Private PAC Learning May be Harder than Online Learning

Mark Bun MBUN@BU.EDU
Department of Computer Science, Boston University

Aloni Cohen ALONI@G.UCHICAGO.EDU
Department of Computer Science and Data Science Institute, University of Chicago

Rathin Desai RATHIN@BU.EDU
Department of Computer Science, Boston University

Editors: Claire Vernade and Daniel Hsu

Abstract

We continue the study of the computational complexity of differentially private PAC learning and how it is situated within the foundations of machine learning. A recent line of work uncovered a qualitative equivalence between the private PAC model and Littlestone's mistake-bounded model of online learning, in particular, showing that any concept class of Littlestone dimension d can be privately PAC learned using $\mathrm{poly}(d)$ samples. This raises the natural question of whether there might be a generic conversion from online learners to private PAC learners that also preserves computational efficiency.

We give a negative answer to this question under reasonable cryptographic assumptions (roughly, those from which it is possible to build indistinguishability obfuscation for all circuits). We exhibit a concept class that admits an online learner running in polynomial time with a polynomial mistake bound, but for which there is no computationally-efficient differentially private PAC learner. Our construction and analysis strengthens and generalizes that of Bun and Zhandry (TCC 2016-A), who established such a separation between private and non-private PAC learner.

1. Introduction

Differential privacy Dwork et al. (2006b) is a formal guarantee of individual-level privacy for the analysis of statistical datasets. Algorithmic research on differential privacy has revealed it to be a central concept to theoretical computer science and machine learning, supplementing the original motivation with deep connections to diverse topics including mechanism design (McSherry and Talwar, 2007; Nissim et al., 2012), cryptography (Beimel et al., 2018), quantum computing Aaronson and Rothblum (2019), generalization in the face of adaptive data analysis (Dwork et al., 2015; Hardt and Ullman, 2014), and replicability in learning (Bun et al., 2023).

To investigate the connections between privacy and machine learning in a simple and abstract setting, Kasiviswanathan et al. (2011) introduced the *differentially private PAC model* for binary clsassification. Numerous papers (Beimel et al., 2014; Bun et al., 2015; Feldman and Xiao, 2015; Beimel et al., 2016; Bun and Zhandry, 2016; Beimel et al., 2019; Alon et al., 2019; Kaplan et al., 2019; Bun, 2020; Sadigurschi and Stemmer, 2021) have since explored the capabilities and limitations (both statistical and computational) of algorithms in this model. A major motivating question in this area is:

Question 1 *When do sample-efficient private PAC learners exist, and if so, when can they be made computationally efficient?*

Early work (Blum et al., 2005; Kasiviswanathan et al., 2011) gave us some important partial answers. On the statistical side, they showed that every finite concept class $\mathcal{F}$ can be privately learned with $O(\log |\mathcal{F}|)$ samples, albeit by an algorithm taking exponential time in general. Computationally, they showed that both "natural" paradigms for polynomial-time non-private PAC learning have differentially private analogs: learners in Kearns' statistical query (SQ) model (Kearns, 1998) and the learner for parities based on Gaussian elimination. Much of the subsequent work on private PAC learning has focused on improving the sample- and computational-efficiency of algorithms for fundamental concept classes, including points, thresholds, conjunctions, halfspaces, and geometric concepts. Meanwhile, Bun and Zhandry (2016) gave an example of a concept class that has a polynomial-time non-private PAC learner, but no computationally-efficient private PAC learner (under strong, but reasonable cryptographic assumptions).

While this work has led to the development of important algorithmic tools and to fascinating connections to other areas of theoretical computer science, a general answer to Question 1 continues to elude us. Some tantalizing progress was made in a recent line of work connecting private PAC learning to the completely different model of mistake-bounded online learning. This connection is summarized as follows.

Theorem 2 (Alon et al. (2022); Ghazi et al. (2021)) *Let $\mathcal{F}$ be a concept class with* Littlestone dimension $d = L(\mathcal{F})$. *Then $\tilde{O}(d^6)$ samples are sufficient to privately learn $\mathcal{F}$ and $\Omega(\log^* d)$ samples are necessary.*

Here, the Littlestone dimension of a class $\mathcal{F}$ measures the best possible mistake bound in Littlestone's model of online learning. Thus, at least qualitatively, online learnability characterizes private PAC learnability. In particular, the "are sufficient" direction of Theorem 2 can be viewed as an elaborate online-to-batch conversion, transforming an online learner into a private PAC learner with only a polynomial blowup from mistake bound to private sample cost. Unfortunately, while it is statistically efficient, the algorithm achieving this (Ghazi et al., 2021) does not preserve computational efficiency in general. Among other steps, it entails computing $L(\mathcal{F}')$ for various subclasses derived from $\mathcal{F}$, which is believed to be computationally hard in general (Schaefer, 1999; Frances and Litman, 1998; Manurangsi and Rubinstein, 2017; Manurangsi, 2023).

One might nonetheless hope for a different transformation that maintains both statistical and computational efficiency. Indeed, when one's goal is to convert an online learner to a *non-private* PAC learner, the transformation is simple and clearly efficient – just present random examples to the online learner and output its eventual state as a classifier (Littlestone, 1989) (see also the variant due to Kearns et al. (1987); Angluin (1988), which is a standard topic in graduate classes on learning theory). Moreover, many key techniques from online learning have found differentially private analogs incurring minimal overhead, including follow-the-regularized-leader (Agarwal and Singh, 2017) and learning from experts (Asi et al., 2023).

Our main result shows that such a generic transformation is unlikely to exist.

Theorem 3 (Informal) *Under (strong, but reasonable) cryptographic assumptions, there is a concept class that is online learnable by a polynomial-time algorithm with a polynomial mistake bound, but not privately PAC learnable in polynomial-time.*

A formal description of this result appears as Theorem 37 in Section 7, along with further discussion of the cryptographic assumptions. Roughly, our assumptions include functional encryption

for all poly-size circuits (an assumption comparable to indistinguishability obfuscation, and recently shown to be obtainable from reasonable assumptions (Jain et al., 2021)), a circuit lower bound for deterministic exponential time, and perfectly sound non-interactive zero knowledge proofs.

Note that owing to the existence of efficient non-private online-to-batch conversions mentioned above, this result strengthens the separation between private and non-private learning from Bun and Zhandry (2016). It also complements a pair of papers studying the possibility of a computationally efficient transformation in the opposite direction. Namely, Gonen et al. (2019) gave conditions under which pure private learners can be efficiently converted into online learners, while Bun (2020) gave a counterexample of a class that is efficiently privately PAC learnable, but not efficiently online learnable. Our result answers an open question from Bun (2020) and, together with that result, shows that polynomial-time online learnability and polynomial-time private PAC learnability are technically incomparable.

1.1. Techniques

Our construction of a concept class that separates online learning from private PAC learning builds on the construction from Bun and Zhandry (2016), so let us briefly review it here. The starting point of their construction was the concept class of one-dimensional threshold functions Thr over a domain of the form $[N] = \{1, \ldots, N\}$. Each function $f_t \in \mathsf{Thr}$ is itself parameterized by a value $t \in [N]$, and takes the value $f_t(x) = 1$ if $x < t$, and $f_t(x) = 0$ otherwise. Threshold functions are easy to PAC learn non-privately. Given a sampled dataset $((x_i, y_i))_{i=1}^n$, a non-private algorithm can simply output f_{x_i} where x_i is the largest value for which the label $y_i = 1$. A standard concentration argument shows that this generalizes to the underlying population from which the sample is drawn as long as n is larger that some constant that is independent of the domain size N.

On the other hand, this algorithm badly fails to be differentially private, as it exposes the sample x_i. In fact, every differentially private learner for this class requires $\Omega(\log^* N)$ samples (Bun et al., 2015; Alon et al., 2022). However, this lower bound is too small to give a computational separation, so the idea in Bun and Zhandry (2016) was to use cryptography to preserve the problem's non-private learnability, while making it much harder to achieve differential privacy. Specifically, they defined a class EncThr by first encrypting each example x_i under an *order-revealing encryption* scheme. Such a scheme allows for ciphertexts to be compared in a manner consistent with the underlying plaintexts, but for nothing else to be revealed besides their order. The ability to make these comparisons is enough for the simple non-private "largest positive example" algorithm to go through. Meanwhile, security of the order-revealing encryption scheme intuitively guarantees that comparisons to the specific ciphertexts appearing in the sample are *all* that efficient learners can do, and hence they cannot be differentailly private.

Turning now to our goal of separating online from private PAC learning, we observe that while the class Thr is efficiently online learnable with mistake bound $\log N$ via binary search (see Section 2.3), the encrypted class EncThr is *not*. Intuitively, order-revealing encryption does not reveal enough information to enable a learner to make efficient use of mistakes, as in binary search. More precisely, suppose the target concept is the (encrypted version) of the middle threshold $f_{N/2}$. Consider an adversary who selects examples by randomly choosing either the smallest positive example not presented so far, or the largest negative example. A computationally-bounded adversary who can only compare these examples to those seen so far cannot distinguish between these cases, and is

hence liable to make super-polynomially (depending on the security of the underlying ORE) many mistakes.

To obtain our main result, we modify the class EncThr to make it efficiently online learnable, while keeping it hard enough to carry out a lower bound against differentially private algorithms. Achieving both goals simultaneously turns out to be a delicate task, and it helps to think about each in more abstract terms. To this end, let L be a *function-revealing encryption* scheme, which (generalizing ORE) enables the revelation ("leakage") of specific structured relationships between plaintexts, but nothing else. Correspondingly, let $\mathsf{LEncThr}$ be the class of one-dimensional thresholds with examples encrypted under L. Given the inadequacy of ORE for efficient online learning, we will think of L as revealing not only the order of plaintexts, but some limited information about the distances between plaintexts as well.

First, we identify sufficient conditions on L to enable the construction of an efficient online learner. Inspired by binary search, we'd like to reveal enough distance information so that every mistake made by an online learner can rule out a constant fraction of the remaining space of consistent concepts. Thinking of distances on a logarithmic scale, we articulate this condition as a "bisection property" of the leakage function (Definition 15) and analyze our analog of binary search in Section 4.

Second, we identify sufficient conditions on L to enable a lower bound against differentially private PAC learners. To do so, we simplify the lower bound argument from Bun and Zhandry (2016), in particular, bypassing their intermediate abstraction of an "example re-identification scheme" and directly showing how to use an accurate, efficient, differentially private learner for EncThr to construct an adversary violating the security of the underlying ORE scheme. This simplified argument goes roughly as follows. Consider running a PAC learner on n uniformly random encryptions labeled by the middle threshold $f_{N/2}$. Accuracy of the learner, together with an averaging argument, implies that for some index i, it can distinguish random encryptions of messages from $[x_{i-1}, x_i)$ from random encryptions from $[x_i, x_{i+1})$ with advantage $\Omega(1/n)$. Now, differential privacy implies that this noticeable distinguishing advantage remains even when the learner is *not* given example x_i, violating the security of the ORE scheme.

This simplified argument makes it easy to use group differential privacy to reason about what happens when not just one, but a small number of examples are removed. In particular, an inverse polynomial distinguishing advantage remains even if we withhold $O(\log n)$ points from the learner. This extra flexibility turns out to be critical in helping us construct pairs of challenge messages in L security games, which not only need to have the same relative order, but respect the stronger constraints imposed by L-leakage. We describe the precise condition we need as "log-invariance" (Definition 16) and show in Section 5 that any L with this condition gives rise to a private PAC learning lower bound.

Our final task is to exhibit a function-revealing encryption scheme L that actually has both the bisection and log-invariance properties. Identifying a leakage function that works turns out to be quite tricky. Our starting point is to reveal the floor of the logarithm of the distance between plaintexts, but this gives too much information for a lower bound to hold. Instead, we reduce this to just comparison information between floor log distances. That is, for any triple of plaintexts $m_0 \leqslant m_1 \leqslant m_2$, we leak whether the floor log distance between m_0 and m_1 is less than that between m_1 and m_2.

2. Preliminaries

2.1. PAC Learning

For each $d \in \mathbb{N}$, let X_d be an instance space (such as $\{0,1\}^d$), where the parameter d represents the size of the elements in X_d. Let $\mathcal{F}_d$ be a set of boolean functions $\{f : X_d \to \{0,1\}\}$. The sequence $(X_1, \mathcal{F}_1), (X_2, \mathcal{F}_2), \ldots$ represents an infinite sequence of learning problems defined over instance spaces of increasing dimension. We will generally suppress the parameter d, and refer to the problem of learning $\mathcal{F}$ as the problem of learning $\mathcal{F}_d$ for every d.

A learner L is given examples sampled from an unknown probability distribution $\mathcal{D}$ over X, where the examples are labeled according to an unknown *target concept* $f \in \mathcal{F}$. The learner must select a hypothesis h from a hypothesis class $\mathcal{H}$ that approximates the target concept with respect to the distribution $\mathcal{D}$. We now define the notion of PAC ("Probably Approximately Correct") learning formally.

Definition 4 *The generalization error of a hypothesis $h : X \to \{0,1\}$ (with respect to a target concept f and distribution $\mathcal{D}$) is defined by $error_{\mathcal{D}}(f,h) = \mathbf{Pr}_{x\sim\mathcal{D}}[h(x) \neq f(x)]$. If $error_{\mathcal{D}}(f,h) \leqslant \alpha$ we say that h is an α-good hypothesis for f on $\mathcal{D}$.*

Definition 5 (PAC Learning, Valiant (1984)) *Let $\mathcal{H}$ be a class of boolean functions over X. An algorithm $L : (X \times \{0,1\})^n \to \mathcal{H}$ is an* (α, β)-accurate PAC learner *for the concept class $\mathcal{F}$ using hypothesis class $\mathcal{H}$ with sample complexity n if for all target concepts $f \in \mathcal{F}$ and all distributions $\mathcal{D}$ on X, given as input n samples $S = ((x_i, f(x_i)), \ldots, (x_n, f(x_n)))$, where each x_i is drawn i.i.d. from $\mathcal{D}$, algorithm L outputs a hypothesis $h \in \mathcal{H}$ satisfying $\mathbf{Pr}[error_{\mathcal{D}}(f,h) \leqslant \alpha] \geqslant 1-\beta$. The probability here is taken over the random choice of the examples in S and the coin tosses of the learner L.*

We are primarily interested in computationally efficient PAC learners, defined as follows.

Definition 6 (Efficient PAC Learning) *A PAC learner L for concept class $\mathcal{F}$ is* efficient *if it runs in time polynomial in the size parameter d, the representation size of the target concept f, and the accuracy parameters $1/\alpha$ and $1/\beta$.*

Note that a necessary (but not sufficient) condition for L to be efficient is that its sample complexity n is polynomial in the learning parameters.

2.2. Differential Privacy

We now define differential privacy and the differentially private PAC model.

Definition 7 *(k-neighboring datasets) Let $S, S' \in Z^n$ for some data domain Z. We say that S and S' are k-neighboring datasets if they differ in exactly k entries. If $k = 1$, we simply say they are neighboring.*

Definition 8 (Differential Privacy, Dwork et al. (2006b,a)) *An algorithm $M : Z^n \to R$ is* (ε, δ)-differentially private *if for all sets $T \subseteq \mathcal{H}$, and neighboring datasets $S, S' \in Z^n$,*

$$\mathbf{Pr}[M(S) \in T] \leqslant e^{\varepsilon} \mathbf{Pr}[M(S') \in T] + \delta.$$

Specializing this definition to the case where $Z = X \times \{0,1\}$ and $R = \mathcal{H}$ is a class of hypotheses, we obtain the differentially private PAC model of Kasiviswanathan et al. (2011).

We now state some simple tools for designing and analyzing differentially private algorithms.

Lemma 9 (Basic Composition Dwork et al. (2006b); Dwork and Lei (2009)) *Let $M_1 : Z^n \to R_1$ be (ε, δ)-differentially private. Let $M_2 : Z^n \times R_1 \to R_2$ be (ε, δ)-differentially private for every fixed value of its second argument. Then the composed algorithm $M : Z^n \to R_2$ defined by $M(S) = M_2(S, M_1(S))$ is $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-differentially private.*

Lemma 10 (Group Privacy) *Let $M : Z^n \to R$ be (ε, δ)-differentially private. Let S and S' be k-neighboring databases. Then for all sets of outcomes T,*

$$\mathbf{Pr}[M(S) \in T] \leqslant e^{k\varepsilon} \cdot \mathbf{Pr}[M(S) \in T] + \frac{e^{k\varepsilon} - 1}{e^{\varepsilon} - 1} \cdot \delta.$$

Lemma 11 (Post-Processing) *Let $M : X^n \to R$ be (ε, δ)-differentially private and let $f : R \to R'$ be an arbitrary randomized function. Then $f \circ M : X^n \to R'$ is (ε, δ)-differentially private.*

2.3. Online Learning, Halving, and Thresholds

We review Littlestone's model of mistake-bounded learning (Littlestone, 1987). It is defined as a two-player game between a learner and an adversary. Let $\mathcal{F}$ be a concept class. Prior to the start of the game, the adversary fixes a concept $f \in \mathcal{F}$. Let $|f|$ represent the description size of the concept and d be the dimension of the instance space. The learning proceeds in rounds. In each round i,

1. The adversary selects an $x_i \in \{0,1\}^d$ and reveals it to the learner.
2. The learner predicts a label $\hat{y}_i \in \{0,1\}$.
3. The adversary reveals the correct label $y_i = f(x_i)$.

A learner makes a mistake every time $\hat{y}_i \neq f(x_i)$. The goal of the learning algorithm is to minimize the number of mistakes it makes in the game. A learning algorithm learns $f \in \mathcal{F}$ with mistake bound M if for every target concept and adversary strategy, the total number of mistakes that the learner makes is at most M. We say that an online learner efficiently learns $\mathcal{F}$ if for every $f \in \mathcal{F}$ it has a mistake bound of $\mathrm{poly}(d, |f|)$ and runs in time $\mathrm{poly}(d, |f|)$ in every round.

A basic algorithm in this setting is the halving algorithm. Halving guarantees a mistake bound of $\log(|\mathcal{F}|)$ and can be made computationally efficient in certain structured cases. One such case is for the simple class of thresholds, which we describe below and study the halving algorithm for.

On data domain $X_d = [2^d]$, the class of thresholds $\mathsf{Thr}_d = \{f_t : X_d \to \{0,1\}\}$ over domain X_d is defined as follows. For each $t \in [2^d]$, define

$$f_t(x) = \begin{cases} 1 & \text{if } x < t \\ 0 & \text{otherwise.} \end{cases}$$

The halving algorithm for learning thresholds is described in Algorithm 1 below.

Whenever the online learner makes a mistake, the set of remaining candidate thresholds is reduced in size by a factor of 2. Since the size of hypothesis space at the start of the game is 2^d, the algorithm terminates after at most d mistakes. We will use a variant of this algorithm to efficiently online learn the concept class we construct later.

Algorithm 1 Halving

Initialize: $t \leftarrow 2^{d-1}, t_+ \leftarrow 1, t_- \leftarrow 2^d$
Input: Stream of $x_i \in [2^d]$ chosen in rounds $i = 1, 2, \ldots$ by an adversary, followed by labels y_i
for $i = 1, 2, \ldots$ **do**
  Set $t = \lfloor \frac{t_- + t_+}{2} \rfloor$
  Predict $\hat{y}_i = h_t(x_i)$ on input x_i
  if $\hat{y}_i = 1 \neq y_i$ **then**
    Update $t_- \leftarrow x_i$
  end
  if $\hat{y}_i = 0 \neq y_i$ **then**
    Update $t_+ \leftarrow x_i$
  end
end

3. Concept Class and its Learnability

3.1. Computational Separation between PAC and Private PAC learning

Bun and Zhandry (2016) proved a computational separation between PAC and Private PAC learning by defining a concept class called EncThr. EncThr intuitively captures the captures the class of threshold functions where examples are encrypted under an Order Revealing Encryption (ORE) scheme. An ORE scheme is defined by four algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Comp})$. We now describe the functionalities of the algorithms.

- $\mathsf{Gen}(1^\lambda, 1^d)$ is a randomized procedure that takes as inputs a security parameter λ and plaintext length d, and outputs a secret encryption/decryption key sk and public parameters params.
- $\mathsf{Enc}(\mathsf{sk}, m)$ is a potentially randomized procedure that takes as input a secret key sk and a message $m \in \{0,1\}^d$, and outputs a ciphertext c.
- $\mathsf{Dec}(\mathsf{sk}, c)$ is a deterministic procedure that takes as input a secret key sk and a ciphertext c, and outputs a plaintext message $m \in \{0,1\}^d$ or a failure symbol $\perp$.
- $\mathsf{Comp}(\mathsf{params}, c_0, c_1)$ is a deterministic procedure that "compares" two ciphertexts, outputting either ">", "<", "=", or $\perp$.

Each concept in the class EncThr is parameterized by a string r that represents the coin tosses of the algorithm Gen and by a threshold $t \in [N]$ for $N = 2^d$, where d represents the length of the plaintext. Let $(\mathsf{sk}^r, \mathsf{params}^r)$ be the secret key and the public parameters output by $\mathsf{Gen}(1^\lambda, 1^d)$ when run on the sequence of coin tosses r. Formally, a concept in EncThr parameterized by t, r is defined as follows:

$$f_{t,r}(c, \mathsf{params}) = \begin{cases} 1 & \text{if } (\mathsf{params} = \mathsf{params}^r) \wedge (\mathsf{Dec}(\mathsf{sk}^r, c) \neq \perp) \wedge (\mathsf{Dec}(\mathsf{sk}^r, c) < t) \\ 0 & \text{otherwise.} \end{cases}$$

Intuitively, each concept $f_{t,r}$ evaluates the threshold function with parameter t on the decryption of the input ciphertext c. The particular syntax of the definition handles various technical complications; further discussion appears in (Bun and Zhandry, 2016).

Order revealing encryption enables determining the plaintext ordering given the ciphertexts. While it can be shown that no private PAC algorithm can efficiently learn EncThr, it also not possible for any online learner to learn EncThr efficiently. To see this, fix a target concept at threshold $t = 2^{d-1}$ and random coin tosses r. As before, the adversary sends examples to the learner where each unlabled example is of the form (c, params^r). The learner can only compare the plaintext ordering given the examples chosen by the adversary. The adversary maintains the largest example with label 1 and smallest example with label 0 that has been presented to the learner so far. In every round, the adversary picks uniformly at random between the smallest available example in the interval on the left side of the threshold (i.e., the smallest available example between the threshold and the largest positive example) and the largest available example in the interval on the right side of the threshold (i.e., the largest available example between the threshold and the largest positive example).

Security of the ORE ensures that for any polynomial time horizon, an efficient learner can do no better than random guessing. Thus, an efficient learner must make super-polynomially many mistakes, so it is not possible to design an efficient online learner for the class EncThr.

To overcome this issue, we design an a concept class similar to EncThr but where the encryption reveals some additional information about the plaintexts that facilitates online learning. That is, we study function-revealing encryption (FRE) schemes that enable a richer class of functionalities over the underlying plaintexts than just comparisons. Tuning this functionality is crucial for proving our separation. On one hand, we need to reveal more than ordering in order to learn online. On the other end, revealing too much information (e.g., the exact distance between the underlying plaintexts) enables constructing an efficient private PAC learner. In fact, even revealing a multiplicative approximation of the distance also allows for constructing efficient private PAC learners – later, we sketch how such a learner can be built using the exponential mechanism.

3.2. Function Revealing Encryption

Function revealing encryption is a cryptographic scheme that lets users evaluate functions on plaintexts given access to only the corresponding ciphertexts. We use FRE that lets us evaluate a "leakage" function leak on plaintexts. We will define a concept class $\mathsf{LEncThr}$ in terms of an abstract FRE such that appropriate conditions on the leakage function leak imply the properties we need for our computational separation.

First, let us describe the general syntax, functionality, and security guarantees we need from FRE, some of which are necessarily non-standard. For instance, the usual definition of FRE allows for function evaluation with overwhelmingly high probability. But for our applications, we require a stronger notion of correct evaluation. We require the evaluation of the function to succeed with probability 1 (perfect correctness). Additionally, we require function evaluation over ciphertexts to always behave consistently with decryption. In particular, the evaluation algorithm should output $\perp$ for ciphertexts that are malformed and do not correspond to any messages (strong correctness).

In general, the leakage function of a function revealing encryption scheme may have arbitrary arity, but we for simplicity we specialize the arity to 3, which captures the way we use such schemes.

Definition 12 *A function revealing encryption scheme* FRE *with functionality* leak *is a tuple of algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *where*

- $\mathsf{Gen}(1^\lambda, 1^d)$ *is a randomized procedure that takes as inputs a security parameter* λ *and plaintext length* d*, and outputs a secret encryption/decryption key* sk *and public parameters* params*.*
- $\mathsf{Enc}(\mathsf{sk}, m)$ *is a randomized procedure that takes as input a secret key* sk *and a message* $m \in \{0,1\}^d$*, and outputs a ciphertext* c*.*
- $\mathsf{Dec}(\mathsf{sk}, c)$ *is a deterministic procedure that takes as input a secret key* sk *and a ciphertext* c*, and outputs a plaintext message* $m \in \{0,1\}^d$ *or a failure symbol* $\perp$*.*
- $\mathsf{Eval}(\mathsf{params}, c_0, c_1, c_2)$ *is a deterministic procedure that aims to reveal the value of* leak *on the plaintexts associated with* c_0, c_1, c_2*.*

Correctness. A FRE scheme must satisfy two separate correctness requirements.

- **Correct Decryption:** This is the standard notion of correctness for an encryption scheme, which says that decryption succeeds. For all security parameters λ and message lengths d, and for all messages m,

$$\mathbf{Pr}[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, m)) = m : (\mathsf{sk}, \mathsf{params}) \leftarrow \mathsf{Gen}(1^\lambda, 1^d)] = 1.$$

- **Correct Evaluation:** We require that the evaluation function succeeds. For every c_0, c_1, c_2 in the ciphertext space, define the auxiliary function $\mathsf{Eval}^{\mathsf{ciph}}_{\mathsf{leak}}(\mathsf{sk}, c_0, c_1, c_2)$ as follows. It first computes $m_b = \mathsf{Dec}(\mathsf{sk}, c_b)$ for $b \in \{0,1,2\}$. If any of m_0, m_1 or m_2 is $\perp$, then $\mathsf{Eval}^{\mathsf{ciph}}_{\mathsf{leak}}$ computes to $\perp$. If $m_0, m_1, m_2 \neq \perp$, then the output is $\mathsf{leak}(\mathsf{Dec}(\mathsf{sk}, c_0), \mathsf{Dec}(\mathsf{sk}, c_1), \mathsf{Dec}(\mathsf{sk}, c_2))$.

  Our definition of "perfect and strong" correctness requires that the evaluation function Eval is always consistent with $\mathsf{Eval}^{\mathsf{ciph}}_{\mathsf{leak}}$. That is, for all security parameters λ, all message lengths d, and all c_0, c_1, c_2 in the ciphertext space,

$$\mathbf{Pr}\left[\mathsf{Eval}(\mathsf{params}, c_0, c_1, c_2) = \mathsf{Eval}^{\mathsf{ciph}}_{\mathsf{leak}}(\mathsf{sk}, c_0, c_1, c_2) : (\mathsf{sk}, \mathsf{params}) \leftarrow \mathsf{Gen}(1^\lambda, 1^d)\right] = 1.$$

Definition 13 (Leakage indistinguishablity security) *An* FRE *scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Comp})$ *is* statically secure *if, for all polynomial-time adversaries* $\mathcal{A}$*,* $|\mathbf{Pr}[W_0] - \mathbf{Pr}[W_1]|$ *is negligible, where* W_b *is the event that* $\mathcal{A}$ *outputs* 1 *in the following interaction between* $\mathcal{A}$ *and a "challenger" algorithm:*

- $\mathcal{A}$ *produces two message sequences* $\{m_1^{(L)}, m_2^{(L)}, \ldots, m_q^{(L)}\}$ *and* $\{m_1^{(R)}, m_2^{(R)}, \ldots, m_q^{(R)}\}$ *such that for all* $i, j, k \in [q]$*,* $\mathsf{leak}(m_i^{(L)}, m_j^{(L)}, m_k^{(L)}) = \mathsf{leak}(m_i^{(R)}, m_j^{(R)}, m_k^{(R)})$*.*
- *The challenger samples* $(\mathsf{sk}, \mathsf{params}) \leftarrow \mathsf{Gen}(1^\lambda, 1^d)$*. It then reveals* params *to* $\mathcal{A}$*, as well as* $c_1, \ldots, c_q$ *where*

$$c_i = \begin{cases} \mathsf{Enc}(\mathsf{sk}, m_i^{(L)}) & \text{if } b = 0 \\ \mathsf{Enc}(\mathsf{sk}, m_i^{(R)}) & \text{if } b = 1. \end{cases}$$

- $\mathcal{A}$ *outputs a guess* b' *for* b*.*

Here, "statically" secure refers to the fact that the adversary must submit all of its challenge messages in a single batch, in contrast to an "adaptive" adversary that may issue challenge messages adaptively depending on the previous ciphertexts received.

On perfectly and strongly correct evaluation. While non-standard, our notions of perfect and strong correctness are important in facilitating our efficient online learner. Note that essentially the same conditions were used in the separation of Bun and Zhandry (2016). For us, these conditions prevent the adversary in the online learning model from either choosing a value of the randomness in Gen that causes the Eval procedure to fail, or by sending the learner malformed ciphertexts as examples.

Let $N = 2^d$ and $[N] = \{1, \dots, N\}$ be a plaintext space. Let $\mathsf{leak} : [N]^3 \to R$ be a (for now, abstract) leakage function with codomain R. Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a statically secure FRE scheme with functionality leak, satisfying our perfect and strong correctness guarantees. We define a concept class LEncThr, which intuitively captures the class of threshold functions where examples are encrypted under the FRE scheme. Following Bun and Zhandry (2016), let $t \in [N+1]$. Let $(\mathsf{sk}^r, \mathsf{params}^r)$ be the secret key and the public parameters output by $\mathsf{Gen}(1^\lambda, 1^d)$ when run on the sequence of coin tosses r. We define

$$f_{t,r}(c, \mathsf{params}) = \begin{cases} 1 & \text{if } (\mathsf{params} = \mathsf{params}^r) \wedge (\mathsf{Dec}(\mathsf{sk}^r, c) \neq \perp) \wedge (\mathsf{Dec}(\mathsf{sk}^r, c) < t) \\ 0 & \text{otherwise.} \end{cases}$$

Note that given t and r, the concept $f_{t,r}$ can be efficiently evaluated.

3.3. Properties of leakage function for separation

We now describe the properties a leakage function that allow us to separate online learning from private PAC learning. We begin by defining the arity-3 leakage function induced by an abstract "distance" function.

Let $X = [2^d]$. Let $\mathsf{dist} : X \times X \to \{-d, \dots, 0, \dots, d\}$. Think of dist as an abstract measure of signed distance between inputs, whose absolute value ranges from $\{0, \dots, d\}$. For example, our construction will eventually take $\mathsf{dist}(x, y) = \mathrm{sgn}(x - y)\lfloor \log_2 |x - y| \rfloor$.The sign of the distance function corresponds to the order of the inputs. That is,

$$\mathsf{dist}(x, y) \begin{cases} < 0 & \text{if } x < y \\ = 0 & \text{if } x = y \\ > 0 & \text{if } x > y. \end{cases}$$

Definition 14 (Distance-induced leakage) *Let* dist *be a signed distance function as described above. We define the leakage function* $\mathsf{leak} : X^3 \to \{<, >, =\}^3 \times \{0, 1\}$ *induced by* dist *as follows.*

$$\mathsf{leak}(x_0, x_1, x_2) = (\mathsf{Comp}(x_0, x_1), \mathsf{Comp}(x_1, x_2), \mathsf{Comp}(x_0, x_2), \mathbb{I}(|\mathsf{dist}(y_0, y_1)| < |\mathsf{dist}(y_1, y_2)|)\,,$$

where Comp *indicates comparison, i.e., " $<$ ", " $>$ " or " $=$ ", and $y_0 \leqslant y_1 \leqslant y_2$ are the inputs x_0, x_1, x_2 in sorted order.*

That is, leak reveals the pairwise comparisons between the inputs x_0, x_1, x_2. It also reveals a bit indicating whether the smaller two plaintexts are closer to each other than the larger two plaintexts.

We now identify the conditions on dist and leak that allow us to prove a computational separation.

3.3.1. SUFFICIENT LEAKAGE FOR ONLINE LEARNING

The online learner we eventually construct for LEncThr is based on the halving algorithm, which exploits each mistake to noticeably decrease the space of remaining consistent hypotheses. Our sufficient condition for online learnability, stated as follows, ensures that each mistake is guaranteed to lead to progress.

Definition 15 *Let* $X = [2^d]$. *Let* dist *and* leak *be the functions as defined in Definition 14. We say* dist *has the* bisection property *if for all* $x, y, z \in X$ *such that* $x < y < z$ *either* $|\mathsf{dist}(y, x)| < |\mathsf{dist}(z, x)|$ *or* $|\mathsf{dist}(z, y)| < |\mathsf{dist}(z, x)|$. *Additionally,* $\mathsf{dist}(x, y) = 0$ *implies that* $x = y$.

We also say that leak *has the bisection property if it is induced by a distance function* dist *with the bisection property.*

3.3.2. SUFFICIENT CONDITION FOR HARDNESS OF PRIVATE LEARNING

Definition 16 *Let* leak *be a leakage function as defined in Definition 14. We say that* leak *has the* log-invariance *property if there exists a polylogarithmic function* κ *and polynomial* ζ *such that for every* $n \in \mathbb{N}$, *the following holds. With probability at least* $1/\zeta(n)$ *over a set* $S = \{x_1, \ldots, x_n\}$ *of points drawn uniformly at random from* X, *for every* $i \in [n]$, *there exists an efficient procedure that outputs a set* R_i *with* $|R_i| \leqslant \kappa(n)$ *such that:*

1. *For all* $m_1, m_2 \in S \setminus R_i$ *and all* z, z' *in the interval* (x_{i-1}, x_{i+1}), *we have* $\mathsf{leak}(m_1, m_2, z) = \mathsf{leak}(m_1, m_2, z')$ *and similarly for all permutations of the inputs.*
2. *For all* $m \in S \setminus R_i$ *and all* z_1, z_2, z_1', z_2' *in the interval* (x_{i-1}, x_{i+1}), *we have* $\mathsf{leak}(m, z_1, z_2) = \mathsf{leak}(m, z_1', z_2')$ *and similarly for all permutations of the inputs.*

That is, with high probability over uniformly random sets S of plaintexts, the leakage function is robust in the following sense. For every i, there is a small (polylogarithmically sized) set of points R_i that can be removed from S such that the leakage function reveals nothing about points in (x_{i-1}, x_{i+1}) via their relationship to points in $S \setminus R_i$.

This is an admittedly technical condition. Intuitively, the "reveals nothing about points in (x_{i-1}, x_{i+1})" condition helps in our lower bound argument to construct pairs of adversarial sequences that respect the constraints imposed by the leakage function. The fact that the set of points R_i has only polylogarithmic size is important for us to use group differential privacy (over removing all of the points in R_i) to preserve an inverse polynomial distinguishing advantage.

4. Efficient Online Learner

We now argue that LEncThr is efficiently online learnable whenever the leak function has the bisection property. Our online learner L (Algorithm 2) operates in two phases. In the first phase, L guesses the label 0 for all examples until it makes its first mistake. This first mistake reveals the correct set of params^r that characterizes the fixed concept. Once L recovers the correct set of parameters, it enters a second phase where it runs a variant of the halving algorithm. That is, it keeps track of the largest example with a positive (1) label and the smallest example with a negative (0) label. In every iteration, L matches the parameters of the received example with params^r to check if the example received is malformed; if so, it predicts label 0. Otherwise, if the example has the

correct public parameters, L uses the Eval function to check if the plaintext corresponding to the plaintext is smaller than the plaintext corresponding to largest positive example seen so far, predicting 1 if this is the case. By the guarantee of perfect and strong correctness of the Eval algorithm, the learner is guaranteed to predict correctly in this case. Similarly, L labels an example with 0 if the plaintext corresponding to the example is greater than the the plaintext corresponding to the smallest example with a negative label. Finally, in the case that the example lies between the largest positive example and the smallest positive example, L checks if it is closer to the largest positive example or the smallest negative example using Eval. It then predicts a label according to whichever point it is closer to.

Since leak satisfies the bisection property, we know that if L makes a mistake in this final case, then the underlying distance dist between the largest positive and smallest negative example reduces by 1. Since dist takes absolute values between 0 and d, the learner can make at most $d+1$ mistakes in this phase.

We now formalize our online learner L. We assume that the first phase is over, so that we've received the correct public params^r. We also assume that we've received at least one positive example and at least one negative example, which will be the case after at most 2 more mistakes.

Simplifying and abusing notation somewhat, let $\mathsf{Comp}(c_0, c_1)$ below denote the information revealed by the leakage evaluation function $\mathsf{Eval}(\mathsf{params}^r, c_0, c_1, c_2)$ about how (the plaintexts underlying) ciphertexts c_0, c_1 compare. Similarly, let $\mathsf{DistComp}(c_0, c_1, c_2)$ denote the information revealed about whether (the plaintexts underlying) c_0, c_1 are closer together, or if c_1, c_2 are closer.

Algorithm 2 Online learner for $\mathsf{LEncThr}$ where leak has the bisection property

Initialize: Public parameters params^r, largest positive example $x_+ = (c_+, \mathsf{params}^r)$, and smallest negative example $x_- = (c_-, \mathsf{params}^r)$

Input: Stream of $x_i = (c_i, \mathsf{params}_i)$ in rounds $i = 1, 2, \ldots$ chosen by an adversary, followed by labels y_i

if $\mathsf{params}_i \neq \mathsf{params}^r$ *or* $\mathsf{Eval}(\mathsf{params}^r, c_+, c_i, c_-) = \perp$ **then**
  Predict $\hat{y}_i = 0$
else if $\mathsf{Comp}(c_i, c_+) =$ " $<$ " *or* $\mathsf{Comp}(c_i, c_+) =$ " $=$ " **then**
  Predict $\hat{y}_i = 1$
else if $\mathsf{Comp}(c_i, c_-) =$ " $>$ " *or* $\mathsf{Comp}(c_i, c_-) =$ " $=$ " **then**
  Predict $\hat{y}_i = 0$
else
  if $\mathsf{DistComp}(c_+, c_i, c_-) = 1$ **then**
    Predict $\hat{y}_i = 1$
    if $\hat{y}_i \neq y_i$ **then**
      Update $c_- \leftarrow c_i$
    end
  if $\mathsf{DistComp}(c_+, c_i, c_-) = 0$ **then**
    Predict $\hat{y}_i = 0$
    if $\hat{y}_i \neq y_i$ **then**
      Update $c_+ \leftarrow c_i$
    end
end

4.1. Analysis

We will use a potential argument to show that the online learner makes at most $d+4$ mistakes. The potential function is simply the absolute value of the distance between the plaintexts underlying c_+ and c_-. That is, for a fixed target function (and hence, choice of secret key sk), define

$$D(c_+, c_-) = |\mathsf{dist}(\mathsf{Dec}(\mathsf{sk}, c_+), \mathsf{Dec}(\mathsf{sk}, c_-)|.$$

(If either decryption fails, then set $D(c_+, c_-) = \perp$.)

We know that $t \in [2^d]$ by the definition of the concept class. We assume that the learner has the knowledge about the correct set of params in our analysis since it takes at most one mistake to recover the correct set of params.

Assume without loss of generality that $c_- = \mathsf{Enc}(\mathsf{sk}, 2^d)$ and $c_+ = \mathsf{Enc}(\mathsf{sk}, 1)$, as different choices of the underlying plaintexts will only improve the analysis below. Note that the learner makes at most two mistakes to get these initial values. Thus, we assume that $D(c_+, c_-) \leqslant d$ at the beginning of the algorithm. Because of the bisection property of the dist function, we know that every time the learner makes a mistake (and hence, either c_+ or c_- gets updated), the value of $D(c_+, c_-)$ shrinks by at least one. Also, $D(c_+, c_-)$ can never fall below 0.

Lemma 17 *If Algorithm 2 has made m mistakes, then $D(c_+, c_-) \leqslant d - m$.*

Proof We prove this statement by induction on m. As our base case, take $m = 0$; then $D(c_+, c_-) \leqslant d$ as observed above.

Now suppose the claim holds for m mistakes. We now show that after an additional mistake, $D(c'_+, c'_-) \leqslant d - (m+1)$ where c'_+ and c'_- are the updated examples.

Let i be the iteration in which the mistake is made. Following the algorithm, there are two cases we need to analyze.

Case 1: $D(c_+, c_i) < D(c_i, c_-)$**.** Here, the learner incorrectly predicted $\hat{y}_i = 1$, giving the update $c'_- = c_i$. The bisection property of dist guarantees that $D(c_+, c_i) = \min\{D(c_+, c_i), D(c_i, c_-)\} < D(c_-, c_+)$. Since $D(c_-, c_+) \leqslant d - m$ by our inductive hypothesis, we have that $D(c'_+, c'_-) = D(c_+, c_i) \leqslant d - (m+1)$.

Case 2: $D(c_+, c_i) \geqslant D(c_i, c_-)$**.** In this case, the learner incorrectly predicted $\hat{y}_i = 0$, resulting in the update $c'_+ = c_i$. Then again, the bisection property guarantees that $D(c_i, c_-) \leqslant \min\{D(c_+, c_i), D(c_i, c_-)\} < D(c_-, c_+)$. Since $D(c_-, c_+) \leqslant d - m$ by our inductive hypothesis, we have that $D(c'_+, c'_-) = D(c_i, c_-) \leqslant d - (m+1)$.

This completes the proof of the inductive step. ■

Combining this lemma with the fact that D can never fall below 0, we obtain the following.

Theorem 18 *Suppose leak has the bisection property. Then Algorithm 2 (with preprocessing described above) learns the associated concept class $\mathsf{LEncThr}$ with mistake bound $d+4$ and polynomial runtime per example.*

5. Hardness of Privately PAC Learning LEncThr

We now prove that there is no computationally efficient private PAC learner for LEncThr whenever the leakage function leak is log-invariant (Definition 16). The goal of this section is to prove the following statement.

Theorem 19 *Let* FRE $=$ (Gen, Enc, Dec, Eval) *be a statically secure* FRE *scheme where* Eval *efficiently evaluates a log-invariant leakage function* leak. *Then there is no polynomial-time differentially private PAC learner for the associated concept class* LEncThr.

We first provide a proof sketch of the theorem statement. The idea is to show that if there were an accurate, efficient, differentially private PAC learner for LEncThr, then we could use it to construct an efficient adversary that violates the FRE scheme.

Implications of accuracy of the learner. Let $N = [2^d]$ be the space of the plaintexts. Fix the target threshold to $t = N/2$ and construct labeled examples $S = \{(x_1 = (\mathsf{params}, \mathsf{Enc}(\mathsf{sk}, m_1)), y_1)$ $, \ldots, (x_n = (\mathsf{params}, \mathsf{Enc}(\mathsf{sk}, m_n)), y_n)\}$ by sampling plaintexts $\{m_1, \ldots m_n\}$ uniformly at random. We can break up the plaintexts space into buckets of the form $B_i = [m_i, m_{i+1})$. Suppose L is an (α, β)-accurate PAC learner. Then with probability at least $1 - \beta$, the hypothesis h produced by L can distinguish encryptions of messages $m < t$ from encryptions of messages $m \geqslant t$ with accuracy at least $(1 - \alpha)$. By an averaging argument, there exists an index $i \in [n]$ such that the hypothesis can distinguish between consecutive points sampled from bucket B_{i-1} versus points sampled from B_i with probability at least $(1 - \alpha)/n$. The ability of the learner to distinguish between consecutive points is crucial for designing the adversary for the FRE security game. Knowing this, let's see how we can construct an adversary that violates the security of the FRE. A natural first attempt is to construct a pair of challenge sequences $m_1 < \ldots m_{i-1} < m_i^{(L)} < m_{i+1} \ldots m_n$ and $m_1 < \ldots m_{i-1} < m_i^{(R)} < m_{i+1} \ldots m_n$, where $m_i^{(L)}$ is randomly chosen from B_{i-1} and $m_i^{(R)}$ is randomly chosen from B_i. Let's assume for now that for any indices of points $u, w \in [n]$, $\mathsf{leak}(m_u, m_i^{(L)}, m_w) = \mathsf{leak}(m_u, m_i^{(R)}, m_w)$ (we will show later how privacy of the learner helps us achieve this). Then if h can distinguish B_{i-1} from B_i , the adversary can distinguish the two sequences. Unfortunately, this approach doesn't quite work. The hypothesis h is only guaranteed to distinguish B_{i-1} from B_i with probability $(1 - \alpha)/n$. If h fails to distinguish the buckets or distinguishes them in the opposite direction then, the adversary's advantage is lost.

Thus, following the approach of Bun and Zhandry (2016), we consider sequences of challenge messages that differ on two messages. For the "left" challenge sequence our adversary samples two messages from the same of either B_{i-1} or B_i. For the "right" challenge sequence our adversary samples from one message from each bucket B_{i-1} and B_i. Both challenge sequences are completed with the same messages $m_1, \ldots m_{i-1}, m_{i+1}, \ldots m_n$. Let c^0 and c^1 be the ciphertexts corresponding to the messages that are different between the two sequences. If the learned hypothesis h agrees on c^0 and c^1, then the challenge messages are more likely to be from the same bucket. If h disagrees, then the challenge sequences are more likely to be from different buckets. With this setup, any advantage h enjoys over random guessing when the learner succeeds is preserved even if it has no advantage when the learner fails.

The difficulty now is ensuring that the "left" and "right" challenge messages are indistinguishable with respect to the leakage function leak. Sampling multiple messages from each bucket makes this task harder still. We now explain how differential privacy helps us overcome this issue.

Use of differential privacy. Differential privacy of the learner permits us to swap out points in its input dataset while preserving its distinguishing advantage. Obviously, deleting the example corresponding to m_i is essential to having any hope of constructing a pair of challenge sequences that agree on leak. But leak imposes more stringent constraints. The log-invariance property of leak that we define ensures that by removing only a polylogarithmic number of samples (which, by group privacy, doesn't hurt our distinguishing advantage too much), we can indeed construct such challenge sequences.

We now formalize the ideas described in the proof sketch. First, we show that an accurate learner is likely to output a hypothesis that can distinguish between two adjacent buckets.

Lemma 20 *Consider the concept class* LEncThr. *Let L be a $(\alpha = 1/4, \beta = 1/4)$-accurate PAC learner for* LEncThr. *Fix any pair* $(\mathsf{sk}, \mathsf{params})$ *in the range of* Gen *and an (encrypted) threshold concept with $t = N/2$.*

Let $S = \{(x_1 = (\mathsf{params}, \mathsf{Enc}(\mathsf{sk}, m_1)), y_1), \ldots, (x_n = (\mathsf{params}, \mathsf{Enc}(\mathsf{sk}, m_n)), y_n)\}$ where m_i are sampled uniformly at random and $y_i = f_t(m_i)$. Then there exists an $i \in [n]$ such that

$$\mathbf{Pr}\left[\left|\mathop{\mathbf{Pr}}_{m\sim B_i}\left[h\left(\mathsf{Enc}\left(m, \mathsf{sk}\right)\right) = 1\right] - \mathop{\mathbf{Pr}}_{m\sim B_{i+1}}\left[h\left(\mathsf{Enc}\left(m, \mathsf{sk}\right)\right) = 1\right]\right| \geqslant \frac{1}{2n}\right] \geqslant \frac{3}{4n}.$$

Here, the outer probability is over the randomness of the samples S and the randomness of the learner, $h \leftarrow L(S)$, and each $B_i = [m_i, m_{i+1})$.

Proof Fix a dataset S and let h be the hypothesis produced by the learner on input S. Let $B_i = [m_i, m_{i+1})$, $\ell_i = |B_i|/2^d$ and $p_i = \mathbf{Pr}_{m\sim B_i}\left[h\left(\mathsf{Enc}\left(m, \mathsf{sk}\right)\right) = 1\right]$ for each $i \in [n]$. Let k be the index of the bucket where the threshold t lies.

Accuracy of the learner implies that with probability at least $1 - \beta \geqslant 3/4$ over the sample S and the learner's coin tosses, we have

$$\sum_{i=1}^{k-1} p_i\ell_i + \sum_{i=k+1}^{n} (1-p_i)\,\ell_i + p_k\ell_a + (1-p_k)\,\ell_b \geqslant 1 - \alpha = \frac{3}{4},$$

where $\ell_a = |t - m_k|/2^d$ and $\ell_b = |m_{k+1} - t - 1|/2^d$.

We claim that if this is the case, then there exist indices $i < j$ such that $|p_i - p_j| \geqslant 1/2$. To see this, assume instead for the sake of contradiction that there exists p such that for all $i \in [n]$, $|p_i - p| < 1/4$. Then

$$\begin{aligned}
&\sum_{i=1}^{k-1} p_i\ell_i + \sum_{i=k+1}^{n} (1-p_i)\,\ell_i + p_k\ell_a + (1-p_k)\,\ell_b \\
&\qquad < (p + 1/4)\underbrace{\sum_{i=1}^{k-1}\ell_i + \ell_a}_{1/2} + (1 - (p - 1/4))\underbrace{\sum_{i=m+1}^{n}\ell_i + \ell_b}_{1/2} \\
&\qquad = \frac{1}{2}\left(p + 1/4 + 1 - (p - 1/4)\right) \\
&\qquad = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}.
\end{aligned}$$

This contradicts our assumed accuracy of the learner.

Thus, we have shown that

$$\mathbf{Pr}\left[\exists i < j \text{ s.t. } \left|\mathop{\mathbf{Pr}}_{m\sim B_i}\left[h\left(\mathsf{Enc}\left(m,\mathsf{sk}\right)\right)=1\right]-\mathop{\mathbf{Pr}}_{m\sim B_j}\left[h\left(\mathsf{Enc}\left(m,\mathsf{sk}\right)\right)=1\right]\right|\geqslant\frac{1}{2}\right]\geqslant\frac{3}{4}.$$

More compactly, if we denote $\mathbf{Pr}_{m\sim B_i}\left[h\left(\mathsf{Enc}\left(m,\mathsf{sk}\right)\right)=1\right]$ by p_i for all $i\in[n]$, that is,

$$\mathbf{Pr}\left[\exists i<j \text{ s.t. } |p_i-p_j|\geqslant\frac{1}{2}\right]\geqslant\frac{3}{4}.$$

If for some $i<j$ we have $|p_i-p_j|\geqslant 1/2$, then the triangle inequality implies $|p_i-p_{i+1}|+|p_{i+1}-p_{i+2}|+\cdots+|p_{j-1}-p_j|\geqslant\frac{1}{2}$. By averaging, this in turn implies that there exists an index k where $i\leqslant k\leqslant j-1$ for which $|p_k-p_{k+1}|\geqslant 1/2n$. Thus we have,

$$\mathbf{Pr}\left[\exists k \text{ s.t. } |p_k-p_{k+1}|\geqslant\frac{1}{2n}\right]\geqslant\frac{3}{4}.$$

Using the union bound we get,

$$\mathbf{Pr}\left[|p_1-p_2|\geqslant\frac{1}{2n}\right]+\mathbf{Pr}\left[|p_2-p_3|\geqslant\frac{1}{2n}\right]+\cdots+\mathbf{Pr}\left[|p_{n-1}-p_n|\geqslant\frac{1}{2n}\right]\geqslant\frac{3}{4}.$$

Now by averaging, we conclude that there exists an $i\in[n]$ such that

$$\mathbf{Pr}\left[|p_i-p_{i+1}|\geqslant\frac{1}{2n}\right]\geqslant\frac{3}{4n}.$$

Unpacking the definition of p_i, equivalently, there exists an $i\in[n]$ such that

$$\mathbf{Pr}\left[\left|\mathop{\mathbf{Pr}}_{m\sim B_i}\left[h\left(\mathsf{Enc}\left(m,\mathsf{sk}\right)\right)=1\right]-\mathop{\mathbf{Pr}}_{m\sim B_{i+1}}\left[h\left(\mathsf{Enc}\left(m,\mathsf{sk}\right)\right)=1\right]\right|\geqslant\frac{1}{2n}\right]\geqslant\frac{3}{4n}.$$

■

We now use group privacy to show that if we switch $\kappa\log n$ points from S to obtain a new dataset S_i to be used as input to the learner, then the gap above still (approximately) holds.

Lemma 21 *Let $\varepsilon\leqslant 1/\kappa(n)$ and $\delta\leqslant 1/10n$. Let L be a $(\alpha=1/4,\beta=1/4)$-accurate and (ε,δ)-differentially private PAC learner for the concept class* LEncThr. *Consider $S=\{(x_1=(\mathsf{params},\mathsf{Enc}(\mathsf{sk},m_1)),y_1),\ldots,(x_n=(\mathsf{params},\mathsf{Enc}(\mathsf{sk},m_n)),y_n)\}$ where m_i are sampled uniformly at random.*

Let i be the index guaranteed by Lemma 20 for which

$$\mathbf{Pr}\left[\left|\mathop{\mathbf{Pr}}_{m\sim B_i}\left[h_S\left(\mathsf{Enc}\left(m,\mathsf{sk}\right)\right)=1\right]-\mathop{\mathbf{Pr}}_{m\sim B_{i+1}}\left[h_S\left(\mathsf{Enc}\left(m,\mathsf{sk}\right)\right)=1\right]\right|\geqslant\frac{1}{2n}\right]\geqslant\frac{3}{4n},$$

where the outer probability is taken over the sample and the coins of the learner, and $h_S\leftarrow L(S)$.

Let $S_i=S\backslash R_i$ where R_i is any set such that $|R_i|\leqslant\kappa(n)$. Then

$$\mathbf{Pr}\left[\left|\mathop{\mathbf{Pr}}_{m\sim B_i}\left[h_{S_i}\left(\mathsf{Enc}\left(m,\mathsf{sk}\right)\right)=1\right]-\mathop{\mathbf{Pr}}_{m\sim B_{i+1}}\left[h_{S_i}\left(\mathsf{Enc}\left(m,\mathsf{sk}\right)\right)=1\right]\right|\geqslant\frac{1}{2n}\right]\geqslant\frac{1}{10n},$$

where $h_{S_i}\leftarrow L(S_i)$.

Proof Consider a postprocessing A of the learner L defined as follows:

$$A(h) = \left| \mathop{\mathbf{Pr}}_{m \sim B_i} \left[h\left(\mathsf{Enc}\left(m, \mathsf{sk}\right)\right) = 1 \right] - \mathop{\mathbf{Pr}}_{m \sim B_{i+1}} \left[h\left(\mathsf{Enc}\left(m, \mathsf{sk}\right)\right) = 1 \right] \right|.$$

Let T be the set of outcomes for which $A(h) \geqslant 1/2n$. By hypothesis, we have $\mathbf{Pr}\left[A(L(S)) \in T\right] \geqslant 3/4n$, where the probability is taken over the sample and the coins of the learner.

Using the post-processing property of differentially private mechanisms, we get that $A \circ L$ is (ε, δ)-differentially private. Switching the input dataset from S to S_i and using group privacy, we get

$$\mathbf{Pr}\left[A\left(L(S)\right) \in T\right] \leqslant e^{|R_i|\varepsilon} \mathbf{Pr}\left[A\left(L(S_i)\right) \in T\right] + \frac{e^{|R_i|\varepsilon} - 1}{e^{\varepsilon} - 1} \cdot \delta.$$

Since $|R_i| \leqslant \kappa(n)$, then as long as $\varepsilon \leqslant 1/\kappa(n)$ and $\delta \leqslant 1/4n$, we get $\mathbf{Pr}\left[A\left(L(S_i)\right) \in T\right] \geqslant \frac{1}{10n}$. ■

Algorithm 3 Adversarial strategy using DP-PAC Learner

1. Set $t = N/2$ and choose uniformly at random $i \sim [n]$.
2. Sample n points uniformly at random and permute in increasing order to get $P = (m_1, \ldots, m_{i-1}, m_i, m_{i+1}, \ldots m_n)$
3. Construct pairs $\left(m_L^0, m_L^1\right)$ and $\left(m_R^0, m_R^1\right)$ as follows. Let $B_{i-1} = [m_{i-1}, m_i)$ and $B_i = [m_i, m_{i+1})$. Sample $m_L^0 < m_L^1$ from B_j for a random choice of $j \in \{i-1, i\}$ and sample m_R^0 from B_{i-1} and m_R^1 from B_i.
4. Let $P_i = P \setminus R_i$, where $|R_i| \leqslant \kappa(n)$ as guaranteed by log-invariance. Challenge on the pair of sequences $P_i \cup \{m_L^0, m_L^1\}$ and $P_i \cup \{m_R^0, m_R^1\}$ (in sorted order) and receive the sequence of ciphertexts $\left(c_1, \ldots, c_i^0, c_i^1, \ldots c_{n-|R_i|}\right)$.
5. Remove c_i^0, c_i^1 from the set of ciphertexts and construct a dataset by attaching public parameters and labels $y_j = f_t(m_j)$, i.e. $S_i = \{(x_1 = (c_1, \mathsf{params}), y_1), \ldots, (x_{n-|R_i|} = (c_{n-|R_i|}, \mathsf{params}), y_{n-|R_i|})\}$. Obtain $h \leftarrow_R L(S_i)$.
6. Set $x_i^0 = (c_i^0, \mathsf{params})$ and $x_i^1 = (c_i^1, \mathsf{params})$. Guess $b' = 0$ if $h(x_i^0) = h(x_i^1)$. Guess $b' = 1$ otherwise.

Theorem 22 *Let L be an $(\alpha = 1/4, \beta = 1/4)$-accurate and (ε, δ)-differentially private PAC learner with $\varepsilon \leqslant 1/\kappa(n)$ and $\delta \leqslant 1/4n$ for the concept class* LEncThr*, where the underlying* FRE *scheme is instantiated using a log-invariant leakage function* leak*. Then there exists an adversary that wins the security game of the* FRE *with advantage at least $1/\mathrm{poly}(n)$.*

Proof We describe our adversarial strategy as Algorithm 3.

Note that a randomly chosen $i \in [n]$ meets the guarantee described in Lemma 20 with probability at least $1/n$. Moreover, since leak is log-invariant, we know that there exists an efficient

procedure that outputs P_i as described in step 4 with probability at least $1/\zeta(n)$ for some polynomial ζ. (Otherwise, our adversary can just output a random guess.) Putting these together, we get the following guarantee.

$$\mathbf{Pr}\left[\left|\mathop{\mathbf{Pr}}_{m\sim B_i}\left[h_{S_i}\left(\mathsf{Enc}\left(m,\mathsf{sk}\right)\right)=1\right]-\mathop{\mathbf{Pr}}_{m\sim B_{i+1}}\left[h_{S_i}\left(\mathsf{Enc}\left(m,\mathsf{sk}\right)\right)=1\right]\right|\geqslant\frac{1}{2n}\right]\geqslant\frac{1}{10n}\cdot\frac{1}{n}\cdot\frac{1}{\zeta(n)}$$
$$=\frac{1}{\mathrm{poly}(n)}. \quad (1)$$

Now fix a realization of S. As before, let $p_i=\mathbf{Pr}_{m\sim B_i}[h_{S_i}(\mathsf{Enc}(m,\mathsf{sk}))=1]$ for each $i\in[n]$. The advantage of the adversary in the security game under this realization of S is

$$\begin{aligned}\mathbf{Pr}\left[b'=b\right]&=\frac{1}{2}\left(\mathbf{Pr}\left[h(x_i^0)=h(x_i^1)\mid b=L\right]+\mathbf{Pr}\left[h(x_i^0)\neq h(x_i^1)\mid b=R\right]\right)\\&=\frac{1}{2}\left(\frac{1}{2}\left(p_i^2+(1-p_i)^2+p_{i+1}^2+(1-p_{i+1})^2\right)+\left(1-p_ip_{i+1}-(1-p_i)\left(1-p_{i+1}\right)\right)\right)\\&=\frac{1}{2}\left(\frac{1}{2}\left(2p_i^2+2p_{i+1}^2+2-4p_ip_{i+1}\right)\right)\\&=\frac{1}{2}\left(1+\left(p_i-p_{i+1}\right)^2\right).\end{aligned}$$

Thus, if $p_i-p_{i+1}\geqslant\frac{1}{2n}$, then the advantage is at least $\frac{1}{4n^2}$. For other values of p_i and p_{i+1}, the advantage is still non-negative. From Equation 1, we know that $p_i-p_{i+1}\geqslant\frac{1}{2n}$ is with probability at least $1/\mathrm{poly}(n)$. Hence, the overall advantage of the adversary over the random choice of S is at least $1/\mathrm{poly}(n)$. ■

6. Identifying an Appropriate Leakage function: tfld

6.1. Results with tfld leakage

In this section, we describe an explicit distance and induced leakage function, denoted tfld, that has both the bisection and log-invariance properties.

Definition 23 *Let $X=[2^d]$ and $m_0,m_1\in X$. We define* fld *as a function that reveals the signed floor-log distance between the inputs i.e.*

$$\mathsf{fld}(m_0,m_1)=\begin{cases}0 & \text{if } m_0=m_1\\ \lfloor\log(m_0-m_1)\rfloor+1 & \text{if } m_0>m_1\\ -\lfloor\log(m_1-m_0)\rfloor-1 & \text{otherwise.}\end{cases}$$

The induced leakage function tfld *is thus*

$$\mathsf{tfld}(m_0,m_1,m_2)=\big(\mathsf{Comp}(m_0,m_1),\mathsf{Comp}(m_1,m_2),\mathsf{Comp}(m_0,m_2),\mathbb{I}(\lfloor\log|m_1-m_0|\rfloor<\lfloor\log|m_2-m_1|\rfloor)\big)$$

where $\mathsf{Comp}(m_0,m_1)$ *reveals if* $m_0<m_1$ *or* $m_0>m_1$ *or* $m_0=m_1$.

Lemma 24 *The floor-log distance function* fld *(and hence, its induced leakage function* tfld*) has the bisection property.*

Proof

First, the definition of fld ensures that it reveals the ordering of inputs. Moreover, it is easy to see that $0 \leqslant \mathsf{fld}(m_0, m_1) \leqslant d$.

We now argue that for any $m_0 < m_1 < m_2$, either $\mathsf{fld}(m_1, m_0) < \mathsf{fld}(m_2, m_0)$ or $\mathsf{fld}(m_2, m_1) < \mathsf{fld}(m_2, m_0)$. Let $z = \mathsf{fld}(m_2, m_0)$ which implies that $2^{z-1} \leqslant m_2 - m_0 < 2^z$. For the sake of contradiction, assume that neither $\mathsf{fld}(m_1, m_0)$ nor $\mathsf{fld}(m_2, m_1)$ are less than z. It is immediate that neither $\mathsf{fld}(m_1, m_0)$ or $\mathsf{fld}(m_2, m_1)$ can be greater than z. In the case that both of them are equal to z, we would have $2^{z-1} \leqslant m_2 - m_1 < 2^z$ and $2^{z-1} \leqslant m_1 - m_0 < 2^z$. This implies that $m_2 - m_0 \geqslant 2^z$ which is a contradiction. ■

Corollary 25 *Algorithm 2 learns* $\mathsf{LEncThr}$ *under leakage function* tfld *with mistake bound* $d + 4$ *and polynomial runtime per example.*

We now argue that tfld has the log-invariance property. First, we establish two helpful senses in which uniformly random points are well-spread.

Lemma 26 *Call a multiset of points* $S = \{m_1, \ldots, m_n\} \subseteq [2^d]$ regular *if for every* i*, we have* $|A_i \cap S| \leqslant 50 \log^2 n$ *where*

$$A_i = \left\{ x \in [2^d] \mid 2^z - \frac{4 \log n \cdot 2^d}{n} \leqslant |x - m_i| \leqslant 2^z + \frac{4 \log n \cdot 2^d}{n} \text{ for some } z \in \{0, 1, \ldots, d-1\} \right\}.$$

A uniformly random set of points S *is regular with probability at least* $1 - 1/n$*.*

Proof First observe that regardless of the realization of m_i, we have

$$|A_i| \leqslant 2 \cdot \frac{4 \log n \cdot 2^d}{n} + \sum_{z=\log(4 \log n \cdot 2^d/n)}^{d} 4 \cdot \frac{4 \log n \cdot 2^d}{n} \leqslant \frac{24 \log^2 n \cdot 2^d}{n}.$$

Given m_i, the remaining points in S remain uniformly random. Therefore, each of these $n - 1$ remaining points intersects A_i independently with probability $24 \log^2 n / n$. By a Chernoff bound, the probability that more than $50 \log^2 n$ of these points intersects A_i is at most $e^{-4 \log^2 n}$. Taking a union bound over $i = 1, \ldots, n$ completes the proof. ■

Lemma 27 *Let* $S = \{m_1, \ldots, m_n\}$ *consist of* n *points drawn uniformly at random from* $[2^d]$*, and arranged in nondecreasing order. With probability at least* $1 - \frac{1}{n}$ *over the sampling of* S*, for all* $i \in \{0, 1, \ldots, n\}$*, we have* $|B_i| \leqslant \frac{4 \log n \cdot 2^d}{n}$*, where* $B_i = [m_i, m_{i+1})$ *and* $m_0 = 0$ *and* $m_{n+1} = 2^d$*.*

Proof Consider dividing $[2^d]$ into disjoint consecutive intervals of length $\frac{2 \log n \cdot 2^d}{n}$. We show that with high probability over a random sampling of n points, every interval will contain at least one sampled point.

Let $U_1, \dots, U_{\mathcal{I}}$ denote these disjoint intervals, where $\mathcal{I} = \frac{n}{2 \log n}$. Our goal is to show that $\mathbf{Pr}[\forall i \in [\mathcal{I}] : \mathsf{count}(U_i) \geqslant 1] \geqslant 1 - \frac{1}{n}$, where $\mathsf{count}(U_i)$ evaluates the total number of sampled points in the interval U_i.

Fix some $i \in [\mathcal{I}]$. Then

$$\mathbf{Pr}[\mathsf{count}(U_i) < 1] = \mathbf{Pr}[\mathsf{count}(U_i) = 0] = \left(\frac{\mathcal{I} - 1}{\mathcal{I}}\right)^n$$

By the union bound, we get

$$\begin{aligned}
\mathbf{Pr}[\exists i \in [\mathcal{I}] : \mathsf{count}(U_i) < 1] &\leqslant \mathcal{I} \cdot \left(\frac{\mathcal{I} - 1}{\mathcal{I}}\right)^n \\
&= \mathcal{I} \cdot \left(1 - \frac{1}{\mathcal{I}}\right)^n \\
&\leqslant \frac{\mathcal{I}}{\exp\left(n/\mathcal{I}\right)} \\
&\leqslant \frac{n}{n^2} = \frac{1}{n}.
\end{aligned}$$

Taking the complement of this event, we conclude that with probability at least $1 - \frac{1}{n}$ over the sampling of S, for all $i \in \{0, 1, \dots, n\}$, we have $|B_i| \leqslant \frac{4 \log n \cdot 2^d}{n}$. ■

Lemma 28 *Let $S = \{m_1, \dots m_n\}$ consist of points sampled uniformly at random from $[2^d]$. Then with probability at least $1 - 2/n$ over the sampling, for all $i \in [n]$, there exists an efficiently computable set of points R_i with $|R_i| \leqslant 50 \log^2 n$ such that for all $y \in S \setminus R_i$, $\mathsf{fld}\,(y, m_{i-1}) = \mathsf{fld}\,(y, m_{i+1})$.*

Proof Let $G = 4 \log n \cdot 2^d / n$. By Lemma 26, we have that S is regular with overwhelming probability. By Lemma 27, we have that all bucket lengths $|B_i| \leqslant G$ with high probability. We will show that if both of these events hold, then we can construct the appropriate sets R_i. For each i, define $R_i = A_i \cap S$, using the notation from Lemma 26, which has the requisite size.

Fix some $m \in S \setminus R_i$ such that $m \leqslant m_{i-1}$. (We can make a symmetric argument for $m \geqslant m_{i+1}$). Let $p = \mathsf{fld}\,(m, m_i)$. By properties of floor-log distance and the construction of the set R_i we know that $|m - m_i| < 2^p - G$. By the triangle inequality, we have $|m - m_{i+1}| \leqslant |m - m_i| + |m_i - m_{i+1}| < 2^p - G + G = 2^p$.

On the other hand, from the construction of R_i, we have that $|m - m_{i+1}| \geqslant |m - m_i| > 2^{p-1} + G$. This implies that $\mathsf{fld}\,(m, m_{i+1}) = p = \mathsf{fld}\,(m, m_i)$.

We now argue that $\mathsf{fld}(m, m_{i-1}) = p$ as well. It is easy to see that $|m - m_{i-1}| < |m - m_i| < 2^p - G$. On the other hand,

$$\begin{aligned}
|m - m_{i-1}| &= |m - m_i| - |m_{i-1} - m_i| \\
&\geqslant |m - m_i| - G \\
&> 2^{p-1} + G - G = 2^{p-1}.
\end{aligned}$$

This proves that $\mathsf{fld}(m, m_{i-1}) = \mathsf{fld}(m, m_i) = \mathsf{fld}(m, m_{i+1}) = p$ for all $m \in S \setminus R_i$.

■

Corollary 29 *The leakage function* tfld *induced by* fld *has the log-invariance property with* $\kappa(n) = 50\log^2 n$.

Proof Let S consist of uniformly random samples and construct the sets R_i as in Lemma 28. It immediately follows that for all $m_1, m_2 \in S \setminus R_i$ and all z, z' in the interval of (m_{i-1}, m_{i+1}), we have $\mathsf{tfld}(m_1, m_2, z) = \mathsf{tfld}(m_1, m_2, z')$ since $\mathsf{fld}(m_1, z) = \mathsf{fld}(m_1, z')$ and $\mathsf{fld}(m_2, z) = \mathsf{fld}(m_2, z')$. It also follows that for all; $m \in S \setminus R_i$ and all z_1, z_2, z_1', z_2' in the interval (m_{i-1}, m_{i+1}), we have $\mathsf{tfld}(m, z_1, z_2) = \mathsf{tfld}(m, z_1', z_2')$. ■

Combining Corollary 29 with Theorem 22 yields the following hardness result.

Corollary 30 *Let* $\mathsf{FRE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Comp})$ *be a statically secure function revealing encryption scheme with leakage function* tfld. *Then there is no* $(\alpha = 1/4, \beta = 1/4)$*-accurate and* (ε, δ)*-differentially private PAC learner for the concept class* $\mathsf{LEncThr}$ *with* $\varepsilon = 1/50\log^2 n$ *and* $\delta = 1/4n$.

We now state a result from Balle et al. (2018) that (building on the "secrecy of the sample" argument from Kasiviswanathan et al. (2011)) that enables efficiently reducing ε parameter of a differentially private algorithm using random sampling. We then use this theorem to obtain our main separation result (Theorem 32).

Theorem 31 *Fix* $\varepsilon \leqslant 1$ *and let* $\mathcal{A}$ *be an* (ε, δ)*-differentially private algorithm operating on datasets of size* m. *For* $n \geqslant 2m$, *construct an algorithm* $\tilde{\mathcal{A}}$ *that, on input a dataset* D *of size* n, *subsamples (without replacement)* m *records from* D *and runs* $\mathcal{A}$ *on the result. Then* $\tilde{\mathcal{A}}$ *is* $(\tilde{\varepsilon}, \tilde{\delta})$*–differentially private for*

$$\tilde{\varepsilon} = \frac{(e^{\varepsilon} - 1)\, m}{n} \quad \textit{and} \quad \tilde{\delta} = \frac{m}{n} \cdot \delta.$$

Theorem 32 *Let* $\mathsf{FRE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Comp})$ *be a statically secure function revealing encryption scheme with leakage function* tfld. *Define the associated concept class* $\mathsf{LEncThr}$. *Then* $\mathsf{LEncThr}$ *is online learnable in polynomial time with a polynomial mistake bound. However, there is no* $(\alpha = 1/4, \beta = 1/4)$*-accurate and* $(\varepsilon = 1, \delta = 1/4n)$*-differentially private PAC learner for* $\mathsf{LEncThr}$.

Proof Corollary 25 shows the existence of an efficient online learner for $\mathsf{LEncThr}$ under the leakage function tfld with mistake bound $d+4$. This proves the efficient learnability of $\mathsf{LEncThr}$ in the online learning model.

We now argue about the computational hardness of $\mathsf{LEncThr}$. Corollary 30 shows that there is no $(\alpha = 1/4, \beta = 1/4)$-accurate and $(\varepsilon = 1/50\log^2 n, \delta = 1/4n)$-differentially private PAC learner for the concept class $\mathsf{LEncThr}$, where the sample size n is any polynomial in the problem description size d. We now use the result that ε can be amplified efficiently by subsampling to show the non-existence of a learner with same accuracy guarantess but worse privacy guarantees.

For the sake of contradiction, assume the existence of a learner $\tilde{L}$ for $\mathsf{LEncThr}$ that is $(\alpha = 1/4, \beta = 1/4)$-accurate and $(\varepsilon = 1, \delta = 1/4m)$-differentially private using $m = \mathrm{poly}(d)$ samples. We now construct a learner L for $\mathsf{LEncThr}$ that is $(\alpha = 1/4, \beta = 1/4)$-accurate and

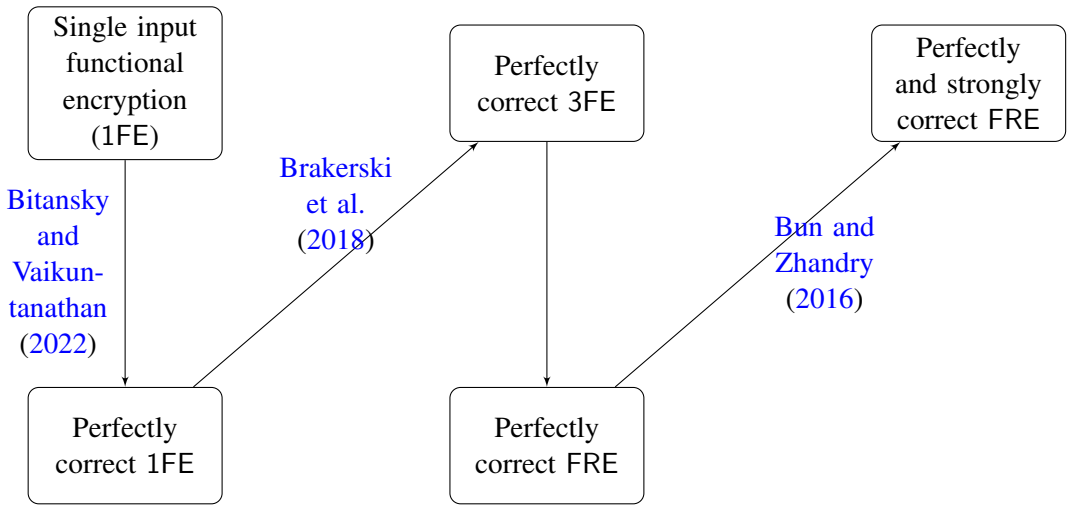

Figure 1: Sketch of construction

$(\varepsilon = 1/50\log^2 n, \delta = 1/4n)$-differentially private using $n = 100m\log^2 m = \mathrm{poly}(d)$ samples. The learner L subsamples a dataset of size m without replacement from its input and runs $\tilde{L}$ on it.

From the guarantees of Theorem 32, we obtain that L is $(\varepsilon = 1/50\log^2 n, \delta = 1/4n)$-differentially private for sufficiently large n. Moreover, since we are running $\tilde{L}$ on subsamples that were sampled without replacement from a dataset whose elements were sampled in an i.i.d. fashion, the output of $\tilde{L}$ is identically distributed to the output of L. This guarantees that L is $(\alpha = 1/4, \beta = 1/4)$-accurate.

However, we have shown in Corollary 30 that such an L cannot exist. So we conclude the non-existence of an $(\alpha = 1/4, \beta = 1/4)$-accurate and $(\varepsilon = 1, \delta = 1/4n)$-differentially private PAC learner for LEncThr. ■

7. Constructing FRE with tfld Evaluation

We now describe sufficient cryptographic and complexity theoretic assumptions to construct function revealing encryption with any leakage computable by poly-size circuits, including tfld.

The "heavy hammer" in our construction is single-input functional encryption for all poly-size circuits. The existence of this primitive is roughly equivalent to indistinguishability obfuscation; a recent breakthrough of Jain et al. (2021) showed that both can be based on a slate of reasonable assumptions described below.

Theorem 33 (Jain et al. (2021)) *Let λ be a security parameter, p be an efficiently sampleable λ-bit prime, and $k = k(\lambda)$ be a large enough polynomial. Assume:*

- *The* SXDH *assumption with respect to a bilinear groups of order p,*
- *The* LWE *assumption with modulus-to-noise ratio 2^{k^ε} where $k = k(\lambda)$ is the dimension of the secret,*

- *The existence of $\gamma-$ secure perturbation resilient generators $\Delta\mathsf{RG} \in (\mathsf{deg}\ 2, \mathsf{deg}\ d)$ over $\mathbb{Z}_p$ for some constant $d \in \mathbb{N}$, with polynomial stretch.*

Then there exists a secret-key functional encryption scheme for polynomial sized circuits having adaptive collusion resistant security, full compactness and perfect correctness.

Figure 1 shows our path for building the perfectly and strongly correct FRE we need from single-input functional encryption.

First, we use the following result of Bitansky and Vaikuntanathan (2022) which gives a complexity-theoretic assumption under which we can guarantee correctness with probability 1.

Theorem 34 (Bitansky and Vaikuntanathan (2022)) *Assume the existence of one-way functions and functions with deterministic (uniform) time complexity $2^{O(n)}$, but non-deterministic circuit complexity $2^{\Omega(n)}$. Then any cryptographic scheme that is secure under parallel repetitions can be made perfectly correct.*

Next, we apply a transformation of Brakerski et al. (2018), who show how to construct a multi-input functional encryption scheme from a single-input functional encryption scheme for all circuits. Note that perfect correctness of the single-input scheme translates into perfect correctness of the resulting multi-input scheme.

Theorem 35 (Brakerski et al. (2018)) *Assume the existence of*

- *A private-key single-input functional encryption scheme for all polynomial-size circuits.*
- *A pseudorandom function family.*

Then there exists a private-key three-input functional encryption scheme for all polynomial-size circuits.

A function revealing encryption scheme is a special case of a multi-input functional encryption scheme where only a single fixed functionality is supported. So what remains is to ensure "strong" correctness. To obtain this, we can invoke a transformation of Bun and Zhandry (2016), who showed how to obtain strong correctness for ORE by attaching a NIZK proof that encryption was performed correctly. Their construction (stated as Theorem 4.1 in their paper) is not specific to ORE and holds for general leakage as stated below.

Theorem 36 (Bun and Zhandry (2016)) *Assuming the existence of a function-revealing encryption scheme with leakage* leak*, a perfectly binding commitment scheme, and perfectly sound non-interactive zero knowledge proofs for* NP*, there is a strongly correct function-revealing encryption scheme with leakage* leak*.*

Perfectly binding commitments can be built from injective one-way functions; moreover, the injectivity requirement can be removed if the circuit lower bound described in Theorem 35 holds (Barak et al., 2007). Perfectly sound NIZKs can be built from bilinear maps (Groth et al., 2012).

Invoking Theorem 32 with the construction we've outlined, and using the fact that (functional) encryption implies the existence of one-way functions, we obtain the following separation.

Theorem 37 *Assume the existence of functional encryption for poly-size circuits (obtainable via the assumptions in Theorem 33), functions computable in time $2^{O(n)}$ with non-deterministic circuit complexity $2^{\Omega(n)}$, and perfectly sound non-interactive zero knowledge proofs for* NP. *Then there exists a concept class that is is online learnable in polynomial time with a polynomial mistake bound. However, this class cannot be learned by a $(\alpha = 1/4, \beta = 1/4)$-accurate and $(\varepsilon = 1, \delta = 1/4n)$-differentially private algorithm.*

8. Conclusion

We conclude with the following open questions.

- Can we build FRE schemes satisfying our bisection and log-invariance properties from weaker assumptions? A beautiful line of work Chenette et al. (2016); Cash et al. (2016) constructs "leaky order-revealing encryption schemes" that enable tantalizingly close functionalities to our tfld. These constructions require much weaker cryptographic assumptions, e,g., just one-way functions and pairings, than what seem to be needed to get multi-input functional encryption for all circuits.

- Can one identify a rich, important class of efficient online learners that *can* be efficiently transformed into private PAC learners?

- Putting computational complexity aside, can we obtain an improved separation between private sample complexity and non-private sample complexity (characterized by VC dimension) of learning? The current best separation is still only a factor of $\log^* |\mathcal{F}|$ (Alon et al., 2022). Similarly, can we improve our general understanding of the sample complexity of private learning?

Acknowledgments

MB was supported by NSF CNS-2046425 and a Sloan Research Fellowship, and thanks Mark Zhandry for helpful conversations about order revealing encryption and its variants. AC was supported by NSF CNS-1915763. RD was supported by NSF CNS-2046425.

References

Scott Aaronson and Guy N. Rothblum. Gentle measurement of quantum states and differential privacy. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 322–333. ACM, 2019. doi: 10.1145/3313276.3316378. URL https://doi.org/10.1145/3313276.3316378.

Naman Agarwal and Karan Singh. The price of differential privacy for online learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 32–40. PMLR, 2017. URL http://proceedings.mlr.press/v70/agarwal17a.html.

Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite Littlestone dimension. In *Proceedings of the 51st Annual ACM Symposium on the Theory of Computing*, STOC '19, New York, NY, USA, 2019. ACM.

Noga Alon, Mark Bun, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private and online learnability are equivalent. *J. ACM*, 69(4):28:1–28:34, 2022. doi: 10.1145/3526074. URL https://doi.org/10.1145/3526074.

Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

Hilal Asi, Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private online prediction from experts: Separations and faster rates. In Gergely Neu and Lorenzo Rosasco, editors, *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023, 12-15 July 2023, Bangalore, India*, volume 195 of *Proceedings of Machine Learning Research*, pages 674–699. PMLR, 2023. URL https://proceedings.mlr.press/v195/asi23a.html.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences, 2018.

Boaz Barak, Shien Jin Ong, and Salil Vadhan. Derandomization in cryptography. *SIAM Journal on Computing*, 37(2):380–400, 2007. doi: 10.1137/050641958.

Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. *Machine Learning*, 94(3):401–437, 2014.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. *Theory of Computing*, 12(1):1–61, 2016.

Amos Beimel, Iftach Haitner, Nikolaos Makriyannis, and Eran Omri. Tighter bounds on multi-party coin flipping via augmented weak martingales and differentially private sampling. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 838–849. IEEE Computer Society, 2018. doi: 10.1109/FOCS.2018.00084. URL https://doi.org/10.1109/FOCS.2018.00084.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of pure private learners. *Journal of Machine Learning Research*, 20(146):1–33, 2019. URL http://jmlr.org/papers/v20/18-269.html.

Nir Bitansky and Vinod Vaikuntanathan. A note on perfect correctness by derandomization. *J. Cryptol.*, 35(3):18, 2022. doi: 10.1007/s00145-022-09428-0. URL https://doi.org/10.1007/s00145-022-09428-0.

Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: The SuLQ framework. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '05, pages 128–138, New York, NY, USA, 2005. ACM.

Zvika Brakerski, Ilan Komargodski, and Gil Segev. Multi-input functional encryption in the private-key setting: Stronger security from weaker assumptions. *J. Cryptol.*, 31(2):434–520, 2018. doi: 10.1007/s00145-017-9261-0. URL https://doi.org/10.1007/s00145-017-9261-0.

Mark Bun. A computational separation between private learning and online learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/ee715daa76f1b51d80343f45547be570-Abstract.html.

Mark Bun and Mark Zhandry. Order-revealing encryption and the hardness of private learning. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 176–206. Springer, 2016.

Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '15, pages 634–649, Washington, DC, USA, 2015. IEEE Computer Society.

Mark Bun, Marco Gaboardi, Max Hopkins, Russell Impagliazzo, Rex Lei, Toniann Pitassi, Satchit Sivakumar, and Jessica Sorrell. Stability is stable: Connections between replicability, privacy, and adaptive generalization. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 520–527. ACM, 2023. doi: 10.1145/3564246.3585246. URL https://doi.org/10.1145/3564246.3585246.

David Cash, Feng-Hao Liu, Adam O'Neill, and Cong Zhang. Reducing the leakage in practical order-revealing encryption. *IACR Cryptol. ePrint Arch.*, page 661, 2016. URL http://eprint.iacr.org/2016/661.

Nathan Chenette, Kevin Lewi, Stephen A. Weis, and David J. Wu. Practical order-revealing encryption with limited leakage. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 474–493. Springer, 2016. doi: 10.1007/978-3-662-52993-5_24. URL https://doi.org/10.1007/978-3-662-52993-5_24.

Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the 41st Annual ACM Symposium on the Theory of Computing*, STOC '09, pages 371–380, New York, NY, USA, 2009. ACM.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '06, pages 486–503, Berlin, Heidelberg, 2006a. Springer.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, Berlin, Heidelberg, 2006b. Springer.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015.

Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. *SIAM Journal on Computing*, 44(6):1740–1764, 2015.

Moti Frances and Ami Litman. Optimal mistake bound learning is hard. *Inf. Comput.*, 144(1): 66–82, 1998. doi: 10.1006/inco.1998.2709. URL https://doi.org/10.1006/inco.1998.2709.

Badih Ghazi, Noah Golowich, Ravi Kumar, and Pasin Manurangsi. Sample-efficient proper PAC learning with approximate differential privacy. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 183–196. ACM, 2021. doi: 10.1145/3406325.3451028. URL https://doi.org/10.1145/3406325.3451028.

Alon Gonen, Elad Hazan, and Shay Moran. Private learning implies online learning: An efficient reduction. *NeurIPS*, 2019.

Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 59(3), jun 2012. ISSN 0004-5411. doi: 10.1145/2220357.2220358. URL https://doi.org/10.1145/2220357.2220358.

Moritz Hardt and Jonathan Ullman. Preventing false discovery in interactive data analysis is hard. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '14, pages 454–463, Washington, DC, USA, 2014. IEEE Computer Society.

Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 60–73. ACM, 2021. doi: 10.1145/3406325.3451093. URL https://doi.org/10.1145/3406325.3451093.

Haim Kaplan, Katrina Ligett, Yishay Mansour, Moni Naor, and Uri Stemmer. Privately learning thresholds: Closing the exponential gap. *CoRR*, abs/1911.10137, 2019. URL http://arxiv.org/abs/1911.10137.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, 1998. doi: 10.1145/293347.293351. URL https://doi.org/10.1145/293347.293351.

Michael J. Kearns, Ming Li, Leonard Pitt, and Leslie G. Valiant. On the learnability of boolean formulae. In Alfred V. Aho, editor, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 285–295. ACM, 1987. doi: 10.1145/28395.28426. URL https://doi.org/10.1145/28395.28426.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1987.

Nick Littlestone. From on-line to batch learning. In Ronald L. Rivest, David Haussler, and Manfred K. Warmuth, editors, *Proceedings of the Second Annual Workshop on Computational Learning Theory, COLT 1989, Santa Cruz, CA, USA, July 31 - August 2, 1989*, pages 269–284. Morgan Kaufmann, 1989. URL http://dl.acm.org/citation.cfm?id=93365.

Pasin Manurangsi. Improved inapproximability of VC dimension and littlestone's dimension via (unbalanced) biclique. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 85:1–85:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi: 10.4230/LIPIcs.ITCS.2023.85. URL https://doi.org/10.4230/LIPIcs.ITCS.2023.85.

Pasin Manurangsi and Aviad Rubinstein. Inapproximability of VC dimension and littlestone's dimension. *CoRR*, abs/1705.09517, 2017. URL http://arxiv.org/abs/1705.09517.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 94–103, Washington, DC, USA, 2007. IEEE Computer Society.

Kobbi Nissim, Rann Smorodinsky, and Moshe Tennenholtz. Approximately optimal mechanism design via differential privacy. In Shafi Goldwasser, editor, *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 203–213. ACM, 2012. doi: 10.1145/2090236.2090254. URL https://doi.org/10.1145/2090236.2090254.

Menachem Sadigurschi and Uri Stemmer. On the sample complexity of privately learning axis-aligned rectangles. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 28286–28297, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ee0e95249268b86ff2053bef214bfeda-Abstract.html.

Marcus Schaefer. Deciding the Vapnik-červonenkis dimension is Σ_3^{p}-complete. *J. Comput. Syst. Sci.*, 58(1):177–182, 1999. doi: 10.1006/jcss.1998.1602. URL https://doi.org/10.1006/jcss.1998.1602.

Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.