Barriers in Cryptography with Weak, Correlated and Leaky Sources

Daniel Wichs *

August 13, 2012

Abstract

There has been much recent progress in constructing cryptosystems that maintain their security without requiring uniform randomness and perfect secrecy. These schemes are motivated by a diverse set of problems such as providing resilience to side-channel *leakage*, using *weak physical sources* of randomness as secret keys, and allowing *deterministic encryption* for high-entropy messages. The study of these problems has significantly deepened our understanding of how randomness is used in cryptographic constructions and proofs.

Nevertheless, despite this progress, some basic and seemingly achievable security properties have eluded our reach. For example, we are unable to prove the security of basic tools for manipulating weak/leaky random sources, such as as *pseudo-entropy generators* and *seed-dependent computational condensers*. We also do not know how to prove leakage-resilient security of any cryptosystem whose secret key is *uniquely determined* by its public key. In the context of deterministic encryption we do not have a standard-model constructions achieving the strongest notion of security proposed by Bellare, Boldyreva and O'Neill (CRYPTO '07), that would allow us to encrypt arbitrarily correlated messages of sufficiently large individual entropy.

In this work, we provide broad black-box separation results, showing that the security of such primitives cannot be proven under virtually *any* standard cryptographic hardness assumption via a reduction that treats the adversary as a *black box*. We do so by formalizing the intuition that "the only way that a reduction can simulate the correctly distributed view for an attacker is to know all the secrets, in which case it does not learn anything useful from the attack". Such claims are often misleading and clever way of getting around them allow us to achieve a wealth of positive results with imperfect/leaky randomness. However, in this work we show that this intuition can be formalized and that it indeed presents a real barrier for the examples given above.

*IBM Research. wichs@cs.nyu.edu

1 Introduction

We look at several related scenarios involving imperfect randomness and secrecy. Although the motivation behind studying these scenarios differs greatly from one to another, the technical means of achieving and analyzing security are strikingly similar between them.

Leakage-Resilient Cryptography. Motivated by the prevalence of various physical *side-channel attacks*, the study of leakage-resilient cryptography strives to construct cryptosystems that maintain their security even if the attacker can observe some partial leakage related to the secret key of the cryptosystem. By now, many different formal models of leakage-resilience have been proposed in the literature – see e.g., [ISW03, MR04, DP08, AGV09, DHLW10a, BKKV10, GR12] and references therein. Perhaps the most basic model, sometimes also called the *bounded-leakage or memory-leakage model* was proposed by Akavia, Goldwasser and Vaikuntanathan [AGV09]. This model allows the attacker to learn any adversarially chosen information about the secret key, as long as the amount of information is not too large. More specifically, a cryptosystem is *ℓ-leakage-resilient*, if its security is maintained even if the attacker can choose any efficient *leakage function* $\mathsf{Leak} : \{0,1\}^* \to \{0,1\}^\ell$ and gets to observe the resulting value $\mathsf{Leak}(sk)$. In the past few years, we have seen a vast number of positive results (e.g., [AGV09, ADW09a, NS09, KV09, ADN$^+$10, GKPV10, BG10, DHLW10b, BSW11, BHK11, HL11] etc.) showing how to construct many such leakage resilient schemes including encryption, signatures and various related primitives, under various (standard) computational assumptions.

Cryptography with Weak Sources. The study of cryptography with *weak randomness* aims to build cryptosystems whose security is maintained even if their secret keys are chosen from some efficiently samplable but otherwise unspecified/adversarial distribution of sufficient min-entropy. This can model various physical sources whose distributions are complex and we know little about them (e.g., password or biometrics). The connection between weak and leaky randomness can be made formal in the information-theoretic setting: conditioned on ℓ-bits of leakage, a uniformly random source can loose at most ℓ bits of entropy. However, such connections are lost in the computational setting since, even if a leakage function is efficiently computable, the distribution of secret keys conditioned on some fixed leakage may not be efficiently samplable. Nevertheless, most of the known positive results for leakage-resilient cryptography do also carry over to the setting of weak randomness.

Deterministic Encryption and Correlated Sources. The issue of imperfect randomness also occurs in the context of *deterministic encryption* [BBO07, BFOR08, BFO08, BS11, FOR12]. In such encryption schemes, we assume that the messages being encrypted already contain some reasonable amount of entropy, allowing us to make the encryption procedure itself deterministic without sacrificing security. The original work of Bellare, Boldyreva and O'Neill [BBO07], which introduced deterministic encryption, suggested that such schemes should be able to securely encrypt multiple distinct messages that come from any arbitrarily correlated distribution, as long as each message individually has sufficient entropy. A corresponding scheme satisfying this notion was then constructed in the *random oracle model*, and achieving this notion in the standard model was left as a major open question. Subsequent works showed how to realize a *weaker* notions of security in the standard model, by assuming that each message contains some fresh entropy conditioned on all others, but the question of achieving the strongest notion of security remains open. Similar issues of maintaining security under correlated random inputs also come up in several related contexts [RS09, HLO10, GOR11] that do not explicitly deal with deterministic encryption.

1.1 The "Paradox" of Cryptography with Imperfect Randomness

One of the challenges in proving the security of cryptosystems with imperfect (weak/leaky/correlated) sources is having the reduction generate a correctly distributed view for the attacker. Let us look at

a simple case study, using the notion of "*leakage resilient one-way functions*" as an example. Such a function f should have the property that, if we choose x at random and give the attacker $y = f(x)$, then it should be hard for the attacker to find a preimage $x' \in f^{-1}(y)$, *even* if the attacker can observe some arbitrary (but sufficiently short) leakage $\mathsf{Leak}(x)$ about the input x. Now imagine we have some candidate construction f and want to prove it secure via a *reduction* from some hard problem. Then the reduction needs to run the attacker and give it a correctly distributed pair $(y, \mathsf{Leak}(x))$. But, it seems that the only way for the reduction to generate a pair of this form is to *already* know x, in which case there is seemingly no point in running the attacker, since it won't yield anything new that the reduction doesn't already know! Similar issues appear to come up in cryptography with weak/correlated sources, where the only way for the reduction to ensure that the attacker gets correctly distributed view is to seemingly just sample all of the secrets itself, in which case the attacker won't produce anything useful that the reduction doesn't already know. We call the above intuition the *"useless attacker paradox"*.

Overcoming the Paradox. Of course, the above paradox is not very formal (as highlighted by the excessive use of the word "seemingly"), and fortunately breaks down on deeper scrutiny. For example, in the case of leakage-resilient one-way functions, even if the reduction produces a correctly distributed pair $(y, \mathsf{Leak}(x))$ by choosing x on its own, the attacker may respond with some other preimage $x' \neq x$, which may not be known to the reduction. Indeed, this loophole can be turned into a positive result, showing that any collision-resistant (and even just second-preimage resistant) function f is already a leakage-resilient one-way function (see e.g., [ADW09b]). In general, the process of explicitly considering the "useless attacker paradox" and seeing where it may break down, is often a good way to arrive at many of the positive results in this area.

Formalizing the Paradox in Special Cases. Nevertheless, despite the success of cryptographic constructions in overcoming the seeming paradox in many cases, there are several important primitives which have eluded the grasp of provably secure constructions so far. In this work, we examine and formalize the "useless attacker paradox" and show that it indeed does present a real barrier in these important instances. For example, returning to our case-study of leakage-resilient one-way functions, let us add an additional requirement that the function f is *injective*. Then, for any given y, there is a unique value $\mathsf{Leak}(x)$ that the attacker expects to see and a unique x that the attacker will produce. Therefore the reduction really is stuck: either it doesn't know x, in which case it will not be able to provide a correctly distributed leakage $\mathsf{Leak}(x)$ to the attacker, or it does know x, in which case the attacker's unique output is useless. In this work we manage to formalize such intuition into a broad *black-box separation result* for several important primitives as highlighted below.

1.2 Our Results

Simulatable Attackers are Useless. Our first result is to formalize the "useless attacker paradox". We say that an *inefficient* attack $\mathcal{A}$ against some cryptographic scheme is *simulatable* if there exists an *efficient* simulator Sim such that getting oracle access to $\mathcal{A}$ is indistinguishable from getting oracle access to Sim. Notice tat the mere existence of an inefficient attack against a cryptographic scheme is usually not interesting or surprising, but the existence of a *simulatable* attack turns out to have an interesting consequence. In particular, we show that the existence of a simulatable attack against some scheme implies that the security of the scheme cannot be proven via a *black-box reduction* from a large class of assumptions modeled as *cryptographic games* between an attacker and challenger. This captures essentially all standard assumptions in cryptography (factoring, CDH, DDH, RSA, LWE etc.).

The intuition is simple - anything a reduction could do with the help of $\mathcal{A}$, it could do efficiently on its own by running Sim. More concretely, assume that some `scheme` has a simulatable attack $\mathcal{A}$ with corresponding simulator Sim, and that there is a black-box reduction $\mathcal{R}$ proving the security of this `scheme` based on some `assumption`. Then the reduction $\mathcal{R}^{\mathcal{A}}$ must be able to use oracle access to $\mathcal{A}$ to

break the security of the `assumption`. But since $\mathcal{A}$ and Sim are indistinguishable, there is also an efficient attack $\mathcal{R}^{\mathsf{Sim}}$ that breaks the security of the `assumption` efficiently, contradicting its hardness.

Notice that this type of a black-box separation shows that the security of a *concrete scheme* cannot be proven from a large class of *concrete assumptions* via a reduction that *treats the attacker as a black box*. This is different from most black-box separations in the literature (e.g., [IR89, Sim98, GKM+00, GMR01, RTV04]) which show that some *target primitive* (e.g., key-agreement) cannot be constructed from an *arbitrary instance of some source primitive* (e.g., a one-way function) if the *construction treats the source primitive as a black box*. Several prior black box separations in the literature [DOPS04, HH09, GW11, Pas11, DHT12] for completely unrelated notions have the same flavor as the one in this work and rely on a similar notion of a "simulatable attack". One of the contributions of this work is to abstract out this useful technique.

Simulatable Attacks for Imperfect Randomness. At first, it may seem that the existence of an inefficient simulatable attack $\mathcal{A}$ against some `scheme` would simply imply that the `scheme` is insecure – if the inefficient $\mathcal{A}$ breaks the security of the `scheme` and Sim is indistinguishable from $\mathcal{A}$, then the efficient simulator Sim should also break the security of the `scheme`. Indeed, the above holds for all schemes whose security can itself be modeled as a cryptographic game. However, this is not the case for the security definitions involving leakage or imperfect randomness that we consider in this work.

For example, in the case of leakage, a valid attacker $\mathcal{A}$ is required to consist of two independent (non-communicating) components $\mathcal{A} = (\mathsf{Leak}, \mathsf{Break})$ and the security game consists of *two stages*: in the first stage the attacker generates leakage $z = \mathsf{Leak}(sk)$ on the secret key sk, and in the second stage the attacker Break attempts to break security (e.g., invert a one-way function of sk) given the leakage z. The attacker $\mathcal{A}$ cannot keep any state in between these two stages. The simulator Sim simulating $\mathcal{A}$, however, may not satisfy these structural requirements of the definition - it is allowed to simulate queries to Leak and Break in a coordinated manner while keeping some state. In other words, the efficient simulator Sim will not have the structure of a valid attack on leakage-resilient security of the scheme and therefore does not contradict its security. A similar phenomenon occurs in security definitions involving weak/correlated sources of randomness, where an attacker $\mathcal{A} = (\mathsf{Sam}, \mathsf{Break})$ consists of two independent components: a sampling algorithm Sam that defines some adversarial distribution and a Break component which attempts to break security when the secrets are sampled according to Sam.

It is precisely because the security definitions involving imperfect sources of randomness consist of multi-stage games with multi-component attackers $\mathcal{A}$, that makes them *different* from standard *cryptographic game* assumptions (e.g., DDH, RSA, LWE etc.) and gives us the opportunity to prove such broad black-box separations.[1] Note that by describing the attacker as consisting of two independent components in a multi-stage game, we are assuming that a black-box reduction treats both components – including the leakage function or the adversarial distribution – as a black box. This type of reduction was also called a *strongly* black-box reduction in the work of [HH09], but we can really think of it as the natural notion of a black-box reduction when dealing with a multi-component attacker.

Primitives with Simulatable Attacks. Our main result is to look at several cryptographic primitives involving imperfect randomness and show that *every* candidate scheme for these primitives has a simulatable attack against it. Therefore, there is no instantiation of these primitives that can be proven secure via a black-box reduction from any cryptographic game assumption. Our results fall into three categories.

***Unique Witnesses*:** We begin by considering *one-way relations with unique witnesses*, which generalizes the example of injective one-way functions and cryptosystems (encryption, signature etc.) where every public key has a unique secret key. We show that there is always an inefficient simulatable

[1] This distinction between security definitions involving multi-stage and single-stage games is also prominent in the work of [RSS11] in the context of indifferentiabiliy.

attack against *leakage-resilient one-wayness* of such relations, when the attacker can observe leakage of super-logarithmic size, and therefore the leakage-resilient security of such schemes cannot be shown via black-box reductions from standard assumptions. Similar separations also apply to security with *weak randomness* in place of leakage-resilience.

***Deterministic Encryption and Correlated Inputs*:** Next, we consider the strong notion of deterministic encryption from Bellare et al. [BBO07], which allows us to deterministically encrypt distinct messages $m_1, \ldots, m_t$ chosen from any distribution that places sufficient entropy on each individual m_i, but otherwise allows the messages to be arbitrarily correlated. The definition of security aims to ensure that the ciphertexts $c_i = \mathsf{Enc}_{pk}(m_i)$ do not reveal essentially any information about the messages. The work of [BBO07] gives a construction of this strong security notion in the random oracle model. Subsequent works [BFOR08, BFO08, BS11, FOR12] were only able to achieve a weaker security notion, where each message m_i must have fresh entropy conditioned on all the others. Here, we focus on the strong notion of security where the messages can be arbitrarily correlated, and we show a simulatable-attack against this notion for any candidate scheme, thus deriving a broad black-box separation result. In fact, our result generalizes to any function family $\{f_{pk}\}$ (even one without a decryption trapdoor), and shows black-box impossibility of proving the one-wayness of $(f_{pk}(m_1), \ldots, f_{pk}(m_t))$ when the distinct inputs $m_1, \ldots, m_t$ can come from an arbitrarily correlated distribution, even if each individual value m_i has high entropy.

***Entropy Generation and Condensation*:** Lastly, we consider basic tools for manipulating entropy of weak/leaky sources. The first such tool is a *pseudo-entropy generator* which takes a random seed x from a weak or leaky distribution and outputs a longer value $y = \mathsf{PEG}(x)$ such that the computational entropy of y is greater than that of x. This tool is used as a building block in several leakage-resilient primitives and was initially explored by Dziembowski and Pietrzak [DP08]. The second such tool is a (seed-dependent) *condenser* which takes a value x from a weak distribution and outputs a shorter value $y = \mathsf{Cond}(x)$ such that the entropy-loss of y (the difference between the length of y and its entropy) is smaller than that of x. This tool was explored in the recent work of Dodis et al. [DRV12]. The only known constructions of these primitives require exponential hardness assumptions and we show that this is necessary under black-box reductions.

2 Preliminaries

Notation. Let n be the security parameter. A function $\mu(n)$ is negligible, denoted $\mu(n) = \mathsf{negl}(n)$, if $\mu(n) = 1/n^{\omega(1)}$. We also use $\mathsf{poly}(n)$ to denote $n^{O(1)}$. If X is a probability distribution or a random variable then $x \leftarrow X$ denotes the process of sampling a value x at random according to X. If S is a set then $s \xleftarrow{\$} S$ denotes sampling s according to the *uniformly random* distribution over the set S. We let $[n]$ denote the set $\{1, \ldots, n\}$.

Entropy. The *min-entropy* of a random variable X is $\mathbf{H}_\infty(X) \stackrel{\text{def}}{=} -\log(\max_x \Pr[X = x])$. This is a standard notion of entropy used in cryptography, since it measures the worst-case predictability of X. We often find it useful to work with a generalized version of min-entropy, called *average conditional min-entropy*, defined by[DORS08] as

$$\mathbf{H}_\infty(X|Z) \stackrel{\text{def}}{=} -\log\left(\mathop{\mathbf{E}}_{z \leftarrow Z}\left[\ \max_x \Pr[X = x|Z = z]\ \right]\right) = -\log\left(\mathop{\mathbf{E}}_{z \leftarrow Z}\left[2^{-\mathbf{H}_\infty(X|Z=z)}\right]\right).$$

This measures the best guess for X by an adversary that may observe an *average-case* correlated variable Z. That is, for all (inefficient) functions $\mathcal{A}$, we have $\Pr[\mathcal{A}(Z) = X] \leq 2^{-\mathbf{H}_\infty(X|Z)}$ and there exists some $\mathcal{A}$ which achieves equality. The following lemma says that conditioning on ℓ bits of information, the min-entropy drops by at most ℓ bits.

Lemma 2.1 ([DORS08]). *Let X, Y, Z be random variables where Y takes on values in a set of size at most 2^ℓ. Then $\mathbf{H}_\infty(X|(Y,Z)) \geq \mathbf{H}_\infty((X,Y)|Z) - \ell \geq \mathbf{H}_\infty(X|Z) - \ell$ and, in particular, $\mathbf{H}_\infty(X|Y) \geq \mathbf{H}_\infty(X) - \ell$.*

3 Cryptographic Games and Black-Box Reductions

Cryptographic Games. We begin by defining a general notion of cryptographic games between a challenger and an attacker. Cryptographic games are used to capture the security requirements of most cryptographic primitives, as well as to define the standard hardness assumptions that we rely on in cryptography.

Definition 3.1 (Cryptographic Game [HH09]). *A* cryptographic game $\mathcal{G} = (\Gamma, c)$ *is defined by a (possibly inefficient) random system Γ, called the* challenger*, and a constant $c \in [0,1)$. On security parameter n, the challenger $\Gamma(1^n)$ interacts with some attacker $\mathcal{A}(1^n)$ and outputs a bit b. We denote this interaction by $b = (\mathcal{A}(1^n) \leftrightarrows \Gamma(1^n))$. The* advantage *of an attacker $\mathcal{A}$ in the game $\mathcal{G}$ is defined as*

$$\mathbf{Adv}^{\mathcal{A}}_{\mathcal{G}}(n) \stackrel{def}{=} \Pr[\ (\mathcal{A}(1^n) \leftrightarrows \Gamma(1^n)) = 1\ ] - c.$$

A cryptographic game $\mathcal{G}$ is secure *if for all* PPT *attackers $\mathcal{A}$, the advantage $\mathbf{Adv}^{\mathcal{A}}_{\mathcal{G}}(n)$ is negligible.*

We say that a (possibly inefficient) attacker $\mathcal{A}$ *successfully breaks the security of* $\mathcal{G}$ if the advantage $\mathbf{Adv}_{\mathcal{G}}(\mathcal{A})$ is *not* negligible. When $c = 0$, the above definition of cryptographic games captures *search problems* such as the one-way hardness of factoring, the discrete logarithm problem, etc. When $c = \frac{1}{2}$, it captures *decisional problems* such as DDH. Note that cryptographic games may be highly interactive and may not even have any a-priori bound on the number of rounds of interaction between $\mathcal{A}, \Gamma$. For example, the security of RSA signatures when instantiated with some concrete hash function family would qualify as a cryptographic game assumption. Lastly, we mention that the work of [GW11] defined a more restricted notion of cryptographic games (called "falsifiable assumptions") where the challenger is also required to be efficient. We do *not* rely on this requirement in the current work. Essentially all common assumptions used in cryptography fall under the framework of cryptographic games.

We can also define a cryptographic game $\mathcal{G}$ to be *δ-exponentially secure* for some constant $\delta > 0$ if for all $\mathcal{A}(1^n)$ running in time $2^{O(n^\delta)}$ the advantage $\mathbf{Adv}^{A}_{\mathcal{G}}(n) = 2^{-\Omega(n^\delta)}$.

Cryptographic Security Properties Beyond Games. The above definition captures most standard cryptographic security notions. However, not all cryptographic properties can be defined via the above template. For example, various definitions involving leakage resilience and weak randomness consider an attacker that consists of two independent (non-communicating) components rather than a single monolithic attacker as needed for cryptographic games. In addition, they can make additional restriction, such as requiring that the adversary defines a distribution of high min-entropy, which cannot easily be checked by a challenger. Therefore, we also give a very general definition of an arbitrary *cryptographic property* $\mathcal{C}$ as a mapping which assigns to each attacker $\mathcal{A}$ some real number $\mathbf{Adv}^{\mathcal{A}}_{\mathcal{C}}(n)$ indicating how successful $\mathcal{A}$ is in breaking the property. We say that $\mathcal{C}$ is secure if for *all* PPT attackers $\mathcal{A}$, the advantage $\mathbf{Adv}^{\mathcal{A}}_{\mathcal{C}}(n)$ is negligible in the security parameter. Although this definition is very general, and most security properties of this type are not very meaningful, it captures several useful notions which we can't easily capture by cryptographic games. In this work, we will only consider several concrete and meaningful instances of "cryptgoraphic properties" involving leakage-resilience and weak/correalated randomness.

Black-Box Reductions. We now define the concept of a *black-box reduction*. Almost all proofs in cryptography have the form of a black-box reductions and hence this captures a meaningful notion. Since the focus of this work is on black-box separations, we want to define as *weak* of a definition as possible. For simplicity, we will therefore assume that any attacker given to the reduction has advantage $\geq \frac{1}{2}$ (rather than insisting that the reduction works for all attackers with any non-negligible advantage).

Definition 3.2 (Black Box Reduction)**.** *Let $\mathcal{C}$ be some cryptographic property and let $\mathcal{G}$ be a cryptographic game. A* black-box reduction deriving the security of $\mathcal{C}$ from the security of $\mathcal{G}$ *is an oracle-access* PPT *machine $\mathcal{B}^{(\cdot)}$ for which there are some constants $c, N_0 > 0$ such that, for all integers $n \geq N_0$ and all (possibly inefficient, non-uniform) oracles $\mathcal{A}_n$ with $\mathbf{Adv}_{\mathcal{C}}^{\mathcal{A}_n}(n) \geq \frac{1}{2}$, we have $\mathbf{Adv}_{\mathcal{G}}^{\mathcal{B}^{\mathcal{A}_n}}(n) \geq n^{-c}$.*
We also consider black-box reductions deriving the security of $\mathcal{C}$ from the δ-exponential security of $\mathcal{G}$, by allowing $\mathcal{B}^{(\cdot)}$ to run in time $2^{O(n^\delta)}$ and only insisting that $\mathbf{Adv}_{\mathcal{G}}^{\mathcal{B}^{\mathcal{A}_n}}(n) \geq 2^{-o(n^\delta)}$.

Remarks on the Definition: In general, the attacker can be a stateful and interactive machine, in which case a black-box reduction can also use *rewinding access* rather than just *oracle access* to the attacker. However, for the security properties $\mathcal{C}$ considered in this work, the attacker is always (without loss of generality) stateless and hence we will safely ignore the issue of rewinding in this work.

Notice that, although we weaken the definition by insisting that the advantage of that attacker $\mathcal{A}$ given to the reduction is at least a half (for all n), we also do strengthen the definition by insisting that the reduction $\mathcal{B}(1^n)$ has a *noticeable* advantage $\geq n^{-c}$ for *every* large-enough n, rather than just insisting on some *non-negligible* advantage. Furthermore, we assume that on security parameter n, the reduction only queries the attacker $\mathcal{A}_n$ on the *same* security parameter n. This may incur some loss of generality, but all the known reductions in the literature that we are aware of are of this type. In particular, their advantage on every security parameters is related by some fixed polynomial to that of the attacker on the *same* security parameter. A less restrictive definition would say that for any $\mathcal{A}$ with $\mathbf{Adv}_{\mathcal{C}}^{\mathcal{A}}(n) \geq 1/2$ we would have $\mathbf{Adv}_{\mathcal{G}}^{\mathcal{B}^{\mathcal{A}}}(n)$ being non-negligible. Proving separations for the less restrictive definition often introduces many subtleties which detract from the main ideas of the result, but it is relatively easy to extend all our proofs to this less restrictive definition as well.

4 Black-Box Separations via a Simulatable Attack

We now describe a general technique for proving strong black-box separation results. Let $\mathcal{C}$ be some cryptographic property (e.g. the leakage-resilience security of some candidate construction). The main idea of the technique is to construct an inefficient but otherwise valid and successful attacker $\mathcal{A}$ against $\mathcal{C}$, along with an efficient simulator Sim so that no oracle-access machine can distinguish interaction with $\mathcal{A}$ and interaction with Sim. Usually, the efficient simulator is *not* going to be a valid attack against $\mathcal{C}$ since it will fail to satisfy some of the required structure. Therefore, the existence of such attack and simulator does not in-itself show the *insecurity* of $\mathcal{C}$ since the attacker $\mathcal{A}$ is valid but inefficient while the simulator Sim is efficient but not valid. However, it will show the impossibility of a black-box reduction proof of security for $\mathcal{C}$.

Definition 4.1. *A $\varepsilon(n)$-*simulatable attack *on a cryptographic property $\mathcal{C}$ consists of: (1) an ensemble of (possibly inefficient) stateless non-uniform attackers $\{\mathcal{A}_{n,h}\}_{n\in\mathbb{N},h\in\mathcal{H}_n}$ where $\mathcal{H}_n$ are some finite sets, and (2) a stateful* PPT *simulator* Sim*. We require that the following two properties hold:*

- *For each $n \in \mathbb{N}, h \in \mathcal{H}_n$, the (inefficient) attacker $\mathcal{A}_{n,h}$ successfully breaks the security of $\mathcal{C}$ with advantage $\mathbf{Adv}_{\mathcal{C}}^{\mathcal{A}_{n,h}}(n) = 1$.*
- *For every (possibly inefficient) oracle access machine $\mathcal{M}^{(\cdot)}$ making at most $q = q(n)$ queries to its oracle: $\mid \Pr_{h\xleftarrow{\$}\mathcal{H}_n, coins(\mathcal{M})}[\mathcal{M}^{\mathcal{A}_{n,h}}(1^n) = 1] \; - \; \Pr_{coins(\mathcal{M},\mathsf{Sim})}[\mathcal{M}^{\mathsf{Sim}(1^n)}(1^n) = 1] \mid \leq \mathsf{poly}(q(n))\varepsilon(n)$.*
  In other words, oracle access to $\mathcal{A}_{n,h}$ for a random $h \xleftarrow{\$} \mathcal{H}_n$ is indistinguishable from that to Sim*.*

We omit the $\varepsilon(n)$ and just say "simulatable attack" as shorthand for an $\varepsilon(n)$-simulatable attack with some negligible $\varepsilon(n) = \mathsf{negl}(n)$.

The following theorem shows that the existence of a simulatable attack against some cryptographic notion $\mathcal{C}$ implies that there is no black-box reduction deriving the security of $\mathcal{C}$ from *any* secure cryptographic game $\mathcal{G}$.

Theorem 4.2. *If there exists a simulatable attack against some cryptographic notion $\mathcal{C}$ and there is a black-box reduction showing the security of $\mathcal{C}$ from the security of some cryptographic game $\mathcal{G}$, then $\mathcal{G}$ is* not *secure. Furthermore, for any constant $\delta > 0$, if there exists an $(\varepsilon(n) = 2^{-\omega(n^\delta)})$-simulatable attack against $\mathcal{C}$ and there is a black-box reduction from the δ-exponential security of $\mathcal{G}$, then $\mathcal{G}$ is* not *δ-exponentially secure.*

Proof. Let $\mathcal{B}$ be the black-box reduction showing the security of $\mathcal{C}$ from that of $\mathcal{G} = (\Gamma, c)$. Let $\{A_{n,h}\}_{n\in\mathbb{N},h\in\mathcal{H}_n}$ and Sim be the simulatable attack as in Definition 4.1. Then, by the definition of a black-box reduction, there is some polynomial $p(\cdot)$ and some $N_0 \in \mathbb{N}$ such that for every $n \geq N_0$ and all $h \in \mathcal{H}_n$ we have

$$\mathbf{Adv}_{\mathcal{G}}^{\mathcal{B}^{\mathcal{A}_{n,h}}}(n) = \Pr_{\mathrm{coins}(\mathcal{B},\Gamma,\mathcal{A}_{n,h})}\left[(\mathcal{B}^{\mathcal{A}_{n,h}}(1^n) \leftrightarrows \Gamma(1^n)) = 1\right] - c \geq 1/p(n).$$

Now, since the above holds for all $h \in \mathcal{H}_n$, we can also take the probability over a random h to get:

$$\Pr_{h\leftarrow\mathcal{H}_n,\mathrm{coins}(\mathcal{B},\Gamma,\mathcal{A}_{n,h})}\left[(\mathcal{B}^{\mathcal{A}_{n,h}}(1^n) \leftrightarrows \Gamma(1^n)) = 1\right] - c \geq 1/p(n) \tag{1}$$

We can now group the reduction $\mathcal{B}$ and the challenger Γ together as a single oracle access machine $\mathcal{M}^{(\cdot)} \stackrel{\text{def}}{=} (\mathcal{B}^{(\cdot)}(1^n) \leftrightarrows \Gamma(1^n))$. Since $\mathcal{B}$ is polynomial-time, there must be some polynomial upper bound $q(n)$ on the number of queries made by $\mathcal{B}$ to its oracle. We use the simulator from Definition 4.1 to get

$$\left| \Pr_{h\stackrel{\$}{\leftarrow}\mathcal{H}_n,\mathrm{coins}(\mathcal{B},\mathcal{A}_{n,h},\Gamma)}\left[(\mathcal{B}^{\mathcal{A}_{n,h}}(1^n) \leftrightarrows \Gamma(1^n)) = 1\right] - \Pr_{\mathrm{coins}(\mathcal{B},\mathsf{Sim},\Gamma)}\left[(\mathcal{B}^{\mathsf{Sim}(1^n)}(1^n) \leftrightarrows \Gamma(1^n)) = 1\right] \right| \leq \mathsf{negl}(n) \tag{2}$$

Now, combining equations (1) and (2), we get

$$\mathbf{Adv}_{\mathcal{G}}^{\mathcal{B}^{\mathsf{Sim}}}(n) = \Pr_{\mathrm{coins}(\mathcal{B},\mathsf{Sim},\Gamma)}\left[(\mathcal{B}^{\mathsf{Sim}(1^n)}(1^n) \leftrightarrows \Gamma(1^n)) = 1\right] - c \ \geq 1/p(n) - \mathsf{negl}(n).$$

Therefore $\mathcal{B}^{\mathsf{Sim}}$ has non-negligible advantage against the game $\mathcal{G}$. Furthermore, the running time of $\mathcal{B}^{\mathsf{Sim}}$ is polynomial since both $\mathcal{B}$ and Sim are PPT machines. Therefore, there is a polynomial-time attack against $\mathcal{G}$ with a non-negligible advantage.

The second part of the argument for δ-exponential security goes the same way by replacing the *polynomial* $p(n)$ with $p(n) = 2^{o(n^\delta)}$ the value $\mathsf{negl}(n)$ with $q(n)\varepsilon(n) \leq 2^{O(n^\delta)}2^{-\omega(n^\delta)} \leq 2^{-\omega(n^\delta)}$. □

5 Unique-Witness One-Way Relations

A *one-way relation* $\mathcal{R} = (\mathsf{Gen}, \mathsf{Ver})$ consists of a PPT sampling algorithm $(y, x) \leftarrow \mathsf{Gen}(1^n)$ that samples an *instance* $y \in \{0,1\}^{m(n)}$ along with a *witness* $x \in \{0,1\}^{k(n)}$, and a deterministic poly-time verification algorithm $\mathsf{Ver}(y, x) \in \{0,1\}$ checks whether the pair (y, x) satisfies the relation. For correctness, we require that $\mathsf{Ver}(y, x) = 1$ for all (y, x) output by $\mathsf{Gen}(1^n)$. Intuitively, one-wayness says that given a randomly generated instance y it should be hard to find a valid witness x' such that $\mathsf{Ver}(y, x') = 1$. Leakage-resilient one-wayness requires that the above holds even if the attacker can gets some leakage on the original witness x for y.

Definition 5.1 (LR-OWR)**.** *We say that a relation $\mathcal{R} = (\mathsf{Gen}, \mathsf{Ver})$ is an $\ell(\cdot)$-*leakage-resilient one-way relation (ℓ-LR-OWR) *if for every* PPT *attacker $\mathcal{A} = (\mathsf{Leak}, \mathsf{Break})$ such that the output domain of $\mathsf{Leak}(1^n, \cdot)$ is $\{0,1\}^{\ell(n)}$, we have:*

$$\Pr\left[\ \mathsf{Ver}(y, x') = 1 \ \middle|\ \begin{array}{c}(y, x) \leftarrow \mathsf{Gen}(1^n)\\ z \leftarrow \mathsf{Leak}(1^n, x), x' \leftarrow \mathsf{Break}(1^n, y, z)\end{array}\right] = \mathsf{negl}(n).$$

We say that the relation has unique witnesses *if for every $y \in \{0,1\}^{m(n)}$ there exists* at most a single *$x \in \{0,1\}^{k(n)}$ such that $\mathsf{Ver}(y, x) = 1$.*

Remarks. The idea of leakage-resilient one-way relations is implicit in essentially all leakage-resilient cryptosystems since, for example, the relation between public and secret keys has to satisfy this definition. One simple observation is that *any* standard one-way relation without leakage is also secure against $\ell = O(\log(n))$ bits of leakage, since such small amount of leakage can just be guessed with good probability. Actually, this "guessing argument" even allows us to prove security for larger values of $\ell(n) = O(n^\delta)$, if we assume the δ-exponential security of the one-way relation without leakage.[2] Most of the work on leakage-resilient cryptosystems (starting with [AGV09]) is about clever ways of beating this trivial "guessing argument" and achieving leakage-resilience for large polynomial values of $\ell(n)$ without making exponential hardness assumptions. For example, [ADW09b] describes a very simple construction of an $\ell(n)$-leakage-resilient one-way functions/relations for *any* arbitrarily large polynomial $\ell(n)$, under only the assumption that standard one-way functions exist.[3]

When it comes to one-way relations with *unique witnesses*, we can construct these in the setting without leakage assuming the existence of injective one-way functions. Therefore, we can also apply the "guessing argument" to get such relations with leakage $\ell = O(\log(n))$ under the same assumption, or $\ell = O(n^\delta)$ under a δ-exponential version. However, unlike the previous case without unique witnesses, we do not know of any clever techniques that beat the trivial guessing argument and allow for larger values of ℓ without exponential assumptions. Here, we show that indeed this is impossible under black-box reductions from cryptographic-game assumptions. In other words, we show that the trivial "guessing argument" is essentially tight.

Theorem 5.2. *For any relation $\mathcal{R} = (\mathsf{Gen}, \mathsf{Ver})$ with unique witnesses, and for any leakage-bound $\ell(\cdot)$, there exists a $2^{-\ell(n)}$-simulatable attack against the ℓ-LR-OWR security of $\mathcal{R}$.*

Proof Idea. We construct an inefficient attacker $\mathcal{A} = (\mathsf{Leak}, \mathsf{Break})$ where $z = \mathsf{Leak}(x)$ is just a random function with ℓ-bit output and $\mathsf{Break}(y, z)$ does an exhaustive search over all possible values to find an x' s.t. $\mathsf{Ver}(y, x') = 1$ and $\mathsf{Leak}(x') = z$: if it finds one, it outputs x' and else $\perp$. Now the only way that the reduction can get something useful from $\mathsf{Break}(y, z)$ is by giving it the correct value $z = \mathsf{Leak}(x)$ for the unique witness x of y (if any exists). But the only way that it can come up with such value is by having previously made a call to $\mathsf{Leak}(x)$ *or by guessing it.* Therefore, we can have a simulator Sim that keeps state and responds to "leak" queries x with random ℓ-bit outputs and to "break" queries (y, z) by checking if the correct witness was given in a previous "leak" query. The simulation matches the outputs of $\mathcal{A}$ up to the $q2^{-\ell}$ probability of the reduction guessing the correct leakage z without querying it within the its q queries to the Break oracle. We translate this idea into a formal proof below.

Proof of Theorem 5.2. Let $\mathcal{H}_n$ be the set of all functions $h : \{0,1\}^{k(n)} \to \{0,1\}^{\ell(n)}$. We define an inefficient class of attackers $\{\mathcal{A}_{n,h} = (\mathsf{Leak}_{n,h}, \mathsf{Break}_{n,h})\}_{n\in\mathbb{N}, h\in\mathcal{H}_n}$ as follows:

$\mathsf{Leak}_{n,h}$ **:** On input x, output $h(x)$.

$\mathsf{Break}_{n,h}$**:** On input (y, z), do an exhaustive search to find $x \in \{0,1\}^{k(n)}$ such that $\mathsf{Ver}(y, x) = 1$. If such x is found and $h(x) = z$ output x, else output $\perp$.

It is easy to see that for each $h \in \mathcal{H}_n$, the attacker $\mathcal{A}_{n,h}$ inefficiently breaks the ℓ-LR one-wayness security of $\mathcal{R} = (\mathsf{Gen}, \mathsf{Ver})$ with advantage 1.

The more interesting property is the second part of the definition showing that, when $h \xleftarrow{\$} \mathcal{H}_n$ is chosen as a uniformly random function, getting oracle access to $\mathcal{A}_{n,h} = (\mathsf{Leak}_{n,h}, \mathsf{Break}_{n,h})$ is *useless* and can be efficiently simulated. Recall that we must construct a simulator $\mathsf{Sim}(1^n)$ such that for any

[2] This "guessing argument" can also be used to similarly prove the leakage-resilience of most other primitives (e.g., signatures, CPA encryption, etc.) with similar parameters.

[3] All these positive results even achieve a stronger definition where the leakage function $\mathsf{Leak}(1^n, x, y)$ also gets the statement y as an input. Since the focus here is on a negative result, we use the weaker and simpler definition stated above.

(possibly inefficient) probabilistic oracle-access machine $\mathcal{M}^{(\cdot)}(1^n)$ making at most $q(n)$ queries in total to its oracle(s), we have:

$$\left| \Pr_{h,\text{coins}(\mathcal{M})}[\mathcal{M}^{(\mathsf{Leak}_{n,h},\mathsf{Break}_{n,h})}(1^n)=1] - \Pr_{\text{coins}(\mathcal{M},\mathsf{Sim})}[\mathcal{M}^{\mathsf{Sim}(1^n)}(1^n)=1] \right| \leq q(n)2^{-\ell(n)}. \tag{3}$$

The simulator $\mathsf{Sim}(1^n)$ is stateful and randomized, and gets to respond to both "leak" and "break" queries in a coordinated manner (as compared to $\mathcal{A}_{n,h} = (\mathsf{Leak}_{n,h}, \mathsf{Break}_{n,h})$ which is stateless and responds to leakage and break queries independently). The simulator works as follows:

- Initialization: Initialize the set $Q := \emptyset$.
- Leakage: On a *leakage* query x, check if, for some z there is already a tuple $(x, z) \in Q$ and, if so, return z. Else select $z \xleftarrow{\$} \{0,1\}^{\ell(n)}$ at random, add the tuple (x, z) to Q and output z.
- Break: On a *break* query (y, z), check if there is a tuple $(x, \hat{z}) \in Q$ for the (unique) witness x s.t. $\mathsf{Ver}(y, x) = 1$. If so, and $\hat{z} = z$ return x. Else return $\perp$.

We now want to show indistinguishability of oracle access to $(\mathsf{Leak}_{n,h}, \mathsf{Break}_{n,h})$ when $h \xleftarrow{\$} \mathcal{H}_n$ is random, and oracle access to $\mathsf{Sim}(1^n)$. First, let us define an event E which occurs if, during the execution $\mathcal{M}^{(\mathsf{Leak}_{n,h},\mathsf{Break}_{n,h})}(1^n)$, the machine $\mathcal{M}$ makes some "break" query (y, z) such that there has been no prior "leak" query on the unique x for which $\mathsf{Ver}(y, x) = 1$, and the oracle outputs x. Then the query that causes E to occur must "guess" the random value $z = h(x)$ for the unique witness x for which $\mathsf{Ver}(y, x) = 1$. The probability that this occurs in any of the $q = q(n)$ queries made by $\mathcal{M}$ is at most $q(n)2^{-\ell(n)}$. Now as a hybrid experiment, let us consider the oracle Hyb that works just like $(\mathsf{Leak}_{n,h}, \mathsf{Break}_{n,h})$ but, on any "break" query that causes E to occur, it just automatically responds with $\perp$. Then

$$\left| \Pr_{\text{coins}(\mathcal{M}),h}[\mathcal{M}^{(\mathsf{Leak}_{n,h},\mathsf{Break}_{n,h})}(1^n)=1] - \Pr_{\text{coins}(\mathcal{M}),h}[\mathcal{M}^{\mathsf{Hyb}}(1^n)=1] \right| \leq \Pr[E] \leq q(n)2^{-\ell(n)}.$$

Lastly, we claim that $\Pr_{\text{coins}(\mathcal{M}),h}[\mathcal{M}^{\mathsf{Hyb}}(1^n)=1] = \Pr_{\text{coins}(\mathcal{M},\mathsf{Sim})}[\mathcal{M}^{\mathsf{Sim}(1^n)}(1^n)=1]$. In particular, the simulator $\mathsf{Sim}(1^n)$ and Hyb work the same way, except that the simulator chooses the outputs of h "on-the-go" as needed to answer leakage queries and break queries. Together the two equations above prove equation (3), concluding the proof. □

As an immediate corollary of the above theorem and Theorem 4.2 we get the following result.

Corollary 5.3. *For any $\ell(n) = \omega(\log(n))$, if there is a black-box reduction showing the ℓ-LR-OWR security of a relation $\mathcal{R}$ with unique witnesses based on the security of some cryptographic game $\mathcal{G}$, then $\mathcal{G}$ is not secure. Moreover, for any $\ell(n) = n^\delta$ with constant $\delta > 0$, if there is such a black-box reduction based on the δ'-exponential security of $\mathcal{G}$ with $\delta' < \delta$ then $\mathcal{G}$ is not δ'-exponentially secure.*

This essentially says that the only way to get $\ell(n)$-leakage-resilience is to assume the $\ell(n)$-exact security of some cryptographic game, meaning that the "guessing" argument is essentially optimal in proving leakage-resilience for unique-witness one-way relations.

Consequences and Implications. Since one-way relations are implicit in all constructions of more advanced cryptosystems, the above theorem gives several interesting implications. Most obviously, it implies similar separations for leakage-resilient *injective* one-way functions and permutations since they immediately give us one-way relations with unique witnesses. Interestingly, the separation does *not* extend to the leakage-resilience of injective *function families* $\mathcal{F} = \{f_{pk}\}$ where each function in the family is described by some $pk \leftarrow \mathsf{Gen}(1^n)$. This is because the corresponding relation $\{(pk, y), x \ : \ y = f_{pk}(x)\}$ with instances (pk, y) and witnesses x does not necessarily have unique witnesses – there may be "invalid"

keys pk which are *not* in the support of Gen for which f_{pk} is not injective. Indeed, the notion of *lossy (trapdoor) functions* [PW08] can be used to construct such injective function families $\mathcal{F}$ with arbitrarily high polynomial leakage ℓ under standard assumptions (DDH, LWE, and others). However, if we were to insist that the function family $\mathcal{F}$ is *certifiably injective*, meaning that we can efficiently verify that pk is a "valid" public key for which the function $f_{pk}(\cdot)$ is injective, then the resulting relation does have unique witnesses and the separation result kicks in. A similar phenomenon occurs in essentially all leakage-resilient *public-key* cryptosystems (e.g., encryption, signatures) where we insist that there is a unique secret key for every public-key. If we insist that the valid public keys of the scheme are efficiently recognizable, then the resulting relation between public and secret keys has unique witnesses and our separation result applies. As another corollary, we also get a similar separation for leakage-resilient *unique signatures* [Lys02] where, for every public key pk (even an adversarially chosen one) and message m, there should be a unique signature that verifies for pk.

Lastly in Appendix Appendix A, we show a similar separation for injective one-way functions where, instead of allowing the attacker to observe leakage on the input x, we allow the attacker to choose any efficiently samplable *weak source* of sufficient min-entropy from which the input x is then sampled. The style of the separation and its parameters are very similar to the leakage case.

6 Deterministic Encryption and Correlated Input Security

Let $\mathcal{F} \stackrel{\text{def}}{=} \{ \ f_{pk} \ : \ \{0,1\}^{k(n)} \to \{0,1\}^{m(n)} \ : \ pk \in \{0,1\}^{p(n)} \}_{n\in\mathbb{N}}$ be some family of efficiently computable functions keyed by a public pk. Let $pk \leftarrow \mathsf{Gen}(1^n)$ be a PPT sampling algorithm for choosing pk. We say that $(\mathcal{F}, \mathsf{Gen})$ is an *injective function family* if for *every pk in the support of* Gen the function f_{pk} is *injective*. Note that we make no requirements on f_{pk} for other values of pk which are not in the support of Gen. In this section, we focus on an adversary that can observe evaluations $f_{pk}(x_1), \ldots, f_{pk}(x_t)$ for several adversarially distributed correlated inputs $x_1, \ldots, x_t$. We only make the restriction that (1) the distribution of $(x_1, \ldots, x_t)$ is independent of pk, (2) each input x_i individually has high entropy and (3) the inputs are distinct. We ask for a simple one-wayness property, that the attacker should be unable to recover all of the x_i values. We show that any candidate scheme has a simulatable attack against this property, implying that we will not be able to prove it via black-box reductions.

Relation to One-Wayness with Weak Keys. Note that we do not require that the set of valid pk to be efficiently recognizable. In particular, there may be other values pk (not in the support of Gen) such that the functions f_{pk} are *not* injective. Therefore the relation $\{((pk, y), x) \ : \ f_{pk}(x) = y\}$ may not have *unique witnesses* and the results from the previous section and from Appendix A do not apply in any meaningful way to $\mathcal{F}$. Indeed, if $\mathcal{F}$ is a lossy trapdoor function family, we are able to prove that (e.g.) the functions f_{pk} remain one-way when evaluated on a single input x, even if x comes from an adversarial source of sufficient min-entropy.

Relation to Deterministic Encryption. The work of [BBO07] studies a somewhat harder problem in the context of *deterministic encryption* where the functions $f_{pk}(x)$ are used as encryptions of the message x. In that context, one needs to make an additional requirement that pk can be sampled with a *decryption trapdoor sk* allowing the recovery of x from $f_{pk}(x)$. Furthermore, the security definition of deterministic encryption is fairly subtle but certainly (and significantly) stronger than one-wayness property that we consider here. The work of [BBO07] was able to achieve this strong notion of security in the *random-oracle model*, assuming the existence of any standard semantically-secure (randomized) public-key encryption scheme. In particular, it shows that $f_{pk}(x) \stackrel{\text{def}}{=} \mathsf{Enc}_{pk}(x; H(x))$ satisfies this definition when H is modeled as a random oracle. This gave a deterministic encryption scheme which could be used to encrypt any message distribution satisfying properties (1),(2) and (3) above. Subsequent work on deterministic encryption [BFOR08, BFO08, BS11] gives constructions in the standard model, but only for

more restricted message distributions where each new message contains some fresh entropy conditioned on the previous ones. Our results therefore imply that some such restrictions on the message distribution are necessary if one wants to prove deterministic encryption security (or even just one-wayness) via black-box reductions.

Defining Correlation Resilience. Let us now make the above definition formal. An algorithm $\mathsf{Sam}(1^n)$ that samples $t(n)$-tuples $\mathbf{x} = (x_1, \ldots, x_t)$ with $x_i \in \{0,1\}^{k(n)}$ produces a (t, ℓ)-*legal correlated distribution* if it satisfies:

- *Distinct Inputs:* For any tuple output by Sam we have $x_i \neq x_j$ for all $i \neq j$.
- *Individual Entropy:* We have $\mathbf{H}_\infty(X_i) \geq k(n) - \ell(n)$, where X_i is a random variable for the ith component x_i produced by $\mathsf{Sam}(1^n)$.

Definition 6.1 (Correlation-Resilient One-Wayness (CROW))**.** *We say that* $(\mathcal{F}, \mathsf{Gen})$ *is* $(t(n), \ell(n))$-correlation-resilient one way *(CROW) if for any* PPT *attacker* $\mathcal{A} = (\mathsf{Sam}, \mathsf{Break})$ *such that* Sam *it a* (t, ℓ)*-legal correlated distribution, we have:*

$$\Pr\left[\mathbf{x}' = \mathbf{x} \;\middle|\; \begin{array}{c} pk \leftarrow \mathsf{Gen}(1^n), \mathbf{x} = (x_1, \ldots, x_t) \xleftarrow{\$} \mathsf{Sam}(1^n), \\ \mathbf{y} = (f_{pk}(x_1), \ldots, f_{pk}(x_t)), \quad \mathbf{x}' \leftarrow \mathsf{Break}(1^n, pk, \mathbf{y}) \end{array} \right] = \mathsf{negl}(n).$$

We say that $(\mathcal{F}, \mathsf{Gen})$ *is* ℓ*-CROW if it is* (t, ℓ)*-CROW for all polynomial* $t(n)$*.*

Theorem 6.2. *For any injective function family* $(\mathcal{F}, \mathsf{Gen})$ *with input size* $k(n)$ *there exists an* $\varepsilon(n)$*-simulatable attack against its* (t, ℓ)*-CROW security for any* $\ell(n) > \log(t(n))$ *with* $\varepsilon(n) \leq 2^{-(t(n)-2k(n))/2+1}$*.*

Proof Intuition. We choose $\bar{h}$ to be a random function which maps 'short'($\approx k$ bit) values z to 'long' tuples $\mathbf{x} = (x_1, \ldots, x_t)$, and define the inefficient sampler $\mathbf{x} \leftarrow \mathsf{Sam}(1^n)$ by having it choose z at random and output $\bar{h}(z)$. With some additional restrictions on the structure of $\bar{h}$, we can ensure that this always yields a (t, ℓ)-legal correlated distribution over the choice of z. Now assume the reduction makes several "Sample" queries and then comes up with some pk and some $\mathbf{y} = (y_1, \ldots, y_k)$. Then we claim that one of the following must hold: either (1) the function $f_{pk}(\cdot)$ is extremely degenerate and (say) half of all inputs x will map to the same output $y = f_{pk}(x)$, or (2) it is extremely unlikely (over the choice of $\bar{h}$) that there is some new value z, which wasn't used by a prior sample query, such that $\mathbf{x} = \bar{h}(z)$ is a pre-image of $\mathbf{y}$ under f_{pk}. The reason is that there are only a few choice of z, each corresponding to some long tuple $\mathbf{x} = (x_1, \ldots, x_t)$ which is uniformly random over the choice of $\bar{h}$. Therefore its extremely unlikely that even one such new z will correspond to a preimage of $\mathbf{y}$ unless f_{pk} is degenerate. So, unless pk is degenerate, getting access to a breaker $\mathsf{Break}(pk, \mathbf{y})$ that does exhaustive search over z to find a preimage $\mathbf{x} = \bar{h}(z)$ is not useful since it will only return preimages that were previously given by Sam. On the other hand, we can efficiently test if pk is degenerate and so can allow $\mathsf{Break}(pk, \mathbf{y})$ to just fail on degenerate pk while maintaining the ability to simulate it. We formalize this intuition in a formal proof.

Proof of Theorem 6.2. We use $t = t(n), k = k(n)$ etc. as shorthand. Let $k' \stackrel{\text{def}}{=} k - \lceil \log(t) \rceil$ and identify tuples $x = (i, x') \in [t] \times \{0,1\}^{k'}$ as members of $\{0,1\}^k$ in the natural way. For every $n \in \mathbb{N}$ let $\mathcal{H}_n$ be the family of all functions $h : [t(n)] \times \{0,1\}^{k'(n)} \to \{0,1\}^{k'(n)}$ such that, for every $i \in [t]$, the projected function $h_i(z) \stackrel{\text{def}}{=} h(i, z)$ is a *permutation* over $\{0,1\}^{k'}$. For any such $h \in \mathcal{H}_n$, we also implicitly define the function

$$\bar{h}(\cdot) : \{0,1\}^{k'} \to \{0,1\}^{k \times t} \text{ defined by } \bar{h}(z) \stackrel{\text{def}}{=} ((1, h_1(z)), \ldots, (t, h_t(z))).$$

For any $h \in \mathcal{H}_n$, we define an inefficient class of attackers $\mathcal{A}_{n,h} = (\mathsf{Sam}_{n,h}, \mathsf{Break}_{n,h})$ as follows:

$\mathsf{Sam}_{n,h}$ **:** On input 1^n, sample $z \xleftarrow{\$} \{0,1\}^{k'}$, and output the vector $\mathbf{x} = (x_1, \ldots, x_t) = \bar{h}(z)$.

$\mathsf{Break}_{n,h}$**:** On input $(1^n, pk, \mathbf{y} = (y_1, \ldots, y_t))$:

1. First check that $f_{pk}(\cdot)$ is *non-degenerate* by performing the following check for each $i \in [t]$: Sample $r = 2t$ random distinct points $x'_1, \ldots, x'_r$ from $\{0,1\}^{k'}$ and compute $y'_j = f_{pk}((i, x'_j))$ for $j = 1, \ldots, r$. If $y'_1, \ldots, y'_r$ are not all distinct, output $\perp$.
2. Do an exhaustive search over $z \in \{0,1\}^{k'}$ for one that satisfies $\bar{h}(z) = \mathbf{x} = (x_1, \ldots, x_t)$ such that $f_{pk}(x_i) = y_i$ for all $i \in [t]$. When the first such z is found, output the corresponding $\mathbf{x}$ and if no such z exists output $\perp$.

Firstly, we check that for each $h \in \mathcal{H}$, the attacker $\mathcal{A}_h = (\mathsf{Sam}_{n,h}, \mathsf{Break}_{n,h})$ breaks the (t,ℓ)-CROW security of $(\mathcal{F}, \mathsf{Gen})$ with advantage 1. Notice that the distribution $\mathbf{x} = (x_1, \ldots, x_t) \leftarrow \mathsf{Sam}_{n,h}(1^n)$ is a (t,ℓ)-legal correlated distribution since: (a) the value $x_i = (i, h_i(z))$ have entropy $k' = k - \lceil \log(t) \rceil \geq k - \ell$ over a random choice of z since h_i is a permutation and (b) $x_i \neq x_j$ for $i \neq j$. Moreover, when calling $\mathsf{Break}_{n,h}$ on $pk \leftarrow \mathsf{Gen}(1^n)$ the non-degeneracy check always passes since f_{pk} is injective. Hence $\mathsf{Break}_{n,h}$ will succeed in recovering the unique pre-image $\mathbf{x}$ of $\mathbf{y}$ under f_{pk} with probability 1.

We now want to show that, when $h \xleftarrow{\$} \mathcal{H}_n$ is chosen uniformly at random, getting oracle access to $\mathcal{A}_h = (\mathsf{Sam}_{n,h}, \mathsf{Break}_{n,h})$ is *not very useful* and can be efficiently simulated.

We construct our simulator $\mathsf{Sim}(1^n)$ as follows:

Initialization: Initialize the set $Q := \emptyset$.

Sam: On a *sample* query, select random $z \xleftarrow{\$} \{0,1\}^{k'}$. If there is a value $(z, \mathbf{x})$ in Q output $\mathbf{x}$. Else, for $i \in [t]$, let Q_i be the set of all x s.t. there is a tuple of the form $(z, \mathbf{x} = (\cdots, (i,x), \cdots)) \in Q$. Choose $\{x_i \xleftarrow{\$} \{0,1\}^{k'} \setminus Q_i\}_{i \in [t]}$, set $\mathbf{x} = ((1,x_1), \ldots, (t,x_t))$, add $(z, \mathbf{x})$ to Q, and output $\mathbf{x}$.

If $|Q| \geq 2^{k'}/4$ then "fill in" the rest of Q by choosing $\mathbf{x}$ as above sequentially for every $z \in \{0,1\}^{k'}$ which is not in Q yet.[4]

Break: On a *break* query $(pk, \mathbf{y} = (y_1, \ldots, y_t))$.

- First check that $f_{pk}(\cdot)$ is *non-degenerate* using the same process as the oracle $\mathsf{Break}_{n,h}$. That is, for each $i \in [t]$, sample $r = 2t(n)$ random distinct points $x'_1, \ldots, x'_r$ from $\{0,1\}^{k'}$ and compute $y'_j = f_{pk}((i, x'_j))$ for $j = 1, \ldots, r$. If $y'_1, \ldots, y'_r$ are not all distinct, output $\perp$.
- Check if there is a tuple $(z, \mathbf{x} = (x_1, \ldots, x_t)) \in Q$ such that $f_{pk}(x_i) = y_i$ for all $i \in [t]$. If so, output $\mathbf{x}$, else output $\perp$.

For the analysis of the simulation, we define a public-key pk as *degenerate* if there is some $i \in [t], y \in \{0,1\}^m$ such that $\Pr_{x' \xleftarrow{\$} \{0,1\}^{k'}}[f_{pk}((i,x')) = y] \geq \frac{1}{2}$. In other words pk is degenerate if f_{pk} is very much non-injective and half all values of the form (i,x') map to the same y. Let δ be the probability that the non-degeneracy test passes for some degenerate pk. Then this only occurs if during the ith iteration of the test, at most 0 or 1 of the x'_j values map to $f_{pk}(i, x'_j) = y$. This occurs with probability $\delta \leq 2^{-r} + r2^{-r+1} \leq 2^{-t}$.

We now want to show indistinguishability of oracle access to $(\mathsf{Sam}_{n,h}, \mathsf{Break}_{n,h})$ when h is random, and oracle access to $\mathsf{Sim}(1^n)$:

$$\left| \Pr_{h, \mathrm{coins}(\mathcal{M})}[\mathcal{M}^{(\mathsf{Sam}_{n,h}, \mathsf{Break}_{n,h})}(1^n) = 1] - \Pr_{\mathrm{coins}(\mathcal{M}, \mathsf{Sim})}[\mathcal{M}^{\mathsf{Sim}(1^n)}(1^n) = 1] \right| \leq q(n) 2^{-(t-2k)/2+1}. \quad (4)$$

[4]The work here is polynomial in the size of the simulator's current state. In other words, amortized over the number of queries, this only takes a polynomial amount of work per query.

First, let us define an event E which occurs if, during the execution $\mathcal{M}^{(\mathsf{Sam}_{n,h},\mathsf{Break}_{n,h})}(1^n)$ with $h \xleftarrow{\$} \mathcal{H}_N$ random, the machine $\mathcal{M}$ makes some "Break" query $(pk, \mathbf{y})$ and:

- It gets a response $\mathbf{x} \neq \perp$ such that $\mathbf{x}$ was *not* the outputs of a previous "Sam" query.
- The number of previous "Sam" queries made by $\mathcal{M}$ is $< 2^{k'}/4$.

Now we claim that the simulation error (the left-hand side of (4)) is at most $\Pr[E]$. To see this, we can define an attacker $\mathcal{A}'_{n,h} = (\mathsf{Sam}'_{n,h}, \mathsf{Break}'_{n,h})$ which acts like $\mathcal{A}_{n,h}$ but outputs $\perp$ on any query that makes E occur. Then we claim that the responses of $\mathcal{A}'_{n,h}$ for $h \leftarrow \mathcal{H}_n$ and $\mathsf{Sim}(1^n)$ are identical since the only (syntactic) difference between them is that $\mathsf{Sim}(1^n)$ chooses the outputs of the function h "on-the-go" for the first $2^{k'}/4$ queries to "Sam". Therefore, we are only left to find an upper bound on $\Pr[E]$.

Let E_i be the event that the ith "Break" query makes E occur. Let D_i be the even that the value pk contained in the ith query $(pk, \mathbf{y})$ is degenerate. Then, $\Pr[E_i \wedge D_i] \leq \Pr[E_i \mid D_i] \leq \delta \leq 2^{-t}$. To calculate $\Pr[E_i \wedge \neg D_i]$, let Z be the set of all values $z \in \{0,1\}^{k'}$ used during previous "sample" queries before query i. Then, conditioned on the view of the attacker prior to the ith query, for each $z \notin Z$, the values of $\{h_j(z)\}_{j\in[t]}$ are mutually uniformly random over the sets $S_j = \{0,1\}^{k'} \setminus h_j(Z)$ of size $|S_j| = 2^{k'} - |Z|$. Moreover, if pk is non-degenerate then at most a $\left(\frac{1}{2} \cdot \frac{2^{k'}}{|S_j|}\right)$ fraction of the values $x \in S_j$ can satisfy $f_{pk}((j,x)) = y$ for any given y. Therefore, since we can assume $|Z| \leq 2^{k'}/4$, we get:

$$\begin{aligned}
\Pr[E_i \wedge \neg D_i] &\leq \Pr[E_i \mid \neg D_i] \leq \Pr_h\left[\ \exists z \in \{0,1\}^{k'} \setminus Z \text{ s.t. } \forall j \in [t] : h_j(z) = y_j\ \right] \\
&\leq 2^{k'}\left(\frac{2^{k'}}{2\cdot(2^{k'} - |Z|)}\right)^t \leq 2^{k'}(\frac{2}{3})^t \leq 2^k(\frac{4}{9})^{k+(t-2k)/2} \leq 2^{-(t-2k)/2}
\end{aligned}$$

So the statistical distance of the simulation is bounded by

$$\Pr[E] \leq \sum_{i=1}^{q(n)} \Pr[E_i] \leq q(n)(2^{-(t-2k)/2} + 2^{-t}) \leq q(n)2^{-(t-2k)/2+1}$$

which concludes the proof of the theorem. □

As an immediate corollary of the above theorem and Theorem 4.2 we get the following.

Corollary 6.3. *Let $(\mathcal{F}, \mathsf{Gen})$ be an injective function family with input size $k(n)$:*

- *If $t(n) \geq 2k(n) + \omega(\log(n))$, $\ell(n) > \log(t(n))$ then there is no black-box reduction showing (t,ℓ)-CROW security of $(\mathcal{F}, \mathsf{Gen})$ based on the security of some cryptographic game $\mathcal{G}$, unless $\mathcal{G}$ is insecure.*
- *If $t(n) \geq 2k(n) + n^\delta$, $\ell(n) > \log(t(n))$, then there is no black-box reduction showing (t,ℓ)-CROW security of $(\mathcal{F}, \mathsf{Gen})$ based on the δ-exponential security of $\mathcal{G}$, unless $\mathcal{G}$ is not δ-exponentially secure.*
- *There is no black-box reduction showing ℓ-CROW security of $(\mathcal{F}, \mathsf{Gen})$ for $\ell(n) = \omega(\log(n))$ under the δ-exponential security of some game $\mathcal{G}$ for some $\delta > 0$, unless $\mathcal{G}$ is not δ-exponentially secure.*

7 Pseudo-Entropy Generation

If x is a random string, and the attacker may get some ℓ-bit leakage $z = \mathsf{Leak}(x)$ about x, then the *conditional entropy* of x conditioned on z may go down by as much as ℓ bits (see Lemma 2.1). Here we ask if we can "increase" the *computational entropy* of x by applying some deterministic function $y = \mathsf{PEG}(x)$, called a *pseudo-entropy generator* (PEG). Since the leakage z may contain (say) the first ℓ bits of y, the conditional computational entropy of y conditioned on z is at most $|y| - \ell$ (under any reasonable

definition). However, if $|y| > |x|$, it is possible that the function PEG *increases* the amount of conditional computational entropy from $|x| - \ell$ possibly all the way to $|y| - \ell$. Indeed, when $\ell = 0$, this is exactly the role of a *pseduorandom generator*. Therefore a PEG generalizes the notion of a pseudorandom generator to weaker entropy requirements on both the input and output. The existence of PEGs with essentially optimal parameters follows easily in the random-oracle model. In the standard model, there has been much interesting work, starting with [DP08, RTTV08], showing that *any* pseduorandom generator is also a PEG for small values of $\ell = O(\log(n))$, or for larger value $\ell = n^{\delta}$ assuming exponential security. Unlike the trivial "guessing argument" used to prove such bounds for leakage-resilient one-way relations, showing the above is highly non-trivial. We now ask if there are clever constructions of PEGs which provably beat the above bounds and allow us to prove security for $\ell = \omega(\log(n))$ under standard (non-exponential) assumptions. We show that the answer is negative if we restrict ourselves to black-box reductions.

We now give a formal definition of PEGs. Since the focus is on negative results, we give a weak definition using a weak variant of *conditional metric entropy* [BSW03]. We also only require that the PEG increases computational entropy by a single bit, from $|x| - \ell$ to $|x| - \ell + 1$.

Definition 7.1. *Let* $\mathsf{PEG} = \{\mathsf{PEG}_n : \{0,1\}^{k(n)} \to \{0,1\}^{m(n)}\}_{n\in\mathbb{N}}$ *be a function ensemble with* $m(n) > k(n)$*. We say that it is an* $\ell(\cdot)$-leaky pseudo-entropy generator *(*ℓ*-LPEG) if the following holds. For every* PPT *attacker* $\mathcal{A} = (\mathsf{Leak}, \mathsf{Dist})$ *such that the output domain of* $\mathsf{Leak}(1^n, \cdot)$ *is* $\{0,1\}^{\ell(n)}$*, there exists some pair of (correlated and not necessarily efficiently samplable) random variables* $\{(Y_n, Z_n)\}_{n\in\mathbb{N}}$ *of high* statistical *conditional entropy* $\mathbf{H}_{\infty}(Y_n \mid Z_n) \geq k(n) - \ell(n) + 1$ *and the distinguishing advantage* $\mathbf{Adv}^{\mathcal{A}}_{LPEG}(n) \leq \mathsf{negl}(n)$ *where*

$$\mathbf{Adv}^{\mathcal{A}}_{LPEG}(n) \stackrel{\text{def}}{=} \left| \Pr_{x \stackrel{\$}{\leftarrow} \{0,1\}^{k(n)}} [\mathsf{Dist}(1^n, \mathsf{PEG}(x), \mathsf{Leak}(x)) = 1\ ] \quad - \quad \Pr_{(y,z)\leftarrow(Y_n,Z_n)} [\mathsf{Dist}(1^n, y, z) = 1] \right|.$$

Theorem 7.2. *For any function* PEG *and for any* $\ell(\cdot)$*, there exists a* $(\varepsilon(n) = 2^{-\ell(n)})$*-simulatable attack against the* ℓ*-LPEG security of* PEG *.*

Proof Intuition. We define the inefficient simulatable attack $\mathcal{A} = (\mathsf{Leak}, \mathsf{Dist})$ as follows. The function $\mathsf{Leak}(x)$is defined by a uniformly random function $h : \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}$ and just outputs $h(\mathsf{PEG}(x))$. On the other hand $\mathsf{Dist}(y, z)$ outputs 1 iff there is a seed x with $y = \mathsf{PEG}(x)$ and $h(y) = z$. Now Dist will always output 1 on pseduorandom inputs with correct leakage, but for any distribution (Y, Z) with sufficient statistical conditional entropy, it will often output 0 since there will not be a seed x explaining the tuple. Therefore this is a valid attack. On the other hand, the only way that a reduction can call $\mathsf{Dist}(y, z)$ and get output 1 for any "new" y for which it did not make a prior call to $\mathsf{Leak}(x)$ with $x \in \mathsf{PEG}^{-1}(y)$, is if the reduction guesses the random ℓ-bit value $h(y)$, which occurs with probability $2^{-\ell}$. Therefore, we can efficiently simulate the attack for the reduction.

Proof of Theorem 7.2. Define $\mathcal{Y}_n \stackrel{\text{def}}{=} \{\mathsf{PEG}(x) : x \in \{0,1\}^{k(n)}\} \subseteq \{0,1\}^{m(n)}$. Let $\mathcal{H}_n$ be the family of all functions $h : \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}$ which are *almost regular over* $\mathcal{Y}_n$, meaning that for all $z \in \{0,1\}^{\ell(n)}$ we have $|h_n^{-1}(z) \cap \mathcal{Y}_n| \leq 2^{k(n)-\ell(n)}$. We define an inefficient class of attackers $\mathcal{A}_{n,h} = (\mathsf{Leak}_{n,h}, \mathsf{Dist}_{n,h})$ with respect to $h \in \mathcal{H}_n$ as follows:

$\mathsf{Leak}_{n,h}$ **:** On input $x \in \{0,1\}^{k(n)}$ output $h_n(\mathsf{PEG}(x))$.

$\mathsf{Dist}_{n,h}$**:** On input y, z, first check that $h(y) = z$ and output 0 if not. Otherwise, do an exhaustive search to find $x \in \{0,1\}^{k(n)}$ such that $\mathsf{PEG}(x) = y$. If x exists output 1 else output 0.

We show that this attack has advantage 1/2. Firstly, given any fixed $h \in \mathcal{H}_n$, we have

$$\Pr_{x \stackrel{\$}{\leftarrow} \{0,1\}^{k(n)}} [\mathsf{Dist}_{n,h}(1^n, \mathsf{PEG}(x), \mathsf{Leak}_{n,h}(x)) = 1] = 1. \tag{5}$$

Secondly, given any r.v. (Y_n, Z_n) with $\mathbf{H}_\infty(Y_n \mid Z_n) \geq k(n) - \ell(n) + 1$, we claim

$$\Pr_{(y,z)\xleftarrow{\$}(Y_n,Z_n)}[\mathsf{Dist}_{n,h}(1^n, y, z) = 1] \leq \Pr[y \in h_n^{-1}(z) \cap \mathcal{Y}_n] \leq \frac{1}{2}. \tag{6}$$

To see the last inequality, let $\mathcal{B}(z)$ be an inefficient predictor that, given z, outputs $y \xleftarrow{\$} h^{-1}(z) \cap \mathcal{Y}_n$. Then

$$2^{-k(n)+\ell(n)-1} \geq \Pr[\mathcal{B}(z) = y] \geq \Pr[y \in h_n^{-1}(z) \cap \mathcal{Y}_n] 2^{-k(n)+\ell(n)}$$

where the left inequality follows from the definition of the conditional min-entropy (Y_n, Z_n), while the right follows by analyzing the strategy of $\mathcal{B}$. This proves equation (6), which together with (5) shows that $\mathcal{A}_{n,h}$ successfully attacks the $\ell(\cdot)$-LPEG security of PEG with advantage $\geq 1/2$.

Now let us show that, when $h \xleftarrow{\$} \mathcal{H}_n$ is chosen uniformly at random, getting oracle access to $\mathcal{A}_{n,h} = (\mathsf{Leak}_{n,h}, \mathsf{Dist}_{n,h})$ can be efficiently simulated. The simulator $\mathsf{Sim}(1^n)$ works as follows:

- Initialization: Initialize the set $Q := \emptyset$.
- Leakage: On a *leakage* query x check if there is already a tuple $(\mathsf{PEG}(x), z) \in Q$ and, if so, return z. Else select $z \xleftarrow{\$} \{0,1\}^{\ell(n)}$ at random, add the tuple $(\mathsf{PEG}(x), z)$ to Q and output z.
- Distinguish: On a *distinguish* query (y, z), check if $(y, z) \in Q$ – if so return 1 else return 0.

We now want to show indistinguishability of oracle access to $(\mathsf{Leak}_{n,h}, \mathsf{Dist}_{n,h})$, and oracle access to $\mathsf{Sim}(1^n)$. That is, for every $\mathcal{M}$ making at most $q(n)$ queries to its oracle, we want to show:

$$\left| \Pr_{h,\mathrm{coins}(\mathcal{M})}[\mathcal{M}^{(\mathsf{Leak}_{n,h},\mathsf{Dist}_{n,h})}(1^n) = 1] - \Pr_{\mathrm{coins}(\mathcal{M},\mathsf{Sim})}[\mathcal{M}^{\mathsf{Sim}(1^n)}(1^n) = 1] \right| \leq (q^2(n) + q(n))2^{-\ell(n)}. \tag{7}$$

First let us define a hybrid oracle Hyb which works just like $\mathcal{A}_h = (\mathsf{Leak}_{n,h}, \mathsf{Dist}_{n,h})$ but instead of selecting $h \xleftarrow{\$} \mathcal{H}_n$, it selects a uniformly random function $\tilde{h} : \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}$, without insisting that it is almost regular. We claim that the statistical distance between q queries to $\mathcal{A}_h$ and Hyb is bounded by $q^2/2^\ell$. This is because, conditioned on the values of any i prior (distinct) queries to $\tilde{h}$ (resp. h), for any new input $y \in \mathcal{Y}_n$ and any $Z \subseteq \{0,1\}^\ell$ we have $\Pr[\tilde{h}(y) \in Z] = |Z|2^{-\ell}$ and $\Pr[h(y) \in Z] = \frac{|Z|2^{k-\ell} - i_Z}{2^k} = |Z|2^{-\ell} - i_Z/2^k$ where i_Z is the number of prior queries for which the response was in Z. Therefore the statistical distance between the responses to the ith query is at most $q2^{-k} \leq q(n)2^{-\ell}$. Under a hybrid argument over the q queries we get the bound $q^2 2^{-\ell}$ as desired.

We define an event E which occurs if, during the execution $\mathcal{M}^{\mathsf{Hyb}}(1^n)$, the machine $\mathcal{M}$ makes some "distinguish" query (y, z) such that it never made a prior leakage query on x with $\mathsf{PEG}(x) = y$, and it gets a response 1. Then, conditioned on E not occurring, the oracles Hyb and $\mathsf{Sim}(1^n)$ are identically distributed, and therefore we can bound the difference in (7) by $\Pr[E] + q^2 2^{-\ell}$. On the other hand, the query (y, z) that causes E to occur must "guess" the random value $z = \tilde{h}(y)$ which is uniformly random and of size $\ell(n)$. The probability that this occurs in any of the $q = q(n)$ queries made by $\mathcal{M}$ is at most $\Pr[E] \leq q(n)2^{-\ell(n)}$. Together these two bounds give us (7) and prove the theorem.

As an immediate corollary of the above theorem and Theorem 4.2 we get the following separation.

Corollary 7.3. *For any $\ell(n) = \omega(\log(n))$, if there is a black-box reduction showing the ℓ-LPEG security of some function PEG based on the security of some cryptographic game $\mathcal{G}$, then $\mathcal{G}$ is not secure. Moreover, for any $\ell(n) = n^\delta$ with constant $\delta > 0$, if there is a reduction based on the δ'-exponential security of $\mathcal{G}$ with $\delta' < \delta$ then $\mathcal{G}$ is not δ'-exponentially secure.*

We mention that the work of [BHK11] gives positive results for an interesting related notion called "leaky pseudo-entropy functions", which does not satisfy our definition of PEGs (Definition 7.1) and therefore does not contradict our barrier. In particular, although "pseudo-entropy functions" never increase the total amount of pseudo-entropy, they ensure that the output of the function at any point is unpredictable even give its outputs at many other points.

8 Entropy Condensation

Another tool studied in the context of manipulating weak sources of randomness is a *condenser*, originally defined in the information theoretic setting by [RR99], with the aim of increasing the *entropy rate* of a source X, defined as the ratio of its entropy to bit size. Unfortunately, much like randomness extractors, such information theoretic condensers for general sources require a random seed chosen independently of the source X. The recent work of [DRV12] suggested an interesting idea of building condensers for a restricted class of *efficiently samplable* sources, which are essentially the only sources we need to worry about in cryptography. Although these condensers (may) still require a seed, the source X can now depend on the seed. Intuitively, such condensers may be plausible since, the only way that an efficient high-entropy distribution on the inputs results in a lower-than-expected entropy distribution on the outputs is if it has a higher-then-expected probability of collisions, which may be hard to achieve efficiently. Indeed, [DRV12] showed that such condensers can be constructed from *collision-resistant hash functions* with sufficient (exponential) security, and gave several interesting cryptographic applications of such condensers. Here we show that, if we want black-box reductions, than there is no clever construction of such condensers which avoids the reliance on exponential security assumptions. In other words, the construction of [DRV12] is essentially tight.

Let $\mathsf{Cond} = \{\mathsf{Cond}_n : \{0,1\}^{k(n)} \times \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}\}_{n\in\mathbb{N}}$ be some function ensemble with $m(n) < k(n)$. We use the notation $\mathsf{Cond}_s(x)$ in place of $\mathsf{Cond}(x,s)$ and call the value s a *seed*.

Definition 8.1 (Seed-Dependent Condenser)**.** *We say that* Cond *is a* $t(\cdot)$*-condenser if for all* PPT *distributions* $\mathsf{Sam}(1^n, s)$ *over* $\{0,1\}^{k(n)}$ *satisfying* $\mathbf{H}_\infty(\mathsf{Sam}(1^n,s)) \geq k(n) - m(n) + t(n)$ *for all* $s \in \{0,1\}^{d(n)}$*, we have* $\mathbf{H}_\infty(\mathsf{Cond}_S(\mathsf{Sam}(1^n,S)) \mid S) \geq t(n)+1$*, where* S *is a uniformly random seed in* $\{0,1\}^{d(n)}$*. A condenser is* regular *if, for every* $s \in \{0,1\}^{d(n)}, y \in \{0,1\}^{m(n)}$ *we have* $|\{x : \mathsf{Cond}_s(x) = y\}| = 2^{k(n)-m(n)}$*.*

Notice that the function Cond which just outputs the first m bits of its input x and ignores the seed s entirely already has output entropy $\geq t$ if the input entropy is $\geq k - m + t$. This is because cutting the last $k-m$ bits of the input can decrease its entropy by at most $k-m$. Therefore, our definition of a condenser only requires us to beat the above trivial construction by only a single bit!

Theorem 8.2. *For any* Cond *which is regular, and for any* $t(\cdot)$*, there exists an* $2^{-t(n)}$*-simulatable attack against the* $t(n)$*-condenser security of* Cond*.*

Proof. Let $H \subseteq \{0,1\}^{m(n)}$ be a sets of size $|H| = 2^{t(n)}$. Let $\mathcal{H}_n$ be the set of all such H. We define an inefficient class of attackers $\mathsf{Sam}_{n,H}$ with respect to the ensemble $H \xleftarrow{\$} \mathcal{H}_n$ as follows:

$\mathsf{Sam}_{n,H}$ **:** On input $s \in \{0,1\}^{d(n)}$, choose a random $y \xleftarrow{\$} H$ and a random pre-image $x \xleftarrow{\$} \{x : \mathsf{Cond}_s(x) = y\}$. Output x.

Firstly, fix any $H \in \mathcal{H}_n$ and define $X_s \stackrel{\text{def}}{=} \{x : \mathsf{Cond}_s(x) \in H\}$. Then, by regularity, $|X_s| = 2^{k(n)-m(n)+t(n)}$. Moreover, the output of $\mathsf{Sam}_{n,H}(1^n,s)$ is just a random sample from X_s and therefore has entropy $\mathbf{H}_\infty(\mathsf{Sam}(1^n,s)) = k(n) - m(n) + t(n)$. On the other hand, for any s, the output of $\mathsf{Cond}_s(\mathsf{Sam}(1^n,s)) \in H$ is contained in a set of size $2^{t(n)}$ and therefore $\mathbf{H}_\infty(\mathsf{Cond}_S(\mathsf{Sam}(1^n,S)) \mid S) = t(n)$. Together, this shows that $\mathsf{Sam}_{n,H}$ is a successful attacker against the $t(n)$-condenser security of Cond.

The simulator $\mathsf{Sim}(1^n)$ simply responds to any query s with a uniformly random value $x \xleftarrow{\$} \{0,1\}^{k(n)}$. We need to show that the simulation is good – namely, that for any (possibly inefficient) probabilistic oracle-access machine $\mathcal{M}^{(\cdot)}(1^n)$ making at most $q(n)$ queries in total to its oracle(s), we have:

$$\left| \Pr_{H\xleftarrow{\$}\mathcal{H}_n,\mathrm{coins}(\mathcal{M})}[\mathcal{M}^{\mathsf{Sam}_{n,H}}(1^n)=1] - \Pr_{\mathrm{coins}(\mathcal{M},\mathsf{Sim})}[\mathcal{M}^{\mathsf{Sim}(1^n)}(1^n)=1] \right| \leq 2q^2(n)2^{-t(n)} \tag{8}$$

Let us define a hybrid oracle Hyb which works just like $\{\mathsf{Sam}_{n,H} : H \xleftarrow{\$} \mathcal{H}\}$ but it remembers the set Q of values y chosen in prior "sample" queries and makes sure that each new query is answered with a fresh $y \xleftarrow{\$} H \setminus Q$. Let E be the event that, during the execution of $\mathcal{M}^{\mathsf{Sam}_H}$ with $H \xleftarrow{\$} \mathcal{H}_n$, some two distinct queries were answered with the same choice of $y \xleftarrow{\$} H$. The, conditioned on E not occurring, the oracles $\mathsf{Sam}_{n,H}$ and Hyb are statistically the same, and therefore the distance between them is bounded by $\Pr[E] \leq q^2(n)2^{-t(n)}$. Let E' be the event that, during the execution of $\mathcal{M}^{\mathsf{Sim}(1^n)}(1^n)$, two queries to Sim map to the same y. Then conditioned on E' not occurring, the oracles Hyb and Sim are statistically the same since on each query s both oracles respond with a random preimage $\mathsf{Cond}_s^{-1}(y)$ for a uniformly random fresh y. Therefore the statistical distance between them is at most $\Pr[E'] \leq q^2(n)2^{-m(n)} \leq q^2(n)2^{-t(n)}$. This proves equation (8) and concludes the proof.

□

As an immediate corollary of the above theorem and Theorem 4.2 we get the following.

Corollary 8.3. *For any $t(n) = \omega(\log(n))$, if there is a black-box reduction showing the t-condenser security of any regular candidate function Cond based on the security of some cryptographic game $\mathcal{G}$, then $\mathcal{G}$ is not secure. Moreover, for any $t(n) = n^\delta$ with constant $\delta > 0$, if there is a reduction based on the δ'-exponential security of $\mathcal{G}$ with $\delta' < \delta$ then $\mathcal{G}$ is not δ'-exponentially secure.*

9 Conclusions and Open Problems

In this work, we consider several cryptographic security notions involving weak, leaky and correlated sources of randomness. These notions cannot be naturally expressed in terms of cryptographic games between an attacker and an adversary, and we show a broad black-box separation, that their security does not follow from that of any standard cryptographic game. One open problem is to apply our approach to other primitives whose security is not expressed as a cryptographic game. Another open problem is to come up with reductions that do not treat the attacker as a black box. One such approach by Barak et al. [BHHI10] has been used in the context of KDM secure encryption, cleverly circumventing the black-box separation of Haitner and Holenstein [HH09] by treating a component of the adversary in a non-black-box way. It does not seem that this technique could apply to the primitives discussed in this work, but it remains an interesting open problem to look for some such approach.

10 Acknowledgements

We thank Yael Tauman Kalai for suggesting the problem of *pseudo-entropy generators*, which lead us to the results of this work. We also thank Yael, Iftach Haitner and Yevgeniy Dodis for many enlightening discussions on the topics and results of this work.

References

[ADN+10] Joël Alwen, Yevgeniy Dodis, Moni Naor, Gil Segev, Shabsi Walfish, and Daniel Wichs. Public-key encryption in the bounded-retrieval model. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 113–134. Springer, 2010.

[ADW09a] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In Halevi [Hal09], pages 36–54.

[ADW09b] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Survey: Leakage resilience and the bounded retrieval model. In *ICITS*, pages 1–18, 2009.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *Sixth Theory of Cryptography Conference — TCC 2007*, volume 5444 of *Lecture Notes in Computer Science*. Springer-Verlag, 2009.

[BBO07] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In Alfred Menezes, editor, *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 535–552. Springer, 2007.

[BFO08] Alexandra Boldyreva, Serge Fehr, and Adam O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In Wagner [Wag08], pages 335–359.

[BFOR08] Mihir Bellare, Marc Fischlin, Adam O'Neill, and Thomas Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In Wagner [Wag08], pages 360–378.

[BG10] Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2010.

[BHHI10] Boaz Barak, Iftach Haitner, Dennis Hofheinz, and Yuval Ishai. Bounded key-dependent message security. In *EUROCRYPT*, pages 423–444, 2010.

[BHK11] Mark Braverman, Avinatan Hassidim, and Yael Tauman Kalai. Leaky pseudo-entropy functions. In Bernard Chazelle, editor, *ICS*, pages 353–366. Tsinghua University Press, 2011.

[BKKV10] Zvika Brakerski, Jonathan Katz, Yael Kalai, and Vinod Vaikuntanathan. Overcomeing the hole in the bucket: Public-key cryptography against resilient to continual memory leakage. In *FOCS* [IEE10], pages 501–510.

[BS11] Zvika Brakerski and Gil Segev. Better security for deterministic public-key encryption: The auxiliary-input setting. In Phillip Rogaway, editor, *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 543–560. Springer, 2011.

[BSW03] Boaz Barak, Ronen Shaltiel, and Avi Wigderson. Computational analogues of entropy. In Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai, editors, *RANDOM-APPROX*, volume 2764 of *Lecture Notes in Computer Science*, pages 200–215. Springer, 2003.

[BSW11] Elette Boyle, Gil Segev, and Daniel Wichs. Fully leakage-resilient signatures. In Paterson [Pat11], pages 89–108.

[Cra12] Ronald Cramer, editor. *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, volume 7194 of *Lecture Notes in Computer Science*. Springer, 2012.

[DHLW10a] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Cryptography against continuous memory attacks. In *FOCS* [IEE10], pages 511–520.

[DHLW10b] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Efficient public-key cryptography in the presence of key leakage. In Masayuki Abe, editor, *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 613–631. Springer, 2010.

[DHT12] Yevgeniy Dodis, Iftach Haitner, and Aris Tentes. On the instantiability of hash-and-sign rsa signatures. In Cramer [Cra12], pages 112–132.

[DOPS04] Yevgeniy Dodis, Shien Jin Ong, Manoj Prabhakaran, and Amit Sahai. On the (im)possibility of cryptography with imperfect randomness. In *45th Symposium on Foundations of Computer Science*, pages 196–205, Rome, Italy, October 17–19 2004. IEEE.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

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

[DRV12] Yevgeniy Dodis, Thomas Ristenpart, and Salil P. Vadhan. Randomness condensers for efficiently samplable, seed-dependent sources. In Cramer [Cra12], pages 618–635.

[FOR12] Benjamin Fuller, Adam O'Neill, and Leonid Reyzin. A unified approach to deterministic encryption: New constructions and a connection to computational entropy. In Cramer [Cra12], pages 582–599.

[FV11] 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*. ACM, 2011.

[GKM+00] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *41st Annual Symposium on Foundations of Computer Science*, pages 325–335, Redondo Beach, California, November 2000. IEEE.

[GKPV10] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In Andrew Chi-Chih Yao, editor, *ICS*, pages 230–240. Tsinghua University Press, 2010.

[GMR01] Yael Gertner, Tal Malkin, and Omer Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *42nd Annual Symposium on Foundations of Computer Science*, Las Vegas, Nevada, October 2001. IEEE.

[GOR11] Vipul Goyal, Adam O'Neill, and Vanishree Rao. Correlated-input secure hash functions. In Ishai [Ish11], pages 182–200.

[GR12] Shafi Goldwasser and Guy N. Rothblum. How to compute in the presence of leakage. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:10, 2012.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Fortnow and Vadhan [FV11], pages 99–108.

[Hal09] Shai Halevi, editor. *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *LNCS*. Springer-Verlag, 2009.

[HH09] Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. In Reingold [Rei09], pages 202–219.

[HL11] Shai Halevi and Huijia Lin. After-the-fact leakage in public-key encryption. In Ishai [Ish11], pages 107–124.

[HLO10] Brett Hemenway, Steve Lu, and Rafail Ostrovsky. Correlated product security from any one-way function and the new notion of decisional correlated product security. Cryptology ePrint Archive, Report 2010/100, 2010. `http://eprint.iacr.org/`.

[IEE10] IEEE. *51th Symposium on Foundations of Computer Science*, Las Vegas, NV, USA, October 23–26 2010.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In D. S. Johnson, editor, *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 44–61, Seattle, Washington, 15–17 May 1989.

[Ish11] Yuval Ishai, editor. *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, volume 6597 of *Lecture Notes in Computer Science*. Springer, 2011.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology—CRYPTO 2003*, volume 2729 of *LNCS*. Springer-Verlag, 2003.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In Mitsuru Matsui, editor, *Advances in Cryptology—ASIACRYPT 2009*, LNCS. Springer-Verlag, 2009. To Appear.

[Lys02] Anna Lysyanskaya. Unique signatures and verifiable random functions from the dh-ddh separation. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 597–612. Springer, 2002.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In Naor [Nao04], pages 278–296.

[Nao04] Moni Naor, editor. *First Theory of Cryptography Conference — TCC 2004*, volume 2951 of *LNCS*. Springer-Verlag, February 19–21 2004.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Halevi [Hal09], pages 18–35.

[Pas11] Rafael Pass. Limits of provable security from standard assumptions. In Fortnow and Vadhan [FV11], pages 109–118.

[Pat11] Kenneth G. Paterson, editor. *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*. Springer, 2011.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Cynthia Dwork, editor, *STOC*, pages 187–196. ACM, 2008.

[Rei09] Omer Reingold, editor. *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, volume 5444 of *Lecture Notes in Computer Science*. Springer, 2009.

[RR99] Ran Raz and Omer Reingold. On recycling the randomness of states in space bounded computation. In Jeffrey Scott Vitter, Lawrence L. Larmore, and Frank Thomson Leighton, editors, *STOC*, pages 159–168. ACM, 1999.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In Reingold [Rei09], pages 419–436.

[RSS11] Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In Paterson [Pat11], pages 487–506.

[RTTV08] Omer Reingold, Luca Trevisan, Madhur Tulsiani, and Salil P. Vadhan. Dense subsets of pseudorandom sets. In *FOCS*, pages 76–85. IEEE Computer Society, 2008.

[RTV04] Omer Reingold, Luca Trevisan, and Salil Vadhan. Notions of reducibility between cryptographic primitives. In Naor [Nao04], pages 1–20.

[Sim98] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions. In Kaisa Nyberg, editor, *Advances in Cryptology—EUROCRYPT 98*, volume 1403 of *LNCS*. Springer-Verlag, May 31–June 4 1998.

[Wag08] David Wagner, editor. *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*. Springer, 2008.

A Weak-Distribution Resilient One-Way Functions

Let $\mathcal{F} \stackrel{\text{def}}{=} \{ \ f_{pk} \ : \ \{0,1\}^{k(n)} \to \{0,1\}^{m(n)} \ \}_{n\in\mathbb{N}}$ be a family of efficiently computable functions keyed by a public $pk \leftarrow \mathsf{Gen}(1^n)$.

Definition A.1 (Weakness Resilient OWF)**.** *Let $(\mathcal{F}, \mathsf{Gen})$ be as above. A probabilistic sampling algorithm Sam is called $\ell(\cdot)$-*weak *if the distribution $\mathsf{Sam}(1^n)$ induces a distribution over $\{0,1\}^{k(n)}$ such that $\mathbf{H}_\infty(\mathsf{Sam}(1^n)) \geq k(n) - \ell(n)$. We say that $(\mathcal{F}, \mathsf{Gen})$ is an $\ell(\cdot)$-*weakness resilient one-way function *if for any PPT algorithms $\mathcal{A} = (\mathsf{Sam}, \mathsf{Break})$ such that Sam is $\ell(n)$-weak, we have*

$$\Pr\left[f_{pk}(x') = y \;\middle|\; \begin{array}{c} pk \leftarrow \mathsf{Gen}(1^n), x \stackrel{\$}{\leftarrow} \mathsf{Sam}(1^n), y = f_{pk}(x) \\ x' \leftarrow \mathsf{Break}(1^n, pk, y) \end{array} \right] = \mathsf{negl}(n)$$

Definition A.2. *We say that the function family $\mathcal{F}$ is* recognizably injective *if there is an efficient algorithm $\mathsf{Ver}(pk) = 1$ iff f_{pk} is injective.*

Theorem A.3. *Let $(\mathcal{F}, \mathsf{Gen})$ be any recognizably injective function family. Then there is a $2^{-\ell(n)}$ simulatable attack against the ℓ-weakness-resilient one-wayness of the family.*

Proof. Let $H \subseteq \{0,1\}^{k(n)}$ be a set of size $|H| = 2^{k(n)-\ell(n)}$ and let $\mathcal{H}_n$ consist of all such sets. We define two (inefficient) oracles $\mathsf{Sam}_{n,H}, \mathsf{Break}_{n,H}$ with respect to $H \in \mathcal{H}$ as follows:

Sam_H **:** On input 1^n, output a random $x \stackrel{\$}{\leftarrow} H$.

Break_H**:** On input (pk, y), verify that $\mathsf{Ver}(pk) = 1$ and output $\perp$ if not. Else do an exhaustive search to find $x \in H$ such that $f_{pk}(x) = y$. If no such x exists output $\perp$ and else output x.

Firstly, it is easy to see that for any $H \in \mathcal{H}_n$, the above oracles break ℓ-WR one-wayness of $(\mathcal{F}, \mathsf{Gen})$ with advantage 1. This is because the entropy of the distribution $\mathsf{Sam}_{n,H}(1^n)$ is always exactly $k(n) - \ell(n)$ and for any $x \stackrel{\$}{\leftarrow} \mathsf{Sam}_{n,H}(1^n, s)$, we have $\mathsf{Break}_{n,H}(1^n, s, f_s(x)) = x$.

Now we show that when H is chosen uniformly at random from $\mathcal{H}_n$, then getting oracle access to $(\mathsf{Leak}^H, \mathsf{Break}_{n,H})$ is *not very useful*. In particular, access to such oracles can be efficiently simulated up to a small statistical distance. The simulator $\mathsf{Sim}(1^n)$ must respond to "sample" queries and to "break" queries and it does so as follows:

1. Initialization: Set $Q := \emptyset$.
2. Sample: On a *sample* query, with probability $p = \frac{|Q|}{2^{k(n)-\ell(n)}}$ output a random $x \xleftarrow{\$} Q$. Otherwise choose $x \xleftarrow{\$} \{0,1\}^{k(n)}$, add the tuple x to Q, and output x.
3. Break: On a *break* query (pk, y), first verify (efficiently) that $\mathsf{Ver}(pk) = 1$ and output $\perp$ if the check fails. Else check if there is some $x \in Q$ such that $f_{pk}(x) = y$ and, if so, output it and else output $\perp$.

We wish to show that the simulation is good and for any oracle-access (possibly inefficient) $\mathcal{M}$ making at most $q(n)$ queries to its oracle, we have:

$$\left| \Pr_{H \xleftarrow{\$} \mathcal{H}_n, \mathrm{coins}(\mathcal{M})} [\mathcal{M}^{(\mathsf{Sam}_{n,H}, \mathsf{Break}_{n,H})}(1^n) = 1] - \Pr_{\mathrm{coins}(\mathcal{M}, \mathsf{Sim})} [\mathcal{M}^{\mathsf{Sim}(1^n)}(1^n) = 1] \right| \leq q(n) 2^{-\ell(n)}.$$

First, let us define the event E to be the event that during the execution $\mathcal{M}^{(\mathsf{Sam}_{n,H}, \mathsf{Break}_{n,H})}(1^n)$, the machine $\mathcal{M}$ makes a "break" query (pk, y) such that $y \neq f_{pk}(x)$ for any x returned by a prior "sample" query, and the response from the oracle is *not* $\perp$. The conditioned on E not occurring, the experiments are the same, since we can think of $\mathsf{Sim}(1^n)$ as just choosing the values of H "on-the-go" as it answers its sample queries. Therefore the statistical distance between the real game and the simulation is bounded by $\Pr[E]$ which is the probability that the query (pk, y) satisfies that its unique preimage $x = f_{pk}^{-1}(y) \in H$, which is the probability of the attacker being able to guess a new value in H after $\leq q$ queries, which is bounded by $\Pr[E] \leq q(n) 2^{-\ell(n)}$ as we wanted to show.

□