Efficient String-Commitment from Weak Bit-Commitment and Full-Spectrum Theorem for Puzzles

Kai-Min Chung*(kmchung@fas.harvard.edu), Feng-Hao Liu (fenghao@cs.brown.edu), Chi-Jen Lu (cjlu@iis.sinica.edu.tw), Bo-Yin Yang (by@crypto.tw)

(Preliminary Version)
December 26, 2009

Abstract

We study security amplification for weak bit-commitment schemes and improve the efficiency of (black-box) transformations in both the information-theoretic and computational settings. Let Com_0 be a (weak) bit-commitment scheme that is *p-hiding* in the sense that no cheating receiver can guess the committed bit correctly with probability better than $(1+p)/2$, and *q-binding* in the sense that no cheating sender can open in two ways with probability better than q, for some constants p, q with $p + q < 1$. The task is to transform Com_0 *efficiently* to a commitment scheme Com that is 2^{-k}-hiding and 2^{-k}-binding, where the efficiency is measured by the number of black-box calls to Com_0.

Our transformation uses only $O(k)$ calls to Com_0 and moreover, we can commit to an $\Omega(k)$-bit string instead of one bit. These results improve on previous work of Damgård et al. [DKS99] and Halevi and Rabin [HR08], whose transformations require $\Omega(k^2)$ black-box calls to Com_0 and commit to only one bit. To obtain our efficiency improvements, we use error-correcting codes and randomness extractors. Similar methods have previously been applied to information-theoretic settings or computational but non-interactive settings.

Our main technical contribution is to carry out the analysis in the *interactive and computational* setting of commitment schemes. In particular, we prove a "Full-Spectrum Theorem" for puzzle systems which says that the hardness of solving at least r puzzles out of n puzzles, where each puzzle can be solved with probability at most δ, amplifies/degrades at the essentially optimal, information-theoretic rate, namely, the probability that n independent Bernoulli random variables with expectation δ have sum at least r. (Independently, Holenstein and Schoenebeck [HS09] obtained similar results about amplification of puzzle systems.)

On the other hand, we provide a way to extract computational entropy in an interactive setting. It is known that in a non-interactive setting, one can extract many bits of computational entropy out using Goldreich-Levin theorem [GL89]. By applying the Halevi-Rabin [HR08] Direct Product Theorem of "sequentially" interactive weakly verifiable puzzles, we carry out the analysis in the interactive setting.

Keywords: commitment schemes, puzzles, computational hardness, hardness amplification, reductions, interactive proofs, entropy, extractors.

*Supported by US-Israel BSF grant 2006060 and NSF grant CNS-0831289.

1 Introduction

Security amplification for weak cryptographic primitives is a basic question that has been studied since the seminal work of Yao [Yao82]. This question has been extensively studied in recent years for a variety of primitives in various settings. For example, amplification has been studied for encryption schemes [DNR04, HR05], commitment schemes [DKS99, HR08], oblivious transfer [DKS99, Wul07] and others, message authentication codes (MACs) [DIJK09], digital signatures, and pseudorandom functions [DIJK09]. Some of these works consider information-theoretic security (e.g., [DKS99]), and others consider computational security. In addition, the various primitives present different interactive settings, which require new proof techniques. For example, commitment schemes are more interactive than encryption schemes, and the chosen-message-attack for MACs introduces a different type of interaction. Proving amplification results is more challenging in an interactive and computational setting.

In this paper, we continue the study of security amplification for commitment schemes in both the information-theoretic and computational settings, which was previously studied in [DKS99, Wul07, HR08]. Suppose a (weak) bit-commitment scheme Com_0 is *p-hiding* in the sense that no cheating receiver can guess the committed bit correctly with probability better than $(1+p)/2$, and *q-binding* in the sense that no cheating sender can open in two ways with probability better than q, and we wish to transform Com_0 to a *secure* commitment scheme Com that is ngl-hiding and ngl-binding, where ngl represents a negligible function of the security parameter. Previous works asked how weak can Com_0 be while such a transformation is possible? In the information-theoretic setting, Damgård, Kilian and Salvail [DKS99] showed that a black-box transformation is possible if and only if $p+q \leq 1-1/\mathrm{poly}(s)$, where s is the security parameter. In the computational setting, one way to do it is to construct a one-way function first [IL89], which can also be done provided $p+q \leq 1-1/\mathrm{poly}(s)$, and then construct a secure commit scheme from the one-way function [Nao89, HILL99]. However, this construction is indirect and very inefficient. Thus, Halevi and Rabin [HR08] analyzed the transformation of [DKS99] in the computational setting and obtained a black-box transformation whenever $p+q \leq 1-1/\mathrm{polylog}(s)$.

Another natural issue to consider is the efficiency of such transformations, which can be measured by the number of black-box calls to the weak commitment scheme Com_0. More concretely:

> **Main question:** Let Com_0 be a p-hiding and q-binding bit commitment scheme against time $2^{\Omega(k)}$ for some constants p, q with $p+q<1$ and some parameter k. How many black-box calls to Com_0 are required to construct a (string) commitment scheme Com that is 2^{-k}-hiding and 2^{-k}-binding against time $2^{\Omega(k)}$? How many bits can Com commit to?

The transformation of Damgård et al. [DKS99] and Halevi and Rabin [HR08] works by composing two transformations alternately — a "secret sharing" transformation, which amplifies hiding ($p \mapsto p^n$, where n is the number of "shares") and degrades binding ($q \mapsto (1-(1-q)^n)$), and a "repetition" transformation, which amplifies binding ($q \mapsto q^n$, where n is the number of "repetition") and degrades hiding ($p \mapsto (1-(1-p)^n)$), up to negligible terms. The number of shares and repetitions correspond to the number of black-box calls to Com_0, and composition multiplies these numbers. Observing that each transformation only improves one property, and it takes an $\Omega(k)$ number of shares (resp., repetitions) to improve the hiding (resp., binding) property from constant to 2^{-k}, their transformation requires at least $\Omega(k^2)$ black-box calls to Com_0.[1] Moreover,

[1] Both [DKS99] and [HR08] did not optimize the efficiency of the transformation. In the information-theoretic

the resulting commitment scheme Com commits to only one bit. The transformation of [Wul07] requires the use of the Goldreich-Micali-Wigderson compiler [GMW86], which is non-black-box and inefficient.

1.1 Our Result and Contribution

We give a transformation that commits to a $\Omega(k)$-bit string using only $O(k)$ black-box calls to Com_0 and achieves 2^{-k} security for both the hiding and binding properties. Our transformation works for both the information-theoretic and computational settings. Both the transformation and the reductions are uniform, and the reductions run in time polynomial in the security we achieved.

On the other hand, for the common framework of asymptotic security where we require *"negligible"* security against all PPT adversaries, i.e. $s^{-\omega(1)}$ for another parametrization of security parameter s, we can achieve a transformation with reduction running time in $s^{\Omega(1)} = \mathrm{poly}(s)$. Furthermore, instead of a bit-commitment scheme, our transformation obtains a standard $\Omega(\log s)$-bit string-commitment scheme using $n'(s)$ black-box calls to Com_0 for any $n'(s) = \omega(\log s)$, improving $\omega(\log^2 s)$ number of calls in the previous results [DKS99, HR08].

To bypass the $\Omega(k^2)$ barrier of the previous transformation, we use error-correcting codes and randomness extractors to amplify both hiding and binding properties *simultaneously*. To illustrate the idea and our construction, 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 $\delta \cdot n'$, and $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^t$ a strong randomness extractor. Our transformation uses Com_0, C and Ext to commit to a string $v \in \{0,1\}^t$ as follows.

- 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.

The intuition is as follows. 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. Intuitively, 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 at most $q \cdot n'$ commit bits. If $q < (0.9)\delta$, the Chernoff bound says that S^* should be able to cheat on at least $\delta \cdot n'$ commit bits with only exponentially small probability in n'. 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, and hides the value v.

The idea of using error correcting codes and randomness extractors to amplify multiple security properties has been used in various settings before. Crépeau [Cré97] used a very similar construction to construct information-theoretically secure bit commitment schemes using a noisy channel. Dwork, Naor, and Reingold [DNR04] studied security amplification for encryption schemes in both

setting, $O(k^2)$ black-box calls suffices for this transformation. In the computational setting, the analysis of[HR08] uses $\omega(k^2)$ black-box calls.

information-theoretic and computational settings. In the information-theoretic setting, they mentioned that error-correcting codes can be used to amplify the security of encryption schemes. However, they did not analyze the construction in computational setting, and instead gave alternative constructions for computational security. Later on, Holenstein and Renner [HR05] used error-correcting codes to achieve one-way key agreement in information-theoretic setting and applied it to bit encryption schemes in computational setting. Note that in the above applications of error-correcting codes and extractors for security amplification, the first three are information-theoretic, and the last one is computational but non-interactive.

Our main technical contribution is to carry out the analysis of the aforementioned transformation in *interactive and computational* setting. In particular, we prove a "Full-Spectrum Theorem" for hardness amplification of puzzle systems, generalizing previous known results of [CHS05, IJK07, HR08, DIJK09].[2] The result can be used to analyze the binding property. For the hiding property, we give a way to extract computational entropy in interactive setting. Our analysis requires the use of a specific type of randomness exactors and systematic codes with very good rate. We elaborate on the challenges and our new ideas in detail below.

1.2 Our Techniques

1.2.1 Binding and Full-Spectrum Theorem for (Interactive) Puzzle Systems

The Task. Informally, we can view the task of breaking the binding property of Com_0 as an interactive puzzle system where the adversarial sender S_0^* is a solver and the honest receiver R_0 is a puzzle generator. The puzzle is the interactively generated commitment. S_0^* successfully solves the puzzle if S_0^* can open the commitment in two ways. Com_0 being q-binding means that no S_0^* can solve the puzzle with probability better than q. Phrased in this way, we can describe the task of breaking the binding property of Com as follows: S^* gets n' sequentially generated puzzles, and for S^* to cheat, S^* is required to solve at least $\delta \cdot n'$ puzzles. Note that the puzzle system is "two-phase" in the sense that the puzzles are generated sequentially (and interactively) in the commit stage, and they are solved *in parallel* (non-interactively) in the reveal stage. Thus, we need a Chernoff-type hardness amplification result for such two-phase puzzle systems saying that if $q < 0.9\delta$, then solving at least $\delta n'$ puzzles is exponentially small.

Context and Previous Work. The question of hardness amplification/degradation of puzzle systems has been studied extensively in recent years. The answer to this question is very sensitive to the model, for example, whether the puzzles are generated in parallel or sequentially, whether they are interactive or non-interactive, and whether they are (weakly) verifiable or not? Negative results where the hardness is not amplified are known for puzzle systems that are not verifiable [CHS05] and interactive puzzle systems [BIN97, PW07]. Positive results are known for (weakly) verifiable puzzle systems [CHS05] and puzzle systems with restricted type of interaction [HR08, DIJK09].

Previously, Chernoff-type Theorems are first proved by Impagliazzo, Jaiswal, and Kabanets [IJK07] for parallel repetition of weakly-verifiable puzzles, and then extended by Dodis, Impagliazzo, Jaiswal, and Kabanets [DIJK09] to dynamic weakly-verifiable puzzles, which are generalizations of weakly-verifiable puzzles with interactive query oracles. Unfortunately, both their reduction algorithms and analyses seem not applicable to the two-phase puzzles.

On the other hand, Halevi and Rabin [HR08] proved a "Direct Product Theorem" and "Hardness Degradation Theorem" for the two-phase puzzles to analyze the binding property. Specifically, they showed that suppose the probability of solving one puzzle is at most q, then the probability of solving

[2] Independently, Holenstein and Schoenebeck [HS09] obtained similar results about amplification of puzzle systems.

all n' puzzles becomes at most $q^{n'}$, and solving at least one puzzle becomes at most $1-(1-q)^{n'}$ (up to negligible terms). Their reduction technique is different from that of [IJK07, DIJK09].

Our Results. We show that the three types of hardness results (Direct Prodcut, Hardness Degradation, Chernoff-type) actually hold for the three aforementioned puzzle systems (weakly-verifiable puzzles, dynamic weakly-verifiable puzzles[3], and two-phase puzzles.) We establish a "Full-Spectrum" Theorem that essentially says that the hardness of solving at least r puzzles out of n' puzzles, where each puzzle can be solved with probability at most q, amplifies/degrades at the optimal, information-theoretic rate, namely, the probability that n' independent Bernoulli random variables with expectation q have sum at least r. In particular, this allows us to analyze the binding property of our transformation.

Independent of our work, Holenstein and Schoenebeck [HS09] achieved similar results for all puzzle systems. They had a cleaner way to deal with the parameters that handles wider range. Furthermore, they consider more general "monotone combining functions" in addition to all threshold functions in our work.

1.2.2 Hiding and Computational Entropy Extraction

The Task. As discussed in Section 1.1, the intuition behind the hiding property of our construction is that the message m should still have a lot of "computational entropy" after the receiver seeing the weak commitments to the bits of the codeword $C(m)$, and thus $\mathrm{Ext}(m,z)$ should be pseudorandom given the receiver's view. Thus, we need to analyze this process of "extracting computational entropy," and need to do so in an *interactive* setting, which is a difficulty for previous approaches.

Context and Previous Work. Extracting computational entropy for non-interactive primitives has been studied for years in cryptography and complexity theory. A classic example is the celebrated construction of pseudorandom generators from one-way functions by Haståd, Impagliazzo, Levin, and Luby [HILL99]. Holenstein [Hol06] recently gave a simpler and more modular proof of their results. Following an approach of Sudan, Trevisan and Vadhan [STV01] in a different setting (namely, nonuniform and non-cryptgraphic), Holenstein used a (new) version of Impagliazzo's Hardcore Lemma [Imp95] to "relate computational entropy to real entropy," so that it suffices to solve the information-theoretic analogue, namely, extracting almost uniform bits from a distribution with high real entropy. Indeed, Holenstein's analysis works with *any* efficient randomness extractor. This approach was later used by Holenstein and Renner [HR05] for amplifying weak encryption schemes.

For the interactive primitives such as commitment schemes, Wullschleger [Wul07] applied the Hardcore Lemma to the "honest-but-curious" model, and proved the security under this model. One can use the Goldreich-Micali-Wigderson compiler [GMW86] to extend the security to against malicious adversaries, but the transformation becomes non-black-box, and less efficient.

To summarize, these previous approaches using the Hardcore Lemma relied on the fact that either the primitive is non-interactive or worked in the honest-but-curious model. Halevi and Rabin [HR08] pointed out the difficulties of generalizing the Hardcore Lemma to interactive primitives in general malicious model. Thus, instead of using Hardcore Lemma, Halevi and Rabin proved an "interactive version" of Yao's XOR lemma, and used this to extract one bit computational entropy in the interactive setting. This indeed bypassed the barrier, but it only gives a commitment to one bit for each XOR (which involves several calls to Com_0.)

Our Result. In this work, we are able to extract computational entropy in the interactive setting

[3] Actually, a variant of dynamic weakly-verifiable puzzles. See discussion in Section 3.

with parameters that are close to the statistical setting. Our construction uses a specific strong randomness extractor, namely the Goldreich-Levin extractor [GL89], and our analysis composes various known techniques in a new way. A key step to handle the interactive setting is to use the Direct Product Theorem of Halevi and Rabin [HR08] for sequential interactive puzzles. To the best of our knowledge, this result is the first one that can extract many bits in the interactive setting.

2 Notations and Definitions

All log's are base 2. s is the security parameter, and $\mathsf{ngl} = \mathsf{ngl}(s)$ denotes a negligible function of the security parameter. We use U_n to denote uniform distribution over n-bit strings. We identify $\{0,1\}$ with $\mathbb{F}_2$, the finite field of size 2. If $x, y \in \{0,1\}^n$ are vectors in $\mathbb{F}_2^n$, then $x \oplus y \in \{0,1\}^n$ denotes their sum, (i.e. bit-wise xor) and $x \cdot y \stackrel{\text{def}}{=} \sum_i x_i y_i \in \{0,1\}$ denotes their inner product.

We review the facts we need about error-correcting codes. The lemma below says that a short random linear code has good minimum distance with overwhelming probability. It can be proved by standard probabilistic methods, and we omit the proof. The constants in the lemma are computationally practical.

Definition 1 *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 parts of codewords $C(x)$ and $C(y)$ such that $x \neq y$.*

Lemma 2 *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.

Lemma 3 (Goldreich-Levin[GL89]) *There is an oracle algorithm $B^{(\cdot)}$ such that for any $x \in \{0,1\}^n$ and an oracle A satisfying*

$$Pr_{r \leftarrow U_n}[A(r) = x \cdot r] > \frac{1}{2} + \gamma,$$

B^A makes $O(\frac{n}{\gamma^2})$ queries to A and then efficiently outputs a list of size $O(\frac{1}{\gamma^2})$ elements such that x is in the list with probability greater equal than $\frac{1}{2}$.

2.1 Commitment Scheme

In this section, we formally define commitment schemes and present our transformation and main theorems. We consider a standard model where the communication is over the classical noiseless channel and the decommitment is non-interactive with perfect correctness [Gol01, HR08].

Definition 4 (Commitment Scheme) *A* commitment scheme *is an interactive protocol* $\mathrm{Com} = (S, R)$ *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 $x = \mathrm{output}((S(v), R)(1^s))$*, and a private output for S, which we call the* decommitment *string $d = \mathrm{output}_S((S(v), R)(1^s))$. 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. *The sender S and receiver R are 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 The assumption of non-interactive reveal stage is essential in both our work and the previous work [HR08]. 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-knowledgeness 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. We will consider both concrete and asymptotic formulations, where the security parameter will be used only for the asymptotic version.

Definition 6 (p-hiding against time T) *A commitment scheme $\mathrm{Com} = (S, R)$ is p-hiding against uniform time T if for every time T cheating receiver R^*, the distributions $(\mathrm{view}_{R^*}(S(U_t), R^*), U_t)$ and $(\mathrm{view}_{R^*}(S(U_t), R^*), U'_t)$ are p-indistinguishable for time T, where U'_t is an i.i.d. copy of U_t. That is, for every time T distinguisher D,*

$$|\Pr[D(\mathrm{view}_{R^*}(S(U_t), R^*), U_t) = 1] - \Pr[D(\mathrm{view}_{R^*}(S(U_t), R^*), U'_t) = 1]| \leq p$$

We say Com *is* p-hiding *if $\mathrm{Com}(1^s)$ is p-hiding against time s^c for every constant c, and sufficiently large security parameter s.*

We remark that the hiding property above is defined as the indistinguishability against *randomly values*, which does not generally imply the standard definition of the indistinguishability against *a pair of efficiently generated values*. Nevertheless, it is known how to transform a commitment scheme with the above hiding property to one with standard hiding property. Moreover, it can be shown that our transformation has the property that if the resulting commitment scheme satisfies the above definition of hiding property, then it satisfies the standard hiding property.

Remark 7 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,

$$\Pr[P(\mathrm{view}_{R^*}(S(U_1), R^*)) = U_1)] \leq 1/2 + p/2.$$

Definition 8 (Binding Game) *The* binding game *for a commitment scheme* $\mathrm{Com} = (S, R)$ *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* $\mathrm{view}_{S^*}(S^*, R)$*.*
2. *In the decommitment finding stage,* F *gets the view* $\mathrm{view}_{S^*}(S^*, R)$*, 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 9 (q-binding against time T) *A commitment scheme* $\mathrm{Com} = (S, R)$ *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* $\mathrm{Com}(1^s)$ *is* q*-binding against time* s^c *for every constant* c*, and sufficiently large security parameter* s*.*

For convenience, we say 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 $\mathrm{Com}(1^s)$ is (s^{-c}, s^{-c})-secure for every constant c, and sufficiently large security parameter s. Let Com_0 be a (p, q)-secure (weak) bit commitment scheme against time $2^{\Omega(k)}$ for sufficiently small constants $p, q \in (0, 1)$. The following black box transformation uses Com_0 to construct a $(2^{-k}, 2^{-k})$-secure t-bit string-commitment scheme against time 2^k with $t = \Omega(k)$.

Definition 10 (Transformation $\mathcal{T}(\mathrm{Com}_0, n, \ell, t)$) *Let* Com_0 *be a bit commitment scheme, and* $n, \ell, t : \mathbb{N} \to \mathbb{N}$ *be efficiently computable functions of the security parameter* s*. We define transformation* $\mathcal{T}$ *that construct a* t*-bit string-commitment scheme* $\mathrm{Com} = (S, R)$ *as follows.*

- *Commit stage: Let* $v \in \{0,1\}^t$ *be the string to which* S *is committing .*
  1. R *samples a uniformly random matrix* $A \leftarrow \{0,1\}^{\ell \times n}$*, and sends* A *to* S*.*
  2. S *samples the following uniformly at random: a vector* $m \leftarrow \{0,1\}^n$ *and a matrix* $Z \leftarrow \{0,1\}^{t \times n}$*.*
  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.*
  4. S *sends* $(Z, v \oplus Zm)$ *to* R*, where* $v \oplus Zm$ *is the bit-wise xor of* v *and* Zm*.*

  In sum, the commitment of v *is simply* $(A, \vec{x}, \vec{y}, Z, v \oplus Zm)$*.*
- *Reveal stage:* S *sends* v *and its coin tosses* r *to* R*, and* R *checks that* v *and* r *are consistant with the honest sender's algorithm.*

The intuition for the above construction is as follows. The matrix A defines a systematic linear code $C(m) \stackrel{\text{def}}{=} (m, Am)$, which has good minimum distance, say at least $\delta \cdot n$, with high probability. The random matrix Z corresponds to a (strong) randomness extractor $\mathrm{Ext}(m, Z) \stackrel{\text{def}}{=} Zm$. The sender picks a random string m, commits to the codeword $C(m)$, and extracts the remaining randomness of m as a one-time pad to commit to v. The binding property is improved because for a 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 $\delta \cdot n$ commit bits out of $n + \ell$ commit bits. The hiding property is improved because after seeing the commitments of $C(m)$, a cheating receiver R^* has only partial information about m. Thus, Ext extracts the remaining entropy from m, and hides the value of v. Formally, we prove the following theorem.

Theorem 11 (main) *The following holds for all sufficiently small constants $p, q \in (0,1)$: For every security parameter s and $k = k(s) \in \mathbb{N}$, if there exists a (p,q)-secure (weak) bit commitment scheme* Com_0 *running in time $2^{O(k)}$ against time $2^{\Omega(k)}$, then there exists a $(2^{-k}, 2^{-k})$-secure $t = \Omega(k)$-bit string-commitment scheme* Com *against time $2^{\Omega(k)}$ that makes $O(k)$ black-box calls to* Com_0*. Specifically,* $\mathrm{Com} = \mathcal{T}(\mathrm{Com}_0, n, \ell, t)$ *for appropriate $n, \ell = O(k)$, and $t = \Omega(k)$.*

By composing our transformation with the transformations of Halevi and Rabin [HR08], we can improve the efficiency of security amplification for weak bit commitment schemes. For every constants p and q with $p + q < 1$, to amplify the security from a (p,q)-secure commitment scheme Com_0, we can reduce the number of call to Com_0 from $\omega(\log^2 s)$ to any $\omega(\log s)$ function, such as, $\log(s) \cdot \log^*(s)$. Furthermore, we can obtain a $\Omega(\log(s))$-bits string commitment scheme instead of bit commitment scheme.

Theorem 12 *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 $n' = \omega(\log s), t = O(\log s)$, there exists a secure t-bit commitment scheme* Com *that makes only n' black-box call to* Com_0 *on security parameter s.*

2.2 Puzzles Systems and Hardness Amplification

In this section, we define the models for the analysis of the binding property. In particular we define two-phase weakly-verifiable puzzle systems, which generalize the weakly-verifiable puzzle systems of Canetti, Halevi, and Steiner [CHS05], and is implicitly studied in [HR08]. Here the two-phase puzzle system considers the case where puzzles can be generated jointly by the puzzle generator and the solver, where previously puzzles are generated only by the puzzle generator [CHS05].

To capture the task of breaking the security conditions of *sequentially* applied protocols, such as breaking the binding or hiding property of n sequentially committed commitments, we consider a composition of such puzzle systems, where n puzzles are generated sequentially, and then the solver S is required to solve all puzzles simultaneously. We remark that this setting is different from the dynamic weakly-verifiable puzzle systems of Dodis *et al.* [DIJK09], which also generalize the model of [CHS05].

Definition 13 (Two-Phase Puzzle System) *A puzzle system P is a* two-phase puzzle system *if $\mathsf{P} = (G, V)$ consists of a randomized algorithm G (puzzle generator) and a deterministic algorithm V (puzzle verifier). Let S be a solver for P. The interaction of $\langle \mathsf{S}, \mathsf{P} \rangle$ 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 $p \leftarrow \langle \mathsf{S}, G(c) \rangle(1^s)$, where c is the private coins of G. The generation of p may take polynomially many rounds.*
- *In the second phase, S sends an answer $a = \mathsf{S}(p)$ to P*
- *In the end of the protocol, P verifies the answer using V and accepts iff $V(c, a) = 1$.*

Definition 14 *We call a puzzle system $\mathsf{P} = (G, V)$ a fully verifiable puzzle system if the verifier V depends only on the puzzle and the answer but not on P's random coins. This is: for all puzzles p and answers a, $V(p, a)$ decides the correctness of the answer to the puzzle. On the other hand, for a weakly verifiable puzzle system, V might depend on the random coins used to generate the puzzle, as stated above $V(c, a) = 1/0$.*

Definition 15 (Hardness of Solving a Puzzle) *A puzzle system* P *is* δ-hard against time T *if for every time* T *solver* S*, the success probability satisfies* $succ_{\mathsf{P}}[S] \leq \delta$.

For simplicity, if we only say a puzzle is δ-hard (without saying against time T), then we mean that it is against all polynomial time adversaries. Note that this only makes sense in the asymptotic setting.

Now, we claim that the puzzle system described above captures the task of breaking the binding property of a commitment scheme $\mathrm{Com}_0 = (S, R)$ as follows. The solver S plays the role of the cheating sender S^* and the generator G plays the role of the receiver R. Then the puzzle is generated jointly by S and G according to the commitment scheme Com_0 and results as the view of S^* i.e. $\mathrm{view}_{S^*}(S^*, R)$. The check information c is the private coin tosses of R, and a valid answer for the puzzle is a pair of decommitment string $((v, d), (v', d'))$ that are 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.

Note that the puzzle system is weakly verifiable in the running time of Com_0. Furthermore, if Com_0 has perfect correctness, then the puzzle system is verifiable, because the solver S can check the acceptance of $((v, d), (v', d'))$ by herself without accessing to the check information (the coin of receiver R) by checking consistency of (v, d) and (v', d') to the commitment.

We proceed to define the sequential generalization of a puzzle system, and the corresponding puzzle solving game.

Definition 16 ((n, r)-Sequential Two-Phase Puzzle System) *Let* $\mathsf{P} = (G, V)$ *be a two-phase puzzle system. We define the corresponding* (n, r)*-sequential two-phase puzzle system* $\mathsf{P}^{n,r}_{seq} = (G^n, V^{n,r})$ *as follows. For a solver* S^n*,* $\langle \mathsf{S}^n, \mathsf{P}^{n,r}_{seq} \rangle$ *in the first phase runs* $\langle \mathsf{S}, G(c_i) \rangle (1^s)$ n *times sequentially to get the puzzle* p_i*, where* c_i *is* G*'s secret coins in* i*-th repetition, for* $i = 1, 2, \ldots, n$*. Then in the second phase,* S *computes answers* $\vec{a} = \mathsf{S}^n(\vec{p})$ *and sends them to* $\mathsf{P}^{n,r}_{seq}$*.* $V^{n,r}(\vec{c}, \vec{a})$ *accepts iff at least* k *copies of* $V(c_i, a_i)$ *accept.*

Since $\mathsf{P}^{n,r}_{seq}$ is simply a two-phase puzzle system in the sense of Definition 13, its hardness is defined via Definition 15. Phrased in this language, the Direct Product Theorems of [CHS05, HR08] refers to the special case $r = n$, which says that if P is δ-hard, then $\mathsf{P}^{n,n}_{seq}$ is δ^n-hard (up to a small slackness), and the Hardness Degradation Theorem of [HR08] refers to the special case $r = 1$, which says that if P is δ-hard, then $\mathsf{P}^{n,1}_{seq}$ is $(1 - (1 - \delta)^n)$-hard (up to a small slackness.)

Observe that the hardness of breaking the binding property of r out of n sequentially committed commitments translates to the hardness of solving r out of n puzzles in the corresponding puzzle system, and that in our transformation in Definition 10, to break the binding property of Com requires breaking $\delta \cdot n$ out of $n + \ell$ invocation of Com_0, where $\delta \cdot n$ is the minimum distance of the code $C(m) = (m, Am)$. A Chernoff-type hardness amplification result for puzzle system implies the q-binding property is improved exponentially fast in n if $q \cdot (n + \ell) < (0.9)\delta \cdot n$.

3 Puzzles and Full Spectrum Theorems

In this section, we present our full spectrum hardness results of puzzle systems. We prove the following theorem, which essentially says that repetition amplifies the hardness of puzzle systems in an optimal, information-theoretic rate. For our application to commitment schemes, we state and analyze the theorem for sequential repetition of two-phase puzzle systems. Our reduction algorithm

can be implemented easily for parallel repetition of weakly-verifiable puzzles and a variant[4] of dynamic weakly-verifiable puzzles of Dodis et al. [DIJK09]. The same analysis goes through and gives theorems of exactly the same form for (parallel repetition of) these models, generalizing the Chernoff-type theorems of Impagliazzo et al. [IJK07] and Dodis et al. [DIJK09].

Theorem 17 *For any constants $\gamma, \delta, \alpha \in (0,1)$ and efficiently computable functions $n, r \in \mathbb{N} \to \mathbb{N}$ with $1 \leq r(s) \leq n(s) \leq \mathrm{poly}(s)$, the following holds. Let $\mathsf{P} = (G, V)(1^s)$ be a two-phase puzzle system. Suppose P is δ-hard, then the following holds:*

1. *The (n,r)-sequential repetition $\mathsf{P}_{seq}^{n,r}$ is $(P(n, r, (1+\alpha)\cdot\delta) + \mathsf{ngl})$-hard.*
2. *If $r \geq (1+\gamma)\delta n$ (i.e., the Chernoff-type regime), then $\mathsf{P}_{seq}^{n,r}$ is $(P(n,r,\delta) + \mathsf{ngl})$-hard.*
3. *If P is* fully-verifiable*, then $\mathsf{P}_{seq}^{n,r}$ is $(P(n,r,\delta) + \mathsf{ngl})$-hard even for efficiently computation and noticeable $\delta : \mathbb{N} \to [0,1]$, where $P(n,r,\delta) = \sum_{i=r}^{n} \binom{n}{i} \delta^i (1-\delta)^{n-i}$, i.e. the probability that the sum of n independent binomial random variable with mean δ is greater than r.*

Our theorem holds for any arbitrary number of repetition $n \leq \mathrm{poly}(s)$ and threshold $r \in [n]$, in comparison to the Chernoff-type theorems of [IJK07, DIJK09], which holds for sufficiently large n with $r \geq (1+\gamma)\delta n$). Independently, Holenstein and Schoenebeck [HS09] used essentially the same idea for the reduction with a cleaner way to deal with error, and thus obtained the result for general puzzles that matches the parameters of the third point in Theorem 17. For the case of parameters in our applications, the differences are not significant. The core part of our proof is the following concrete version of lemma (Lemma 18).

Lemma 18 *For any $n, r, \eta, T \in \mathbb{N}, \delta \in [0,1]$ with $r \leq n$, the following holds. Let $\mathsf{P} = (G, V)$ be a two-phase puzzle system with $T_G, T_V \leq T$, and $\mathsf{P}_{seq}^{n,r}$ the corresponding (n,r)-sequential puzzle system. Suppose there exists a time T solver S^n for $\mathsf{P}_{seq}^{n,r}$ with $\mathit{succ}_{\mathsf{P}_{seq}^{n,r}}[\mathsf{S}^n] \geq P(n,r,\delta)$. Then there exists a solver S for P such that $\mathit{succ}_{\mathsf{P}}[\mathsf{S}] \geq \delta \cdot (1 - 1/\eta)$, and S runs in time $T' = \mathrm{poly}(n, \eta, \delta^{-r}, (1-\delta)^{-(n-r)}) \cdot T$, given n, r, η, T and δ.*

Although the reduction runs in time $\mathrm{poly}(\delta^{-r}, (1-\delta)^{-(n-r)})$, which may seem inefficient, an efficient reduction for Theorem 17 can be obtained by applying a simple reduction that converts a solver S^n to a solver $\mathsf{S}^{n'}$ for a smaller n' before applying Lemma 18. The slightly improved parameter range for fully-verifiable puzzles comes from a more efficient reduction in the following lemma.

Lemma 19 *For any $n, r, \eta, T \in \mathbb{N}, \delta \in [0,1]$ with $r \leq n$, the following holds. Let $\mathsf{P} = (G, V)$ be a* fully-verifiable *two-phase puzzle system with $T_G, T_V \leq T$, and $\mathsf{P}_{seq}^{n,r}$ the corresponding (n,r)-sequential puzzle system. Suppose there exists a time T solver S^n for $\mathsf{P}_{seq}^{n,r}$ with $\mathit{succ}_{\mathsf{P}_{seq}^{n,r}}[\mathsf{S}^n] \geq P(n,r,\delta)$. Then there exists a solver S for P such that $\mathit{succ}_{\mathsf{P}}[\mathsf{S}] \geq \delta \cdot (1 - 1/\eta)$, and S runs in time $T' = \mathrm{poly}(n, \eta, \delta^{-1}, P(n,r,\delta)^{-1}) \cdot T$, given n, r, η, T and δ.*

We defer all proofs to Appendix.

[4]The variant is implicit in Lemma 5 of [DIJK09], which is the model where they actually prove the hardness amplification.

4 The Proof of Main Theorem

This section is devoted to prove the main Theorem 11. We analyze the binding and hiding properties separately in two lemmas below. Due to the space limit, we just outline the proofs and defer the rigorous ones to the appendix.

Lemma 20 (Binding) *Let d_0 be the universal constant in Lemma 2. There exist universal constants c_1, c' 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) $c'k \geq n \geq c_1 \cdot k/q$, (iii) $2^k \geq \ell \geq d_0 \cdot (3q) \cdot \log(1/3q) \cdot n$, and (iv) $t \leq 2^k$, if a bit-commitment scheme $\mathrm{Com}_0 = (S_0, R_0)$ running in time T_0 is q-binding against time T, then $\mathrm{Com} = \mathcal{T}(\mathrm{Com}_0, n, \ell, t)$ is 2^{-k}-binding against time $T' = T/\mathrm{poly}(2^k, T_0)$.*

(Outline) We give an outline of the proof as follow. By the setting of the parameters above, Lemma 2 shows that with high probability, the linear code has minimum distance at least δn. Conditioning on the code being good, we show the probability that adversary S^* breaks at least δn Com_0's decreases exponentially fast, using the Chernoff-type result of the Lemma 18. This completes the proof.

Lemma 21 (hiding) *There exist universal constants c_2, c'' such that the following holds. For every $\alpha \in (0,1), n, k, \ell, t, T_0, T \in \mathbb{N}$ satisfying (i) $2^k \geq n \geq c_2 \cdot k/\alpha$, (ii) $\ell \leq c''k$, (iii) $t \leq \alpha n/12$, if $\mathrm{Com}_0 = (S_0, R_0)$ running in time T_0 is a $(1-\alpha)$-hiding against time T, then $\mathrm{Com} = \mathcal{T}(\mathrm{Com}_0, n, \ell, t)$ is 2^{-k}-hiding against time $T' = T/\mathrm{poly}(2^k, T_0)$.*

(Outline) We give an outline of the proof as follows. We prove the contrapositive statement that if we can break the hiding property of Com, then we can break the hiding property of Com_0. First, suppose there exists a distinguisher that distinguishes $(\mathrm{Com}(U_t), U_t)$ and $(\mathrm{Com}(U_t), U'_t)$ with probability non-negligibly larger than $1/2$, then we can obtain a next bit predictor P that predicts the i-th bit of U_t given the first $i-1$ bits of U_t and $\mathrm{Com}(U_t)$. From our construction and P, we can use Goldreich-Levin algorithm to obtain the message U_t with non-negligible probability. Then using the Direct Product Theorem for weakly verifiable puzzles in [HR08] and Lemma 18, we can predict U_1 given $\mathrm{Com}_0(U_1)$ with high probability, which implies we can break the hiding property of Com_0.

Proof. (**of Main Theorem**) We set the parameters n, k, ℓ as follows: $n = \max\{\frac{c_1 k}{q}, \frac{c_2 k}{1-p}\} = O(k)$, $\ell = d_0(3q)\log(3q) \cdot n$, and $t = \frac{(1-p)n}{12} = \Omega(k)$, where c_1, c_2, d_0 are the constants in the Lemma 2, 20, and 21. Then the proof follows directly from the lemmas.

■

5 Commitment Schemes with Standard Asymptotic Security

In this section, we present the ideas for the proof to Theorem 12, which says that for all constants p, q with $p + q < 1$, given a weak (p, q)-secure bit commitment scheme Com_0, we can securely commit to $\Omega(\log s)$ bit by applying Com_0 only n' times for any function $n' = \omega(\log s)$. Recall that the standard asymptotic security refers to (s^{-c}, s^{-c})-secure against time s^c for every constant c for sufficiently large s. Working with this definition incurs some subtleties. We outline the construction with an informal discussion below.

Let p_0, q_0 be constants with $p_0 + q_0 < 1$, and Com_0 be a (p_0, q_0)-secure bit commitment scheme. Since our transformation only works for sufficiently small p and q, we apply the transformations of

[HR08] first to bring p_0, q_0 down to sufficiently small constants p_1, q_1. Since we amplify the security from constant to constant, this requires only a constant number of calls to Com_0. Let Com_1 be the resulting (p_1, q_1)-secure bit commitment scheme.

The next step is to apply our transformation to Com_1 to obtain a t-bit string commitment scheme Com_2. However, since the reductions can only run in $\mathrm{poly}(s)$ time, we can only apply Theorem 11 with $k = O(\log(s))$ and thus obtain an (s^{-c}, s^{-c})-secure for a constant c. Then we apply the transformations of [HR08] to Com_2 to amplify the security to negligible.

Here there are some subtleties that we need to deal with. Our transformation gives a "string"-commitment scheme instead of a "bit"-commitment scheme, so we need to generalize the "repetition transformation" and the "secret-sharing transformation" of [HR08]. For the "repetition transformation" of [HR08], it is not hard to generalize the analysis for both the binding and hiding properties. For the "secret-sharing," it is not hard to achieve the degradation of the binding property, as we can view that as solving puzzles. Recently, Maurer and Tessaro [MT09] generalized the XOR lemma to the setting of string-commitment schemes. Putting these together, we are able to transform a (s^{-c}, s^{-c})-secure commitment scheme to a $(\mathsf{ngl}, \mathsf{ngl})$-secure one with $\omega(1)$ calls of that. Thus in total, we obtain a secure commitment scheme with $\omega(\log s)$ calls to Com_0.

We put the detailed transformation and proofs in the appendix.

References

[BIN97] Mihir Bellare, Russell Impagliazzo, and Moni Naor. Does parallel repetition lower the error in computationally sound protocols? In *FOCS*, pages 374–383, 1997.

[CHS05] Ran Canetti, Shai Halevi, and Michael Steiner. Hardness amplification of weakly verifiable puzzles. In *TCC*, pages 17–33, 2005.

[Cré97] Claude Crépeau. Efficient cryptographic protocols based on noisy channels. In *EUROCRYPT*, pages 306–317, 1997.

[DIJK09] Yevgeniy Dodis, Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. Security amplification for interactivecryptographic primitives. In *TCC*, pages 128–145, 2009.

[DKS99] 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.

[DNR04] Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing encryption schemes from decryption errors. In *EUROCRYPT*, pages 342–360, 2004.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32, 1989.

[GMW86] 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.

[Gol01] Oded Goldreich. *Foundations of Cryptography. Basic tools*. Cambridge University Press, 2001.

[HILL99] 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.

[Hol06] Thomas Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In *TCC*, pages 443–461, 2006.

[HR05] 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.

[HR08] Shai Halevi and Tal Rabin. Degradation and amplification of computational hardness. In *TCC*, pages 626–643, 2008.

[HS09] Thomas Holenstein and Grant Schoenebeck. General hardness amplification of predicates and puzzles. Unpublished manuscript, 2009.

[IJK07] Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. Chernoff-type direct product theorems. In *CRYPTO*, pages 500–516, 2007.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *FOCS*, pages 230–235, 1989.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *FOCS*, pages 538–545, 1995.

[MT09] 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.

[Nao89] Moni Naor. Bit commitment using pseudo-randomness. In *CRYPTO*, pages 128–136, 1989.

[PW07] Krzysztof Pietrzak and Douglas Wikström. Parallel repetition of computationally sound protocols revisited. In *TCC*, pages 86–102, 2007.

[STV01] Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the xor lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

[Wul07] Jürg Wullschleger. Oblivious-transfer amplification. In *EUROCRYPT*, pages 555–572, 2007.

[Yao82] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *FOCS*, pages 80–91, 1982.

A Proofs in Section 4

This section we put the missing proofs for the two main lemmas in section 4.

A.1 Binding Lemma

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^*, and then (S^*, R) is supposed to use Com_0 $(n+\ell)$ times to commit each bit of a random message pair (m, Am). Let $C : \{0,1\}^n \to \{0,1\}^{n+\ell}$ be a linear code defined by $C(m) = (m, Am)$. For S^* to win the game, he needs to decommit the $(n+\ell)$ bits into two valid codewords in C.

We observe that

$$\Pr[S^* \text{ succeed}] \quad \leq \quad \Pr[C \text{ has min-distance} < \delta n] + \Pr[S^* \text{ succeeds } \& \ C \text{ has min-distance} \geq \delta n]$$

and will bound the two probabilities.

First we want to apply Lemma 2 to say that C is a good code with high probability. Let d_0, d_1 be the constants in the Lemma 2, $\delta = 3q$, and $\gamma = d_0 \delta \log(1/\delta)$. We will set $c_1 > 3d_1$ so that $n \geq d_1 \cdot k/\delta$. By Lemma 2, the code C has minimum distance at least $\delta \cdot n$ with probability at least $1 - 2^{-k}/2$.

Then we want to show that S^*'s success probability is small with the code C having min-distance δn. Suppose to the contrary, that S^* succeeds with probability greater than $2^{-k}/2$, as in the section 1.2.1 and 2.2, we can view breaking the binding property as the task of solving $(n+\ell, \delta n)$-sequential two-phase puzzle system $\mathsf{P}_{seq}^{n+\ell,\delta n} = (G^{n+\ell}, V^{n+\ell,\delta n})$. From S^* we can obtain a solver that solves $\mathsf{P}_{seq}^{n+\ell,\delta n}$ with success probability greater equal to $P(n+\ell, \delta \cdot n, 1.02q) \stackrel{\text{def}}{=} \rho$ running within time $T'' \stackrel{\text{def}}{=} \max\{T', T_0\}$, then let $\eta = 100$, and by Lemma 18, there exists another solver that succeeds of breaking P with probability greater than $1.02 \cdot q \cdot (1 - 1/100) \geq q$ running in time $T = \mathrm{poly}(n+\ell, \eta, q^{-\delta n}, (1-q)^{-n-\delta n}) \cdot T''$. This translates to the fact that there exists a time T adversary S_0^* that breaks q-binding of Com_0. Here 1.02 is an arbitrary constant greater to 1, and by setting this, we can make the proof work.

Recall that $\delta = 3q$, and $\gamma \in [0,1]$. Thus, $\delta n = 3qn > 1.02qn(1+\gamma) = 1.02q(n+\ell)$. Then by standard Chernoff bound, we have

$$P(n+\ell, \delta \cdot n, 1.02q) \leq 2^{-\delta n/c}$$

for some constant c that is independent of q and k. Thus, we can set $c_1 = \max\{3d_1, 6c\}$, $c' = \max\{\frac{1}{3q}\log(\frac{1}{q}), \frac{1}{1+3q}\log(\frac{1}{1-q})\}$, so that $c'k \geq n \geq c_1 \cdot k/q$ implies $2^{-\delta n/c} \leq 2^{-k}/2$, $q^{-\delta n} \leq 2^k$, and $(1-q)^{-n-\delta n} \leq 2^k$. Also we have $T'' = \max\{T_0, T'\}$, and therefore $T = T' \cdot \mathrm{poly}(2^k, T_0)$. This contradicts the fact that Com_0 is q-hard against time $T = T' \cdot \mathrm{poly}(2^k, T_0)$.

Thus we have $\Pr[S^* \text{ succeeds}] \leq 2^{-k}/2 + 2^{-k}/2 \leq 2^{-k}$.

■

Remark 22 We observe that in the condition (ii), "$c'k \geq n$" is only used to bound the reduction time to be polynomial in 2^k. If Com_0 has perfect correctness, where the binding property can be modeled as solving *verifiable* puzzle systems, then we have a *more efficient* reduction algorithm as Lemma 19. Thus we can relax the condition (ii) as $2^k \geq n \geq c_1 \cdot k/q$.

A.2 Hiding Lemma

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, 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 Algorithm as Lemma 3, if $\Pr_{z_i}[P(\sigma, \vec{x}, Z_{-i}, z_1 \cdot m, \ldots, z_{i-1} \cdot m, z_i) = z_i \cdot m] > 1/2 + 2^{-k}/2t$, we can make $O(n \cdot 2^{2k})$ 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 $(T' + O(t)) \cdot O(n \cdot 2^{2k})$ 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 $2^{-(\ell+i-1)} \cdot \Omega(2^{-3k}/t^3)$. The idea is try 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

[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 $((\sigma, \vec{x}, Z, U'_t \oplus U_t), U'_t) = ((\sigma, \vec{x}, Z, U_t \oplus Zm), U'_t)$.

$x_{n+1}, \ldots, x_{n+\ell}$ 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'. Since S is honest, Z_{-i} is uniformly random, and easy to prepare. Finally, we can simply guess the value of $(z_1 \cdot m, \ldots, z_{i-1} \cdot m)$. 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'. Since we only need to guess $(\ell + i - 1)$ bits, we can guess it correctly with probability $2^{-(\ell+i-1)}$. Therefore, we have a time $(T' + O(t)) \cdot O(n \cdot 2^{2k})$ algorithm B' such that

$$\Pr[B'(\sigma, x_1, \ldots, x_n) = m] \geq 2^{-(\ell+i-1)} \cdot \Omega(2^{-3k}/t^3) = \Omega(2^{-(3k+\ell+i-1)}/t^3)$$

Now, we can view B' as solving a sequentially generated stateful weakly verifiable puzzle system P^n as follows. The set of puzzles $p \in \mathcal{P}$ in P is the set of possible views $\mathrm{view}_{R^*_0}(S_0(b), R^*_0)$ of a cheating receiver R^*_0 in Com_0. The answer is simply the bit b that a honest S_0 commits to. The check information is the private coins of S_0, or the decommitment string. The puzzle generator G simulates the interaction of a honest sender $S_0(b)$ who wants to commit a uniformly random bit b, with a cheating receiver R^*_0 provided by the solver S. The state of G is a description of R^*_0 together with a configuration of R^*_0 that G uses to simulate the interaction. Thus, B''s task is to solve all n sequentially generated puzzles correctly.

By Remark 7, Com_0 being $(1-\alpha)$-hiding against time T means that for every time T cheating receiver R^*_0, and time T predictor P,

$$\Pr[P(\mathrm{view}_{R^*_0}(S_0(U_1), R^*_0)(1^k)) = U_1] \leq 1/2 + (1-\alpha)/2 = 1 - \alpha/2$$

This means that the puzzle system P is $(1-\alpha/2)$-hard against time T. Now, we want to apply Lemma 18 with $\eta = 6/\alpha$ so that $(1-\alpha/4)(1-1/\eta) > (1-\alpha/2)$, and $r = n$. We set the parameters c_2, c'' such that the conditions (i)(ii)(iii) imply $\Omega(2^{-(3k+\ell+i-1)}/t^4) \geq e^{-\alpha n/4} \geq (1-\alpha/4)^n$, and we know B' is a solver running in $T' \cdot O(n2^{2k})$ that solves all n sequentially generated puzzles with probability $(1-\alpha/4)^n$. Then by the theorem we can obtain a solver B'' that solves a single puzzle with probability $(1-1/\eta)(1-\alpha/4) \geq (1-\alpha/2)$, running in time $T'' = T' \cdot O(n2^{2k}) \cdot \mathrm{poly}(n, \eta, 2^{(3k+\ell+i-1)}) = T' \cdot \mathrm{poly}(n, 2^k)$ with the parameter settings.

This is a contradiction to the fact that Com_0 is $(1-\alpha)$-hiding against time T, where $T = T' \cdot \mathrm{poly}(n, 2^k) = T'' \cdot \mathrm{poly}(n, 2^k)$. ■

B Detailed Transformation in Section 5

We define the repetition and secret-sharing transformation of [HR08] first.

Definition 23 (Repetition $\mathcal{R}(\mathrm{Com}_0, u)$) *Let* Com_0 *be a* t*-bit string-commitment scheme, and* $u \in \mathbb{N}$*. The* repetition transformation $\mathcal{R}(\mathrm{Com}_0, u)$ *defines a* t*-bit string-commitment scheme* $\mathrm{Com} = (S, R)$ *as follows. In the commit stage, to commit a value* $v \in \{0,1\}^t$ *to* R*,* S *simply uses* Com_0 *sequentially* u *times to commit the same value* v *to* R*.*

Definition 24 (Secret-Sharing $\mathcal{SS}(\mathrm{Com}_0, u)$) *Let* Com_0 *be a* t*-bit string-commitment scheme, and* $u \in \mathbb{N}$*. The* secret-sharing transformation $\mathcal{SS}(\mathrm{Com}_0, u)$ *defines a* t*-bit string-commitment scheme* $\mathrm{Com} = (S, R)$ *as follows. In the commit stage, to commit a value* $v \in \{0,1\}^t$ *to* R*,* S *first obtains random* $v_1, v_2, \ldots v_t \in \{0,1\}^t$ *with* $\bigoplus_{i \in [t]} v_i = v$*, i.e. a share of* v*, and then uses* Com_0 *sequentially* t *times to commit to each* v_i *to* R*.*

We proceed to present the analysis of the aforementioned transformation as the following lemmas. The first lemma says that using the transformation of [HR08], we can amplify the security from constant to constant (resp., $o(1)$) using constant (resp., $\omega(1)$) number of black-box calls, and from s^{-c} to negligible using $\omega(1)$ black-box calls.

Lemma 25 ([HR08]) *For all constants $p_0, q_0, p, q > 0$ with $p_0 + q_0 < 1$, we can transform a (p_0, q_0)-secure bit-commitment scheme Com_0 into a (p, q)-secure bit-commitment scheme Com, which makes $O(1)$ black-box calls to Com_0.*

We next analyze the effect of the repetition and secret-sharing transformation to a string commitment scheme.

Lemma 26 *Let Com be a t-bit string-commitment scheme, and $\mathrm{Com}' = \mathcal{R}(\mathrm{Com}, u)$ with efficiently computable $u = u(s) \leq \mathrm{poly}(s)$. If Com is q-binding, then Com' is $(q^u + \mathsf{ngl})$-binding, for some negligible function ngl in the security parameter s.*

Proof. (sketch) Observe that breaking the binding property of Com' requires breaking the binding property of all u calls of Com. The lemma follows by Theorem 18 with $r = u$ (which is the Direct Product Theorem of [HR08].) ■

Lemma 27 *Let Com be a t-bit string-commitment scheme, and $\mathrm{Com}' = \mathcal{R}(\mathrm{Com}, u)$ with efficiently computable $u = u(s) \leq \mathrm{poly}(s)$. If Com is ngl-hiding, so does Com'.*

Proof. (sketch) We sketch the argument by the following informal notation. Let $(\mathrm{Com}(U_t), U_t) \approx_c (\mathrm{Com}(U_t), U'_t)$ represents the ngl-hiding property of Com, where $\mathrm{Com}(U_t)$ is the distribution of a commitment of random value U_t and U'_t is an i.i.d. copy of U_t. Note that the condition is equivalent to $(\mathrm{Com}(U_t), U_t) \approx_c (\mathrm{Com}(U'_t), U_t)$. By definition, $(\mathrm{Com}'(U_t), U_t) = (\mathrm{Com}(U_t), \ldots, \mathrm{Com}(U_t), U_t)$. Observing the we can efficiently generate $\mathrm{Com}(U_t)$ from U_t, $(\mathrm{Com}(U_t), U_t) \approx_c (\mathrm{Com}(U'_t), U_t)$ implies

$$(\mathrm{Com}(U_t), \ldots, \mathrm{Com}(U_t), U_t) \approx_c (\mathrm{Com}(U_t^1), \mathrm{Com}(U_t), \ldots, \mathrm{Com}(U_t), U_t),$$

where U_t^1 denotes an i.i.d. copy of U_t. Note that $\mathrm{Com}(U_t^1)$ can be generated efficiently as well, by the same argument, we have

$$(\mathrm{Com}(U_t^1), \mathrm{Com}(U_t), \ldots, \mathrm{Com}(U_t), U_t) \approx_c (\mathrm{Com}(U_t^1), \mathrm{Com}(U_t^2), \mathrm{Com}(U_t), \ldots, \mathrm{Com}(U_t), U_t).$$

Iteratively applying this argument, we have

$$(\mathrm{Com}'(U_t), U_t) = (\mathrm{Com}(U_t), \ldots, \mathrm{Com}(U_t), U_t) \approx_c (\mathrm{Com}(U_t^1), \ldots, \mathrm{Com}(U_t^u), U_t).$$

On the other hand, observing that $(\mathrm{Com}(U_t), U_t) \approx_c (\mathrm{Com}(U'_t), U_t)$ implies

$$(\mathrm{Com}(U_t), \mathrm{Com}(U_t), \ldots, \mathrm{Com}(U_t)) \approx_c (\mathrm{Com}(U'_t), \mathrm{Com}(U_t), \ldots, \mathrm{Com}(U_t)),$$

we have

$$\begin{aligned}
(\mathrm{Com}'(U_t), U'_t) &= (\mathrm{Com}(U_t), \mathrm{Com}(U_t), \ldots, \mathrm{Com}(U_t), U'_t) \\
&\approx_c (\mathrm{Com}(U_t^1), \mathrm{Com}(U_t), \ldots, \mathrm{Com}(U_t), U'_t) \\
&\approx_c (\mathrm{Com}(U_t^1), \mathrm{Com}(U_t^2), \ldots, \mathrm{Com}(U_t), U'_t) \\
&\vdots \\
&\approx_c (\mathrm{Com}(U_t^1), \mathrm{Com}(U_t^2), \ldots, \mathrm{Com}(U_t^u), U'_t) \\
&= (\mathrm{Com}(U_t^1), \mathrm{Com}(U_t^2), \ldots, \mathrm{Com}(U_t^u), U_t).
\end{aligned}$$

Therefore, $(\mathrm{Com}'(U_t), U_t) \approx_c (\mathrm{Com}'(U_t), U'_t)$, as desired.

■

Lemma 28 *Let* Com *be a t-bit string-commitment scheme, and* $\mathrm{Com}' = \mathcal{SS}(\mathrm{Com}, u)$ *with efficiently computable $u = u(s) \leq \mathrm{poly}(s)$. If* Com *is q-binding, then* Com' *is $(uq + \mathsf{ngl})$-binding, for some negligible function* ngl *in the security parameter s.*

Proof. (sketch) Observe that breaking the binding property of Com' requires breaking the binding property of all u calls of Com. The lemma follows by Lemma 18 with $r = 1$ (which is the Hardness Degradation Theorem of [HR08]. Also, a simple hybrid argument also yields this.) ■

Lemma 29 ([MT09]) *Let* Com *be a t-bit string-commitment scheme, and* $\mathrm{Com}' = \mathcal{SS}(\mathrm{Com}, u)$ *with efficiently computable $u = u(s) \leq \mathrm{poly}(s)$. If* Com *is p-hiding, then* Com' *is $(p^u + \mathsf{ngl})$-binding, for some negligible function* ngl *in the security parameter s.*

Proof. (sketch) This lemma generalizes the bit XOR lemma [HR08] to the string one, and it was proved by Maurer and Tessaro [MT09]. We state it in this form for convenience. ■

We are ready to prove both parts of Theorem 12.

Proof. (of Theorem 12, sketch) Let p_0, q_0 be constants with $p_0 + q_0 < 1$, and Com_0 be a (p_0, q_0)-secure bit commitment scheme. We (i) apply Lemma 25 to obtain a bit commitment scheme Com_1 that is (p_1, q_1)-secure for sufficiently small constants p_1, q_1, (ii) apply our transformation with $k = O(\log s)$ in Theorem 11 to obtain an $t = \Omega(\log s)$-bit *string*-commitment scheme that is (s^{-c}, s^{-c})-secure for some constant c, denoted as Com_1 and then (iii) let a be an arbitrary $\omega(1)$ function with $a = o(\log s)$. We apply the secret-sharing transformation $\mathrm{Com}_2 := \mathcal{SS}(\mathrm{Com}_1, a)$, and then apply the repetition transformation $\mathcal{R}(\mathrm{Com}_2, a)$. Then lemma 29 with $u = a$ shows that Com_2 is a (ngl, as^{-c})-secure t-bit string commitment, and lemma 26 shows that the final scheme is $(\mathsf{ngl}, \mathsf{ngl})$ secure. The total number of calls is $O(1) \cdot O(\log s) \cdot a^2 = n' = \omega(\log s)$.

■

C Proofs for the Puzzles

This section devotes to the proofs of Theorem 17 and Lemma 18, 19. We outline the organization of this section. First we prove the fully-verifiable case, namely Lemma 19. Then we prove Lemma 18 with a different framework. Finally we sketch how to achieve Theorem 17 from both lemmas.

Before the proof, we first observe that if there exists a PPT solver for a puzzle system, then there exists a deterministic solver that also has almost the same success probability. Thus, in the reduction, we may assume a solver S^n for n-fold repetition $\mathsf{P}^{n,k}$ of a puzzle system P to be deterministic by the reasoning. We formalize the lemma as below.

Lemma 30 *For all puzzle system P, for all $\eta(s) \leq \mathrm{poly}(s)$, if there exists a PPT solver T such that $\mathrm{succ}_{\mathsf{P}}[T](s) \geq \delta(s)$, then there exists a PPT S such that with probability $(1 - \mathsf{ngl}(s))$ over S's random coins r, let S_r be the solver S with fixed random coins r, we have $\mathrm{succ}_{\mathsf{P}}[\mathsf{S}_r](s) \geq \delta \cdot (1 - \frac{1}{\eta})$. Note that S_r is S with the fixed coins, so S_r is deterministic.*

C.1 Fully Verifiable Two-Phase Puzzle System

In this section, we are going to prove Lemma 19. Our strategy is to show the contrapositive argument: suppose there exists a solver S^n with success probability $P(n, r, \delta)$ over n puzzles, then there exists a solver S with success probability δ over a single puzzle. We use the notion of game that relates success probabilities of the optimal solver S_{opt} and the efficient solver $\mathsf{S}_{\mathit{eff}}$. Then we can focus on the analysis of S_{opt}, and have a cleaner proof.

C.1.1 Reduction Game

Definition 31 *A* game $\mathcal{G} = (\mathcal{Q}, \mathcal{W}, \mathcal{J})$ *consists of two* players $\mathcal{Q}, \mathcal{W}$ *and a* judge $\mathcal{J}$*. In the game,* $\mathcal{Q}$ *and* $\mathcal{W}$ *plays a fixed number of rounds, denoted by* $\ell \in \mathbb{N}$*.* $\mathcal{Q}$ *always plays the first round, and in each round,* $\mathcal{Q}$ *or* $\mathcal{W}$ *makes a* move *alternatively. At the end of the game, on the entire history of* $\mathcal{Q}$ *and* $\mathcal{W}$*'s moves, the judge* $\mathcal{J}$ *outputs* 1 *or* 0 *to indicate whether* $\mathcal{Q}$ succeeds *in the game.*

We denote the sequence of moves by $q_1, w_1, \ldots, q_m, w_m$*, and let* $\bar{h}^i_q = (q_1, w_1, \ldots, q_{i-1}, w_{i-1}, q_i)$ *and* $\bar{h}^i_w = (q_1, w_1, \ldots, q_i, w_i)$ *to denote the* history *of the game up to* $(2i-1)$*-st and* $2i$*-th round, respectively. The game* $\mathcal{G}$ *naturally induces a game tree, where each* $\bar{h}^i_q$ *(respectively,* $\bar{h}^i_w$*) corresponds to a* $\mathcal{W}$-move *node (respectively,* $\mathcal{Q}$-move *node) in the game tree. Thus, the entire histories, which correspond to the leaf nodes of the game tree, are either of the form* $\bar{h}^m_q$ *when* $\ell = 2m-1$ *is odd, or* $\bar{h}^m_w$ *when* $\ell = 2m$ *is even.*

We are interested the settings where $\mathcal{W}$ plays a fixed randomized strategy, and our goal is to maximize the success probability of $\mathcal{Q}$. Below, we formally define strategies of players and express the success probability of $\mathcal{Q}$.

Definition 32 *Let* $\mathcal{G} = (\mathcal{Q}, \mathcal{W}, \mathcal{J})$ *be a game. A* strategy *of player* $\mathcal{Q}$ *is a randomized algorithm (not necessarily efficient)* $\mathcal{Q}_s$ *that at each* $\mathcal{Q}$*-move node* $\bar{h}^{i-1}_w$*, computes* $\mathcal{Q}$*'s next move* q_i *basing on the history* $\bar{h}^{i-1}_w$*. Thus,* $\mathcal{Q}_s$ *induces a distribution* $\mathcal{D}_s(\bar{h}^{i-1}_w)$ *over* $\bar{h}^{i-1}_w$*'s children* $\bar{h}^i_q = \bar{h}^{i-1}_w \circ q_i$ *for every* $\mathcal{Q}$*-move node* $\bar{h}^{i-1}_w$*. We define a* strategy $\mathcal{W}_t$ *of player* $\mathcal{W}$ *and the induced distribution* $\mathcal{D}_t(\bar{h}^i_q)$ *analogously.*

Since $\mathcal{W}$ always plays a fixed strategy $\mathcal{W}_t$ in our settings, the success probability of $\mathcal{Q}$ depends only on $\mathcal{Q}$'s strategy.

Definition 33 *Let* $\mathcal{G} = (\mathcal{Q}, \mathcal{W}, \mathcal{J})$ *be a game with* $\mathcal{W}$ *playing a fixed strategy* $\mathcal{W}_t$*. Let* $\mathcal{Q}_s$ *be a strategy of* $\mathcal{Q}$*. We inductively define* $\gamma_s(\cdot)$ *to express the success probability of* $\mathcal{Q}_s$ *at each node of the game tree as follows.*

$$\begin{cases} \gamma_s(\bar{h}) = \mathcal{J}(\bar{h}) \quad \textit{for every leaf node } \bar{h} \textit{ (of the form either } \bar{h}^m_q \textit{ or } \bar{h}^m_w\textit{)}, \\ \gamma_s(\bar{h}^i_q) = \mathrm{E}_{w_i \leftarrow \mathcal{D}_t(\bar{h}^i_q)}[\gamma_s(\bar{h}^i_q \circ w_i)] \quad \textit{for every } \mathcal{W}\textit{-move node } \bar{h}^i_q, \\ \gamma_s(\bar{h}^{i-1}_w) = \mathrm{E}_{q_i \leftarrow \mathcal{D}_s(\bar{h}^{i-1}_w)}[\gamma_s(\bar{h}^{i-1}_w \circ q_i)] \quad \textit{for every } \mathcal{Q}\textit{-move node } \bar{h}^{i-1}_w. \end{cases}$$

In particular, γ_s *is the success probability of* $\mathcal{Q}_s$*.*

We will consider two strategies of $\mathcal{Q}$, an optimal strategy $\mathcal{Q}_{\mathrm{opt}}$ and an "efficient" strategy $\mathcal{Q}_{\mathrm{eff}}$. Let us consider an *optimal* strategy $\mathcal{Q}_{\mathrm{opt}}$ first. $\mathcal{Q}_{\mathrm{opt}}$ is a deterministic algorithm that in each round, picks a deterministic move that maximize the success probability of $\mathcal{Q}_{\mathrm{opt}}$ after taking the move. In

other words, for every $\mathcal{Q}$-move node $\bar{h}_w^{i-1}$, the distribution $\mathcal{D}_{\text{opt}}(\bar{h}_w^{i-1})$ concentrate on a single move q_i^* that maximize $\gamma_{\text{opt}}(\bar{h}_w^{i-1} \circ q_i)$, and thus we have

$$\gamma_{\text{opt}}(\bar{h}_w^{i-1}) = \mathop{\mathrm{E}}_{q_i \leftarrow \mathcal{D}_{\text{opt}}(\bar{h}_w^{i-1})}[\gamma_{\text{opt}}(\bar{h}_w^{i-1} \circ q_i)] = \gamma_{\text{opt}}(\bar{h}_w^{i-1} \circ q_i^*) = \max_{q_i}\{\gamma_{\text{opt}}(\bar{h}_w^{i-1} \circ q_i)\}.$$

The optimality of $\mathcal{Q}_{\text{opt}}$ is easy to show by induction. The problem with $\mathcal{Q}_{\text{opt}}$ is that $\mathcal{Q}_{\text{opt}}$ may not be efficient in general. Thus, we try to approximate $\mathcal{Q}_{\text{opt}}$ by the following strategy $\mathcal{Q}_{\text{eff}}$ that finds a good move by sampling. In each round, $\mathcal{Q}_{\text{eff}}$ samples several moves, estimates the success probability of $\mathcal{Q}_{\text{eff}}$ *itself* after taking each move by sampling, and then takes the best move among them according to the estimation.

Definition 34 *Let $\mathcal{G} = (\mathcal{Q}, \mathcal{W}, \mathcal{J})$ be a game with ℓ rounds and $\mathcal{W}$ playing a fixed strategy $\mathcal{W}_t$. Let $\mathcal{S}(\cdot)$ be (efficiently samplable) distributions such that for every $\mathcal{Q}$-move node $\bar{h}_w^{i-1}$, $\mathcal{S}(\bar{h}_w^{i-1})$ is a distribution over the next moves q_i. Let $\varepsilon \in (0, 1]$ be a real number. We define a strategy $\mathcal{Q}_{\text{eff}}(\varepsilon)$ of $\mathcal{Q}$ as follows. At each $\mathcal{Q}$-move node $\bar{h}_w^{i-1}$, $\mathcal{Q}_{\text{eff}}(\varepsilon)$ does the following.*

1. *Let $M = \Theta((1/\varepsilon) \cdot (\log(1/\varepsilon) + \ell))$. Sample M moves $q_{i,1}, \ldots, q_{i,M} \leftarrow \mathcal{S}(\bar{h}_w^{i-1})$ independently.*
2. *For every $j \in [M]$, estimate the success probability $\gamma_{\text{eff}}(\bar{h}_w^{i-1} \circ q_{i,j})$ of $\mathcal{Q}_{\text{eff}}$* itself *after taking the move $q_{i,j}$ by sampling.*
   - *Let $M_i = \Theta((4^i/\varepsilon)^2 \cdot (\log M + \log(1/\varepsilon) + \ell))$. Independently simulate M_i times the game $\mathcal{G}$ with $\mathcal{Q}$ playing $\mathcal{Q}_{\text{eff}}$ and $\mathcal{W}$ playing $\mathcal{W}_t$ from the $2i$-th round with history $\bar{h}_w^{i-1} \circ q_{i,j}$.*
   - *Let $\gamma_{\text{est}}(\bar{h}_w^{i-1} \circ q_{i,j}) \stackrel{\text{def}}{=} (\#\ \mathcal{Q}\text{-successful simulations})/M_i$ be the fraction of simulations that $\mathcal{Q}$ succeeds, which is an estimation of $\gamma_{\text{eff}}(\bar{h}_w^{i-1} \circ q_{i,j})$.*
3. *Let $j^* = \operatorname{argmax}_j\{\gamma_{\text{est}}(\bar{h}_w^{i-1} \circ q_{i,j})\}$. Take the move q_{i,j^*} that maximize the estimated success probability $\gamma_{\text{est}}(\bar{h}_w^{i-1} \circ q_{i,j})$.*

Note that in the second step, $\mathcal{Q}_{\text{eff}}$ needs to simulate itself in the remaining rounds, which uses sampling too. Thus, what $\mathcal{Q}_{\text{eff}}(\varepsilon)$ does is to use recursive sampling to select his next move. We note that although we do not define efficiency explicitly in our notion of game, it is conceivable that if $\mathcal{W}_t$ is efficient, the distribution $\mathcal{S}$ is efficiently samplable and the number of rounds ℓ is constant, then $\mathcal{Q}_{\text{eff}}(\varepsilon)$ can be implemented efficiently.

However, it is impossible for $\mathcal{Q}_{\text{eff}}(\varepsilon)$ to always approximate the optimal strategy $\mathcal{Q}_{\text{opt}}$ well, because $\mathcal{Q}_{\text{eff}}(\varepsilon)$ may never see the best move from distribution $\mathcal{S}$, and the best move can be significantly better than all the other moves. Nevertheless, $\mathcal{Q}_{\text{eff}}(\varepsilon)$ can see one of the ε-fraction of best moves (according to the distribution $\mathcal{S}$) with high probability. Thus, it is possible for $\mathcal{Q}_{\text{eff}}(\varepsilon)$ to achieve comparable success probability to an optimal strategy $\tilde{\mathcal{Q}_{\text{opt}}}$ of $\tilde{\mathcal{Q}}$ in a modified game $\tilde{\mathcal{G}}$ where ε-fraction of best moves are "turned off". Indeed, we will define the modified game $\tilde{\mathcal{G}}$ formally and show that the success probability γ_{eff} of $\mathcal{Q}_{\text{eff}}(\varepsilon)$ in $\mathcal{G}$ is close to the success probability $\gamma_{\tilde{\text{opt}}}$ of $\tilde{\mathcal{Q}_{\text{opt}}}$ in the modified game $\tilde{\mathcal{G}}$.

Definition 35 *Let $\mathcal{G} = (\mathcal{Q}, \mathcal{W}, \mathcal{J})$ be a game with $\mathcal{W}$ playing a fixed strategy $\mathcal{W}_t$. Let $\mathcal{S}(\cdot)$ be distributions such that for every $\mathcal{Q}$-move node $\bar{h}_w^{i-1}$, $\mathcal{S}(\bar{h}_w^{i-1})$ is a distribution over the next moves q_i. Let $\varepsilon \in (0, 1]$ be a real number. Let $\mathcal{Q}_{\text{eff}}(\varepsilon)$ be the strategy defined in Definition 34.*

- *A $\mathcal{W}$-move node $\bar{h}_q^i = (\bar{h}_w^{i-1} \circ q_i)$ is* strong *if $\bar{h}_q^i$ has the ε-fraction of largest $\gamma_{\mathrm{eff}}(\bar{h}_q^i)$ among the children $(\bar{h}_w^{i-1} \circ q_i')$ of $\bar{h}_w^{i-1}$ according to $\mathcal{S}(\bar{h}_w^{i-1})$. That is,*

$$\Pr_{q_i' \leftarrow \mathcal{S}(\bar{h}_w^{i-1})} \left[\gamma_{\mathrm{eff}}(\bar{h}_w^{i-1} \circ q_i) \le \gamma_{\mathrm{eff}}(\bar{h}_w^{i-1} \circ q_i')\right] \le \varepsilon.$$

  Let H denote the set of strong $\mathcal{W}$-move nodes.

- *Let $\tilde{\mathcal{G}} = (\tilde{\mathcal{Q}}, \tilde{\mathcal{W}}, \tilde{\mathcal{J}})$ be a game that is the same as $\mathcal{G}$ except that the judge $\tilde{\mathcal{J}}$, which is defined below, is different from $\mathcal{J}$. For every leaf node $\bar{h}$, if there exists an ancestor $\mathcal{W}$-move node $\bar{h}_q^i$ of $\bar{h}$ such that $\bar{h}_q^i$ is strong, then $\tilde{\mathcal{J}}(\bar{h}) = 0$. Otherwise, $\tilde{\mathcal{J}}(\bar{h}) = \mathcal{J}(\bar{h})$.*

Lemma 36 *Let $\mathcal{G} = (\mathcal{Q}, \mathcal{W}, \mathcal{J})$ be a game with $\mathcal{W}$ playing a fixed strategy $\mathcal{W}_t$. Let $\mathcal{S}(\cdot)$ be distributions such that for every $\mathcal{Q}$-move node $\bar{h}_w^{i-1}$, $\mathcal{S}(\bar{h}_w^{i-1})$ is a distribution over the next moves q_i. Let $\varepsilon \in (0, 1]$ be a real number. Let $\mathcal{Q}_{\mathrm{eff}}(\varepsilon)$ be the strategy defined in Definition 34 and $\tilde{\mathcal{G}}$ the corresponding modified game. We have $\gamma_{\mathrm{eff}} \ge \tilde{\gamma_{\mathrm{opt}}} - \varepsilon$.*

Proof. We prove the statement by induction on the level of the game tree from the leaf level. The inductive hypotheses are

$$\begin{cases} \gamma_{\mathrm{eff}}(\bar{h}_q^i) \ge \tilde{\gamma_{\mathrm{opt}}}(\bar{h}_q^i) - \varepsilon/4^i & \text{for every } \mathcal{W}\text{-move node } \bar{h}_q^i, \\ \gamma_{\mathrm{eff}}(\bar{h}_w^i) \ge \tilde{\gamma_{\mathrm{opt}}}(\bar{h}_w^i) - \varepsilon/4^i & \text{for every } \mathcal{Q}\text{-move node } \bar{h}_w^i. \end{cases}$$

The base case is trivial. For every leaf node $\bar{h}$ (of the form either $\bar{h}_w^m$ or $\bar{h}_q^m$), we have

$$\gamma_{\mathrm{eff}}(\bar{h}) = \mathcal{J}(\bar{h}) \ge \tilde{\mathcal{J}}(\bar{h}) \ge \tilde{\gamma_{\mathrm{opt}}}(\bar{h}) - \varepsilon/4^m.$$

There are two cases in the inductive step. Let us prove the simpler case first. Suppose that the inductive hypothesis is true for every $\mathcal{Q}$-move node $\bar{h}_w^i$ in the $2i$-th level. Then for every $\mathcal{W}$-move node $\bar{h}_q^i$ in the $(2i-1)$-st level, we have

$$\gamma_{\mathrm{eff}}(\bar{h}_q^i) = \mathop{\mathrm{E}}_{w_i}[\gamma_{\mathrm{eff}}(\bar{h}_q^i \circ w_i)] \ge \mathop{\mathrm{E}}_{w_i}[\tilde{\gamma_{\mathrm{opt}}}(\bar{h}_q^i \circ w_i) - (\varepsilon/4^i)] = \tilde{\gamma_{\mathrm{opt}}}(\bar{h}_q^i) - (\varepsilon/4^i),$$

where the middle inequality follows by the inductive hypothesis. Thus, the inductive hypothesis holds for the $(2i-1)$-st level.

Now, suppose that for every $\mathcal{W}$-move node $\bar{h}_q^i$ in the $(2i-1)$-th level, $\gamma_{\mathrm{eff}}(\bar{h}_q^i) \ge \tilde{\gamma_{\mathrm{opt}}}(\bar{h}_q^i) - \varepsilon/4^i$. We want to show that for every $\mathcal{Q}$-move node $\bar{h}_w^{i-1}$ in the $(2i-2)$-nd level,

$$\gamma_{\mathrm{eff}}(\bar{h}_w^{i-1}) \ge \tilde{\gamma_{\mathrm{opt}}}(\bar{h}_w^{i-1}) - \varepsilon/4^{i-1}.$$

Consider an arbitrary $\mathcal{Q}$-move node $\bar{h}_w^{i-1}$. Let T the set of children $(\bar{h}_w^{i-1} \circ q_i)$ of $\bar{h}_w^{i-1}$ with largest $\gamma_{\mathrm{eff}}(\bar{h}_q^i)$ among $(\bar{h}_w^{i-1} \circ q_i')$'s not in H. That is,

$$T = \{(\bar{h}_w^{i-1} \circ q_i) \notin H : \gamma_{\mathrm{eff}}(\bar{h}_w^{i-1} \circ q_i) = \max_{(\bar{h}_w^{i-1} \circ q_i') \notin H} \gamma_{\mathrm{eff}}(\bar{h}_w^{i-1} \circ q_i')\}.$$

The disired inequality follows by the following three claims.

Claim 37 *In the first step of $\mathcal{Q}_{\mathrm{eff}}$, with probability at least $1-(1-\varepsilon)^M$ over $q_{i,1}, \ldots, q_{i,M} \leftarrow \mathcal{S}(\bar{h}_w^{i-1})$, there exists a $j_0 \in [M]$ such that $\bar{h}_w^{i-1} \circ q_{i,j_0} \in H \cup T$. In this case, we call the first step is* good.

Proof of claim: Observe that

$$\Pr_{q_i \leftarrow \mathcal{S}(\bar{h}_w^{i-1})}[\bar{h}_w^{i-1} \circ q_i \in H \cup T] > \varepsilon.$$

Since $q_{i,1}, \ldots, q_{i,M}$ are sampled from $\mathcal{S}(\bar{h}_w^{i-1})$ independently, the claim follows. □

Claim 38 *In the second step of $\mathcal{Q}_{\text{eff}}$, with probability at least $1 - M \cdot 2^{-\Omega(M_i \cdot (\varepsilon/4^i)^2)}$, the estimation $\gamma_{\text{est}}(\bar{h}_w^{i-1} \circ q_{i,j})$ of $\gamma_{\text{eff}}(\bar{h}_w^{i-1} \circ q_{i,j})$ is within additive error $\varepsilon/4^i$ for every $j \in [M]$. That is,*

$$|\gamma_{\text{est}}(\bar{h}_w^{i-1} \circ q_{i,j}) - \gamma_{\text{eff}}(\bar{h}_w^{i-1} \circ q_{i,j})| \leq (\varepsilon/4^i) \text{ for every } j \in [M].$$

In this case, we call the second step is good.

Proof of claim: Follow by a standard Chernoff bound and a union bound. □

Claim 39 *Suppose in the execution of $\mathcal{Q}_{\text{eff}}$, both the first two steps are good. Then $\mathcal{Q}_{\text{eff}}$ will take a move q_i^* with $\gamma_{\text{eff}}(\bar{h}_w^{i-1} \circ q_i^*) \geq \tilde{\gamma_{\text{opt}}}(\bar{h}_w^{i-1}) - (3\varepsilon/4^i)$.*

Proof of claim: Let $q_{i,1}, \ldots, q_{i,M} \leftarrow \mathcal{S}(\bar{h}_w^{i-1})$ be the M samples drawn in the first step of $\mathcal{Q}_{\text{eff}}$. Since the first step is good, there exists a $j_0 \in [M]$ such that $\bar{h}_w^{i-1} \circ q_{i,j_0} \in H \cup T$. We first argue that $\gamma_{\text{eff}}(\bar{h}_w^{i-1} \circ q_{i,j_0}) \geq \tilde{\gamma_{\text{opt}}}(\bar{h}_w^{i-1}) - (\varepsilon/4^i)$. Note that $\tilde{\gamma_{\text{opt}}}(\bar{h}_w^{i-1} \circ q_i) = 0$ for every $\bar{h}_w^{i-1} \circ q_i \in H$ by our construction of $\tilde{\mathcal{G}}$, and $\tilde{\gamma_{\text{opt}}}(\bar{h}_w^{i-1} \circ q_i) \leq \gamma_{\text{eff}}(\bar{h}_w^{i-1} \circ q_i) + (\varepsilon/4^i)$ for every $\bar{h}_w^{i-1} \circ q_i$ by the induction hypothesis. We have

$$\begin{aligned}
\tilde{\gamma_{\text{opt}}}(\bar{h}_w^{i-1}) &= \max_{q_i : (\bar{h}_w^{i-1} \circ q_i) \notin H} \{\tilde{\gamma_{\text{opt}}}(\bar{h}_w^{i-1} \circ q_i)\} \\
&\leq \max_{q_i : (\bar{h}_w^{i-1} \circ q_i) \notin H} \{\gamma_{\text{eff}}(\bar{h}_w^{i-1} \circ q_i) + (\varepsilon/4^i)\} \\
&\leq \gamma_{\text{eff}}(\bar{h}_w^{i-1} \circ q_{i,j_0}) + (\varepsilon/4^i).
\end{aligned}$$

However, $\mathcal{Q}_{\text{eff}}$ takes q_i^* that maximize $\gamma_{\text{est}}(\bar{h}_w^{i-1} \circ q_{i,j})$, which may not be q_{i,j_0}. Nevertheless, since the second step is good, $|\gamma_{\text{est}}(\bar{h}_w^{i-1} \circ q_{i,j}) - \gamma_{\text{eff}}(\bar{h}_w^{i-1} \circ q_{i,j})| \leq (\varepsilon/4^i)$ for every $j \in [M]$. Therefore,

$$\begin{aligned}
\gamma_{\text{eff}}(\bar{h}_w^{i-1} \circ q_i^*) &\geq \gamma_{\text{est}}(\bar{h}_w^{i-1} \circ q_i^*) - (\varepsilon/4^i) \\
&\geq \gamma_{\text{est}}(\bar{h}_w^{i-1} \circ q_{i,j_0}) - (\varepsilon/4^i) \\
&\geq \gamma_{\text{eff}}(\bar{h}_w^{i-1} \circ q_{i,j_0}) - (2\varepsilon/4^i) \\
&\geq \tilde{\gamma_{\text{opt}}}(\bar{h}_w^{i-1}) - (3\varepsilon/4^i),
\end{aligned}$$

as desired. □

By the above claims and a union bound, with probability at least $1-(1-\varepsilon)^M - M \cdot 2^{-\Omega(M_i \cdot (\varepsilon/4^i)^2)} \geq 1 - \varepsilon/4^i$, $\mathcal{Q}_{\text{eff}}$ takes a move q_i^* with $\gamma_{\text{eff}}(\bar{h}_w^{i-1} \circ q_i^*) \geq \tilde{\gamma_{\text{opt}}}(\bar{h}_w^{i-1}) - (3\varepsilon/4^i)$. Thus,

$$\gamma_{\text{eff}}(\bar{h}_w^{i-1}) \geq \tilde{\gamma_{\text{opt}}}(\bar{h}_w^{i-1}) - (3\varepsilon/4^i) - \varepsilon/4^i = \tilde{\gamma_{\text{opt}}}(\bar{h}_w^{i-1}) - \varepsilon/4^{i-1},$$

as desired. Since the argument holds for every $\mathcal{Q}$-move node $\bar{h}_w^{i-1}$, the inductive hypothesis holds for the $(2i-2)$-nd level, which completes the proof. ■

C.1.2 Proof of Lemma 19

We introduce the following notation. Let S^n be a solver for $\mathsf{P}^{n,r}_{seq}$. For every $i \in [n]$, we use $\langle \mathsf{S}^n, \mathsf{P}_i \rangle$ to denote the i-th round repetition of $\langle \mathsf{S}^n, \mathsf{P}^{n,r}_{seq} \rangle$. We say that S^n succeeds on P_i if $\mathsf{P}^{n,r}_{seq}$ accepts the i-th round repeatition, and define indicator random variables $T_i = 1$ iff S^n succeeds on P_i. In this section, we prove the following concrete version of full-spectrum hardness theorem for sequential repetition of puzzle systems.

We recall that the theorem says that, for every puzzle system P, if there is a time T solver S^n solving the n-fold sequential repetition $\mathsf{P}^{n,r}_{seq}$ of P with probability greater than $P(n, r, \delta)$, then we can construct a solver S from S^n that can successfully solve one puzzle P with probability at least $\delta \cdot (1 - 1/\eta)$, where η is a slackness parameter. Furthermore, the reduction solver S runs in time polynomial in all parameters $n, \eta, 1/\delta, 1/P(n, r, \delta)$ and T. We remark that we use multiplicative slackness in both sides to make the proof cleaner.

We follow the outline described in the introduction to prove Lemma 19. We will define a game $\mathcal{G}$ played between S and P, analyze the success probability of an optimal strategy $\mathsf{S}_{\mathrm{opt}}$, and apply Lemma 36 to show that there exists an efficient strategy $\mathsf{S}_{\mathrm{eff}}$ succeeds with probability close to δ.

Let P be a puzzle system and $\mathsf{P}^{n,r}_{seq}$ the corresponding (n, r)-sequential puzzle system. Let S^n be a solver with success probability $succ_{\mathsf{P}^{n,r}_{seq}}[\mathsf{S}^n] \geq P(n, r, \delta)$ for $\mathsf{P}^{n,r}_{seq}$. Without loss of generality we can assume that S^n is deterministic (he can sample the best coin tosses and uses these particular fixed tosses.) Thus, the outcome of $\langle \mathsf{S}^n, \mathsf{P}^{n,r}_{seq} \rangle$ depends only on each P_i's private coin tosses p_i, which can be viewed as the *puzzle* of each P_i.

We now define a game $\mathcal{G}(\mathsf{S}^n) = (\mathsf{S}, \mathsf{P}, \mathcal{J})$ that captures a natural approach for S to solve P. The game $\mathcal{G}(\mathsf{S}^n)$ consists of only three rounds. In the first round, S picks some coordinate $i \in [n]$ and puzzles $p_1, \ldots, p_{i-1}$ as her move $(i; p_1, \ldots, p_{i-1})$. In the second round, P randomly pick a move p_i. Finally S picks the suffix $(p_{i+1}, p_{i+2}, \ldots, p_n)$. This corresponds to S simulates the first $i - 1$ rounds of $\langle \mathsf{S}^n, \mathsf{P}^{n,r}_{seq} \rangle$ using $p_1, \ldots, p_{i-1}$ as $\mathsf{P}^{n,r}_{seq}$'s coins and then simulates the i-th round $\langle \mathsf{S}^n, \mathsf{P}_i \rangle$ with real P. Finally S simulates the last $n - i$ rounds using the suffix. We define $\mathcal{J}(i; p_1, \ldots, p_{i-1}, p_i, p_{i+1}, \ldots p_n) = T_i(p_1, \ldots, p_n)$, since S succeeds iff S^n succeeds on P_i.

The next step is to analyze the optimal strategy $\mathsf{S}_{\mathrm{opt}}$. We prove the following lemma, which says that if $succ_{\mathsf{P}^{n,r}_{seq}}[\mathsf{S}^n] \geq P(n, r, \delta)$, then the optimal strategy has success probability $succ_{\mathsf{P}}[\mathsf{S}_{\mathrm{opt}}] \geq \delta$.

Lemma 40 *Let n, k be positive integers with $k \leq n$, and $\delta \in (0, 1)$ a real number. Let P be a puzzle system, $\mathsf{P}^{n,r}_{seq}$ the corresponding (n, k)-sequential puzzle system, and S^n a deterministic solver. Let $\mathcal{G}(\mathsf{S}^n)$ be the corresponding game defined as above and $\mathsf{S}_{\mathrm{opt}}$ be an optimal strategy. If $succ_{\mathsf{P}^{n,r}_{seq}}[\mathsf{S}^n] \geq P(n, r, \delta)$, then $succ_{\mathsf{P}}[\mathsf{S}_{\mathrm{opt}}] \geq \delta$.*

Proof. We will prove the contrapositive statement. Let us consider the success probability $\gamma_{\mathrm{opt}}(\cdot)$ of $\mathsf{S}_{\mathrm{opt}}$ at each node of the game tree. We have

$$\begin{cases} \gamma_{\mathrm{opt}}(i; p_1, \ldots, p_n) = T_i|_{(p_1, \ldots, p_n)} & \text{for each leaf node } (i; p_1, \ldots, p_n). \\ \gamma_{\mathrm{opt}}(i; p_1, \ldots, p_{i-1}, p_i) = 1 \text{ iff } \exists\ (p_{i+1}, p_{i+2}, \ldots, p_n)\ s.t.\ T_i|_(p_1, p_2, \ldots, p_n) = 1;\ \text{otherwise } 0. \\ \text{for the second } \mathsf{S}\text{-move node } (i; p_1, \ldots, p_{i-1}, p_i), \\ \gamma_{\mathrm{opt}}(i; p_1, \ldots, p_{i-1}) = \mathrm{E}_{p_i}[\gamma_{\mathrm{opt}}(i; p_1, \ldots, p_i)] & \text{for each } \mathsf{P}\text{-move node } (i; p_1, \ldots, p_{i-1}), \\ \gamma_{\mathrm{opt}} = \max_{(i; p_1, \ldots, p_{i-1})}\{\gamma_{\mathrm{opt}}(i; p_1, \ldots, p_{i-1})\} & \text{for the root.} \end{cases}$$

Thus, $\gamma_{\mathrm{opt}}(i; p_1, \ldots, p_{i-1}) \leq \gamma_{\mathrm{opt}}$ for every $i \in [n]$ and every $p_1, \ldots, p_{i-1}$. Given this, we claim that the random variables $(T_1, \ldots, T_n)$ can be coupled with i.i.d. binary random variables $(R_1, \ldots, R_n)$

with $\Pr[R_i = 1] = \gamma_{\mathrm{opt}}$ such that $T_i \le R_i$ for every i with probability 1. It follows that

$$succ_{\mathsf{P}_{seq}^{n,r}}[\mathsf{S}^n] = \Pr\left[\sum_i T_i \ge r\right] \le \Pr\left[\sum_i R_i \ge r\right] = P(n,r,\gamma) < P(n,r,\delta),$$

as desired. Note that the last inequality holds because $f(\alpha) \stackrel{\text{def}}{=} P(n,r,\alpha)$ is a strictly increasing function in α for every n and $r \in [n]$.

It remains to prove the claim. First we define the conditional random variable $R'_i|_{(p_1,\ldots,p_{i-1})} = 1$ if there exists a suffix $p_i, p_{i+2}, \ldots, p_n$ such that $T|_{p_1,p_2,\ldots,p_n} = 1$, and otherwise 0. It is easy to see that $T_i \le R'_i$ conditioning on all possible prefixes. On the other hand, since $\gamma_{\mathrm{opt}}(i; p_1, \ldots, p_{i-1}) = \Pr_{p_i}[R'_i|_{p_1,\ldots,p_{i-1}} = 1]$ by the construction of R'_i, and thus we have $\Pr_{p_i}[R'_i|_{p_1,\ldots,p_{i-1}} = 1] \le \gamma_{\mathrm{opt}}$. Then we define R_i by defining the conditional random variable $R_i|_{(p_1,\ldots,p_{i-1})}$ for every $i \in [n]$ and every $p_1, \ldots, p_{i-1}$. We want $R_i|_{(p_1,\ldots,p_{i-1})}$ to satisfy (i) $R'_i|_{(p_1,\ldots,p_{i-1})} \le R_i|_{(p_1,\ldots,p_{i-1})}$ and (ii) $\Pr[R_i|_{(p_1,\ldots,p_{i-1})} = 1] = \gamma_{\mathrm{opt}}$, which is doable because $\Pr[R'_i|_{(p_1,\ldots,p_{i-1})}] \le \gamma_{\mathrm{opt}}$. For every $i \in [n]$ and every $p_1, \ldots, p_i$, if $R'_i|_{(p_1,\ldots,p_{i-1})} = 1$, we set $R_i|_{(p_1,\ldots,p_{i-1})} = 1$, and if $R'_i|_{(p_1,\ldots,p_{i-1})} = 0$, we toss independent coins and set $R_i|_{(p_1,\ldots,p_{i-1})} = 1$ with probability $(\gamma_{\mathrm{opt}} - \gamma(i; p_1, \ldots, p_{i-1}))/(1 - \gamma(i; p_1, \ldots, p_{i-1}))$. It is easy to verify that (i) and (ii) are satisfied for every i and $p_1, \ldots, p_{i-1}$, which implies $T_i \le R_i$ for every $i \in [n]$ with probability 1 and $\Pr[R_i = 1] = \gamma_{\mathrm{opt}}$ for every $i \in [n]$. To check the independence, for every $r_1, \ldots, r_{i-1} \in \{0,1\}$, we have

$$\begin{aligned}
&\Pr[R_i = 1 | R_1 = r_1, \ldots, R_{i-1} = r_{i-1}] \\
&= \mathop{\mathrm{E}}_{(p_1,\ldots,p_{i-1})}[\Pr[R_i = 1 | p_1, \ldots, p_{i-1}, R_1 = r_1, \ldots, R_{i-1} = r_{i-1}]] \\
&= \mathop{\mathrm{E}}_{(p_1,\ldots,p_{i-1})}[\Pr[R_i = 1 | p_1, \ldots, p_{i-1}]] \\
&= \gamma_{\mathrm{opt}},
\end{aligned}$$

where the second-to-third line follows because once we conditioning on $p_1, \ldots, p_{i-1}$, R_i is independent of $R_1, \ldots, R_{i-1}$ by our construction. ■

The next step is to use Lemma 36 to show that if there is an efficient deterministic solver S^n with $succ_{\mathsf{P}_{seq}^{n,r}}[\mathsf{S}^n] \ge P(n,r,\delta)$, then there is an efficient strategy $\mathsf{S}_{\mathrm{eff}}$ that succeeds with probability at least $\delta \cdot (1 - 1/\eta)$ by the following argument.

- For the S-move nodes of the game tree, the root, define an efficiently samplable distribution $\mathcal{S}$ over the possible S moves, and set the parameter $\varepsilon = \min\{\delta, P(n,r,\delta)\}/(n\eta)$. This defines a strategy $\mathsf{S}_{\mathrm{eff}}$ and a modified game $\tilde{\mathcal{G}}$ by Definition 34 and 35.
- Define a modified solver $\tilde{\mathsf{S}}^n$ from S^n such that $succ_{\mathsf{P}_{seq}^{n,r}}[\tilde{\mathsf{S}}^n] \ge P(n,r,\delta)(1 - 1/2\eta)$ and the corresponding game $\mathcal{G}(\tilde{\mathsf{S}}^n) = \tilde{\mathcal{G}}$. It follows by Lemma 40 that $succ_{\mathsf{P}}[\tilde{\mathsf{S}_{\mathrm{opt}}}] \ge \delta$ and by Lemma 36 that $succ_{\mathsf{P}}[\mathsf{S}_{\mathrm{eff}}] \ge succ_{\mathsf{P}}[\tilde{\mathsf{S}_{\mathrm{opt}}}] - \varepsilon \ge \delta \cdot (1 - 1/\eta)$.

Lemma 41 *Let $n, r, \eta, T : \mathbb{N} \to \mathbb{N}$ and $\delta : \mathbb{N} \to [0,1]$ be efficiently computable functions with $r \le n$. Let P be a puzzle system, $\mathsf{P}_{seq}^{n,r}$ the corresponding (n,r)-sequential puzzle system, and S^n a* deterministic *solver such that $\langle \mathsf{S}^n, \mathsf{P}_{seq}^{n,r} \rangle(1^s)$ runs in time $T(s)$. If $succ_{\mathsf{P}_{seq}^{n,r}}[\mathsf{S}^n] \ge P(n,k,\delta) \cdot (1 - 1/\eta)$, then there exists a solver S for P runs in time* $\mathrm{poly}(n, q, \delta^{-1}, P(n,r,\delta)^{-1}, T)$ *with success probability $succ_{\mathsf{P}}[\mathsf{S}] \ge \delta \cdot (1 - 1/\eta)$.*

Proof. Let $\mathcal{G}(\mathsf{S}^n)$ be the corresponding game. We consider the solver $\mathsf{S}_{\mathrm{eff}}$ that solves P by playing the game $\mathcal{G}(\mathsf{S}^n)$ with P as defined in Definition 34. To specify $\mathsf{S}_{\mathrm{eff}}$, we need to set the

parameter ε and define distributions $\mathcal{S}(\cdot)$ for every S-move nodes in the game tree. We set $\varepsilon = \min\{\delta, P(n,r,\delta)\}/(n\eta)$. Note that $\mathcal{G}(\mathsf{S}^n)$ has three rounds. The corresponding distribution $\mathcal{S}$ over $\{(i; p_1, \ldots, p_{i-1})\}$ is as follows. $i \leftarrow [n]$ is chosen uniformly at random, and $(p_1, \ldots, p_{i-1})$ is generated by the first $(i-1)$ rounds of $\langle \mathsf{S}^n, \mathsf{P}^{n,r}_{seq} \rangle$. The next move's ($\mathsf{P}$-move) distribution is the i-round of the repetition $\mathsf{P}^{n,r}_{seq}$ and it outputs the puzzle p_i. Finally the last move is the suffix $(p_{i+1}, \ldots, p_n)$ of the system $\mathsf{P}^{n,r}_{seq}$. More precisely, the corresponding solver S_{eff} is as follows.

1. Let $\varepsilon = \min\{\delta, P(n,k,\delta)\}/(n\eta)$ and $M = \Theta((1/\varepsilon) \cdot (\log(1/\varepsilon)))$. For every $j \in [M]$, independently sample $i_j \leftarrow [n]$ uniformly at random and generate $(p_{1,j}, \ldots, p_{i_j-1,j})$ by simulating $\langle \mathsf{S}^n, \mathsf{P}^{n,r}_{seq} \rangle$ for $(i-1)$ rounds.

2. For every $j \in [M]$, estimate the success probability $\gamma_{\text{eff}}(i_j; p_{1,j}, \ldots, p_{i_j-1,j})$ by sampling as follows. Let $M' = \Theta((1/\varepsilon)^2 \cdot (\log M + \log(1/\varepsilon)))$. Independently simulate M' times the interaction $\langle \mathsf{S}^n, \mathsf{P}_i \rangle$ with history $(p_{1,j}, \ldots, p_{i_j-1,j})$. Let $\gamma_{\text{est}}(i_j; p_{1,j}, \ldots, p_{i_j-1,j})$ be the fraction of simulation that S^n succeeds on P_i.

3. Let $j^* = \text{argmax}_j\{\gamma_{\text{est}}(i_j; p_{1,j}, \ldots, p_{i_j-1,j})\}$. Let the real puzzle P plays the role of $\mathsf{P}_{i_{j^*}}$, and solve P by simulating $\langle \mathsf{S}^n, \mathsf{P}_{i_{j^*}} \rangle$ with history $(p_{1,j^*}, \ldots, p_{i_{j^*}-1,j^*})$.

In other words, S_{eff} selects a good prefix puzzles $(p_1, \ldots, p_{i-1})$ by sampling, and solves the real puzzle P using S^n with P playing the role of P_i and history $(p_1, \ldots, p_{i-1})$. Note that S_{eff} runs in time $\text{poly}(n, \eta, \delta^{-1}, P(n,r,\delta)^{-1}, T)$, as desired.

Now, recall that the corresponding modified game $\tilde{\mathcal{G}}$ is obtained by "turning off" all strong P-move nodes, where a P-move node $(i; p_1, \ldots, p_{i-1})$ is strong if it has the ε-fraction of largest $\gamma_{\text{eff}}(i; p_1, \ldots, p_{i-1})$ among all P-move nodes. That is,

$$\Pr_{(i'; p'_1, \ldots, p'_{i-1}) \leftarrow \mathcal{S}} \left[\gamma_{\text{eff}}(i; p_1, \ldots, p_{i-1}) \leq \gamma_{\text{eff}}(i'; p'_1, \ldots, p'_{i-1}) \right] \leq \varepsilon.$$

For every leaf node $(i; p_1, \ldots, p_i)$, if its P-move parent $(i; p_1, \ldots, p_{i-1})$ is strong, then $\tilde{\mathcal{J}}(i; p_1, \ldots, p_i) = 0$; otherwise, $\tilde{\mathcal{J}}(i; p_1, \ldots, p_i) = \mathcal{J}(i; p_1, \ldots, p_i) = T_i|_{(p_1, \ldots, p_i)}$. Let $\tilde{\mathsf{S}}^n$ be a solver for $\mathsf{P}^{n,r}_{seq}$ and $\tilde{T}_1, \ldots, \tilde{T}_n$ the indicator random variables such that $\tilde{T}_i = 1$ iff $\tilde{\mathsf{S}}^n$ succeeds on P_i. Note that if $\tilde{T}_i|_{(p_1, \ldots, p_i)} = 0$ when $(i; p_1, \ldots, p_i)$ is strong and otherwise, $\tilde{T}_i|_{(p_1, \ldots, p_i)} = T_i|_{(p_1, \ldots, p_i)}$, then $\tilde{\mathcal{G}} = \mathcal{G}(\tilde{S}^n)$.[6] Thus, we can lower bound the success probability of $\tilde{\mathsf{S}}_{\text{opt}}$ by lower bounding the success probability of $\tilde{\mathsf{S}}^n$ and applying Lemma 40.

Claim 42 *The statistical distance* $\Delta((T_1, \ldots, T_n), (\tilde{T}_1, \ldots, \tilde{T}_n)) \leq n \cdot \varepsilon \leq 1/\eta$, *and thus*

$$succ_{\mathsf{P}^{n,r}_{seq}}[\tilde{\mathsf{S}}^n] \geq succ_{\mathsf{P}^{n,r}_{seq}}[\mathsf{S}^n] - 1/\eta \geq P(n,r,\delta).$$

Proof of claim: Note that for every i and $p_1, \ldots, p_i$, $T_i|_{(p_1, \ldots, p_i)} \neq \tilde{T}_i|_{(p_1, \ldots, p_i)}$ only when $(i; p_1, \ldots, p_{i-1})$ is strong, and the fraction of strong nodes is at most ε. That is,

$$\Pr_{(i; p_1, \ldots, p_{i-1}) \leftarrow \mathcal{S}} [(i; p_1, \ldots, p_{i-1}) \in H] \leq \varepsilon.$$

[6] One realization of such $\tilde{S}^n$ is to let $\tilde{S}^n$ to behave exactly the same as S^n but for each $i \in [n]$ abort the interaction with P_i when $(i; p_1, \ldots, p_{i-1})$ is strong. Note that such $\tilde{S}^n$ may not be efficient and needs to know the private coins of $\mathsf{P}^{n,r}_{seq}$ to decide whether to abort or not. Nevertheless, Lemma 40 is still applicable.

Intuitively, $\vec{T} = (T_1, \ldots, T_n)$ and $\tilde{\vec{T}} = (\tilde{T}_1, \ldots, \tilde{T}_n)$ can only differ on at most $(n \cdot \varepsilon)$-fraction of probability mass because $\Pr[(i; p_1, \ldots, p_{i-1})] = (1/n) \cdot \Pr[(p_1, \ldots, p_{i-1})]$. Formally, we have

$$\begin{aligned}
\Delta(\vec{T}, \tilde{\vec{T}}) &\leq \Pr[\vec{T} \neq \tilde{\vec{T}}] \\
&\leq \Pr[\exists i \in [n] \text{ s.t. } (i; p_1, \ldots, p_{i-1}) \in H] \\
&\leq \sum_i \Pr[(i; p_1, \ldots, p_{i-1}) \in H] \\
&= n \cdot \Pr_{(i;p_1,\ldots,p_{i-1}) \leftarrow \mathcal{S}}[(i; p_1, \ldots, p_{i-1}) \in H] \\
&\leq n \cdot \varepsilon
\end{aligned}$$

□

Putting things together, by Lemma 40 and 36, we have

$$succ_{\mathsf{P}}[\mathsf{S}_{\text{eff}}] \geq succ_{\mathsf{P}}[\tilde{\mathsf{S}}_{\text{opt}}] - \varepsilon \geq \delta \cdot (1 - 1/\eta),$$

as desired. ■

Lemma 19 follows easily from Lemma 30 and 41.

Proof. (of Lemma 19) By Lemma 30 with slackness parameter $\min\{\delta, P(n,k,\delta)\}/(2\eta)$, there exists a solver $\hat{\mathsf{S}}^n$ such that $\langle \hat{\mathsf{S}}^n, \mathsf{P}^{n,r}_{seq} \rangle$ runs in time $\text{poly}(\eta, \delta^{-1}, P(n,r,\delta)^{-1}, T)$ and with probability at least $1 - \delta/(2\eta)$ over $\hat{\mathsf{S}}^n$'s random coins c, the deterministic solver $\hat{\mathsf{S}}^n_c$ obtained by fixing $\hat{\mathsf{S}}^n$'s coins to c has success probability $succ_{\mathsf{P}^{n,r}_{seq}}[\hat{\mathsf{S}}^n_c] \geq P(n,k,\delta) \cdot (1 - 1/(2\eta))$. Consider a solver S that samples random coins c for $\hat{\mathsf{S}}^n$, and solves P by playing strategy $\mathsf{S}_{\text{eff},c}$ in the corresponding game $\mathcal{G}(\hat{\mathsf{S}}^n_c)$. It is not hard to see that S runs in time $\text{poly}(n, \eta, \delta^{-1}, P(n,r,\delta)^{-1}, T)$. By Lemma 41 with slackness parameter 2η, if $succ_{\mathsf{P}^{n,r}_{seq}}[\hat{\mathsf{S}}^n_c] \geq P(n,r,\delta) \cdot (1 - 1/(2\eta))$, then $succ_{\mathsf{P}}[\mathsf{S}_{\text{eff},c}] \geq \delta \cdot (1 - 1/(2\eta))$. Since $succ_{\mathsf{P}^{n,r}_{seq}}[\hat{\mathsf{S}}^n_c] \geq P(n,r,\delta) \cdot (1 - 1/(2\eta))$ with probability at least $1 - \delta/(2\eta)$ over c, it follows that

$$succ_{\mathsf{P}}[\mathsf{S}] \geq \delta \cdot (1 - 1/(2\eta)) - \delta/(2\eta) = \delta \cdot (1 - 1/\eta).$$

■

C.2 General Two-Phase Model

In this section, we consider a more general two-phase model where the puzzle systems are only weakly verifiable. We wonder whether Full Spectrum Hardness can be achieved here where the solver cannot verify the answer to the puzzle not generated by himself. Let S^n be the solver with success probability at least $P(n,r,\delta)$ of solving $\mathsf{P}^{n,r}_{seq}$, from this our goal is to construct another solver S that solves P with success probability δ. By the same argument as previous, without loss of generality we assume S^n to be deterministic.

In the first place, it is natural to look at the previous framework (i.e. Fully verifiable Two-Phase system) and see if the same reasoning also applies. To our observation, there are some technical barriers with which it is unclear how the same framework can apply. Recall in this framework, given a solver S^n for $\mathsf{P}^{n,r}_{seq}$, we then get an induced game $\mathcal{G} = (\mathsf{S}, \mathsf{P}, \mathcal{J})$ for the single puzzle system P and a solver S. Here the judge $\mathcal{J}$'s rule depends on the behavior of S^n that whether he solves the n-fold puzzle. The game consists of the following moves. First the S finds a prefix, then P gives a puzzle, then S finds a suffix. Then S succeeds iff $\mathcal{J}$ on these moves outputs 1. Then we construct

the players S_{opt} and $\mathsf{S}_{\mathit{eff}}$ who play the optimal strategy and the efficient strategy respectively and relate the success probabilities of those, and argue that if S^n has high success probability to $\mathsf{P}^{n,r}_{seq}$, then the player $\mathsf{S}_{\mathit{eff}}$ also has high success probability.

Here in the weakly verifiable case, the judge's rule is different from that for the verifiable case since whether the answer to any puzzle is correct or not depends on the puzzle generator's secret coins. Thus for the same answer to a particular answer, the judge may give different results. Thus, the judge must be probabilistic. Here by the same argument in Lemma 36, we can argue that the optimal strategy still has high success probability. However, when we want to approximate it by sampling, it is unclear how to make it. We observe that in this approach, the judge cannot be efficiently implementable. This is because no efficient algorithm can sample P's random coins that are consistent to the puzzle it generates, and then makes the judgement according to the consistent coins. This results in the failure of the sampling technique of $\mathsf{S}_{\mathit{eff}}$ in that we cannot efficiently estimate the success probability at each node. To be more concrete, after a prefix and a puzzle are chosen, it is not clear how S estimates the success probability and cannot choose the best suffix consequently.

Although it seems hard to bypass the barrier in the above direction, there are other techniques used in the previous works and had positive results for some special cases. In the above, we identified the barrier that it is hard to estimate the success probability of any particular suffix. Canetti, Halevi, and Steiner [CHS05] gave a new idea that chooses a good suffix by "*conditioning*." That is, even though we cannot estimate the success probability of each suffix, we believe that S^n gives a correct answer conditioning on some particular event with the suffix. They gave an algorithm that finds a good prefix, and with conditioning, they proved the Direct Product Theorem. Followed by a similar idea by Halevi and Rabin [HR08], the Hardness Degradation can also be achieved. We summarize these two algorithms and give a high level analysis to see why this idea works.

For the Direct Product Theorem [CHS05], the algorithm firstly uses recursion to find a good prefix. Let $\mathcal{E}_i(k)$ be the event that S^n solves k puzzles over $\{p_i, p_{i+1}, \ldots, p_n\}$, and $\Pr[\mathcal{E}_i(k)]$ is the probability that this event happens with the randomness over the rest of the puzzles $\{p_i, p_{i+1}, \ldots, p_n\}$. Starting from prefix $i = 1, \vec{p} = \emptyset$, the recursion iteratively extends it by the condition: if there exists a puzzle p_i such that $\Pr[\mathcal{E}_{i+1}(n-i)] \geq \delta^{n-i}$, then $\vec{p} = \vec{p} \circ \{p_i\}$, $i = i + 1$ and it recurs. We observe that when the algorithm recurs, it reduces the problem to the smaller subproblem with the same structure, (i.e. S^n with the prefix solves the other puzzles with probability better than δ^{n-i}, and this is the smaller subproblem.) On the other hand, when the recursion condition does not hold, we claim $\vec{p}$ is the good prefix. Then with this prefix, the solver S embeds the puzzle from the P, then he finds a suffix conditioning on the event $\mathcal{E}_{i+1}(n-i-1)$, which means conditioning on S solves *all* of the puzzles in the suffix, the algorithm outputs S^n's answer to the real puzzle.

Here we give a high level analysis of this algorithm. Let $b \in \{0, 1, *\}$, and for convenience we introduce another similar notation $\mathcal{E}_i(b, k)$ to be the event that S^n solves k puzzles among $p_{i+1}, p_{i+2}, \ldots, p_n$ and with $b = 0$ it fails to solve p_i, $b = 1$ it succeeds, and $b = *$, it does not care. Observe that the recursion deals with the problem of the same structure, so we analyze the case when the recursion stops at the first place, i.e. $i = 1, \vec{p} = \emptyset$. With the same reasoning, we can handle the case whenever the recursion stops. Then we claim that the success probability of the algorithm is $\mathrm{E}_{p_1}[\mathcal{E}_1(1,*)|\mathcal{E}_1(*,n-1)] = \mathrm{E}_{p_1}\left[\frac{\mathcal{E}_1(1,n-1)}{\mathcal{E}_1(*,n-1)}\right]$. This is because for every p_1 the algorithm has success probability $\Pr\left[\frac{\mathcal{E}_1(1,n-1)}{\mathcal{E}_1(*,n-1)}\middle| p_1\right]$ of solving it correctly (by conditioning), and thus the average value over p_1 is the success probability of the algorithm.

At the beginning, we have (i) $\Pr[\mathcal{E}_1(1, n-1)] \geq \delta^n$ from the assumption. Since the recursion

stops at $i = 1$, we then have (ii) for any first puzzle p_1, $\Pr[\mathcal{E}_1(*, n-1)|p_1] \leq \delta^{n-1}$. Then we have

$$\mathop{\mathrm{E}}_{p_1}\left[\frac{\mathcal{E}_1(1,n-1)}{\mathcal{E}_1(*,n-1)}\right] \geq \mathop{\mathrm{E}}_{p_1}\left[\frac{\mathcal{E}_1(1,n-1)}{\delta^{n-1}}\right] \geq \frac{\delta^n}{\delta^{n-1}} = \delta,$$

where the first inequality is by (ii), and the second is by (i). This gives the result we desire for the Direct Product Theorem.

For the other case of Hardness Degradation in [HR08], the authors uses the same framework with some variations. Still the algorithm first finds a good prefix by recursion. Starting from prefix $i = 1, \vec{p} = \emptyset$, the recursion iteratively extends it by the condition: if there exists a puzzle p_i such that $\Pr[\mathcal{E}_{i+1}(\geq 1)] \geq 1 - (1-\delta)^{n-i}$, then $\vec{p} = \vec{p} \circ \{p_i\}$, $i = i + 1$ and it recurs. On the other hand, when the recursion condition does not hold, we claim $\vec{p}$ is the good prefix. Then with this prefix, the solver S embeds the puzzle from the P, then he finds a suffix conditioning on the event $\mathcal{E}_{i+1}(0)$, which means conditioning on S solves *none* of the puzzles in the suffix, the algorithm outputs S^n's answer to the real puzzle.

With the same reason for the Direct Product Theorem, we consider the case when the recursion stops in the beginning $i = 1$, and then we have the success probability $\mathrm{E}_{p_1}[\mathcal{E}_1(1,*)|\mathcal{E}_1(*,0)] = \mathrm{E}_{p_1}\left[\frac{\mathcal{E}_1(1,0)}{\mathcal{E}_1(*,0)}\right]$. At the beginning we have (i) $\Pr[\mathcal{E}_1(1,0) + \mathcal{E}_1(*,\geq 1)] \geq 1 - (1-\delta)^n$ from the assumption. Then since the recursion stops, we have (ii) for all p_1, $\Pr[\mathcal{E}(*,\geq 1)|p_1] < 1-(1-\delta)^{n-1}$. From (ii) we have for all p_1, $\Pr[\mathcal{E}_1(0,0)|p_1] = 1 - \Pr[\mathcal{E}_1(*,\geq 1)|p_1] > (1-\delta)^{n-1}$. By (i) and (ii) we have $\Pr[\mathcal{E}_1(1,0)] \geq 1 - (1-\delta)^n - (1-(1-\delta)^{n-1}) = \delta \cdot (1-\delta)^{n-1}$. Then we have the success probability, and Jensen's inequality:

$$\mathop{\mathrm{E}}_{p_1}\left[\frac{\mathcal{E}_1(1,0)}{\mathcal{E}_1(*,0)}\right] = \mathop{\mathrm{E}}_{p_1}\left[\frac{\mathcal{E}_1(1,0)}{\mathcal{E}_1(1,0) \cup \mathcal{E}(0,0)}\right] \geq \mathop{\mathrm{E}}_{p_1}\left[\frac{\mathcal{E}_1(1,0)}{\mathcal{E}_1(1,0) + (1-\delta)^{n-1}}\right] \geq \frac{\mathrm{E}[\mathcal{E}_1(1,0)]}{\mathrm{E}[\mathcal{E}_1(1,0)] + (1-\delta)^{n-1}} \geq \delta.$$

These two directions gave positive results in the two special cases; however, there are no direct clues of how we can generalize it to the case r between 2 to $n-1$. First, the recursion condition is not clear. Which subproblem the algorithm should recur to is not clear, since the original condition seems not to work directly. The other problem is which condition the algorithm believes the output to be correct, and the previous conditions that all the other puzzles are simultaneously solved correctly or incorrectly at the same time seem not work in the general case.

In this work, we find the correct recursive subproblems and the corresponding conditional event that the algorithm believes the answer to be correct. We slightly modify the previous recursion that in the generalized case, it occurs with either of the following two facts holds: for $n \in \mathbb{N}, r \in [n]$, starting from the empty prefix $\vec{p} = \emptyset, i = 1$, the algorithm checks if there exists a p_i such that (1) $\Pr[\mathcal{E}_i(*,k)|\vec{p},p_i] \geq P(n-i,r,\delta)$, or (2) $\Pr[\mathcal{E}_i(*,k-1)|\vec{p},p_i] \geq P(n-i,r-1,\delta)$, then the algorithm recurs with $\vec{p} = \vec{p} \circ p_i, i = i+1$ (to the case that the condition holds). If both conditions fail, then it claims $\vec{p}$ is the good prefix, and it believes the answer from S^n if it solves exactly $r-1$ puzzles in the remaining puzzles, i.e. conditioning on the event $\mathcal{E}_i(*,r-1)$.

In the following, we are going to argue the above scheme succeeds with probability more than δ given the S^n has success probability more than $P(n,r,\delta)$. Here we first give a formal definition of the events and the main theorem as follows:

Definition 43 *Let $n, r, i, t \in \mathbb{N}, b \in \{0,1,*\}$, P be a two-phase puzzle system and $\mathsf{P}^{n,r}_{seq}$ the corresponding (n,r)-sequential two-phase puzzle system. Let S^n be the solver to $\mathsf{P}^{n,r}_{seq}$. Given an n-fold puzzle $(p_1, p_2, \ldots, p_n)$ from $\mathsf{P}^{n,r}_{seq}$, define the following events:*

1. *$\mathcal{E}_i(\geq t)$: S^n solves* at least *t puzzles over $\{p_i, p_{i+1}, \ldots, p_n\}$.*

2. $\mathcal{E}_i(t)$: S^n *solves* exactly t *puzzles over* $\{p_i, p_{i+1}, \ldots, p_n\}$

3. $\mathcal{E}_i(b, \geq t)$: S^n *solves* at least t *puzzles over* $\{p_{i+1}, p_{i+2}, \ldots, p_n\}$; *and it solves* p_i *if* $b = 1$; *it does not if* $b = 0$; *else* $b = *$, $\mathcal{E}_i(*, \geq t) = \mathcal{E}_i(1, \geq t) \cup \mathcal{E}_i(0, \geq t)$.

4. $\mathcal{E}_i(b, t)$: S^n *solves* exactly t *puzzles over* $\{p_{i+1}, p_{i+2}, \ldots, p_n\}$; *and it solves* p_i *if* $b = 1$; *it does not if* $b = 0$; *else* $b = *$, $\mathcal{E}_i(*, t) = \mathcal{E}_i(1, t) \cup \mathcal{E}_i(0, t)$.

Let $\vec{p}_{i-1} = (p_1, p_2, \ldots, p_{i-1})$ *be a prefix with length* $i-1$, *and* p_i *be the next puzzle. We define the conditional probability* $T(i, \vec{p}_{i-1} \circ p_i; \mathcal{E}_{i+1}(\geq t)) \stackrel{\text{def}}{=} \Pr[\mathcal{E}_{i+1}(\geq t) | \vec{p}_{i-1} \circ p_i]$. *Similarly, the other types of events defined as above can be the parameters of* T *in the same way.*

Now we are going to prove Lemma 18. Similarly we first consider a lemma in below that considers the solver S^n to be deterministic, and it follows from the argument of Lemma 30 that this is without loss of generality. Specifically, we are going to prove the lemma below:

Lemma 44 *Let* $n, r, \eta, T : \mathbb{N} \to \mathbb{N}$, *and* $\delta : \mathbb{N} \to [0, 1]$ *be efficiently computable functions with* $r \leq n$. *Let* P *be a two-phase puzzle system and* $\mathsf{P}^{n,r}_{seq}$ *the corresponding* (n, r)*-sequential two-phase puzzle system. Let* S^n *be a deterministic solver for* $\mathsf{P}^{n,r}_{seq}$ *such that* $\langle \mathsf{S}^n, \mathsf{P}^{n,r}_{seq} \rangle (1^s)$ *runs in time* $T(s)$, *and* $succ_{\mathsf{P}^{n,r}_{seq}}[\mathsf{S}^n] \geq P(n, r, \delta)$. *Then there exists a solver* S *for* P *that achieves* $\langle \mathsf{S}, \mathsf{P} \rangle (1^s)$ *running in time* $T'(s) = \mathrm{poly}(n, r, \eta, 1/\delta^r, T)$ *and* $succ_{\mathsf{P}}[\mathsf{S}] \geq \delta \cdot (1 - 1/\eta^{0.99})$. *In particular the solver is defined in the below.*

Given a deterministic solver S^n that has success probability at least $P(n, r, \delta)$, we construct a reduction algorithm as follow.

Definition 45 *From the premises of the lemma above, we define a reduction algorithm* S. *Let* $i = 1, \vec{p} = \emptyset$ *as the initial condition. Then on input* $(i, \vec{p}, n, k)$ *and oracle access to* S^n, S *does:*

- *repeat the following procedure: (prefix finding)*

  1. *independently sample* $M = O(\frac{\eta}{\delta} \log(4\eta n/\delta))$ p_i*'s denoted as* $p_{i,1}, p_{i,2}, \ldots, p_{i,M}$
  2. *for each* $j \in [M]$, *let* $\eta_i = P(n - i, r, \delta)/\delta$, *and then use* $M'_i = O(\eta_i^2 \log(4\eta n M/\delta))$ *independent samples to estimate the probability of the events* $\mathcal{E}_i(*, \geq r)$ *and* $\mathcal{E}_i(*, \geq r-1)$ *conditioning on* $\vec{p} \circ p_{i,j}$, *denoted as* $\tilde{T}(i, \vec{p} \circ p_{i,j}; \mathcal{E}_i(*, \geq r))$, *and* $\tilde{T}(i, \vec{p} \circ p_{i,j}; \mathcal{E}_i(*, \geq r-1))$
  3. *if there exists a* j^* *such that the estimation of* $\tilde{T}(i, \vec{p} \circ p_{i,j^*}; \mathcal{E}_i(*, \geq r)) \geq P(n - i, r, \delta)$ *then* $\vec{p} = \vec{p} \circ p_{i,j^*}, i = i + 1$; *else if that* $\tilde{T}(i, \vec{p} \circ p_{i,j^*}; \mathcal{E}_i(*, \geq r-1)) \geq P(n - i, r - 1, \delta)$ *then* $\vec{p} = \vec{p} \circ p_{i,j^*}, i = i + 1, r = r - 1$; *then repeat with the new values of* $(i, \vec{p}, n, r)$
  4. *if both conditions above do not hold or it* $i = n$, *then exit the repeat loop*

  the algorithm gets a puzzle p_i *from* P *and tries to solve it: (puzzle solving)*

- *repeat the following procedure:*

  1. *sample a suffix* $p_{i+1}, p_{i+2}, \ldots, p_n$
  2. *if* S^n *on input* $\vec{p}, p_i, p_{i+1}, p_{i+2}, \ldots, p_n$ *solves exactly* $r - 1$ *puzzles among the suffix, then output* a_i, *the answer from* S^n *to the puzzle* p_{i+1}

Proof. From a solver S^n, we define S as in the definition. Let i be the index that S stops the prefix finding and begins the puzzle solving. Then we have the following claims:

Claim 46 *In the execution of the algorithm, with probability* $1 - \frac{\delta}{4\eta n M}$ *over the randomness of the algorithm, the estimation of the event* $\mathcal{E}_i(*, \geq r)$ *has precision error at most* $\frac{P(n-i,r,\delta)}{\eta}$ *additively. That is, for all* $\tilde{T}(i, \vec{p} \circ p_{i,j}; \mathcal{E}_i(*, \geq r))$*, we have*

$$\Pr\left[\left|\tilde{T}(i, \vec{p} \circ p_{i,j^*}; \mathcal{E}_i(*, \geq r)) - T(i, \vec{p} \circ p_{i,j^*}; \mathcal{E}_i(*, \geq r))\right| < \frac{P(n-i,r,\delta)}{\eta}\right] > 1 - \frac{\delta}{4\eta n M}.$$

Proof of claim: This is by Chernoff Bound since we take $M_i = O(\eta_i^2 \log(4\eta n M/\delta))$ with $q_i = \frac{\eta}{P(n-i,r,\delta)}$ independent samples. □

We can bound the total number of estimations in the algorithm by Mn since for each $i \in [n]$ the algorithm takes M independent estimations. Thus, by union bound and the claim above, we have with probability $1 - \frac{\delta}{4\eta n}$, all the estimations at loop i have precision $\frac{P(n-i,r,\delta)}{\eta}$. We define this event as *Good_1*, and we have $\Pr[\textit{Good\_1}] > 1 - \frac{\delta}{4\eta}$.

Claim 47 *Let* $i \in [n]$*,* $\vec{p}$ *be the prefix found in the algorithm in the round* i*, and* $\mathcal{E}_i(*, \geq r)$ *be the event that it is going to estimate. Define* $B(i, \vec{p}, \mathcal{E}_i(*, \geq r)) = \{p_i : T(i, \vec{p} \circ p_i; \mathcal{E}_i(*, \geq r)) \geq (1 + 1/\eta) \cdot P(n-i, r, \delta)\}$*.*

Then for a given i*, conditioning on the event Good_1, with probability* $1 - \frac{\delta}{4\eta n}$ *over the randomness of the algorithm, we have the following fact. Suppose for all* $j \in [M]$*, the estimations* $\tilde{T}(i, \vec{p} \circ p_{i,j}; \mathcal{E}_i(*, \geq r)) \leq P(n-i, r, \delta)$*, then we have* $\Pr_{p_i}[p_i \in B(i, \vec{p}, \mathcal{E}_i(*, \geq r))] < \frac{\delta}{\eta}$*.*

Proof of claim: This is equivalent to prove that conditioning on the event *Good_1*, with probability $1 - \frac{\delta}{4\eta n}$ over the randomness of the algorithm, suppose we have $\Pr_{p_i}[p_i \in B(i, \vec{p}, \mathcal{E}_i(*, \geq r))] \geq \frac{\delta}{\eta}$, then we have at least one j such that $\tilde{T}(i, \vec{p} \circ p_{i,j}; \mathcal{E}_i(*, \geq r)) \geq P(n-i, r, \delta)$.

Since we condition on the event *Good_1*, which means all the estimations are within precision errors at most $\frac{P(n-i,r,\delta)}{\eta}$, then given a $p_i \in B(i, \vec{p}, \mathcal{E}_i(*, \geq r))$, the algorithm will correctly identify this. Thus suppose $\Pr_{p_i}[p_i \in B(i, \vec{p}, \mathcal{E}_i(*, \geq r))] \geq \frac{\delta}{\eta}$, then taking $M = O(\frac{\eta}{\delta} \log(4\eta n/\delta))$ samples without hitting any set element is at most $(1 - \frac{\delta}{\eta})^M = O(\frac{\delta}{4\eta n})$. Thus hitting at least one element has probability at least $1 - \frac{\delta}{4\eta n}$, as desired. □

Let the event *Good_2*(i) to be: given i, suppose for all $j \in [M]$, the estimations $\tilde{T}(i, \vec{p} \circ p_{i,j}; \mathcal{E}_i(*, \geq r)) \leq P(n-i, r, \delta)$, then we have $\Pr_{p_i}[p_i \in B(i, \vec{p}, \mathcal{E}_i(*, \geq r))] < \frac{\delta}{\eta}$. Let the event *Good_2* to be: for all i event *Good_2*(i) holds. From the claim above and by union bound, we have $\Pr[\textit{Good\_2} | \textit{Good\_1}] > 1 - \frac{\delta}{4\eta}$. Also by the previous claim we have $\Pr[\textit{Good\_1} \& \textit{Good\_2}] > (1 - \frac{\delta}{4\eta})^2 > 1 - \frac{\delta}{2\eta}$.

Then we are going to calculate the success probability that the algorithm outputs conditioning on the both events *Good_1* and *Good_2*.

Claim 48 *Let $i \in [n]$ be the index when the algorithm stops recurring, and $\vec{p}$ be the prefix it found. Conditioning on both events Good_1 and Good_2 hold, for all sufficiently small constant $c \in (0,1)$ and sufficiently large η the algorithm has success probability greater equal to $(1 - \Theta(1/\eta^{1-c})) \cdot \delta$. In particular, $\eta = \Omega\left(\left(P(n-i-1,r,\delta)/\delta\binom{n-i-1}{r-1}\right)^{1/c}\right)$ is sufficient.*

Proof of claim: Since the algorithm stops at i, found $\vec{p}$ as the prefix, and both *Good_1*, *Good_2* hold, by claim 46 and 47, we can deduce: for at least $1-\frac{\delta}{\eta}$ fraction of p_i, we have (i) $T(i, \vec{p}\circ p_i; \mathcal{E}_i(*, \geq r)) \leq (1+1/\eta)\cdot P(n-i, r, \delta)$, (ii) $T(i, \vec{p}\circ p_i; \mathcal{E}_i(*, \geq r-1)) \leq (1+1/\eta)\cdot P(n-i, r-1, \delta)$. Also we have (iii) $T(i, \vec{p}; \mathcal{E}_i(\geq r)) \geq (1-1/\eta)\cdot P(n-i+1, r, \delta)$ from the recursion criteria in the previous level $i-1$. Similarly to the discussion of the analysis of Direct Product Theorem and Hardness Degradation [CHS05, HR08], we can express the success probability of the algorithm as $\mathrm{E}_{p_i}\left[\mathcal{E}_i(1,*)|\mathcal{E}_i(*,r-1)\right] = \mathrm{E}_{p_i}\left[\frac{\mathcal{E}_i(1,r-1)}{\mathcal{E}_i(*,r-1)}\right]$. Thus we have for every p_i,

$$\begin{aligned}
&\frac{T(i,\vec{p}\circ p_i;\mathcal{E}_i(1,r-1))}{T(i,\vec{p}\circ p_i;\mathcal{E}_i(*,r-1))} \\
&= \frac{T(i,\vec{p}\circ p_i;\mathcal{E}_i(\geq r)) - T(i,\vec{p}\circ p_i;\mathcal{E}_i(*,\geq r))}{T(i,\vec{p}\circ p_i;\mathcal{E}_i(*,\geq r-1)) - T(i,\vec{p}\circ p_i;\mathcal{E}_i(*,\geq r))} \\
&\geq \frac{T(i,\vec{p}\circ p_i;\mathcal{E}_i(\geq r)) - (1+1/\eta)\cdot P(n-i-1,k,\delta)}{(1+1/\eta)\cdot P(n-i-1,r-1,\delta) - (1+1/\eta)\cdot P(n-i-1,r,\delta)}
\end{aligned}$$

Then by taking the expected value over p_i, we get

$$\mathrm{E}_{p_i}\left[\frac{T(i,\vec{p}\circ p_i;\mathcal{E}_i(1,r-1))}{T(i,\vec{p}\circ p_i;\mathcal{E}_i(*,r-1))}\right] \geq \frac{(1-1/\eta)\cdot P(n-i,r,\delta) - (1+1/\eta)\cdot P(n-i-1,r,\delta)}{(1+1/\eta)\cdot P(n-i-1,r-1,\delta) - (1+1/\eta)\cdot P(n-i-1,r,\delta)}$$

$$\geq (1-1/\eta^{1-c})\delta, \text{ for all } \eta \geq \left(\frac{2P(n-i-1,r,\delta)}{\delta\binom{n}{r-1}}\right)^{1/c}.$$

□

■

Thus $succ_{\mathsf{P}}[\mathsf{S}] \geq \Pr[\mathsf{S} \text{ succeeds } | \mathit{Good\_1}\&\mathit{Good\_2}]\Pr[\mathit{Good\_1}\&\mathit{Good\_2}] > (1-1/\eta^{0.99})\delta$ by taking $c = 0.01$. This completes the proof of Lemma 44. Then together with Lemma 30, it is easy to prove Lemma 18 as the previous section for fully-verifiable puzzle systems.

C.2.1 The Asymptotic Result: Proof of Theorem 17

Similar to [CHS05, HR08], we want to establish the asymptotic result from the lemmas with concrete parameters developed in the previous section. Lemma 19 in section C.1 directly gives us the asymptotic result of Theorem 12, since the reduction time depends is polynomial in $n, r, \eta, 1/\delta, 1/P(n,r,\delta)$. However, for weakly verifiable puzzles, we are not able to apply Lemma 18 directly in that the reduction time is polynomial in δ^{-r}, which is super-polynomial when δ is a constant and $r = \mathrm{poly}(s)$ for the security parameter s. However, through a trick that composes two reduction algorithms, we can still achieve the task. For simplicity, we present the form whose parameters are within Chernoff range, and remark that for the other regions, the same argument directly follows.

Lemma 49 *Let $\gamma, \delta \in (0,1)$ be any arbitrary small constant, $n, r \in \mathbb{N} \to \mathbb{N}$ be any efficiently computable and polynomially bounded functions with $(1+\gamma)\delta n(s) \leq r(s) \leq n(s)$, $\mathsf{P} = (G, V)(1^s)$ be any two-phase weakly verifiable puzzle system where s is the security parameter. Suppose P is δ-hard, then $\mathsf{P}_{seq}^{n,r}$ is $(P(n,r,\delta) + \mathsf{ngl})$-hard.*

Proof. (sketch) We prove the contrapositive argument. Suppose $\mathsf{P}_{seq}^{n,r}$ is not $P(n,r,\delta)$-hard, i.e. there exists a solver with success probability greater equal than $P(n,r,\delta) + \epsilon$ for some noticeable function ϵ, then we want to construct a single puzzle solver S with success probability greater equal than $\delta + \epsilon'$ for another noticeable function ϵ', which means that P is not δ-hard.

Suppose $n = O(\log s)$, then Lemma 18 already gives us a reduction algorithm that runs in $\mathrm{poly}(n, r, \delta^{-r}, (P(n,r,\delta)+\epsilon)^{-1}) = \mathrm{poly}(s)$. Thus we are done. For larger n, r, we take the following strategy.

Let $C = \frac{1}{(\gamma^2/4)(1+\gamma/2)\delta}$ be a constant, $n' = C\log(\frac{n}{\epsilon}) = O(\log s)$ since $n/\epsilon = \mathrm{poly}(s)$, and $t = n/n'$, $r' = r/t$. First we observe that from S^n we can construct a solver $\tilde{\mathsf{S}}$ to the system $\mathsf{P}_{seq}^{n',r'}$ with success probability $\frac{P(n,r,\delta)+\epsilon}{t}$. This can be done by a simple average argument where $\tilde{\mathsf{S}}$ first randomly sample $i \leftarrow [t-1]$, then sample a prefix ($i \cdot n'$ puzzles) by simulating a puzzle system $\mathsf{P}_{seq}^{it,tr'}$, then embed the n' puzzles from $\mathsf{P}_{seq}^{n',r'}$, and finally sample the suffix for the solver S^n.

Let δ' be the parameter such that $P(n',r',\delta') = \frac{P(n,r,\delta)+\epsilon}{t}$ holds. Given $\tilde{\mathsf{S}}$, by Lemma 18, we can construct a solver S that solves a single puzzle with success probability greater equal than δ', running time in $\mathrm{poly}(n', r', \delta^{-r'}, 1/\epsilon) = \mathrm{poly}(s)$.

Our goal is to show that $\delta' \geq (1+\gamma/2)\delta \geq \delta + \epsilon'$ for $\epsilon' = \gamma\epsilon/2$ being noticeable. First we observe that $P(n',r',\rho)$ is a increasing function with ρ given fixed n', r'. Then we claim that $P(n',r',\rho) < P(n',r',\delta')$ for $\rho = (1+\gamma/2)\delta$, and thus we will have $\delta' > \rho = (1+\gamma/2)\delta$. Since $r' \geq (1+\gamma)\delta n' = (1+\gamma)/(1+\gamma/2)\rho n' > (1+\gamma/2)\rho n'$, by standard Chernoff bound, we have $P(n',r',\rho) < e^{-(\gamma^2/4)\rho n'} = e^{-(\gamma^2/4)(1+\gamma/2)\delta n'} = \epsilon/n < P(n',r',\delta')$. Thus our goal is fulfilled.

■