On the Complexity of Simulating Auxiliary Input

Yi-Hsiu Chen[1⋆], Kai-Min Chung[2⋆⋆], and Jyun-Jie Liao[2]

[1] Harvard John A. Paulson School Of Engineering And Applied Sciences, Harvard University, USA
[2] Institute of Information Science, Academia Sinica, Taipei, Taiwan

Abstract. We construct a simulator for the simulating auxiliary input problem with complexity better than all previous results and prove the optimality up to logarithmic factors by establishing a black-box lower bound. Specifically, let ℓ be the length of the auxiliary input and ϵ be the indistinguishability parameter. Our simulator is $\tilde{O}(2^\ell \epsilon^{-2})$ more complicated than the distinguisher family. For the lower bound, we show the relative complexity to the distinguisher of a simulator is at least $\Omega(2^\ell \epsilon^{-2})$ assuming the simulator is restricted to use the distinguishers in a black-box way and satisfy a mild restriction.

1 Introduction

In the *simulating auxiliary inputs* problem [JP14], a joint distribution (X, Z) over $\{0,1\}^n \times \{0,1\}^\ell$ is given. the goal is to find a "low complexity" simulator function $h : \{0,1\}^n \to \{0,1\}^\ell$ such that (X, Z) and $(X, h(X))$ are indistinguishable by a family of distinguishers. The non-triviality comes from the "low complexity" requirement. Otherwise, one can simply hardcode the distributions $Z|_{X=x}$ for each x to approximate Z. We call the lemma that addresses this problem *Leakage Simulation Lemma*.

Theorem 1 (Leakage Simulation Lemma, informal). *Let $\mathcal{F}$ be a family of deterministic distinguishers from $\{0,1\}^n \times \{0,1\}^\ell$. For every joint distribution (X, Z) over $\{0,1\}^n \times \{0,1\}^\ell$, There exists a simulator function $h : \{0,1\}^n \to \{0,1\}^\ell$ with complexity $\mathrm{poly}(2^\ell, \epsilon^{-1})$ relative to $\mathcal{F}$ such that for all $f \in \mathcal{F}$,*

$$\left|\Pr\left[f(X, Z) = 1\right] - \Pr\left[f(X, h(X))\right] = 1\right| \leq \epsilon.$$

The "relative complexity" means if we have oracle gates that compute functions in $\mathcal{F}$, then what is the circuit complexity of h when considering those oracle gates [JP14]. A typical choice of a family of distinguishers is a set of all circuits of size s. In that case, we can get a simulator of size $s \cdot \mathrm{poly}(2^\ell, \epsilon^{-1})$.

⋆ Supported by NSF grant CCF-1749750.
⋆⋆ This research is partially supported by the 2016 Academia Sinica Career Development Award under Grant no. 23-17 and Ministry of Science and Technology, Taiwan, under Grant no. MOST 106-2628-E-001-002-MY3.

The Leakage Simulation Lemma implies many theorems in computational complexity and cryptography. For instance, Jetchev and Pietrzak [JP14] used the lemma to give a simpler and quantitatively better proof for the leakage-resilient stream-cipher [DP08]. Also, Chung, Lui, and Pass [CLP15] apply the lemma[3] to study connections between various notions of Zero-Knowledge. Moreover, the leakage simulation lemma can be used to deduce the technical lemma of Gentry and Wichs [GW11] (for establishing lower bounds for succinct arguments) and the Leakage Chain Rule [JP14] for *relaxed-HILL pseudoentropy* [HILL99,GW11].

Before Jetchev and Pietrzak described the Leakage Simulation Lemma as in Theorem 1, Trevisan Tulsiani and Vadhan proved a similar lemma called *Regularity Lemma* [TTV09], which can be viewed as a special case of the Leakage Simulation Lemma by restricting the family of distinguishers in certain forms. In [TTV09], they also showed that all Dense Model Theorem [RTTV08], Impagliazzo Hardcore Lemma [Imp95] and Weak Szemerédi Regularity Lemma [FK99] can be derived from the Regularity Lemma. That means the Leakage Simulation Lemma also implies all those theorems.

As the Leakage Simulation Lemma has many implications, achieving the better complexity bound in $\mathrm{poly}(\epsilon^{-1}, 2^{\ell})$ is desirable. Notably, in certain parameter settings, the provable security level of a leakage-resilient stream-cipher can be improved significantly if we can prove the better bound for the Leakage Simulation Lemma with better complexity bound. (See the next section for a concrete example). Therefore, an interesting question is what is the optimal parameter complexity bound we can get for the Leakage Simulation Lemma? In this paper, we provide an improved upper bound and also show the bound is "almost" optimal.

1.1 Upper Bound Results

Previous Results. In [TTV09], they provided two different approaches for proving the Regularity Lemma. One is by the min-max theorem, and another one is via boosting-type of proof. Although it is not known whether the Regularity Lemma implies the Leakage Simulation Lemma directly, [JP14] adopted both techniques and used them to show the Leakage Chain Rule with complexity bound $\tilde{O}(2^{4\ell}\epsilon^{-4})$.[4]. On the other hand, Vadhan and Zheng derived the Leakage Simulation Lemma [VZ13, Lemma 6.8] using so-called "uniform min-max theorem", which is proved via multiplicative weight update (MWU) method incorporating with KL-projections. The circuit complexity of the simulator they got is $\tilde{O}(s \cdot 2^{\ell}\epsilon^{-2} + 2^{\ell}\epsilon^{-4})$ where s is the size of the distinguisher circuits. Recently, Skórski also used the boosting-type method to achieve the bound $\tilde{O}(2^{5\ell}\epsilon^{-2})$ [Skó16a], then later improved it to $\tilde{O}(2^{3\ell}\epsilon^{-2})$ by incorporating the subgradient method [Skó16b]. Note that the complexity bound in [VZ13] has an additive term, so their result is incomparable to the others.

[3] They also consider the interactive version.

[4] In the original paper, they claimed to achieve the bound $\tilde{O}(2^{3\ell}\epsilon^{-2})$. However, Skórski pointed out some analysis flaws [Skó16a].

Our Results. In this paper, we achieve the bound $\tilde{O}(2^{\ell}\epsilon^{-2})$ for relative complexity, which contains the best components out of three complexity bounds mentioned above. The algorithm we use is also of multiplicative weight update (MWU) method as in [VZ13] but without going through the uniform min-max theorem argument. The additive term $2^{\ell}\epsilon^{-4}$ in [VZ13] is due to the precision issue when performing multiplication of "real numbers". The saving of the additive term is based on the observation mentioned in [VZ13] – the KL-projection step in their MWU algorithm is not needed when proving the Leakage Simulation Lemma. Thus we can potentially simplify the circuit construction. Indeed, we prove that certain level of truncation on weights does not effect the accuracy too much but helps us reducing the circuit complexity. In table 1, we list out and compare all previous results to ours.

Paper	Techinque	Complexity of Simulator
[JP14]	Min-max / Boosting	$\tilde{O}(s \cdot 2^{4\ell}\epsilon^{-4})$
[VZ13]	Boosting with KL-projection	$\tilde{O}(s \cdot 2^{\ell}\epsilon^{-2} + 2^{\ell}\epsilon^{-4})$
[Skó16a]	Boosting with self-defined projection	$\tilde{O}(s \cdot 2^{5\ell}\epsilon^{-2})$
[Skó16b]	Boosting with Subgradient Method	$\tilde{O}(s \cdot 2^{3\ell}\epsilon^{-2})$
This work	Boosting	$O(s \cdot \ell 2^{\ell}\epsilon^{-2})$
	Black-box lower bound	$\Omega(s \cdot 2^{\ell}\epsilon^{-2})$

Table 1. Summary of exisiting upper bound results and our results.

Implication of Our Results As mentioned before, our result yields a proof of better security in leakage-resilient stream-cipher. All previous results suffer from the term ϵ^{-4} [JP14,VZ13][5] or the $2^{3\ell}$ multiplicative factor [Skó16b] in the complexity bound. In particular, Skórski's gave legitimate examples [Skó16a] where the bounds in [JP14] and [VZ13] only guarantee trivial security bounds when ϵ is set to be 2^{-40}. On the other hand, the factor $2^{3\ell}$ (or even $2^{5\ell}$) is significant and makes the guaranteed security bound trivial when the leakage is more than few bits. Therefore, in some reasonable parameter settings, our bound is the only one that can achieve a useful security. Here is a concrete example. If we consdier the stream cipher in [JP14] and follow the settings in [Skó16a, Section 1.6]: The underlying weak PRF has 256 bits security, the target cipher security is $\epsilon' = 2^{-40}$ and the round is 16. If the leakage is $\lambda = 17$ per rounds, then using our bound, we can guarantee the security against 2^{50}-size circuit but all the analyses in [JP14,VZ13,Skó16a] guarantee nothing.

[5] It appears as an additive complexity in [VZ13] and/or a multiplicative term in [JP14].

1.2 Lower Bound Results

Our Results. We show that the simulator must have a "relative complexity" $\Omega(2^\ell\epsilon^{-2})$ to the distinguisher family by establishing a black-box lower bound, where a simulator can only use the distinguishers in a black-box way. Our lower bound requires an additional mild assumption that the simulator on a given input x, does not make a query an $x' \neq x$ to distinguishers.[6] Querying at points different from the input seems not helpful, but that makes the behaviors on different inputs not completely independent, which causes a problem in analysis. Indeed, all the known upper bound algorithms (including the one in this work) satisfy the assumptions we made. Still, we leave it as an open problem to close this gap completely.

Comparison to Related Results. In [JP14], they proved a $\Omega(2^\ell)$ lower bound for relative complexity under a hardness assumption for one-way functions. Besides, there are also lower bound results on the theorems that implied by the Leakage Simulation Lemma, including Regularity Lemma [TTV09], Hardcore Lemma [LTW11], Dense Model Theorem [Zha11], Leakage Chain Rule [PS16] and Hardness Amplification [SV10,AS11]. The best lower bound one can obtain before this work is $\Omega(\epsilon^{-2})$ (from [LTW11,SV10,Zha11]) or $\Omega(2^\ell\epsilon^{-1})$ (from [PS16]). Thus our lower bound is the first tight lower bound $\Omega(2^\ell\epsilon^{-2})$ for Leakage Simulation Lemma. See Section 4.2 for more detailed comparison.

Proof Overview We define an oracle and a joint distribution $(X, Z) \in \{0,1\}^n \times \{0,1\}^\ell$. Considering a family of the distinguishers that each of them makes a single query to the oracle, the simulator has to query the oracle at least $\Omega(2^\ell\epsilon^{-2})$ times to fool all the distinguishers in the family. Therefore, if the only way to access the oracle is through the distinguishers, the simulator must use at least $\Omega(2^\ell\epsilon^{-2})$ distinguishers.

We can treat Z as a randomized function of X. That is, if can define $g : \{0,1\}^n \to \{0,1\}^\ell$ such that $\Pr[g(x) = z] = \Pr[Z = z|X = x]$, then $(X, Z) = (X, g(X))$. The distribution we consider is that the function g is deterministic, but the images are "hidden" from the simulator. Note that it is impossible for a simulator to hardwire all 2^n images. If the oracle receives a query $(x, z) \in \{0,1\}^n \times \{0,1\}^\ell$ with $z = g(x)$, it returns an answer based on the distribution $\mathrm{Bern}(1/2 + \epsilon)$. Otherwise, use the distribution $\mathrm{Bern}(1/2)$. Intuitively, the goal of the simulator is to find $g(x)$ for a given input x. For each z, due to the anti-concentration bound, it has to make $\Omega(\epsilon^{-2})$ many queries to check if $g(x) = z$. And if it has to check a constant fraction of all $z \in \{0,1\}^n$, then the total query complexity is $\Omega(2^\ell\epsilon^{-2})$.

[6] Many black-box lower bounds in related contexts [LTW11,Zha11,PS16] (implicitly) make the same mild assumption. See Section 4.2 for more details.

2 Preliminaries

2.1 Basic Definitions

Notations. For a natural number n, $[n]$ denotes the set $\{1, 2, \ldots, n\}$ and U_n denotes the uniform distribution over $\{0,1\}^n$. For a finite set $\mathcal{X}$, $|\mathcal{X}|$ denotes its cardinality, and $U_{\mathcal{X}}$ denotes the uniform distribution over $\mathcal{X}$. For a distribution X over $\mathcal{X}$, $x \leftarrow X$ means x is a random sample drawn from X. $\mathrm{Bern}(p)$ denotes the Bernoulli distribution with parameter $0 \leq p \leq 1$. For any function f, $\tilde{O}(f)$ means $O(f \log^k f)$ and $\tilde{\Omega}(f)$ means $\Omega(f/\log^k f)$ for some constant $k > 0$.

Definition 1 (Statistical Distance). *Let X and Y be two random variables. The statistical distance (or total variation) between X and Y is denoted as*

$$\Delta(X,Y) = \sum_x \frac{1}{2} \left| \Pr[X = x] - \Pr[Y = x] \right|.$$

Also, we say X and Y are ϵ-close *if $\Delta(X,Y) \leq \epsilon$.*

Definition 2 (Indistinguishability). *Let X, Y be distributions over $\{0,1\}^n$. We say X and Y are* (s,ϵ)-indistinguishable *if for every circuit $f : \{0,1\}^n \to \{0,1\}$ of size s,*

$$\left| \mathop{\mathbb{E}}_{x \leftarrow X}[f(x)] - \mathop{\mathbb{E}}_{y \leftarrow Y}[f(y)] \right| \leq \epsilon.$$

2.2 Multiplicative Weight Update

Consider the following prediction game. In each round, a predictor makes a prediction and receive a payoff. There are N experts that the predictor can refer to. That is, the predictor can (randomly) choose an expert to follow. The goal of the predictor is to minimize total payoff in many rounds. We called the difference between the total payoff of predictor and of the best expert *regret*, which is the criterion we use to measure the performance of the predictor. The Multiplicative weight update (MWU) algorithm provides a good probabilistic strategy for prediction. The overview of the algorithm is as follows. In the first round, the predictor simply chooses an expert uniformly at random. In the following rounds, the predictor updates the probabilities of choosing experts "multiplicatively" according to their performances in the previous round. The formal algorithm and the guarantees by the MWU algorithm is stated below.

Lemma 1 (Multiplicative weight update [AHK12]). *Consider a T-round game such that in t-th round, the predictor chooses a distribution D_t over $[N]$, and obtains a payoff according to the function $f_t : [N] \to [0,1]$. Let $0 < \eta \leq 1/2$ be an update rate. If player 1 chooses D_t as in Algorithm 1, then for every $i \in [N]$,*

$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{j \leftarrow D_t}[f_t(j)] \leq \sum_{t=1}^{T} f_t(i) + \frac{\log N}{\eta} + T\eta.$$

In particular, if we set $\eta = \sqrt{\log N/T}$*, we have*

$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{j \leftarrow D_t}[f_t(j)] \leq \sum_{t=1}^{T} f_t(i) + O\left(\sqrt{T \log N}\right)$$

Algorithm 1: Multiplicative weight update

```
For all i ∈ [N] set w_i := 1.
for t := 1 to T do
    Choose D_t such that D_t(i) ∝ w_i.
    for i := 1 to N do
        w_i := w_i · (1 − η)^{f_t(i)};
```

As the regret grows sub-linearly to T, the predictor can achieve δ average regret when T is large enough.

Corollary 1. *There exists* $T = O\left(\frac{\ln N}{\epsilon^2}\right)$ *such that for all* $i \in [N]$*,*

$$\frac{1}{T}\sum_{t} \mathop{\mathbb{E}}_{j \leftarrow D_t}[f_t(j)] \leq \frac{1}{T}\sum_{t} f_t(i) + \epsilon.$$

Freund and Schapire discovered the connection between MWU algorithm and zero sum game [FS96] by treating the best response of Player 2 as the payoff function. MWU algorithm not only gives a new proof of von Neumanns Min-Max Theorem, but also provides a way to "approximate" the universal strategy obtained by the Min-Max Theorem[7].

Lemma 2 ([FS96]). *Consider a zero-sum game between Player 1 and Player 2 whose (pure) strategy spaces are* $\mathcal{P}$ *and* $\mathcal{Q}$*, respectively, and* $|\mathcal{P}| = N$*. The payoff to Player 2 is defined by the function* $u : \mathcal{P} \times \mathcal{Q} \to [0, 1]$*. We apply the MWU algorithm (Algorithm 1) in the following way to get the mixed strategy* P^* *and* Q^**.*

1. *Treat each pure strategy in* $|\mathcal{P}|$ *as an expert. Let* P_t *denote the mixed strategy described by* D_t *(the* i*-th pure strategy is chosen with probability* $D_t(i)$*).*
2. *Let* Q_t *denote the best response of Player 2 to* P_t*. Namely*

$$Q_t = \min_{Q} \mathop{\mathbb{E}}_{p \leftarrow P_t, q \leftarrow Q} f(p, q)$$

3. *Set the payoff function in the MWU algorithm as* $f_t(\cdot) = M(\cdot, Q_t)$*.*
4. *Let* $P^* = \frac{1}{T}\sum_t P_t$ *and* $Q^* = \frac{1}{T}\sum_t Q_t$*.*

[7] It is called Non-uniform Min-Max Theorem in [VZ13]

If we conduct the above procedure for $T = O(\log N/\epsilon^2)$ rounds, the mixed strategies P^*, Q^* are almost the equilibrium strategies. That is

$$\begin{aligned}\max_q \mathop{\mathbb{E}}_{p\leftarrow P^*}[u(p,q)] - \epsilon &\le \max_Q \min_P \mathop{\mathbb{E}}_{p\leftarrow P, q\leftarrow Q}[u(p,q)] \\ &= \min_P \max_Q \mathop{\mathbb{E}}_{p\leftarrow P, q\leftarrow Q}[u(p,q)] \le \min_p \mathop{\mathbb{E}}_{q\leftarrow Q^*}[u(p,q)] + \epsilon.\end{aligned}$$

3 Simulating Auxiliary Inputs

The formal description of Leakage Simulation Lemma with our improved parameters is as follows.

Theorem 2 (Leakage Simulation Lemma). *Let $n, \ell \in \mathbb{N}$, $\epsilon > 0$ and $\mathcal{F}$ be a collection of deterministic distinguishers $f : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}$. For every distribution (X, Z) over $\{0,1\}^n \times \{0,1\}^\ell$, there exists a simulator circuit $h : \{0,1\}^n \to \{0,1\}^\ell$ such that*

1. *h has complexity $\tilde{O}(2^\ell \epsilon^{-2})$ relative to $\mathcal{F}$. i.e., h can be computed by an oracle-aided circuit of size $\tilde{O}(2^\ell \epsilon^{-2})$ with oracle gates are functions in $\mathcal{F}$.*
2. *(X, Z) and $(X, h(X))$ are indistinguishable by $\mathcal{F}$. That is, for every $f \in \mathcal{F}$,*

$$\left| \mathop{\mathbb{E}}_{(x,z)\leftarrow(X,Z)}[f(x,z)] - \mathop{\mathbb{E}}_{h,x\leftarrow X}[f(x,h(x))] \right| \le \epsilon.$$

Set $\mathcal{F}$ to be a set of Boolean circuits of size at most s, we immediate have the following corollary.

Corollary 2. *Let $s, n, \ell \in \mathbb{N}$ and $\epsilon > 0$. For every distribution (X, Z) over $\{0,1\}^n \times \{0,1\}^\ell$, there exists a simulator circuit of size $s' = \tilde{O}(s \cdot 2^\ell \epsilon^{-2})$ such that (X, Z) and $(X, h(X))$ are (s, ϵ)-indistinguishable.*

3.1 Boosting

There are numbers of proof of Leakage Simulation Lemma as discussed in the introduction. We focus on the "boosting" type of proof as it usually gives us better circuit complexity. The boosting framework has the following structure:

1. Choose a proper initial simulator h.
2. If h satisfies the constraint above, return h. Otherwise, find $f \in \mathcal{F}'$ which violates the constraint.
3. Update h with f and repeat.

Previous proofs in the framework are different in how they update h and correspondingly how they prove the convergence. If the algorithm converges fast and each update does not take too much time, we can get an efficient simulator. Starting from [TTV09], then followed [JP14] and [Skó16a], they use *additive* update on the probability mass function of each $h(x)$. However, additive update

may cause negative weights, so they need an extra efforts (Both algorithm-wise and complexity-wise) to fix it. Vadhan and Zheng use multiplicative weight update instead [VZ13], which not only avoids the issue above but also converges faster. However, the number of bits to represent weights increases drastically after multiplications, and that causes the $O(2^\ell \epsilon^{-4})$ additive term in the complexity. Since the backbone of our algorithm is same as in [VZ13], we review their idea first in the next section, and then show how the additive term can be eliminated in Section 3.3.

3.2 Simulate Leakage with MWU

In this section, we show how MWU algorithm helps in simulating auxiliary inputs and why we can achieve the low round complexity. It is convenient to think Z as a randomized function of X. That is, we can define $g : \{0,1\}^n \to \{0,1\}^\ell$ such that $\Pr[g(x) = z] = \Pr[Z = z | X = x]$, then $(X, Z) = (X, g(X))$. Essentially, the goal is to find an "efficient function" h to simulate g.

Now we show that how the simulation problem problem is related to a zero-sum game, thus can be solved via MWU algorithm. The first step is to remove the one-sided error constraint. Let $\mathcal{F}'$ denote the closure of $\mathcal{F}$ under complement, namely, $\mathcal{F}' = \{f, 1 - f : f \in \mathcal{F}\}$. Then the indistinguishability constraint is equivalent to

$$\forall f \in \mathcal{F}' , \quad \mathop{\mathbb{E}}_{h,x \leftarrow X} [f(x, h(x))] - \mathop{\mathbb{E}}_{g,x \leftarrow X} [f(x, g(x))] \leq \epsilon.$$

Then consider the following zero-sum game: Player 1 choose a simulator h, Player 2 choose a distinguisher f, and the payoff to Player 2 is

$$\mathop{\mathbb{E}}_{h,x \leftarrow X} [f(x, h(x))] - \mathop{\mathbb{E}}_{g,x \leftarrow X} [f(x, g(x))] .$$

One can get a bounded relative complexity of g by simply applying Lemma 2 with treating all functions from $\{0,1\}^n$ to $\{0,1\}^\ell$ as pure strategies of Player 1. However, relative complexity is $O(s \cdot 2^n \ell \epsilon^{-2})$ and hence is inefficient. To solve the above issue, Vadhan and Zheng observed that the marginal distribution of X-part is fixed. Thus we can consider the MWU algorithm for every $X = x$, where in each run of MWU, the Player 1 strategy space is simply a distribution over $\{0,1\}^\ell$, hence the round complexity is merely $O(\ell/\epsilon^2)$.

While the framework Vadhan and Zheng's considered is more general, the proof is also more complicated. Below we give a simpler proof which only uses the no-regret property of MWU.[8] Note that any no-regret algorithms for expert learning will work for this proof. Indeed, by applying online gradient descent instead of MWU we will get an additive boosting simulator. Nevertheless, multiplicative weight update is optimal in expert learning, which explains why MWU converges faster than additive boosting proofs.

[8] We say an online decision-making algorithm is *no-regret* if the average regret tends to zero as T approaches infinity. See, e.g., [Rou16].

Algorithm 2: Construction of Simulator h

1 **Input**: $x \in \{0,1\}^n$
2 **Parameter**: $\epsilon > 0$
3 Let $T = O(n/\epsilon^2)$, $\eta = \sqrt{\log N/T}$.
4 For all $z \in \{0,1\}^\ell$, set $w_x(z) = 1$.
5 Let h_0 be a randomized function such that $\Pr[h_0(x) = z] \propto w_x(z)$.
6 **for** $t = 1 \to T$ **do**
7 *Let* $f_t \in \mathcal{F}' = \arg\max_{f \in \mathcal{F}'} \mathbb{E}_{h_{t-1}, x \leftarrow X}[f(x, h_{t-1}(x))] - \mathbb{E}_{g, x \leftarrow X}[f(x, g(x))]$.
8 **if** $\mathbb{E}_{h_{t-1}, x \leftarrow X}[f(x, h_{t-1}(x))] - \mathbb{E}_{g, x \leftarrow X}[f(x, g(x))] \le \epsilon$ **then**
9 **Return** $h_{t-1}(x)$ *as the output* $h(x)$
10 *For all* $z \in \{0,1\}^\ell$, *set* $w_x(z) = w_x(z) \cdot (1-\eta)^{f_t(x,z)}$
11 *Let* h_t *be a randomized function such that* $\Pr[h_t(x) = z] \propto w_x(z)$.
12 **Return** $h_T(x)$ *as the output* $h(x)$

Lemma 3. *Let X be a distribution over $\{0,1\}^n$ and $g : \{0,1\}^n \to \{0,1\}^\ell$ be a randomized function. For a given error parameter ϵ, the function h defined by Algorithm 2 satisfies*

$$\forall f \in \mathcal{F}', \quad \mathop{\mathbb{E}}_{x \leftarrow X}[f(x, h(x))] - \mathop{\mathbb{E}}_{x \leftarrow X}[f(x, g(x))] \le \epsilon.$$

Proof. For a fixed x, if there exists $f \in \mathcal{F}'$ such that

$$\mathop{\mathbb{E}}_{h}[f(x, h(x))] - \mathop{\mathbb{E}}_{g}[f(x, g(x))] > \epsilon,$$

then the algorithm returns at the line 12. That means for all $t \in [T]$, we have

$$\mathop{\mathbb{E}}_{h_{t-1}}[f_t(x, h_{t-1}(x))] - \mathop{\mathbb{E}}_{g}[f_t(x, g(x))] > \epsilon, \tag{1}$$

and so

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{h_{t-1}}[f_t(x, h_{t-1}(x))] - \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{g}[f_t(x, g(x))] > \epsilon, \tag{2}$$

However, by Corollary 1, for every $z \in \{0,1\}^\ell$,

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{h_{t-1}}[f_t(x, h_{t-1}(x))] \le \frac{1}{T}\sum_{t=1}^{T} f_t(x, z) + \epsilon.$$

By taking z over $g(x)$, we get a contradiction. Therefore, for all $f \in \mathcal{F}$,

$$\mathop{\mathbb{E}}_{h}[f(x, h(x))] - \mathop{\mathbb{E}}_{g}[f(x, g(x))] > \epsilon.$$

Take the expectation of x over X, we conclude the lemma.

3.3 Efficient Approximation

Algorithm 2 provides a simulator which fools all distinguishers in $\mathcal{F}$ by error up to ϵ. However, we have only proved a bound for the number of iterations, but not for the complexity of h_T itself. Actually, the circuit complexity of a naive implementation of Algorithm 2 is not better than using additive boosting. Nevertheless, we will show that there exists an efficient way to implement h_T approximately, of which the complexity is not much larger than evaluating the distinguishers T times.

In below, we assume all functions $f \in \mathcal{F}$ has circuit compleixty at most s. From Algorithm 2, we can see $h_T(x)$ returns z with probability proportional to $(1-\eta)^{\sum_i f_i(x,z)}$. A natural way to approximate h_T is to compute $(1-\eta)^{\sum_i f_i(x,z)}$ for each z and apply a rejection sampling. Without loss of generality, we can assume that $(1-\eta)$ can be represented in $O(\log \frac{1}{\eta})$ bits, and thus, it takes at most $O(k \log \frac{1}{\eta})$ to represent $(1-\eta)^k$ for $k \in \mathbb{N}$. Since $\sum_i f_i(x,z)$ is at most T, it takes $O(Ts + T^2 \log^2 \frac{1}{\eta})$ complexity to compute $(1-\eta)^{\sum_i f_i(x,z)}$ by naive multiplication, or $O(Ts + T^2 \log T \log \frac{1}{\eta})$ via lookup table. Therefore there exists an approximation of h_T of size $O((T^2 \log^2 \frac{1}{\eta} + Ts) \cdot 2^\ell)$, which is $\tilde{O}(s \cdot 2^\ell \epsilon^{-2} + 2^\ell \epsilon^{-4}))$ after expanding T and η. This is the complexity claimed in [VZ13]. As mentioned in [Skó16a], the $\tilde{O}(2^\ell \epsilon^{-4})$ term may dominate in some settings, so the bound in [VZ13] is not always better.

Now we state the idea of approximating normalized weights efficiently. Observe that weights are of the form $(1-\eta)^{\sum_i f_i(x,z)}$. If the total weight is guaranteed to be at least 1, then intuitively, truncating the weight at each $z \in \{0,1\}^\ell$ a little amount does not influence the result distribution too much. Hopefully, if the truncated values can be stored with a small number of bits, a lookup table which maps $\sum_i f_i(x,z)$ to the truncated value of $(1-\eta)^{\sum_i f_i(x,z)}$ is affordable. In the lemma below we formalize the above intuition.

Lemma 4. *Suppose there are two sequences of positive real numbers $\{\gamma_i\}_{i\in[n]}$, $\{w_i\}_{i\in[n]}$ such that $\forall i \in [n]$, $\gamma_i \le w_i$. Let $r = \sum_i \gamma_i / \sum_i w_i$ and X, X' be a distribution over $[n]$ such that $\Pr[X = i] \propto w_i$ and $\Pr[X' = i] \propto (w_i - \gamma_i)$, respectively. Then $\Delta(X, X') \le \frac{r}{1-r}$.*

Proof.

$$\begin{aligned}
\Delta(X, X') &= \frac{1}{2} \sum_z \left| \frac{w_z}{\sum_i w_i} - \frac{w_z - \gamma_z}{\sum_i (w_i - \gamma_i)} \right| \\
&= \frac{1}{2} \sum_z \left| \frac{\gamma_z \sum_i w_i - w_z \sum_i \gamma_i}{(\sum_i w_i)^2 (1-r)} \right| \\
&\le \frac{1}{2} \sum_z \frac{w_z \sum_i \gamma_i + \gamma_z \sum_i w_i}{(\sum_i w_i)^2 (1-r)} \\
&= \frac{\sum_i w_i \sum_i \gamma_i}{(\sum_i w_i)^2 (1-r)} = \frac{r}{1-r}
\end{aligned}$$

where the inequality follows from the triangle inequality.

Corollary 3. *Let $h' : \{0,1\}^n \to \{0,1\}^\ell$ be a function which satisfies*

$$\Pr[h'(x) = z] = \frac{(1-\eta)^{\sum_i f_i(x,z)} - \gamma_{x,z}}{\sum_{z'} \left((1-\eta)^{\sum_i f_i(x,z')} - \gamma_{x,z'}\right)}$$

where

$$\gamma_{x,z} \leq \min\left\{(1-\eta)^{\sum_i f_i(x,z)}\ ,\ \frac{\eta}{2^\ell(1+\eta)} \cdot \sum_{z'} (1-\eta)^{\sum_i f_i(x,z')}\right\}.$$

Then for any $x \in \mathcal{X}$, $h'(x)$ is η-close to $h_T(x)$.

By the above corollary, the following procedure gives a good approximation of h_T.

1. For every $z \in \{0,1\}^\ell$, compute $Adv(x,z) = \sum_i f_i(x,z) - \min_{z'}(\sum_i f_i(x,z'))$. This can be done by a circuit of size $O(2^\ell \cdot (sT + T\log T))$.
2. Because there is z_0 such that $Adv(x,z_0) = 0$, we have $\sum_z (1-\eta)^{Adv(x,z)} \geq 1$. Let $k = O(\ell \log(1/\delta))$ be the smallest integer which satisfies $2^{-k} \leq \frac{\eta}{2^\ell(1+\eta)}$. By Corollary 3, if we truncate $(1-\eta)^{Adv(x,z)}$ down to the closest multiple of 2^{-k}, the corresponding distribution is still η-close to $h_T(x)$. Let $h'(x)$ denote the truncated distribution.
3. Observe that the truncated value is positive only if $Adv(x,z)$ is less than some threshold $t = O(k/\eta)$. Therefore we can build a lookup table consists of the truncated value of $(1-\eta)^j$ for $j \in [t]$. Such table is of size $O(t \log t \cdot k)$. With this table we can query truncated value of $(1-\eta)^{Adv(x,z)}$ for each z.
4. By rejection sampling, we can sample a η-approximation of $h'(x)$ in at most $O(2^\ell \log(1/\delta))$ rounds, and each round takes only $O(k)$ time.

Let h^* be the circuit which uses above steps to approximate h_T. Since $\eta = O(\epsilon)$ and $h(x)$ is 2η-close to $h_T(x)$, we have

$$\mathop{\mathbb{E}}_{h,x\leftarrow X}[f(x,h(x))] - \mathop{\mathbb{E}}_{g,x\leftarrow X}[f(x,g(x))] \leq \epsilon + 2\eta = O(\epsilon)$$

for any $f \in \mathcal{F}'$. (Note that we can always rescale ϵ to make the final gap is at most ϵ.) Since the complexity of the first step dominates all other steps, h is of complexity $O(2^\ell \cdot (sT + T\log T)) = \tilde{O}(s \cdot 2^\ell \epsilon^{-2})$.

4 Lower Bound for Leakage Simulation

We have seen that there exists an MWU algorithm which combines only $O(\ell\epsilon^{-2})$ distinguishers to make a good simulator h. Besides, for every chosen distinguisher f the algorithm queries $f(x,z)$ for every $z \in \{0,1\}^\ell$ when computing $h(x)$. Therefore the algorithm makes $O(\ell 2^\ell \epsilon^{-2})$ queries in total. In the previous section, we also showed that evaluating the $O(\ell\epsilon^{-2})$ chosen distinguishers is the bottleneck of the simulation. Then a natural question arises: can we construct

a simulator which makes fewer queries? It might be possible to find a boosting procedure using fewer distinguishers, or maybe we can skip some $z \in \{0,1\}^\ell$ when querying $f(x,z)$ for some f. However, in this section we will show that the MWU approach is almost optimal: any *black-box* simulator which satisfies an independence restriction has to make $\Omega(2^\ell \epsilon^{-2})$ queries to fool the distinguishers.

4.1 Black-Box Model

To show the optimality of the MWU approach, we consider black-box simulation, which means we only use only the distinguishers as black-box and does not rely on how they are implemented. Note that all known results of leakage simulation ([JP14,Skó16a,VZ13]) are black-box. Indeed, all the leakage simulation results are in the following form: first learn a set of distinguishers $\{f_1, \ldots, f_{q'}\}$ which is common for each x, then query $f_i(x,z)$ for each $z \in \{0,1\}^\ell$ and $i \in [q']$, and finally combine them to obtain the distribution of $h(x)$. The model we consider is more general than this form, so it also rules out some other possible black-box approaches.

Definition 3 (Simulator). *Given a function $g : \{0,1\}^n \to \{0,1\}^\ell$, a distribution X over $\{0,1\}^n$ and a set $\mathcal{F}$ of functions $\{0,1\}^{n+\ell} \to \{0,1\}$, we say function $h : \{0,1\}^n \to \{0,1\}^\ell$ is an* $(\epsilon, X, \mathcal{F})$-simulator of g *if*

$$\forall f \in \mathcal{F}\ , \left| \mathop{\mathbb{E}}_{g,x\leftarrow X}[f(x,g(x))] - \mathop{\mathbb{E}}_{h,x\leftarrow X}[f(x,h(x))] \right| \le \epsilon.$$

Definition 4 (Black-Box Simulator). *Let $\ell, m, a \in \mathbb{N}$ and $\epsilon > 0$. We say an oracle-aid simulation circuit $D^{(\cdot)}$ which takes two inputs $x \in \{0,1\}^n$ and $\alpha \in \{0,1\}^a$ is a black-box (ϵ, ℓ, m, a)-simulator with query complexity q if it satisfies the follows. For every function $g : \{0,1\} \to \{0,1\}^\ell$, distribution X over $\{0,1\}^n$ and a set of distinguishers $\mathcal{F}$ with $|\mathcal{F}| \le m$, there exists $\alpha \in \{0,1\}^a$ (which we call "advice string") such that $D^{\mathcal{F}}(\cdot, \alpha)$ is an $(\epsilon, X, \mathcal{F})$-simulator for g and D uses at most q oracle gates.*

We say a black-box simulator is a same-input *black-box simulator if for every $f \in \mathcal{F}$, D only queries $f(x,\cdot)$ when computing on input x. We say a black-box simulator is* non-adaptive *if the choice of the oracle queries (including the choice of f and query input) does not depend on any response of the oracle.*

Remark 1. A reasonable range of parameters are $\epsilon^{-1}, 2^\ell, \log|\mathcal{F}| < 2^{o(n)}$ since all the simulations we know is of complexity $\mathrm{poly}(\epsilon^{-1}, 2^\ell, \log|\mathcal{F}|)$. Note that when we consider $\mathcal{F}$ to be the set of every distinguisher of size at most s, $\log|\mathcal{F}| = O(s \log s)$. Besides, we also assume $a = 2^{o(n)}$ so that the simulator cannot trivially take α as an expression of g.

The lower bound we prove in this paper is for *same-input black-box simulator*. The same-input assumption is also made in related works including [LTW11,Zha11,PS16]. See the next section for more discussions about the black-box models in related results.

It is not hard to see that all the boosting approaches we mentioned above are in this model: the advice α is of length $O(q \log |\mathcal{F}|)$ and stands for "which distinguishers should be chosen", and D queries every chosen distinguisher f with input (x, z) for every $z \in \{0,1\}^\ell$ when computing $D^{\mathcal{F},\alpha}(x)$. Moreover, these simulation algorithms are non-adaptive. We can write the MWU approach as the following corollary:

Corollary 4. *For every $0 < \epsilon < \frac{1}{2}$, $\ell, m \in \mathbb{N}$, there exists an non-adaptive same-input black-box (ϵ, ℓ, m, a)-simulator with query complexity $q = O(\ell 2^\ell \epsilon^{-2})$ and $a = \tilde{O}(q \log |\mathcal{F}|)$.*

Besides capturing all known simulators, our lower bound also rules out the adaptive approaches. Whether there exists a faster simulation not satisfying the same-input restriction is left open, but it is hard to imagine how querying different input is useful.

4.2 Main Theorem and Related Results

Theorem 3. *For every $2^{-o(n)} < \epsilon < 0.001$, $\ell = o(n)$, $\omega(2^\ell/\epsilon^3) < m < 2^{2^{o(n)}}$ and $a = 2^{o(n)}$, a same-input black-box (ϵ, ℓ, m, a)-simulator must have query complexity $q = \Omega(2^\ell \epsilon^{-2})$.*

Remark 2. For ϵ we require it to be smaller than some constant so that $\mathrm{Bern}(\frac{1}{2} + \Theta(\epsilon))$ is well defined. Besides, we also require the size of distinguisher set m to be large enough to guarantee that the simulator must "simulate" the function instead of fooling distinguishers one by one. As we saw in Remark 1, the range of parameters here is reasonable.

Before this paper, there were some lower bounds either for Leakage Simulation Lemma itself or for its implications. We classify these results by their models as follows.

– **Non-Adaptive Same-Input Black-Box Lower Bounds.** Recall that Leakage Simulation implies Hardcore Theorem and Dense Model Theorem. Lu, Tsai and Wu [LTW11] proved an $\Omega(\log(\frac{1}{\delta})/\epsilon^2)$ lower bound for query complexity in Hardcore Lemma proof where δ denotes the density of the hardcore set. By taking $\delta = \Theta(1)$ we can obtain an $\Omega(1/\epsilon^2)$ lower bound for query complexity of Leakage Simulation. Similarly, Zhang [Zha11] proved a lower bound for query complexity in Dense Model Theorem proof which implies the same $\Omega(1/\epsilon^2)$ lower bound.[9] Besides, Pietrzak and Skórski [PS16] proved a $\Omega(2^\ell/\epsilon)$ lower bound for leakage chain rule, which also implies a $\Omega(2^\ell/\epsilon)$ lower bound for Leakage Simulation. These lower bounds assume both the non-adaptivity and the independence of inputs.[10]

[9] The black-box model these results considered is more restricted. Actually, the black-box model in [LTW11] does not contain Holenstein's proof [Hol05]. Nevertheless, their proof for query lower bound also works for the model we define here.

[10] Interestingly, in the reduction from Leakage Chain Rule to Leakage Simulation, there exists a distinguisher in the reduction which only need to be queried on one

– **Non-Adaptive Black-Box Lower Bounds.** Impagliazzo [Imp95] proved that the Hardcore Lemma implies Yao's XOR Lemma [GNW95,Yao82], which is an important example of hardness amplification. Since the reduction is black-box, it is not hard to see that the $\Omega(\log(\frac{1}{\delta})/\epsilon^2)$ lower bound for hardness amplification proved by Shaltiel and Viola [SV10] is also applicable to Hardcore Lemma. Similarly, by setting $\delta = \Theta(1)$ we get a $\Omega(1/\epsilon^2)$ lower bound for Leakage Simulation. Moreover, this lower bound does not require the same-input assumption.[11] Nevertheless, the proof highly relies on non-adaptivity.
– **General Black-Box Lower Bounds.** Artemenko and Shaltiel [AS11] proved an $\Omega(1/\epsilon)$ lower bound for a simpler type of hardness amplification, and removed the non-adaptivity. Their result implies a general black-box lower bound for Leakage Simulation, but the lower bound is far from optimal.
– **Non-Black-Box Lower Bounds.** Trevisan, Tulsiani and Vadhan show that the simulator cannot be much more efficient than the distinguishers [TTV09, Remark 1.6]. Indeed, for any large enough $s \in \mathbb{N}$ they construct a function g such that any simulator h of complexity s can be distinguished from g by a distinguisher of size $\tilde{O}(ns)$. Jetchev and Pietrzak [JP14] also show an $\Omega(2^\ell \cdot s)$ lower bound under some hardness assumptions for one-way functions.

None of the existing results imply an optimal lower bound for Leakage Simulation. However, proving a lower bound for Leakage Simulation might be a simpler task, and it turns out that we can prove a lower bound of $\Omega(2^\ell \epsilon^{-2})$. The basic ideas is as follows, and would be further explained in the proof. To capture the 2^ℓ factor, for each distinguisher f and input x we hide information at $f(x, z)$ for a random z, similar to the proof in [PS16]. Then checking all z over $\{0,1\}^\ell$ is necessary. Although the claim seems trivial, the analysis would be more complicated in our adaptive model. To capture the ϵ^{-2} factor, we utilize the anti-concentration of almost uniform Bernoulli distribution $\mathrm{Bern}(\frac{1}{2} + \Theta(\epsilon))$, so that $\Omega(1/\epsilon^2)$ samples are needed to distinguish it from uniform distribution with constant probability. A similar concept can be found for example in [Fre95,LTW11,PS16]. Note that in [PS16] they only require an advantage of ϵ when distinguishing such Bernoulli distribution from uniform, which causes an $O(1/\epsilon)$ loss in complexity.

4.3 Proof of Theorem 3

Overview We would like to show that there exists a function g and a set of distinguisher $\mathcal{F}$ such that any simulator h with limited queries to $\mathcal{F}$ cannot approximate g well. Since $|\mathcal{F}|$ is much larger than the number of queries, there exist some distinguishers which can distinguish g and any bad simulator h "fairly",

adaptively chosen input. In this case non-adaptivity causes a 2^ℓ additive loss. This can be viewed as an evidence that adaptivity might be useful.

[11] Actually, such assumption is not even natural in hardness amplification.

i.e. these distinguishers are independent of h. Therefore more queries are required to successfully simulate g and fool $\mathcal{F}$. We will prove the existence of g and $\mathcal{F}$ by probabilistic argument.

To make the simulation task as hard as possible, let g be a random function. Besides, for any distinguisher $f \in \mathcal{F}$, let $f(x, z)$ be a random bit drawn from $\mathrm{Bern}(\frac{1}{2}+c_1\epsilon)$ for some constant c_1 if $z = g(x)$, or from $\mathrm{Bern}(\frac{1}{2})$ otherwise, so that a query to f provides least possible information.[12] To understand such setting, we can imagine that there exists a random oracle $\mathcal{O}$ which takes input (x, z) and only return biased bit at $z = g(x)$ for each x. Then $g(x)$ is considered as the *key* to the oracle, and our goal is to find out the correct key. Each $f \in \mathcal{F}$ can be viewed as a collection of samples from the oracle with certain randomness. Intuitively, since $f(x, g(x))$ is only $\Theta(\epsilon)$ away from uniform, f can distinguish g and any bad simulator h which does not approximate g with constant probability. To approximate g well, we need to test all 2^ℓ keys to find the correct one. Besides, it requires $\Omega(1/\epsilon^2)$ samples to distinguish $\mathrm{Bern}(\frac{1}{2} + \Theta(\epsilon))$ and $\mathrm{Bern}(\frac{1}{2})$ with constant probability, so $\Omega(1/\epsilon^2)$ queries are required for each key to make sure we can distinguish the real key from other fake keys. Therefore a successful simulator h should make at least $\Omega(\epsilon^{-2}2^\ell)$ queries.

Now we proceed to the formal proof. Assume for contradiction that D is a black-box (ϵ, ℓ, m, a)-simulator with query complexity $q \le c_2(2^\ell\epsilon^{-2})$, where $c_2 = \frac{1}{360000}$. Let $g : \{0,1\}^n \to \{0,1\}^\ell$ be a random function such that for every $x \in \{0,1\}^n$, $g(x)$ is chosen uniformly at random from $\{0,1\}^\ell$. Let $\mathcal{F}$ be a set of random function defined in previous paragraph, and we specify that $c_1 = 30$. First we prove that given any fixed advice string α, the decision function $D^{\mathcal{F}}(, \alpha)$ cannot guess g correctly with high enough probability over the choice of $\mathcal{F}$ and g.

Lemma 5. *Fix α and let $h = D^{\mathcal{F},\alpha}$. For any $x \in \{0,1\}^n$, we have $\Pr[h(x) = g(x)] \le 1 - \frac{3}{c_1}$, where the probability is taken over the choice of $g(x)$, $f(x, \cdot)$ for every $f \in \mathcal{F}$ (abbreviated as $\mathcal{F}(x)$), and the randomness of h.*

Proof. Without loss of generality, assume that h has no randomness other than oracle queries. (We can obtain the same bound for probabilistic h by taking average over deterministic circuits.) We also assume that h always make q different queries by adding dummy queries.

Consider h as a decision tree where queries are the nodes and different answers represent different branches. For every fixed $g(x)$ and $\mathcal{F}(x)$, the computation of $h(x)$ corresponds to a root-to-leaf path denoted as $t = \{a_1, \ldots, a_q\}$ where a_i is the answer to the i-th query, and we call t *transcript*. Let T be a random variable over $\{0,1\}^q$ which represents such transcript. Note that the output of $h(x)$ is uniquely determined by its transcript. Let $Dec : \{0,1\}^q \to \{0,1\}^\ell$ denote the corresponding decision function from transcript to output. Then we have

$$\Pr[h(x) = g(x)] = \Pr[Dec(T) = g(x)] = \sum_{t,k} \Pr[T = t, g(x) = k, Dec(t) = k].$$

[12] Note that $\mathcal{F}$ should be able to distinguish g from easy functions with advantage ϵ, otherwise the simulation is trivial.

To prove the upper bound for $\Pr[h(x) = g(x)]$, first we consider an ideal case such that each function in $\mathcal{F}$ is an uniform random function. In this case, for every $(t, k) \in \{0,1\}^q \times \{0,1\}^\ell$, $\Pr[T^* = t, g(x) = k] = 2^{-(q+\ell)}$ where T^* is the ideal transcript, i.e., uniform distribution over $\{0,1\}^q$. Since for each t there exists a unique k where $Dec(t) = k$, only 2^q pairs (t, k) are *correct* (i.e. $Dec(t) = k$). In such ideal case, we have $\Pr[h^*(x) = g(x)] = 2^{-\ell}$ where h^* denotes the ideal variant of h. In the real case, $\Pr[T = t, g(x) = k]$ can be at most $2^{-\ell}(\frac{1}{2} + c_1\epsilon)^q$, in the case that h queries with correct key in every query and all the responses are 1. However, there does not exist too many extreme cases like this. Besides, we have seen that most of the pairs (t, k) over $\{0,1\}^q \times \{0,1\}^\ell$ do not satisfy $Dec(t) = k$. Therefore we can expect that a large fraction of pairs are *normal* (i.e. chosen with probability $\Theta(2^{-(q+\ell)})$) and *wrong* (i.e. $Dec(t) \neq k$). Such statement implies a lower bound for $\Pr[h(x) \neq g(x)]$.

Next we formally prove the statement above. Consider any transcript $t = \{a_1, a_2, \ldots, a_q\}$. Recall that the queries made by h are uniquely determined by t: the first query is fixed, the second query is determined by the first bit of t, and so on. Let $\{z_1, z_2, \ldots, z_q\}$ be the sequence of key such that the i-th query is $f_i(x, z_i)$ for some $f_i \in \mathcal{F}$. For any $k \in \{0,1\}^\ell, t \in \{0,1\}^q$, let u_i denote the index of the i-th *useful query*, which means the i-th index satisfying $z_{u_i} = k$. Then we define $N_b(t, k) = \sum_i [a_{u_i} = b]$ for $b \in \{0,1\}$, which represents the number of useful queries with response b. Besides, let $N(t, k) = N_0(t, k) + N_1(t, k)$ and $N_\Delta(t, k) = N_0(t, k) - N_1(t, k)$. Similarly, for $j \leq N(t, k)$, we define $N_b(t, k, j) = \sum_{i=1}^{j} [a_{u_i} = b]$ for $b \in \{0,1\}$ and $N_\Delta(t, k, j) = N_0(t, k, j) - N_1(t, k, j)$, which only consider the first j useful queries. Recall that for any $f \in \mathcal{F}$, $f(x, z)$ is uniform when $z \neq g(x)$ and biased when $z = g(x)$. For any fixed (t, k),

$$\begin{aligned}
\Pr[g(x) = k, T = t] &= \left(\frac{1}{2}\right)^{(\ell+q-N(t,k))} \left(\frac{1}{2} - c_1\epsilon\right)^{N_0(t,k)} \left(\frac{1}{2} + c_1\epsilon\right)^{N_1(t,k)} \\
&= \left(\frac{1}{2}\right)^{(\ell+q)} (1 - 2c_1\epsilon)^{N_\Delta(t,k)} \left(1 - 4c_1^2\epsilon^2\right)^{N_1(t,k)} \\
&\geq \left(\frac{1}{2}\right)^{(\ell+q)} (1 - 2c_1\epsilon)^{N_\Delta(t,k)} \left(1 - 4c_1^2\epsilon^2\right)^{N(t,k)} \qquad (3)
\end{aligned}$$

Therefore a pair (t, k) is normal if $N_\Delta(t, k) = O(1/\epsilon)$ and $N(t, k) = O(1/\epsilon^2)$. We claim that a large enough fraction of pairs over $\{0,1\}^q \times \{0,1\}^\ell$ are wrong and normal as following:

Claim. Let $q' = 5q/2^\ell \leq 5c_2\epsilon^{-2}$. Then for at least $\frac{1}{5}$ fraction of pairs (t, k) over $\{0,1\}^q \times \{0,1\}^\ell$ satisfies the following conditions:

1. $Dec(t) \neq k$.
2. $N(t, k) < q'$.
3. $N_\Delta(t, k) < \sqrt{5q'}$.

Proof. We will prove upper bounds for correct pairs and extreme cases to make sure a large fraction of normal and wrong pairs are left. More precisely, we prove upper bound for the contrary of each condition one by one.

1. Only $2^{-\ell}$ of pairs are correct:
   This obviously holds because (t, k) is correct only when $Dec(t) = k$.
2. At most $\frac{1}{5}$ of pairs (t, k) satisfy $N(t, k) \geq q'$:
   For any t we have $\mathbb{E}_{k \leftarrow U_\ell}[N(t, k)] = \frac{q}{2^\ell}$. By Markov's inequality, at most $\frac{q}{2^\ell q'} = \frac{1}{5}$ of pairs satisfy $N(t, k) \geq q'$.
3. For at most $\frac{1}{10}$ of pairs (t, k), $N(t, k) < q'$ and $N_\Delta(t, k) > \sqrt{5q'}$:
   Fix k. Let T^* be a random transcript which is uniform over $\{0, 1\}^q$. Consider a sequence of random variable $\{Y_j\}$ depending on T^* such that

$$Y_j = \begin{cases} N_\Delta(T^*, k, j), & \text{if } j < N(T^*, k) \\ N_\Delta(T^*, k), & \text{otherwise.} \end{cases}$$

   It's not hard to see that $\{Y_i\}$ is a martingale with difference at most 1. By Azuma's inequality, we have $\Pr[Y_{q'} \geq \sqrt{5q'}] \leq e^{-5q'/2q'} < 0.1$. Since T^* is uniform, the statement above is the same as saying for at most 0.1 fraction of $t \in \{0, 1\}^q$, $Y_{q'}(t) \geq \sqrt{5q'}$. When restricted to t satisfying $N(t, k) < q'$ we have $N_\Delta(t, k) = Y_{q'}(t) \geq \sqrt{5q'}$.

By union bound, all three conditions in the claim hold simultaneously for at least $\frac{1}{5}$ of pairs over $\{0, 1\}^q \times \{0, 1\}^\ell$.

Now consider any pair (t, k) which satisfies condition 2 and 3 in the claim above, in other word a *normal* pair. By inequality (3), we have

$$\begin{aligned}
\Pr[g(x) = k, T = t] &\geq (1/2)^{(\ell+q)} (1 - 2c_1\epsilon)^{N_\Delta(t,k)} (1 - 4c_1^2\epsilon^2)^{N(t,k)} \\
&\geq (1/2)^{(\ell+q)} (1 - 2c_1\epsilon)^{\sqrt{5q'}} (1 - 4c_1^2\epsilon^2)^{q'} \qquad (4) \\
&= (1/2)^{(\ell+q)} (1 - 2c_1\epsilon)^{5\sqrt{c_2}\epsilon^{-1}} (1 - 4c_1^2\epsilon^2)^{5c_2\epsilon^{-2}} \\
&\geq (1/2)^{(\ell+q)} (0.3)^{10c_1\sqrt{c_2}} (0.3)^{20c_1^2c_2} \qquad (5) \\
&\geq (1/2)^{(\ell+q)} \cdot 0.5 \qquad (6)
\end{aligned}$$

The inequality (5) holds because $(1 - \delta)^{1/\delta} \geq 0.3$ for any $0 < \delta \leq 0.1$. Since $\frac{1}{5}$ of pairs satisfy the conditions above, we have

$$\Pr[h(x) \neq g(x)] = \sum_{k,t} [g(x) = k, T = t, Dec(t) \neq k] \geq 0.1. \qquad (7)$$

Therefore $\Pr[h(x) = g(x)] \leq 0.9 = 1 - \frac{3}{c_1}$.

With the lemma above, we can finish the proof simply with a concentration bound and probabilistic method. Consider the probabilistic distinguisher f_R which is a uniform distribution over all distinguishers in $\mathcal{F}$. Fix any advice α and consider $h(\cdot) = D^{\mathcal{F}}(\cdot, \alpha)$. For any $x \in \{0, 1\}^n, f \in \mathcal{F}$ such that f is not queried by $h(x)$, we have $\mathbb{E}[f(x, h(x))] = \frac{1}{2} + \Pr[h(x) = g(x)] \cdot c_1\epsilon$ by definition of f. Since h makes at most q query when computing $h(x)$, f_R chooses a query

coincident with queries in h with probability $\frac{q}{m}$. Even in the worst case that f_R returns 1 in all these cases, we still have

$$\mathbb{E}\left[f_R(x, g(x))\right] \leq \frac{1}{2} + \Pr[g(x) = h(x)] \cdot c_1\epsilon + \frac{q}{m} \tag{8}$$

$$\leq \frac{1}{2} + (c_1 - 2)\epsilon \tag{9}$$

because m is large enough. Also we have $\mathbb{E}\left[f_R(x, g(x))\right] = \frac{1}{2} + c_1\epsilon$ by definition. Therefore, $\mathbb{E}\left[f_R(x, g(x)) - f_R(x, h(x))\right] \geq 2\epsilon$. Let X be the uniform distribution. Note that for different x, $g(x)$ and $\mathcal{F}(x)$ are chosen independently. Therefore $\mathbb{E}_h\left[f_R(x, g(x)) - f_R(x, h(x))\right]$[13] for each x are independent random variables since it is only influenced by randomness of $g(x)$ and $\mathcal{F}(x)$. By Chernoff-Hoeffding bound, $\mathbb{E}_{x \leftarrow X}[f_R(x, g(x)) - f_R(x, h(x))] < \epsilon$ holds with probability $2^{-\Omega(\epsilon^2 2^n)}$ over the choice of $\mathcal{F}$ and g. By taking union bound over α, we have

$$\forall \alpha \in \{0,1\}^{2^{o(n)}} \ , \ \mathop{\mathbb{E}}_{x \leftarrow X}\left[f_R(x, g(x)) - f_R(x, D^{\mathcal{F}}(x, \alpha))\right] \leq \epsilon \tag{10}$$

with probability $2^{-\Omega(\epsilon^2 2^n) + 2^{o(n)}}$, which is less than 1 for large enough n. By the probabilistic argument there exists a function g and a set $\mathcal{F}$ such that

$$\mathop{\mathbb{E}}_{x \leftarrow X}[f_R(x, g(x)) - f_R(x, D^{\mathcal{F}}(x, \alpha))] > \epsilon. \tag{11}$$

By averaging argument, for any α, there exists $f \in \mathcal{F}$ such that f can distinguish $(X, D^{\mathcal{F}}(X, \alpha))$ and $(X, g(X))$. Therefore the simulation fails no matter what α is, which contradicts to our assumption. Thus there is no simulator with query complexity $c_2(2^\ell \epsilon^{-2})$.

To summarize, we proved an $\Omega(2^\ell \epsilon^{-2})$ lower bound for black-box (ϵ, ℓ, k, a)-simulator, while the upper bound is only $O(\ell 2^\ell \epsilon^{-2})$. Note that in order to apply Chernoff bound, we need the same-input assumption (i.e. $D(x)$ cannot query $\mathcal{F}(x')$ for $x' \neq x$) to guarantee the independence of different x, even though querying with different input seems useless. A general black-box tight lower bound is left for future work.

References

AHK12. Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

AS11. Sergei Artemenko and Ronen Shaltiel. Lower bounds on the query complexity of non-uniform and adaptive reductions showing hardness amplification. In Leslie Ann Goldberg, Klaus Jansen, R. Ravi, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 14th International Workshop, APPROX 2011,*

[13] The expectation is taken over the local randomness of h, which does not need to be considered in the probabilistic argument.

and 15th International Workshop, RANDOM 2011, Princeton, NJ, USA, August 17-19, 2011. Proceedings, volume 6845 of *Lecture Notes in Computer Science*, pages 377–388. Springer, 2011.

CLP15. Kai-Min Chung, Edward Lui, and Rafael Pass. From weak to strong zero-knowledge and applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, volume 9014 of *Lecture Notes in Computer Science*, pages 66–92. Springer, 2015.

DBL08. *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*. IEEE Computer Society, 2008.

DP08. Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA* [DBL08], pages 293–302.

FK99. Alan M. Frieze and Ravi Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999.

Fre95. Yoav Freund. Boosting a weak learning algorithm by majority. *Inf. Comput.*, 121(2):256–285, 1995.

FS96. Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In Avrim Blum and Michael Kearns, editors, *Proceedings of the Ninth Annual Conference on Computational Learning Theory, COLT 1996, Desenzano del Garda, Italy, June 28-July 1, 1996.*, pages 325–332. ACM, 1996.

GNW95. Oded Goldreich, Noam Nisan, and Avi Wigderson. On yao's xor-lemma. *Electronic Colloquium on Computational Complexity (ECCC)*, 2(50), 1995.

GW11. Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 99–108. ACM, 2011.

HILL99. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

Hol05. Thomas Holenstein. Key agreement from weak bit agreement. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 664–673. ACM, 2005.

HS16. Martin Hirt and Adam D. Smith, editors. *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, volume 9985 of *Lecture Notes in Computer Science*, 2016.

Imp95. Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, 23-25 October 1995*, pages 538–545. IEEE Computer Society, 1995.

JP14. Dimitar Jetchev and Krzysztof Pietrzak. How to fake auxiliary input. In Yehuda Lindell, editor, *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014.*

Proceedings, volume 8349 of *Lecture Notes in Computer Science*, pages 566–590. Springer, 2014.

LTW11. Chi-Jen Lu, Shi-Chun Tsai, and Hsin-Lung Wu. Complexity of hard-core set proofs. *Computational Complexity*, 20(1):145–171, 2011.

PS16. Krzysztof Pietrzak and Maciej Skórski. Pseudoentropy: Lower-bounds for chain rules and transformations. In Hirt and Smith [HS16], pages 183–203.

Rou16. Tim Roughgarden. *No-Regret Dynamics*, pages 230–246. Cambridge University Press, 2016.

RTTV08. Omer Reingold, Luca Trevisan, Madhur Tulsiani, and Salil P. Vadhan. Dense subsets of pseudorandom sets. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA* [DBL08], pages 76–85.

Skó16a. Maciej Skórski. Simulating auxiliary inputs, revisited. In Hirt and Smith [HS16], pages 159–179.

Skó16b. Maciej Skórski. A subgradient algorithm for computational distances and applications to cryptography. *IACR Cryptology ePrint Archive*, 2016:158, 2016.

SV10. Ronen Shaltiel and Emanuele Viola. Hardness amplification proofs require majority. *SIAM J. Comput.*, 39(7):3122–3154, 2010.

TTV09. Luca Trevisan, Madhur Tulsiani, and Salil P. Vadhan. Regularity, boosting, and efficiently simulating every high-entropy distribution. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 126–136. IEEE Computer Society, 2009.

VZ13. Salil P. Vadhan and Colin Jia Zheng. A uniform min-max theorem with applications in cryptography. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 93–110. Springer, 2013.

Yao82. Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 80–91. IEEE Computer Society, 1982.

Zha11. Jiapeng Zhang. On the query complexity for showing dense model. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:38, 2011.