Electronic Colloquium on Computational Complexity, Report No. 195 (2024)

On White-Box Learning and Public-Key Encryption

Yanyi Liu[*] Noam Mazor[†] Rafael Pass[‡]

November 28, 2024

Abstract

We consider a generalization of the Learning With Error problem, referred to as the *white-box learning problem*: You are given the code of a sampler that with high probability produces samples of the form $y, f(y) + \epsilon$ where ϵ is small, and f is computable in polynomial-size, and the computational task consist of outputting a polynomial-size circuit C that with probability, say, $1/3$ over a new sample y' according to the same distributions, approximates $f(y')$ (i.e., $|C(y') - f(y')|$ is small). This problem can be thought of as a generalizing of the Learning with Error Problem (LWE) from linear functions f to polynomial-size computable functions.

We demonstrate that worst-case hardness of the white-box learning problem, conditioned on the instances satisfying a notion of computational shallowness (a concept from the study of Kolmogorov complexity) not only *suffices* to get public-key encryption, but is also *necessary*; as such, this yields the first problem whose worst-case hardness characterizes the existence of public-key encryption. Additionally, our results highlights to what extent LWE "overshoots" the task of public-key encryption.

We complement these results by noting that worst-case hardness of the same problem, but restricting the learner to only get black-box access to the sampler, characterizes one-way functions.

[*]Cornell Tech. E-mail: `yl2866@cornell.edu`. Research partly supported by NSF CNS-2149305.

[†]Tel Aviv University. E-mail: `noammaz@gmail.com`. Research partly supported by NSF CNS-2149305, AFOSR Award FA9550-23-1-0312 and AFOSR Award FA9550-23-1-0387 and ISF Award 2338/23.

[‡]Cornell Tech, Technion and Tel Aviv University. E-mail: `rafael@cs.cornell.edu`. Supported in part by NSF Award CNS 2149305, AFOSR Award FA9550-23-1-0387, AFOSR Award FA9550-23-1-0312 and AFOSR Award FA9550-24-1-0267 and ISF Award 2338/23. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government, or the AFOSR.

ISSN 1433-8092

1 Introduction

Public-Key Encryption (PKE) [DH76; RSA78] is one of the most central cryptographic primitives enabling secure communication on the Internet: it is the primitive that enables entities that have not physically met to engage in confidential conversations and collaborations.

In contrast to private-key primitives, such as symmetric-key encryption and digital signatures, that can be securely build from the minimal primitive of one-way functions (and for which many candidate problems are known), we only know of a handful of candidate hard problems from which public-key encryption can be constructed. More specifically, these include (a) *number-theory problems* based on either factoring [RSA78; Rab79] or discrete logarithms [DH76; ElG84], (b) coding-theory based problems [McE78], (c) lattice problems as finding shortest/longest vectors in lattices [AD97; Reg09; BCNHR22], and (d) noisy linear-algebra based problems [Ale03; ABW10]. Out of these, the number-theory based problems can be efficiently solved by quantum algorithms [Sho99], and the coding-theory, lattice and noisy linear algebra problems are all very related—in essence, they can all be viewed as different instances of solving noisy systems of *linear* equations (on either particular natural distributions, or even in the worst-case, when restricting attention to systems of equations satisfying the condition that appropriate solutions exist.[1]

The main purpose of this paper is to provide an assumption that generalizes all these assumptions (i.e., is implied by all of them), yet suffices for obtaining secure PKE. Indeed, the main result of this paper is that hardness of a notion of *white-box learning* achieves this goal.

White-box Learning Perhaps the most common noisy linear algebra-based assumption is the hardness of the *Learning With Error (LWE)* problem [Reg09], which, in essence, stipulates the hardness of recovering a vector $\mathbf{x}$ given $\mathbf{A}, \mathbf{A}\mathbf{x} + \mathbf{e}$, where $\mathbf{A}$ is a matrix, $\mathbf{e}$ is some "small" noise vector and all arithmetic is modulo some prime p. In more detail, we typically require an stronger condition: not only that is hard to recover x, but also that it is hard to compute a value "close" to $\mathbf{a}^T\mathbf{x}$ for a random vector $\mathbf{a}$. In other words, we can think of $\mathbf{x}$ as the description of a function $f_{\mathbf{x}}(\mathbf{a}) = \mathbf{a}^T\mathbf{x}$ that we are trying to *improperly* approximately learn given noisy samples—thus the name "learning with error".

In fact, there is also a different way to think about the LWE problem that will be useful for our purposes (which follows from the construction of Regev's PKE [Reg09]): We are given the code of a sampler P_x that enables providing samples of the form $(\mathbf{a}, f_{\mathbf{x}}(\mathbf{a})+\mathbf{e})$, and the goal is to approximate $f_{\mathbf{x}}(\mathbf{a}')$ on a new fresh sample $\mathbf{a}'$ according to the same distribution as $\mathbf{a}$.[2] We refer to this alternative way of thinking of LWE as an instance of *(improper) white-box learning*, where, more generally, we are given the code of a sampler P that generates noisy samples of the form $(y, f(y)+\epsilon)$ and the goal is to approximate $f(y')$ for a fresh sample y' according to the same distribution, using a polynomial-size circuit. In essence, this problem is generalizing the LWE problem from (improperly) learning *linear functions* from noisy samples, to (improperly) *learning a polynomial-size circuit* from noisy samples, and given white-box access to the sampler[3]; the white-box access feature can be viewed as a generalization of Valiant classic PAC-learning model [Val84] to a setting where the learner not

[1] In more detail, we require worst-case hardness to hold when conditioning on instances that define a lattice where the shortest vector is long compared to amount of noise.

[2] The reason for this is that given $\mathbf{A}, \mathbf{A}\mathbf{x} + \mathbf{e}$, we can generate noisy samples of $f_{\mathbf{x}}(\mathbf{a}')$ by taking linear combinations. See [Reg09] and Section 6.1 for more details.

[3] As we will discuss in more detail later, it is also more general than the LWE problem in the aspect the LWE *sampler* has a particular form, but we here may consider more general classes of samplers.

only gets random samples, but also gets the code of the sampler.

In more detail, given a sampler circuit P that samples "labeled instances" (y, z) (where we think of $z \in \mathbb{N}$ as a, perhaps, noisy label for y), let $\mathsf{Comp}_{\epsilon}^{\Delta}(P)$ denote the set of circuits f that with probability $1-\epsilon$ over $(y, z) \leftarrow P$ satisfy the property that $|z - f(y)| \leq \Delta$ (when interpreting both z and $f(z)$ as elements in $\mathbb{N}$). For a function $\Delta\colon \{1\}^* \to \mathbb{N}$, let Δ-$\mathsf{WBLearn}$ denote the following learning problem:

- **Input:** Circuit $P \in \{0,1\}^n$, with the promise that there exists a circuit $\widehat{C} \in \{0,1\}^n$ such that $\widehat{C} \in \mathsf{Comp}_{n^{-10}}^{\Delta(1^n)/n^{10}}(P)$.
- **Output:** Circuit $C \in \mathsf{Comp}_{1/3}^{\Delta(1^n)}(P)$.[4]

In other words, we are given a sampler that with very high $(1 - n^{-10})$ probability outputs labelled samples where the label is very close $(\Delta(1^n)/n^{10})$ to being correct, and the goal is to, given the code of the sampler P, simply find a circuit C that with probability $2/3$ approximates the label within $\Delta(1^n)$ (i.e., with a factor n^{10} higher error). We use $\mathsf{ExactWBLearn}$ to denote Δ-$\mathsf{WBLearn}$ when $\Delta(1^n) = 0$.

Hardness of Learning v.s. (Public-Key) Cryptography Roughly speaking, the main result of this paper will be to show that under certain restrictions on samplers P (which come from the study of *time-bounded Kolmogorov complexity* [Kol68; Sol64; Cha69; Ko86; Har83; Sip83] and in particular the notion of *computational depth* [AFVMV06]), this generalization of the LWE problem not only suffices to realize a PKE, but will also be necessary. In more detail, worst-case hardness of Δ-$\mathsf{WBLearn}$ under these conditions will *characterize* the existence of public-key encryption. As such, our results yield insight on the extent with which LWE "overshoots" the task of public-key encryption.

We highlight that we are not the first ones to consider connections between learning-theory and cryptography; however, as far as we know, all earlier connections were between the hardness of learning and *private-key cryptography* (i.e., the notion of one-way functions), as opposed to public-key cryptography. Indeed, classic results from [KV94; BFKL93] demonstrate the equivalence of the hardness of a notion of *average-case* PAC-learning of polynomial-size circuits (i.e., black-box learning) and one-way functions.[5] In contrast, our focus here is on PKE, and instead of considering black-box learning, we consider white-box learning. An additional difference is that we consider worst-case hardness, as opposed to average-case hardness, of the learning theory problem. As was recently shown in [HN23], this issue can be overcome (in the context of characterizing one-way functions) through the use of a (alternative) notion of computational depth (more details on the relationship with this work below). To put out result in context, we additionaly show that exactly the same problem that we demonstrate characterizes PKE, also characterizes one-way functions once modified to only provide the learner black-box access to the sampler.

[4]We remark that for efficient algorithms, outputting a circuit that approximates a function f is essentially equivalent to approximating the value of $f(y')$ on a random sample y'. Indeed, the circuit implementation of an algorithm for the latter task, is a valid output for the first task.

[5]And the results of [IL90] can be thought of as characterizing one-way functions through a different type of learning.

1.1 Our Results

Towards explaining our results in more detail, let us first recall the notion of *computational depth* [AFVMV06]. Let the *t-computational depth* of x, denoted $cd^t(x)$, be defined as $cd^t(x) = \mathrm{K}^t(x) - \mathrm{K}(x)$ where $\mathrm{K}(x)$ denotes the Kolmogorov complexity of x and K^t denotes the t-bounded Kolmogorov complexity of x. That is, $cd^t(x)$ measures how much more x could be compressed if the decompression algorithm may be computationally unbounded, as opposed to it running in time bounded by $t(|x|)$.

We will focus on instances P of the Δ-$\mathsf{WBLearn}$ problem having the property that $(P, \widehat{C})$ is "computationally shallow", where $\widehat{C}$ is a circuit that agrees with P with high probability: let Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t}$ denote Δ-$\mathsf{WBLearn}$ with the additional promise that $cd^t(P, \widehat{C}) \leq 2 \log n$.[6]

Characterizing PKE through Hardness of White-Box Learning Our main results is that the hardness of this problem *characterizes* the existence of public-key encryption:

Theorem 1.1. *Let $\epsilon > 0$ be a constant and let $\Delta \colon \{1\}^* \to \mathbb{N}$ be an efficiently computable function such that $\Delta(1^n) \leq 2^{n^{(1-\epsilon)}}$, and $t \colon \mathbb{N} \to \mathbb{N}$ be polynomial such that $t(n) \geq n^{1+\epsilon}$. Then, following are equivalent:*

- *PKE exists.*
- Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$.

Computational depth is typically thought of as measure of "unnaturality" of strings: strings with low computational depth are considered "natural" and those with high computational depth are considered "unnatural".[7] Given this interpretation, our characterization of public key encryption is thus in terms of the worst-case hardness of white-box learning for "natural" instances.

As far as we know, this thus yields the first problem whose worst-case hardness not only suffices for public-key encryption (such as e.g., [Reg09]) but also is *necessary*.

On the Use of Computational Depth We note that Antunes and Fortnow [AF09] elegantly used computational depth to connect worst-case hardness of a problem when restricting attention to elements with small computational depth and *errorless* average-case hardness on sampleable distributions; errorless hardness, however, is not sufficient for cryptographic applications. Nevertheless, inspired by the work of [AF09], worst-case hardness conditioned on instances with small computational depth was used in [LP23] (and independently using a variant of this notion in [HN23]) to characterize one-way functions; additionally, an (interactive) variant of such a notion was also implicitly used in [BLMP23] to characterize key exchange protocols. Our techniques are similar to those employed in [LP23; BLMP23] but instead of applying them to study the hardness of a time-bounded Kolmogorov complexity problem (following [LP20]), we here instead apply them to study a learning theory problem (namely, white-box learning).

[6] We note that there is nothing special about the constant 2; it can be anything that is strictly larger than 1.

[7] The reason why low computational depth captures "natural string" is as follows: random strings are known to have low computational depth; furthermore, known results (c.f. slow growth laws [HN23]) show (at least under derandomization assumptions) that one needs to have a long running time to even find a string with high computational depth. So strings with high computational depth are rare and "hard to find", which is why they can be thought of as "unnatural".

We note that learning theory problems conditioned on small computational depth were recently used in [HN23] to characterize one-way functions, but our techniques here are more similar to [LP23; BLMP23]. In particular, [HN23] does not actually use the standard notion of computational depth but instead define a new alternative variant; in contrast, we here rely on just the standard notion.

Relating Exact and Approximate White-Box Learning Note that in Theorem 1.1, the equivalence hold for any (sufficiently small) choice of Δ, as such we directly get as a corollary the equivalence of Exact and Approximate White-box Learning:

Corollary 1.2. *Let $\epsilon > 0$ be a constant and let $\Delta \colon \{1\}^* \to \mathbb{N}$ be any efficiently computable function such that $\Delta(1^n) \leq 2^{n^{(1-\epsilon)}}$, and let $t \colon \mathbb{N} \to \mathbb{N}$ be any polynomial such that $t(n) \geq n^{1+\epsilon}$. Then the following are equivalent:*

- $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$
- $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$

Bounded-Degree Learning and LWE While in the Δ-$\mathsf{WBLearn}$ problem, we allow the function we are trying to learn to be any polynomial-size circuit, we may also consider a restricted version of the problem, denoted $\Delta\text{-}\mathsf{WBLearn}^d_q$, where we restrict attention to functions that can be computed by a degree d polynomial, and we assume that arithmetic is now over $\mathbb{Z}_q$.

We first remark that our main theorem can next be generalized (basically using padding) to show that it suffices to use learning of degree-n^ϵ polynomials to characterize PKE:

Theorem 1.3. *Let $\epsilon > 0$ be a constant and let $\Delta, q \colon \{1\}^* \to \mathbb{N}$ be efficiently computable functions with $\Delta(1^n) \leq q(1^n)/4$ and $q(1^n) \leq 2^{n^{(1-\epsilon)}}$, and $t \colon \mathbb{N} \to \mathbb{N}$ be polynomial such that $t(n) \geq n^{1+\epsilon}$. The following are equivalent:*

- *PKE exists.*
- $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$*.*
- $\Delta\text{-}\mathsf{WBLearn}^n_q|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$*.*
- $\Delta\text{-}\mathsf{WBLearn}^{n^\epsilon}_q|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$*.*

Additionally, as informally discussed above, we note that the hardness of the LWE problem, implies hardness of the $\Delta\text{-}\mathsf{WBLearn}^1_q$ problem (i.e., white-box learning of *linear* functions.)

Lemma 1.4 (Lemma 6.4, informal)**.** *Assuming the hardness of LWE, there exists a polynomial t such that $\Delta\text{-}\mathsf{WBLearn}^1_q|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$ where $\Delta = q/4$.*

Lemma 1.4 thus shows that white-box learning of linear functions is at least as weak an assumption as LWE. At first sight, one would expect that a converse result may also hold due to known worst-case to average-case reductions for the LWE problem [Reg09; Pei09; BLPRS13] and thus that LWE is equivalent to white-box learning of linear functions. However, the problem with proving the converse direction is that the known worst-case to average-case reductions for LWE only work when the LWE instance defines a lattice where the shortest vector is long compared to amount of noise. Instances sampled from P may not necessarily satisfy this promise, and thus is it not clear how to use an LWE oracle to generally solving white-box learning of linear functions. Thus, it would seem that hardness even of just $\Delta\text{-}\mathsf{WBLearn}_q^1|_{\mathsf{CS}^t}$ is seemingly a weaker (and therefore more general) assumption than LWE. (Of course, it may be that a stronger worst-case to average-case reduction can be established for the LWE problem, in which case equivalence would hold.)

We finally investigate what happens in the regime of "intermediate-degree" polynomials. We remark that using standard linearization techniques, the constant-degree problem is equivalent to the case of degree 1:

Lemma 1.5. *For every constant $d \in \mathbb{N}$, and functions $t, q, \Delta \colon \mathbb{N} \to \mathbb{N}$, there exist $t', q', \Delta' \colon \mathbb{N} \to \mathbb{N}$ such that*

$$\Delta\text{-}\mathsf{WBLearn}_q^d|_{\mathsf{CS}^t} \leq_p \Delta'\text{-}\mathsf{WBLearn}_{q'}^1|_{\mathsf{CS}^{t'}},$$

and

$$\Delta\text{-}\mathsf{WBLearn}_q^d \leq_p \Delta'\text{-}\mathsf{WBLearn}_{q'}^1.$$

Black-box Learning Finally, to put our results in context, we consider the standard PAC learning model [Val84] where the learner only get access to samples: Let $\Delta\text{-}\mathsf{BBLearn}$ denote identically the same problem as $\mathsf{WBLearn}$ with the exception that the learner gets oracle access (i.e., black-box access) to the sampler (as opposed to white-box access to the sampler). This notion is equivalent to the notion of improper Δ-approximate PAC learning for polynomial-size circuits (and when $\Delta = 0$ to simply improper PAC learning).

As before, let $\Delta\text{-}\mathsf{BBLearn}|_{\mathsf{CS}^t}$ denote the problem $\Delta\text{-}\mathsf{BBLearn}$ with the additional promise that $cd^t(P, \widehat{C}) \leq 2\log n$, and let $\mathsf{ExactBBLearn}$ and $\mathsf{ExactBBLearn}|_{\mathsf{CS}^t}$ to denote $\Delta\text{-}\mathsf{BBLearn}$ and $\Delta\text{-}\mathsf{BBLearn}|_{\mathsf{CS}^t}$ when $\Delta(1^n) = 0$.

The following theorem can be viewed as the worst-case analog of the classic result of [KV94; BFKL93] characterizing one-way functions through the hardness of *average-case PAC learning.*

Theorem 1.6. *Let $\epsilon > 0$ be a constant and let $\Delta \colon \{1\}^* \to \mathbb{N}$ be any efficiently computable function such that $\Delta(1^n) \leq 2^{n^{(1-\epsilon)}}$, and let $t \colon \mathbb{N} \to \mathbb{N}$ be any polynomial such that $t(n) \geq n^{1+\epsilon}$. Then the following are equivalent:*

- *One-way function exists*
- $\Delta\text{-}\mathsf{BBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$

As mentioned, [HN23] also recently obtained a worst-case characterization of one-way functions through a learning problem, and using a notion of computational depth. The problems, however, are somewhat different. As opposed to [HN23], we here consider the standard PAC learning problem (whereas they consider a more general learning problem), and condition on the standard notion of low computational depth (whereas they condition on a new notion of low computational depth that they introduce).[8]

We remark that our Theorem 1.6 differs from [KV94; BFKL93] not only in the worst-case condition, but also generalizes those results in the sense that we handle the hardness of Δ-approximate learning for any Δ. As a consequence, we again get an equivalence of approximate and exact black-box learning:

[8]Of course, on a conceptual level, these results are similar; the key point we are trying to make here is that exactly the same learning problem characterizes either one-way functions or PKE, depending on whether the learner gets black-box or white-box access to the sampler.

Corollary 1.7. *Let $\epsilon > 0$ be a constant and let $\Delta\colon \{1\}^* \to \mathbb{N}$ be any efficiently computable function such that $\Delta(1^n) \leq 2^{n^{(1-\epsilon)}}$, and let $t\colon \mathbb{N} \to \mathbb{N}$ be any polynomial such that $t(n) \geq n^{1+\epsilon}$. Then the following are equivalent:*

- $\mathsf{ExactBBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$
- $\Delta\text{-}\mathsf{BBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$

Open Problems We leave as an intriguing open problem the question of whether white-box learning of polynomial, or even logarithmic-degree polynomials, also can be collapsed down to the case of constant-degree functions (and thus to linear functions); if this were possible, it would show that PKE, in essence, inherently requires the structure of the LWE problem.

Additionally, as discussed above, even when just restricting attention to learning linear functions, our learning problem generalizes LWE. It would appear that despite this, cryptographic applications of LWE (e.g., to obtain fully homomorphic encryption [Gen09; BV14]) nevertheless may still be possible from this generalized version; we leave an exploration of this for future work.

Finally, it is an intriguing open problem to relate black-box and white-box learning. By our results, doing so is equivalent to relating public-key and secret-key encryption. We note that relating black-box and white-box learning is interesting even just for the case of linear functions (which by our results is equivalent to $O(1)$-degree polynomials). Indeed, Regev's construction of a PKE [Reg09] can be thought of a reduction from black-box learning to white-box learning of linear functions for a *specific* distribution; it is possible that a similar reduction may be applicable more generally.

1.2 Proof Overview

We here provide a detailed proof overview for the proof of Theorem 1.1. For simplicity, we will show the equivalence between PKE and the worst-case hardness of the exact version, $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$. We start with the construction of PKE based on the hardness of $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$.

Weak PKE First, we use the well-known fact that a PKE is simply a two-rounds key-agreement protocol. Moreover, by the Key-agreement Amplification Theorem of Holenstein [Hol06] and an application of the Goldreich-Levin theorem [GL89], to obtain (full-fledged) PKE, it suffices to obtain a weak form of two-rounds key-agreemnt, which we simply refer to as *Weak PKE* defined as follows: There exist some $\epsilon = 1/\text{poly}$ such that *agreement* between A and B happens with probability $1 - \epsilon$. *Security* requires that Eve cannot guess the key (output by Alice) with probability better than $1 - 20\epsilon$.

The Weak PKE protocol We will next define the weak PKE protocol. We note that this construction resembles the universal key-agreement construction from [HKNRR05], but with some crucial difference that enable our security proof. The parties A and B on input n perform the following steps:

- *Sample random program:* A samples a random length $\lambda \in [2n]$, and a random program Π of length λ.

- *Run random program:* Next, A runs the program Π for at most $t(n)$ steps to get an output, and interprets the output as a pair of circuits $P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$ and $C\colon \{0,1\}^k \to \{0,1\}^\ell$. (Think of P as the sampler for a white-box learning instance, and of C as a potential solution to the problem.) If the output of Π is not a valid encoding of such a pair, A sets $P = P_0$ and $C = C_0$ for two fixed circuits P_0, C_0 that always output 0.

- *Agreement estimation:* A estimates the "agreement probability" of P and C (i.e., checking whether C indeed is a good solution): it samples n^{20} random inputs $w_1, \ldots, w_{n^{20}}$ for P, and computes $(x_i, s_i) = P(w_i)$. It then lets $\widehat{\alpha} = \Pr_{i \leftarrow [n^{20}]}[C(x_i) = s_i]$. If $\widehat{\alpha} \leq 1 - n^{-9}$, A reset the pair $(P, C) = (P_0, C_0)$.

- *First message:* A sends (the "sampler") P to B.

- *Second message:* B applies P on a random input w, and computes $(x, s) = P(w)$. It then sends x to A.

- *Outputs:* A outputs $C(x)$ and Bob outputs s.

Agreement We claim that with probability $1 - 1/n^8$, Alice and Bob will agree (i.e., the final outputs are the same). Note that if $(P, C) = (P_0, C_0)$, Alice and Bob always agree. Moreover, let $\alpha = \Pr_{(x,s) \leftarrow P(U_r)}[C(x) = s]$. Then, the probability of Alice and Bob to agree given that Alice uses (P, C) as the circuits in the protocol, is exactly α. We observe that by the Chernoff bound, the probability that $\widehat{\alpha}$ is far from α is small, and thus Alice uses (P, C) only when α is larger than $1 - n^{-8}$.

Security We claim that Eve that can guess s with probability $1 - n^{-7}$ can be used to solve $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$. In more detail, consider such Eve, and let P be an input for $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$. The idea is to construct an algorithm $\mathsf{Learner}$, that on input P outputs a circuit C, such that $C(x)$ simply simulates Eve on the messages P and x. C then outputs Eve's output. For simplicity, in the following we assume that Eve (and thus $\mathsf{Learner}$) is deterministic. (We note that this is a *non-black-box* reduction: We are using the code of Eve to generate C—in particular, we are including the code of Eve into this circuit.[9])

Let $\mathbf{P}$ be a random variable distributed according to the same distribution of the first message in the above protocol. By assumption, we have that

$$\Pr\bigl[C' = \mathsf{Learner}(\mathbf{P}); (x, s) \leftarrow \mathbf{P}(U_r); C'(x) = s\bigr] \geq 1 - n^{-7}.$$

It follows by a simple averaging argument that

$$\Pr_{\mathbf{P}, C' = \mathsf{Learner}(\mathbf{P})}\Bigl[\Pr_{w}\bigl[(x, s) \leftarrow \mathbf{P}(w); C'(x) = s\bigr] \geq 2/3\Bigr] \geq 1 - 3n^{-7}.$$

Namely, $\mathsf{Learner}$ solves $\mathsf{ExactWBLearn}$ with probability at least $1 - 3n^{-7}$ over the distribution of $\mathbf{P}$. We next use ideas from [LP23; BLMP23] to show this implies that $\mathsf{Learner}$ solves $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$ in the worst case.

[9]This particular non-black usage of Eve is not inherent. We could have considered a different formalization of the learning theory problem which simply requires the attacker to succeed on a randomly sampled instance. Subsequent parts of the argument, however, will use non-black-box access to Eve more inherently.

Indeed, assume that $\mathsf{Learner}$ fails on some instance P of $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$. By the promise of the problem, there exists a circuit $\widehat{C}$ such that $cd^t(P,\widehat{C}) \le 2\log n$, and $\widehat{C}$ agrees with P with probability at least $1-n^{-10}$. Let $\ell = \mathrm{K}^t(P,\widehat{C})$. Our goal is to show that $\mathrm{K}(P,\widehat{C}) < \ell - 2\log n$, which is a contradiction.

Toward this goal, let Let $\mathcal{S}_\ell$ be the set of all pairs of circuits $(P',\widehat{C'})$ with $\mathrm{K}^t(P',\widehat{C'}) = \ell$ that agree with probability at least $1-n^{-10}$, and on which $\mathsf{Learner}$ fails, so that $(P,\widehat{C}) \in \mathcal{S}_\ell$. Fix $(P',\widehat{C'}) \in \mathcal{S}_\ell$, and let Π be the length ℓ program that outputs $(P',\widehat{C'})$ in time t. Observe that the probability of A to sample $(P',\widehat{C'})$ in the first step of the above protocol is at least its probability to sample the program Π, which is $1/2n \cdot 2^{-\ell}$. Since $(P',\widehat{C'})$ agree with high probability, the equality test it the third step of the protocol will pass with high probability, and thus A will send P' to B with probability at least $1/4n \cdot 2^{-\ell}$. In other words,

$$\Pr\big[\mathbf{P} = P'\big] \ge 1/4n \cdot 2^{-\ell}$$

for every $(P',\widehat{C'}) \in \mathcal{S}_\ell$, and thus

$$\Pr[(\mathbf{P},\cdot) \in \mathcal{S}_\ell] \ge |\mathcal{S}_\ell| \cdot 1/4n \cdot 2^{-\ell}$$

(here we say that $(P,\cdot) \in \mathcal{S}_\ell$ if there is some C such that $(P,C) \in \mathcal{S}_\ell$).

On the other hand, by definition of $\mathcal{S}_\ell$, $\mathsf{Learner}$ fails on every $(P',\widehat{C'}) \in \mathcal{S}_\ell$. Since $\mathsf{Learner}$ fails with probability at most $3n^{-7}$ on $\mathbf{P}$, it must holds that

$$3n^{-7} \ge \Pr[(\mathbf{P},\cdot) \in \mathcal{S}_\ell].$$

Combining the above, we get that

$$|\mathcal{S}_\ell| \le 12n^{-6} \cdot 2^{\ell}.$$

We can now use the bound on $\mathcal{S}_\ell$ to bound the Kolmogorov complexity of $(P,\widehat{C})$. To describe $(P,\widehat{C})$, it is enough to describe the set $\mathcal{S}_\ell$, and the index of the pair $(P,\widehat{C})$ in this set. That is,

$$\mathrm{K}(P,\widehat{C}) \le \mathrm{K}(\mathcal{S}_\ell) + \log|\mathcal{S}_\ell| + O(\log\log n) \le \mathrm{K}(\mathcal{S}_\ell) + \ell - 6\log n + O(\log\log n).$$

We conclude the proof with the observation that to describe $\mathcal{S}_\ell$ it is enough to describe n (which can be done using $\log n + O(\log\log n)$ bits), ℓ ($\log n$ bits) and Eve (that can be described with constant many bits[10]). Thus, $\mathrm{K}(\mathcal{S}_\ell) \le 3\log n$, and we get that

$$\mathrm{K}(P,\widehat{C}) < \ell - 2\log n,$$

as we wanted to show.

Hardness of $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$ from PKE. We next show that the existence of PKE implies the hardness of $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$. We now sketch the proof. Let Gen be the algorithm the generate pair (pk,sk) of public and secret key, and let $\mathsf{Enc},\mathsf{Dec}$ be the encryption and decryption protocols. For a random pair of keys $(pk,sk) \leftarrow \mathsf{Gen}(r)$, we construct two circuits, $P(w,s) = (x = \mathsf{Enc}_{pk}(s;w), s)$, and $\widehat{C}(x) = \mathsf{Dec}_{sk}(x)$.

[10]This non-black-box usage of Eve (which is taken from [LP23; BLMP23]) is seemingly inherent to our proof technique. Note that we are here relying on the fact that Eve is a uniform algorithm, but as we discuss in the formal section, the argument can be extended to work also in the non-uniform setting.

By the security of the PKE scheme, it follows that with high probability over the randomness of Gen, it is hard to learn the function of C, as the circuit P only uses the public key, and the function C computes is the decryption of a random encryption. This already implies that $\mathsf{ExactWBLearn}$ is hard, but we still need to show that P is inside the promise of the problem $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$.

It follows by the correctness of the PKE scheme that $\widehat{C}$ computes the function that is sampled by P. Thus, to be in the promise of $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$, we only need that $(P, \widehat{C})$ has small computational depth. This, however, is not necessarily the case. To solve this, we simply pad $\widehat{C}$ with the randomness used by Gen. That is, we let $\widehat{C'}$ be a functionally equivalent circuit to $\widehat{C}$, with r encoded to it, where r is such that $\mathsf{Gen}(r) = (pk, sk)$. It follows that when t is large enough, $\mathrm{K}^t(P, \widehat{C'}) \leq |r| + O(1)$ (as we can describe them by simply describing the randomness and the algorithm Gen). On the other hand, $\mathrm{K}(P, \widehat{C'}) \geq \mathrm{K}(r)$ (since r can be obtained from $\widehat{C'}$), which is at least $|r| - O(1)$ with high probability. Together, we conclude that with high probability over the randomness of Gen, the circuits $(P, \widehat{C'})$ are in the promise of $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$.

Comparison with [BLMP23] We remark that at a high-level, our construction of a two-message key-agreement protocol shares many similaries with the key-agreement protocol developing in [BLMP23], relying on the hardness of an interactive notion of time-bounded Kolmogorov complexity, conditionned on an analog of computational shallowness. They central difference is that the protocol in [BLMP23] requires at least 3 rounds due to the use of an equility check protocol to determine whether Alice and Bob managed to agree on a key. In contrast, our protocol does not rely on such an equality check step (and thus it can be executed in only two rounds, which is crucial to get PKE); indeed, we replaced the equality protocol with a step where Alice on her own can determine whether the message she sends will enable agreement to happen. Of course, the reason why we can do this is that we are reducing security from a different problem.

Other features of the protocol are quite similar; this is because we also rely on the worst-case hardness of a problem "conditioning on computationally shallow instances". Indeed, as described above, our security proof shares many features with those of [LP20; BLMP23].

Comparison with Universal Constructions Universal constructions of PKE are known (see [HKNRR05]); that is, constructions having the property that they are secure if (any) PKE exist. We emphasize that while the details of the protocol from [HKNRR05] are somewhat different, the "agreement estimation" step performed there is very similar to what we do. Furthermore, we can interpret our protocol as an alternative (variant) universal PKE protocol in which Alice chooses a random (key-generation) program Gen and executes it to get an encryption scheme Enc_{pk} and description scheme Dec_{sk}, with the keys pk, sk hardcoded to the scheme.[11] Alice then estimates the agreement probability of the encryption, and if it is high enough, she sends the encryption scheme as the public-key, and uses the decryption scheme as the private-key. If PKE exists, with a noticeable probability over the choice of Gen, this scheme will be secure. We emphasize that since we base the security of our protocol (and thus also the above universal one) on the hardness of the white-box learning problem, it enables an approach for measuring the concrete security of the protocol by relating it to the security of the learning theory problem.

[11]In the protocol of [HKNRR05], Alice would instead choose random programs for Alice and Bob to run; we do not know how to prove the security of such a protocol under our assumption.

2 Preliminaries

2.1 Notations

All logarithms are taken in base 2. We use calligraphic letters to denote sets and distributions, bold uppercase for random variables, and lowercase for values and functions. Let poly stand for the set of all polynomials. Let PPT stand for probabilistic poly-time, and n.u.-poly-time stand for non-uniform poly-time. An n.u.-poly-time algorithm A is equipped with a (fixed) poly-size advice string set $\{z_n\}_{n\in\mathbb{N}}$ (that we typically omit from the notation), and we let A_n stand for A equipped with the advice z_n (used for inputs of length n). For a randomized algorithm A, we denote by $\mathsf{A}(\cdot;r)$ the algorithm A with fixed randomness $r \in \{0,1\}^*$. Let neg stand for a negligible function. Given a vector $v \in \Sigma^n$, let v_i denote its i^{th} entry, let $v_{<i} = (v_1,\ldots,v_{i-1})$ and $v_{\leq i} = (v_1,\ldots,v_i)$. Similarly, for a set $\mathcal{I} \subseteq [n]$, let $v_{\mathcal{I}}$ be the ordered sequence $(v_i)_{i\in\mathcal{I}}$. For $x,y \in \{0,1\}^*$, we use $x||y$ to denote the concatenation of x and y. For a set $\mathcal{S} \subseteq \{0,1\}^*$, we use $\mathcal{S}||y$ to denote the set $\{x||y \colon x \in \mathcal{S}\}$.

2.2 Distributions and Random Variables

When unambiguous, we will naturally view a random variable as its marginal distribution. The support of a finite distribution $\mathcal{P}$ is defined by $\mathrm{Supp}(\mathcal{P}) := \{x \colon \Pr_{\mathcal{P}}[x] > 0\}$. For a (discrete) distribution $\mathcal{P}$, let $x \leftarrow \mathcal{P}$ denote that x was sampled according to $\mathcal{P}$. Similarly, for a set $\mathcal{S}$, let $x \leftarrow \mathcal{S}$ denote that x is drawn uniformly from $\mathcal{S}$. For $m \in \mathbb{N}$, we use $\mathbf{U}_m$ to denote a uniform random variable over $\{0,1\}^m$ (that is independent from other random variables in consideration). The statistical distance (also known as, variation distance) of two distributions $\mathcal{P}$ and $\mathcal{Q}$ over a discrete domain $\mathcal{X}$ is defined by $\mathrm{SD}(\mathcal{P},\mathcal{Q}) := \max_{\mathcal{S}\subseteq\mathcal{X}}|\mathcal{P}(\mathcal{S}) - \mathcal{Q}(\mathcal{S})| = \frac{1}{2}\sum_{x\in\mathcal{S}}|\mathcal{P}(x) - \mathcal{Q}(x)|$.

The following lemma is proven in Appendix A.

Lemma 2.1. *There exists an efficient oracle-aided algorithm* A *such that the following holds. Let* $(\mathbf{X},\mathbf{Y})$ *be a pair of jointly distributed random variables over* $\{0,1\}^* \times \{0,1\}^n$*, and let* $\mathbf{R}$ *be a uniform independent random variable over* $\{0,1\}^n$*. Let* E *be an algorithm such that* $\Pr[\mathsf{E}(\mathbf{X},\mathbf{R}) = \langle \mathbf{Y},\mathbf{R}\rangle] \geq 1-\epsilon$*, for* $0 \leq \epsilon$*. Then* $\Pr\left[\mathsf{A}^{\mathsf{E}}(1^n,\mathbf{X}) = \mathbf{Y}\right] \geq 1 - 8\epsilon - \mathrm{neg}(n)$*.*

2.3 Circuits

In this paper we consider circuits over the De-Morgan Basis, which contains the following gates: $\wedge$ ("and" gate with fan-in 2), $\vee$ ("or" gate with fan-in 2), and $\neg$ ("not" gate with fan-in one). The size of a circuit C, is the number of gates in C.

We consider encoding of a circuit as a string over $\{0,1\}^*$ in the following natural way: Given a circuit C, we first encode the length of C using a prefix-free encoding, and then for every gate g, according to a topological order, we encode its type (input, output, $\wedge$,$\vee$, or $\neg$), and the (up to two) gates wired into g.

Observe that every circuit C of size s can be encoded to a string of length $O(s\log s)$. Moreover, given the encoding and an input x to C, $C(x)$ can be computed efficiently (in polynomial time in the encoding length).

We identify a string in $\{0,1\}^*$ with the circuit it encodes. For example, for $C \in \{0,1\}^n$ we use $C(x)$ to denote the value of the circuit encoded by C applied on the input x. We denote by $|C|$ the length of the encoding of C.

Finally, by encoding first the size of C, we assume that for every encoding of a circuit C, and for every $z \in \{0,1\}^*$, the string $C||z$ encodes the same circuit as C.[12]

2.4 Entropy

For a random variable $\mathbf{X}$, let $\mathrm{H}(\mathbf{X}) = \mathrm{E}[\log \frac{1}{\Pr[\mathbf{X}=x]}]$ denote the (Shannon) entropy of $\mathbf{X}$, and let $\mathrm{H}_\infty(\mathbf{X}) = \min_{x \in Supp(\mathbf{X})} \log \frac{1}{\Pr[\mathbf{X}=x]}$ denote the *min-entropy* of $\mathbf{X}$.

$$\min_{x \in \mathrm{Supp}(\mathbf{X})} \log \frac{1}{\Pr[\mathbf{X}=x]}.$$

For a random variable $\mathbf{X}$ and an event E, we use $\mathrm{H}_\infty(\mathbf{X} \mid E)$ to denote the min-entropy of the distribution $\mathbf{X}|_E$. We will use the following facts.

Fact 2.2. *Let $\mathbf{X}$ and $\mathbf{Y}$ be independent random variables. Then $\mathrm{H}_\infty(\mathbf{X}, \mathbf{Y}) = \mathrm{H}_\infty(\mathbf{X}) + \mathrm{H}_\infty(\mathbf{Y})$.*

2.5 Complexity Classes

We define the complexity classes FBPP and ioP/poly.

Definition 2.3 (Infinitely-often FBPP (ioFBPP))**.** *A binary relation $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ is in* ioFBPP *if there exists* PPT *algorithms* A *such that the following holds for infinitely many n's:*

For every $x \in \{0,1\}^n$ such that there exists $y \in \{0,1\}^*$ with $(x,y) \in \mathcal{R}$,

$$\Pr\Big[(x, \mathsf{A}(x, 1^k)) \in \mathcal{R}\Big] \geq 1 - 1/k,$$

for every $k > 0$. $\mathcal{R}$ is in ioP/poly *if the above holds with respect to $n.u. - poly - time$ algorithms* A.

2.6 One-Way Function

Definition 2.4 (One-way function)**.** *A polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ is called a* one-way function *if for every polynomial-time algorithm* A*,*

$$\Pr_{x \leftarrow \{0,1\}^n}\big[\mathsf{A}(1^n, f(x)) \in f^{-1}(f(x))\big] = \mathrm{neg}(n)$$

Definition 2.5 (Weak one-way function)**.** *Let $m \in \mathrm{poly}$ be a polynomial-time computable function. A polynomial-time computable function $f : \{0,1\}^{m(n)} \to \{0,1\}^*$ is called α-*weak one-way function *if for every polynomial-time algorithm* A*, for every large enough n,*

$$\Pr_{x \leftarrow \{0,1\}^{m(n)}}\big[\mathsf{A}(1^n, f(x)) \in f^{-1}(f(x))\big] \leq 1 - \alpha(n)$$

f is a weak one-way function *if it is $1/p$-weak one way function, for some $p \in \mathrm{poly}$.*

Theorem 2.6 (Weak to strong OWFs, [Yao82])**.** *One-way functions exist if and only if weak-one way functions exist.*

[12]We actually only use the fact that we can encode z into the circuit. This can be done by adding dummy gates that do not change the output of C, where each gate g_i is either $\wedge$ or $\vee$ according to the ith bit of z.

2.7 Public-Key Encryption

Definition 2.7 (Public-key encryption scheme (PKE))**.** *A triplet of randomized, efficiently computable functions* (Gen, Enc, Dec) *is a* $(\alpha(n), \beta(n))$-public-key encryption scheme (PKE) *if the following holds:*

- *Correctness: For every large enough $n \in \mathbb{N}$ and any $m \in \{0,1\}$,*

$$\Pr_{(sk,pk)\leftarrow \mathsf{Gen}(1^n)}[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m] \geq 1 - \alpha(n)$$

- *Security: For every* PPT Eve*, for every large enough $n \in \mathbb{N}$,*

$$\Pr_{(sk,pk)\leftarrow \mathsf{Gen}(1^n), m\leftarrow\{0,1\}}[\mathsf{Eve}(pk, \mathsf{Enc}(pk, m)) = m] \leq 1/2 + \beta(n).$$

Such a scheme is a PKE *if it is $(1/n^c, 1/n^c)$-PKE for every $c \in \mathbb{N}$.*

The following lemma shows that it is possible to amplify an 1-bit weak key-agreement protocol into a key-agreement. This lemma is a simple case of the more general result of Holenstein [Hol06].

Lemma 2.8 (PKE amplification, [Hol06])**.** *The following holds for every constants $c_1 > c_2$. Assume there exists an $(n^{-c_1}, 1/2 - n^{-c_2})$-PKE. Then, there exists a PKE.*

We also define weak-PKE.

Definition 2.9 (Weak Public-key encryption scheme (weak-PKE))**.** *For an efficiently computable function $d \colon \{1\}^* \to \mathbb{N}$, a triplet of randomized, efficiently computable functions* (Gen, Enc, Dec) *is a* $(\alpha(n), \beta(n), \gamma(n))$-weak-public-key encryption scheme (weak-PKE) *if the following holds:*

- *For every $n \in \mathbb{N}$, given pk and randomness r, $\mathsf{Enc}(pk; r)$ outputs a message $m(pk; r)$ and an output $o(pk; r) \in \mathbb{N}$.*

- *Correctness: For every large enough $n \in \mathbb{N}$*

$$\Pr_{(sk,pk)\leftarrow \mathsf{Gen}(1^n), r}[|\mathsf{Dec}(sk, m(pk, r)) - o(pk, r)| \leq d(1^n)] \geq 1 - \alpha(n)$$

- *Security: For every* PPT Eve*, for every large enough $n \in \mathbb{N}$,*

$$\Pr_{(sk,pk)\leftarrow \mathsf{Gen}(1^n), r}[|\mathsf{Eve}(pk, m(pk, r)) - o(pk, r)| \leq \gamma(n) \cdot d(1^n)] \leq \beta(n).$$

We prove the following lemma, stating that weak-PKE can be used to construct PKE.

Lemma 2.10 (Weak-PKE amplification)**.** *The following holds for every constants $c_1 > c_2$. Assume there exists an $(n^{-c_1}/2, 1 - 10n^{-c_2}, 2n^{c_1})$-weak-PKE. Then, there exists a PKE.*

Proof. Let (Gen, Enc, Dec) be a $(n^{-c_1}/2, 1 - 10n^{-c_2}, 2n^{c_1})$-weak-PKE with parameter d. Consider a scheme $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ where Gen' is identical to Gen, and Enc', Dec' defined as follows:

- Given a pk, a bit b to encrypt and randomness r, Enc' first executes Enc to get $m(pk,r)$ and $o(pk,r)$. Then Enc' chooses a random shift $\sigma \leftarrow [0, 2n^{c_1}d(1^n)]$ and a random vector $v \leftarrow \{0,1\}^{\ell}$, where ℓ is (an upper bound on) the length of the bit representation of $o(pk,r)$. It then computes $\widehat{o} = \left\lceil \frac{o(pk,r)+\sigma}{2n^{c_1}d(1^n)} \right\rceil$, and outputs $(m(pk,r), \sigma, v, \langle v, \widehat{o}\rangle \oplus b)$.

- Let Dec' be the algorithm that given sk and (m,σ,v,c), first executes $\mathsf{Dec}(sk,m)$ to get a number o', and then outputs $\langle \left\lceil \frac{o'+\sigma}{2n^{c_1}d(1^n)} \right\rceil \rangle \oplus c)$.

We next show that $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ is an $(1-n^{-c_1}, 1/2 - n^{-c_2})$-PKE, which implies the lemma by Lemma 2.8.

For correctness, it is enough to show that $\left\lceil \frac{o'+\sigma}{2n^{c_1}d(1^n)} \right\rceil$ is equal to $\left\lceil \frac{o(pk,r)+\sigma}{2n^{c_1}d(1^n)} \right\rceil$ with high probability. Let $\mathbf{O}, \mathbf{O}'$ be the values of the $o(pk,r)$ and o' in a random execution of the above protocol. Recall that by the correctness of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, it holds that $\Pr[|\mathbf{O} - \mathbf{O}'| \leq d(1^n)] \geq 1 - n^{-c_1}/2$. Moreover, for any two numbers O, O' such that $|O - O'| \leq d(1^n)$, it holds that

$$\Pr_{\sigma}\left[\left\lceil \frac{O+\sigma}{2n^{c_1}d(1^n)} \right\rceil \neq \left\lceil \frac{O'+\sigma}{2n^{c_1}d(1^n)} \right\rceil\right] \leq n^{-c_1}/2.$$

Indeed, assume without loss of generality that $O \leq O'$,

$$\begin{aligned}
\Pr_{\sigma}\left[\left\lceil \frac{O+\sigma}{2n^{c_1}d(1^n)} \right\rceil \neq \left\lceil \frac{O'+\sigma}{2n^{c_1}d(1^n)} \right\rceil\right] &\leq \Pr_{\sigma}\left[\left\lceil \frac{O+\sigma}{2n^{c_1}d(1^n)} \right\rceil \neq \left\lceil \frac{O+d(1^n)+\sigma}{2n^{c_1}d(1^n)} \right\rceil\right] \\
&= \Pr_{\sigma}\left[\left\lceil \frac{\sigma}{2n^{c_1}d(1^n)} \right\rceil \neq \left\lceil \frac{d(1^n)+\sigma}{2n^{c_1}d(1^n)} \right\rceil\right] \\
&\leq d(1^n)/2n^{c_1}d(1^n) = n^{-c_1}/2.
\end{aligned}$$

Thus we conclude that $\Pr[\mathbf{O} = \mathbf{O}'] \geq 1 - n^{c_1}$.

For security, assume there exists an PPT algorithm Eve that given $(m(pk,r), \sigma, v, \langle v, \left\lceil \frac{o(pk,r)+\sigma}{2n^{c_1}d(1^n)} \right\rceil \rangle \oplus b)$ can guess the value of b with probability $1 - n^{-c_2}$. Then, there exists an algorithm Eve' that can guess the value of $\langle v, \left\lceil \frac{o(pk,r)+\sigma}{2n^{c_1}d(1^n)} \right\rceil \rangle$ with the same probability. By Lemma 2.1, such an algorithm Eve' can be used to output $w = \left\lceil \frac{o(pk,r)+\sigma}{2n^{c_1}d(1^n)} \right\rceil$ with probability $1 - 10n^{-c_2}$ (given $m(pk,r), \sigma$). Since $w \cdot (2n^{c_1}d) - \sigma$ is an approximation of $o(pk,r)$ within distance at most $2n^{c_1}d(1^n)$, we get that there exists an efficient algorithm, that breaks the assumed security of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. $\square$

2.8 Kolmogorov Complexity and Computational Depth

Roughly speaking, the *t-time-bounded Kolmogorov complexity*, $\mathrm{K}^t(x)$, of a string $x \in \{0,1\}^*$ is the length of the shortest program $\Pi = (M, y)$ such that, when simulated by an universal Turing machine, Π outputs x in $t(|x|)$ steps. Here, a program Π is simply a pair of a Turing Machine M and an input y, where the output of P is defined as the output of $M(y)$. When there is no running time bound (i.e., the program can run in an arbitrary number of steps), we obtain the notion of Kolmogorov complexity.

In the following, fix universal TM U with polynomial simulation overhead, and let $\mathsf{U}(\Pi, 1^t)$ denote the output of Π when emulated on U for t steps. We now define the notion of Kolmogorov complexity with respect to the universal TM U.

Definition 2.11. *Let t be a polynomial. For all $x \in \{0,1\}^*$, define the* t-bounded Kolmogorov complexity *of x*

$$\mathrm{K}^t(x) = \min_{\Pi \in \{0,1\}^*} \{|\Pi| : \mathsf{U}(\Pi, 1^{t(|x|)}) = x\}$$

where $|\Pi|$ is referred to as the description length *of Π. When there is no time bound, the* Kolmogorov complexity *of x is defined as*

$$\mathrm{K}(x) = \min_{\Pi \in \{0,1\}^*} \{|\Pi| : \exists t \in \mathbb{N} \mathsf{U}(\Pi, 1^t) = x\}.$$

The computational depth *of x [AFVMV06], denoted by $cd^{t,\infty}(x)$, is defined to be*

$$cd^{t,\infty}(x) = \mathrm{K}^t(x) - \mathrm{K}(x).$$

We use $\mathsf{K}(x,y)$ to denote the Kolmogorov complexity of some generic self-delimiting encoding of the pair x, y. Recall that we use $\mathsf{K}(x||y)$ to denote the complexity of the concatenation of x and y. We will use the following well-known fact:

Fact 2.12. *For every $x, y \in \{0,1\}^*$,*

$$\mathsf{K}(x,y) \le \mathsf{K}(x) + \mathsf{K}(y) + \log(\mathsf{K}(x)) + 2\log\log(\mathsf{K}(x)) + O(1).$$

We will also use the following bound on the Kolmogorov complexity of strings sampled from distributions with high min-entropy.

Lemma 2.13. *For every $n \in \mathbb{N}$, and every distribution $\mathcal{D}$, it holds that*

$$\Pr_{x \leftarrow \mathcal{D}}[\mathsf{K}(x) \ge \mathrm{H}_\infty(\mathcal{D}) - \log n] \ge 1 - 1/n.$$

Proof. There are at most $2^{\mathrm{H}_\infty(\mathcal{D}) - \log n}$ strings x with $\mathsf{K}(x) < \mathrm{H}_\infty(\mathcal{D}) - \log n$. Thus,

$$\Pr_{x \leftarrow \mathcal{D}}[\mathsf{K}(x) < \mathrm{H}_\infty(\mathcal{D}) - \log n] \le 2^{\mathrm{H}_\infty(\mathcal{D}) - \log n} \cdot 2^{-\mathrm{H}_\infty(\mathcal{D})} = 1/n.$$

□

We will also use the well-known Chernoff bound in our proof.

Fact 2.14 (Hoeffding's inequality). *Let $\mathbf{A}_1, ..., \mathbf{A}_n$ be independent random variables s.t. $\mathbf{A}_i \in \{0,1\}$. Let $\widehat{\mathbf{A}} = 1/n \cdot \Sigma_{i=1}^n \mathbf{A}_i$ and $\mu = \mathrm{E}\left[\widehat{\mathbf{A}}\right]$. For every $\epsilon \in [0,1]$ it holds that:*

$$\Pr\left[\left|\widehat{\mathbf{A}} - \mu\right| \ge \epsilon\right] \le 2 \cdot e^{-\epsilon^2 \cdot n}.$$

3 White-Box Distributional Learning

Let $P \colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$ be a circuit that, given r bits of randomness, samples labeled instances (x, s). In the following we view s as a binary representation of a number in $\mathbb{N}$ (and respectively all the operations below are over $\mathbb{N}$, and we use $|\cdot|$ to denote the absolute value). We define the set of all circuits that approximate P with high probability,

$$\mathsf{Comp}_\epsilon^\Delta(P) = \left\{C \colon \{0,1\}^k \to \{0,1\}^\ell \colon \Pr_{(x,s) \leftarrow P(\mathbf{U}_r)}[|C(x) - s| \le \Delta] \ge 1 - \epsilon\right\}.$$

We define the following white-box learning problem $\mathsf{WBLearn}$:

Definition 3.1 (Δ-$\mathsf{WBLearn}$)**.** *For a function $\Delta\colon \{1\}^* \to \mathbb{N}$, let Δ-$\mathsf{WBLearn}$ be the following learning problem:*

- *Input: Circuit $P \in \{0,1\}^n$, with the promise that there exists a circuit $\widehat{C} \in \{0,1\}^n$ such that $\widehat{C} \in \mathsf{Comp}_{n^{-10}}^{\Delta(1^n)/n^{10}}(P)$.*
- *Output: Circuit $C \in \mathsf{Comp}_{1/3}^{\Delta(1^n)}(P)$*

In this work we are focusing on inputs P for which the circuit $\widehat{C}$ that agree with P with high probability, is such that the description of $\widehat{C}$ and P is computational shallow. Formally, for a time function $t\colon \mathbb{N} \to \mathbb{N}$, we denote by Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t}$ the problem Δ-$\mathsf{WBLearn}$ with the additional promise that $cd^t(P,\widehat{C}) \le 2\log n$.

We use $\mathsf{ExactWBLearn}$ and $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t}$ to denote Δ-$\mathsf{WBLearn}$ and Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t}$ when $\Delta(1^n) = 0$. Note that $\mathsf{ExactWBLearn}$ and Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t}$ are incomparable: While in Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t}$ we only need to find a circuit that approximates P, the promise in $\mathsf{ExactWBLearn}$ is stronger. Yet, there is a simple reduction from $\mathsf{ExactWBLearn}$ to Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t}$.

Lemma 3.2. *For every $\Delta\colon \{1\}^* \to \mathbb{N}$ such that $\Delta \in 2^{o(n/\log n)}$, it holds that*

$$\mathsf{ExactWBLearn} \le_p \Delta\text{-}\mathsf{WBLearn}.$$

Similarly, for any such Δ and a function $t\colon \mathbb{N} \to \mathbb{N}$, there exists $t'\colon \mathbb{N} \to \mathbb{N}$, such that

$$\mathsf{ExactWBLearn}|_{\mathsf{CS}^t} \le_p \Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^{t'}}.$$

Proof. We start with the first reduction $\mathsf{ExactWBLearn} \le_p \Delta$-$\mathsf{WBLearn}$. Given a circuit $P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$ of length n, the reduction outputs a circuit $P'\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^{\ell+n}$, which is equivalent to P, with additional n output gates that always output 0. That is, $P'(w) = (x, s')$, where $P(w) = (x,s)$ and $s' = 2^n \cdot s$. As we only added n gates to P, P' can be encoded using $O(n\log n)$ bits. Using padding we assume that $|P'| \in \Theta(n\log n)$.

For correctness, observe that if there exists $\widehat{C} \in \mathsf{Comp}_{n^{-10}}^{0}(P)$, then there exists such $\widehat{C'} \in \mathsf{Comp}_{n^{-10}}^{0}(P')$ (where $\widehat{C'}$ is defined similarly to P' using $\widehat{C}$). Moreover, an approximation of the output s' of P' within a distance of $2^n/4$ is equivalent to the exact output s of P. Indeed, given an $2^n/4$-approximation of s' we can find s by simply dividing s' by 2^n and rounding to the closest integer.

Finally, to see the second reduction, it is enough to show that $cd^{t'}(P',\widehat{C'}) \le 2\log|P'|$. Since $P', \widehat{C'}$ can be efficiently constructed given $P, \widehat{C}$, and similarly $P, \widehat{C}$ can be efficiently constructed given $P', \widehat{C'}$, it holds that for large enough (polynomial time t', $cd^{t'}(P',\widehat{C'}) \le cd^t(P,\widehat{C}) + O(1) \le 2\log n + O(1) \le 2\log n + 2\log\log n \le 2\log|P'|$. $\square$

We prove the following theorem.

Theorem 3.3. *Let $\epsilon > 0$ be a constant and let $\Delta\colon \{1\}^* \to \mathbb{N}$ be any efficiently computable function such that $\Delta(1^n) \le 2^{n^{(1-\epsilon)}}$, and let $t\colon \mathbb{N} \to \mathbb{N}$ be any polynomial such that $t(n) \ge n^{1+\epsilon}$. Then the following are equivalent:*

- *PKE exists*

- $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$

As a corollary, we get a result of independent interest relating exact and approximate white-box learning:

Corollary 3.4. *Let $\epsilon > 0$ be a constant and let $\Delta\colon \{1\}^* \to \mathbb{N}$ be any efficiently computable function such that $\Delta(1^n) \leq 2^{n^{(1-\epsilon)}}$, and let $t\colon \mathbb{N} \to \mathbb{N}$ be any polynomial such that $t(n) \geq n^{1+\epsilon}$. Then the following are equivalent:*

- $\mathsf{ExactWBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$
- $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$

Theorem 3.3 follows by Theorems 4.1 and 5.1 which are stated and proven in Sections 4 and 5.

4 Worst-Case hardness of $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \implies$ PKE

In this section we prove the following theorem, that states that the worst-case hardness of $\mathsf{WBLearn}|_{\mathcal{Q}_t}$ implies the existence of public-key encryption scheme.

Theorem 4.1. *Let $t(n)$ be a polynomial and $\Delta\colon \{1\}^* \to \mathbb{N}$ an efficiently computable function. Then if $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$, PKE exist.*

In the following, let $C_0\colon \{0,1\} \to \{0,1\}$ be the circuit that always outputs 0, and $P_0\colon \{0,1\} \to \{0,1\} \times \{0,1\}$ be the circuit that always outputs $(0,0)$.

To prove the above theorem, fix $t = t(n)$ and $\Delta = \Delta(1^n)$, and consider the following scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$:

Algorithm 4.2 (Gen)**.**

Parameter: function $t\colon \mathbb{N} \to \mathbb{N}$.

Input: 1^n.

Operation:

1. *Sample $\lambda \leftarrow [3n]$ and $\Pi \leftarrow \{0,1\}^{\lambda}$.*
2. *Run Π for $t(2n)$ steps to get circuits $C\colon \{0,1\}^k \to \{0,1\}^{\ell}, P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^{\ell}$ for some k, r, ℓ. If the output of Π is not two such circuits, set $C = C_0$, $P = P_0$, $r = k = \ell = 1$.*
3. *Randomly sample $(x_1, s_1), \ldots, (x_{n^{20}}, s_{n^{20}}) \leftarrow P(U_r)$, and compute $\widehat{\alpha} = \Pr_{i \leftarrow [n^{20}]}\left[|C(x_i) - s_i| \leq \Delta(1^n)/n^{10}\right]$. If $\widehat{\alpha} < 1 - 2n^{-8}$, reset $C = C_0$ and $P = P_0$.*
4. *Output (k, ℓ, C) as the secret key, and (r, k, ℓ, P) as the public key.*

Algorithm 4.3 (Enc)**.**

Input: public-key $(r \in \mathbb{N}, k \in \mathbb{N}, \ell \in \mathbb{N}, P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^{\ell})$.

Operation:

1. *Sample randomness* $z \leftarrow \{0,1\}^r$.

2. *Compute* $P(z)$ *to get* (x, s).

3. *Output* $m = x$ *and* $o = s$.

Algorithm 4.4 (Dec).
Input: secret-key $(k \in \mathbb{N}, \ell \in \mathbb{N}, C\colon \{0,1\}^k \to \{0,1\})$, *cipher* $x \in \{0,1\}^k$.
Operation:

1. *Compute* $C(x) = s'$.

2. *Output* s'.

Observe that the size of the circuits C and P sampled by $\mathsf{Gen}(1^n)$ is at most $t(2n)$. Thus, all of the above algorithms can be implemented efficiently. Below we bound the correctness and the security of the above scheme. For every $n \in \mathbb{N}$, let $(\mathbf{K}_n, \mathbf{L}_n, \mathbf{C}_n)$ and $(\mathbf{R}_n, \mathbf{K}_n, \mathbf{L}_n, \mathbf{P}_n)$ be the random variables distributed according to the secret and public keys $(k, \ell, C), (r, k, \ell, P)$ in a random execution of $\mathsf{Gen}(1^n)$. Let $\mathbf{M}_n$ and $\mathbf{O}_n$ be the output of $\mathsf{Enc}(\mathbf{R}_n, \mathbf{K}_n, \mathbf{L}_n, \mathbf{P}_n)$ in a random execution.

4.1 Correctness

We start by analyzing the correctness probability of the scheme.

Lemma 4.5. *For every* $n \in \mathbb{N}$, *it holds that*

$$\Pr\left[|\mathsf{Dec}((\mathbf{K}_n, \mathbf{L}_n, \mathbf{C}_n), \mathbf{M_n}) - \mathbf{O}_n| \leq \Delta(1^n)/n^{10}\right] \geq 1 - n^{-7}.$$

To prove Lemma 4.5 we will use the following simple claim, which is immediate from the Hoffeding bound.

Claim 4.6. *Let* $C\colon \{0,1\}^k \to \{0,1\}^\ell$ *and* $P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$ *be two circuits. Let* $\mathbf{A}$ *be a random variable distributed according to the following process:*
Sample $(x_1, s_1) \ldots, s(x_{n^{20}}, s_{n^{20}}) \leftarrow P(\mathbf{U}_r)$, *and let* $\mathbf{A} = \Pr_{i \leftarrow [n^{20}]}[|C(x_i) - s_i| \leq \Delta]$.
Let $\mu = \Pr_{(x,s) \leftarrow P(\mathbf{U}_r)}[|C(x) - s| \leq \Delta]$. *Then* $\Pr\left[|\mathbf{A} - \mu| \geq n^{-9}\right] \leq 2^{-n}$.

Proof. Immediate by Fact 2.14. □

Proof of Lemma 4.5. When $\mathsf{Gen}(1^n)$ outputs C_0 and P_0, the scheme has perfect correctness. Moreover, when the secret and public key are (k, ℓ, C) and (r, k, ℓ, P), it holds that

$$\Pr\left[|\mathsf{Dec}((k, \ell, C), \mathbf{M})) - \mathbf{O}| \leq \Delta(1^n)/n^{10}\right] = \Pr_{(x,s) \leftarrow P(\mathbf{U}_r)}\left[|C(x) - s| \leq \Delta(1^n)/n^{10}\right].$$

Thus,

$$\begin{aligned}
&\Pr\left[|\mathsf{Dec}((\mathbf{K}_n, \mathbf{L}_n, \mathbf{C}_n), \mathbf{M}_n) - \mathbf{O}_n| \le \Delta(1^n)/n^{10}\right] \\
&\ge (1 - 3n^{-8}) \cdot \Pr_{\mathbf{R}_n, \mathbf{K}_n, \mathbf{P}_n, \mathbf{C}_n}\left[\Pr_{(x,s) \leftarrow \mathbf{P}_n(\mathbf{U}_{\mathbf{R}_n})}\left[|\mathbf{C}_n(x) - s| \le \Delta(1^n)/n^{10}\right] \ge 1 - 3n^{-8}\right] \\
&\ge 1 - 3n^{-8} - \Pr_{\mathbf{R}_n, \mathbf{K}_n, \mathbf{P}_n, \mathbf{C}_n}\left[\Pr_{(x,s) \leftarrow \mathbf{P}_n(\mathbf{U}_{\mathbf{R}_n})}\left[|\mathbf{C}_n(x) - s| \le \Delta(1^n)/n^{10}\right] < 1 - 3n^{-8}\right].
\end{aligned}$$

The lemma now follows since by Lemma 4.5, the probability of circuits P, C with

$$\Pr_{(x,s) \leftarrow P(\mathbf{U}_r)}\left[|\mathbf{C}_n(x) - s| \le \Delta(1^n)/n^{10}\right] < 1 - 3n^{-8}$$

to pass the test in Step 3 is at most 2^{-n}. □

4.2 Security

We next bound the leakage of the scheme.

Lemma 4.7. *Assume there exists an algorithm* Eve *such that*

$$\Pr[|\mathsf{Eve}(1^n, (\mathbf{R}_n, \mathbf{K}_n, \mathbf{L}_n, \mathbf{P}_n), \mathbf{M_n}) - \mathbf{O}_n| \le \Delta(1^n)] \ge 1 - n^{-6}$$

for infinitely many n's. Then $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \in \mathsf{ioBPP}$.

In the following, let Eve be an algorithm that uses $r_{\mathsf{Eve}}(n)$ bits of randomness and guesses $\mathbf{M}$ with probability at least $1 - n^{-6}$. Recall that for $w \in \{0,1\}^{r_{\mathsf{Eve}}(n)}$, $\mathsf{Eve}(1^n, (r, k, \ell, P), m; w)$ denotes the execution of Eve when its randomness is fixed to be w.

Algorithm 4.8.

Input: $P \in \{0,1\}^n$.

Operation:

1. *Let* r, k, ℓ *be such that* $P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$.
2. *Sample* $w \leftarrow \{0,1\}^{r_{\mathsf{Eve}}(n)}$ *uniformly at random.*
3. *Construct a circuit* C' *such that* $C'(x) = \mathsf{Eve}(1^n, (r, k, \ell, P), x; w)$.
4. *Return* C'.

We prove the following lemma.

Lemma 4.9. *Assume that*

$$\Pr[|\mathsf{Eve}(1^n, (\mathbf{R}_n, \mathbf{K}_n, \mathbf{L}_n, \mathbf{P}_n), \mathbf{M}_n) - \mathbf{O}_n| \le \Delta(1^n)] \ge 1 - n^{-6}$$

for infinitely many n's. Then the following holds for infinitely many n's. For every input $P \in \{0,1\}^n \cap \mathcal{Q}_t^\Delta$, *Algorithm 4.8 outputs* $C \in \mathsf{Comp}_{1/4}^\Delta(P)$ *with probability at least* $1/2$.

We prove Lemma 4.9 below, but first we use it to prove Lemma 4.7.

Proof of Lemma 4.7. Assume there exists an algorithm Eve such that

$$\Pr[|\mathsf{Eve}(1^n, (\mathbf{R}_n, \mathbf{K}_n, \mathbf{L}_n, \mathbf{P}_n), \mathbf{M}_n) - \mathbf{O}_n| \leq \Delta(1^n)] \geq 1 - n^{-6}$$

for infinitely many n's.

By Lemma 4.9, for infinitely many n's, Algorithm 4.8 outputs $C \in \mathsf{Comp}^{\Delta}_{1/4}(P)$ with probability at least $1/2$, for every $P \in \{0,1\}^n \cap \mathcal{Q}^{\Delta}_t$. We want to construct an algorithm $\textsc{Sol}$ that, given P and $1^{1/\delta}$, outputs $C \in \mathsf{Comp}_{1/3}(P)$ with probability at least $1-\delta$, for every such n.

Let $\textsc{Sol}$ be the algorithm that, given input $(P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell, 1^{1/\delta})$, first run Algorithm 4.8 on P for $2\lceil \log 1/\delta \rceil$ times, to get circuits $C_1, \ldots, C_{2\lceil \log 1/\delta \rceil}$. Then, for every circuit C_i, $\textsc{Sol}$ samples $(x_1, s_1), \ldots, (x_{(100\lceil \log 1/\delta \rceil)^2}, s_{(100\lceil \log 1/\delta \rceil)^2}) \leftarrow P(\mathbf{U}_r)$, and computes

$$p_i = \Pr_{j \leftarrow [(100\lceil \log 1/\delta \rceil)^2]}[|C_i(x_j) - s_j| \leq \Delta(1^n)].$$

Finally, $\textsc{Sol}$ outputs the circuit C_i for the index i with maximal value of p_i.

We now analyze the success probability of $\textsc{Sol}$. Using *Fact* 2.14, for every i, the probability that $\left|p_i - \Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|C_i(x) = s| \leq \Delta(1^n)]\right| \geq 1/25$ is at most

$$2^{-4(\lceil \log 1/\delta \rceil)^2+1} \leq 1/(\delta(4\lceil \log 1/\delta \rceil)).$$

Thus, by the union bound, the probability that $\left|p_i - \Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|C_i(x) = s| \leq \Delta(1^n)]\right| \geq 1/25$ for some i is at most $\delta/2$. Next, by the success probability of Algorithm 4.8, with probability at least

$$1 - (1-1/2)^{2\log 1/\delta} \geq 1 - \delta/2,$$

at least one of the circuits $C_1, \ldots, C_{2\log 1/\delta}$ is in $\mathsf{Comp}^{\Delta}_{1/4}(P) \subseteq \mathsf{Comp}^{\Delta}_{1/3}(P)$. Let i^* be the index of such a circuit. Then, with probability at least $1 - \delta/2 - \delta/2 = 1 - \delta$, such i^* exists, and p_{i^*} is at least $(3/4 - 1/25)$, while for every i with $C_i \notin \mathsf{Comp}^{\Delta}_{1/3}(P)$, p_i is at most

$$(2/3 + 1/25) < (3/4 - 1/25) \leq p_{i^*},$$

which implies that the output of $\textsc{Sol}$ is in $\mathsf{Comp}^{\Delta}_{1/3}(P)$. □

4.3 Proving Lemma 4.9

To prove Lemma 4.9, we start with the following claim.

Claim 4.10. *Let $P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$ be a circuit, and assume that*

$$\Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|\mathsf{Eve}(1^n, (r,k,\ell,P), x) - s| \leq \Delta(n)] \geq 9/10.$$

Then, on input P, Algorithm 4.8 outputs $C \in \mathsf{Comp}^{\Delta}_{1/4}(P)$ with probability at least $1/2$.

Proof of Claim 4.10. Let $P\colon \{0,1\}^r \to \{0,1\}^k \times \{0,1\}^\ell$ be a circuit such that

$$\Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|\mathsf{Eve}(1^n, (r,k,\ell,P), x) - s| \leq \Delta(n)] \geq 9/10.$$

Then, by definition it holds that

$$\mathrm{E}_{w\leftarrow\{0,1\}^{r_{\mathsf{Eve}}(n)}}\left[\Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|\mathsf{Eve}(1^n,(r,k,\ell,P),x)-s|>\Delta(n)]\right]\leq 1/10.$$

Using Markov's inequality, we gets that

$$\Pr_{w\leftarrow\{0,1\}^{r_{\mathsf{Eve}}(n)}}\left[\Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|\mathsf{Eve}(1^n,(r,k,\ell,P),x)-s|>\Delta(n)]>1/4\right]\leq 1/2,$$

which implies that the circuit $C'=\mathsf{Eve}(1^n,(r,k,\ell,P),\cdot;w)$ is in $\mathsf{Comp}^{\Delta}_{1/4}(P)$ with probability at least $1/2$ over the choice of w, as we wanted to show. □

Given Claim 4.10, we are now ready to prove Lemma 4.9.

Proof of Lemma 4.9. Assume that Eve is such that

$$\Pr[|\mathsf{Eve}(1^n,(\mathbf{R}_n,\mathbf{K}_n,\mathbf{L}_n,\mathbf{P}_n),\mathbf{M}_n)-\mathbf{O}_n|\leq\Delta(1^n)]\geq 1-n^{-6}$$

for infinitely many n's. In the following, fix such large enough $n\in\mathbb{N}$. We show that

$$\Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|\mathsf{Eve}(1^n,(r,k,\ell,P),x)-s|\leq\Delta(n)]\geq 9/10 \tag{1}$$

for every $P\colon\{0,1\}^r\to\{0,1\}^k\times\{0,1\}^\ell$ with $P\in\mathcal{Q}^{\Delta}_t\cap\{0,1\}^n$. The proof then follows by Claim 4.10. To see the above, let $\mathbf{Pk}=(\mathbf{R}_n,\mathbf{K}_n,\mathbf{L}_n,\mathbf{P}_n))$, and notice that

$$\Pr_{\substack{\mathbf{R}_n,\mathbf{K}_n,\mathbf{L}_n,\mathbf{P}_n,\mathbf{C}_n\\ \mathbf{Pk}=(\mathbf{R}_n,\mathbf{K}_n,\mathbf{L}_n,\mathbf{P}_n)}}\left[\Pr_{\mathbf{M}_n,\mathbf{O}_n,\mathbf{W}\leftarrow\{0,1\}^{r}_{\mathsf{Eve}}}[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n;\mathbf{W})-\mathbf{O}_n|\leq\Delta(n)]<9/10\right]\leq 10n^{-6}. \tag{2}$$

Indeed, it holds that

$$\begin{aligned}
1-n^{-6}&\leq\Pr[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n)-\mathbf{O}_n|\leq\Delta(n)]\\
&\leq\Pr_{\mathbf{Pk}}[\Pr[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n)-\mathbf{O}_n|\leq\Delta(n)]\geq 9/10]\\
&\qquad+9/10\cdot\Pr_{\mathbf{Pk}}[\Pr[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n)-\mathbf{O}_n|\leq\Delta(n)]<9/10]\\
&=(1-\Pr_{\mathbf{Pk}}[\Pr[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n)-\mathbf{O}_n|\leq\Delta(n)]<9/10])\\
&\qquad+9/10\cdot\Pr_{\mathbf{Pk}}[\Pr[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n)-\mathbf{O}_n|\leq\Delta(n)]<9/10]\\
&=1-1/10\cdot\Pr_{\mathbf{Pk}}[\Pr[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n)-\mathbf{O}_n|\leq\Delta(n)]<9/10]
\end{aligned}$$

which implies that Equation (2) holds. Next, we use Equation (2) and the upper bound on the computational depth of instances in $\mathcal{Q}^{\Delta}_t$, to show that Equation (1) holds for every $P\in\mathcal{Q}^{\Delta}_t$. To do so, fix $P\in\mathcal{Q}^{\Delta}_t\cap\{0,1\}^n$ and let $C\in\{0,1\}^n\cap\mathsf{Comp}^{\Delta/n^{10}}_{n^{-10}}(P)$ with $cd^{t,\infty}(C,P)\leq 2\log n$ be the circuit promised by the definition of $\mathcal{Q}^{\Delta}_t$. Assume towards a contradiction that

$$\Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|\mathsf{Eve}(1^n,(r,k,\ell,P),x)-s|\leq\Delta(n)]<9/10.$$

We want to upperbound $\mathsf{K}(P,C)$, to get a lower bound on the computational depth of (P,C). To this end, let $\mathcal{S}$ be the set of all pairs $(P',C') \in \{0,1\}^n \times \{0,1\}^n$, such that $C' \in \mathsf{Comp}_{n^{-10}}^{\Delta/n^{10}}(P')$, $\mathsf{K}^t(P',C') = \mathsf{K}^t(P,C)$, and on which Eve fails to approximate s with probability more than $1/10$.

By our assumption on (P,C), it holds that $(P,C) \in \mathcal{S}$. We next bound the size of $\mathcal{S}$. First, we claim that for every $(P',C') \in \mathcal{S}$,

$$\Pr\big[\mathbf{P}_n = P', \mathbf{C}_n = C'\big] \geq 1/6n \cdot 2^{-\mathsf{K}^t(C,P)}. \tag{3}$$

Indeed, by definition there exists a program Π of length $\mathsf{K}^t(C,P)$ that outputs (C',P') in at most $t(2n)$ steps. Thus, Gen samples Π with probability at least $1/3n \cdot 2^{-\mathsf{K}^t(C,P)}$. Next, by Claim 4.6, the test in Step 3 of Gen passes with probability at least $1-2^{-n} > 1/2$, since by definition of $\mathsf{Comp}_{n^{-10}}^{\Delta/n^{10}}$, $\Pr_{(x,s)\leftarrow P'(\mathbf{U}_r}\big[|C'(x)-s| \leq \Delta(1^n)/n^{10}\big] - n^{-9} \geq 1 - n^{-10} - n^{-9} \geq 1 - 2n^{-7}$. In this case that the test passes, P' and C' are the output of Gen, and thus Equation (3) holds.

By Equation (3) and the definition of $\mathcal{S}$, we get that

$$\Pr_{\substack{\mathbf{R}_n,\mathbf{K}_n,\mathbf{L}_n,\mathbf{P}_n,\mathbf{C}_n \\ \mathbf{Pk}=(\mathbf{R}_n,\mathbf{K}_n,\mathbf{L}_n,\mathbf{P}_n)}}\left[\Pr_{\mathbf{M}_n,\mathbf{O}_n,\mathbf{W}\leftarrow\{0,1\}^r_{\mathsf{Eve}}}[|\mathsf{Eve}(1^n,\mathbf{Pk},\mathbf{M}_n;\mathbf{W}) - \mathbf{O}_n| \leq \Delta(n)] < 9/10\right] \geq |\mathcal{S}| \cdot 1/6n \cdot 2^{-\mathsf{K}^t(C,P)}.$$

Combining the above with Equation (2) yields that

$$|\mathcal{S}| \leq \frac{10n^{-6}}{1/6n \cdot 2^{-\mathsf{K}^t(C,P)}} \leq 2^{\mathsf{K}^t(C,P)+\log n - 6\log n + 6}.$$

Observe that $\mathcal{S}$ can be (inefficiently) computed given $n, \mathsf{K}^t(C,P)$ and Eve. Thus, to encode (P,C), it is enough to encode $\mathcal{S}$ and the index of (P,C) in $\mathcal{S}$ (according to the lexicographic order). We conclude that,

$$\begin{aligned}
\mathsf{K}(P,C) &\leq \mathsf{K}(n,\lambda,\mathsf{Eve}) + 2\log(\mathsf{K}(n,\lambda,\mathsf{Eve})) + \log|\mathcal{S}| + O(1) \\
&\leq 2\log n + 4\log\log n + \mathsf{K}^t(C,P) - 5\log n + O(1) \\
&< \mathsf{K}^t(C,P) - 2\log n,
\end{aligned}$$

where the last inequality holds for every large enough n. By the above, $\mathsf{K}^t(C,P) - \mathsf{K}(P,C) > 2\log n$, in contradiction to the choice of (P,C). This yields that Equation (1) holds for every $P \in \mathcal{Q}_t$, as we wanted to show. □

4.4 Proving Theorem 4.1.

We are now ready to use Lemma 2.8 in order to prove Theorem 4.1.

Proof of Theorem 4.1. By Lemmas 4.5 and 4.7, $(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ is a $(n^{-7}, 1-n^{-6}, n^{10})$-weak-PKE (for $d(1^n) = \Delta(1^n)/n^{10}$), and thus it is also a $(n^{-6.9}/2, 1-10n^{-6.1}, 2n^{6.9})$-weak-PKE. Thus, by Lemma 2.8, $(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ can be amplified into a PKE. □

Remark 4.11 (The non-uniform setting). *A similar theorem can be proven when assuming that* $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioP/poly}$*, and when the PKE is secure against non-uniform adversaries. In this case, we assume that* Eve *is a non-uniform algorithm that breaks the PKE protocol, and want to construct a non-uniform (randomized) algorithm that decides* $\mathsf{WBLearn}|_{\mathcal{Q}_t}$.

The issue with the above proof is that we cannot simply use Eve *to bound the Kolmogorov complexity of* (C, P) *as done in the proof of Lemma 4.7, as* Eve *does not have constant size. However, we can find* Eve *using a small Turing machine: Let* M *be the (inefficient) Turing machine that, given a constant* c *such that* n^c *is a bound on the size of* Eve*, and an input for* Eve*, first find the circuit* E'_n *of size at most* n^c *that maximize the advantage in predicting* M *given an encryption of* M *by* Enc*, and then execute* E' *on the input. Observe that* M *has prediction advantage at least as the advantage of* Eve*. The theorem now follows using the same proof, by replacing* Eve *in the proof of Lemma 4.7 with* M*, and replacing* Eve *in Algorithm 4.8 with* $E' = \{E'_n\}_{n\in\mathbb{N}}$.

5 PKE $\implies$ Hardness of Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t}$

In this part, it is shown that if a public-key encryption scheme exists, Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t}$ is hard. We prove the following theorem.

Theorem 5.1. *Assume there exists a PKE. Then for any constant* $\epsilon > 0$ *and any* $t(n) \geq n^{1+\epsilon}$ *and any efficiently computable* $\Delta\colon \{1\}^* \to \mathbb{N}$ *such that* $\Delta(1^n) \leq 2^{n^{(1-\epsilon)}}$, Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$.

To prove Theorem 5.1, it will be convenient to assume that in the PKE scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, the public and secret keys are simply the circuits used to encrypt and decrypt, respectively. For this, we define *circuit-PKE (cPKE)* and show that we can assume the above without loss of generality.

Definition 5.2 (circuit PKE (cPKE))**.** *A randomized, function* Gen' *is a* $(\alpha(n), \beta(n))$-circuit PKE (cPKE) *if the following holds:*

- $\mathsf{Gen}(n)$ *outputs two circuits* $pk\colon \{0,1\} \times \{0,1\}^{r(n)} \to \{0,1\}^{k(n)}$ *and* $sk\colon \{0,1\}^{k(n)} \to \{0,1\}$.
- *Correctness: For every large enough* $n \in \mathbb{N}$ *and any* $m \in \{0,1\}$,

$$\Pr_{(pk,sk)\leftarrow\mathsf{Gen}(1^n), w\leftarrow\{0,1\}^{r(n)}}[sk(pk(m,w)) = m] \geq 1 - \alpha(n)$$

- *Security: For every* PPT Eve*, for every large enough* $n \in \mathbb{N}$,

$$\Pr_{(pk,sk)\leftarrow\mathsf{Gen}(1^n), m\leftarrow\{0,1\}, w\leftarrow\{0,1\}^{r(n)}}[\mathsf{Eve}(1^n, pk, pk(m,w)) = m] \leq 1/2 + \beta(n).$$

Such a scheme is a cPKE *if it is* $(1/n^c, 1/n^c)$*-cPKE for every constant* $c \in \mathbb{N}$.

We say that cPKE has key of length ℓ if the description size of pk, sk is at most $n/2$ bits.

Lemma 5.3. *Assume there exists a PKE. Then there exists a cPKE.*
Moreover, under the same assumption there exists a cPKE such that $\mathsf{Gen}(n)$ *runs in time* n^ϵ *and has keys of length* n^ϵ.

Proof. Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE, and assume without loss of generality that Dec is deterministic algorithm (by adding its randomness to the secret-key). Let $t(n)$ be an upper bound on the running time of Enc when executed on pk sampled from $(\cdot, pk) \leftarrow \mathsf{Gen}(1^n)$ and a message $m \in \{0,1\}$. Observe that $t(n)$ bounds both the randomness used by Enc and the length of the output of $\mathsf{Enc}(pk, m)$. By using appropriate padding, assume without loss of generality that $|\mathsf{Enc}(pk, m)| = t(n)$ for any m and every $(sk, pk) \leftarrow \mathsf{Gen}(1^n)$, and let $r(n) = k(n) = t(n)$.

Let $\mathsf{Gen}'(n)$ be the algorithm that execute $\mathsf{Gen}(1^n)$ to get (sk, pk), and then construct circuits $sk'\colon \{0,1\}^{k(n)} \to \{0,1\}$ and $pk'\colon \{0,1\} \times \{0,1\}^{r(n)} \to \{0,1\}^{k(n)}$ such that $sk'(e) = \mathsf{Dec}(sk, e)$ for every $e \in \{0,1\}^{k(n)}$, and $pk'(m, w) = \mathsf{Enc}(pk, m; w)$. It is not hard to see that Gen' can be implemented in polynomial time, and that it has the same correctness and security as $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

The moreover part follows by a simple security leveraging. Let $t(n)$ be an upper bound on the length of sk', pk' and the running time of Gen'. Let δ be a constant such that $t(n^\delta) \le n^\epsilon$ (which exists since t can be bounded with a polynomial). Let $\mathsf{Gen}''(1^n) := \mathsf{Gen}'(1^{n^\epsilon})$. It is not hard to see that Gen'' has a running time and keys of length n^ϵ, where the security follows by the security of Gen. $\square$

The main lemma in this part states that the conclusion of Theorem 5.1 holds assuming cPKE exists.

Lemma 5.4. *Assume that for any constant ϵ there is a cPKE such that $\mathsf{Gen}(n)$ runs in time n^ϵ, and has keys of length n^ϵ. Then for any constant ϵ, Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$ for every $t(n) \ge n^\epsilon$ and any $\Delta \le 2^{n^{1-\epsilon}}/4$.*

Proof of Lemma 5.4. Let Gen' be a cPKE. Let Gen be tha algorithm that samples $(pk', sk') \leftarrow \mathsf{Gen}'(1^n; r)$, and construct a circuit pk'' that on input (m, w) outputs $pk'(m, w), (m||0^{n^{(1-2\epsilon)}})$ (where $m||0^{(n^{1-\epsilon})}$ is the binary representation of the number $m \cdot 2^{n^{(1-\epsilon)}}$). Similarly, let sk'' be the circuit that on input e outputs $sk(e)||0^{(n^{1-\epsilon})}$. Note that $|sk|, |pk| \le n/2$. let $pk = pk''||0^{|pk''|-n/2}$ and $sk = sk''||r$. In the following, assume toward a contradiction that Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t} \in \mathsf{ioFBPP}$, and let $\textsc{Sol}$ be the algorithm that, for infinitely many n's, solves Δ-$\mathsf{WBLearn}|_{\mathsf{CS}^t}$. We will show that $\textsc{Sol}$ contradicts the secrecy assumption of Gen. We observe that assuming we can approximating the output of pk, it follows using rounding that we can guess the MSB of the output with the same probability (as the output of pk is either 0 or $2^{n^{1-\epsilon}}$). That is, we can guess the value of m. In the following we assume that $\textsc{Sol}$ simply outputs the MSB of the output of pk.

Specifically, consider the following algorithm Eve that uses $\textsc{Sol}$ to guess the message m given an encryption $e = pk(m, w)$.

Algorithm 5.5 (Eve)**.**

Input: pk, e.

Operation:

1. *Execute* $\textsc{Sol}(pk, 1^{100})$ *to get a circuit C.*[13]

2. *Compute $C(e)$ and output its output.*

. . . .

First, notice that Eve is an efficient algorithm if $\textsc{Sol}$ is. Indeed, since C is the output of $\textsc{Sol}$, its size is bounded by the running time of $\textsc{Sol}$, and thus C can be evaluated on e in polynomial time.

Next, observe that if for infinitely many n's,

$$\Pr_{\substack{(sk,pk)\leftarrow\mathsf{Gen}(n)|E_n,\\ m\leftarrow\{0,1\},e\leftarrow pk(m,\mathbf{U}_{r(n)})}}[C(e) = m] \ge 1/2 + 1/50, \tag{4}$$

[13]Recall that the second input for $\textsc{Sol}$ is the accuracy parameter, see Definition 2.3.

then Eve breaks the security of Gen. By assumption, for every $P \in \mathcal{Q}_{t'}$, with probability at least $1 - 1/100$, $\textsc{Sol}(P, 1^{100})$ finds a circuit $C \in \mathsf{Comp}_{1/3}(P)$. In this case, C agrees with P with probability at least $2/3$. Thus,

$$\Pr_{\substack{(sk,pk)\leftarrow\mathsf{Gen}(n),\\ m\leftarrow\{0,1\},e\leftarrow pk(m,\mathbf{U}_{r(n)})\\ C\leftarrow\textsc{Sol}(pk,1^{100})}}[C(e) = m] \geq \Pr_{(sk,pk)\leftarrow\mathsf{Gen}(n)}[pk \in \mathcal{Q}_{t'}] \cdot (1 - 1/100) \cdot 2/3.$$

So to show Equation (4), it is enough to show that

$$\Pr_{(sk,pk)\leftarrow\mathsf{Gen}(n)}[pk \in \mathcal{Q}_{t'}] \geq 4/5. \tag{5}$$

To do so, recall that pk and sk are two circuits with description size n. Moreover, by the correctness of the PKE scheme, with all but negligible probability over the choice of $(sk, pk) \leftarrow \mathsf{Gen}(n)$,

$$\Pr_{m\leftarrow\{0,1\},e\leftarrow pk(m;\mathbf{U}_{r(n)})}[sk(e) = m] \geq 1 - n^{-10}.$$

Thus, we are only left to show that $cd^{t',\infty}(C, P) \leq 2 \log n$ with high probability.

To bound the computational depth, first notice that $\mathsf{K}^{t'}(C, P) \leq \mathsf{K}^{t}(C, P) \leq s(n) + \log n + O(1)$ where $s(n)$ is the amount of randomness used by Gen. Indeed, to encode pk and sk it is enough to encode the algorithms Gen together with n and the randomness used by Gen to sample pk, sk.

On the other hand, since sk contains the randomness used by Gen, the distribution of (pk, sk) sampled by $\mathsf{Gen}(n)$ has min-entropy $s(n)$, we get by Lemma 2.13 that with probability at least $1 - 1/100$, $\mathsf{K}(P, C) \geq s(n) - 0.5 \log n$. Combining the above together, we get that $cd^{t',\infty}(C, P) \leq 1.5 \log n + O(1) < 2 \log n$ with probability at least $1 - 1/100$. Overall, the correctness property and the above bound on the computational depth holds simultaneously with probability strictly larger than $4/5$, and so Equation (5) holds. $\square$

Proving Theorem 5.1

Proof of Theorem 5.1. The proof follows by Lemmas 5.3 and 5.4. $\square$

6 Bounded-Degree Learning

In this part we consider the $\mathsf{WBLearn}$ problem, when we restrict the target function to be a low degree polynomial. In the following we fix some $q = q(n)$, and all the operations are with respect to $\mathbb{Z}_q$. We fix an encoding of elements in $\mathbb{Z}_q$ in $\{0, 1\}^{\lceil \log q \rceil}$, and consider pairs of circuit P and circuit C of the form $P \colon \{0, 1\}^r \to \mathbb{Z}_q^k \times \mathbb{Z}_q$ and $C \colon \mathbb{Z}_q^k \to \mathbb{Z}_q$. We define

$$\mathsf{Comp}_{\epsilon}^{\Delta}(P) = \left\{ C \colon \mathbb{Z}_q^k \to \mathbb{Z}_q \colon \Pr_{(x,s)\leftarrow P(\mathbf{U}_r)}[|C(x) - s| \leq \Delta] \geq 1 - \epsilon \right\},$$

where the "$-$" operation is now over $\mathbb{Z}_q$. For any bound d on the degree, we define,

Definition 6.1 (Δ-$\mathsf{WBLearn}^d$)**.** *For functions $\Delta, q \colon \{1\}^* \to \mathbb{N}$, let Δ-$\mathsf{WBLearn}_q^d$ be the following learning problem:*

- *Input: Circuit $P \in \{0,1\}^n$, with the promise that there exists a circuit $\widehat{C} \in \{0,1\}^n$ such that $\widehat{C} \in \mathsf{Comp}_{n^{-10}}^{\Delta(1^n)/n^{10}}(P)$ and $\widehat{C}$ computes a degree d polynomial (over $\mathbb{Z}_{q(1^n)}$) on its input.*
- *Output: Circuit $C \in \mathsf{Comp}_{1/3}^{\Delta(1^n)}(P)$*

We similarly define $\Delta\text{-}\mathsf{WBLearn}_q^d|_{\mathsf{CS}^t}$ to be the above problem with the additional promise that $cd^t(\widehat{C}, P) \leq 2\log n$.

The next theorem states that the hardness of $\Delta\text{-}\mathsf{WBLearn}_q^d|_{\mathsf{CS}^t}$ is equivalent to the existence of PKE in which the decoding procedure (for any fixed secret key) is a degree d polynomial. Recall that for circuit-PKE, the secret key is a circuit (computing the function $f(\cdot) = \mathsf{Dec}(sk, \cdot)$). We thus say that the secret key computes a low degree polynomial if the above function f can be computed by a low degree polynomial.

Theorem 6.2. *Let $\epsilon > 0$ be a constant and let $\Delta, q\colon \{1\}^* \to \mathbb{N}$ be efficiently computable functions with $\Delta(1^n) \leq q(1^n)/4$, and $t\colon \mathbb{N} \to \mathbb{N}$ be polynomial such that $t(n) \geq n^{1+\epsilon}$. Assume circuit-PKE such that the secret key computes a degree d polynomial over $\mathbb{Z}_q$. Then $\Delta\text{-}\mathsf{WBLearn}_{q'}^d|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$ for some efficiently computable function q'.*

Theorem 6.2 follows by the exact same proof of Theorem 5.1. We also note that any boolean function over n variables is computable by a exponentially long polynomial of degree n (where we use the natural embedding of $\{0,1\}$ in $\mathbb{Z}_q$). Using this fact in the proof of Lemma 5.3, we get the following stronger theorem:

Theorem 6.3. *Let $\epsilon > 0$ be a constant and let $\Delta, q\colon \{1\}^* \to \mathbb{N}$ be efficiently computable functions with $\Delta(1^n) \leq q(1^n)/4$, $q(1^n) \leq 2^{n^{1-\epsilon}}$, and $t\colon \mathbb{N} \to \mathbb{N}$ be polynomial such that $t(n) \geq n^{1+\epsilon}$. The following are equivalent:*

- *PKE exists.*
- $\Delta\text{-}\mathsf{WBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$.
- $\Delta\text{-}\mathsf{WBLearn}_q^n|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$.
- $\Delta\text{-}\mathsf{WBLearn}_q^{n^\epsilon}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$.

6.1 Learning with Errors and $\Delta\text{-}\mathsf{WBLearn}_q^1|_{\mathsf{CS}^t}$

We next discuss LWE as a specific case of $\Delta\text{-}\mathsf{WBLearn}_q^d|_{\mathsf{CS}^t}$. First, we recall Regev's PKE [Reg09]. This encryption scheme (parametrized by n, m, q, B) can be described as follows (where all the operation are in $\mathbb{Z}_q$):

- The secret key is a random vector $\mathbf{x} \in \mathbb{Z}_q^n$.
- The public key is a pair $(\mathbf{A}, \mathbf{b})$ where $\mathbf{A}$ is a matrix of size $m \times n$ and $\mathbf{b}$ is a vector of size m, for some $m = m(n) \geq \Omega(n \log q)$. $\mathbf{A}$ is a uniformly random matrix in $\mathbb{Z}_q^{m \times n}$, and $\mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{e}$ where each e_i is sampled from a fixed noise distribution such that $|e_i| \leq B$. (Let $c \geq 1$ be a constant such that the public key can be sampled in time n^c.)
- To encrypt a bit $m \in \{0,1\}$, we sample a random $\mathbf{r} \in \{0,1\}^m$, and compute $\mathbf{a}' = \mathbf{r}^T\mathbf{A}, u = \mathbf{r} \cdot \mathbf{b} + m \cdot (q/2)$.

- To decrypt, we compute $\mathbf{a}' \cdot \mathbf{x} - u$, and check if the result is closer to 0 (in which case we output 0) or to $q/2$ (in which case we output 1).

Observe that Regev's PKE has a decryption algorithm that can be implemented by an inner product (which is a linear function over $\mathbb{Z}_q$) together with a rounding gate. Combining this with Theorem 6.2, it follows that the security of Regev's PKE implies the hardness of the learning problem $\Delta\text{-}\mathsf{WBLearn}^d_q|_{\mathsf{CS}^t}$ where d is the degree of rounding. Next, we aim to show that the implication still holds if d is fixed to be 1; that is, it implies the hardness of learning linear functions.

Lemma 6.4. *Assume that Regev's PKE scheme is secure with $B \le q/(4mn^{20c})$. Then, there exists a polynomial t and an efficiently computable function $q' \colon \mathbb{N} \to \mathbb{N}$ such that $\Delta\text{-}\mathsf{WBLearn}^1_{q'}|_{\mathsf{CS}^t} \notin$ ioFBPP where $\Delta = q'(1^n)/4$.*

Proof. Consider $t(n) = n^{4c}$ (where c defined as above is the exponent in the runtime bound of public key generation). For any $\mathbf{x} \in \mathbb{Z}_q^n$, define P as the circuit (hardcoding $\mathbf{A}$ and $\mathbf{b}$) that on input $r \in \{0,1\}^m$, outputs $\mathbf{a}' = \mathbf{r}^T\mathbf{A}, b' = \mathbf{r} \cdot \mathbf{b}$. In addition, let C be the circuit (hardcoding $\mathbf{x}$ together with all the randomness used to generate $\mathbf{A}$ and $\mathbf{b}$) that on input $\mathbf{a} \in \mathbb{Z}_q^n$, outputs $\mathbf{a} \cdot \mathbf{x} \in \mathbb{Z}_q$. Notice that (the description length) $|P|, |C| \le n^{2c}$ (since $(\mathbf{A}, \mathbf{b})$ can be generated in time n^c).

We move on to proving that P satisfies the promise in our learning problem $\Delta\text{-}\mathsf{WBLearn}^1_q|_{\mathsf{CS}^t}$. The "witness" for the circuit P is just the circuit C. Observe that C is a linear function over $\mathbb{Z}_q$. Next, we show that P produces instances together with noised labels that C computes. Note that $|C(\mathbf{a}') - b'| = |\mathbf{r}^T\mathbf{A}\mathbf{x} - \mathbf{r}\cdot\mathbf{b}| = |\mathbf{r}\cdot\mathbf{e}| \le m \cdot B \le \Delta/(n^{2c})^{10}$. It remains to argue that (C, P) has small computational depth. Since (C, P) together hardcodes all the randomness, it follows from the last two paragraphs in the proof of Lemma 5.4 that this happens with probability at least $1 - 1/100$.

Given the public key $(\mathbf{A}, \mathbf{b})$, we prepare the circuit P and feed it to the $\Delta\text{-}\mathsf{WBLearn}^1_q|_{\mathsf{CS}^t}$ attacker. If P satisfies the promise in $\Delta\text{-}\mathsf{WBLearn}^1_q|_{\mathsf{CS}^t}$ (which happens with probability at least 1/100), it will find a circuit C' (with probability $1 - 1/100$) such that

$$\Pr_{\mathbf{r} \leftarrow \{0,1\}^m}[|C'(\mathbf{a}') - b'| \le q/4] \ge 2/3$$

This will allow us to decrypt a ciphertext $(\mathbf{a}', u)$ by computing $C'(\mathbf{a}') - u$ and rounding to either 0 or $q/2$. Thus, we can distinguish the encryption of 0 and encryption of 1 with advantage at least $2/3 - 2/100$. Finally, we need to deal with the issue that the $\Delta\text{-}\mathsf{WBLearn}^1_q|_{\mathsf{CS}^t}$ attacker only succeeds on infinitely many input lengths. We will instead prepare many padded copies of P, one copy for each description length in $[n^{2c}, (n+1)^{2c})$, and feed all of them to the attacker. The attacker will find a circuit C' for each copy, and we can test whether C' breaks the PKE scheme by generating encryption of zeros and ones and estimating the distinguishing gap. We will deploy C' with the best (empirical) distinguishing gap. $\square$

6.2 $\Delta\text{-}\mathsf{WBLearn}^d_q$ with constant degree

We next observe that for any constant degree d, there is a reduction between $\Delta\text{-}\mathsf{WBLearn}^d_q$ and $\Delta'\text{-}\mathsf{WBLearn}^1_q$ (with slightly different parameters). For simplicity, we prove it for the exact version of the problems.

Lemma 6.5. *For every constant $d \in \mathbb{N}$ and efficiently computable function $q\colon \{1\}^* \to \mathbb{N}$, there exists an efficiently computable function q' such that* $\mathsf{ExactWBLearn}^d_q \leq_p \mathsf{ExactWBLearn}^1_{q'}$.

Note that the other direction holds trivially.

Proof. Given a circuit $P\colon \{0,1\}^r \to \mathbb{Z}^k_q \times \mathbb{Z}_q$, the reduction outputs the following circuit P'. Let $P'\colon \{0,1\}^r \to \mathbb{Z}_q^{\binom{k+1}{d+1}} \times \mathbb{Z}_q$ be the circuit that given an input w computes $P(w) = (s, m)$, where $s \in \mathbb{Z}^k_q$, sets $s' \in \mathbb{Z}_q^{\binom{k+1}{d+1}}$ to be all the degree at most d monomials over s, and outputs (s, m). The correctness follows since every circuit C that computes a degree-d polynomial over s, computes a linear function of s', and vice versa. $\square$

A similar reduction gives the following theorem.

Lemma 6.6. *For every constant $d \in \mathbb{N}$, and functions $t\colon \mathbb{N} \to \mathbb{N}, q\colon \{1\}^* \to \mathbb{N}$, there exists $t'\colon \mathbb{N} \to \mathbb{N}, q'\colon \{1\}^* \to \mathbb{N}$ such that* $\mathsf{ExactWBLearn}^d_q|_{\mathsf{CS}^t} \leq_p \mathsf{ExactWBLearn}^1_{q'}|_{\mathsf{CS}^{t'}}$.

Proof. The proof follows by the same reduction as Lemma 6.5, where we pad P' such that $|P'| \geq n^{d+1}$ (where n^{d+1} is an upper bound on the length of P'). Since P' can be computed efficiently from P, and P can be computed efficiently from P', we have that, for large enough t' and every circuit C, $cd^{t'}(P', C) \leq cd^t(P, C) \pm O(1)$. We get that $cd^{t'}(P', \widehat{C}) \leq cd^t(P, \widehat{C}) + O(1) \leq 2\log n + O(1) \leq \log|P'|$. $\square$

Using the same reduction we get the following result.

Lemma 6.7. *For every constant $d \in \mathbb{N}$, and functions $t\colon \mathbb{N} \to \mathbb{N}$, $q, \Delta\colon \{1\}^* \to \mathbb{N}$ there exist $t'\colon \mathbb{N} \to \mathbb{N}$ and $q', \Delta'\colon \{1\}^* \to \mathbb{N}$ such that*

$$\Delta\text{-}\mathsf{WBLearn}^d_q|_{\mathsf{CS}^t} \leq_p \Delta'\text{-}\mathsf{WBLearn}^1_{q'}|_{\mathsf{CS}^{t'}},$$

and

$$\Delta\text{-}\mathsf{WBLearn}^d_q \leq_p \Delta'\text{-}\mathsf{WBLearn}^1_{q'}.$$

7 Black-Box Distributional Learning and One-Way Functions

In this section, we consider the standard PAC learning model [Val84] where the learner only get access to samples. The following definition is identical to the definition of $\mathsf{WBLearn}$ with the exception that the learner gets oracle access (i.e., black-box access) to the sampler (as opposed to white-box access to the sampler). It is equivalent to the notion of improper Δ-approximate PAC learning for polynomial-size circuits (and when $\Delta = 0$ to simply improper PAC learning).

Definition 7.1 (Δ-$\mathsf{BBLearn}$)**.** *For a function $\Delta\colon \{1\}^* \to \mathbb{N}$, let Δ-$\mathsf{BBLearn}$ be the following learning problem:*

- *Input: Oracle access to samples from the output distribution of a circuit $P \in \{0,1\}^n$, with the promise that there exists a circuit $\widehat{C} \in \{0,1\}^n$ such that $\widehat{C} \in \mathsf{Comp}^{\Delta(1^n)/n^{10}}_{n^{-10}}(P)$.*
- *Output: Circuit $C \in \mathsf{Comp}^{\Delta(1^n)}_{1/3}(P)$*

As before, let $\Delta\text{-}\mathsf{BBLearn}|_{\mathsf{CS}^t}$ denote the problem $\Delta\text{-}\mathsf{BBLearn}$ with the additional promise that $cd^t(P, \widehat{C}) \le 2\log n$, and let $\mathsf{ExactBBLearn}$ and $\mathsf{ExactBBLearn}|_{\mathsf{CS}^t}$ to denote $\Delta\text{-}\mathsf{BBLearn}$ and $\Delta\text{-}\mathsf{BBLearn}|_{\mathsf{CS}^t}$ when $\Delta(1^n) = 0$.

The following theorem can be view as a worst-case analog of the classic result of [KV94; BFKL93] characterizing one-way functions through *average-case PAC learning*. (We remark that our theorem differs not only in the worst-case condition, but also generalizes [KV94; BFKL93] in the sense that we handle the hardness of Δ-approximate learning for any Δ.)

Theorem 7.2. *Let $\epsilon > 0$ be a constant and let $\Delta\colon \{1\}^* \to \mathbb{N}$ be any efficiently computable function such that $\Delta(1^n) \le 2^{n^{(1-\epsilon)}}$, and let $t\colon \mathbb{N} \to \mathbb{N}$ be any polynomial such that $t(n) \ge n^{1+\epsilon}$. Then the following are equivalent:*

- *One-way function exists*
- $\Delta\text{-}\mathsf{BBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$

To prove Theorem 7.2, we actually prove that the hardness of $\Delta\text{-}\mathsf{BBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$ is equivalent to the existence of *secret-key* encryption scheme. As a result, the proof of Theorem 7.2 is almost identical to the proof of Theorem 3.3. We will use an encryption scheme with security against Chosen Plaintext Attack (CPA).

Definition 7.3 (CPA-secure private-key encryption scheme)**.** *A triplet of randomized, efficiently computable functions* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a* $(\alpha(n), \beta(n))$-$\mathsf{CPA\text{-}secure\ private\text{-}key\ encryption\ scheme}$ *if the following holds:*

- *Correctness: For every large enough $n \in \mathbb{N}$ and any $m \in \{0,1\}$,*

$$\Pr_{sk \leftarrow \mathsf{Gen}(1^n), \mathsf{Enc}}[\mathsf{Dec}(sk, \mathsf{Enc}(sk, m)) = m] \ge 1 - \alpha(n)$$

- *CPA-Security: For every oracle-aided* PPT Eve*, for every large enough $n \in \mathbb{N}$,*

$$\Pr_{sk \leftarrow \mathsf{Gen}(1^n), m \leftarrow \{0,1\}}\left[\mathsf{Eve}^{\mathsf{Enc}_{sk}}(\mathsf{Enc}(sk, m)) = m\right] \le 1/2 + \beta(n),$$

  where Enc_{sk} is the (randomized) function defined by $\mathsf{Enc}_{sk}(m) = \mathsf{Enc}(sk, m)$.

Such a scheme is a $\mathsf{CPA\text{-}secure\ encryption\ scheme}$ *if it is $(1/n^c, 1/n^c)$-PKE for every $c \in \mathbb{N}$.*

The existence of such a scheme is equivalent to the existence of one-way functions.

Proposition 7.4 ([Gol04])**.** *CPA-secure encryption scheme exists if and only if one-way function exists.*

We can naturally define weak variant of CPA-secure encryption, that corresponds to our definition of weak-PKE (Definition 2.9).

Definition 7.5 (Weak CPA-secure encryption scheme)**.** *For an efficiently computable function $d\colon \{1\}^* \to \mathbb{N}$, a triplet of randomized, efficiently computable functions* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a* $(\alpha(n), \beta(n), \gamma(n))$-$\mathsf{weak\ CPA\text{-}secure\ encryption\ scheme}$ *if the following holds:*

- *For every $n \in \mathbb{N}$, given sk and randomness r,* $\mathsf{Enc}(sk; r)$ *outputs a message $m(sk; r)$ and an output $o(sk; r) \in \mathbb{N}$.*
- *Correctness: For every large enough $n \in \mathbb{N}$*

$$\Pr_{sk \leftarrow \mathsf{Gen}(1^n), r}[|\mathsf{Dec}(sk, m(sk,r)) - o(sk,r)| \le d(1^n)] \ge 1 - \alpha(n)$$

- *Security: For every oracle-aided* PPT Eve*, for every large enough $n \in \mathbb{N}$,*

$$\Pr_{sk \leftarrow \mathsf{Gen}(1^n), r}\left[\left|\mathsf{Eve}^{\mathsf{Enc}_{sk}}(1^n, m(sk,r)) - o(sk,r)\right| \le \gamma(n) \cdot d(1^n)\right] \le \beta(n),$$

  where Enc_{sk} *is the (randomized, no-input) function defined by* $\mathsf{Enc}_{sk}() = \mathsf{Enc}(sk)$*.*

We remark that such a weak CPA-secure encryption scheme can be amplified to a CPA-secure encryption scheme. This result may be known to follow using the techniques of [Hol06] but, as far as we know, such a result has not previously been stated. For sake of self containment, we present a direct and simple proof of the following (secret-key) analog of Lemma 2.10, and defer the proof to Appendix A.2:

Lemma 7.6 (CPA-secure encryption amplification)**.** *The following holds for every constants $c_1 > c_2$. Assume there exists an $(n^{-c_1}/2, 1 - 10n^{-c_2}, 2n^{c_1})$-weak CPA-secure encryption scheme. Then, there exists a CPA-secure encryption scheme.*

Theorem 7.2 next follows by the following two theorems.

Theorem 7.7. *Let $t(n)$ be a polynomial and $\Delta\colon \{1\}^* \to \mathbb{N}$ an efficiently computable function. Then if $\Delta\text{-}\mathsf{BBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$, CPA-secure encryption scheme exist.*

Proof sketch. The proof is almost identical to the proof of Algorithm 4.2, where we change the algorithm Gen (Algorithm 4.2) to output (r, k, ℓ, C, P) as the secret-key, and Enc, Dec gets are defined similarly. In the security prove, we give Eve oracle access to samples from P instead of the description of P as input (but assume that Eve gets r, k, ℓ as input). □

Theorem 7.8. *Assume there exists a CPA-secure encryption scheme. Then for any constant $\epsilon > 0$ and any $t(n) \ge n^{1+\epsilon}$ and any efficiently computable $\Delta\colon \{1\}^* \to \mathbb{N}$ such that $\Delta(1^n) \le 2^{n^{(1-\epsilon)}}$, $\Delta\text{-}\mathsf{BBLearn}|_{\mathsf{CS}^t} \notin \mathsf{ioFBPP}$.*

The proof of Theorem 7.8 is similar to the proof of Theorem 4.1, where we replace cPKE with the following defined circuit-Encryption.

Definition 7.9 (Circuit Encryption)**.** *A randomized, function* Gen' *is a $(\alpha(n), \beta(n))$-*circuit Encryption *if the following holds:*

- $\mathsf{Gen}(n)$ *outputs two circuits $pk\colon \{0,1\} \times \{0,1\}^{r(n)} \to \{0,1\}^{k(n)}$ and $sk\colon \{0,1\}^{k(n)} \to \{0,1\}$.*
- *Correctness: For every large enough $n \in \mathbb{N}$ and any $m \in \{0,1\}$,*

$$\Pr_{(pk,sk) \leftarrow \mathsf{Gen}(1^n), w \leftarrow \{0,1\}^{r(n)}}[sk(pk(m,w)) = m] \ge 1 - \alpha(n)$$

- *Security: For every oracle aided* PPT Eve*, for every large enough* $n \in \mathbb{N}$*,*

$$\Pr_{(pk,sk)\leftarrow \mathsf{Gen}(1^n), m \leftarrow \{0,1\}, w \leftarrow \{0,1\}^{r(n)}} \left[\mathsf{Eve}^{\mathcal{O}_{pk}}(1^n, pk(m,w)) = m\right] \leq 1/2 + \beta(n),$$

  where $\mathcal{O}_{pk}$ *is an oracle that samples from the distribution of* $pk(U_1, U_r)$*.*

Such a scheme is a circuit encryption *if it is* $(1/n^c, 1/n^c)$*-circuit encryption for every constant* $c \in \mathbb{N}$*.*

Proof sketch. The proof follows similarly to the proof of Theorem 4.1, where we first construct circuit encryption from CPA secure encryption (letting $sk'(e) = \mathsf{Dec}(sk, e)$ and $pk'(m, w) = \mathsf{Enc}(sk, m; w)$), and then replace the use of cPKE with the circuit encryption scheme. □

Theorem 7.2 follows directly by Theorems 7.7 and 7.8 and Proposition 7.4.

Proof of Theorem 7.2. Follows by Theorems 7.7 and 7.8 and Proposition 7.4. □

7.1 Bounded-Degree Black-Box Learning

Similarly to the white-box case, we can define black-box learning of bounded degree functions. For a degree d, let $\Delta - \mathsf{BBLearn}^d_q$ and $\Delta\text{-}\mathsf{BBLearn}^d_q|_{\mathsf{CS}^t}$ be defined similarly to $\Delta - \mathsf{WBLearn}^d_q$ and $\Delta\text{-}\mathsf{WBLearn}^d_q|_{\mathsf{CS}^t}$ in the black-box setting.

Similarly to Lemma 6.7 and using the same reduction, we get the following result.

Lemma 7.10. *For every constant* $d \in \mathbb{N}$*, and functions* $t\colon \mathbb{N} \to \mathbb{N}$*,* $q, \Delta\colon \{1\}^* \to \mathbb{N}$ *there exist* $t'\colon \mathbb{N} \to \mathbb{N}$ *and* $q', \Delta'\colon \{1\}^* \to \mathbb{N}$ *such that*

$$\Delta\text{-}\mathsf{BBLearn}^d_q|_{\mathsf{CS}^t} \leq_p \Delta'\text{-}\mathsf{BBLearn}^1_{q'}|_{\mathsf{CS}^{t'}},$$

and

$$\Delta\text{-}\mathsf{BBLearn}^d_q \leq_p \Delta'\text{-}\mathsf{BBLearn}^1_{q'}.$$

References

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. "Public-key cryptography from different assumptions". In: *Proceedings of the forty-second ACM symposium on Theory of computing.* 2010, pp. 171–180 (cit. on p. 2).

[AD97] Miklós Ajtai and Cynthia Dwork. "A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence". In: *stoc29.* See also ECCC TR96-065. 1997, pp. 284–293 (cit. on p. 2).

[AF09] Luis Antunes and Lance Fortnow. "Worst-case running times for average-case algorithms". In: *2009 24th Annual IEEE Conference on Computational Complexity.* IEEE. 2009, pp. 298–303 (cit. on p. 4).

[AFVMV06] Luis Antunes, Lance Fortnow, Dieter Van Melkebeek, and N Variyam Vinodchandran. "Computational depth: concept and applications". In: *Theoretical Computer Science* 354.3 (2006), pp. 391–404 (cit. on pp. 3, 4, 15).

[Ale03] Michael Alekhnovich. "More on average case vs approximation complexity". In: *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* IEEE. 2003, pp. 298–307 (cit. on p. 2).

[BCNHR22] Andrej Bogdanov, Miguel Cueto Noval, Charlotte Hoffmann, and Alon Rosen. "Public-Key Encryption from Homogeneous CLWE". In: *Theory of Cryptography: 20th International Conference, TCC 2022, Chicago, IL, USA, November 7–10, 2022, Proceedings, Part II.* Springer. 2022, pp. 565–592 (cit. on p. 2).

[BFKL93] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J Lipton. "Cryptographic primitives based on hard learning problems". In: *Annual International Cryptology Conference.* Springer. 1993, pp. 278–291 (cit. on pp. 3, 6, 29).

[BLMP23] Marshall Ball, Yanyi Liu, Noam Mazor, and Rafael Pass. "Kolmogorov Comes to Cryptomania: On Interactive Kolmogorov Complexity and Key-Agreement". In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS).* IEEE. 2023, pp. 458–483 (cit. on pp. 4, 5, 8–10).

[BLPRS13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. "Classical hardness of learning with errors". In: *Proceedings of the forty-fifth annual ACM symposium on Theory of computing.* 2013, pp. 575–584 (cit. on p. 5).

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. "Efficient fully homomorphic encryption from (standard) LWE". In: *Journal of the ACM* 43.2 (2014), pp. 831–871 (cit. on p. 7).

[Cha69] Gregory J. Chaitin. "On the Simplicity and Speed of Programs for Computing Infinite Sets of Natural Numbers". In: *J. ACM* 16.3 (1969), pp. 407–422 (cit. on p. 3).

[DH76] Whitfield Diffie and Martin E. Hellman. "New Directions in Cryptography". In: *IEEE Transactions on Information Theory* (1976), pp. 644–654 (cit. on p. 2).

[ElG84] Taher ElGamal. "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms". In: *Annual International Cryptology Conference (CRYPTO).* 1984, pp. 10–18 (cit. on p. 2).

[Gen09] Craig Gentry. "Fully homomorphic encryption using ideal lattices". In: *Proceedings of the forty-first annual ACM symposium on Theory of computing.* 2009, pp. 169–178 (cit. on p. 7).

[GL89] Oded Goldreich and Leonid A. Levin. "A Hard-Core Predicate for all One-Way Functions". In: *Proceedings of the twenty-first annual ACM symposium on Theory of computing (STOC).* 1989, pp. 25–32 (cit. on p. 7).

[Gol04] Oded Goldreich. *Foundations of Cryptography – VOLUME 2: Basic Applications.* Cambridge University Press, 2004 (cit. on p. 29).

[Har83] J. Hartmanis. "Generalized Kolmogorov complexity and the structure of feasible computations". In: *24th Annual Symposium on Foundations of Computer Science (sfcs 1983).* 1983, pp. 439–445. DOI: 10.1109/SFCS.1983.21 (cit. on p. 3).

[HKNRR05] Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. “On robust combiners for oblivious transfer and other primitives”. In: *Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005. Proceedings 24*. Springer. 2005, pp. 96–113 (cit. on pp. 7, 10).

[HN23] Shuichi Hirahara and Mikito Nanashima. “Learning in Pessiland via Inductive Inference”. In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2023, pp. 447–457 (cit. on pp. 3–6).

[Hol06] Thomas Holenstein. “Strengthening key agreement using hard-core sets”. PhD thesis. ETH Zurich, 2006 (cit. on pp. 7, 13, 30).

[IL90] Russell Impagliazzo and Leonid A. Levin. “No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random”. In: *focs31*. 1990, pp. 812–821 (cit. on p. 3).

[Ko86] Ker-I Ko. “On the Notion of Infinite Pseudorandom Sequences”. In: *Theor. Comput. Sci.* 48.3 (1986), pp. 9–33. DOI: `10.1016/0304-3975(86)90081-2`. URL: `https://doi.org/10.1016/0304-3975(86)90081-2` (cit. on p. 3).

[Kol68] A. N. Kolmogorov. “Three approaches to the quantitative definition of information”. In: *International Journal of Computer Mathematics* 2.1-4 (1968), pp. 157–168 (cit. on p. 3).

[KV94] Michael Kearns and Leslie Valiant. “Cryptographic limitations on learning boolean formulae and finite automata”. In: *Journal of the ACM (JACM)* 41.1 (1994), pp. 67–95 (cit. on pp. 3, 6, 29).

[LP20] Yanyi Liu and Rafael Pass. “On one-way functions and Kolmogorov complexity”. In: *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2020, pp. 1243–1254 (cit. on pp. 4, 10).

[LP23] Yanyi Liu and Rafael Pass. “On One-way Functions and the Worst-case Hardness of Time-Bounded Kolmogorov Complexity”. In: *Cryptology ePrint Archive* (2023) (cit. on pp. 4, 5, 8, 9).

[McE78] Robert J McEliece. “A public-key cryptosystem based on algebraic”. In: *Coding Thv* 4244 (1978), pp. 114–116 (cit. on p. 2).

[Pei09] Chris Peikert. “Public-key cryptosystems from the worst-case shortest vector problem”. In: *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 2009, pp. 333–342 (cit. on p. 5).

[Rab79] Michael O Rabin. *Digitalized signatures and public-key functions as intractable as factorization*. Tech. rep. Massachusetts Inst of Tech Cambridge Lab for Computer Science, 1979 (cit. on p. 2).

[Reg09] Oded Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: *Journal of the ACM (JACM)* 56.6 (2009), pp. 1–40 (cit. on pp. 2, 4, 5, 7, 26).

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. “A method for obtaining digital signatures and public-key cryptosystems”. In: *Communications of the ACM* 21.2 (1978), pp. 120–126 (cit. on p. 2).

[Sho99] Peter W Shor. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”. In: *SIAM review* 41.2 (1999), pp. 303–332 (cit. on p. 2).

[Sip83] Michael Sipser. “A Complexity Theoretic Approach to Randomness”. In: 1983, pp. 330–335 (cit. on p. 3).

[Sol64] R.J. Solomonoff. “A formal theory of inductive inference. Part I”. In: *Information and Control* 7.1 (1964), pp. 1 –22. ISSN: 0019-9958. DOI: https://doi.org/10.1016/S0019-9958(64)90223-2 (cit. on p. 3).

[Val84] Leslie G Valiant. “A theory of the learnable”. In: *Communications of the ACM* 27.11 (1984), pp. 1134–1142 (cit. on pp. 2, 6, 28).

[Yao82] Andrew C. Yao. “Theory and Applications of Trapdoor Functions”. In: *Annual Symposium on Foundations of Computer Science (FOCS).* 1982, pp. 80–91 (cit. on p. 12).

A Missing proofs

A.1 Proving Lemma 2.1

.

To prove Lemma 2.1, we use the following weak version of GL.

Lemma A.1. *There exists a* PPT *oracle-aided algorithm* Dec *such that the following holds. Let* $n \in N$ *be a number,* $x \in \{0,1\}^n$*, and and let* Pred *be an algorithm such that*

$$\Pr_{r \leftarrow \{0,1\}^n}[\mathsf{Pred}(r) = \mathrm{GL}(x,r)] > 3/4 + 0.01,$$

where $\mathrm{GL}(x,r) := \langle x, r \rangle$ *is the Goldreich-Levin predicate. Then* $\Pr\left[\mathsf{Dec}^{\mathsf{Pred}}(1^n) = x\right] = 1 - \mathrm{neg}(n)$.

Proof of Lemma A.1. We use Pred to decode each bit of x separately. For every i, let e_i be the vector that has 1 in the i-th entry, and 0's everywhere else. Observe that, for a uniformly chosen $R \leftarrow \{0,1\}^n$,

$$\Pr[\mathsf{Pred}(R) = \mathrm{GL}(x,R) \wedge \mathsf{Pred}(R \oplus e_i) = \mathrm{GL}(x, R \oplus e_i)] \geq 1/2 + 0.01.$$

Thus,

$$\Pr[\mathsf{Pred}(R) \oplus \mathsf{Pred}(R \oplus e_i) = \mathrm{GL}(x,R) \oplus \mathrm{GL}(x, R \oplus e_i)] \geq 1/2 + 0.01.$$

By linearity of the inner product we get that,

$$\Pr[\mathsf{Pred}(R) \oplus \mathsf{Pred}(R \oplus e_i) = x_i] \geq 1/2 + 0.01.$$

Let Dec be the algorithm that for every i, computes $\mathsf{Pred}(R) \oplus \mathsf{Pred}(R \oplus e_i)$ for n random values of R, and let x'_i to be the majority of the outputs. Then, Dec outputs $x' = x'_1, \ldots, x'_n$. By Chernoff bound, x'_i is equal to x_i with all but negligible probability. By the union bound, the above is true for all i's simultaneously with all but negligible probability, as we wanted to show. ☐

We are now ready to prove Lemma 2.1.

Proof of Lemma 2.1. Let A be the algorithm that given $1^n, X$ and an oracle to E, executes $\mathsf{Dec}^{\mathsf{Pred}}(1^n)$ where Dec is the algorithm promised by Lemma A.1, and $\mathsf{Pred}(r) = \mathsf{E}(X, r)$. It is enough to show that $\Pr_X[\Pr_R[\mathsf{E}(X, R) = \langle X, R\rangle] \geq 3/4 + 0.01] \geq 1 - 8\epsilon$.

Assume toward a contradiction that $\Pr_X[\Pr_R[\mathsf{E}(X, R) = \langle X, R\rangle] \geq 3/4 + 0.01] < 1 - 8\epsilon$. Then,

$$\Pr_{X,R}[\mathsf{E}(X, R) = \langle X, R\rangle] < (1 - 8\epsilon) \cdot 1 + 8\epsilon \cdot (3/4 + 0.01) < 1 - \epsilon,$$

which is a contradiction to the assumption. □

A.2 Proving Lemma 7.6

Lemma A.2 (CPA-secure encryption amplification, restated)**.** *The following holds for every constants $c_1 > c_2$. Assume there exists an $(n^{-c_1}/2, 1 - 10n^{-c_2}, 2n^{c_1})$-weak CPA-secure encryption scheme. Then, there exists a CPA-secure encryption scheme.*

Proof. We prove a stronger claim - that $(n^{-c_1}, 1 - 5n^{-c_1}, 1)$-weak CPA-secure encryption scheme implies the existence of a one-way function. This implies the lemma using Proposition 7.4. Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be such scheme. For a large enough polynomial k to be chosen later, we define a weak one-way function f as follows:

$$f(1^n, r_{\mathsf{Gen}}, r_1, \ldots, r_k) = 1^n, \mathsf{Enc}(\mathsf{Gen}(1^n; r_{\mathsf{Gen}}); r_1), \ldots, \mathsf{Enc}(\mathsf{Gen}(1^n; r_{\mathsf{Gen}}); r_k), \sum_i b_i,$$

where $b_i \in \{0, 1\}$ is equal to one if $|\mathsf{Dec}(\mathsf{Gen}(1^n; r_{\mathsf{Gen}}), m_i) - o_i| \leq d(1^n)$ for $(m_i, o_i) = \mathsf{Enc}(\mathsf{Gen}(1^n; r_{\mathsf{Gen}}); r_i)$. We next prove that f is indeed n^{-c_1}-weak one-way function. Using Theorem 2.6, we get that one-way function exists.

Assume that there exists an efficient algorithm A that inverts f with probability at least $1-n^{-c_1}$ for infinitely many n's. We claim that we can use A to break the security of the weak CPA-secure encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Indeed, let Eve be the algorithm that on input $1^n, m$ and oracle Enc_{sk}, samples k pairs $(m_1, o_1), \ldots, (m_k, o_k)$ using the oracle. Let $r_1, \ldots, r_k$ be the randomness used by the oracle (so that $(m_i, o_i) = \mathsf{Enc}(sk, r_i)$). It then uses A to invert f on $1^n, (m_1, o_1), \ldots, (m_k, o_k), v$ for every value of $v \in [k]$. Let $1^n, r'_{\mathsf{Gen}}, r'_1, \ldots, r'_k$ be the output of A on the largest value of v for which A successfully found a preimage. Finally, Eve computes a secret-key sk' using $\mathsf{Gen}(r'_{\mathsf{Gen}})$ and outputs $o' = \mathsf{Dec}(sk', m)$. Let $b'_i \in \{0, 1\}$ be 1 if $|\mathsf{Dec}(sk', m_i) - o_i| \leq d(1^n)$, or 0 otherwise.

We first notice that on a random key $sk \leftarrow \mathsf{Gen}(1^n)$, and for $v^* = \sum b_i$, the distribution of Eve's query $(1^n, (m_1, o_1), \ldots, (m_k, o_k), v^*)$ to A is the exact distribution of the image of f on a random input. Thus, with probability at least $1 - n^{-c_1}$, A outputs $r'_{\mathsf{Gen}}, r'_1, \ldots, r'_k$ such that $\mathsf{Enc}(sk'; r'_i) = (m_i, o_i)$ for every $i \in [k]$ for $sk' = \mathsf{Gen}(1^n; r'_{\mathsf{Gen}})$, and where $\sum b'_i \geq v^*$.

In the following we show that with probability at least $1 - n^{-c_1}$ over sk and the randomness of the oracle calls $r_1, \ldots, r_k$, it follows that any such $r'_{\mathsf{Gen}}, r'_1, \ldots, r'_k$ with $\sum b'_i \geq v^*$,

$$\Pr_{(m,o)\leftarrow\mathsf{Enc}(sk)}\left[\left|\mathsf{Dec}(sk', m) - o\right| \leq d(1^n)\right] \geq v^*/k - n^{-c_1}. \tag{6}$$

Using the union bound, we will then get that with probability at least $1 - 2n^{-c_1}$ Eve outputs a key sk' for which Equation (6) holds. The proof then follows by the observation that the expected value of v^*/k is at least $1 - n^{-c_1}$.

To see that Equation (6) holds with high probability, fix sk and sk' such that Equation (6) does not hold. We will show that with probability at least $1 - n^{-c_1}/2^{|sk'|}$ (over the randomness of the oracle calls $r_1, \ldots, r_k$), any $r'_{\mathsf{Gen}}, r'_1, \ldots, r'_k$ such that $sk' = \mathsf{Gen}(1^n; r'_{\mathsf{Gen}})$ is not a valid pre-image of $(m_1, o_1), \ldots, (m_k, o_k), v$ for $v \geq v^*$. Equation (6) then follows using the union bound on all possible keys sk'.

By our choice of sk and sk', it holds that

$$\Pr_{(m,o)\leftarrow\mathsf{Enc}(sk)}\left[\left|\mathsf{Dec}(sk', m) - o\right| \leq d(1^n)\right] < v^*/k - n^{-c_1}. \tag{7}$$

Using Fact 2.14, for $k \geq n^{2c_1}(|sk'| + c_1 \log n + 1)$, we get that

$$\Pr_{(m_1,o_1),\ldots,(m_k,o_k)\leftarrow\mathsf{Enc}_{sk}}\left[\left|\left\{i\colon \left|\mathsf{Dec}(sk', m_i) - o_i\right| \leq d(1^n)\right\}\right| \geq v^*\right] \leq n^{-c_1}/2^{|sk'|}.$$

The proof follows as $\sum_i b'_i = |\{i\colon |\mathsf{Dec}(sk', m_i) - o_i| \leq d(1^n)\}|$. □

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il