (n-1) \ldots (n-t+1)} = \prod_{x=0}^{t-1} \frac{k-x}{n-x}.$$

We express each term $(k-x)/(n-x)$ of the form $(1+\beta(x)) \cdot \delta$, where

$$\beta(x) = \frac{\gamma\delta n - (1-\delta)x}{\delta(n-x)}.$$

Using inequality $(1+\beta) \geq e^{(2/3)\cdot\beta}$ (holds for $\beta \in [0,1]$), we obtain

$$\prod_{x=0}^{t-1} \frac{k-x}{n-x} = \prod_{x=0}^{t-1} (1+\beta(x)) \cdot \delta \geq e^{(2/3)\cdot\sum_{x=0}^{t-1}\beta(x)} \cdot \delta^t.$$

Observing that for $0 \leq x \leq t$,

$$\beta(x) = \frac{\gamma\delta n - (1-\delta)x}{\delta(n-x)} \geq \frac{\gamma\delta n - (1-\delta)x}{\delta n} \stackrel{\text{def}}{=} \lambda(x),$$

we can lower bound $\sum_{x=0}^{t-1}\beta(x)$ by $\sum_{x=0}^{t-1}\lambda(x)$, which is a sum of arithmetic progression. Note that $\lambda(0) = \gamma$, and $\lambda(t-1) \geq 0$ for $t \leq (\gamma\delta n)/(1-\delta)$, we have

$$\sum_{x=0}^{t-1} \lambda(x) \geq \frac{t \cdot \gamma}{2}.$$

It follows that

$$\frac{\binom{k}{t}}{\binom{n}{t}} \geq e^{(2/3)\cdot\sum_{x=0}^{t-1}\beta(x)} \cdot \delta^t \geq e^{(2/3)\cdot\sum_{x=0}^{t-1}\lambda(x)} \cdot \delta^t \geq e^{t\cdot\gamma/3} \cdot \delta^t,$$

as desired. ■

We proceed to proof Theorem 4.3.

Proof. (of Theorem 4.3) We prove it by contradiction. By Theorem 3.16, $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ implies that for every efficiently computable t, the parallel protocol $\langle \mathsf{P}^t, \mathsf{V}^{t,t} \rangle$ has soundness error $\delta^{t/2} + \mathsf{ngl}$. We will derive a contradiction to this statement assuming the conclusion is not true.

Suppose the conclusion is not true, then there exists a PPT parallel prover P^{n*} and a noticeable η such that for infinitely many $s \in \mathbb{N}$, there exists some x with security parameter s such that

$$\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle(x) = 1] > e^{-\lfloor \gamma^2\sqrt{\delta}\cdot n/3 \rfloor} + \eta(s) \stackrel{\text{def}}{=} \varepsilon_1.$$

Applying transformation in Figure 4.1 with

$$t = \min\left\{\left\lfloor \frac{\gamma\sqrt{\delta}\cdot n}{1-\sqrt{\delta}}\right\rfloor, \left\lceil \frac{3}{\gamma}\ln\frac{2}{\eta}\right\rceil\right\},$$

we obtain an efficient parallel prover strategy P^{t*} for direct product verifier $\mathsf{V}^{t,t}$. By Lemma 4.1, we have

$$\Pr[\langle \mathsf{P}^{t*}, \mathsf{V}^{t,t}\rangle(x) = 1] \geq \left(\varepsilon_1 \cdot \frac{\binom{k}{t}}{\binom{n}{t}}\right) \stackrel{\text{def}}{=} \varepsilon_2.$$

Note that by the choice of t and the fact that δ is a constant, δ^t is non-negligible. To contradict to the fact that $\langle \mathsf{P}^t, \mathsf{V}^{t,t}\rangle$ has soundness error $\delta^{t/2} + \mathsf{ngl}$, it suffices to show that $\varepsilon_2 \geq \delta^{t/2}(1+\eta')$, for some non-negligible η'.

We again use Lemma 4.4 to estimate $\binom{k}{t}/\binom{n}{t}$. Substituting δ by $\sqrt{\delta}$ in Lemma 4.4, we have

$$\frac{\binom{k}{t}}{\binom{n}{t}} \geq e^{\gamma t/3}\cdot\left(\sqrt{\delta}\right)^t.$$

Hence,

$$\varepsilon_2 = \left(e^{-\lfloor\gamma^2\sqrt{\delta}\cdot n/3\rfloor} + \eta\right)\cdot\frac{\binom{k}{t}}{\binom{n}{t}} \geq \left(e^{-\lfloor\gamma^2\sqrt{\delta}\cdot n/3\rfloor} + \eta\right)\cdot e^{\gamma t/3}\cdot\left(\sqrt{\delta}\right)^t \geq (1+\eta')\cdot\delta^{t/2},$$

for some non-negligible η', where the last inequality follows by the fact that

$$e^{t\cdot\gamma/3} \geq \min\left\{e^{\lfloor\gamma^2\sqrt{\delta}\cdot n/3\rfloor}, \frac{2}{\eta}\right\}.$$

In sum, the prover strategy P^{t*} obtained above is efficient and

$$\Pr[\langle \mathsf{P}^{t*}, \mathsf{V}^{t,t}\rangle(x) = 1] > \delta^{t/2}\cdot(1+\eta').$$

This contradicts to the fact that $\langle \mathsf{P}^t, \mathsf{V}^{t,t}\rangle$ has soundness error $\delta^{t/2}+\mathsf{ngl}$ and completes the proof. ■

4.1.1 Discussion

As mentioned in the introduction, the Chernoff-type theorems resulting from our reduction for public-coin protocols and computationally simulatable protocols are incomparable to the Chrenoff-type theorems of Haståd, Pass, Wikström, and Pietrzak [20]. They stated their theorems in a different form. We first present their theorems (but in our notation), and then compare the bounds.

Theorem 4.5 ([20]) *Assume $\varepsilon \leq 1/2$ and let $\mathsf{V} \in$ PPT be a m-round verifier and let P^{n*} be a polynomial-time parallel prover. Then there exists a prover strategy P^* running in time $\mathrm{poly}(|x|, n, m, 1/\varepsilon)$ such that for every $x \in \{0,1\}^*$ where*

$$\Pr[\langle \mathsf{P}^{n*}(n,k,\varepsilon), \mathsf{V}^{n,k}\rangle(x) = 1] \geq \varepsilon,$$

for threshold $k = (1-\alpha)n$ with $0 \leq \alpha < 1$, we have that

1. *if V is "1-simulatable with verdict", then*

$$\Pr[\langle \mathsf{P}^{*(\mathsf{P}^{n*})}, \mathsf{V}\rangle(x) = 1] \geq 1 - \alpha - 2\sqrt{\frac{\log(1/\varepsilon)}{n}} - \sqrt{\frac{1}{n}}, \text{ and}$$

2. *if V is "1-simulatable without verdict", then*

$$\Pr[\langle \mathsf{P}^{*(\mathsf{P}^{n*})}, \mathsf{V}\rangle(x) = 1] \geq 1 - \alpha - O\left(\sqrt{\frac{m \cdot \log(1/\varepsilon)}{n}} + \sqrt{\frac{m}{n}} \cdot \log(mn)\right)$$

Haståd, et al. [20] state their results in terms of the "simulatability" property of protocols (which we refer to as weak simulatability property). Informally, "1-simulatable with verdict" and "1-simulatable without verdict" essentially correspond to public-coin protocols and computationally simulatable protocols, respectively.[1] They also state their results using different parameterizations. They fix the the success probability ε of the parallel prover P^{n*}, and lower bound the success probability of the reduction prover strategy P^*. In contrast, we assume the original protocol $\langle \mathsf{P}, \mathsf{V}\rangle$ has soundness error δ, and upper bound the soundness error of the parallel protocol $\langle \mathsf{P}^n, \mathsf{V}^{n,k}\rangle$.

To compare the results, we translate their bounds to our parameter settings. For public-coin protocols, their bound says that if $\langle \mathsf{P}, \mathsf{V}\rangle$ has soundness error δ, then the corresponding parallel protocol $\langle \mathsf{P}^n, \mathsf{V}^{n,k}\rangle$ with $k = (1+\gamma)\delta n$ and $\gamma \in (0,1)$ has soundness error $e^{-\gamma^2\delta^2 n/4} + \mathsf{ngl}$, which is slightly worse than our bound $e^{-\gamma^2\delta n/3} + \mathsf{ngl}$. For computationally simulatable protocols, their bound is $e^{-\Omega(\gamma^2\delta^2 n/m)} + \mathsf{ngl}$, which depends on the number m of rounds. On the other hand, our reduction does not give result for threshold $k = (1+\gamma)\delta n$. We need a larger threshold $k \geq (1+\gamma)\sqrt{\delta} \cdot n$ to guarantee the decrease of soundness error. We obtain bound $e^{-\gamma^2\sqrt{\delta}\cdot n/3} + \mathsf{ngl}$, which is independent of m. Our bound is better when δ is close to 1 and m is large.

[1] The definitions are not exactly equivalent, but for example, the results hold for the case of 1-simulatable without verdict also hold for the case of computationally simulatable protocols, and vice versa, since the reduction can be implemented in the corresponding settings.

4.2 Efficient Threshold Repetition Theorem for Three-Message Protocols

In this section, we generalize the reduction of Canetti, Halevi, and Steiner [2] presented in Section 3.3 to prove a tight threshold repetition theorem for three-message protocols.

In the literature, there have been many proofs of parallel repetition theorems for three-message protocols (where some of them were presented in different models). Bellare, Impagliazzo, Naor [1] proved the first direct product theorem, which says that n-fold parallel repetition decreases the soundness error from $(1-\varepsilon)$ to $\varepsilon^{\Omega(\varepsilon^2 n)}+\mathsf{ngl}$. The bound is later improved to optimal by Canetti, Halevi, and Steiner [2] in the context of "weakly verifiable puzzle systems". The first Chernoff-type theorem was proved by Impagliazzo, Jaiswal, and Kabanets [23, 24]. They showed that for threshold $k = (1+\gamma)\delta n$, n-fold parallel repetition with threshold verifier $\mathsf{V}^{n,k}$ decreases the soundness error from δ to $\varepsilon^{\Omega(\gamma^2\delta n)} + \mathsf{ngl}$. In the context of security amplification for commitment scheme, Helavi and Rabin [17] proved a "hardness degradation theorem", which corresponds to threshold repetition theorem with threshold $k = 1$.

We prove a threshold repetition theorem for three-message protocols, which generalizes and improves the above previous results. In particular, for the Chernoff-type region (i.e., the threshold $k \geq (1+\gamma)\delta n$ for some constant γ), our bounds match the information-theoretic bounds up to a necessary negligible term. For convenience, we introduce the following notation to refer to the information-theoretic bounds.

Definition 4.6 *Let $n, k \in \mathbb{N}$ and $\delta \in (0,1)$ be parameters. We define $X_1, \ldots, X_n$ to be i.i.d. binary random variable with $\Pr[X_i = 1] = \delta$, and*

$$P(n,k,\delta) \stackrel{\text{def}}{=} \Pr\left[\sum_{i=1}^{n} X_i \geq k\right],$$

i.e., the probability that at least k out of n independent events happens, where each event happens with probability δ.

We prove that if $\langle \mathsf{P}, \mathsf{V}\rangle$ has constant soundness error δ, then $\langle \mathsf{P}^n, \mathsf{V}^{n,k}\rangle$ with $k \geq (1+\gamma)\delta n$ has soundness error $P(n,k,\delta) + \mathsf{ngl}$. For general threshold k, we obtain a slightly worse bound $P(n,k,\delta+\alpha) + \mathsf{ngl}$, where α is an arbitrarily small constant slackness parameter. As mentioned, we prove our result by a natural generalization of the reduction of Canetti et al. [2].

Independent of our work, Holenstein and Schoenebeck [22] proved tight efficient parallel repetition theorem for any monotone 3-message verifier. They considered the same reduction as ours and came up with a better analysis to handle errors, which allows them to prove an optimal parallel repetition theorem. There is another independent work of Jutla [26], which improved the Chernoff-type theorem of Impagliazzo

et al. [24] by a variant of their reduction. Jutla improved the constant in the exponent of the bound to match that of the Chernoff bound. However, the bound is worse than the optimal information-theoretic bounds obtained by us and Holenstein and Schoenebeck.

Formally, we prove the following threshold repetition theorem for three-message protocols in this section. We present the better analysis of Holenstein and Schoenebeck [22] since it is more elegant and gives better parameters. We will discuss how our original analysis was different from that of Holenstein and Schoenebeck later in this section. The following theorem says that if $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ, then the corresponding parallel protocol $\langle \mathsf{P}^n, \mathsf{V}^{n,k} \rangle$ has soundness error $P(n,k,\delta) + \mathsf{ngl}$.

Theorem 4.7 *Let $\mathsf{V} \in$ PPT be a three-message verifier. There exists a prover strategy P^* such that for every common input $x \in \{0,1\}^*$, every $n, k \in \mathbb{N}$ with $k \leq n$, every $\delta, \xi \in (0,1)$, and every parallel prover strategy P^{n*},*

1. $\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle(x) = 1] \geq P(n,k,\delta) + \xi \Rightarrow$

$$\Pr[\langle \mathsf{P}^{*(\mathsf{P}^{n*})}(n,k,\delta,\xi), \mathsf{V} \rangle(x) = 1] \geq \delta + \frac{\xi}{10n}.$$

2. $\mathsf{P}^{*(\cdot)}(x,n,k,\delta,\xi)$ *runs in time* $\mathrm{poly}(|x|, n, \xi^{-1})$ *given oracle access to* $\mathsf{P}^{n*}(x)$.

We first recall the notation and ideas for proving direct product theorems from Section 3.3. Recall that the three messages of $\langle \mathsf{P}^*, \mathsf{V} \rangle$ and $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle$ are denoted by w, v, p and $\vec{w}, \vec{v}, \vec{p}$ respectively. We use c to denote V's private coins, and write $v = \mathsf{V}(w, c)$ or simply $v = \mathsf{V}(c)$ when the prover's first message w is clear from the context. We assume without loss of generality that P^{n*} is *deterministic*, and hence $\vec{w}$ is fixed and the outcome of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle(x)$ is determined by $\mathsf{V}^{n,k}$'s private coins $\vec{c} = (c_1, \ldots, c_n)$. With a slight abuse of notation, we write "$\mathsf{P}^{n*}(\vec{c})$ convinces V_{-i}," or "$\mathsf{P}^{n*}(v_i, \vec{c}_{-i})$ convinces V_{-i}," to denote the subverifiers V_{-i} accepting in the corresponding interaction $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle(x)$. Similarly, "$\mathsf{P}^{n*}(\vec{c})$ convinces $\geq t$ of $\mathsf{V}^{n,k}$," and "$\mathsf{P}^{n*}(\vec{c})$ convinces $\geq t$ of V_{-1}," mean at least t subverifiers of $\mathsf{V}^{n,k}$ and V_{-1} accept in the corresponding interaction, respectively.

Also recall the common framework that the interaction of $\langle \mathsf{P}^*, \mathsf{V} \rangle$ simulates the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle$, where (1) P^* first selects coordinate i, (2) V generates random coins $c = c_i$, which determine the message $v_i = \mathsf{V}_i(c_i)$, and then (3) P^* selects the remaining $n-1$ sequences of coins $\vec{c}_{-i}$, which determine messages $\vec{v}_{-i}$. P^* succeeds iff P^{n*} convinces V_i in the corresponding interaction (i.e., V_i with coins c_i accepts the interaction (w_i, v_i, p_i).).

Finally, recall that the challenge is that P^* can only compute the decision of the $n-1$ internal subverifiers V_{-i} but cannot predict V's decision from the transcript, since P^* does not know V's private coin c_i. In Section 3.3, we considered a naive strategy, where P^* only checks whether all internal subverifiers accept. More precisely, P^*

uses sampling to select a $\vec{c}_{-i}$ such that P^{n*} convinces all V_{-i} in the corresponding interaction.

We observed that this naive strategy does not work when there are"bad correlations" in the "success pattern" of P^{n*}. Then, we presented a correlation reduction in Figure 3.5, which exploits such bad correlations to convert a parallel prover strategy P^{n*} to another parallel prover strategy $\mathsf{P}^{n'*}$ with smaller $n' < n$ and without such bad correlations.

More precisely, we sample coins c_1^* of the first subverifier V_1, and estimate by sampling the probability that P^{n*} convinces V_{-1} conditioned on V_1's coins are c_1^*. If the probability is at least δ^{n-1}, then we can construct a prover strategy P^{n-1*} that convinces $\mathsf{V}^{n-1,n-1}$ with probability at least δ^{n-1}, and we can iterate this process on P^{n-1*}. On the other hand, if we cannot find such c_1^* after many samples, we obtain an extra property that for most coins c_1^*, $\Pr[\mathsf{P}^{n*}$ convinces $\mathsf{V}_{-1}|c_1 = c_1^*] \leq \delta^{n-1}$.[2] We proved that the naive strategy works when applies to a prover strategy P^{n*} with this extra property. Intuitively (though not accurate[3]), this is because

$$\begin{aligned} \Pr[\mathsf{P}^* \text{ convinces } \mathsf{V}] \quad &\text{“=”} \quad \Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}^{n,n} | \mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-1}] \\ &= \quad \frac{\Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}^{n,n}]}{\Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_{-1}]} \geq \delta \end{aligned}$$

For the case of threshold verifiers, we consider the following natural generalization. The starting point is a parallel prover strategy P^{n*} such that

$$\Pr[\mathsf{P}^{n*} \text{ convinces} \geq k \text{ of } \mathsf{V}^{n,k}] \geq P(n,k,\delta).$$

We apply a natural generalization of the correlation reduction procedure to convert P^{n*} to some $\mathsf{P}^{n'*}$. There are two natural conditions to consider. First, if there exist coins c_1^* of subverifier V_1 such that

$$\Pr[\mathsf{P}^{n*} \text{ convinces} \geq k \text{ of } \mathsf{V}_{-1} | c_1 = c_1^*] \geq P(n-1,k,\delta),$$

then we can construct a P^{n-1*} that convinces $\mathsf{V}^{n-1,k}$ with probability $P(n-1,k,\delta)$, and iterate the process on P^{n-1*}. Similarly, if there exist coins c_1^* of subverifier V_1 such that

$$\Pr[\mathsf{P}^{n*} \text{ convinces} \geq k-1 \text{ of } \mathsf{V}_{-1} | c_1 = c_1^*] \geq P(n-1,k-1,\delta),$$

[2]For intuition, we omit various sampling errors and necessary slackness parameters in the following informal discussion.

[3]The success probability of P^* is actually $\mathrm{E}_{c_1^*}[\Pr[\mathsf{P}^{n*}$ convinces $\mathsf{V}^{n,n} | \mathsf{P}^{n*}$ convinces $\mathsf{V}_{-1} | c_1 = c_1^*]]$, and hence, to lower bound the success probability of P^*, we need the property that $\Pr[\mathsf{P}^{n*}$ convinces $\mathsf{V}_{-1} | c_1 = c_1^*] \leq \delta^{n-1}$ for most c_1^*, as opposed to only $\Pr[\mathsf{P}^{n*}$ convinces $\mathsf{V}_{-1}] \leq \delta^{n-1}$. Nevertheless, it is instructive to think of the above (over-simplified) formula for intuition.

then we can construct a P^{n-1*} that convinces $\mathsf{V}^{n-1,k-1}$ with probability $P(n-1,k-1,\delta)$, and iterate the process on P^{n-1*}. On the other hand, if we cannot find such c_1^* after many samples, we obtain extra properties that for most coins c_1^*,

- $\Pr[\mathsf{P}^{n*} \text{ convinces} \geq k \text{ of } \mathsf{V}_{-1} | c_1 = c_1^*] \leq P(n-1,k,\delta)$,
- $\Pr[\mathsf{P}^{n*} \text{ convinces} \geq k-1 \text{ of } \mathsf{V}_{-1} | c_1 = c_1^*] \leq P(n-1,k-1,\delta)$.

Intuitively, since P^{n*} cannot convince V_{-1} well by the above conditions, in order to convinces $\mathsf{V}^{n,k}$ with good probability, P^{n*} must convince V_1 well, especially when P^{n*} convinces *exactly* $k-1$ of V_{-1}. Therefore, we consider a variant of the naive strategy, where P^* embeds $\mathsf{V} = \mathsf{V}_1$ at the first coordinate of $\mathsf{V}^{n,k}$, and uses sampling to select a $\vec{c}_{-1}$ such that P^{n*} convinces exactly $k-1$ subverifiers of V_{-1}. Again, intuitively (though not accurate), we have

$$
\begin{aligned}
&\Pr[\mathsf{P}^* \text{ convinces } \mathsf{V}] \\
\text{“}=\text{”} \quad &\Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_1 \wedge (k-1 \text{ of } \mathsf{V}_{-1})] \\
= \quad &\frac{\Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}_1 \wedge (k-1 \text{ of } \mathsf{V}_{-1})]}{\Pr[\mathsf{P}^{n*} \text{ convinces } k-1 \text{ of } \mathsf{V}_{-1}]} \\
= \quad &\frac{\Pr[\mathsf{P}^{n*} \text{ convinces} \geq k \text{ of } \mathsf{V}^{n,k}] - \Pr[\mathsf{P}^{n*} \text{ convinces} \geq k \text{ of } \mathsf{V}_{-1}]}{\Pr[\mathsf{P}^{n*} \text{ convinces} \geq k-1 \text{ of } \mathsf{V}_{-1}] - \Pr[\mathsf{P}^{n*} \text{ convinces} \geq k \text{ of } \mathsf{V}_{-1}]} \\
\geq \quad &\frac{P(n,k,\delta) - \Pr[\mathsf{P}^{n*} \text{ convinces} \geq k \text{ of } \mathsf{V}_{-1}]}{P(n-1,k-1,\delta) - \Pr[\mathsf{P}^{n*} \text{ convinces} \geq k \text{ of } \mathsf{V}_{-1}]} \\
\geq \quad &\frac{P(n,k,\delta) - P(n-1,k,\delta)}{P(n-1,k-1,\delta) - P(n-1,k,\delta)} \\
= \quad &\frac{\Pr[X_1 = 1 \wedge \sum_{i=2}^n X_i = k-1]}{\Pr[\sum_{i=2}^n X_i = k-1]} = \delta,
\end{aligned}
$$

where

- The numerator of the third equality says that the event that P^{n*} convinces V_1 and exactly $k-1$ of V_{-1} is equivalent to the event that P^{n*} convinces at least k of $\mathsf{V}^{n,k}$ but less than k of V_{-1}.
- The denominator of the third equality says that the event that P^{n*} convinces exactly $k-1$ of V_{-1} is equivalent to the event that P^{n*} convinces at least $k-1$ of V_{-1} but less than k of V_{-1}.
- The first inequality follows by the fact that $\Pr[\mathsf{P}^{n*} \text{ convinces} \geq k \text{ of } \mathsf{V}^{n,k}] \geq P(n,k,\delta)$ and $\Pr[\mathsf{P}^{n*} \text{ convinces} \geq k-1 \text{ of } \mathsf{V}_{-1}] \leq P(n-1,k-1,\delta)$.
- The second inequality follows by the fact that $\Pr[\mathsf{P}^{n*} \text{ convinces} \geq k \text{ of } \mathsf{V}_{-1}] \leq P(n-1,k,\delta)$ and the inequality

$$\frac{b}{a} \geq \frac{b-x}{a-x}$$

for any $a \geq b \geq x \geq 0$.

Note that the key of the above calculation is the third equality, where we decompose the events that used by the naive reduction prover strategy into the events that controlled by the correlation reduction procedure.

The aforementioned natural generalization of the reduction of Canetti et al. [2] presented in Section 3.3 indeed works. However, it requires a careful implementation. In particular, note that the generalized correlation reduction procedure converts a P^{n*} for $\mathsf{V}^{n,k}$ with success probability $P(n,k,\delta)$ to some $\mathsf{P}^{n'*}$ for $\mathsf{V}^{n',k'}$ with success probability $P(n',k',\delta)$, for some $n' \leq n$ and $k' \leq k'$. It is possible that $P(n',k',\delta) \ll P(n,k,\delta)$. For example, $k = n/2$, $\delta = 1/2$ and $n' = k' = n/2$. Note that $P(n,n/2,1/2) \geq 1/2$, but $P(n/2,n/2,1/2) = 2^{-n/2}$. If $n = \omega(\log s)$, then the returned $\mathsf{P}^{n'*}$ has negligible success probability, which is not useful for efficient black-box reduction.

To resolve this issue, Holenstein and Schoenebeck [22] exploited the slackness parameter ξ carefully to keep the success probability noticeable (always at least $\Omega(\xi/n)$) during the correlation reduction procedure. To make this work, it also requires a careful analysis of the reduction. Therefore, they were able to make the reduction run in time $\mathrm{poly}(|x|,n,\xi^{-1})$ and prove tight efficient parallel repetition theorem for all parameter ranges. Furthermore, they extended the above reduction to handle any monotone verifier and obtained a tight monotone repetition theorem.

On the other hand, we overlooked their way of exploiting the slackness parameter, and hence, our implementation of the above reduction ran in time $\mathrm{poly}(|x|,\delta^{-n},(1-\delta)^{-n})$, which is only efficient for constant δ and $n = O(\log s)$. Instead, we exploited the slackness parameter in a different way to obtain parallel repetition theorem for general $n,k = \mathrm{poly}(s)$. Roughly speaking, we applied generic transformations (in the spirit of the one presented in Section 4.1) to convert P^{n*} to some $\mathsf{P}^{n'*}$ with $n' = O(\log s)$ before applying the above reduction. As mentioned, we obtained the same bounds for Chernoff-type region, but slightly worse bounds when the threshold is small. Our result had an additional limitation that it required the initial soundness error δ to be constant.

In the following sections, we formalize the above reduction, and present the better analysis of Holenstein and Schoenebeck [22].

4.2.1 Correlation Reduction for Threshold Verifiers

In this section, we present a formal description of the generalized correlation reduction procedure for threshold verifiers in Figure 4.2, and state its property in Lemma 4.8. Recall that the starting point is a (deterministic) parallel prover P^{n*} for $\mathsf{V}^{n,k}$ with success probability at least $P(n,k,\delta)+\xi$ for some slackness parameter ξ. The procedure is similar to that in Figure 3.5 of Section 3.3, and is informally discussed in the previous section. We note that although the value $P(n,k,\delta)$ can decrease exponentially during the iteration of the procedure, the slackness parameter

remains at least ξ/n throughout the procedure.

Sub-Routine $\mathsf{FindC}(\mathsf{P}^{n*}, n, k, \delta, \xi)$
/* Find correlations in the success pattern of P^{n*} on the first coordinate. */
/* Return $\mathsf{P}^{(n-1)*}$ if such correlation is found. Otherwise, return $\perp$ */

Repeat the following at most $M_1 = O\left(\frac{n}{\xi} \cdot \log \frac{n}{\xi}\right)$ times:

- Sample random coins c_1^* and estimate
  - $p_k(c_1^*) \stackrel{\text{def}}{=} \Pr[\mathsf{P}^{n*} \text{ convinces} \geq k \text{ of } \mathsf{V}_{-1} | c_1 = c_1^*]$, and
  - $p_{k-1}(c_1^*) \stackrel{\text{def}}{=} \Pr[\mathsf{P}^{n*} \text{ convinces} \geq k-1 \text{ of } \mathsf{V}_{-1} | c_1 = c_1^*]$

  by sampling. Namely, randomly sample $M_2 = O\left(\frac{n^2}{\xi^2} \cdot \log \frac{n}{\xi}\right)$ independent copies of $\vec{c}_{-1}$'s, check if P^{n*} convinces $\geq k$ (resp., $k-1$) of V_{-1} on $(c_i^*, \vec{c}_{-i})$, and compute estimators
  - $\hat{p}_k(c_i^*) = |\{\vec{c}_{-i} : \ \mathsf{P}^{n*} \text{ convinces} \geq k \text{ of } \mathsf{V}_{-1} \text{ on } (c_1^*, \vec{c}_{-1})\}|/M_2$.
  - $\hat{p}_{k-1}(c_i^*) = |\{\vec{c}_{-i} : \ \mathsf{P}^{n*} \text{ convinces} \geq k-1 \text{ of } \mathsf{V}_{-1} \text{ on } (c_1^*, \vec{c}_{-1})\}|/M_2$.
- If $\hat{p}_k(c_1^*) \geq P(n-1, k, \delta) + (1 - (1/n)) \cdot \xi$, then return integer k and a parallel prover $\mathsf{P}^{(n-1)*}(c_1^*)$ for $\mathsf{V}^{n-1,k}$ defined as follows: $\mathsf{P}^{(n-1)*}(c_1^*)$ interacts with $\mathsf{V}^{n-1,k}$ by simulating the interaction of P^{n*} and $\mathsf{V}^{n,k}$, where $\mathsf{P}^{(n-1)*}(c_1^*)$ simulates P^{n*} and the first coordinate V_1 with coin c_1^* honestly, and $\mathsf{V}^{n-1,k}$ plays the remaining coordinates V_{-1} of $\mathsf{V}^{n,k}$.
- If $\hat{p}_{k-1}(c_1^*) \geq P(n-1, k-1, \delta) + (1 - (1/n)) \cdot \xi$, then return integer $k-1$ and the same parallel prover $\mathsf{P}^{(n-1)*}(c_1^*)$ as above (but for $\mathsf{V}^{n-1,k-1}$).

Return $\perp$ after M_1 (failure) attempts.

$\mathsf{CR}(\mathsf{P}^{n*}, n, k, \delta, \xi)$
/* Implicitly, there are a PPT verifier V and an input x as part of the input. */
/* Iteratively exploit correlation in the success pattern of P^{n*} to obtain $\mathsf{P}^{(n-1)*}$.*/
Iteratively apply FindC until $n = 1$ or FindC returns $\perp$, namely

- Call $\mathsf{FindC}(\mathsf{P}^{n*}, n, k, \delta, \xi)$. If FindC returns a number k' and $\mathsf{P}^{(n-1)*}$, then set $\xi \leftarrow \left(1 - \frac{1}{n}\right) \cdot \xi$, $n \leftarrow n-1$, and $k \leftarrow k'$ (so that P^{n*} refers to the prover strategy returned by FindC).

Return the final P^{n*} and the corresponding threshold k.

Figure 4.2: Correlation reduction for threshold verifiers.

The property of the generalized correlation reduction procedure is summarized in the following lemma, which informally says that the returned $\mathsf{P}^{n'*}$ (with threshold k') satisfies that either (1) $n' = 1$ and $\mathsf{P}^{n'*}$ convinces V with probability at least δ, or (2) $\mathsf{P}^{n'*}$ convinces $\mathsf{V}^{n',k'}$ with probability at least $P(n', k', \delta) + \Omega(n'\xi/n)$, and for most coins c_1^*, $\Pr[\mathsf{P}^{n'*}$ convinces $\geq k'$ of $\mathsf{V}_{-1}|c_1 = c_1^*] \leq P(n'-1, k', \delta)+O(n'\xi/n)$ and $\Pr[\mathsf{P}^{n'*}$ convinces $\geq k'-1$ of $\mathsf{V}_{-1}|c_1 = c_1^*] \leq P(n'-1, k'-1, \delta) + O(n'\xi/n)$.

Consider a PPT verifier V (not necessarily three-message), an input $x \in \{0,1\}^*$, parameters $n, k \in N$, $\delta, \xi \in (0,1)$, and a deterministic parallel prover P^{n*} for $\mathsf{V}^{n,k}$ be given as in Figure 4.2.

Lemma 4.8 *If P^{n*} has success probability at least $(P(n,k,\delta)+\xi)$ in convincing $\mathsf{V}^{n,k}$ on input x, then with probability at least $(1-(\xi/10n))$ over the randomness of CR, $\mathsf{CR}(\mathsf{P}^{n*}, n, k, \delta, \xi)$ outputs a deterministic prover strategy $\mathsf{P}^{n'*}$ with threshold k' satisfying the following properties.*

- $\Pr[\langle \mathsf{P}^{n'*}, \mathsf{V}^{n',k'}\rangle(x) = 1] \geq P(n', k', \delta) + ((10n'-1)/10n)\cdot\xi$.
- *Either $n' = 1$, or with probability at least $(1-(\xi/10n))$ over V_1's coins c_1^*,*
  - $\Pr[\mathsf{P}^{n'*}$ *convinces* $\geq k'$ *of* $\mathsf{V}_{-1}|c_1 = c_1^*] \leq P(n'-1, k', \delta) + \frac{10(n'-1)+1}{10n}\cdot\xi$, *and*
  - $\Pr[\mathsf{P}^{n'*}$ *convinces* $\geq k'-1$ *of* $\mathsf{V}_{-1}|c_1 = c_1^*] \leq P(n'-1, k'-1, \delta) + \frac{10(n'-1)+1}{10n}\cdot\xi$.

Furthermore, $\mathsf{CR}(\mathsf{P}^{n*}, n, k, \delta, \xi)$ can be implemented with oracle access to P^{n*} with runtime $\mathrm{poly}(|x|, n, \xi^{-1})$, and the output $\mathsf{P}^{n'*}$ can be implemented in time $\mathrm{poly}(|x|, n)$ given oracle access to P^{n*}.

Lemma 4.8 can be proved by a straightforward generalization of the proof of Lemma 3.13. We omit the proof to avoid repetitive arguments.

4.2.2 Reduction Prover Strategy P^*

In this section, we present a formal description of our reduction prover strategy P^* for proving Theorem 4.7 in Figure 4.3. As discussed, P^* first applies the correlation reduction procedure CR to the given (deterministic) parallel prover P^{n*} to obtain a $\mathsf{P}^{n'*}$ with extra properties stated in Lemma 4.8. Then P^* applies a variant of the naive strategy to $\mathsf{P}^{n'*}$, namely, P^* embeds V in the first coordinate of $\mathsf{V}^{n',k'}$, and uses sampling to select a $\vec{c}_{-1}$ such that $\mathsf{P}^{n'*}$ convinces exactly $k'-1$ of V_{-1}.

We proceed to analyze the success probability of P^*. As before, it is instructive to analyze the reduction in the ideal case where there are no sampling errors. Specifically, we consider an ideal scenario (where there are no sampling errors in both the CR procedure and sampling $\vec{c}_{-1}$) that satisfies the following properties:

Prover Strategy $\mathsf{P}^*(x, n, k, \delta, \xi)$
/* P^* interacts with V and is given oracle access to P^{n*}. */

- P^* applies CR to P^{n*} to obtain $\mathsf{P}^{n'*} = \mathsf{CR}(\mathsf{P}^{n*}, n, k, \delta, \xi)$, and a threshold k'.
- If $n' = 1$, P^* interacts with V by running $\mathsf{P}^{n'*}$; otherwise, P^* does the following.
- P^* runs $\mathsf{P}^{n'*}$ to generate $\vec{w}$, and sends w_1 to V.
- Upon receiving $v_1 = v$ from V, P^* repeats the following at most $M = O\left(\frac{n}{\xi} \cdot \log \frac{n}{\xi}\right)$ times.
  - P^* generates random coins $\vec{c}_{-1} = (c_2, \ldots, c_{n'})$. Then P^* checks if $\mathsf{P}^{n'*}(v_1, \vec{c}_{-1})$ convinces exactly $k' - 1$ of V_{-1}. Namely, P^* computes $\vec{v}_{-1} = \mathsf{V}_{-1}(\vec{c}_{-1})$ and $\vec{p} = \mathsf{P}^{n'*}(v_1, \vec{v}_{-1})$, and checks if V_j with coin c_j accepts transcript (w_j, v_j, p_j) for exactly $k' - 1$ of $j \in \{2, \ldots, n\}$. If so, P^* sends p_1 to V and terminates.
- If all the M attempts fail, P^* sends $\perp$ to V (or simply abort).

Figure 4.3: Reduction prover strategy P^* for three-message protocols with threshold verifiers.

- The $\mathsf{P}^{n'*}$ and threshold k' returned by CR (assuming that $n' > 1$ as the $n' = 1$ case is trivial) always satisfy the following properties:
  - $\Pr[\langle \mathsf{P}^{n'*}, \mathsf{V}^{n',k'} \rangle(x) = 1] \geq P(n', k', \delta) + \frac{n'}{n} \cdot \xi$.
  - For every c_1^*,

$$\Pr[\mathsf{P}^{n'*} \text{ convinces} \geq k' \text{ of } \mathsf{V}_{-1} | c_1 = c_1^*] \leq P(n' - 1, k', \delta) + \frac{n' - 1}{n} \cdot \xi,$$

$$\Pr[\mathsf{P}^{n'*} \text{ convinces} \geq k' - 1 \text{ of } \mathsf{V}_{-1} | c_1 = c_1^*] \leq P(n' - 1, k' - 1, \delta) + \frac{n' - 1}{n} \cdot \xi.$$

- If there exist $\vec{c}_{-1}$ such that $\mathsf{P}^{n'*}$ convinces exactly $k' - 1$ of V^{-1} on $(v_1, \vec{c}_{-1})$, then P^* selects a uniformly random such $\vec{c}_{-1}$. (This is achieved by randomly sampling a unbounded number of $\vec{c}_{-1}$ as opposed to at most M times in Figure 4.3.)

We argue that (in this ideal scenario) the success probability of P^* can be expressed in the following formula:

$$\Pr[\langle \mathsf{P}^*, \mathsf{V} \rangle(x) = 1] = \mathop{\mathrm{E}}_{c_1^*} \left[\frac{\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}_1 \wedge (k' - 1 \text{ of } \mathsf{V}_{-1}) | c_1 = c_1^*]}{\Pr[\mathsf{P}^{n'*} \text{ convinces } k' - 1 \text{ of } \mathsf{V}_{-1} | c_1 = c_1^*]} \right].^4$$

[4] Here, we take a convention that $0/0 = 0$.

First, the expectation operator corresponds to that V uses uniformly random coins c_1^*. Second, note that P^* samples a random $\vec{c}_{-1}$ conditioning on $\mathsf{P}^{n'*}$ convinces exactly $k'-1$ of V_{-1} on $(c_1^*, \vec{c}_{-1})$, and P^* succeeds iff $\mathsf{P}^{n'*}$ also convinces V_1 on $(c_1^*, \vec{c}_{-1})$. Conditioned on V's coin being c_1^*, the success probability of P^* is precisely

$$\frac{\Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}_1 \wedge (k'-1 \text{ of } \mathsf{V}_{-1}) | c_1 = c_1^*]}{\Pr[\mathsf{P}^{n'*} \text{ convinces } k'-1 \text{ of } \mathsf{V}_{-1} | c_1 = c_1^*]}.$$

We will use the above properties of $\mathsf{P}^{n'*}$ to lower bound the above expectation. To simplify the notation, we first introduce the following shorthand notations.

- We use $(\geq k')$ to denote the event that $\mathsf{P}^{n'*}$ convinces at least k' of $\mathsf{V}^{n',k'}$, and

$$p(\geq k') \stackrel{\text{def}}{=} \Pr[\mathsf{P}^{n'*} \text{ convinces} \geq k' \text{ of } \mathsf{V}^{n',k'}],$$

$$p(\geq k' | c_1^*) \stackrel{\text{def}}{=} \Pr[\mathsf{P}^{n'*} \text{ convinces} \geq k' \text{ of } \mathsf{V}^{n',k'} | c_1 = c_1^*].$$

- We use $(1, k'-1)$ to denote the event that $\mathsf{P}^{n'*}$ convinces V_1 and exactly $k'-1$ of V_{-1}, and $(*, \geq k')$ to denote the event that $\mathsf{P}^{n'*}$ convinces at least k' of V_{-1} (i.e., ignore V_1's verdict). Similarly, we define

$$p(1, k'-1) \stackrel{\text{def}}{=} \Pr[\mathsf{P}^{n'*} \text{ convinces } \mathsf{V}_1 \wedge (k'-1 \text{ of } \mathsf{V}_{-1})],$$

$$p(*, \geq k' | c_1^*) \stackrel{\text{def}}{=} \Pr[\mathsf{P}^{n'*} \text{ convinces} \geq k' \text{ of } \mathsf{V}_{-1} | c_1 = c_1^*].$$

We perform the following calculation similar to that in previous section to lower bound the expectation.

$$\begin{aligned}
&\mathop{\mathrm{E}}_{c_1^*}\left[\frac{p(1, k'-1|c_1^*)}{p(*, k'-1|c_1^*)}\right] \\
&= \mathop{\mathrm{E}}_{c_1^*}\left[\frac{p(\geq k'|c_1^*) - p(*, \geq k'|c_1^*)}{p(*, \geq k'-1|c_1^*) - p(*, \geq k'|c_1^*)}\right] \\
&\geq \mathop{\mathrm{E}}_{c_1^*}\left[\frac{p(\geq k'|c_1^*) - \left(P(n'-1, k', \delta) + \frac{n'-1}{n}\cdot\xi\right)}{\left(P(n'-1, k'-1, \delta) + \frac{n'-1}{n}\cdot\xi\right) - \left(P(n'-1, k', \delta) + \frac{n'-1}{n}\cdot\xi\right)}\right] \\
&\geq \frac{\left(P(n', k', \delta) + \frac{n'}{n}\cdot\xi\right) - \left(P(n'-1, k', \delta) + \frac{n'-1}{n}\cdot\xi\right)}{P(n'-1, k'-1, \delta) - P(n'-1, k', \delta)} \\
&= \frac{\Pr[X_1 = 1 \wedge \sum_{i=2}^{n'} X_i = k'-1] + \frac{1}{n}\cdot\xi}{\Pr[\sum_{i=2}^{n'} X_i = k'-1]} \\
&\geq \delta + \frac{\xi}{n}.
\end{aligned}$$

This completes the analysis in the ideal scenario.

We proceed to analyze the actual (non-ideal) prover strategy P^*, where the challenge is to show that the sampling errors do not lower the success probability too much. The sampling errors are handled in a similar way to the analysis for the direct product case in Section 3.3. We shall show that if P^{n*} has success probability at least $P(n,k,\delta)+\xi$ in convincing $\mathsf{V}^{n,k}$, then P^* can succeed with probability at least $\delta+(\xi/10n)$.

Recall that by Lemma 4.8, with probability at least $(1-(\xi/10n))$ over the randomness of CR, the $\mathsf{P}^{n'*}$ returned by CR satisfies two good properties. Let us call a $\mathsf{P}^{n'*}$ returned by CR *good* if $\mathsf{P}^{n'}$ satisfies the two properties stated in the lemma. Informally, Lemma 4.8 allows us to focus on good $\mathsf{P}^{n'*}$'s with a loss of of at most $(\xi/10n)$ on the success probability, since $\mathsf{P}^{n'*}$ is not good with probability at most $(\xi/10n)$. Also observe that by definition, when $\mathsf{P}^{n'*}$ is good and $n'=1$, P^* can succeed with probability at least $\delta+(9/10n)\cdot\xi$. Therefore, it remains to analyze the success probability of P^* for the case that the $\mathsf{P}^{n'*}$ returned by CR is good and $n'>1$.

Fix a good $\mathsf{P}^{n'*}$ with threshold k' returned by CR with $n'>1$. Lemma 4.8 says that $\mathsf{P}^{n'*}$ satisfies

- $p(\geq k') = \mathrm{E}_{c_1^*}[p(\geq k'|c_1^*)] \geq P(n',k',\delta)+((10n'-1)/10n)\cdot\xi$.
- With probability at least $(1-(\xi/10n))$ over c_1^*,
  - $p(*,\geq k'|c_1^*) \leq P(n'-1,k',\delta)+\frac{10(n'-1)+1}{10n}\cdot\xi$, and
  - $p(*,\geq k'-1|c_1^*) \leq P(n'-1,k'-1,\delta)+\frac{10(n'-1)+1}{10n}\cdot\xi$.

Let us call c_1^* is good and denote it by $c_1^*\in\mathsf{Good}$ if the above inequalities holds. In other words, we define

$$\mathsf{Good}=\{c_1^*:\ c_1^* \text{ satisfies the above inequalities.}\}.$$

By a similar argument as that in the analysis of the ideal scenario, we observe that the success probability of P^* can be expressed as

$$\Pr[\langle\mathsf{P}^*,\mathsf{V}\rangle(x)=1|\mathsf{P}^{n'}] = \mathop{\mathrm{E}}_{c_1^*}\left[\frac{p(1,k'-1|c_1^*)}{p(*,k'-1|c_1^*)}\cdot\left(1-(1-p(1,k'-1|c_1^*))^M\right)\right],$$

where $\left(1-(1-p(1,k'-1|c_1^*))^M\right)$ is the probability that P^* can find a $\vec{c}_{-1}$ such that $\mathsf{P}^{n'*}$ convinces exactly $k'-1$ of V_{-1} on $(c_1^*,\vec{c}_{-1})$ from at most M random samples of $\vec{c}_{-1}$. Our goal is to lower bound the expectation.

We recall Claim 3.14 in Section 3.3 to handle the extra error term $\left(1-(1-p(1,k'-1|c_1^*))^M\right)$.

Claim 4.9 *Let $\gamma\in(0,1)$. If $M\in\mathbb{N}$ satisfies $(1-\gamma)^M\leq\gamma$, then for every α,β with $0\leq\alpha\leq\beta\leq1$, we have*

$$\frac{\alpha}{\beta}\cdot\left(1-(1-\beta)^M\right)\geq\frac{(\alpha-\gamma)_+}{\beta},$$

where $(x)_+ \stackrel{\text{def}}{=} \max\{x, 0\}$.[5]

Choosing the constant in $M = O\left(\frac{n}{\xi}\log\frac{n}{\xi}\right)$ properly so that $(1-(\xi/10n))^M \leq (\xi/10n)$, and using the above claim, we have

$$\mathop{\mathrm{E}}_{c_1^*}\left[\frac{p(1,k'-1|c_1^*)}{p(*,k'-1|c_1^*)}\cdot\left(1-(1-p(*,k'-1|c_1^*))^M\right)\right] \geq \mathop{\mathrm{E}}_{c_1^*}\left[\frac{(p(1,k'-1|c_1^*)-(\xi/10n))_+}{p(*,k'-1|c_1^*)}\right].$$

Also, we can get rid of bad c_1^* by

$$\mathop{\mathrm{E}}_{c_1^*}\left[\frac{(p(1,k'-1|c_1^*)-(\xi/10n))_+}{p(*,k'-1|c_1^*)}\right] \geq \mathop{\mathrm{E}}_{c_1^*}\left[\frac{(p(1,k'-1|c_1^*)-(\xi/10n))_+\cdot\mathbf{1}[c_1^*\in\mathsf{Good}]}{p(*,k'-1|c_1^*)}\right],$$

where $\mathbf{1}[\mathcal{E}] = 1$ if the event $\mathcal{E}$ is true, and 0 otherwise. We can then perform a similar calculation as in the ideal scenario:

$$\begin{aligned}
&\mathop{\mathrm{E}}_{c_1^*}\left[\frac{(p(1,k'-1|c_1^*)-(\xi/10n))_+\cdot\mathbf{1}[c_1^*\in\mathsf{Good}]}{p(*,k'-1|c_1^*)}\right]\\
&= \mathop{\mathrm{E}}_{c_1^*}\left[\frac{(p(\geq k'|c_1^*)-p(*,\geq k'|c_1^*)-(\xi/10n))_+\cdot\mathbf{1}[c_1^*\in\mathsf{Good}]}{p(*,\geq k'-1|c_1^*)-p(*,\geq k'|c_1^*)}\right]\\
&\geq \mathop{\mathrm{E}}_{c_1^*}\left[\frac{\left(p(\geq k'|c_1^*)-\left(P(n'-1,k',\delta)+\frac{10(n'-1)+1}{10n}\cdot\xi\right)-(\xi/10n)\right)_+\cdot\mathbf{1}[c_1^*\in\mathsf{Good}]}{\left(P(n'-1,k'-1,\delta)+\frac{10(n'-1)+1}{10n}\cdot\xi\right)-\left(P(n'-1,k',\delta)+\frac{10(n'-1)+1}{10n}\cdot\xi\right)}\right]\\
&= \frac{\mathrm{E}_{c_1^*}\left[\left(p(\geq k'|c_1^*)-P(n'-1,k',\delta)-\frac{10(n'-1)+2}{10n}\cdot\xi\right)_+\cdot\mathbf{1}[c_1^*\in\mathsf{Good}]\right]}{\Pr\left[\sum_{i=2}^{n'}X_i = k'-1\right]},
\end{aligned}$$

where we can bound the numerator by

$$\begin{aligned}
&\mathop{\mathrm{E}}_{c_1^*}\left[\left(p(\geq k'|c_1^*)-P(n'-1,k',\delta)-\frac{10(n'-1)+2}{10n}\cdot\xi\right)_+\cdot\mathbf{1}[c_1^*\in\mathsf{Good}]\right]\\
&\geq \mathop{\mathrm{E}}_{c_1^*}[p(\geq k'|c_1^*)\cdot\mathbf{1}[c_1^*\in\mathsf{Good}]]-\left(P(n'-1,k',\delta)+\frac{10(n'-1)+2}{10n}\cdot\xi\right)\\
&\geq \mathop{\mathrm{E}}_{c_1^*}[p(\geq k'|c_1^*)]-\Pr[c_1^*\notin\mathsf{Good}]-\left(P(n'-1,k',\delta)+\frac{10(n'-1)+2}{10n}\cdot\xi\right)\\
&\geq \left(P(n',k',\delta)+\frac{10n'-1}{10n}\cdot\xi\right)-\frac{\xi}{10n}-\left(P(n'-1,k',\delta)+\frac{10(n'-1)+2}{10n}\cdot\xi\right)\\
&= \Pr\left[X_1=1\wedge\sum_{i=2}^{n'}X_i=k'-1\right]+\frac{10n'-4}{10n}\cdot\xi.
\end{aligned}$$

[5] Again, we use a convention that $0/0 \stackrel{\text{def}}{=} 0$.

Putting things together, for every good $\mathsf{P}^{n'*}$ with threshold k' returned by CR with $n' > 1$, we have

$$\Pr[\langle \mathsf{P}^*, \mathsf{V}\rangle(x) = 1 | \mathsf{P}^{n'*}] \geq \frac{\Pr\left[X_1 = 1 \wedge \sum_{i=2}^{n'} X_i = k' - 1\right] + \frac{10n'-4}{10n} \cdot \xi}{\Pr\left[\sum_{i=2}^{n'} X_i = k' - 1\right]} \geq \delta + \frac{2\xi}{10n}.$$

It follows that

$$\begin{aligned}
&\Pr[\langle \mathsf{P}^*, \mathsf{V}\rangle(x) = 1] \\
&\quad \geq \Pr[\langle \mathsf{P}^*, \mathsf{V}\rangle(x) = 1 | \mathsf{P}^{n'*} \text{ is good }] - \Pr[\mathsf{P}^{n'*} \text{ is not good }] \\
&\quad \geq \delta + \frac{2\xi}{10n} - \frac{\xi}{10n} \\
&\quad \geq \delta + \frac{\xi}{10n},
\end{aligned}$$

which completes the analysis.

4.2.3 Discussion

For the sake of completeness, we formally state the threshold repetition theorem we obtained and the more general parallel repetition theorem with monotone verifiers of Holenstein and Schoenebeck [22], and briefly discuss how our original analysis is different. We first state the result of Holenstein and Schoenebeck [22].

Theorem 4.10 ([22]) *Let $\mathsf{V} \in$ PPT be a three-message verifier. There exists a prover strategy P^* such that for every common input $x \in \{0,1\}^*$, every $n \in \mathbb{N}$, $\delta, \xi \in (0,1)$, every efficient combining function $g : \{0,1\}^n \to \{0,1\}$ (given as a circuit), and parallel prover strategy P^{n*}, let $\varepsilon \stackrel{\text{def}}{=} \Pr[g(X_1, \ldots, X_n) = 1]$ where $X_1, \ldots, X_n$ are i.i.d. random bits with $\Pr[X_i = 1] = \delta$, we have*

1. $\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n}\rangle(x) = 1] \geq \varepsilon + \xi \Rightarrow$

$$\Pr[\langle \mathsf{P}^{*(\mathsf{P}^{n*})}(n, g, \delta, \xi), \mathsf{V}\rangle(x) = 1] \geq \delta + \frac{\xi}{10n},$$

2. $\mathsf{P}^{*(\cdot)}(x, n, \delta, \xi)$ *runs in time* $\text{poly}(|x|, n, \xi^{-1})$ *given oracle access to* $\mathsf{P}^{n*}(x)$.

As mentioned earlier in this section, both we and Holenstein and Schoenebeck [22] consider the same reduction idea, but our implementation is suboptimal so that the runtime of our reduction is $\text{poly}(|x|, \delta^{-n}, (1-\delta)^{-n})$ instead of $\text{poly}(|x|, n, \xi^{-1})$. Hence, our reduction is efficiently only when $n = O(\log s)$. Nevertheless, we can still prove the following threshold repetition theorems for general $n, k = \text{poly}(s)$.

Theorem 4.11 *Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a three-message protocol with input domain Λ, let $\delta, \alpha \in (0,1)$ be constants, and let $n, k : \mathbb{N} \to \mathbb{N}$ be efficiently computable functions with $1 \leq k \leq n \leq \mathrm{poly}(s)$. If $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ,*

1. *then $\langle \mathsf{P}^n, \mathsf{V}^{n,k} \rangle$ has soundness error $P(n, k, (1+\alpha)\delta) + \mathsf{ngl}$.*
2. *if in addition, $k \geq (1+\gamma)\delta n$ for some constant $\gamma > 0$, then $\langle \mathsf{P}^n, \mathsf{V}^{n,k} \rangle$ has soundness error $P(n, k, \delta) + \mathsf{ngl}$.*

We briefly sketch our idea of proving the above theorem, where we omit various sampling errors and rounding issues for simplicity. Our idea is to apply the following simple transformation to reduce the number of repetition n to $O(\log s)$ before applying our reduction. Given a parallel prover strategy P^{n*} for $\mathsf{V}^{n,k}$ with success probability ε, our transformation outputs a parallel prover strategy $\mathsf{P}^{n'*}$ for $\mathsf{V}^{n',k'}$ with $k' = k \cdot (n'/n)$ and $\varepsilon' = \varepsilon \cdot (n'/n)$.

The $\mathsf{P}^{n'*}$ is defined as follows. $\mathsf{P}^{n'*}$ interacts with $\mathsf{V}^{n',k'}$ by simulating the interaction of P^{n*} and $\mathsf{V}^{n,k}$. $\mathsf{P}^{n'*}$ partitions the n coordinates of $\mathsf{V}^{n,k}$ into $t = (n/n')$ (disjoint) blocks, embeds $\mathsf{V}^{n',k'}$ in a random block of $\mathsf{V}^{n,k}$, and then simply simulates P^{n*} and the remaining $t-1$ blocks of $\mathsf{V}^{n,k}$ honestly. Note that when P^{n*} convinces $\mathsf{V}^{n,k}$, by an averaging argument, there must be a block $i \in [t]$ such that P^{n*} convinces at least k' out of n' subverifiers in this block. $\mathsf{P}^{n'*}$ can guess this block correctly with probability $1/t$, and hence $\mathsf{P}^{n'*}$ can successfully convince $\mathsf{V}^{n',k'}$ with probability at least $\varepsilon' = \varepsilon/t$.

At the first glance, the loss of factor t in the success probability is large so that when we apply our reduction to $\mathsf{P}^{n'*}$, we may not be able to obtain a good bound. Fortunately, note that we only need to apply the transformation when $n = \omega(\log s)$, and thanks to the necessary negligible slackness presented in Theorem 4.11, the resulting reduction prover strategy can indeed succeed with desired probability.

To see this, let us consider the Chernoff-type case where the threshold $k \geq (1+\gamma)\delta n$ for some constant $\gamma > 0$. Recall that the starting point of the reduction is a P^{n*} with success probability $\varepsilon = P(n, k, \delta) + \xi$ for some noticeable slackness ξ. Note that when $n = \omega(\log s)$, the probability $P(n, k, \delta) = e^{-\Omega(\gamma^2 \delta n)} = s^{-\omega(1)}$ is negligible. Hence, ξ is dominant term in $\varepsilon = P(n, k, \delta) + \xi$. By choosing $n' = c \log s$ for sufficiently large constant c, we can have $\varepsilon' = \varepsilon/t \geq P(n', k', \delta)$. It follows that when we apply our reduction to $\mathsf{P}^{n'*}$, we can obtain a prover strategy with success probability at least δ. A similar argument can be applied to the general case to prove the first bullet of the above Theorem 4.11.

4.3 Efficient Parallel Repetition Theorem for Constant-round Public-Coin Protocols

In this section, we prove a tight efficient parallel repetition theorem for *constant-round* public-coin protocols with arbitrary monotone verifiers, which says that sound-

ness error behaves as if the different executions were completely independent. More precisely, if a constant-round public-coin protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ has soundness error δ, then the parallel protocol $\langle \mathsf{P}^n, \mathsf{V}^{n,g} \rangle$ has soundness error $\Pr[g(X_1, \ldots, X_n) = 1] + \mathsf{ngl}$, where g is any monotone combining function, and $X_1, \ldots, X_n$ are i.i.d. random bits with $\Pr[X_i = 1] = \delta$.

We prove it by analyzing a "recursive sampling" strategy of Pass and Venkitasubramaniam [30], who used the reduction to prove a tight direct product theorem for constant-round public-coin protocols. We show that the same reduction actually gives more general parallel repetition theorems. Namely, if there exists a P^{n*} with success probability at least $\Pr[g(X_1, \ldots, X_n) = 1]$ in convincing $\mathsf{V}^{n,g}$, then the reduction prover strategy P^* can succeed with probability at least roughly δ. This reduction is very different from the reduction in Section 3.2 for general public-coin protocols.

Formally, we prove the following theorem. Note that the reduction runs in time exponential in the number of rounds m, and hence only gives efficient parallel repetition theorem when the number of rounds is constant.

Theorem 4.12 *Let $\mathsf{V} \in \mathrm{PPT}$ be a m-round public-coin verifier. There exists a prover strategy P^* such that for every common input $x \in \{0,1\}^*$, every $n \in \mathbb{N}$, $\delta, \xi \in (0,1)$, every efficient combining function $g : \{0,1\}^n \to \{0,1\}$, and parallel prover strategy P^{n*}, if we let $\varepsilon \stackrel{\text{def}}{=} \Pr[g(X_1, \ldots, X_n) = 1]$ where $X_1, \ldots, X_n$ are i.i.d. random bits with $\Pr[X_i = 1] = \delta$, we have*

1. $\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) = 1] \geq \varepsilon + \xi$

$$\Rightarrow \Pr[\langle \mathsf{P}^{*(\mathsf{P}^{n*})}(n, g, \delta, \xi), \mathsf{V} \rangle(x) = 1] \geq \delta,$$

2. $\mathsf{P}^{*(\cdot)}(x, n, \delta, \xi)$ *runs in time* $\mathrm{poly}(|x|, 2^{m^2}, n^m, \xi^{-m})$ *given oracle access to* $\mathsf{P}^{n*}(x)$.

We first recall the notation for public-coin protocols, and then discuss the reduction prover strategy for proving the above theorem. Recall that we assume the verifier speaks first, and we denote the verifier V's (resp., the prover P's) messages by $v_1, \ldots, v_m$ (resp., $p_1, \ldots, p_m$). The messages of the n-fold parallel repetition $\langle \mathsf{P}^n, \mathsf{V}^{n,g} \rangle$ of $\langle \mathsf{P}, \mathsf{V} \rangle$ are denoted by $\vec{v}_1 = (v_{1,1}, \ldots, v_{1,n}), \vec{v}_2, \ldots, \vec{v}_m$, and $\vec{p}_1, \ldots, \vec{p}_m$, respectively. We use $d_1, \ldots, d_n$ to denote the verdict bits of $\mathsf{V}^{n,g}$, and $d = g(d_1, \ldots, d_n)$ is the verdict of $\mathsf{V}^{n,g}$. Throughout this section, $\vec{X} = (X_1, \ldots, X_n)$ are i.i.d. random bits with $\Pr[X_i = 1] = \delta$, and $\varepsilon \stackrel{\text{def}}{=} \Pr[g(\vec{X}) = 1] = \Pr[g(X_1, \ldots, X_n) = 1]$.

Similarly to Lemma 2.5 and 2.6, we assume without loss of generality that P^{n*} is *deterministic*, and hence the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,g} \rangle(x)$ is determined solely by $\mathsf{V}^{n,g}$'s messages $(\vec{v}_1, \ldots, \vec{v}_m)$. For convenience, we ignore P^{n*}'s messages and refer to $(\vec{v}_1, \ldots, \vec{v}_m)$ as the transcript of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,g} \rangle(x)$. We also write $\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,n} \rangle(x) =$

$1|\vec{v}_1, \ldots, \vec{v}_j]$ as the success probability of P^{n*} conditioned on the partial transcript being $(\vec{v}_1, \ldots, \vec{v}_j)$. Similarly, $\Pr[\mathsf{P}^{n*}$ convinces $\mathsf{V}_i|\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i}]$ is the probability that P^{n*} convinces the i-th subverifier V_i conditioned on partial transcript $(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})$.

We proceed to discuss the recursive sampling strategy of Pass and Venkitasubramaniam [30]. Recall the common framework that P^* interacts with V by simulating the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,g} \rangle$, where P^* embeds $\mathsf{V} = \mathsf{V}_i$ in some coordinate $i \in [n]$ of $\mathsf{V}^{n,g}$, and they jointly select the parallel verifier $\mathsf{V}^{n,g}$'s messages $\vec{v}_1, \ldots, \vec{v}_m$. This can be viewed as a game played between P^* and V, where P^*'s moves are $(i, \vec{v}_{1,-i}, \vec{v}_{2,-i}, \ldots, \vec{v}_{m,-i})$ and V's moves are $(v_{1,i}, v_{2,i}, \ldots, v_{m,i})$. In the game, V plays uniformly random strategy, and the goal is to design strategies for P^* to achieve a good success probability.

Pass and Venkitasubramaniam [30] considered an optimal strategy P^*_{opt}, where at each round, P^*_{opt} selects an optimal move (either $i \in [n]$ or $\vec{v}_{j,-i}$) that maximizes his success probability in the future. They proved that, for the direct product case, if P^{n*} has success probability δ^n, then P^*_{opt} can succeed with probability at least δ. However, the issue is that the optimal strategy P^*_{opt} cannot be implemented efficiently.

They further observed that, although P^*_{opt} cannot be implemented efficiently, P^*_{opt} can be "approximated" by an efficient recursive sampling strategy P^*_{rec}, which uses sampling to find approximately optimal moves. The name "recursive sampling" comes from that, to approximate the optimal strategy, P^*_{rec} needs to estimate *his own* success probability after taking a certain move, which requires P^*_{rec} to simulate himself *recursively*. This is the reason that P^*_{rec} has runtime exponential in the number of rounds m, and hence only gives efficient parallel repetition theorems for constant-round protocols. They proved that P^*_{rec} can also succeed with probability at least roughly δ, which gives a tight direct product theorem.[6]

We observe that the recursive sampling strategy, while being only efficient for constant-round protocols, gives tight parallel repetition theorems for the most general monotone verifiers. This contrasts to the case of general (super-constant-round) protocols, where we only know how to prove direct product and Chernoff-type theorem. We extends the above two steps to prove the more general parallel repetition theorem. The second step is the same, where the same analysis of Pass and Venkitasubramaniam [30] relates lower bounds on success probability of P^*_{rec} to that of P^*_{opt}.

Our contribution is in the first step, where we lower bound the success probability of the optimal strategy P^*_{opt} when given a parallel prover strategy P^{n*} for a general verifier $\mathsf{V}^{n,g}$ with some monotone combining function g. For the direct product case, Pass and Venkitasubramaniam [30] lower bound the success probability of P^*_{opt} by induction with the same induction hypothesis as our analysis of rejection sampling

[6] As a technical remark, we emphasize that the success probability of P^*_{opt} can be much higher than that of P^*_{rec}. What Pass and Venkitasubramaniam [30] proved actually relates the lower bounds on the success probabilities of P^*_{rec} and P^*_{opt}.

strategy in Section 3.2. Namely, the induction hypothesis says that for every partial interaction $(\vec{v}_1, \ldots, \vec{v}_j)$,

$$\prod_{i=1}^{n} \Pr[\mathsf{P}^*_{opt} \text{ convinces } \mathsf{V}|i, \vec{v}_1, \ldots, \vec{v}_j] \geq \Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}^{n,g}|\vec{v}_1, \ldots, \vec{v}_j],$$

where $\Pr[\mathsf{P}^*_{opt} \text{ convinces } \mathsf{V}|i, \vec{v}_1, \ldots, \vec{v}_j]$ denotes the success probability of P^*_{opt} conditioned on P^*_{opt} plays moves $(i, \vec{v}_{1,-i}, \ldots, \vec{v}_{j,-i})$ and V plays moves $(v_{1,i}, \ldots, v_{j,i})$. It is unclear how to modify the induction hypothesis to analyze the success probability of P^*_{opt} when given P^{n*} for a general $\mathsf{V}^{n,g}$.

We instead use a *coupling argument* to analyze the general case. Let $(D_1, \ldots, D_n)$ be indicator random variables of the verdict bits of the n subverifiers in $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,g} \rangle$. We define indicator random variables $(R_1, \ldots, R_n)$ coupled with $(D_1, \ldots, D_n)$ such that

1. $R_i \geq D_i$ for every $i \in [n]$ with probability 1, and
2. R_i's are *mutually independent* with $\Pr[R_i = 1] = \Pr[\mathsf{P}^*_{opt} \text{ convinces } \mathsf{V}|i]$.

Note that the success probability of P^*_{opt} is $\max_i\{\Pr[\mathsf{P}^*_{opt} \text{ convinces } \mathsf{V}|i]\}$. Suppose $\Pr[\mathsf{P}^{n*} \text{ convinces } \mathsf{V}^{n,g}] \geq \varepsilon$, then we have

$$\Pr[g(\vec{R}) = 1] \geq \Pr[g(\vec{D}) = 1] \geq \varepsilon = \Pr[g(\vec{X}) = 1],$$

where the first inequality follows by the monotonicity of g. Furthermore, the probability $\Pr[g(\vec{R}) = 1]$ is a monotone increasing function of the probabilities $\Pr[R_i = 1]$. This implies that P^*_{opt} can succeed with probability at least

$$\max_i\{\Pr[R_i = 1]\} \geq \Pr[X_i = 1] = \delta.$$

We shall present and analyze the optimal strategy P^*_{opt} formally in the next section, and then present the recursive sampling strategy P^*_{rec} in Section 4.3.2.

4.3.1 Optimal Prover Strategies P^*_{opt}

In this section, we formally define and analyze the optimal strategy P^*_{opt}. Recall that we view the interaction of P^* and V as a game, where P^* first takes a move $i \in [n]$, and then V and P^* take moves $v_{j,i}$ and $\vec{v}_{j,-i}$ alternately for $j = 1, \ldots, m$. At each round of the game, V simply plays a uniformly random move, and the optimal strategy P^*_{opt} selects an optimal move $i \in [n]$ or $\vec{v}_{j,-i}$ that maximizes his success probability in the future.

To formally define the optimal strategy P^*_{opt}, we define functions $\gamma_i(\cdot)$, which correspond to the success probability of P^*_{opt} conditioned on partial interactions of $\langle \mathsf{P}^*_{opt}, \mathsf{V} \rangle$.

Definition 4.13 *Let V be a m-round public-coin verifier, $n \in \mathbb{N}$, P^{n*} a n-fold parallel prover strategy, and $x \in \{0,1\}^*$ an input. We define $[0,1]$-valued functions $\gamma_i(\cdot)$ corresponding to the interaction $\langle \mathsf{P}^{n*}, \mathsf{V}^n \rangle(x)$ inductively as follows.*[7]

First, for every $i \in [n]$ and complete transcript $(\vec{v}_1, \ldots, \vec{v}_m)$, we define

$$\gamma_i(\vec{v}_1, \ldots, \vec{v}_m) \stackrel{\text{def}}{=} \begin{cases} 1 \text{ if } \mathsf{P}^{n*} \text{ convinces } \mathsf{V}_i \text{ on interaction } (\vec{v}_1, \ldots, \vec{v}_m), \\ 0 \text{ otherwise.} \end{cases}$$

Then, for $j = m, m-1, \ldots, 1$, for every $i \in [n]$ and partial transcript $(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})$, we define

$$\begin{aligned} \gamma_i(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i}) &\stackrel{\text{def}}{=} \max_{\vec{v}_{j,-i}} \{\gamma_i(\vec{v}_1, \ldots, \vec{v}_j)\}, \\ \gamma_i(\vec{v}_1, \ldots, \vec{v}_{j-1}) &\stackrel{\text{def}}{=} \mathop{\mathrm{E}}_{v_{j,i}} [\gamma_i(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})]. \end{aligned}$$

Finally, we define

$$\gamma \stackrel{\text{def}}{=} \max_{i \in [n]} \{\gamma_i\}.$$

With the above definition, we give a formal description of P^*_{opt} in Figure 4.4. By construction, it is not hard to verify inductively that for every partial interaction $(i, \bar{h}) = (i, \vec{v}_1, \ldots, \vec{v}_j)$ and $(i, \bar{h}) = (i, \vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})$,

$$\Pr[\mathsf{P}^*_{opt} \text{ convinces } \mathsf{V} | i, \bar{h}] = \gamma_i(\bar{h}).$$

We proceed to analyze the success probability of P^*_{opt}. We shall show that if P^{n*} convinces $\mathsf{V}^{n,g}$ with probability $\varepsilon = \Pr[g(\vec{X}) = 1]$, then P^*_{opt} can convince V with probability at least δ. Formally, we prove the following Lemma.

Lemma 4.14 *Let V be a public-coin verifier. Let $n \in \mathbb{N}, \delta \in (0,1)$ and let $g : \{0,1\}^n \to \{0,1\}$ be a monotone function. Let P^{n*} be a parallel prover strategy and $x \in \{0,1\}^*$ an input such that $\Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,g} \rangle(x) = 1] \geq \varepsilon = \Pr[g(\vec{X}) = 1]$, where $\vec{X} = (X_1, \ldots, X_n)$ are i.i.d. random bits with $\Pr[X_i = 1] = \delta$. Then we have*

$$\Pr[\langle \mathsf{P}^{*(\mathsf{P}^{n*})}_{opt}(n), \mathsf{V} \rangle(x) = 1] \geq \delta.$$

Proof. As outlined in the previous section, we prove the lemma by a coupling argument. Define $\vec{D} = (D_1, \ldots, D_n)$ to be the indicator random variables of the verdict bits of the n subverifiers of $\mathsf{V}^{n,g}$ in the interaction $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,g} \rangle(x)$. Note that the randomness of this probability space is the verifier's messages $\vec{v}_1, \ldots, \vec{v}_m$, which are uniformly random strings. We will construct indicator random variables $\vec{R} = (R_1, \ldots, R_n)$ coupled with $\vec{D}$ that satisfy the following two conditions.

[7] The definition does not depend on the combining function g used by the parallel verifier V^n so we drop the superscript g from the notation.

Prover Strategy $\mathsf{P}^*_{opt}(x, n)$
/* P^*_{opt} interacts with V and is given oracle access to P^{n*} for some $\mathsf{V}^{n,g}$. */
/* The interaction of $\langle \mathsf{P}^*_{opt}, \mathsf{V} \rangle$ simulates the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,g} \rangle$. */

- P^*_{opt} selects a coordinate i such that $\gamma_i = \gamma$.
- For each round $j \in [m]$, upon receiving $v_{j,i} = v_j$ from V, P^*_{opt} selects a $\vec{v}_{j,-i}$ such that

$$\gamma_i(\vec{v}_1, \ldots, \vec{v}_j) = \gamma_i(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i}) = \max_{\vec{v}'_{j,i}} \{\gamma_i(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i}, \vec{v}'_{j,-i})\}$$

  as his move in this round, and runs P^{n*} on $(v_{j,i}, \vec{v}_{j,-i})$ to generate $\vec{p}_j$ and sends $p_{j,i}$ to V.

Figure 4.4: The optimal prover strategy P^*_{opt} for public-coin protocols.

1. $R_i \geq D_i$ for every $i \in [n]$ with probability 1, and
2. R_i's are *mutually independent* with $\Pr[R_i = 1] = \Pr[\mathsf{P}^*_{opt} \text{ convinces } \mathsf{V}|i] = \gamma_i$.

We will construct the desired random variables $\vec{R} = (R_1, \ldots, R_n)$ inductively. We start by defining $\vec{R}_m = \vec{D}$, and inductively construct $\vec{R}_j = (R_{j,1}, \ldots, R_{j,n})$ for $j = m-1, \ldots, 0$ in the same probability space that satisfies the following two invariant conditions.

1. $R_{j,i} \geq D_i$ for every $i \in [n]$ with probability 1.
2. For every partial transcript $\bar{h} = (\vec{v}_1, \ldots, \vec{v}_j)$, the conditional random variables $(R_{j,1}, \ldots, R_{j,n})|_{\bar{h}}$ are mutually independent with $\Pr[R_{j,i} = 1] = \gamma_i(\bar{h})$.

Note that the final $\vec{R}_0 = (R_{0,1}, \ldots, R_{0,n})$ are the desired random variables. Also, it is easy to verify that the above two invariant conditions hold for the base case $j = m$ trivially, as $\vec{R}_m = \vec{D}$ and there is no randomness after conditioning on the complete transcript $\vec{v}_1, \ldots, \vec{v}_m$.

We proceed to construct $\vec{R}_{j-1}$ from $\vec{R}_j$. We define $\vec{R}_{j-1}$ by defining its conditional distribution $\vec{R}_{j-1}|_{\vec{v}_1,\ldots,\vec{v}_j}$ conditioned on every partial transcript $\vec{v}_1, \ldots, \vec{v}_j$. By the invariant conditions, $(R_{j,1}, \ldots, R_{j,n})|_{\vec{v}_1,\ldots,\vec{v}_j}$ are independent bits with $\Pr[(R_{j,i}|_{\vec{v}_1,\ldots,\vec{v}_j}) = 1] = \gamma_i(\vec{v}_1, \ldots, \vec{v}_j)$. Since

$$\gamma_i(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i}) = \max_{\vec{v}'_{j,-i}} \{\gamma_i(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i}, \vec{v}'_{j,-i})\} \geq \gamma_i(\vec{v}_1, \ldots, \vec{v}_j)$$

for every $i \in [n]$, we can define $(R_{j-1,1}, \ldots, R_{j-1,n})|_{\vec{v}_1,\ldots,\vec{v}_j}$ with the following two properties easily.[8]

- $R_{j,i}|_{\vec{v}_1,\ldots,\vec{v}_j} \leq R_{j-1,i}|_{\vec{v}_1,\ldots,\vec{v}_j}$ for $i = 1, \ldots, n$ with probability 1.
- $(R_{j-1,1}, \ldots, R_{j-1,n})|_{\vec{v}_1,\ldots,\vec{v}_j}$ are independent bits with $\Pr[R_{j-1,i}|_{\vec{v}_1,\ldots,\vec{v}_j} = 1] = \gamma_i(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})$.

This completes the definition of $\vec{R}_{j-1}$. We now check that the constructed $\vec{R}_{j-1}$ satisfies the two invariant conditions. The first condition holds because $D_i \leq R_{j,i} \leq R_{j-1,i}$ for every $i = 1, \ldots, n$ with probability 1. The second condition holds because once we fix $\vec{v}_1, \ldots, \vec{v}_{j-1}$, the probability of $R_{j-1,i} = 1$ depends only on the $v_{j,i}$ component of $\vec{v}_j$, and the $v_{j,i}$'s are independent. More formally, for every $\vec{v}_1, \ldots, \vec{v}_{j-1}$, every $i = 1, \ldots, n$ and every $\vec{r}_{-i} = (r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_n) \in \{0,1\}^{n-1}$, we have

$$\begin{aligned}
&\Pr[R_{j-1,i} = 1 | \vec{v}_1, \ldots, \vec{v}_{j-1}, \vec{R}_{j-1,-i} = \vec{r}_{-i}] \\
&= \mathop{\mathrm{E}}_{\vec{v}_j}[\Pr[R_{j-1,i} = 1 | \vec{v}_1, \ldots, \vec{v}_j, \vec{R}_{j-1,-i} = \vec{r}_{-i}]] \\
&= \mathop{\mathrm{E}}_{\vec{v}_j}[\gamma_i(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})] \quad (\because R_{j-1,i} \text{ and } \vec{R}_{j-1,-i} \text{ are independent given } \vec{v}_1, \ldots, \vec{v}_j) \\
&= \mathop{\mathrm{E}}_{v_{j,i}}[\gamma_i(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})] \\
&= \gamma_i(\vec{v}_1, \ldots, \vec{v}_{j-1})
\end{aligned}$$

To summarize, we constructed $\vec{R} = \vec{R}_0$ such that $R_i \geq D_i$ for every $i \in [n]$ with probability 1, and R_i's are mutually independent with $\Pr[R_i = 1] = \gamma_i$. Recall that the success probability of P^*_{opt} is $\gamma = \max_i\{\gamma_i\}$. By the monotonicity of g, we have

$$\Pr[g(\vec{R}) = 1] \geq \Pr[g(\vec{D}) = 1] \geq \varepsilon = \Pr[g(\vec{X}) = 1].$$

Furthermore, the probability $\Pr[g(\vec{R}) = 1]$ is a monotone increasing function of the probabilities γ_i's. This implies that P^*_{opt} can succeed with probability at least

$$\gamma = \max_i\{\Pr[R_i = 1]\} \geq \Pr[X_i = 1] = \delta,$$

as desired. ■

4.3.2 Recursive Sampling Strategy P^*_{rec}

In this section, we present the recursive sampling strategy P^*_{rec} of Pass and Venkitasubramaniam [30], which is an efficient reduction prover strategy that approximates the optimal strategy P^*_{opt} presented in the previous section.

---

[8] For example, when $R_{j,i} = 1$, we set $R_{j-1,i} = 1$, and when $R_{j,i} = 0$, we toss fresh independent coins and set $R_{j-1,i} = 1$ with appropriate probability to make $\Pr[R_{j-1,i} = 1] = \gamma_i(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})$.

The idea is intuitive, although the analysis is involved. Recall that the optimal strategy P^*_{opt} selects an optimal move that maximizes his success probability after taking the move. P^*_{rec} tries to find a nearly optimal move by sampling. Consider for example that P^*_{rec} needs to select a move $\vec{v}_{j,-i}$ at j-th round. P^*_{rec} uses sampling to generate several candidate moves $\vec{v}^1_{j,-i}, \ldots, \vec{v}^M_{j,-i}$, uses sampling again to estimate *his own* success probability after taking these moves, and select the best one of them (with highest estimated success probability) as his move. Note that in order to estimate his own success probability after taking a certain move $\vec{v}_{j,-i}$, P^*_{rec} needs to simulate the interaction of himself and V in the remaining rounds, which uses sampling to estimate his own success probability again. Hence, P^*_{rec} performs sampling recursively and is called the recursive sampling strategy.

Recall that in $\langle \mathsf{P}^*, \mathsf{V} \rangle$, the moves of P^* and V are $(i, \vec{v}_{1,-i}, \ldots, \vec{v}_{m,-i})$ and $(v_{1,i}, \ldots, v_{m,i})$, respectively. Partial interactions of $\langle \mathsf{P}^*, \mathsf{V} \rangle$ can be described by $(i, \vec{v}_1, \ldots, \vec{v}_j)$ and $(i, \vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})$. We define $\gamma^{rec}(\cdot)$ as follows to denote the success probability of P^*_{rec} conditioned on partial interactions.

$$\gamma^{rec}_i(\vec{v}_1, \ldots, \vec{v}_j) \stackrel{\text{def}}{=} \Pr[\langle \mathsf{P}^*_{rec}, \mathsf{V} \rangle(x) = 1 | i, \vec{v}_1, \ldots, \vec{v}_j], \text{ and}$$

$$\gamma^{rec}_i(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i}) \stackrel{\text{def}}{=} \Pr[\langle \mathsf{P}^*_{rec}, \mathsf{V} \rangle(x) = 1 | i, \vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i}].$$

A formal description of P^*_{rec} can be found in Figure 4.5.

We proceed to analyze the success probability of P^*_{rec}. We first note that P^*_{rec} cannot always approximate the optimal strategy P^*_{opt} with similar success probability, since P^*_{rec} may never see the best move from sampling, and it could be the case that the best move is significantly better than all other moves. Instead, our goal is to show that P^*_{rec} can approximately achieve the lower bound on the success probability we proved for P^*_{opt}.

Observe that although P^*_{rec} cannot find the best move, from the M samples, P^*_{rec} can find a sample that is one of the "top" α-fraction of moves with high probability. Hence, P^*_{rec} can select some top moves at every round. Now, if the optimal strategy P^*_{opt} is not allowed to choose these top α-fraction of moves, then intuitively, P^*_{opt} should not be able to succeed with probability significantly higher than P^*_{rec}.

Specifically, we consider a modified parallel prover strategy $\hat{\mathsf{P}}^{n*}$ from P^{n*}, where for every partial interaction $\vec{v}_1, \ldots, \vec{v}_j$, if the messages $\vec{v}_{j,-i}$ is of the top α-fraction of moves for P^*_{rec} for corresponding partial interaction $(i, \vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})$, then $\hat{\mathsf{P}}^{n*}$ simply aborts (so that $\hat{\mathsf{P}}^{n*}$ fails to convinces any V_i). Intuitively, given oracle access to this $\hat{\mathsf{P}}^{n*}$ effectively turns off the option of P^*_{opt} to take the top α-fraction of moves of P^*_{rec}. Formally, for a given parameter $\alpha \in (0, 1)$ and interaction $\langle \mathsf{P}^{*(\mathsf{P}^{n*})}_{rec}, \mathsf{V} \rangle(x)$, we define α-*top* partial interactions as follows.

Definition 4.15 *We say that a partial interaction* $(i, \vec{v}_1, \ldots, v_{j,i}, \vec{v}_{j,-i}) = (i, \bar{h}, \vec{v}_{j,-i})$

Prover Strategy $\mathsf{P}^*_{rec}(x, n, \delta, \xi)$
/* P^*_{rec} interacts with V and is given oracle access to P^{n*}. */
/* The interaction of $\langle \mathsf{P}^*_{rec}, \mathsf{V} \rangle$ simulates the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^n \rangle$. */

- Let $\alpha = (\xi/2mn)$ and $\eta = (\xi/16n)$.
- In the first move, for every $i \in [n]$, P^*_{rec} estimates his success probabilities after taking move i
$$\gamma_i^{rec} = \Pr[\mathsf{P}^*_{rec} \text{ convinces } \mathsf{V} | i]$$
by sampling, and selects the one with highest (estimated) probability as his first move.
  - Namely, P^*_{rec} simulates the continuation of the interaction after taking first move i for $M_0 = O\left(\frac{1}{\eta^2} \log \frac{n}{\eta}\right)$ times, and computes estimators
$$\tilde{\gamma}_i^{rec} \stackrel{\text{def}}{=} (\#\text{ successful interaction})/M_0.$$
- For each round $j \in [m]$, upon receiving $v_{j,i} = v_j$ from V, P^*_{rec} samples $M = O\left(\frac{1}{\alpha} \cdot \left(m + \log \frac{1}{\eta}\right)\right)$ random moves $\vec{v}^1_{j,-i}, \ldots, \vec{v}^M_{j,-i}$, estimates his success probabilities after taking each move $\vec{v}^\ell_{j,-i}$ for $\ell \in [M]$
$$\gamma_i^{rec}(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i}, \vec{v}^\ell_{j,-i}) = \Pr[\mathsf{P}^*_{rec} \text{ convinces } \mathsf{V} | i, \vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i}, \vec{v}^\ell_{j,-i}]$$
by sampling, and selects the one with highest (estimated) probability as his move.
  - Namely, P^*_{rec} simulates the continuation of the interaction after taking the move $\vec{v}^\ell_{j,-i}$ for $M_j = O\left(\left(\frac{4^j}{\eta}\right)^2 \cdot \left(j + \log \frac{M}{\eta}\right)\right)$ times, and computes estimators
$$\tilde{\gamma}_i^{rec}(\vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i}, \vec{v}^\ell_{j,-i}) \stackrel{\text{def}}{=} (\#\text{ successful interaction})/M_j.$$
  - Let $\vec{v}_{j,-i}$ be the selected move. P^*_{rec} runs P^{n*} on $(v_{j,i}, \vec{v}_{j,-i})$ to generate $\vec{p}_j$ and sends $p_{j,i}$ to V.

Figure 4.5: Recursive sampling strategy P^*_{rec} for constant-round public-coin protocols.

of the interaction of $\langle \mathsf{P}^{*(\mathsf{P}^{n*})}_{rec}, \mathsf{V} \rangle(x)$ *to be* α-top *if*
$$\Pr_{\vec{v}'_{j,-i}}[\gamma_i^{rec}(\bar{h}, \vec{v}_{j,-i}) \leq \gamma_i^{rec}(\bar{h}, \vec{v}'_{j,-i})] \leq \alpha,$$
and we call the corresponding move $\vec{v}_{j,-i}$ *as an* α*-top move with respect to* $(i, \bar{h})$.

With the above definition, a formal description of the modified parallel prover $\hat{\mathsf{P}}^{n*}$

Prover Strategy $\hat{\mathsf{P}}^{n*}(x)$

/* $\hat{\mathsf{P}}^{n*}$ depends on a parameter $\alpha \in (0,1)$ and interaction $\langle \mathsf{P}_{rec}^{*(\mathsf{P}^{n}*)}, \mathsf{V}\rangle(x)$. */

/* $\hat{\mathsf{P}}^{n*}$ "turns off" the α-top moves of $\langle \mathsf{P}_{rec}^{*(\mathsf{P}^{n}*)}, \mathsf{V}\rangle(x)$ for some parameter $\alpha \in (0,1)$ */

- For each round $j \in [m]$, upon receiving $\vec{v}_j$ from V^n, $\hat{\mathsf{P}}^{n*}$ checks if there exists an $i \in [n]$ such that $(i, \vec{v}_1, \ldots, \vec{v}_j)$ is a α-top partial interaction of $\langle \mathsf{P}_{rec}^{*(\mathsf{P}^{n}*)}, \mathsf{V}\rangle(x)$. If so, $\hat{\mathsf{P}}^{n*}$ aborts and fails; otherwise, $\hat{\mathsf{P}}^{n*}$ runs P^{n*} on $\vec{v}_j$, and sends the outputted $\vec{p}_j$ to V^n.

Figure 4.6: Modified parallel prover strategy $\hat{\mathsf{P}}^{n*}$ that "turns off" the α-top moves of $\langle \mathsf{P}_{rec}^{*(\mathsf{P}^{n}*)}, \mathsf{V}\rangle(x)$.

can be found in Figure 4.6. We proceed to lower bound the success probability of P_{rec}^{*} given oracle access to P^{n*} to that of P_{opt}^{*} given oracle access to $\hat{\mathsf{P}}^{n*}$ in the following lemma.

Lemma 4.16 *Let V be a public-coin verifier, $n \in \mathbb{N}, \delta, \xi \in (0,1)$ parameters, $x \in \{0,1\}^*$ an input, P^{n*} a parallel prover strategy, and $\hat{\mathsf{P}}^{n*}$ the modified prover strategy with respect to $\alpha = (\xi/2mn)$ and $\langle \mathsf{P}_{rec}^{*(\mathsf{P}^{n}*)}(n,\delta,\xi), \mathsf{V}\rangle(x)$. We have*

$$\Pr[\langle \mathsf{P}_{rec}^{*(\mathsf{P}^{n*})}(n,\delta,\xi), \mathsf{V}\rangle(x) = 1] \geq \Pr[\langle \mathsf{P}_{opt}^{*(\hat{\mathsf{P}}^{n*})}(n), \mathsf{V}\rangle(x) = 1] - 4\eta,$$

where $\eta = (\xi/16n)$.

Proof. We prove the statement by backward induction on the round $j = m, m-1, \ldots, 1, 0$. Recall that $\gamma_i^{rec}(\cdot)$ denotes the success probability of $\mathsf{P}_{rec}^{*(\mathsf{P}^{n}*)}$ conditioned on partial interactions. We use $\hat{\gamma}_i(\cdot)$ to denote the success probability of $\mathsf{P}_{opt}^{*(\tilde{\mathsf{P}}^{n}*)}$. We use the following two induction hypotheses for each round j.

1. For every partial interaction $(i, \bar{h}) = (i, \vec{v}_1, \ldots, \vec{v}_j)$, $\gamma_i^{rec}(\bar{h}) \geq \hat{\gamma}_i(\bar{h}) - \eta/4^j$.
2. For every partial interaction $(i, \bar{h}) = (i, \vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})$, $\gamma_i^{rec}(\bar{h}) \geq \hat{\gamma}_i(\bar{h}) - \eta/4^{j-1}$.

The base case of our induction is the first induction hypothesis with $j = m$, where we conditioned on complete interactions. The base case holds by observing that the probabilities are 1 iff P^{n*} (resp, $\hat{\mathsf{P}}^{n*}$) convinces V_i on interaction $(\vec{v}_1, \ldots, \vec{v}_m)$. We proceed to show that for every $j \in m$,

- The first induction hypothesis with round j implies the second induction hypothesis with round j.
- The second induction hypothesis with round j implies the first induction hypothesis with round $j-1$.

We prove the second claim first, as it is simpler. Consider a fixed $j \in [m]$ and a partial interaction $(i, \bar{h}) = (i, \vec{v}_1, \ldots, \vec{v}_{j-1})$. Note that in both interactions $\langle \mathsf{P}_{rec}^{*(\mathsf{P}^{n*})}, \mathsf{V} \rangle(x)$ and $\langle \mathsf{P}_{opt}^{*(\hat{\mathsf{P}}^{n*})}, \mathsf{V} \rangle(x)$, V plays moves $v_{j,i}$ uniformly at random. We have

$$\gamma_i^{rec}(\bar{h}) = \mathop{\mathrm{E}}_{v_{j,i}}\left[\gamma_i^{rec}(\bar{h}, v_{j,i})\right] \text{ and } \hat{\gamma}_i(\bar{h}) = \mathop{\mathrm{E}}_{v_{j,i}}\left[\hat{\gamma}_i(\bar{h}, v_{j,i})\right].$$

It follows by the second induction hypothesis that

$$\gamma_i^{rec}(\bar{h}) = \mathop{\mathrm{E}}_{v_{j,i}}\left[\gamma_i^{rec}(\bar{h}, v_{j,i})\right] \geq \mathop{\mathrm{E}}_{v_{j,i}}\left[\hat{\gamma}_i(\bar{h}, v_{j,i}) - (\eta/4^{j-1})\right] = \hat{\gamma}_i(\bar{h}) - (\eta/4^{j-1}).$$

We proceed to prove the first claim. Again, fix a $j \in [m]$ and a partial interaction $(i, \bar{h}) = (i, \vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})$. Recall that $\mathsf{P}_{opt}^{*(\hat{\mathsf{P}}^{n*})}$ plays optimal strategy, we have

$$\hat{\gamma}_i(\bar{h}) = \max_{\vec{v}_{j,-i}}\{\hat{\gamma}_i(\bar{h}, \vec{v}_{j,-i})\}.$$

Let $\vec{v}_{j,-i}^{*}$ be the move selected by $\mathsf{P}_{opt}^{*(\hat{\mathsf{P}}^{n*})}$ that maximizes $\hat{\gamma}_i(\bar{h}, \vec{v}_{j,-i})$. Since P_{opt}^{*} is given $\hat{\mathsf{P}}^{n*}$, where the α-top moves are turned off, we know that $\vec{v}_{j,-i}^{*}$ cannot be a α-top move.

On the other hand, $\mathsf{P}_{rec}^{*(\mathsf{P}^{n*})}$ selects his $\vec{v}_{j,-i}$ by sampling M candidates $\vec{v}_{j,-i}^{1}, \ldots, \vec{v}_{j,-i}^{M}$, and select the best one according to his estimation. We shall argue that with high probability, $\mathsf{P}_{rec}^{*(\mathsf{P}^{n*})}$ selects a move $\vec{v}_{j,-i}'$ that is comparable to the move $\vec{v}_{j,-i}^{*}$ for $\mathsf{P}_{opt}^{*(\hat{\mathsf{P}}^{n*})}$.

First, let us consider all moves $\vec{v}_{j,-i}$ that is at least as good as $\vec{v}_{j,-i}^{*}$ for $\mathsf{P}_{rec}^{*(\mathsf{P}^{n*})}$. Namely, we consider the set of moves

$$H \stackrel{\text{def}}{=} \{\vec{v}_{j,-i} : \gamma_i^{rec}(\bar{h}, \vec{v}_{j,-i}) \geq \gamma_i^{rec}(\bar{h}, \vec{v}_{j,-i}^{*})\}.$$

Note that H is strictly larger than the set of α-top moves, since $\vec{v}_{j,-i}^{*}$ is not a α-top move. Furthermore, we have

$$\Pr_{\vec{v}_{j,-i}}[\vec{v}_{j,-i} \in H] \geq \alpha.$$

Now, by choosing the constants in the big-O notations of M and M_j's properly, we have

- With probability at least $1 - (\eta/(2 \cdot 4^j))$ over the randomness of choosing $\vec{v}_{j,-i}^{1}, \ldots, \vec{v}_{j,-i}^{M}$, there exists some $\ell^* \in [M]$ such that $\vec{v}_{j,-i}^{\ell^*} \in H$.
- With probability at least $1 - (\eta/(2 \cdot 4^j))$ over the randomness of estimating $\tilde{\gamma}_i^{rec}(\cdot)$, for every $\ell \in [M]$,

$$|\gamma_i^{rec}(\bar{h}, \vec{v}_{j,-i}^{\ell}) - \tilde{\gamma}_i^{rec}(\bar{h}, \vec{v}_{j,-i}^{\ell})| \leq (\eta/4^j).$$

When both events hold, $\mathsf{P}_{rec}^{*(\mathsf{P}^{n*})}$ selects a move $\vec{v}'_{j,-i}$ such that

$$\begin{aligned}
&\gamma_i^{rec}(\bar{h}, \vec{v}'_{j,-i}) \\
&\geq \tilde{\gamma}_i^{rec}(\bar{h}, \vec{v}'_{j,-i}) - (\eta/4^j) \quad \text{(estimators have small error)} \\
&\geq \tilde{\gamma}_i^{rec}(\bar{h}, \vec{v}^{\ell *}_{j,-i}) - (\eta/4^j) \quad (\mathsf{P}^*_{rec} \text{ selects } \vec{v}'_{j,-i} \text{ with highest estimators}) \\
&\geq \gamma_i^{rec}(\bar{h}, \vec{v}^{\ell *}_{j,-i}) - (2\eta/4^j) \quad \text{(estimators have small error)} \\
&\geq \gamma_i^{rec}(\bar{h}, \vec{v}^{*}_{j,-i}) - (2\eta/4^j) \quad \text{(by definition of } H) \\
&\geq \hat{\gamma}_i(\bar{h}, \vec{v}^{*}_{j,-i}) - (3\eta/4^j) \quad \text{(by induction hypothesis)} \\
&= \hat{\gamma}_i(\bar{h}) - (3\eta/4^j).
\end{aligned}$$

By an union bound, both events hold with probability at least $1 - (\eta/4^j)$, and hence

$$\gamma_i^{rec}(\bar{h}) \geq (\hat{\gamma}_i(\bar{h}) - 3\eta/4^j) - (\eta/4^j) = \hat{\gamma}_i(\bar{h}) - (\eta/4^{j-1}).$$

This finish the proof of induction. We proceed to use the first induction hypothesis with $j = 0$ to prove the lemma. The induction hypothesis says that

$$\gamma_i^{rec} \geq \hat{\gamma}_i - \eta \quad \forall i \in [n].$$

Let $i^* \in [n]$ be the coordinate that maximizes $\hat{\gamma}_i$. By a similar argument as above, with probability at least $1 - \eta$, all estimators have error less than η, and in this case, $\mathsf{P}_{rec}^{*(\mathsf{P}^{n*})}$ selects a coordinate i' with

$$\gamma_{i'}^{rec} \geq \tilde{\gamma}_{i'}^{rec} - \eta \geq \tilde{\gamma}_{i*}^{rec} - \eta \geq \gamma_{i*}^{rec} - 2\eta \geq \hat{\gamma}_{i*}^{rec} - 3\eta.$$

It follows that the success probability of $\mathsf{P}_{rec}^{*(\mathsf{P}^{n*})}$ is at least

$$\Pr[\langle \mathsf{P}_{rec}^{*(\mathsf{P}^{n*})}, \mathsf{V} \rangle(x) = 1] \geq (\hat{\gamma}_{i*}^{rec} - 3\eta) - \eta = \Pr[\langle \mathsf{P}_{opt}^{*(\hat{\mathsf{P}}^{n*})}, \mathsf{V} \rangle(x) = 1] - 4\eta,$$

as desired. ■

Finally, we claim that by construction, $\hat{\mathsf{P}}^{n*}$ aborts with probability at most $nm \cdot \alpha = \xi/2$, and hence

$$\Pr[\langle \hat{\mathsf{P}}^{n*}, \mathsf{V}^{n,g} \rangle(x) = 1] \geq \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,g} \rangle(x) = 1] - \xi/2.$$

This is because that, by definition, for every partial interaction $(i, \vec{v}_1, \ldots, \vec{v}_{j-1}, v_{j,i})$ of $\langle \mathsf{P}^*_{rec}, \mathsf{V} \rangle$,

$$\Pr_{\vec{v}_{j,-i}}[\vec{v}_{j,-i} \text{ is an } \alpha\text{-top move}] \leq \alpha.$$

Hence, conditioned on every partial transcript $(\vec{v}_1, \ldots, \vec{v}_{j-1})$, $\hat{\mathsf{P}}^{n*}$ aborts at the j-th round with probability at most $n \cdot \alpha$. By a union bound over the number of rounds, $\hat{\mathsf{P}}^{n*}$ abort with probability at most $nm \cdot \alpha$.

We are ready to prove Theorem 4.12.

Proof. (of Theorem 4.12) Consider the corresponding $\hat{\mathsf{P}}^{n*}$ with respect to $\alpha = (\xi/2mn)$ and $\langle \mathsf{P}_{rec}^{*(\mathsf{P}^{n*})}(n,\delta,\xi), \mathsf{V}\rangle(x)$. By the above claim, we have

$$\Pr[\langle \hat{\mathsf{P}}^{n*}, \mathsf{V}^{n,g}\rangle(x) = 1] \geq \Pr[\langle \mathsf{P}^{n*}, \mathsf{V}^{n,g}\rangle(x) = 1] - \xi/2 \geq \varepsilon + \xi/2.$$

Now, consider i.i.d. random bits $\vec{X'} = (X'_1, \ldots, X'_n)$ with $\Pr[X'_i = 1] = \delta' \stackrel{\text{def}}{=} \delta + (\xi/4n)$. By a union bound, the statistical distance

$$\Delta(\vec{X}, \vec{X'}) \leq n \cdot \frac{\xi}{4n} = \frac{\xi}{4}.$$

Since g is a deterministic function, we have

$$\Pr[g(\vec{X'}) = 1] \leq \Pr[g(\vec{X}) = 1] + \xi/4 < \varepsilon + \xi/2.$$

Hence, by Lemma 4.14, we have $\Pr[\langle \mathsf{P}_{opt}^{*(\hat{\mathsf{P}}^{n*})}, \mathsf{V}\rangle(x) = 1] \geq \delta'$, and by Lemma 4.16,

$$\Pr[\langle \mathsf{P}_{rec}^{*(\mathsf{P}^{n*})}, \mathsf{V}\rangle(x) = 1] \geq \Pr[\langle \mathsf{P}_{opt}^{*(\hat{\mathsf{P}}^{n*})}, \mathsf{V}\rangle(x) = 1] - 4\eta \geq \left(\delta + \frac{\xi}{4n}\right) - \frac{\xi}{4n} = \delta,$$

which gives the desired lower bound on the success probability of P_{rec}^*.

Finally, for the runtime of P_{rec}^*, observe that at each move, P_{rec}^* calls itself recursively $\text{poly}(2^m, n, \xi^{-1})$ times. The runtime of P_{rec}^* is $\text{poly}(|x|, 2^{m^2}, n^m, \xi^{-m})$. ■

Chapter 5

Applications to Security Amplification for Cryptographic Primitives

In this chapter, we present applications of efficient parallel repetition theorems (and the reductions for proving the theorems) to the field of security amplification for cryptographic primitives.

Security amplification for cryptographic primitives is a basic question in cryptography that has been studied since the seminal work of Yao [38]. In contrast to general cryptographic tasks where we construct cryptographic primitives from more basic primitives and computational hardness assumptions, the task of security amplification asks whether we can construct a *fully* secure cryptographic primitive from *the same* primitive with *weak* security. For example, given a weak one-way function, where no PPT algorithm can invert on more than 99% of the inputs, can we construct a fully secure one-way function, where no PPT algorithm can invert on a non-negligible fraction of the inputs?

In addition to being a natural question in its own right, security amplification is a useful step in constructing cryptographic primitives. Instead of constructing a primitive from scratch, we first construct one with weaker security first, and then amplify its security to obtain a fully secure one. Security amplification is also a way to understand and minimize our assumptions, as it asks what is the quantitatively weakest form of security that implies the desired security.

Security amplification has been extensively studied in recent years for a variety of primitives with different types of security properties. To name a few, it has been studied for encryption schemes [9, 21], commitment schemes [5, 37, 17], oblivious transfer [5, 37], CAPTCHAs [2, 24, 26], message authentication codes (MACs), digital signatures, and pseudorandom functions (PRFs) [7].

In such tasks, it is also desirable that the constructions are *efficient*, where the efficiency can be measured in various ways, such as communication complexity, round

complexity, or the number of calls to the weak primitive. (Parallel) repetition, when it works, usually gives a simple and efficient way to achieve security amplification. For example, one simple way to amplify the security of one-way functions is to concatenate the outputs of a given weak one-way function on several independent inputs.

As mentioned in the introduction (Chapter 1), efficient parallel repetition theorems are useful tools to analyze such constructions. For example, the security of one-way functions can be captured as the soundness of a two-message interactive protocols, where the verifier sends $f(x)$ and accepts if he receives some x' from the prover with $f(x') = f(x)$. The above security amplification of one-way function corresponds exactly to a parallel repetition of this protocol, and hence the corresponding parallel repetition theorem implies security of the constructed one-way function.

In many other cases, although the security property of primitives can also be captured by the soundness property of certain interactive protocols, the primitives are more interactive in nature so that their security cannot be captured by a class of protocols where parallel repetition theorems are available. Nevertheless, the security property may have additional structure so that the black-box reductions used to prove parallel repetition theorems can be implemented in the corresponding settings to prove corresponding repetition theorems. There are also cases that require more involved constructions than (parallel) repetitions to amplify the security, but parallel repetition theorems are still useful in analyzing the constructions.

We present two such examples in this chapter, where we improve the efficiency of security amplification for several primitives by proving/improving the corresponding parallel repetition theorems and/or proposing better constructions. We propose a more efficient security amplification for commitment schemes in Section 5.1. Then in Section 5.2, we improve the analysis of parallel repetition for "dynamic weakly verifiable puzzle systems" of Dodis, Impagliazzo, Jaiswal, and Kabanets [7]. As a consequence, this improves the efficiency of security amplification for message authentication codes (MACs), digital signatures, and pseudorandom functions (PRFs).

In addition, we show in Section 5.1.2 that the threshold repetition theorem for "two-phase puzzle systems" implies sequential repetition theorem for computationally sound protocols with threshold verifiers.

5.1 Security Amplification for Commitment Schemes

Commitment schemes are interactive protocols that are digital analogues of safes, where (in the commit stage), a sender Alice can put a value inside the safe and send it to a receiver Bob without leaking any information about the value (hiding property), and later on (in the reveal stage), Alice can only open the safe in one way to reveal a unique value to Bob (binding property). The goal of security amplification is to turn a weak bit-commitment scheme Com_0, where both properties can be broken with a bounded but non-negligible probability, to a fully secure one, where both properties

can be broken with only a negligible probability.

More precisely, we say a bit-commitment scheme Com_0 is *p-hiding* if no (PPT or unbounded) adversarial receiver, who may deviate from the prescribed protocol arbitrarily, can guess the committed bit correctly with probability better than $(1 + p)/2$, and *q-binding* if no (PPT or unbounded) adversarial sender can open in two ways with probability better than q. The goal is to construct a secure (i.e., ngl-hiding and ngl-binding for some negligible function ngl) commitment scheme Com from a weak Com_0 that is p-hiding and q-binding for, say, constant p and q. It is desirable that the transformation is black-box (i.e., uses Com_0 in a black-box matter) since it is simpler and more efficient.

Security amplification for commitment schemes has been studied in [5, 37, 17] from information-theoretic to computational settings. Damgård, Kilian and Salvail [5] studied the question in the (simpler) information-theoretic setting, where the security holds against unbounded adversaries. Wullschleger [37] extended the result to the computational and passive setting, where the security holds against efficient (PPT) and semi-honest adversaries. The result of Wullschleger can be further extend to the active setting by applying the generic Goldreich-Micali-Wigderson compiler [15]. However, the compiler makes the construction non-black-box and blows up the complexity significantly.[1] Finally, Halevi and Rabin [17] proved security amplification in the general computational and active setting.

All previous works focus on feasibility results. Namely, for what values of p and q is the security amplification achievable. In the information-theoretic setting, Damgård, Kilian and Salvail [5] showed that a black-box transformation is possible *if and only if* $p + q \leq 1 - 1/\mathrm{poly}(s)$, where s is a security parameter. Wullschleger [37] showed the same result holds for the computational and passive setting. Halevi and Rabin [17] analyzed the transformation of [5] in the computational setting and proved that a black-box transformation is possible whenever $p + q \leq 1 - 1/\mathrm{polylog}(s)$. Recently and independent of our work, Holenstein and Schoenebeck [22] improved the result to be tight. Namely. they showed that in the computational setting, black-box security amplification is achievable if and only if $p + q \leq 1 - 1/\mathrm{poly}(s)$.

However, in terms of efficiency, the existing transformations are suboptimal. To measure the efficiency, we consider the number of black-box calls to Com_0 that Com makes when p and q are constants with $p + q < 1$. We note that in the computational setting, black-box calls to Com_0 need to be done sequentially,[2] and hence the number of black-box calls affects not only the communication complexity, but also the round

[1] Another non-black-box solution is to first construct a one-way function from Com_0, which can be done provided $p + q \leq 1 - 1/\mathrm{poly}(s)$ [25], and then construct a secure commitment scheme from the one-way function [29, 18]. However, this construction is indirect and also highly inefficient.

[2] In general, the commit stage can consist of multiple rounds. If the black-box calls are done in parallel, one can show, by modifying the negative example of Bellare, Impagliazzo, and Naor [1] for computationally sound protocols, that the security may not be amplified at all.

	Efficiency (constants p, q)			Feasibility
Work	Number of black-box calls	Length of committed string	Rate	Applicable range of parameters
[17]	$\omega(\log^2 s)$	1	$\omega(\log^2 s)$	$p + q < 1 - 1/\mathrm{poly}\log(s)$
[22]	$\omega(\log^2 s)$	1	$\omega(\log^2 s)$	$p + q < 1 - 1/\mathrm{poly}(s)$
Ours	$\omega(\log s)$	$O(\log s)$	$\omega(1)$	$p + q < 1 - 1/\mathrm{poly}\log(s)$
Ours + [22]	$\omega(\log s)$	$O(\log s)$	$\omega(1)$	$p + q < 1 - 1/\mathrm{poly}(s)$

Figure 5.1: Summary of results on security amplification for commitment schemes in the computational setting. Efficiency measures the cost of amplifying commitment schemes from constant security to negligible security. Feasibility refers to the parameter range that security amplification is achievable.

complexity of the resulting protocol.

All existing solutions requires $\omega(\log^2 s)$ black-box calls to securely commit a single bit, where s is the security parameter. At a high level, the reason is that they amplify the hiding and binding property *separately*. Amplifying each property from constant to negligible seems to require $\omega(\log s)$ black-box calls, which is the case of the existing constructions and results in $\omega(\log^2 s)$ black-box calls in total.

Furthermore, the existing transformations construct *bit*-commitment schemes from a (weak) bit commitment scheme Com_0. When a sender Alice wants to commit to a string $x \in \{0,1\}^*$, she needs to use the resulting bit-commitment schemes to commit to x bit by bit separately, each of which requires $\omega(\log^2 s)$ black-box calls to Com_0. It is natural to ask if we can do better in terms of the *rate*, i.e., the number of black-box calls per committed bit. For example, can we commit to a string with $o(\log s)$ black-box calls per bit?

Our improvements. We give a transformation that amplifies a (weak) bit commitment scheme Com_0 with constant security to a $O(\log s)$-bit string commitment scheme with negligible security using only $\omega(\log s)$ black-box calls to Com_0, where $O(\log s)$ (resp., $\omega(\log s)$) denotes an arbitrary $O(\log s)$ (resp., $\omega(\log s)$) function of the security parameter s. In terms of rate, we achieve $\omega(1)$ black-box calls per committed bit. We use error-correcting codes and randomness extractors to amplify both hiding and binding properties *simultaneously*, which allow us to bypass the $\omega(\log^2 s)$ barrier of the previous results. A summary of our result and existing results can be found in Figure 5.1.

Application of parallel repetition theorems. In analyzing our construction, we need to upper bound the probability that an adversarial sender (resp., receiver) breaks the binding (resp., hiding) property of at least r out of n invocations of Com_0 for $1 \leq r \leq n$. Note that, although we argued that the calls to Com_0 in the commit

stage needs to be done *sequentially*, all commitments (of Com_0) are decommitted *in parallel* in the reveal stage. We model the security of commitment schemes as (the hardness of) solving "two-phase" (interactive) puzzle systems (implicit in [17]), and study the hardness of the puzzle systems under this type of repetition.

We observe that, while the scenario is different from parallel repetition for interactive protocols, the reduction algorithm for three-message protocols presented in Section 4.2 can be implemented in this scenario, yielding a tight threshold repetition theorem for two-phase puzzle systems.

We remark that, independent of our work, Holenstein and Schoenebeck [22] came up with the same observation. In fact, Holenstein and Schoenebeck [22] proved their more general tight monotone repetition theorem (described in Section 4.2.3) in the context of two-phase puzzle systems (which they referred to interactive weakly verifiable puzzle systems), and used it to analyze their construction of security amplification for commitment schemes.

We proceed to present our construction in following sections. We start with a formal definition of commitment schemes and some preliminaries in Section 5.1.1. Then we define two-phase puzzle system and present a corresponding threshold repetition theorem in Section 5.1.2. As a small digression, we also argue that the threshold repetition theorem of two-phase puzzle system implies sequential repetition theorem for computationally sound protocols in Section 5.1.2. We present our construction in Section 5.1.3, and analyze it in Section 5.1.4 and 5.1.5.

5.1.1 Preliminaries and Theorem Statement

In this section, we give a formal definition of commitment schemes and present our theorem statement. We will also state two preliminary lemmas that we will use at the end of this section. We consider a standard model where the communication is over a noiseless channel and the decommitment is non-interactive [13, 17].

Definition 5.1 (Commitment Scheme) *A* **commitment scheme** *is an interactive protocol* $\text{Com} = (S, R)$ *with the following properties:*

1. *Scheme* Com *consists of two stages: a* **commit stage** *and a* **reveal stage**. *In both stages, the* **sender** S *and the* **receiver** R *receive a security parameter* 1^s *as common input.*

2. *At the beginning of the commit stage, sender* S *receives a private input* $v \in \{0,1\}^t$*, which denotes the string to which* S *is supposed to commit. The commitment stage results in a joint output, which we call the* **commitment** $x = \text{output}((S(v), R)(1^s))$*, and a private output for* S*, which we call the* **decommitment** *string* $d = \text{output}_S((S(v), R)(1^s))$*. Without loss of generality,* x *can be taken to be the full transcript of the interaction between* S *and* R*, and* d *to be the private coin tosses of* S*.*

3. *In the reveal stage, sender S sends the pair (v, d), where d is the decommitment string for string v. Receiver R accepts or rejects based on v, d, x.*

4. *Both sender S and receiver R are efficient, i.e., both run in probabilistic polynomial time in the security parameter s.*

5. *R will always accept with probability $1 - \mathsf{ngl}$ if both the sender S and the receiver R follow their prescribed strategy. If R accepts with probability 1, we say* Com *has* **perfect correctness**.

6. *When the commit string v is just a bit in $\{0, 1\}$, we call* Com *a* **bit-commitment scheme**. *Otherwise, we call* Com *a t-***bit string-commitment scheme**.

Remark 5.2 The assumption of a non-interactive reveal stage is essential in both our work and the previous work [17]. This assumption can be made without loss of generality as long as no additional property (e.g., if the sender wants to decommit in a zero-knowledge manner) is required, because in the reveal stage, the sender S can send his coin tosses to the receiver R, who can check the consistency and simulate the protocol. On the other hand, the assumption of perfect correctness can be relaxed to $(1 - \mathsf{ngl})$-correctness in both works.

We proceed to define the hiding and binding properties of commitment schemes. To facilitate the presentation of our results and analysis, we are precise about the adversary's running time in the definition and define the binding property in terms of binding games.

Definition 5.3 (p-hiding against time T) *A commitment scheme* Com $= (S, R)$ *is* p-hiding against uniform time T *if for every probabilistic time T cheating receiver R^*, the distributions* $(\text{view}_{R^*}(S(U_t), R^*), U_t)$ *and* $(\text{view}_{R^*}(S(U_t), R^*), U'_t)$ *are p-indistinguishable for time T, where U'_t is an i.i.d. copy of U_t. That is, for every probabilistic time T distinguisher D,*

$$|\Pr[D(\text{view}_{R^*}(S(U_t), R^*), U_t) = 1] - \Pr[D(\text{view}_{R^*}(S(U_t), R^*), U'_t) = 1]| \leq p/2$$

We say Com *is* p**-hiding** *if for every constant c,* $\text{Com}(1^s)$ *is p-hiding against time s^c for sufficiently large security parameter s.*

We remark that the hiding property above is defined as the indistinguishability for *random values*, which does not generally imply the standard semantic security for the hiding property. Nevertheless, it is easy to transform a commitment scheme Com with the above hiding property to one with standard semantic security – one can use Com to commit to a random string $r \in_R \{0, 1\}^t$, and use r as a one-time pad to hide the actual string v that the sender wants to commit to.

Remark 5.4 For bit-commitment schemes, p-hiding is equivalent to saying that the receiver can guess the committed bit with probability at most $1/2 + p/2$. Formally, for every time T predictor P,

$$\Pr[P(\text{view}_{R^*}(S(U_1), R^*)) = U_1)] \leq 1/2 + p/2.$$

Definition 5.5 (Binding Game) *The* **binding game** *for a commitment scheme* $\text{Com} = (S, R)$ *is played between a honest receiver* R*, and* (S^*, F)*, a cheating sender* S^* *with a decommitment finder* F*. The game consists of two stages:*

1. *In the commit stage,* S^* *interacts with* R *to produce a view* $\text{view}_{S^*}(S^*, R)$*.*
2. *In the decommitment finding stage,* F *gets the view* $\text{view}_{S^*}(S^*, R)$*, and produces two decommitment strings* (s, d) *and* (s', d')*.*

(S^*, F) **succeeds** *if in the reveal stage,* R *accepts both decommitment strings* (s, d) *and* (s', d')*.*

Definition 5.6 (q-binding against time T) *A commitment scheme* $\text{Com} = (S, R)$ *is* q**-binding against time** T*, if in the binding game, for every time* T *pair* (S^*, F)*, the success probability of* (S^*, F) *is at most* q*. We say* Com *is* q**-binding** *if for every constant* c*,* $\text{Com}(1^s)$ *is* q*-binding against time* s^c *for sufficiently large security parameter* s*.*

Definition 5.7 (security of commitment schemes) *A commitment scheme* Com *is* (p, q)**-secure (against time** T**)** *if* Com *is* p*-hiding and* q*-binding (against time* T*).* Com *is* **secure** *if for every constant* c*,* $\text{Com}(1^s)$ *is* (s^{-c}, s^{-c})*-secure for sufficiently large security parameter* s*.*

Theorem statement. We proceed to state our theorem on efficient security amplification for commitment schemes. The following theorem says that we can securely commit to a $O(\log s)$-bit string using only $\omega(\log s)$ black-box call to a weak commitment scheme Com_0 with constant hiding and binding properties.

Theorem 5.8 *Let* $p, q \in (0, 1)$ *be constants with* $p + q < 1$*. Suppose there exists a* (p, q)*-secure bit commitment scheme* Com_0*. Then for every* $t(s) \leq \text{poly}(s)$*,* $n(s) = \omega(t + \log s)$ *where* s *is the security parameter, there exists a secure* t*-bit string-commitment scheme* Com *that makes only* n *black-box calls to* Com_0*.*

Preliminary lemmas. We proceed to state two preliminary lemmas that we use in the analysis of our construction. The first lemma says that a random (systematic) linear codes is a good error correcting code (in terms of min-distance of the code) with overwhelming probability. The lemma can be proved by standard probabilistic methods. The constants in the lemma are actually small.

Definition 5.9 *The* Hamming distance *of two strings x and y is the number of coordinates i such that $x_i \neq y_i$. Let $C : \{0,1\}^n \to \{0,1\}^{n'}$ be a code. The* minimum distance *of C is the minimum Hamming distance over all pairs of codewords $C(x)$ and $C(y)$ such that $x \neq y$.*

Lemma 5.10 *There exist universal constants d_0, d_1 such that the following holds. Let k be a positive integer, and $\gamma, \delta \in [0,1]$ be numbers such that $\gamma > d_0 \cdot \delta \log(1/\delta)$. Let n be an integer such that $n > d_1 \cdot k/\delta$. Let $C : \{0,1\}^n \to \{0,1\}^{(1+\gamma)n}$ be a random linear code defined by $C(m) = (m, Am)$, where $A \in \{0,1\}^{\gamma n \times n}$ is a random 0-1 matrix. Then C has minimum distance at least $\delta \cdot n$ with probability at least $1 - 2^{-k}/2$.*

We also need the classic Goldreich-Levin theorem, which says that we can extract pseudorandom bits from (computationally) unpredictable distributions.

Lemma 5.11 (Goldreich-Levin [14]) *There is an oracle algorithm $B^{(\cdot)}$ such that for any $x \in \{0,1\}^n$ and any oracle A satisfying*

$$\Pr_{r \leftarrow U_n}[A(r) = x \cdot r] > \frac{1}{2} + \gamma,$$

B^A makes $O(n/\gamma^2)$ queries to A and then efficiently outputs a list of size $O(1/\gamma^2)$ elements such that x is in the list with probability greater equal than $\frac{1}{2}$.

5.1.2 Two-Phase Puzzles Systems

In this section, we define two-phase puzzle systems to capture the security (both binding and hiding) properties of commitment schemes. Informally, a two-phase puzzle system $\mathbf{P}$ consists of a *puzzle generation* phase and a *puzzle solving* phase. In the puzzle generation phase, a solver $\mathbf{S}$ *interacts* with $\mathbf{P}$ to generate a puzzle p, and then in the puzzle solving phase, $\mathbf{S}$ sends an answer a to $\mathbf{P}$. $\mathbf{P}$ accepts if the answer a is correct. We mention that this model generalizes the "weakly verifiable puzzle systems" of Canetti, Halevi, and Steiner [2] in that we allow interactive puzzle generation.

Definition 5.12 (Two-Phase Puzzle System) *A* **two-phase puzzle system** $\mathbf{P} = (G, V)$ *consists of a* PPT *(interactive) puzzle generator G and a deterministic polynomial-time puzzle verifier V. Let $\mathbf{S}$ be a solver for $\mathbf{P}$. The interaction of $\langle \mathbf{S}, \mathbf{P} \rangle$ consists of two phases, where the first phase corresponds to the* **puzzle generation phase***, and the second is the* **puzzle solving phase***. More precisely,*

- *In the first phase, the solver and the generator jointly generate a puzzle $p \leftarrow \langle \mathbf{S}, G(c) \rangle(1^s)$, where c is the private coins of G. The generation of p may take polynomially many rounds.*

- *In the second phase,* $\mathbf{S}$ *sends an answer* $a = \mathbf{S}(p)$ *to* $\mathbf{P}$
- *In the end of the protocol,* $\mathbf{P}$ *verifies the answer using* V *and accepts iff* $V(c, a) = 1$ *(i.e., the answer* a *is* correct*).*

Note that $\mathbf{S}$ does not know the coins c of G, so $\mathbf{S}$ may not be able to verify the correctness of an answer a.

Definition 5.13 (Hardness of Solving a Puzzle) *A two-phase puzzle system* $\mathbf{P}$ *is* δ-hard against time $T = T(s)$ *if for every time* T *solver* $\mathbf{S}$*, the success probability of* $\mathbf{S}$ *is at most* δ*. We say* $\mathbf{P}$ *is* δ*-hard if* $\mathbf{P}$ *is* δ*-hard against time* s^c *for all constant* c*.*

Interested in relating the hardness of solving (at least) r out of n puzzles to the hardness of solving a single puzzle, we proceed to define (n, r)-repetition of a two-phase puzzle system.

Definition 5.14 ((n, r)-Repetition of Two-Phase Puzzle Systems) *Let* $\mathbf{P} = (G, V)$ *be a two-phase puzzle system. We define the* (n, r)*-repetition of* $\mathbf{P}$ *to be a two-phase puzzle system* $\mathbf{P}^{n,r} = (G^n, V^{n,r})$ *such that (1) in the first phase,* $\mathbf{P}^{n,r}$ sequentially *generates* n *puzzles with a solver, and (2)* $\mathbf{P}^{n,r}$ *accepts the* n*-fold answer received from the solver in the second phase if at least* r *out of* n *answers are correct. More precisely, let* $\mathbf{S}^n$ *be a solver for* $\mathbf{P}^{n,r}$*. The interaction of* $\langle \mathbf{S}^n, \mathbf{P}^{n,r} \rangle$ *is defined below.*

- *In the first phase,* $\langle \mathbf{S}^n, G^n(c_1, \ldots, c_n) \rangle$ *generates* n *puzzles* $(p_1, \ldots, p_n)$ *sequentially by running* $\langle \mathbf{S}^n, G(c_i) \rangle(1^s)$ *for* $i = 1, 2, \ldots, n$*.*
- *In the second phase,* $\mathbf{S}^n$ *sends a* n*-fold answer* $\vec{a} = (a_1, \ldots, a_n) \leftarrow \mathbf{S}^n(\vec{p})$ *to* $\mathbf{P}^{n,r}$*.*
- *At the end of the protocol,* $\mathbf{P}^{n,r}$ *accepts iff at least* r *copies of* $V(c_i, a_i)$ *accept.*

We remark that although in the above definition, the puzzles are generated sequentially, it captures the *parallel repetition* of weakly verifiable puzzle systems considered by Canetti, Halevi and Steiner [2]. In our model, the solver starts to solve the puzzles after all of them are generated. Thus, when the puzzles are generated solely by $\mathbf{P}$, which is the case of the weakly-verifiable puzzle systems, parallel generation and sequential generation are equivalent. We also remark that in order to obtain hardness amplification results, we cannot consider parallel repetition in the (interactive) puzzle generation phase. Indeed, the negative example of Bellare, Impagliazzo, and Naor [1] for interactive arguments can be adapted to our model, showing that the hardness may not be amplified for the case of parallel puzzle generation.

As an example, we argue that the hardness of two-phase puzzle systems captures the binding property of a commitment scheme $\mathrm{Com}_0 = (S, R)$ as follows. The solver $\mathbf{S}$ plays the role of a cheating sender S^* and the generator G plays the role of the

honest receiver R. Then the puzzle is the commitment generated jointly by $\mathbf{S}$ and G according to the commitment scheme Com_0. A valid answer for the puzzle is a pair of decommitment strings $((v,d),(v',d'))$ that are both accepted by the receiver R. Thus, Com_0 being q-binding against time T corresponds to the puzzle system begin q-hard against time T, and the hardness of breaking the binding property of at least r out of n sequentially committed commitments translates to the hardness of (n,r)-repetition of the corresponding puzzle system.

We proceed to present a tight threshold repetition theorem for two-phase puzzle systems, which says that the hardness of two-phase puzzle systems behaves as independent events under (n,r)-repetition. Namely, if a two-phase puzzle system $\mathbf{P}$ is δ-hard, then its (n,r)-repetition $\mathbf{P}^{n,r}$ has hardness $P(n,r,\delta)+\mathsf{ngl}$, where ngl is a negligible function and $P(n,r,\delta)=\Pr[\sum_i X_i \geq r]$ with X_i's being i.i.d. random bits with $\Pr[X_i=1]=\delta$.

Theorem 5.15 *Let $\mathbf{P}$ be a two-phase puzzle system. There exists a solver $\mathbf{S}^*$ such that for every security parameter $s\in\mathbb{N}$, every $n,r\in\mathbb{N}$ with $r\leq n$, every $\delta,\xi\in(0,1)$, and every solver $\mathbf{S}^{n*}$ for $\mathbf{P}^{n,r}$,*

1. $\Pr[\langle \mathbf{S}^{n*},\mathbf{P}^{n,r}\rangle(1^s)=1]\geq P(n,r,\delta)+\xi$

$$\Rightarrow \Pr[\langle \mathbf{S}^{*(\mathbf{S}^{n*})}(n,r,\delta,\xi),\mathbf{P}\rangle(1^s)=1]\geq \delta+\frac{\xi}{10n}.$$

2. $\mathsf{P}^{*(\cdot)}(1^s,n,r,\delta,\xi)$ *runs in time* $\mathrm{poly}(s,n,\xi^{-1})$ *given oracle access to* $\mathbf{S}^{n*}(1^s)$.

As mentioned earlier in this section, although (n,r)-repetition of two-phase puzzle system is different from parallel repetition of interactive protocols, the reduction algorithm presented in Section 4.2 for proving threshold repetition theorem for three-message protocols can be implemented in this model. Note that the puzzle $\mathbf{P}$ plays the role of the verifier V and the solver $\mathbf{S}^*$ plays the role of prover P^*. To verify this claim, let us investigate the correlation reduction procedure in Figure 4.2 and the reduction prover strategy defined in Figure 4.3.

- Recall that in the correlation reduction CR, P^* does not interact with V at all. In CR, P^* simply runs the interaction $\langle \mathsf{P}^{n*},\mathsf{V}^{n,k}\rangle$ internally with different coins, and constructs a prover $\mathsf{P}^{n'*}$ for $\mathsf{V}^{n',k'}$ for some $k'\leq n'\leq n$. The constructed $\mathsf{P}^{n'*}$ interacts with $\mathsf{V}^{n',k'}$ by simulating the interaction of $\langle \mathsf{P}^{n*},\mathsf{V}^{n,k}\rangle$, where $\mathsf{P}^{n'*}$ plays P^{n*} and the first $n-n'$ subverifiers of $\mathsf{V}^{n,k}$ with certain fixed coins $c_1^*,\ldots,c_{n-n'}^*$.

  It is not hard to see that the correlation reduction CR can be implemented in two-phase puzzle systems as well, where $\mathbf{S}^*$ converts $\mathbf{S}^{n*}$ for $\mathbf{P}^{n,r}$ into $\mathbf{S}^{n'*}$ for $\mathbf{P}^{n',r'}$ for some $r'\leq n'\leq n$.

- Similarly, recall that P^* interacts with V by simulating the interaction of $\mathsf{P}^{n'}$ and $\mathsf{V}^{n',k'}$ with V embedded in the first coordinate of $\mathsf{V}^{n',k'}$, and when P^* receives message $v = v_1$ from V, P^* repeatedly samples coins $\vec{c}_{-1}$ of V_{-1} and checks if $\mathsf{P}^{n'*}$ convinces exactly $k' - 1$ of V_{-1}. Once P^* finds such coins, he sends P^{n*}'s answer to v_1 to V.

  Recall that in (n', r')-repetition, the puzzles are generated *sequentially*. $\mathbf{S}^*$ can first run $\mathbf{S}^{n'*}$ to interact with $\mathbf{P}$ and generate the actual puzzle $p = p_1$. Then $\mathbf{S}^*$ repeatedly samples coins $\vec{c}_{-1}$ of $\mathbf{P}_{-1}$, simulates $\langle \mathbf{S}^{n'*}, \mathbf{P}^{n',r'} \rangle$, and checks if $\mathbf{S}^{n'*}$ makes exactly $r' - 1$ of $\mathbf{P}_{-1}$ accept. Once $\mathbf{S}^*$ finds such coins, he sends $\mathbf{S}^{n'*}$'s answer to p_1 to $\mathbf{P}$.

It follows that Theorem 5.15 can be proved by exactly the same analysis of the same reduction presented in Section 4.2.

Digression – Application to Sequential Repetition for Computationally Sound Protocols As mentioned in the introduction, while it is believed that computational soundness behaves well under sequential repetition, it seems that only a direct product theorem is found in literature [6]. Here, we observe that sequential repetition of interactive protocols can be viewed as a degenerate case of repetition of two-phase puzzle system defined above. Indeed, we can view an interactive protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ as a two-phase puzzle system $\mathbf{P}$, where P plays the role of a solver $\mathbf{S}$, V play the role of puzzle generator in $\mathbf{P}$, and there is no puzzle solving phase (i.e., $\mathbf{P}$ decides to accept/reject only based on the "puzzle" generated in the puzzle generation phase). Recall that in a two-phase puzzle system, the repetition of puzzle generation phase is done sequentially, which corresponds to the sequential repetition of $\langle \mathsf{P}, \mathsf{V} \rangle$.

Therefore, Theorem 5.15 implies tight sequential repetition theorem for computationally sound protocols with threshold verifiers. In fact, the result of Holenstein and Schoenebeck [22] implies that tight sequential repetition theorem holds for computationally sound protocols with monotone verifiers.

5.1.3 Outline of Our Construction

In this section, we discuss our construction of efficient black-box security amplification for commitment schemes in the computational setting, where the security holds against PPT and active adversaries. We start by reviewing the previous construction of Halevi and Rabin [17], and then discuss its limitations and our improvement. The construction in [17] uses the following two transformations, each of which improves one property significantly without hurting the other property too much.

- **Secret-sharing transformation.** Let Com_0 be a bit commitment scheme, and $n \in \mathbb{N}$ be a parameter. The transformation gives a bit commitment scheme $\mathrm{Com} = (S, R)$ as follows. To commit to a bit $b \in \{0, 1\}$, S generates random

$b_1, b_2, \ldots b_n \in \{0,1\}$ such that $\bigoplus_{i\in[n]} b_i = b$, i.e. a secret sharing of b, and then uses Com_0 to sequentially commit to each b_i to R.

Intuitively, this transformation improves the hiding property, since an adversarial R^* needs to learn all bits $b_1, \ldots, b_n$ to recover b, but it hurts the binding property, since an adversarial S^* only needs to cheat on any single bit b_i to decommit in two ways. Indeed, Halevi and Rabin proved that if Com_0 is (p,q)-secure, then Com is $(p^n, 1-(1-q)^n)$-secure.[3]

- **Repetition transformation.** Let Com_0 be a bit commitment scheme, and $n \in \mathbb{N}$ be a parameter. The transformation gives a bit commitment scheme $\mathrm{Com} = (S, R)$ as follows. To commit a bit $b \in \{0,1\}$ to R, S sequentially uses Com_0 n times to commit to the same bit b to R.

  Intuitively, this transformation improves the binding property, since an adversarial S^* needs to cheat on all copies of Com_0 to decommit in two ways, but it hurts the hiding property, since an adversarial R^* can learn the bit b from any copy of the commitments. Indeed, Halevi and Rabin proved that if Com_0 is (p,q)-secure, then Com is $(1-(1-p)^n, q^n)$-secure.

Halevi and Rabin showed that, as long as p and q satisfy $p+q \le 1-1/\mathrm{polylog}(s)$, then given a (p,q)-secure (weak) bit commitment scheme Com_0, one can apply the above two transformations alternately to obtain a secure bit commitment scheme Com. To measure the efficiency, consider the case where both p and q are constants with $p+q<1$. Since improving either hiding or binding property from constant to negligible requires $\omega(\log s)$ invocations to Com_0, and the above transformations improve two properties *separately*, the construction of Halevi and Rabin requires at least $\omega(\log^2 s)$ black-box calls to Com_0.

Remark 5.16 Independent of our work, Holenstein and Schoenebeck [22] present a different construction that improves the result of Halevi and Rabin in the following sense. For any (p,q)-secure bit commitment scheme Com_0 with $p+q \le 1-1/\mathrm{poly}(s)$ (rather than $1-1/\mathrm{polylog}(s)$), their construction gives a secure bit commitment scheme Com using $\mathrm{poly}(s)$ black-box calls to Com_0. Their construction uses Valiant's monotone formula for majority [33] to improve both properties. However, a closer inspection shows that their construction is equivalent to applying the secret sharing transformation and a variant of repetition transformation (with the same effect on the parameters) alternately. Hence, in terms of the efficiency, their construction also requires at least $\omega(\log^2 s)$ black-box calls to amplify a (p,q)-secure weak commitment scheme with constant p and q to a secure one.

To bypass the $\omega(\log^2 s)$ barrier of the existing constructions, our main idea is to use error-correcting codes and randomness extractors to amplify both hiding and

[3] We omit the negligible slackness in the informal discussion.

binding properties *simultaneously*. For intuition, we give an informal description of our transformation first. Let us informally use $\mathrm{Com}_0(b)$ to denote a commitment of a bit b, and let $C : \{0,1\}^n \to \{0,1\}^{n'}$ be an error-correcting code with minimum distance at least $\delta \cdot n'$. Also, let $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^t$ a strong randomness extractor — Ext is a function such that for every source X over $\{0,1\}^n$ with sufficiently large (min-)entropy, the distribution $(U_d, \mathrm{Ext}(X, U_d))$, where U_d denotes a uniformly random d-bit string, is statistically close to uniform (i.e., Ext extracts randomness from X using seed U_d).

Our transformation uses Com_0, C and Ext to commit to a string $v \in \{0,1\}^t$ as follows (recall that we obtain string commitment schemes as opposed to bit commitment schemes of other existing constructions).

- **Commit Stage**: the sender S samples a message $m \in_R \{0,1\}^n$ uniformly at random, and sequentially commits to each bit of the codeword $C(m)$ using Com_0, which generates commitments

  $$\mathrm{Com}_0(C(m)) \stackrel{\mathrm{def}}{=} (\mathrm{Com}_0(C(m)_1), \ldots, \mathrm{Com}_0(C(m)_{n'})).$$

  Then S samples a uniform seed $z \in_R \{0,1\}^d$, and sends the seed z with $v \oplus \mathrm{Ext}(m, z)$ to the receiver R. In sum, the commitment is

  $$\mathrm{Com}(v) = (\mathrm{Com}_0(C(m)), z, v \oplus \mathrm{Ext}(m, z)).$$

- **Reveal Stage**: the sender S sends the value v, the message m and reveals each committed bit of $C(m)$ to R, who checks consistency and accepts or rejects accordingly.

Intuitively, the binding property is improved because for an adversarial sender S^* to cheat, S^* needs to decommit $C(m)$ into two valid codewords. Since the code C has good minimum distance, S^* needs to successfully cheat on at least $\delta \cdot n'$ committed bits out of n' commit bits. The q-binding property of Com_0 says that, for each committed bit, S^* can cheat with probability at most q. Thus, in expectation, S^* can cheat on only $q \cdot n'$ commit bits. If $q < (0.9)\delta$, the Chernoff bound suggests that S^* should be able to cheat on at least $\delta \cdot n'$ commit bits with only exponentially small probability in n'. On the other hand, the hiding property is improved because after seeing the commitments of $C(m)$, an adversarial receiver R^* has only partial information about m by the p-hiding property of Com_0. Thus, Ext extracts the remaining (computational) entropy from m, which is used to hide the value v. Ideally, when both p and q are constants, we can set both $n, n' = \omega(\log s)$ and commit to $\Omega(n)$-bit string. However, there are a few difficulties:

- First, although it is not difficult to formalize the above intuition in the information-theoretic setting, analyzing the above construction in the computational setting requires new ideas. We will discuss this issue in detail in Section 5.1.4.

- The second issue is that the above construction requires both p and q are small constants.[4] This is not a big issue, since we can apply the transformation of Halevi and Rabin [17] first to make both p and q sufficiently small using constant number of black box calls.

- Finally, another limitation of the above construction is that, we can only prove security of our construction in the "known-security" setting. Namely, we can only amplify the security to a fixed polynomial s^{-c} (using $O(\log s)$ black-box calls and committing to a $O(\log s)$-bit string) as opposed to an unspecified negligible function. The reason is that the reduction of our security proof for the hiding property is efficient only when $n' = O(\log s)$. A natural idea is to achieve the standard asymptotic security is to apply the transformations of Halevi and Rabin to amplify the security to negligible. This indeed works for bit-commitment scheme and blows the number of black-box calls by only a $\omega(1)$ factor. However, since the output of our construction is a string commitment scheme, we need to generalize the transformations of Halevi and Rabin to a string version. Fortunately, this can be done, and the details can be found in Section 5.1.5.

In sum, our efficient security amplification for commitment schemes consists of three steps: given a (p, q)-secure bit commitment scheme Com_0 with constants $p+q < 1$, (1) we first apply the transformations of Halevi and Rabin to obtain a (p', q')-secure bit commitment scheme Com_1 with sufficiently small constants p', q', which costs a constant number of black box calls, (2) we apply the above construction to obtain a (s^{-c}, s^{-c})-secure $O(\log s)$-bit string commitment scheme Com_2, which costs $O(\log s)$ black box calls, and (3) we apply a string version of the transformations of Halevi and Rabin [17] to obtain a secure $O(\log s)$-bit string commitment scheme Com_3, which costs $\omega(1)$ black box calls. The number of black-box calls multiply over steps, and hence the resulting Com_3 uses $\omega(\log s)$ black-box calls to Com_0.

We proceed to give a formal description of the above construction and its analysis in Section 5.1.4, and present a string version of the transformations of Halevi and Rabin used in the third step and prove Theorem 5.8 in Section 5.1.5.

---

[4]The reason for this limitation is a bit involved: The sender S commits to a random codeword $C(m)$ of length n', which consists of only n bits of entropy. Informally, the commitments $\mathrm{Com}_0(C(m))$ may leak $n' \cdot p$ bits of information. Hence, we need $n > n' \cdot p$ so that there is entropy left for extraction. So p is upper-bounded by the rate n/n' of the code. On the other hand, q is upper-bounded by the distance δ of the code. Thus both are bounded due to the rate-distance tradeoffs for binary error-correcting codes.

5.1.4 Efficient Security Amplification in the Known-Security Setting

In this section, we present a transformation that converts a (p, q)-secure bit commitment scheme Com_0 to a (s^{-c}, s^{-c})-secure $O(\log s)$-bit string commitment scheme Com using $O(\log s)$ black-box calls to Com_0, where c is an arbitrary constant. Our transformation uses error-correcting codes and randomness extractors to amplify both hiding and binding properties *simultaneously*. The transformation requires using a *systematic* code with good distance and the "Goldreich-Levin" extractor. We will discuss the reason when we discuss the proof of security below. A formal description of our transformation can be found in Figure 5.2.

We will show that if Com_0 is a (p, q)-secure bit commitment scheme for small constants p, q, then by setting $n, \ell, t = O(\log s)$ properly, the resulting string commitment scheme is (s^{-c}, s^{-c})-secure for some constant c. Note that both parties in Com run in time polynomial in n, ℓ, t, and the running time of Com_0, which is efficient. Formally, we prove the following theorem.

Theorem 5.17 *The following holds for all sufficiently small constants $p, q \in (0, 1)$, and $k = O(\log s)$: Suppose there exists a (p, q)-secure (weak) bit-commitment scheme Com_0, then there exists an efficient $(2^{-k}, 2^{-k})$-secure $t = \Omega(k)$-bit string-commitment scheme* Com *that makes $O(k)$ black-box calls to* Com_0*. Specifically,* $\mathrm{Com} = \mathcal{T}(\mathrm{Com}_0, n, \ell, t)$ *for appropriate chosen $n, \ell = O(k)$, and $t = \Omega(k)$.*

We proceed to formalize the aforementioned intuition to analyze the hiding and binding properties.

Analysis of the Binding Property

We first recall the intuition of why the binding property is improved with a bit more detail. Recall that in the construction, the sender S is supposed to commit to each bit of a valid codeword $C(m) = (m, Am)$ using Com_0, where C is a random linear code chosen by the receiver R. By Lemma 5.10, C has good min-distance $\delta \cdot n$ with overwhelming probability. For an adversarial sender S^* to cheat, S^* needs to decommit the $n + \ell$ commitments into two valid codewords $C(m_1), C(m_2)$, which means that S^* needs to successfully cheat on at least $\delta \cdot n$ commitments out of $n + \ell$ commitments. Intuitively, if breaking the binding property of each commitment were independent events with success probability at most q, and if $\delta \cdot n \geq (1.1) \cdot q \cdot (n + \ell)$, then by Chernoff bounds, the success probability of S^* should be exponentially small in n.

Of course, the events are not independent since S^* has chance to correlate his strategy for different instances. However, breaking the binding property of sequentially committed bits can be modeled as repetition of two-phase puzzle systems, and hence the above intuition follows by the the threshold repetition theorem (Theorem 5.15),

Transformation $\mathcal{T}(\mathrm{Com}_0, n, \ell, t)$:

- **Inputs.** A bit commitment scheme Com_0, and parameters $n, \ell, t \in \mathbb{N}$.
- **Outputs.** A t-bit string-commitment scheme $\mathrm{Com} = (S, R)$ defined as follows.
- **Commit Stage.** Let $v \in \{0,1\}^t$ be the string to which S is committing to.
  1. R samples a uniformly random matrix $A \leftarrow \{0,1\}^{\ell \times n}$, and sends A to S.
     /* i.e., R selects a random systematic linear code $C(m) \stackrel{\text{def}}{=} (m, Am)$. */
  2. S samples the following uniformly at random: a message $m \leftarrow \{0,1\}^n$ and a matrix $Z \leftarrow \{0,1\}^{t \times n}$.
     /* Z is a random seed for the (strong) randomness extractor $\mathrm{Ext}(m, Z) \stackrel{\text{def}}{=} Zm$.*/
  3. S uses Com_0 to commit to each bit of m and each bit of Am to R sequentially. Let $\vec{x} = (x_1, \ldots, x_n)$ and $\vec{y} = (y_1, \ldots, y_\ell)$ denote the commitment of each bit respectively.
     /* i.e., S commits to each bit of the codeword $C(m)$. */
  4. S sends $(Z, v \oplus Zm)$ to R, where $v \oplus Zm$ is the bit-wise xor of v and Zm.
     /* i.e., S uses $\mathrm{Ext}(m, Z)$ as a one-time pad to hide the commit string v. */

  In sum, the commitment of v is $(A, \vec{x}, \vec{y}, Z, v \oplus Zm)$.
- **Reveal Stage.** S sends v and its coin tosses r to R, and R checks that v and r are consistent with the honest sender's algorithm.

Figure 5.2: Our black-box transformation $\mathcal{T}(\mathrm{Com}_0, n, \ell, t)$.

which says the success probability of S^* behaves the same as the case of independent events.

Formally, we prove the following lemma, which essentially says that when q is sufficiently smaller than the min-distance of the code, the binding property is amplified in an exponential rate. We formulate the lemma in concrete parameters for preciseness. For intuition, think of $n, \ell = \Theta(k)$, $k = O(\log s)$, and $T = s^{\omega(1)}$.

Lemma 5.18 (Binding) *Let d_0 be the universal constant in Lemma 5.10. There exists a universal constant c_1 such that the following holds.*

For any $q \in (0,1), n, k, \ell, t, T_0$, $T \in \mathbb{N}$ *satisfying (i)* $d_0 \cdot (3q) \cdot \log(1/3q) < 1$, *(ii)* $n \geq c_1 \cdot k/q$, *(iii)* $n > \ell \geq d_0 \cdot (3q) \cdot \log(1/3q) \cdot n$, *and (iv)* $n, t \leq \text{poly}(s)$, *if a bit-commitment scheme* $\text{Com}_0 = (S_0, R_0)$ *is* q*-binding against time* T, *then* $\text{Com} = \mathcal{T}(\text{Com}_0, n, \ell, t)$ *is* 2^{-k}*-binding against time* $T' = T/\text{poly}(s, 2^k)$.

Proof. Let S^* be a time T' cheating sender. We want to show that in the binding game,

$$\Pr[S^* \text{ succeeds}] \leq 2^{-k}.$$

Recall that in the binding game, the honest receiver R first sends a random 0-1 matrix A to S^*, which defines a systematic linear code $C : \{0,1\}^n \to \{0,1\}^{n+\ell}$ by $C(m) = (m, Am)$, and then (S^*, R) is supposed to use Com_0 $(n+\ell)$ times to commit each bit of a random codeword $C(m)$. For S^* to win the game, he needs to decommit the $(n+\ell)$ bits into two valid codewords in C. We use

$$\begin{aligned}&\Pr[S^* \text{ succeeds}]\\ &\leq \Pr[C \text{ has min-distance} < \delta n] + \Pr[S^* \text{ succeeds } \wedge \ C \text{ has min-distance} \geq \delta n],\end{aligned}$$

and upper bound both probabilities.

First we want to apply Lemma 5.10 to say that C is a good code with high probability. Let d_0, d_1 be the constants in the Lemma 5.10, $\delta = 3q$, and $\gamma = d_0 \delta \log(1/\delta)$. We set $c_1 > 3d_1$ so that $n \geq d_1 \cdot k/\delta$. By Lemma 5.10, we have $\Pr[C \text{ has min-distance} < \delta n] \leq 2^{-k}/2$.

Then we want to upper bound the second probability by $2^{-k}/2$. Suppose to the contrary that

$$\Pr[S^* \text{ succeeds } \wedge \ C \text{ has min-distance} \geq \delta n] > 2^{-k}/2.$$

As argued in Section 5.1.2, we can view breaking the binding property of Com_0 as solving a two-phase puzzle system $\mathbf{P} = (G, V)$, where G plays the role of R_0, and V verifies if both two decommitments are valid. It follows that the above events implies S^* solves at least δn out of $n + \ell$ puzzles, i.e., succeeds in the $(n + \ell, \delta n)$-repetition $\mathbf{P}^{n+\ell,\delta n} = (G^{n+\ell}, V^{n+\ell,\delta n})$. In other words, S^* is a time T' solver $\mathbf{S}^{n*}$ for $\mathbf{P}^{n+\ell,\delta n}$ with success probability at least $2^{-k}/2$.

By Theorem 5.15 (with parameter $\xi = 2^{-k}/4$), there is a solver $\mathbf{S}^*$ for $\mathbf{P}$ with success probability at least δ', provided δ' satisfies $P(n+\ell, \delta n, \delta') \leq 2^{-k}/4$, and $\mathbf{S}^*$ runs in time $\text{poly}(s, n+\ell, (2^{-k}/4)^{-1}) \cdot T'$ (since $\mathbf{S}^*$ makes at most $\text{poly}(s, n+\ell, (2^{-k}/4)^{-1})$ oracle calls to $\mathbf{S}^{n*}$, each of which can be simulated in time T'). In other words, there exists a S_0^* that breaks the binding property of Com_0 with probability δ' with the above runtime.

Recall that $\delta = 3q$, and $n > \ell$, we have $\delta n = 3qn > 1.5q(n + \ell)$. By a standard Chernoff bound, we have

$$P(n + \ell, \delta n, 1.2q) \leq 2^{-\delta n/c}$$

for some constant c that is independent of q and k. Thus, we can set $c_1 = \max\{3d_1, 6c\}$ so that $n \geq c_1 \cdot k/q$ and $n > \ell$ implies $2^{-\delta n/c} \leq 2^{-k}/4$. Hence, we may set $\delta' = 1.2q$ to obtain a solver $\mathbf{S}^*$ for $\mathbf{P}$ with success probability at least q and $\mathbf{S}^*$ runs in time $\text{poly}(s, 2^k) \cdot T'$. This contradicts the fact that Com_0 is q-binding against time $T = T' \cdot \text{poly}(2^k, T_0)$.

Since both probabilities are at most $2^{-k}/2$, we have $\Pr[S^* \text{ succeed}] \leq 2^{-k}$, as desired. ■

Analysis of the Hiding Property

We first recall the intuitive entropy argument of why the hiding property is improved with a bit more detail. Recall that in the construction, the sender S samples a random n-bit message m, which contains n bits of entropy. Then S commits to each bit of the codeword $C(m) = (m, Am)$, each of them leaking a little ($\approx p$ bits) of information about m. Intuitively, if we set the parameters so that there is (computational) entropy left in m, S can use randomness extractor to extract a string $\text{Ext}(x, Z)$ that is (pseudo-)random from an adversarial receiver R^*'s point of view, and use it as one-time-pad to hide the commit value v.

To formalize this intuition in the computational setting, we argue that it is very hard for R^* to predict the whole message m after he sees the $n + \ell$ commitments, and hence one can apply the Goldreich-Levin theorem to extract pseudo-random bits. This is why our transformation requires to use the Goldreich-Levin extractor. To argue that m is hard to predict from the commitments $(\vec{x}, \vec{y})$, we first argue that m is hard to predict from $\vec{x}$. We can view predicting n sequentially committed message bits of m from the commitments $\vec{x}$ as n-fold direct product of a two-phase puzzle system. By the threshold repetition theorem (Theorem 5.15), the success probability of R^* is at most $((1 + p)/2)^n$ (up to a negligible term). Observing that $\vec{y}$ contains at most ℓ bits of information about m, the success probability of R^* to predict m from $(\vec{x}, \vec{y})$ is at most $2^\ell \cdot ((1 + p)/2)^n$. Hence, by the Goldreich-Levin theorem, we can extract $\Omega(\log(2^\ell \cdot ((1 + p)/2)^n))$ pseudorandom bits.

Formally, we prove the following lemma, which essentially says that we can extract $\Omega(\log(2^\ell \cdot ((1+p)/2)^n))$ pseudorandom bits. Again, we formulate the lemma in concrete parameters for preciseness, and we use parameter $\alpha = 1 - p$ for clarity. For intuition, think of $n, \ell = \Theta(k)$, $k = O(\log s)$ and $T = s^{\omega(1)}$.

Lemma 5.19 (Hiding) *There exists a universal constant c_2 such that the following holds. For every $\alpha \in (0, 1), n, k, \ell, t, T_0, T \in \mathbb{N}$ satisfying (i) $2c_2 \cdot k/\alpha \geq n \geq c_2 \cdot k/\alpha$ and (ii) $\ell, t \leq \alpha n/12 \leq \text{poly}(s)$, if $\text{Com}_0 = (S_0, R_0)$ is a $(1 - \alpha)$-hiding against time T, then $\text{Com} = \mathcal{T}(\text{Com}_0, n, \ell, t)$ is 2^{-k}-hiding against time $T' = T/\text{poly}(s, 2^k)$.*

Proof. We prove the contrapositive statement. Suppose Com is not 2^{-k}-hiding against time T', then there exists a time T' cheating receiver R^*, and a time T'

distinguisher D such that

$$|\Pr[D(\text{view}_{R^*}(S(U_t), R^*)(1^k), U_t) = 1] - \Pr[D(\text{view}_{R^*}(S(U_t), R^*)(1^k), U'_t) = 1]| > 2^{-k}$$

Let us understand the view of R^* better. In the commit stage, R^* tosses some coins r, sends some 0-1 matrix A to S, and reaches some configuration σ. We can assume without loss of generality that σ contains r and A. Next, the honest sender S plays the role of S_0 in Com_0, and commits to n random bits $m \leftarrow \{0,1\}^n$, and ℓ parity bits Am. Again, let $C : \{0,1\}^n \to \{0,1\}^{n+\ell}$ be the linear code defined by $C(m) = (m, Am)$. In each interaction $i = 1, \ldots, n+\ell$, R^* plays a cheating receiver $R^*_{0,i}$, and gets a view $x_i \stackrel{\text{def}}{=} \text{view}_{R^*_{0,i}}(S_0(C(m)_i), R^*_{0,i})$. Let $\vec{x} = (x_1, \ldots, x_{n+\ell})$. Finally, R^* receives a random matrix Z, and $s \oplus Zm$, where s is the string that S commits to. In sum, the view of R^* in $(S(s), R^*)(1^k)$ can be described by $(\sigma, \vec{x}, Z, s \oplus Zm)$. Thus, we have,

$$|\Pr[D((\sigma, \vec{x}, Z, U_t \oplus Zm), U_t) = 1] - \Pr[D((\sigma, \vec{x}, Z, U_t \oplus Zm), U'_t) = 1]| > 2^{-k}$$

This implies the existence of time $T' + O(t)$ distinguisher D' such that[5]

$$|\Pr[D'((\sigma, \vec{x}, Z, Zm) = 1] - \Pr[D'((\sigma, \vec{x}, Z, U_t) = 1]| > 2^{-k}$$

Let $Z = (z_1, \ldots, z_t)$, where each z_i is a row of Z. We can write Zm as $(z_1 \cdot m, \ldots, z_t \cdot m)$. By the equivalence of pseudorandomness and next-bit unpredictability, there is a time $T' + O(t)$ next-bit-predictor P such that

$$\Pr[P(\sigma, \vec{x}, Z, z_1 \cdot m, \ldots, z_{i-1} \cdot m) = z_i \cdot m] > 1/2 + 2^{-k}/t$$

where the probability is also taken on a random choice of $i \in [t]$.

For convenience, we let $Z_{-i} = (z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_t)$, and write $(\sigma, \vec{x}, Z, z_1 \cdot m, \ldots, z_{i-1} \cdot m)$ as $(\sigma, \vec{x}, Z_{-i}, z_1 \cdot m, \ldots, z_{i-1} \cdot m, z_i)$ (i.e., move z_i to the last coordinate). By a Markov argument, with probability at least $2^{-k}/2t$ over random $(i, \sigma, \vec{x}, Z_{-i}, z_1 \cdot m, \ldots, z_{i-1} \cdot m)$,

$$\Pr_{z_i}[P(\sigma, \vec{x}, Z_{-i}, z_1 \cdot m, \ldots, z_{i-1} \cdot m, z_i) = z_i \cdot m] > 1/2 + 2^{-k}/2t$$

We can view $P(\sigma, \vec{x}, Z_{-i}, z_1 \cdot m, \ldots, z_{i-1} \cdot m, \cdot)$ as a corrupted Hadamard encoding of m. By the Goldreich-Levin Theorem (Lemma 5.11), if

$$\Pr_{z_i}[P(\sigma, \vec{x}, Z_{-i}, z_1 \cdot m, \ldots, z_{i-1} \cdot m, z_i) = z_i \cdot m] > 1/2 + 2^{-k}/2t,$$

---

[5] On input $(\sigma, \vec{x}, Z, a)$, D' simply samples a fresh copy of uniform bits U'_t, and feeds $((\sigma, \vec{x}, Z, U'_t \oplus a), U'_t)$ to D. If a is drawn from Zm, then D gets distribution $((\sigma, \vec{x}, Z, U'_t \oplus Zm), U'_t)$, and if a is drawn from U_t, then D gets $((\sigma, \vec{x}, Z, U'_t \oplus U_t), U'_t) = ((\sigma, \vec{x}, Z, U_t \oplus Zm), U'_t)$.

we can make $O(n \cdot t^2 \cdot 2^{2k})$ queries to $P(\sigma, \vec{x}, Z_{-i}, z_1 \cdot m, \ldots, z_{i-1} \cdot m, \cdot)$ and guess m correctly with probability $\Omega((2^{-k}/t)^2)$. Therefore, there exists a time $(T' + O(t)) \cdot O(n \cdot t^2 \cdot 2^{2k})$ algorithm B such that

$$\Pr[B(\sigma, \vec{x}, Z_{-i}, z_1 \cdot m, \ldots, z_{i-1} \cdot m) = m] \geq (2^{-k}/2t) \cdot \Omega((2^{-k}/t)^2) = \Omega((2^{-k}/t)^3)$$

Now, suppose we only get input σ and $x_1, \ldots, x_n$, we claim that we can still guess m correctly with probability at least $2^{-(\ell+t-1)} \cdot \Omega(2^{-3k}/t^3)$. The idea is to generate the rest of the input $(x_{n+1}, \ldots, x_{n+\ell}, Z_{-i}, z_1 \cdot m, \ldots, z_{i-1} \cdot m)$ with correct distribution, and feed it to B. Observe that $x_{n+1}, \ldots, x_{n+\ell}$ are generated by the interaction of a honest S, who plays the role of S_0 to commit each bit of (Am), and a cheating receiver R^*, who has the view $(\sigma, x_1, \ldots, x_n)$ and plays a cheating sender $R^*_{0,i}$. Since we have the view $(\sigma, x_1, \ldots, x_n)$ of R^*, if we can guess (Am) correctly, then we can simulate the interaction of S and R^*, and generate the correct distribution of $(x_{n+1}, \ldots, x_{n+\ell})$ in time $T' \cdot \mathrm{poly}(s, \ell)$. Finally, we can simply guess the value of $(z_1 \cdot m, \ldots, z_{i-1} \cdot m)$, which is at most $t - 1$ bits. In sum, if we can guess the value of (Am) and $(z_1 \cdot m, \ldots, z_{i-1} \cdot m)$ correctly, then we can generate the correct distribution of B's input $(\sigma, \vec{x}, Z_{-i}, z_1 \cdot m, \ldots, z_{i-1} \cdot m)$ in time $T' \cdot \mathrm{poly}(s, \ell)$. Since we only need to guess at most $(\ell + t - 1)$ bits, we can guess it correctly with probability at least $2^{-(\ell+t-1)}$. Therefore, we have a time $(T' + O(t)) \cdot O(n \cdot t^2 \cdot 2^{2k}) + T' \cdot \mathrm{poly}(s, \ell) = T' \cdot \mathrm{poly}(s, 2^k)$ algorithm B' such that

$$\Pr[B'(\sigma, x_1, \ldots, x_n) = m] \geq 2^{-(\ell+t-1)} \cdot \Omega(2^{-3k}/t^3) = \Omega(2^{-(3k+\ell+t-1)}/t^3)$$

Next, we observe that breaking the weak hiding property of Com_0 can also be viewed as a two-phase puzzle system $\mathbf{P}$, where the puzzle generator plays the role of the honest sender S_0 who commits to a random bit b, and the solver $\mathbf{S}^*$ is required to guess the commit bit b correctly. Note that in Com, the first n bits to which S commits are independent random bits m, we can view the combination of R^*, who interacts with S and generates $(\sigma, x_1, \ldots, x_n)$, and B', who takes $(\sigma, x_1, \ldots, x_n)$ as input and guesses m, as a solver $\mathbf{S}^{n*}$ for the (n, n)-repetition $\mathbf{P}^{n,n}$. Furthermore, the above inequality says that the success probability of $\mathbf{S}^{n*}$ is at least $\Omega(2^{-(3k+\ell+t-1)}/t^3)$, and the runtime of $\mathbf{S}^{n*}$ is $T' \cdot \mathrm{poly}(2^k, s)$. By Theorem 5.15 (with the slackness parameter $\xi = \Omega(2^{-(3k+\ell+t-1)}/t^3)$), there is a solver $\mathbf{S}^*$ for $\mathbf{P}$ with success probability at least δ', provided δ' satisfies $\delta'^n \leq \Omega(2^{-(3k+\ell+t-1)}/t^3)$, and $\mathbf{S}^*$ runs in time $\mathrm{poly}(s, n, (\delta')^{-n}) \cdot T' = T' \cdot \mathrm{poly}(s, 2^k)$. In other words, there exists a R^*_0 with runtime $T' \cdot \mathrm{poly}(s, 2^k)$ that guesses a random committed bit correctly with probability at least δ'.

By Remark 5.4, Com_0 being $(1 - \alpha)$-hiding against time T means that for every time T cheating receiver R^*_0, and time T predictor P,

$$\Pr[P(\mathrm{view}_{R^*_0}(S_0(U_1), R^*_0)(1^k)) = U_1] \leq 1/2 + (1 - \alpha)/2 = 1 - \alpha/2.$$

To get a contradiction, we set $\delta' = (1 - \alpha/4)$, and set c_2 large enough so that the conditions (i) and (ii) imply

$$\Omega(2^{-(3k+\ell+t-1)}/t^4) \geq e^{-\alpha n/4} \geq (1 - \alpha/4)^n.$$

It follows that the R_0^* constructed above runs in time $T' \cdot \text{poly}(s, 2^k) < T$ such that

$$\Pr[P(\text{view}_{R_0^*}(S_0(U_1), R_0^*)(1^k)) = U_1] > 1 - \alpha/2,$$

a contradiction. ■

Finally, we use Lemma 5.18 and 5.19 with properly chosen parameters to prove Theorem 5.17.

Proof. (**of Theorem 5.17**) We set the parameters n, k, and ℓ as follows: $n = \max\{\frac{c_1 k}{q}, \frac{c_2 k}{1-p}\} = \Theta(k)$, $\ell = d_0(3q)\log(3q) \cdot n$, and $t = \frac{(1-p)n}{12} = \Omega(k)$, where c_1, c_2, d_0 are the constants in the Lemma 5.10, 5.18, and 5.19. The theorem follows directly from Lemma 5.18 and 5.19. ■

5.1.5 Security Amplification for String Commitment Schemes

In this section, we generalize the transformations of Halevi and Rabin [17] to the case of string commitment schemes, with the goal of amplifying the (s^{-c}, s^{-c})-secure string commitment scheme obtained from our transformation to achieve negligible security. This is a simpler task since the starting point is an almost secure commitment scheme, and this task can be done by applying a secret-sharing transformation first and then a repetition transformation. A formal description of the transformations can be found in Figure 5.3.

Secret-sharing $\mathcal{SS}(\text{Com}_0, u)$. Let Com_0 be a t-bit string commitment scheme, and $u \in \mathbb{N}$ be a parameter. The transformation gives a t-bit string commitment scheme $\text{Com} = (S, R)$ as follows. To commit a value $v \in \{0,1\}^t$ to R, S generates random $v_1, v_2, \ldots v_n \in \{0,1\}^t$ such that $v_1 \oplus v_2 \oplus \cdots \oplus v_u = v$, where $\oplus$ denotes the bit-wise xor of v_i's (i.e. a secret sharing of v), and then uses Com_0 to sequentially commit to each v_i to R.

Repetition $\mathcal{R}(\text{Com}_0, u)$. Let Com_0 be a t-bit string commitment scheme, and $u \in \mathbb{N}$ be a parameter. The transformation gives a t-bit string commitment scheme $\text{Com} = (S, R)$ as follows. To commit a value $v \in \{0,1\}^t$ to R, S sequentially uses Com_0 u times to commit to the same value v to R.

Figure 5.3: Secret-sharing and repetition transformation for string commitment schemes.

We proceed to analyze the binding and hiding properties of the resulting commitment schemes of the two transformations. For the binding property, the analysis is essentially the same as in [17]: for repetition, it requires to break all u commitments of Com_0, and for secret-sharing, it requires to break only 1 out of u commitments of

Com_0, which can be modeled as solving corresponding repetition of two-phase puzzles. Theorem 5.15 (or the direct product and hardness degradation theorems of Halevi and Rabin [17]) implies the following lemma.

Lemma 5.20 ([17]) *Let* Com_0 *be a* t*-bit string-commitment scheme,* $u = u(s) \leq \mathrm{poly}(s)$ *a efficiently computable function, and* $q \in (0,1)$ *a constant. Suppose* Com_0 *is* q*-binding, then* $\mathcal{R}(\mathrm{Com}_0, u)$ *is* $(q^u + \mathsf{ngl})$*-binding, and* $\mathcal{SS}(\mathrm{Com}_0, u)$ *is* $(1-(1-q)^u + \mathsf{ngl})$*-binding.*

On the other hand, analyzing the hiding property is tricker. For the secret-sharing transformation, we need a string version of XOR Lemma to show that the hiding property is amplified. Maurer and Tessaro [27] proved a more general result (Theorem 2 of [27]) in the context of composing "random systems," which implies the following lemma.

Lemma 5.21 ([27]) *Let* Com_0 *be a* t*-bit string-commitment scheme, and* $\mathrm{Com} = \mathcal{SS}(\mathrm{Com}_0, u)$ *with efficiently computable* $u = u(s) \leq \mathrm{poly}(s)$*. If* Com_0 *is* p*-hiding, then* Com *is* $(p^u + \mathsf{ngl})$*-hiding.*

We next show that repetition transformation preserves the (negligible) hiding property. This is sufficient for our purpose since we will apply the secret-sharing transformation to amplify the hiding property to negligible before applying the repetition transformation.

Lemma 5.22 *Let* $\mathrm{Com}_0 = (S_0, R_0)$ *be a* t*-bit string-commitment scheme, and* $\mathrm{Com} = \mathcal{R}(\mathrm{Com}_0, u)$ *with efficiently computable* $u = u(s) \leq \mathrm{poly}(s)$*. If* Com_0 *is* ngl*-hiding, so is* Com*.*

Proof. Recall that the hiding property of Com_0 says that for every PPT adversarial receiver R_0^*, the distribution $(\mathrm{view}_{R_0^*}(S_0(U_t), R_0^*), U_t)$ and $(\mathrm{view}_{R_0^*}(S_0(U_t), R_0^*), U_t')$ are computationally indistinguishable, where U_t' is an independent copy of U_t. To simplify the notation, we use $\mathrm{Com}_0(U_t)$ to denote $\mathrm{view}_{R_0^*}(S_0(U_t), R_0^*)$, and so hiding property is represented by $(\mathrm{Com}_0(U_t), U_t) \approx_c (\mathrm{Com}_0(U_t), U_t')$. Also, by abusing the notation, we denote $\mathrm{Com}(U_t) = (\mathrm{Com}_0(U_t), \ldots, \mathrm{Com}_0(U_t))$. Our goal is to prove that if $(\mathrm{Com}_0(U_t), U_t) \approx_c (\mathrm{Com}_0(U_t), U_t')$, then $(\mathrm{Com}(U_t), U_t) \approx_c (\mathrm{Com}(U_t), U_t')$, i.e.,

$$(\mathrm{Com}_0(U_t), \ldots, \mathrm{Com}_0(U_t), U_t) \approx_c (\mathrm{Com}_0(U_t), \ldots, \mathrm{Com}_0(U_t), U_t').$$

Note that in the above distributions, the u copies of $\mathrm{Com}_0(U_t)$ are not independent, since implicitly, R^* can correlate different copies.

We will show that both distributions are computationally indistinguishable to

$$(\mathrm{Com}(U_t^1), \mathrm{Com}(U_t^2), \mathrm{Com}(U_t^3), \ldots, \mathrm{Com}(U_t^u), U_t),$$

where the U_t^i's are i.i.d. copies of U_t. This implies $(\mathrm{Com}(U_t), U_t) \approx_c (\mathrm{Com}(U_t), U_t')$.

Claim 5.23 $(\mathrm{Com}_0(U_t^1), \ldots, \mathrm{Com}_0(U_t^u), U_t) \approx_c (\mathrm{Com}_0(U_t), \ldots, \mathrm{Com}_0(U_t), U_t)$.

Proof of claim: We prove the claim by contradiction. Assume that there exists a PPT adversary R^* (who sequentially interacts with S_0) such that the distributions

$$(\mathrm{Com}_0(U_t^1), \ldots, \mathrm{Com}_0(U_t^u), U_t) \text{ and } (\mathrm{Com}_0(U_t), \ldots, \mathrm{Com}_0(U_t), U_t)$$

are distinguishable by a PPT distinguisher D^* with noticeable advantage ε. We show by a hybrid argument that there exist a PPT adversary R_0^* (interacting with S_0) and a PPT distinguisher D_0^* that distinguishes distributions

$$(\mathrm{view}_{R_0^*}(S_0(U_t), R_0^*), U_t) \text{ and } (\mathrm{view}_{R_0^*}(S_0(U_t'), R_0^*), U_t)$$

with probability ε/u.

The adversary R_0^* (interacting with either $S_0(U_t)$ or $S_0(U_t')$) is defined as follows. R_0^* selects a uniformly random coordinate $i \in [u]$ and simulates R^* internally. For $j = 1, 2, \ldots, i-1$, R_0^* internally simulates the interaction of R^* and $S_0(U_t^j)$ sequentially, where U_t^j's are i.i.d. copies of U_t. Then R_0^* interacts with the external S_0 by continuing running R^*.

We proceed to define the distinguisher D_0^*. Note that the view of R_0^* contains i, $\mathrm{Com}_0(U_t^1), \ldots, \mathrm{Com}_0(U_t^{i-1})$, either $\mathrm{Com}_0(U_t)$ or $\mathrm{Com}_0(U_t')$, and the state of R^* after i sequential interactions with S_0. Also note that D_0^* receives U_t as well. Hence, we can let D_0^* simulate the continuation of $u-i$ sequential interactions of R^* and $S_0(U_t)$, and generate either distribution

$$(\mathrm{Com}_0(U_t^1), \ldots, \mathrm{Com}_0(U_t^{i-1}), \mathrm{Com}_0(U_t), \mathrm{Com}_0(U_t), \ldots, \mathrm{Com}_0(U_t), U_t),$$

or distribution

$$(\mathrm{Com}_0(U_t^1), \ldots, \mathrm{Com}_0(U_t^{i-1}), \mathrm{Com}_0(U_t'), \mathrm{Com}_0(U_t), \ldots, \mathrm{Com}_0(U_t), U_t).$$

Then D_0^* runs D^* on the above distribution.

Now, for $i \in \{0, 1, \ldots, u\}$, we define hybrid distributions

$$H_i \stackrel{\mathrm{def}}{=} (\mathrm{Com}_0(U_t^1), \ldots, \mathrm{Com}_0(U_t^{i-1}), \mathrm{Com}_0(U_t^i), \mathrm{Com}_0(U_t), \ldots, \mathrm{Com}_0(U_t), U_t).$$

Observe that when R_0^* selects coordinate $i \in [u]$ and interacts with $S_0(U_t)$ (resp., $S_0(U_t')$), D_0^* feeds in D^* the hybrid H_{i-1} (resp., H_i). By a standard hybrid argument, D_0^* can distinguish

$$(\mathrm{view}_{R_0^*}(S_0(U_t), R_0^*), U_t) \text{ and } (\mathrm{view}_{R_0^*}(S_0(U_t'), R_0^*), U_t)$$

with probability ε/u. This contradicts the hiding property of Com_0 and completes the proof of the claim. □

Now, observe that the above claim also implies

$$(\mathrm{Com}(U_t), \ldots, \mathrm{Com}(U_t)) \approx_c (\mathrm{Com}(U_t^1), \ldots, \mathrm{Com}(U_t^u)),$$

which implies

$$(\mathrm{Com}_0(U_t), \ldots, \mathrm{Com}_0(U_t), U_t') \approx_c (\mathrm{Com}_0(U_t^1), \ldots, \mathrm{Com}_0(U_t^u), U_t').$$

Hence, we have

$$\begin{aligned} &(\mathrm{Com}_0(U_t), \ldots, \mathrm{Com}_0(U_t), U_t) \\ \approx_c \ &(\mathrm{Com}_0(U_t^1), \ldots, \mathrm{Com}_0(U_t^u), U_t) \\ \approx_c \ &(\mathrm{Com}_0(U_t), \ldots, \mathrm{Com}_0(U_t), U_t'), \end{aligned}$$

as desired. ■

Finally, we present a formal description of our final construction of efficient security amplification for commitment schemes in Figure 5.4, and complete the proof of Theorem 5.8.

Final Construction.

- **Inputs.** A (p, q)-secure bit commitment scheme Com_0 with $p + q < 1$.
- **Outputs.** A secure t-bit string-commitment scheme Com with $t = O(\log s)$.

1. Apply the transformations of Halevi and Rabin alternately to obtain a (p', q')-secure bit commitment scheme Com_1 with sufficiently small constants p', q'.
2. Apply our transformations $\mathcal{T}(\mathrm{Com}_1, n, \ell, t)$ to obtain a (s^{-c}, s^{-c})-secure t-bit string commitment scheme Com_2, where $n, \ell = O(\log s)$, and c is some constant.
3. Let $a = a(s)$ be an arbitrary $\omega(1)$ function. Apply $\mathcal{SS}(\mathrm{Com}_2, a)$ to obtain a $(\mathsf{ngl}, a \cdot s^{-c} + \mathsf{ngl})$-secure t-bit string commitment scheme Com_3.
4. Apply $\mathcal{R}(\mathrm{Com}_3, a)$ to obtain a secure t-bit string commitment scheme Com.

Figure 5.4: Efficient security amplification of commitment schemes.

Theorem 5.24 (Theorem 5.8 restated) *Let $p, q \in (0, 1)$ be constants with $p + q < 1$. Suppose there exists a (p, q)-secure bit commitment scheme Com_0. Then for every*

$t(s) \leq \text{poly}(s)$, $n(s) = \omega(t + \log s)$ *where* s *is the security parameter, there exists a secure* t*-bit string-commitment scheme* Com *that makes only* n *black-box call to* Com_0.

Proof. We first prove the theorem for the case where $t(s) = O(\log s)$. In this case, the desired commitment scheme Com is defined in Figure 5.4. The fact that Com is a secure string commitment scheme follows straightforwardly from Theorem 5.17 and Lemma 5.20, 5.21, 5.22. Observing the Com_1 makes $O(1)$ black-box calls to Com_0, Com_2 makes $O(\log s)$ black-box calls to Com_1, Com_3 makes $\omega(1)$ black-box calls to Com_2, and finally Com makes $\omega(1)$ black-box calls to Com_3, the total number of black-box calls that Com makes to Com_0 is $\omega(\log s)$, as desired.

For general $t(s) \leq \text{poly}(s)$, we simply divide the t-bit string into blocks of length $O(\log s)$, and use Com to commit to each block. Standard hybrid arguments show that the security of committing each block implies the security of committing the whole string. ■

5.2 Security Amplification for Dynamic Weakly Verifiable Puzzles

In this section, we briefly discuss the work of Dodis, Impagliazzo, Jaiswal, and Kabanets [7] on security amplification for "dynamic weakly verifiable puzzle systems," and our improved analysis to their corresponding Chernoff-type theorem.

In [7], Dodis et al. defined dynamic weakly verifiable puzzle systems to capture the security properties of several cryptographic primitives such as message authentication codes (MACs), signature schemes (SIGs), and pseudorandom functions (PRFs). They considered parallel repetition of dynamic WVPs, proved a Chernoff-type theorem for this model, and used it to amplify the security of the corresponding primitives efficiently. Our contribution is to improve the bound of their Chernoff-type theorem to almost match the corresponding information-theoretic bound. As a consequence, we improve the efficiency of security amplification for the corresponding cryptographic primitives.

We will not present a full analysis in this section, but only outline the analysis of Dodis et al. and discuss our improvement. Hence, we try to state the definitions of Dodis et al. verbatim for the convenience of reference, except for some small notational change to make it more compatible with the rest of this thesis.

5.2.1 Dynamic Weakly Verifiable Puzzle Systems

We proceed to introduce the dynamic weakly verifiable puzzle systems (dynamic WVPs, for short) of Dodis, Impagliazzo, Jaiswal, and Kabanets [7], which are a generalization of weakly verifiable puzzle systems of Canetti, Halevi, and Steiner [2]. It

is illustrative to motivate the model by considering the example of message authentication codes (MACs) under chosen-message attacks. Hence, we start by reviewing the definition of message authentication codes.

Message authentication codes are cryptographic primitives that allows two parties Alice and Bob, who share a key sk, to ensure the the integrity and authenticity of the messages they exchange. A MAC Π consists of two algorithms $\Pi = (\mathsf{Tag}, \mathsf{Ver})$. When Alice wants to send a message m to Bob, Alice also computes a tag $\sigma = \mathsf{Tag}_{\mathsf{sk}}(m)$ and sends (m, σ) to Bob, who can verify the integrity of the message by checking if $\mathsf{Ver}_{\mathsf{sk}}(m, \sigma) = 1$. We require both *completeness* and *security* properties for MACs. The completeness says that when Alice is honest, Bob should accept with probability 1. For security, one standard way to formalize security is to consider the following chosen message attack (CMA) game.

The game is played between a challenger C and an adversary A. First, C generates a uniformly random key sk. Then C plays the role of an oracle, which allows A to make an arbitrary number of Tag and Ver queries. Namely, A can either send a message m to C and get back $\mathsf{Tag}_{\mathsf{sk}}(m)$, or sends a message-tag pair (m, σ) to C and get back $\mathsf{Ver}_{\mathsf{sk}}(m, \sigma)$. A succeeds if A ever makes a Ver query (m^*, σ^*) such that $\mathsf{Ver}_{\mathsf{sk}}(m^*, \sigma^*) = 1$, but he never queried Tag on m^* before.

A MAC Π is *secure* (or *unforgeable under chosen message attack*) if no PPT adversary A can succeed with non-negligible probability. For weaker security, we say that Π is δ-*secure* if no PPT adversary A can succeed with probability higher than δ. Security amplification for MACs asks if we can convert a δ-secure MAC into a fully secure one. A natural way to do it is by repetition. Namely, we can consider a n-fold repetition $\Pi^n = (\mathsf{Tag}^n, \mathsf{Ver}^n)$ of Π, where Tag^n uses n copies of keys $\vec{\mathsf{sk}} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$, and on input a message m, outputs $\vec{\sigma} = \mathsf{Tag}^n_{\vec{\mathsf{sk}}}(m) = (\mathsf{Tag}_{\mathsf{sk}_1}(m), \ldots, \mathsf{Tag}_{\mathsf{sk}_n}(m))$. For verification, Ver^n can either accept if all $\mathsf{Ver}_{\mathsf{sk}_i}(m, \sigma_i) = 1$ for every $i \in [n]$, or accept if at least a certain threshold number of tags are valid.

We remark that the above security property of a MAC Π can be viewed as the soundness of a corresponding interactive protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ (with unspecified number of rounds), where P and V play the role of A and C, respectively. Furthermore, the security of Π^n corresponds to the soundness of the n-fold parallel repetition of $\langle \mathsf{P}, \mathsf{V} \rangle$. However, $\langle \mathsf{P}, \mathsf{V} \rangle$ is a private-coin protocol with more than four messages, and hence, by the negative example of [1, 31] presented in Section 1.1.1, a parallel repetition theorem is not available for analyzing the security of Π^n.

Instead, Dodis et al. [7] model the security of MACs as follows.

Definition 5.25 (Dynamic Weakly Verifiable Puzzles [7]) *A* dynamic weakly verifiable puzzle $\mathbf{P}$ *is defined by an efficiently samplable distribution $\mathcal{D}$ on pairs of strings (p, c), where w.l.o.g., c is a sequence of uniformly random coins. Unlike the case of weakly verifiable puzzles, the string p defines a set of puzzles, (p, q) for $q \in Q$ (for some set Q of indices). There is a* PPT *computable relation R that specifies which answers are solutions for which of these puzzles: $R(c, q, r) = 1$ iff response r is*

correct answer to puzzle q in the collection determined by c. Finally, there is also a PPT *computable* hint *function $H(c, q)$.*

The solver $\mathbf{S}$ *receives the string p from* $\mathbf{P}$*, and can make a number of queries to* $\mathbf{P}$*: query hint(q) asks for $H(c, q)$, the hint for puzzle number q; a verification query $V(q, r)$ asks whether $R(c, q, r) = 1$. The solver* $\mathbf{S}$ succeeds *if* $\mathbf{S}$ *makes an accepting verification query for a q where it has not previously made a hint query on q.*

It is not hard to see that dynamic WVPs captures the security of MACs. The coins c correspond to the key sk, and the string p is just an empty string for the case of MACs. The set Q are set of messages m. Finally, the functions H and R correspond to Tag and Ver, respectively. Dodis et al. argued that dynamic WVPs can be used to capture the security of signature schemes (SIGs), and pseudorandom functions (PRFs) as well.

As in Section 5.1 for two-phase puzzle systems, we say that $\mathbf{P}$ is *δ-hard* against time T if for every solver $\mathbf{S}$ with runtime at most T, the success probability of $\mathbf{S}$ is at most δ. We proceed to define parallel repetition of dynamic WVPs to capture n-fold repetition Π^n of a MAC Π.

Definition 5.26 (Parallel Repetition of Dynamic WVPs **[7])** *Given a dynamic* WVP $\mathbf{P}$ *with $\mathcal{D}, R, Q$ and H, and integers $n \geq k \geq 1$, its (n, k)-parallel repetition is a dynamic* WVP $\mathbf{P}^{n,k}$ *with the product distribution $\mathcal{D}^n$ producing n-tuples $(p_1, c_1), \ldots, (p_n, c_n)$. For a given n-tuple $\vec{c} = (c_1, \ldots, c_n)$ and a query $q \in Q$, the new hint function is $H^n(\vec{c}, q) = (H(c_1, q), \ldots, H(c_n, q))$. The new relation $R^{n,k}((c_1, \ldots, c_n), q, (r_1, \ldots, r_n))$ evaluates to true if there is a subset $S \subset [n]$ of size at least k such that $\wedge_{i \in S} R(c_i, q, r_i)$.*

A solver $\mathbf{S}^n$ *for the (n, k)-repetition* $\mathbf{P}^{n,k}$ *may ask hint queries hint$^n(q)$, getting $H^n(\vec{c}, q)$ as the answer. A verification query $V^n(q, \vec{r})$ asks if $R^{n,k}(\vec{c}, q, \vec{r}) = 1$, for an n-tuple $\vec{r} = (r_1, \ldots, r_n)$. We say that the solver succeeds if it makes an accepting verification query for a q where it has not previously made a hint query on q.*

Again, it is not hard to see that the (n, k)-repetition $\mathbf{P}^{n,k}$ of $\mathbf{P}$ corresponds to n-fold repetition $\Pi^{n,k}$ of a MAC Π, where the verification algorithm $\mathsf{V}^{n,k}$ accepts iff at least k out of n tags are valid. Dodis et al. [7] proved the following Chernoff-type theorem for dynamic WVPs, which says that if $\mathbf{P}$ is $(1 - \delta)$-hard, then $\mathbf{P}^{n,k}$ with $k = n \cdot (1 - (1 - \gamma)\delta)$ is $e^{-\Omega(\gamma^2 \delta n)}$-hard.

Theorem 5.27 (Chernoff-type Theorem for Dynamic WVPs **[7])** *Let $n, k \in \mathbb{N}$ and $\delta, \gamma \in (0, 1)$ be parameters such that $k = n \cdot (1 - (1 - \gamma)\delta)$. Let* $\mathbf{P}$ *be a dynamic* WVP *with runtime t' and* $\mathbf{P}^{n,k}$ *the corresponding (n, k)-repetition. Suppose there exists a time t solver* $\mathbf{S}^n$ *for* $\mathbf{P}^{n,k}$ *with success probability at least ε, where*

$$\varepsilon \geq (800/\gamma\delta) \cdot (h + v) \cdot e^{-\gamma^2 \delta n/40},$$

and h is the number of hint queries, and v the number of verification queries made by $\mathbf{S}^n$. Then there is a solver $\mathbf{S}$ for $\mathbf{P}$ with success probability at least $(1-\delta)$ and runtime $\mathrm{poly}(t, t', h, v, \varepsilon^{-1}, \log(1/\gamma\delta))$.[6]

In comparison, the information-theoretic Chernoff bounds (if applicable) gives a upper bound $e^{-\gamma^2\delta n/2}$ on the success probability, but the above theorem gives a bound $e^{-\gamma^2\delta n/40}$, which is suboptimal. We improve the bound to almost match the corresponding information theoretical bound. In particular, our bound improve the constant in the exponent to 2. The constant improvement could be significant, since it means that for the purpose of security amplification, we can reduce the number of repetitions by a multiplicative factor of 20 for achieving the desired security.

We will outline the analysis of Dodis et al. in next section, and discuss our improvement in Section 5.2.3.

5.2.2 Outline of the Analysis of Dodis et al. [7]

In this section, we outline the analysis of Dodis, Impagliazzo, Jaiswal, and Kabanets [7] for proving Theorem 5.31.

Dodis et al. proved the theorem by an efficient black-box reduction. They observed that the "soft-decision" reduction algorithm of Impagliazzo, Jaiswal, and Kabanets [24] for weakly verifiable puzzles can be generalized to the dynamic WVP setting. Recall we mentioned in Section 3.3.4 and 3.4.8 that the idea of soft-decision was first used by Bellare, Impagliazzo, and Naor [1] for proving a direct product theorem for three-message protocols, and was later used by several works [24, 7, 20].

For simplicity, let us assume $\mathbf{S}^n$ makes only one verification query in the following informal discussion. The idea of soft-decision is as follows. The soft-decision reduction solver $\mathbf{S}$ solves the puzzle $\mathbf{P}$ by simulating the given parallel solver $\mathbf{S}^n$ for solving $\mathbf{P}^{n,k}$. $\mathbf{S}$ embeds the real puzzle $\mathbf{P}$ in a random coordinate $i \in [n]$ of $\mathbf{P}^{n,k}$, and simulates $\mathbf{S}^n$, and the remaining $n-1$ puzzles $\mathbf{P}_{-i}$ by himself. More precise, after receiving $p = p_i$ from $\mathbf{P}$, $\mathbf{S}$ generates $(\vec{p}_{-i}, \vec{c}_{-i})$ of $\mathbf{P}_{-i}$, and simulates $\mathbf{S}^n$. To handle $\mathbf{S}^n$'s hint query $hint(q)$, $\mathbf{S}$ simply forwards $hint(q)$ to get $H(c_i, q)$, and prepares the answers of $\mathbf{P}_{-i}$ by himself. When $\mathbf{S}^n$ comes up with a solution and makes a verification query $V^n(q^*, \vec{r}^*)$, $\mathbf{S}$ uses "soft-decision" to decide whether to accept and forward the solution (q^*, r_i^*) to $\mathbf{P}$. Specifically, if the solution $(q^*, \vec{r}^*_{-i})$ solves at least k of $\mathbf{P}_{-i}$, then $\mathbf{S}$ accepts and forwards the solution to $\mathbf{P}$. Otherwise, $\mathbf{S}$ forwards the solution with probability ρ^{k-t}, where $\rho \in (0,1)$ is a parameter, and t is the number of puzzles $\mathbf{P}_{-i}$ that accept $(q^*, \vec{r}^*_{-i})$. Namely, the probability that $\mathbf{S}$ accepts the solution decays exponentially in

[6] We note that the parametrization of the above Chernoff-type theorem of Dodis et al. is slightly different from the parametrization we used in Section 4.1. In Section, we assume the soundness error of $\langle \mathsf{P}, \mathsf{V} \rangle$ is δ, and consider threshold $k = (1+\gamma)\delta n$. Here, Dodis et al. assume hardness of $\mathbf{P}$ is $(1-\delta)$, and consider threshold $k = n - (1-\gamma)\delta n = (1-(1-\gamma)\delta)n$. Nevertheless, in both parameter ranges, the standard Chernoff bounds give upper bounds $e^{-\gamma^2\delta n/2}$ on probability.

the number of internal puzzles not solved by $(q^*, \vec{r}^*_{-i})$. When $\mathbf{S}$ decides to not accept the solution (q^*, r_i^*), $\mathbf{S}$ restarts the whole process again. Namely, $\mathbf{S}$ selects another random coordinate in which to embed $\mathbf{P}$, generates new $n-1$ internal puzzles, and so on.

However, note that $\mathbf{S}$ only solves $\mathbf{P}$ if $\mathbf{S}$ can come up with a *fresh* solution (q, r), where $\mathbf{S}$ never queried $hint(q)$ before. In the above reduction, $\mathbf{S}$ may simulate $\mathbf{S}^n$ many times, each of which makes several hint queries, and the final solution (q^*, r_i^*) forwarded by $\mathbf{S}$ may not be fresh due to the hint queries made by previous simulation of $\mathbf{S}^n$. Dodis et al. use a Valiant-Vazirani type argument [34] to resolve this issue. Roughly, they use a hash function $hash$ to partition the set Q into a solution set $Q_0 = hash^{-1}(0)$, and a hint set $Q_1 = Q \backslash Q_0$, and modify the given parallel solver $\mathbf{S}^n$ to a solver $\tilde{\mathbf{S}}^n$ that only makes hint queries with $q \in Q_1$ and one verification query with $q \in Q_0$. More precisely, $\tilde{\mathbf{S}}^n$ simply runs $\mathbf{S}^n$ with the following modification:

- $\tilde{\mathbf{S}}^n$ terminates if $\mathbf{S}^n$ makes a hint query with $q \notin Q_1$.
- $\tilde{\mathbf{S}}^n$ only forwards the first verification query of $\mathbf{S}^n$ with $q \in Q_0$, and ignores all the remaining verification queries of $\mathbf{S}^n$ (i.e., simply returns 0 to $\mathbf{S}^n$ and continues to run $\mathbf{S}^n$).

Let ε be the success probability of $\mathbf{S}^n$ and h and v the number of hint and verification queries made by $\mathbf{S}^n$. They show that one can efficiently find a hash function $hash$ such that the modified solver $\tilde{\mathbf{S}}^n$ has success probability $\varepsilon/8(h+v)$. Note that the issue with the freshness of solutions goes away when the soft-decision reduction is applied to $\tilde{\mathbf{S}}$, which gives good success probability for solving $\mathbf{P}$.

Dodis et al. [7] formalized the above argument in the following two lemmas, and Theorem 5.31 follows straightforwardly by the lemmas. Let $n, k \in \mathbb{N}$ be parameters, $\mathbf{P}$ a dynamic WVP and $\mathbf{P}^{n,k}$ the (n,k)-repetition of $\mathbf{P}$.

Lemma 5.28 *Let $\mathbf{S}^n$ be a solver for $\mathbf{P}^{n,k}$ with success probability ε and runtime t that makes at most h hint queries and v verification queries. Then there is a probabilistic algorithm runs in time* $\mathrm{poly}(t, h, v, \varepsilon^{-1})$ *and with high probability outputs a hash function $hash$ such that the corresponding modified solver $\tilde{\mathbf{S}}^n$ has success probability at least $\varepsilon/8(h+v)$. Furthermore, $\tilde{\mathbf{S}}^n$ only makes one verification query.*

Lemma 5.29 *Let $\tilde{\mathbf{S}}^n$ be the modified solver of $\mathbf{S}^n$ as in the conclusion of Lemma 5.28 with respect to $\mathbf{P}^{n,k}$ and a hash function $hash$. If $\tilde{\mathbf{S}}^n$ has runtime t and success probability at least $\varepsilon' = (100/\gamma\delta) \cdot e^{-\gamma^2\delta n/40}$ for some $\gamma, \delta \in (0,1)$, then there is a probabilistic solver $\mathbf{S}$ for $\mathbf{P}$ with success probability at least $1-\delta$ and runtime* $\mathrm{poly}(t, t', \varepsilon'^{-1}, \log(1/\gamma\delta))$*, where t' is the runtime of $\mathbf{P}$.*

5.2.3 Our Improvement

As in the setting of two-phase puzzle systems in Section 5.1, we observe that not only the soft-decision reduction but also our reduction for three-message protocols presented in Section 4.2 can be implemented in the setting of dynamic WVPs, which gives bounds on the success probability that match the information-theoretic bounds. However, like the soft-decision reduction, there is an issue of "freshness" of the solution of $\mathbf{S}^n$, and hence, the reduction can only be applied to the modified solver $\tilde{\mathbf{S}}^n$ as before. This improves the bound of the second lemma (Lemma 5.29) of Dodis et al. to optimal.

Lemma 5.30 *Let $\tilde{\mathbf{S}}^n$ be the modified solver of $\mathbf{S}^n$ with respect to a hash function hash. If $\tilde{\mathbf{S}}^n$ has runtime t and success probability $\varepsilon' = P(n, k, (1-\delta)) + \xi$ for some $\delta, \xi \in (0,1)$, then there is a probabilistic solver $\mathbf{S}$ for $\mathbf{P}$ with success probability at least $(1-\delta) + (\xi/10n)$ and runtime $\mathrm{poly}(t, t', \varepsilon'^{-1})$, where t' is the runtime of $\mathbf{P}$.*

To prove Lemma 5.30, we verify that both the correlation reduction procedure in Figure 4.2 and the reduction prover strategy defined in Figure 4.3 can be implemented in this model. Then the lemma follows by exactly the same analysis presented in Section 4.2.

- Recall that in the correlation reduction CR, P^* does not interact with V at all. In CR, P^* simply runs the interaction $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle$ internally with different coins, and constructs a prover $\mathsf{P}^{n'*}$ for $\mathsf{V}^{n',k'}$ for some $k' \leq n' \leq n$. The constructed $\mathsf{P}^{n'*}$ interacts with $\mathsf{V}^{n',k'}$ by simulating the interaction of $\langle \mathsf{P}^{n*}, \mathsf{V}^{n,k} \rangle$, where $\mathsf{P}^{n'*}$ plays P^{n*} and the first $n - n'$ subverifiers of $\mathsf{V}^{n,k}$ with certain fixed coins $c^*_1, \ldots, c^*_{n-n'}$.

  It is not hard to see that the correlation reduction CR can be implemented in dynamic WVPs as well, where $\mathbf{S}$ converts $\tilde{\mathbf{S}}^n$ for $\mathbf{P}^{n,k}$ into $\tilde{\mathbf{S}}^{n'}$ for $\mathbf{P}^{n',k'}$ for some $k' \leq n' \leq n$. Note that the resulting $\tilde{\mathbf{S}}^{n'}$ preserves the partition structure of $Q = Q_0 \cup Q_1$, and makes only one verification query like $\tilde{\mathbf{S}}^n$.

- Similarly, recall that P^* interacts with V by simulating the interaction of $\mathsf{P}^{n'}$ and $\mathsf{V}^{n',k'}$ with V embedded in the first coordinate of $\mathsf{V}^{n',k'}$, and when P^* receives message $v = v_1$ from V, P^* repeatedly samples coins $\vec{c}_{-1}$ of V_{-1} and checks if $\mathsf{P}^{n'*}$ convinces exactly $k' - 1$ of V_{-1}.

  Upon receiving a string p from $\mathbf{P}$, $\mathbf{S}$ can also repeatedly sample $(\vec{c}_{-1}, \vec{p}_{-1})$ of $\mathbf{P}_{-1}$, simulate $\mathbf{S}^{n'}$ solving $\mathbf{P}^{n',k'}$ and forward the hint queries to $\mathbf{P} = \mathbf{P}_1$, and check if $\mathbf{S}^{n'}$ makes exactly $k' - 1$ of $\mathbf{P}_{-1}$ accept.

Combining our Lemma 5.30 and the first lemma (Lemma 5.28) of Dodis et al. [7], we improve the bound of Theorem 5.31.

Theorem 5.31 (Chernoff-type Theorem for Dynamic WVPs**)** *Let $n, k \in \mathbb{N}$ and $\delta, \gamma \in (0,1)$ be parameters such that $k = n \cdot (1-(1-\gamma)\delta)$. Let $\mathbf{P}$ be a dynamic* WVP *with runtime t' and $\mathbf{P}^{n,k}$ the corresponding (n,k)-repetition. Suppose there exists a time t solver $\mathbf{S}^n$ for $\mathbf{P}^{n,k}$ with success probability at least ε, where*

$$\varepsilon \geq 16(h+v) \cdot P(n,k,(1-\delta)),$$

and h is the number of hint queries, and v the number of verification queries made by $\mathbf{S}^n$. Then there is a solver $\mathbf{S}$ for $\mathbf{P}$ with success probability at least $(1-\delta)$ and runtime $\mathrm{poly}(t, t', h, v, \varepsilon^{-1})$.

In particular, note that by a standard Chernoff bound, $P(n,k,(1-\delta)) \leq e^{-\gamma^2 \delta n/2}$, we improve the constant in the exponent from 40 to 2.

Bibliography

[1] Mihir Bellare, Russell Impagliazzo, and Moni Naor. Does parallel repetition lower the error in computationally sound protocols? In *FOCS*, pages 374–383, 1997.

[2] Ran Canetti, Shai Halevi, and Michael Steiner. Hardness amplification of weakly verifiable puzzles. In *TCC*, pages 17–33, 2005.

[3] Kai-Min Chung and Feng-Hao Liu. Parallel repetition theorems for interactive arguments. In *TCC*, pages 19–36, 2010.

[4] Kai-Min Chung, Feng-Hao Liu, Chi-Jen Lu, and Bo-Yin Yang. Efficient string-commitment from weak bit-commitment. In Masayuki Abe, editor, *Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT '10)*. Springer-Verlag, 5-9 December 2010. To appear.

[5] Ivan Damgård, Joe Kilian, and Louis Salvail. On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In *EUROCRYPT*, pages 56–73, 1999.

[6] Ivan Damgård and Birgit Pfitzmann. Sequential iteration of interactive arguments and an efficient zero-knowledge argument for np. In Kim Guldstrand Larsen, Sven Skyum, and Glynn Winskel, editors, *ICALP*, volume 1443 of *Lecture Notes in Computer Science*, pages 772–783. Springer, 1998.

[7] Yevgeniy Dodis, Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. Security amplification for interactivecryptographic primitives. In *TCC*, pages 128–145, 2009.

[8] Richard Durrett. *Probability: Theorey and Examples. Third Edition.* Duxbury, 2004.

[9] Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing encryption schemes from decryption errors. In *EUROCRYPT*, pages 342–360, 2004.

[10] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 465–482. Springer, 2010.

[11] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.

[12] Oded Goldreich. *Modern Cryptography, Probabilistic Proofs, and Pseudorandomness*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.

[13] Oded Goldreich. *Foundations of Cryptography. Basic tools*. Cambridge University Press, 2001.

[14] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32, 1989.

[15] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In *FOCS*, pages 174–187, 1986.

[16] Iftach Haitner. A parallel repetition theorem for any interactive argument. In *FOCS*, 2009.

[17] Shai Halevi and Tal Rabin. Degradation and amplification of computational hardness. In *TCC*, pages 626–643, 2008.

[18] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[19] Johan Håstad, Rafael Pass, Krzysztof Pietrzak, and Douglas Wikström. An efficient parallel repetition theorem. Unpublished manuscript, 2008.

[20] Johan Håstad, Rafael Pass, Douglas Wikström, and Krzysztof Pietrzak. An efficient parallel repetition theorem. In Micciancio [28], pages 1–18.

[21] Thomas Holenstein and Renato Renner. One-way secret-key agreement and applications to circuit polarization and immunization of public-key encryption. In *CRYPTO*, pages 478–493, 2005.

[22] Thomas Holenstein and Grant Schoenebeck. General hardness amplification of predicates and puzzles. *CoRR*, abs/1002.3534, 2010.

[23] Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. Chernoff-type direct product theorems. In *CRYPTO*, pages 500–516, 2007.

[24] Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. Chernoff-type direct product theorems. *J. Cryptology*, 22(1):75–92, 2009.

[25] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *FOCS*, pages 230–235, 1989.

[26] Charanjit S. Jutla. Almost optimal bounds for direct product threshold theorem. In Micciancio [28], pages 37–51.

[27] Ueli Maurer and Stefano Tessaro. Computational indistinguishability amplification: Tight product theorems for system composition. In Shai Halevi, editor, *Advances in Cryptology — CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 350–368. Springer-Verlag, August 2009.

[28] Daniele Micciancio, editor. *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, volume 5978 of *Lecture Notes in Computer Science*. Springer, 2010.

[29] Moni Naor. Bit commitment using pseudo-randomness. In *CRYPTO*, pages 128–136, 1989.

[30] Rafael Pass and Muthuramakrishnan Venkitasubramaniam. An efficient parallel repetition theorem for arthur-merlin games. In *STOC*, pages 420–429, 2007.

[31] Krzysztof Pietrzak and Douglas Wikström. Parallel repetition of computationally sound protocols revisited. In *TCC*, pages 86–102, 2007.

[32] Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998.

[33] Leslie G. Valiant. Short monotone formulae for the majority function. *J. Algorithms*, 5(3):363–366, 1984.

[34] Leslie G. Valiant and Vijay V. Vazirani. Np is as easy as detecting unique solutions. *Theor. Comput. Sci.*, 47(3):85–93, 1986.

[35] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2010.

[36] Douglas Wikström. An efficient concurrent repetition theorem. Cryptology ePrint Archive, Report 2009/347, 2009.

[37] Jürg Wullschleger. Oblivious-transfer amplification. In *EUROCRYPT*, pages 555–572, 2007.

[38] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *FOCS*, pages 80–91, 1982.