Impossibility of Strong KDM Security with Auxiliary Input

Cody Freitag* Ilan Komargodski† Rafael Pass‡

March 13, 2019

Abstract

In this note, we show that a strong notion of KDM security cannot be obtained by any encryption scheme in the auxiliary input setting, assuming Learning With Errors (LWE) and one-way permutations. The notion of security we deal with guarantees that for any (possibly inefficient) function f, it is computationally hard to distinguish between an encryption of $\vec{0}$ and an encryption of $f(\mathsf{pk}, z)$, where pk is the public key and z is the auxiliary input. Furthermore, we show that this holds even when restricted to bounded-length auxiliary input where z is much shorter than pk under the additional assumption that (non-leveled) fully homomorphic encryption exists.

1 Introduction

An encryption scheme is said to be key-dependent message (KDM) secure if it is secure even against adversaries who have access to encryptions of messages that depend on the secret key. This notion captures settings where there might be correlations between the secret key and the encrypted messages. Since its introduction by Black et al. [BRS02], this notion has been extensively studied and many different definitions have been proposed and used.

In this note, we address a strong notion of KDM security with auxiliary input and show that no encryption scheme can satisfy it under widely believed cryptographic assumptions (Learning With Errors and one-way permutations). At a high level, our notion of security, called *strong KDM security with auxiliary input*, requires that, for a distribution D used to (maliciously) sample public keys together with some auxiliary input, if

$$\{(z, \mathsf{pk}^*) \leftarrow D(1^\lambda) : (z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(0))\}_{\lambda\in\mathbb{N}} \approx_c \{(z, \mathsf{pk}^*) \leftarrow D(1^\lambda) : (z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(1))\}_{\lambda\in\mathbb{N}},$$

then for every (not necessarily efficient) function f such that $f(\mathsf{pk}^*, z) \in \{0,1\}^s$ for $s \in \mathsf{poly}(\lambda)$,

$$\{(z, \mathsf{pk}^*) \leftarrow D(1^\lambda) : (z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(0^s))\}_{\lambda\in\mathbb{N}} \approx_c \{(z, \mathsf{pk}^*) \leftarrow D(1^\lambda) : (z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(f(\mathsf{pk}^*, z)))\}_{\lambda\in\mathbb{N}}.$$

Our main result is that the above notion of security cannot exist under Learning with Errors (LWE) and one-way permutations. More generally, our result can be instantiated with any implementation of a *compute-and-compare* obfuscation (introduced and constructed by Goyal, Koppula,

*Cornell Tech, cfreitag@cs.cornell.edu
†Cornell Tech, komargodski@cornell.edu
‡Cornell Tech, rafael@cs.cornell.edu

and Waters [GKW17] and by Wichs and Zirdelis [WZ17]) for a specific high pseudo-entropy distribution. We use the construction of [GKW17, WZ17] of such an obfuscator from LWE and use one-way permutations to create the aforementioned distribution.

Theorem 1. *Assuming LWE and one-way permutations, no semantically secure encryption scheme satisfies strong KDM security with auxiliary input.*

Furthermore, we show that even if the auxiliary input z might be short, even sublinear in the size of pk, we can use *succinct* compute-and-compare obfuscation to break the assumption. Such an obfuscator is constructed in [WZ17] from (non-leveled) fully homomorphic encryption, which exists under LWE plus an additional circular security assumption.

Remark 1. *In a recent work, Deshpande and Kalai [DK18a] use this notion of strong KDM security with auxiliary input to construct a 2-message witness hiding protocol. In a preliminary version [DK18a], they define the notion without auxiliary input, but their proof of witness hiding implicitly relies on strong KDM security with respect to auxiliary input. We have communicated a preliminary version of this note to the authors of [DK18a], and they have acknowledged the issue with their definition. In a follow-up [DK18b], they show that considering short auxiliary input suffices for their result. However, as we show in this note, even strong KDM security with short auxiliary input is impossible, assuming standard cryptographic assumptions.*

1.1 Related Work

Brzuska and Mittelbach [BM14] show the impossibility of multi-bit point obfuscation with auxiliary input, assuming indistinguishability obfuscation. Canetti, Kalai, Varia, and Wichs [CKVW10] show that symmetric encryption schemes with strong KDM security plus an additional "wrong-key detection" property imply multi-bit point obfuscation. This strong additional property requires that if a ciphertext is encrypted under one key, then with high probability, it is an invalid ciphertext under any other key. Together, these results rule out strong KDM security with auxiliary input for semantically secure encryption schemes that also satisfy wrong-key detection assuming indistinguishability obfuscation. Our result rules out the possibility of strong KDM security with auxiliary input for *all* semantically secure encryption schemes instead assuming LWE and one-way permutations.

Following [BM14], Bellare, Stepanovs, and Tessaro [BST16] show the impossibility of key-message leakage-resilient (KM-LR) symmetric encryption assuming indistinguishability obfuscation. KM-LR requires semantic security to hold as long as the key is computationally unpredictable given the auxiliary input, which may depend on both the key and the encrypted message. While the settings are different, our impossibility rules out KM-LR security as well.

Canetti, Chen, Reyzin, and Rothblum [CCRR18] define and give candidate encryption schemes for a similar notion of strong KDM security for symmetric key encryption schemes, but their notion does not consider auxiliary input. As such, it is not ruled out by our impossibility.

2 Preliminaries

A function μ is negligible if for every polynomial p and all sufficiently large $\lambda \in \mathbb{N}$, $\mu(\lambda) \leq 1/p(\lambda)$. A probabilistic, polynomial time (PPT) algorithm $\mathcal{A}$ is a Turing machine with access to an infinite random tape that on input x halts in time $p(|x|)$ for some polynomial p. A non-uniform PPT

algorithm $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$ also receives polynomial-size non-uniform advice $z = z(\lambda) \in \{0,1\}^{\mathsf{poly}(\lambda)}$. We can equivalently consider each $\mathcal{A}_\lambda$ as a circuit of polynomial size. For a probabilistic algorithm $\mathcal{A}$, we write $\mathcal{A}(x;r)$ to denote running $\mathcal{A}$ on input x with a fixed random tape r, and when r is not provided, we assume that it is generated from a uniform distribution. Without loss of generality, we assume that for a non-uniform PPT algorithm $\mathcal{A}_\lambda$ receives 1^λ as its first input for all $\lambda \in \mathbb{N}$.

2.1 Computational Indistinguishability and Pseudo-Entropy

An ensemble is a sequence $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ where for each $\lambda \in \mathbb{N}$, X_λ is a probability distribution over $\{0,1\}^*$. Let X be a probability distribution and $\mathcal{A}$ be an algorithm, then we write $\mathcal{A}(X)$ to denote the probability distribution formed by first drawing $x \leftarrow X$ and outputting $\mathcal{A}(x)$. We define the computational indistinguishability of two ensembles as follows.

Definition 1 (Computational Indistinguishability)**.** Let $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ be ensembles. We say that X and Y are *computationally indistinguishable*, written $X \approx_c Y$, if for every non-uniform PPT algorithm $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$ there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$,

$$|\Pr[\mathcal{A}_\lambda(1^\lambda, X_\lambda) = 1] - \Pr[\mathcal{A}_\lambda(1^\lambda, Y_\lambda) = 1]| \leq \mu(\lambda).$$

We also define what it means for an ensemble X to have "pseudo-entropy" conditioned on Y. Informally, we say that X has high pseudo-entropy conditioned on Y if there exists an ensemble X' that is computationally indistinguishable from X such that X' has high min-entropy conditioned on Y. The conditional min-entropy of two random variables X and Y, denoted $H_\infty(X \mid Y)$, is defined as follows,

$$H_\infty(X \mid Y) = -\log\left(\mathrm{E}_{y\leftarrow Y}\left[\max_x \Pr[X = x \mid Y = y]\right]\right).$$

Definition 2 (Conditional (HILL) Pseudo-Entropy [HILL99, HLR07])**.** Let $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ be (possibly dependent) ensembles. We say that X *has $\ell(\lambda)$-pseudo-entropy conditioned on Y*, denoted by $H_{\mathsf{HILL}}(X \mid Y) \geq \ell(\lambda)$, if there exists some $X' = \{X'_\lambda\}_{\lambda\in\mathbb{N}}$ jointly distributed with Y such that $(X, Y) \approx_c (X', Y)$ and for all $\lambda \in \mathbb{N}$, $H_\infty(X'_\lambda \mid Y_\lambda) \geq \ell(\lambda)$.

Furthermore, when it is clear from context, we may say that a random variable X_λ has $\ell(\lambda)$-pseudo-entropy condition on Y_λ (which holds only for sufficiently large $\lambda \in \mathbb{N}$) when the associated ensembles X and Y satisfy $H_{\mathsf{HILL}}(X \mid Y) \geq \ell(\lambda)$.

2.2 One-way Functions and Pseudo-Random Generators

A one-way function is a function that can be computed easily but is hard to invert for a random input, defined formally as follows.

Definition 3. A polynomial-time computable function $f\colon \{0,1\}^* \to \{0,1\}^*$ is a *one-way function* if for every non-uniform PPT algorithm $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$,

$$\Pr[x \leftarrow \{0,1\}^\lambda; y \leftarrow f(x) : f(\mathcal{A}_\lambda(1^\lambda, y)) = y] \leq \mu(\lambda).$$

Given a one-way function f, a hard-core predicate h for f is a function that outputs a single bit $h(x)$ that is hard to predict given only $f(x)$.

Definition 4. A predicate $h\colon \{0,1\}^\lambda \to \{0,1\}$ is a *hard-core predicate for* f if h is polynomial-time computable, and for every non-uniform PPT algorithm $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$,

$$\Pr[x \leftarrow \{0,1\}^\lambda : \mathcal{A}_\lambda(1^\lambda, f(x)) = h(x)] \leq 1/2 + \mu(\lambda).$$

Goldreich and Levin [GL89] construct a hard-core predicate for every one-way function.

A pseudo-random generator (PRG) is an efficiently computable function that is expanding and whose output is computationally indistinguishable from uniform random bits. It is well known that the existence of one-way functions imply the existence of pseudo-random generators [HILL99].

Definition 5 (Pseudo-Random Generator (PRG))**.** Let $m\colon \mathbb{N} \to \mathbb{N}$ be a function such that $m(\lambda) > \lambda$ for all $\lambda \in \mathbb{N}$. An efficiently computable function $G\colon \{0,1\}^* \to \{0,1\}^*$ is a *pseudo-random generator for length* m if

$$\{x \leftarrow U_\lambda : G(x)\}_{\lambda\in\mathbb{N}} \approx_c \{U_{m(\lambda)}\}_{\lambda\in\mathbb{N}}.$$

2.3 Semantically Secure Encryption

A public-key encryption scheme consists of key generation, encryption, and decryption algorithms. At a high level, semantic security guarantees that anything that can be learned (by a non-uniform PPT algorithm) about a plaintext given an encryption of the plaintext can be learned without the encryption. Goldwasser and Micali [GM84] first introduced this security notion and showed that it is equivalent to an indistinguishability-based notion, which guarantees that the encryptions of any pair of messages are computationally indistinguishable. We use the indistinguishability-based notion, defined formally as follows.

Definition 6. A semantically secure public-key encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies the following:

- **Correctness:** For every message $m \in \{0,1\}^*$ and $\lambda \in \mathbb{N}$,

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

- **Semantic security:** For ever pair of messages $m, m' \in \{0,1\}^*$ such that $|m| = |m'|$,

$$\{(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^\lambda) : (\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(m))\}_{\lambda\in\mathbb{N}} \approx_c \{(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(1^\lambda) : (\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(m'))\}_{\lambda\in\mathbb{N}}.$$

2.4 Compute-and-Compare Obfuscation

Compute-and-compare obfuscation (also known as lockable obfuscation) was first defined and constructed concurrently by [WZ17, GKW17]. A compute-and-compare program $\mathsf{CC}[f, u]$ has hard coded a function f and a target value u. $\mathsf{CC}[f, u](x)$ outputs 1 if $f(x) = u$ and 0 otherwise. A compute-and-compare obfuscator $\mathcal{O}$ is an efficient algorithm that takes as input a compute-and-compare program $\mathsf{CC}[f, u]$ and outputs an obfuscated circuit $\widetilde{\mathsf{CC}}$ that satisfies distributional indistinguishability for specified class of distributions $\mathcal{D}$. We define this formally as follows.

Definition 7 (Compute-and-compare Obfuscation)**.** A compute-and-compare obfuscator $\mathcal{O}$ for a class of distributions $\mathcal{D}$ is a PPT algorithm that satisfies:

1. **Correctness:** for any circuit $f\colon \{0,1\}^n \to \{0,1\}^m$ and $u \in \{0,1\}^m$, there exists a negligible function ν such that

$$\Pr[\widetilde{\mathsf{CC}} \leftarrow \mathcal{O}(1^\lambda, \mathsf{CC}[f,u]) : \widetilde{\mathsf{CC}}(x) = \mathsf{CC}[f,u](x)] \geq 1 - \nu(x);$$

2. **Simulation:** there exists a simulator Sim such that for every distribution $D \in \mathcal{D}$ where $(z,f,u) \leftarrow D(1^\lambda)$, it holds that

$$\{(z,f,u) \leftarrow D(1^\lambda) : (z, \mathcal{O}(1^\lambda, \mathsf{CC}[f,u]))\}_{\lambda\in\mathbb{N}} \approx_c \{(z,f,u) \leftarrow D(1^\lambda) : (z, \mathsf{Sim}(1^\lambda, 1^\ell))\}_{\lambda\in\mathbb{N}},$$

where f is an ℓ-size circuit for $\ell \in \mathsf{poly}(\lambda)$.

Wichs and Zirdelis [WZ17] construct compute-and-compare obfuscation for polynomial-time samplable distributions D where $(z,f,u) \leftarrow D(1^\lambda)$ such that D is in the class of $\alpha(\cdot)$-psuedo-entropy distributions $\mathcal{D}_{\alpha\text{-pe}}$, where $H_{\mathsf{HILL}}(u \mid z, f) \geq \alpha(\lambda)$, They show how to construct compute-and-compare obfuscation for $\mathcal{D}_{\lambda^\epsilon\text{-pe}}$ for any $\epsilon > 0$ under LWE.

Let f be a circuit of size t with depth d. We say that a compute-and-compare obfuscator $\mathcal{O}$ is *succinct* if $|\mathcal{O}(1^\lambda, \mathsf{CC}[f,u])| \in \mathsf{poly}(\lambda, |f|, |u|, \log t)$ and is *weakly succinct* if $|\mathcal{O}(1^\lambda, \mathsf{CC}[f,u])| \in \mathsf{poly}(\lambda, |f|, |u|, \log t, d)$. We may also consider compute-and-compare obfuscation for Turing machines where $t(n)$ is the bound on the running time for inputs of length n. In [WZ17], they show how to use (non-leveled) FHE for Turing machines (which exists under LWE plus additional circular security assumptions [GSW13]) to achieve succinct compute-and-compare obfuscation. Relying on only leveled FHE (known from LWE), they also give a *weakly succinct* obfuscator.

Finally, we note that compute-and-compare obfuscation implies one-way functions [BGI+12, KMN+14] and hence pseudo-random generators [HILL99].

3 KDM Security

We consider a definition of KDM security where a distribution D (maliciously) samples auxiliary input in addition to a public key for *any* semantically secure encryption scheme. This definition with auxiliary input captures how KDM security is often used when composed in applications.

Definition 8 (Strong KDM Security with Auxiliary Input)**.** A semantically secure public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be *strong KDM secure with auxiliary input* if for every efficiently computable (by a non-uniform PPT algorithm) distribution D used to (maliciously) sample public keys and auxiliary input it holds that if

$$\{(z, \mathsf{pk}^*) \leftarrow D(1^\lambda) : (z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(0))\}_{\lambda\in\mathbb{N}} \approx_c \{(z, \mathsf{pk}^*) \leftarrow D(1^\lambda) : (z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(1))\}_{\lambda\in\mathbb{N}},$$

then for every (not necessarily efficient) function f such that $f(\mathsf{pk}^*, z) \in \{0,1\}^s$ for $s \in \mathsf{poly}(\lambda)$,

$$\{(z, \mathsf{pk}^*) \leftarrow D(1^\lambda) : (z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(0^s))\}_{\lambda\in\mathbb{N}} \approx_c \{(z, \mathsf{pk}^*) \leftarrow D(1^\lambda) : (z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(f(\mathsf{pk}^*, z)))\}_{\lambda\in\mathbb{N}}.$$

We lastly consider a more general definition where the auxiliary input z is restricted to be bounded by some arbitrary function α of the relevant parameters. Specifically, we say that a semantically secure public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be *strong KDM secure with $\alpha(\cdot)$-bounded auxiliary input* if the above definition holds when D is restricted to outputting auxiliary input z such that $|z| \leq \alpha(\lambda)$. We recover the case of no auxiliary input when $\alpha = 0$ and Definition 8 when $\alpha = \infty$, i.e., unbounded auxiliary input.

4 Breaking KDM Security with Auxiliary Input

We show that strong KDM security with auxiliary input can be generically broken for any semantically secure encryption scheme assuming the existence of compute-and-compare obfuscation for class of λ-pseudo-entropy distributions $\mathcal{D}_{\lambda\text{-pe}}$. Specifically, for any semantically secure encryption scheme, we construct a distribution D outputting a public key pk^* and auxiliary input z such that for some function f, z makes it possible to distinguish encryptions of 0^s and $f(\mathsf{pk}^*, z)$ but not encryptions of 0 and 1.

Theorem 2 (Restatement of Theorem 1)**.** *Assuming LWE and one-way permutations, no semantically secure encryption scheme satisfies strong KDM security with auxiliary input.*

Proof. Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be any semantically secure encryption scheme and $\mathcal{O}$ be a compute-and-compare obfuscator for λ-pseudo-entropy distributions of [WZ17] (which exists based on LWE). Let $G\colon \{0,1\}^\lambda \to \{0,1\}^\lambda \times \{0,1\}^\lambda$ be a PRG that satisfies the following two properties:

1. *Uniqueness:* There exists a (possibly inefficient) function g such that
$$\Pr[s \leftarrow \{0,1\}^\lambda; (y_1, y_2) = G(s) : g(y_2) = y_1] = 1.$$
2. *Indistinguishability:* The following two ensembles are computationally indistinguishable:
$$\{(y_1, y_2)\}_{\lambda\in\mathbb{N}} \approx_c \{r \leftarrow \{0,1\}^\lambda : (r, y_2)\}_{\lambda\in\mathbb{N}},$$
where $(y_1, y_2) \leftarrow G(s)$ for $s \leftarrow \{0,1\}^\lambda$.

Given such a PRG G, it immediately follows that y_1 has λ-pseudo-entropy conditioned on y_2 where $(y_1, y_2) = G(s)$ for $s \leftarrow \{0,1\}^\lambda$. Specifically, $H_{\mathsf{HILL}}(y_1 \mid y_2) \geq \lambda$ since by assumption y_1 is indistinguishable from U_λ given y_2 and $H_\infty(U_\lambda \mid y_2) = \lambda$.

We note that the standard construction of a PRG from any one-way permutation and hardcore predicate satisfies this notion. Specifically, let P be a one-way permutation with hard-core bit h, and let $P^{(i)}$ be the composition of P i times. We define

$$G(x) = (h(x) \,||\, h(P(x)) \,||\, \ldots \,||\, h(P^{(\lambda-1)}(x)), P^{(\lambda)}(x)) = (y_1, y_2).$$

Uniqueness follows since P is a permutation and hence $P^{(\lambda)}$ has a unique preimage that can be used to compute y_1. Using a standard hybrid argument, indistinguishability follows by the security of the hard-core predicate h and since $P^{(i)}$ is a one-way permutation for all $i \in \mathbb{N}$.

Using the above ingredients, we consider the following distribution D.

$D(1^\lambda)$:

1. Sample $s \leftarrow \{0,1\}^\lambda$, $(y_1, y_2) = G(s)$, and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$.
2. Compute $\widetilde{\mathsf{CC}} \leftarrow \mathcal{O}(1^\lambda, \mathsf{CC}[\mathsf{Dec}_{\mathsf{sk}}, y_1])$.
3. Let $z = (y_2, \widetilde{\mathsf{CC}})$.
4. Output (z, pk).

We show that the definition of strong KDM security with auxiliary input is broken for $\mathcal{E}$ with respect to the distribution D for function $f(\mathsf{pk}, z) = g(y_2) = y_1$, where g is the (possibly inefficient) function that exists by the uniqueness property of G. Namely, for $(z, \mathsf{pk}^*) \leftarrow D(1^\lambda)$, we show

$$\{(z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(0^s))\}_{\lambda\in\mathbb{N}} \not\approx_c \{(z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(f(\mathsf{pk}^*, z)))\}_{\lambda\in\mathbb{N}}, \tag{A}$$

but

$$\{(z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(0))\}_{\lambda\in\mathbb{N}} \approx_c \{(z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(1))\}_{\lambda\in\mathbb{N}}. \tag{B}$$

It remains to prove that (A) and (B) hold.

Proof of (A). We construct a non-uniform PPT algorithm $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$ such that there exists a negligible function μ that for all $\lambda \in \mathbb{N}$

$$|\Pr[\mathcal{A}_\lambda(z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(0^s)) = 1] - \Pr[\mathcal{A}_\lambda(z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(f(\mathsf{pk}^*, z)))]| \geq 1 - \mu(\lambda), \tag{1}$$

for $(z, \mathsf{pk}^*) \leftarrow D(1^\lambda)$. Each $\mathcal{A}_\lambda$ is defined as follows.

$\mathcal{A}_\lambda(1^\lambda, z, \mathsf{pk}^*, \mathsf{ct})$:

1. Parse z as $(y_2, \widetilde{\mathsf{CC}})$.
2. Output $\widetilde{\mathsf{CC}}(\mathsf{ct})$.

We first bound $\Pr[\mathcal{A}_\lambda(1^\lambda, z, \mathsf{pk}^*, \mathsf{ct}) = 1]$ for $\mathsf{ct} = \mathsf{Enc}_{\mathsf{pk}^*}(f(\mathsf{pk}^*, z))$. By definition, $\mathsf{CC}[\mathsf{Dec}_{\mathsf{sk}}, y_1](\mathsf{ct})$ outputs 1 if and only if $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}) = y_1$. By the correctness guarantee of compute-and-compare obfuscation, there is a negligible function ν_1 such that $\widetilde{\mathsf{CC}}(\mathsf{ct}) = 1$ when $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}) = y_1$ with probability at least $1 - \nu_1(\lambda)$. Thus,

$$\Pr[\mathcal{A}_\lambda(z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(f(\mathsf{pk}^*, z))) = 1] \geq 1 - \nu_1(\lambda).$$

Next we bound $\Pr[\mathcal{A}_\lambda(1^\lambda, z, \mathsf{pk}^*, \mathsf{ct}) = 1]$ for $\mathsf{ct} = \mathsf{Enc}_{\mathsf{pk}^*}(0^\lambda)$. Let BAD be the event that $y_1 = 0^\lambda$. By the security of the PRG G, it must be the case that $\Pr[\mathsf{BAD}] \leq \nu_2(\lambda)$ for some negligible function ν_2. Suppose otherwise that $\Pr[\mathsf{BAD}] > 1/q(\lambda)$ for some polynomial q. Then we can construct a non-uniform PPT algorithm that distinguishes $G(U_\lambda)$ from $U_{2\lambda}$ with noticeable probability by checking if the first λ bits are all 0, which would happen with $2^{-\lambda}$ probability for $U_{2\lambda}$ and at least $1/q(\lambda)$ probability for $G(U_\lambda)$ by assumption.

Given BAD doesn't occur, $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}) \neq u$ by correctness of the underlying encryption scheme $\mathcal{E}$. Then by the correctness guarantee of compute-and-compare obfuscation, $\Pr[\mathcal{A}_\lambda(1^\lambda, z, \mathsf{pk}^*, \mathsf{ct}) = 1 \mid \neg\mathsf{BAD}] \leq \nu_1(\lambda)$. It follows then that for infinitely many $\lambda \in \mathbb{N}$,

$$\begin{aligned}\Pr[\mathcal{A}_\lambda(1^\lambda, z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(0^\lambda)) = 1] &\leq \Pr[\mathcal{A}_\lambda(1^\lambda, z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(0^\lambda)) = 1 \mid \neg\mathsf{BAD}] + \Pr[\mathsf{BAD}] \\ &\leq \nu_1(\lambda) + \nu_2(\lambda).\end{aligned}$$

Finally, we note that Equation (1) holds for negligible $\mu = 2\nu_1 + \nu_2$, which completes the proof of (A).

Before proving (B), we define a hybrid distribution D_{Sim} that replaces $\widetilde{\mathsf{CC}}$ with the simulated circuit Sim, which is now independent of y_1.

$D_{\mathsf{Sim}}(1^\lambda)$:

1. Sample $s \leftarrow \{0,1\}^\lambda$, $(y_1, y_2) = G(s)$, and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$.
2. Compute $\widetilde{\mathsf{CC}} \leftarrow \mathsf{Sim}(1^\lambda, 1^\ell)$ where ℓ is the size of the circuit computing $\mathsf{Dec}_{\mathsf{sk}}$.
3. Let $z = (y_2, \widetilde{\mathsf{CC}})$.
4. Output (z, pk).

Fix any message m and then consider sampling $z = ((y_2, \mathcal{O}(1^\lambda, \mathsf{CC}[\mathsf{Dec}_{\mathsf{sk}}, y_1])), \mathsf{pk}) \leftarrow D(1^\lambda)$. Since m, sk, and pk are independent of y_1 and y_1 has λ-pseudo-entropy given y_2, it follows that y_1 still has λ-pseudo-entropy given $(y_2, \mathsf{Dec}_{\mathsf{sk}}, \mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(m))$. This allows us to indistinguishably replace D with D_{Sim} as long as ct is an encryption of a message m that is independent of y_1. Specifically, by the simulation guarantee of compute-and-compare obfuscation, the following equation holds for m independent of y_1,

$$\{(z, \mathsf{pk}^*) \leftarrow D(1^\lambda) : (z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(m))\}_{\lambda\in\mathbb{N}} \approx_c \{(z, \mathsf{pk}^*) \leftarrow D_{\mathsf{Sim}}(1^\lambda) : (z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(m))\}_{\lambda\in\mathbb{N}} \quad (2)$$

We are now ready to prove (B).

Proof of (B). We construct a sequence of computationally indistinguishable hybrid ensembles H_0, H_1, H_2, H_3.

- H_0: This ensemble is the left-hand side of (B).
$$H_0 = \{(z, \mathsf{pk}^*) \leftarrow D(1^\lambda) : (z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(0))\}_{\lambda\in\mathbb{N}}$$
- H_1: This ensemble is H_0 except D is replaced with D_{Sim}.
$$H_1 = \{(z, \mathsf{pk}^*) \leftarrow D_{\mathsf{Sim}}(1^\lambda) : (z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(0))\}_{\lambda\in\mathbb{N}}$$
- H_2: This ensemble is H_1 except $\mathsf{Enc}_{\mathsf{pk}^*}(0)$ is replaced with $\mathsf{Enc}_{\mathsf{pk}^*}(1)$.
$$H_2 = \{(z, \mathsf{pk}^*) \leftarrow D_{\mathsf{Sim}}(1^\lambda) : (z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(1))\}_{\lambda\in\mathbb{N}}$$
- H_3: This ensemble is the right-hand side of (B).
$$H_3 = \{(z, \mathsf{pk}^*) \leftarrow D(1^\lambda) : (z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(1))\}_{\lambda\in\mathbb{N}}$$

$H_0 \approx_c H_1$ and $H_2 \approx_c H_3$: These follow from Equation (2).

$H_1 \approx_c H_2$*:* Suppose by way of contradiction that H_1 and H_2 are not computationally indistinguishable. Namely, there exists a non-uniform PPT algorithm $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$ and a polynomial p such that for infinitely many $\lambda \in \mathbb{N}$,

$$|\Pr[\mathcal{A}_\lambda(1^\lambda, z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(0)) = 1] - \Pr[\mathcal{A}_\lambda(1^\lambda, z, \mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(1)) = 1] > 1/p(\lambda),$$

for $(z, \mathsf{pk}^*) \leftarrow D_{\mathsf{Sim}}(1^\lambda)$. We use $\mathcal{A}$ to construct a non-uniform PPT algorithm $\mathcal{B} = \{\mathcal{B}_\lambda\}_{\lambda\in\mathbb{N}}$ that breaks the semantic security of $\mathcal{E}$. For $\lambda \in \mathbb{N}$, $\mathcal{B}_\lambda$ has g, Sim, and $\ell(\lambda)$ hard coded where $\ell \in \mathsf{poly}(\lambda)$ is the size of the circuit computing $\mathsf{Dec}_{\mathsf{sk}}$. We define $\mathcal{B}_\lambda$ as follows.

$\mathcal{B}_\lambda(1^\lambda, \mathsf{pk}, \mathsf{ct})$:

1. Sample $s \leftarrow \{0,1\}^\lambda$ and compute $(y_1, y_2) = G(s)$.
2. Compute $\widetilde{\mathsf{CC}} \leftarrow \mathsf{Sim}(1^\lambda, 1^\ell)$.
3. Output $\mathcal{A}_\lambda(1^\lambda, (y_2, \widetilde{\mathsf{CC}}), \mathsf{pk}, \mathsf{ct})$.

Because $\mathcal{B}_\lambda$ computes $(y_2, \widetilde{\mathsf{CC}})$ identically to D_{Sim}, $\mathcal{B}_\lambda$ outputs 1 with the same probability as $\mathcal{A}_\lambda$. Thus, for infinitely many $\lambda \in \mathbb{N}$,

$$|\Pr[\mathcal{B}_\lambda(1^\lambda, \mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(0)) = 1] - \Pr[\mathcal{B}_\lambda(1^\lambda, \mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(1)) = 1] > 1/p(\lambda),$$

contradicting the semantic security of $\mathcal{E}$.

This completes the proof of (B) and hence Theorem 2. □

4.1 Dealing with Bounded Auxiliary Input

We show how to deal with auxiliary input sublinear in the size of pk, which we assume to be equal to λ. We rely on the following two new ideas to deal with bounded auxiliary input:

1. We replace the compute-and-compare obfuscation with a succinct one (which requires the stronger assumption of (non-leveled) FHE).
2. We use a PRG to generate the randomness needed for key generation, which allows us to use a compressed version of the secret key sk in the obfuscation.

We note that for encryption schemes $\mathcal{E}$ where decryption can be computed by a low depth circuit, *weakly succinct* compute-and-compare obfuscation suffices, which is known from leveled FHE and can be based on LWE alone.

Theorem 3. *Assuming LWE, (non-leveled) FHE, and one-way permutations, no semantically secure encryption scheme satisfies strong KDM security with λ^δ-bounded auxiliary input for any $\delta > 0$.*

Proof. For a constant $\epsilon \in (0,1)$ (to be chosen later), we instantiate the attack of Theorem 2 with the following modifications:

1. We use a standard PRG G with seed r to maliciously choose the keys for the encryption scheme, which allows us to compute $\mathsf{Dec}_{\mathsf{sk}}$ with a function Dec^*_r such that $|\mathsf{Dec}^*_r| \leq \lambda^\epsilon$.
2. We use a PRG $G' \colon \{0,1\}^{\lambda^\epsilon} \to \{0,1\}^{\lambda^\epsilon} \times \{0,1\}^{\lambda^\epsilon}$, i.e., the seed s is sampled from $\{0,1\}^{\lambda^\epsilon}$, satisfying uniqueness and indistinguishability as in Theorem 2. Since ϵ is constant, G' satisfies polynomial security in λ.
3. We use λ^ϵ as the security parameter for a *succinct* compute-and-compare obfuscation for λ^ϵ-pseudo-entropy distributions, i.e., $\widetilde{\mathsf{CC}} = \mathcal{O}(1^{\lambda^\epsilon}, \mathsf{CC}[\mathsf{Dec}^*_r, y_1])$. Such a succinct obfuscator exists using (non-leveled) FHE by [WZ17]. Also, since ϵ is constant, $\mathcal{O}$ satisfies polynomial security in λ.

Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be any semantically secure encryption scheme such that Gen uses $k(\lambda)$ bits of randomness, and let G be a secure PRG for length $k^{1/\epsilon}(\lambda) \in \mathsf{poly}(\lambda)$. Consider the following distribution D.

$D(1^\lambda)$:

1. Sample $s \leftarrow \{0,1\}^{\lambda^\epsilon}$, $(y_1, y_2) = G'(s)$, $r \leftarrow \{0,1\}^{\lambda^\epsilon}$, and $(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow \mathsf{Gen}(1^\lambda; G(r))$.
2. Construct Dec^*_r to be the function that, on input ct, first computes $(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow \mathsf{Gen}(1^\lambda; G(r))$ and outputs $\mathsf{Dec}_{\mathsf{sk}^*}(\mathsf{ct})$.
3. Compute $\widetilde{\mathsf{CC}} \leftarrow \mathcal{O}(1^{\lambda^\epsilon}, \mathsf{CC}[\mathsf{Dec}^*_r, y_1])$.
4. Let $z = (y_2, \widetilde{\mathsf{CC}})$.
5. Output (z, pk^*).

We note that we can represent Dec^*_r with $O(1) + \lambda^\epsilon$ bits as a constant size Turing machine that runs in polynomial-time with r hard-coded. Since the compute-and-compare obfuscator is succinct, this implies that $|z| \leq (\lambda^\epsilon)^c$ for some constant c. We choose $\epsilon \leq \delta/c$ such that $|z| \leq \lambda^\delta$.

To finish the proof of the theorem, it remains to argue that correctness and semantic security of the encryption scheme still hold when maliciously using a PRG to generate the keys as in D. Correctness holds since $\mathcal{E}$ satisfies perfect correctness, i.e., correctness holds no matter what randomness is used by Gen. Semantic security holds by the PRG security of G and semantic security of the underlying encryption scheme. Specifically, if we could distinguish encryptions of m and m' when the randomness for Gen is generated by G, we could either distinguish $G(U_{\lambda^\epsilon})$ from $U_{k(\lambda)}$ or distinguish encryptions of m and m' for $\mathcal{E}$. The rest of the proof is identical to that of Theorem 2. □

References

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6:1–6:48, 2012.

[BM14] Christina Brzuska and Arno Mittelbach. Indistinguishability obfuscation versus multi-bit point obfuscation with auxiliary input. In *ASIACRYPT*, pages 142–161, 2014.

[BRS02] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *Selected Areas in Cryptography, 9th Annual International Workshop, SAC*, pages 62–75, 2002.

[BST16] Mihir Bellare, Igors Stepanovs, and Stefano Tessaro. Contention in cryptoland: Obfuscation, leakage and UCE. In *TCC*, pages 542–564, 2016.

[CCRR18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. Fiat-shamir and correlation intractability from strong kdm-secure encryption. In *Advances in Cryptology - EUROCRYPT*, pages 91–122, 2018.

[CKVW10] Ran Canetti, Yael Tauman Kalai, Mayank Varia, and Daniel Wichs. On symmetric encryption and point obfuscation. In *TCC*, pages 52–71, 2010.

[DK18a] Apoorvaa Deshpande and Yael Kalai. Proofs of ignorance and applications to 2-message witness hiding. *IACR Cryptology ePrint Archive*, 2018:896, 2018. Version dated: 25-Sep-2018.

[DK18b] Apoorvaa Deshpande and Yael Kalai. Proofs of ignorance and applications to 2-message witness hiding. *IACR Cryptology ePrint Archive*, 2018:896, 2018. Version dated: 02-Mar-2019.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 612–621, 2017.

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

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer, 2013.

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

[HLR07] Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In *EUROCRYPT*, pages 169–186, 2007.

[KMN+14] Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon Yogev. One-way functions and (im)perfect obfuscation. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 374–383, 2014.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 600–611, 2017.