Key Derivation Without Entropy Waste

Yevgeniy Dodis * Krzysztof Pietrzak † Daniel Wichs ‡

April 9, 2013

Abstract

We revisit the classical question of converting an imperfect source X of min-entropy k into a usable m-bit cryptographic key for some underlying application P. If P has security δ (against some class of attackers) with a uniformly random m-bit key, we seek to design a key derivation function (KDF) h that allows us to use $R = h(X)$ as the key for P and results in comparable security $\delta' \approx \delta$. Seeded randomness extractors provide a generic way to solve this problem for all applications, provided that $k \geq m + 2\log(1/\delta)$, and this lower bound on k (called "RT-bound") is known to be tight in general. Unfortunately, in many situations the loss of $2\log(1/\delta)$ bits of entropy is unacceptable, motivating the study KDFs with less entropy waste for important special classes of sources X or applications P.

In this work we obtain the following new positive and negative results in this regard:

- Efficient samplability of the source X does not help beat the RT-bound for general applications. This resolves the SRT (samplable RT) conjecture of Dachman-Soled et al. [DGKM12] in the affirmative, and also shows that the existence of computationally-secure extractors beating the RT-bound implies the existence of one-way functions.
- We continue the line of work initiated by Barak et al. [BDK$^+$11] by constructing information-theoretic KDFs which beat the RT-bound bound for large but restricted classes of applications. Specifically, we design efficient KDFs that work for *all unpredictability applications* P (signatures, MACs, one-way functions, etc.) and can either: (1) extract *all* of the entropy $k = m$ with a very modest security loss $\delta' = O(\delta \cdot \log(1/\delta))$, or alternatively, (2) achieve optimal security $\delta' = O(\delta)$ with a very modest entropy loss $k \geq m + \operatorname{loglog}(1/\delta)$. In comparison, the best prior results from [BDK$^+$11] for this class of applications achieved $\delta' = O(\sqrt{\delta})$ when $k = m$ and needed $k \geq m + \log(1/\delta)$ for $\delta' = O(\delta)$.
- The weaker bounds of [BDK$^+$11] hold for a larger class of so-called "square-friendly" applications (which includes all unpredictability, but also some important indistinguishability, applications). Unfortunately, we show that these bounds are tight for this strictly larger class of applications.
- We abstract out a clean, information-theoretic notion of (k, δ, δ')-*unpredictability extractors*, which guarantee "induced" security δ' for any δ-secure unpredictability application P, and characterize the parameters achievable for such unpredictability extractors. Of independent interest, we also relate this notion to the previously-known notion of (min-entropy) *condensers*, and improve the state-of-the-art parameters for such condensers.

*New York Unviersity. E-mail: `dodis@cs.nyu.edu`

†Institute of Science and Technology Austria. E-mail:`pietrzak@ist.ac.at`

‡Northeastern University. E-mail: `wichs@ccs.neu.edu`

1 Introduction

Key Derivation is a fundamental cryptographic task arising in a wide variety of situations where a given application P was designed to work with a uniform m-bit key R, but in reality one only has a "weak" n-bit random source X. Examples of such sources include biometric data [DORS08, BDK$^+$05], physical sources [BST03, BH05], secrets with partial leakage, and group elements from Diffie-Hellman key exchange [GKR04, Kra10], to name a few. We'd like to have a *Key Derivation Function* (KDF) $h : \{0,1\}^n \to \{0,1\}^m$ with the property that the derived key $h(X)$ can be safely used by P, even though the original security of P was only analyzed under the assumption that its key R is uniformly random.

Of course, good key derivation is generally impossible unless X has some amount of entropy k to begin with, where the "right" notion of entropy in this setting is *min-entropy* $\mathbf{H}_\infty(X)$: for any $x \in \{0,1\}^n$ we must have $\Pr[X = x] \leq 2^{-k}$. We call such a distribution X an (n,k)*-source*, and generally wish to design a KDF h which "works" for all such (n,k)-sources X. A bit more formally, assuming P was δ-secure (against some class of attackers) with the uniform key $R \equiv U_m$, we would like to conclude that P is still δ'-secure (against nearly the same class of attackers) when using $R = h(X)$ instead. The two most important parameters are: (1) ensuring that the new security δ' is "as close as possible" to the original security δ, and (2) allowing the source entropy k to be "as close as possible" to the application's key length m. Minimizing this threshold k is very important in many practical situations. For example, in the setting of biometrics and physical randomness many natural sources are believed to have very limited entropy, while in the setting of Diffie-Hellman key exchange reducing the size of the Diffie-Hellman group (which is roughly 2^k) results in substantial efficiency improvements. Not surprisingly, the design of good KDFs has received a lot of attention from both theoreticians and practitioners. Unfortunately, as we detail below, there is a huge gap between the theory and practice of key derivation, not only in terms of the actual functions used, but also in the min-entropy threshold k which is considered "safe".

Key Derivation in Practice. In practice, one would typically use so called "cryptographic hash function" h, such as SHA or MD5, for key derivation. As discussed in detail by [DGH$^+$04, Kra10, DRV12], there are several important reasons for this choice. From the perspective of this work, we will focus on the arguably the most important such reason — the common belief that cryptographic hash function achieve excellent security $\delta' \approx \delta$ already when $k \approx m$. This can be easily *justified in the random oracle model*; assuming the KDF h is a random oracle which can be evaluated on at most q points (where q is the upper bound of the attacker's running time), one can upper bound $\delta' \leq \delta + q/2^k$, where $q/2^k$ is the probability the attacker evaluates $h(X)$. In turn, in time q the attacker can also test about q out of 2^m possible m-bit keys, and hence achieve advantage $q/2^m$. This means that the ideal security δ of P cannot be lower than $q/2^m$, implying $q \leq \delta \cdot 2^m$. Plugging this bound on q in the bound of $\delta' \leq \delta + q/2^k$ above, we get that using a random oracle (RO) as a KDF achieves "real security"

$$\delta' \leq \delta_{\mathsf{RO}} \stackrel{\text{def}}{=} \delta + \delta \cdot 2^{m-k} \tag{1}$$

In particular, $\delta' \leq 2\delta$ even when $k = m$. For example, to derive a 128-bit key for a CBC-MAC with security $\delta \approx \delta' \approx 2^{-64}$, one needs $k \approx 128$ bits of min-entropy.

Of course, as any analysis in the random oracle model [CGH98], the bound above is ultimately a heuristic. Moreover, as was pointed out by [DGH$^+$04, Kra10], existing hash functions, such as SHA and MD5, as far from ideal, since they use a highly structured Merkle-Damgard mode of operation when processing long inputs. In particular, the provable "extraction bounds" one gets when taking this structure into account are nowhere close to the amazing bound in (1), even under the generous assumption that the "compression function" f of h is "ideal". Still, the lack of realistic attacks on any of the most common uses of cryptographic hash functions as KDFs when $k \approx m$, leaves us with the following central question of this work:

Our Main Question: *Can one find reasonable application scenarios where one can design a* **provably-secure** *KDF achieving "real security" $\delta' \approx \delta$ when $k \approx m$ (matching the heuristic bound in (1))?*

More generally, for a given (class of) applications P,

(A) *What is the best (provably) achievable security* δ' *when* $k = m$*?*

(B) *What is the smallest (provable) entropy threshold* k *to achieve security* $\delta' = O(\delta)$*?*

In this work we will provide several positive and negative answers to our main question, including a general way to (nearly) *match the bound from Equation (1) information-theoretically for all unpredictability applications.* But first we turn to what is known in the theory of key derivation.

Randomness Extractors. In theory, the cleanest way to design a general KDF is by using so called (strong) *randomness extractors* [NZ96]. Such a (k, ε)-extractor Ext has the property that the output distribution $\mathsf{Ext}(X)$ is ε-statistically close to the uniform distribution U_m, which means that using $\mathsf{Ext}(X)$ as a key will degrade the original security δ of *any* application P by at most ε: $\delta' \leq \delta + \varepsilon$. However, the sound use of randomness extractors comes with two important caveats. The first caveat comes from the fact that no deterministic extractor Ext can work for all (n, k)-sources [CG89] when $k < n$, which means that extractors must be probabilistic, or "seeded". This by itself is not a big limitation, since the extracted randomness $\mathsf{Ext}(X; S)$ is ε-close to U_m even *conditioned on the seed* S, which means that the seed S can be reused and globally shared across many applications.[1] From our perspective, though, a more important limitation/caveat of randomness extractors comes from a non-trivial tradeoff between the min-entropy k and the security ε one can achieve to derive an m-bit key $\mathsf{Ext}(X; S)$. The best randomness extractors, such as the one given by the famous Leftover Hash Lemma (LHL) [HILL99], can only achieve security $\varepsilon = \sqrt{2^{m-k}}$. This gives the following very general bound on δ' for *all* applications P:

$$\delta' \leq \delta_{\mathsf{ALL}} \stackrel{\text{def}}{=} \delta + \sqrt{2^{m-k}} \tag{2}$$

As we can see, this provable (and very general) bound is much worse than the heuristic bound in Equation (1). In particular, we get no meaningful security when $k = m$ (giving no answer to Question (A)), and must assume $k \geq m + 2\log(1/\delta)$ to ensure that $\delta' = O(\delta)$ for Question (B). For example, to derive a 128-bit key for a CBC-MAC with security $\delta \approx \delta' \approx 2^{-64}$, one needs $k \approx 256$ bits of min-entropy.

Of course, part of the reason why the provable bounds are so much worse is their generality: extractors work for *all* (n, k)-sources X and *all* applications P. Unfortunately, Radhakrishnan and Ta-shma [RTS00] showed that in this level of generality nothing better is possible: any (k, ε)-extractor must have $k \geq m + 2\log(1/\varepsilon)$ (we will refer to this as the "RT-bound"). This implies that for any candidate m-bit extractor Ext there exists some (possibly *indistinguishability*) application P, some (possibly *inefficiently samplable*) source X of min-entropy k and some (possibly *exponential time*) attacker A, such that $A(S)$ can break P with key $R = \mathsf{Ext}(X; S)$ with advantage $\sqrt{2^{m-k}}$.

Thus, there is hope that better results are possible if one restricts the type of applications P (e.g., unpredictability applications), sources X (e.g., efficiently samplable) or attackers A (e.g., polynomial-time) considered. We discuss such options below, stating what was known together with our new results.

Efficiently Samplable Sources. This natural restriction is known to be useful for relaxing the assumption that the source distribution X is independent of the seed S [TV00, DRV12], which was the first caveat in using randomness extractors. Unfortunately, it was not clear if efficient samplability of X helps with reducing the entropy loss $L = k - m$ below $2\log(1/\varepsilon)$. In fact, Dachman-Soled et al. [DGKM12] conjectured that this is indeed not the case when Ext is also efficient, naming this conjecture the "SRT assumption" (where SRT stands for "samplable RT").

SRT Assumption [DGKM12]: *For any efficient extractor* Ext *with* m*-bit output there exists an efficiently samplable (polynomial in* n*) distribution* X *of min-entropy* $k = m + 2\log(1/\varepsilon) - O(1)$ *and a (generally inefficient) distinguisher* D *which has at least an* ε*-advantage in distinguishing* $(S, R = \mathsf{Ext}(X; S))$ *from* $(S, R = U_m)$*.*

[1] However, it does come with an important assumption that the source distribution X must be independent of the seed S. Although this assumption could be problematic in some situations, such as leakage-resilient cryptography (and has led to some interesting research [TV00, CDH+00, KZ03, DRV12]), in many situations, such as the Diffie-Hellman key exchange or biometrics, the independence of the source and the seed could be naturally enforced/assumed.

As our first result, we show that the SRT assumption is indeed (unfortunately) true, even *without* restricting the extractor Ext to be efficient.

Theorem 1.1. *(Informal) The SRT assumption is true for any (possibly inefficient) extractor* Ext*. Thus, efficiently samplability does not help to reduce the entropy loss of extractors below* $2\log(1/\varepsilon)$*.*

SQUARE-FRIENDLY APPLICATIONS. The next natural restriction is to limit the class of applications P in question. The idea is that for such applications one can argue that the derived key $R = h_s(X)$ is still "good enough" for P despite *not* being statistically close to U_m (given s). This approach was recently pioneered by Barak et al [BDK+11], and then further extended and generalized by Dodis et al. [DRV12, DY13]. In these works the authors defined a special class of cryptographic applications, called *square-friendly*, where the pessimistic RT-bound can be provably improved. Intuitively, while any traditional application P demands that the expectation (over the uniform distribution $r \leftarrow U_m$) of the attacker's advantage $f(r)$ on key r is at most δ, square-friendly applications additionally require that the expected value of $f(r)^2$ is also bounded by δ. The works of [BDK+11, DY13] then showed that the class of square-friendly applications includes *all unpredictability applications* (signatures, MACs, one-way functions, etc.), and *some, but not all, indistinguishability applications* (including chosen plaintext attack secure encryption, weak pseudorandom functions and others). Additionally, for all such square-friendly applications P it was shown that universal (and thus also the stronger pairwise independent) hash functions $\{h_s\}$ yield the following improved bound on the security δ' of the derived key $R = h_s(X)$:

$$\delta' \leq \delta_{\mathsf{SQF}} \stackrel{\text{def}}{=} \delta + \sqrt{\delta \cdot 2^{m-k}} \tag{3}$$

This provable and still relatively general bound lies somewhere in between the idealized bound (1) and the fully generic bound (2): in particular, for the first time we get a meaningful security $\delta' \approx \sqrt{\delta}$ when $k = m$ (giving partial answer to Question (A)), or, alternatively, we get full security $\delta' = O(\delta)$ provided $k \geq m + \log(1/\delta)$ (giving a partial answer to Question (B)). For example, to derive a 128-bit key for a CBC-MAC having ideal security $\delta = 2^{-64}$, we can either settle for much lower security $\delta' \approx 2^{-32}$ with $k = 128$, or get full security $\delta' \approx 2^{-64}$ with $k = 192$. However, both bounds are still far from the expected bound $\delta' \approx 2^{-64}$ with $k = 128$, raising the question if further improvements are possible.

As a simple (negative) result, we show that the bound in Equation (3) cannot be improved in general for *all* square-friendly applications. Interestingly, the proof of this result uses the proof of Theorem 1.1 to produce the desired source X for the counter-example.

Theorem 1.2. *(Informal) There exists a δ-square friendly application P with an m-bit key such that for any family $\mathcal{H} = \{h_s\}$ of m-bit key derivation functions there exists (even efficiently samplable) (n,k)-source X and a (generally inefficient) distinguisher D such that $D(S)$ has at least $\delta' = \Omega(\sqrt{\delta \cdot 2^{m-k}})$ advantage in breaking P with the derived key $R = h_S(X)$ (for random seed S).*

Hence, to improve the parameters in Equation (3) and still have information-theoretic security, we must place more restrictions on the class of applications P we consider.

UNPREDICTABILITY APPLICATIONS. This brings us to our main (positive) result: we get improved information-theoretic key derivation for all unpredictability applications.

Theorem 1.3. *(Main Result; Informal) Assume P is any unpredictability[2] application which is δ-secure with a uniform m-bit key against some class of attackers $\mathcal{C}$. Then, there is an efficient family of hash functions $\mathcal{H} = \{h_s : \{0,1\}^n \rightarrow \{0,1\}^m\}$, such that for any (n,k)-source X, the application P with the derived key $R = h_S(X)$ (for random public seed S) is δ'-secure against class $\mathcal{C}$, where:*

$$\delta' \leq \delta_{\mathsf{UNP}} \stackrel{\text{def}}{=} O\left(1 + \log(1/\delta) \cdot 2^{m-k}\right)\delta. \tag{4}$$

In particular, we get the following nearly optimal answers to Questions (A) and (B):

[2]Recall, an unpredictability application is any cryptographic game where the attacker's advantage is simply the probability of him "winning the game". (In contrast, for indistinguishability applications one subtracts 1/2 from this probability.)

- $\delta' \le (1 + \log(1/\delta))\delta$ *when* $k = m$ *(answering Question (A)).*
- $\delta' \le 3\delta$ *provided* $k \ge m + \operatorname{loglog}(1/\delta) + 4$ *(answering Question (B)).*

In fact, our basic KDF hash family $\mathcal{H}$ is simply a t-wise independent hash function where $t = O(\log(1/\delta))$. Hence, by using higher than pairwise independence (which was enough for weaker security given by Equations (2) and (3)), we get a largely improved entropy loss: $\operatorname{loglog}(1/\delta)$ instead of $\log(1/\delta)$.

As we can see, the *provable* bound on δ' above nearly matches the idealized bound $\delta' \le \delta + \delta \cdot 2^{m-k}$ from Equation (1), except for the mild security loss $O(\log(1/\delta))$, or alternatively, mild entropy loss $\operatorname{loglog}(1/\delta)$. For example, to derive a 128-bit key for a CBC-MAC having ideal security $\delta = 2^{-64}$ (so that $\operatorname{loglog}(1/\delta) = 6$), we can either have excellent security $\delta' \le 2^{-57.9}$ with $k = 128$, or get full security $\delta' \le 2^{-62.4}$ with $k = 138$. Thus, for the first time we obtained an efficient, theoretically-sound key derivation scheme which nearly matches the parameters achieved by heuristic KDFs. In particular, for the first time we can offer a provably-secure *alternative* to the existing practice of using cryptographic hash functions as KDFs and achieve nearly optimal parameters.

Unpredictability Extractors and Condensers. To better understand the proof of Theorem 1.3, it is helpful to abstract the notion of an *unpredictability extractor* UExt which we define in this work. Recall, standard (k, ε)-extractors ε-fool any distinguisher $D(R, S)$. In contrast, when dealing with δ-secure unpredictability applications, we only care about "fooling" so called δ-distinguishers D: these are distinguishers s.t. $\Pr[D(U_m, S) = 1] \le \delta$, which directly corresponds to the emulation of P's security experiment between the "actual attacker" A and the challenger C. Thus, we define (k, δ, δ')-*unpredictability extractors* as having the property that $\Pr[D(\mathsf{UExt}(X; S), S) = 1] \le \delta'$ for any δ-distinguisher D.[3] With this cleaner notion in mind, our main Theorem 1.3 can be equivalently restated as follows:

Theorem 1.4. *(Main Result; Restated) A family* $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^m\}$ *which is* $O(\log(1/\delta))$*-wise independent defines a* $(k, \delta, O(1 + \log(1/\delta) \cdot 2^{m-k})\delta)$*-unpredictability extractor* $\mathsf{UExt}(x; s) = h_s(x)$.

In turn, we observe that unpredictability extractors are closely connected to the related notion of a *randomness condenser* [RR99, RSW06]: such a (k, ℓ, ε)-condenser $\mathsf{Cond} : \{0,1\}^n \to \{0,1\}^m$ has the property that the output distribution $\mathsf{Cond}(X; S)$ is ε-close (even given the seed S) to some distribution Y s.t. the conditional min-entropy $\mathbf{H}_\infty(Y|S) \ge m - \ell$ whenever $\mathbf{H}_\infty(X) \ge k$. In particular, instead of requiring the output to be close to uniform, we require it to be close to having almost full entropy, with some small "gap" ℓ. While $\ell = 0$ gives back the definition of (k, ε)-extractors, permitting a small non-zero "entropy gap" ℓ has recently found important applications for key derivation [BDK+11, DRV12, DY13]. In particular, it is easy to see that a (k, ℓ, ε)-condenser is also a $(k, \delta, \varepsilon + \delta \cdot 2^\ell)$-unpredictability extractor. Thus, to show Theorem 1.4 it suffices to show that $O(\log(1/\delta))$-wise independent hashing gives a (k, ℓ, δ)-condenser, where $\ell \approx \operatorname{loglog}(1/\delta)$.

Theorem 1.5. *(Informal) A family* $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^m\}$ *of* $O(\log(1/\delta))$*-wise independent hash functions defines a* (k, ℓ, δ)*-condenser* $\mathsf{Cond}(x; s) = h_s(x)$ *for either of the following settings:*

- No Entropy Loss*: min-entropy* $k = m$ *and entropy gap* $\ell = \operatorname{loglog}(1/\delta)$.
- Constant Entropy Gap*: min-entropy* $k = m + \operatorname{loglog}(1/\delta) + O(1)$ *and entropy gap* $\ell = 1$.

It is instructive to compare this result with the RT-bound for (k, δ)-extractors: to have no entropy gap $\ell = 0$ requires us to start with entropy $k \ge m + 2\log(1/\delta)$. However, already 1-bit entropy gap $\ell = 1$ allows us to get away with $k = m + \operatorname{loglog}(1/\delta)$, while further increasing the gap to $\ell = \operatorname{loglog}(1/\delta)$ results in no entropy loss $k = m$.

Balls and Bins, Max-Load and Balanced Hashing. Finally, to prove Theorem 1.5 (and, thus, Theorem 1.4 and Theorem 1.3) we further reduce the problem of condensers to a very simple balls-and-bins problem. Indeed, we can think of our (k, ℓ, δ)-condenser as a way to hash 2^k items (out of a universe

[3] This notion can also be viewed as "one-sided" slice extractors [RTS00]. Unlike this work, though, the authors of [RTS00] did not use slice extractors as an interesting primitive by itself, and did not offer any constructions of such extractors.

of size 2^n) into 2^m bins, so that the *load* (number of items per bin) is not too much larger than the expected 2^{k-m} for "most" of the bins. More concretely, it boils down to analyzing a version of average-load: if we choose a random item (and a random hash function from the family) then the probability that the item lands in a bin with more than $2^\ell(2^{k-m})$ items should be at most ε. We use Chernoff-type bounds for limited independence [Sie89, BR94] to analyze this version of average load when the hash function is $O(\log 1/\delta)$-independent.

Optimizing Seed Length. The description length d of our $O(\log(1/\delta))$-wise independent KDF h_s is $d = O(n\log(1/\delta))$, which is much larger than that needed by universal hashing for standard extractors. We show how to adapt the elegant "gradual increase of independence" technique of Celis et al. [CRSW11] to reduce the seed length to nearly linear: $d = O(n \log k)$ (e.g., for $k = 128$ and $\delta = 2^{-64}$ this reduces the seed length from $128n$ to roughly $7n$). It is an interesting open problem if the seed length can be reduced even further (and we show non-constructively that the answer is positive).

Computational Extractors. So far we considered information-theoretic techniques for designing theoretically-sound KDFs trying to approach the heuristic bound in Equation (1). In contrast, the derivation of Equation (1) critically used the fact that the attacker D can only make a bounded number q of random oracle queries.[4] On the one hand, this explains why the impossibility results in Theorem 1.1, Theorem 1.2 all had to use *inefficient* attackers D (as otherwise, by contradicting Equation (1), they would find unexpected weakness in existing cryptographic hash functions). On the other hand, it opens the possibility that one can actually match the bound in Equation (1), or perhaps overcome some of our impossibility results, by explicitly assuming that the attacker D is *computationally bounded*.

We largely leave the exploration of this exciting direction to future research, here only making the following two initial observations. Both observations are negative results about *computational extractors* [DGH+04, Kra10, DGKM12] Ext, whose output $R = \mathsf{Ext}(X; S)$ looks *pseudorandom* to D (given S) for any *efficiently samplable* (n, k)-source X, which would suffice for our KDF goals if very strong results were possible for such extractors.

Expand-then-Extract Approach. One very natural way to build computational extractors is the folklore *extract-then-extract* approach (recently explored in more detail by [Kra10, DGKM12]). The idea is to define $\mathsf{Ext}(X; S) = \mathsf{Prg}(\mathsf{Ext}'(X; S))$, where $\mathsf{Prg} : \{0,1\}^{m'} \to \{0,1\}^m$ is a computationally $(t, \delta_{\mathsf{PRG}})$-secure pseudorandom generator (PRG), and Ext' is an information-theoretic (k, ε)-extractor with an m'-bit output. It is clear that the resulting computational extractor has has security $\delta_{\mathsf{PRG}} + \varepsilon$, which means that $R = \mathsf{Ext}(X; S)$ can be used in any computationally (t, δ)-secure application P, and result in (t, δ')-security, where $\delta' \le \delta + \varepsilon + \delta_{\mathsf{PRG}}$. In particular, it is tempting to set $\delta_{\mathsf{PRG}} \approx \varepsilon \approx \delta$, which gives $\delta' = O(\delta)$, and ask what is the smallest entropy threshold for k where such setting of parameters is possible. In other words, how good is the extract-then-expand approach for answering our Question (B)?

Unfortunately, we show that the resulting parameters must be poor, at least for the low-entropy settings we care about. Indeed, since the best information-theoretic security δ for the extractor Ext' is $\delta = \sqrt{2^{k-m'}}$ [RTS00], we get that the best value of k we can hope for is $k = m' + 2\log(1/\delta)$, where m' is the smallest possible seed length for a (t, δ)-secure PRG. However, it is well known (e.g., see [DTT10]) than any non-trivial (m', δ)-secure PRG with an m'-bit seed must have seed length $m' > 2\log(1/\delta)$. This gives a lower bound $k > 4\log(1/\delta)$ even for linear-time distinguishers (and the bound actually gets worse when t grows). For example, if $\delta = 2^{-64}$, we get $k > 256$, which is already worse that the naive bound we directly got from an information-theoretic secure extractor when $m = 128$ (see Equation (2)). Indeed, in this case the PRG itself must have a longer seed $m' > 128$ than the derived 128-bit key we are looking for! Thus, although the extract-then-expand approach is indeed useful for medium-to-high rage values of k (e.g., $k \gg 256$), it does not appear to be of any use for the more important low-entropy (e.g., $k < 256$) scenarios.

[4]For example, unlike our bound in Equation (4), one cannot apply the heuristic bound from Equation (1) to derive a key for an *information-theoretically* secure MAC.

Beating RT-bound Implies OWFs. Despite provably failing to solve our Main Question for low-entropy regimes, the extract-then-expand approach at least showed that computational assumptions help in "beating" the RT-bound $k \geq m+2\log(1/\varepsilon)$ for any (k,ε)-secure extractor, as applying the PRG allows one to increase m essentially arbitrarily (while keeping $k = m'+2\log(1/\varepsilon)$). Motivated by this, Dachman-Soled et al. [DGKM12] asked an interesting theoretical question if the existence of one-way functions (and, hence, PRGs [HILL99]) is *essential* for beating the RT-bound for unconditional extractors. They also managed to give an affirmative answer to this question *under the SRT assumption* mentioned earlier. Since we unconditionally prove the SRT assumption (see Theorem 1.1), we immediately get the following Corollary, removing the conditional clause from the result of [DGKM12]:

Theorem 1.6. *(Informal) If* Ext *is an efficient* (k,ε)*-computational extractor with an* m*-bit output, where* $m > k - 2\log(1/\varepsilon) - O(1)$*, then one-way functions (and, hence, PRGs) exist.*

2 Preliminaries

We recap some definitions and results from probability theory. Let X, Y be random variables with supports S_X, S_Y, respectively. We define their *statistical difference* as

$$\Delta(X,Y) = \frac{1}{2}\sum_{u \in S_X \cup S_Y} \left|\Pr[X=u] - \Pr[Y=u]\right|.$$

We write $X \approx_\varepsilon Y$ and say that X and Y are ε-statistically close to denote that $\Delta(X,Y) \leq \varepsilon$.

The *min-entropy* of a random variable X is $\mathbf{H}_\infty(X) \stackrel{\text{def}}{=} -\log(\max_x \Pr[X=x])$, and measures the "best guess" for X. The *conditional min-entropy* is defined by $\mathbf{H}_\infty(X|Y=y) \stackrel{\text{def}}{=} -\log(\max_x \Pr[X=x|Y=y])$. Following Dodis et al. [DORS08], we define the *average* conditional min-entropy:

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

We say that a random variable X is an (n,k)-*source* if the support of X is $\{0,1\}^n$ and the entropy of X is $\mathbf{H}_\infty(X) \geq k$.

Lemma 2.1 (A Tail Inequality [BR94])**.** *Let $q \geq 4$ be an even integer. Suppose $X_1, \ldots, X_n$ are q-wise independent random variables taking values in $[0,1]$. Let $X := X_1 + \cdots + X_n$ and define $\mu := \mathbf{E}[X]$ be the expectation of the sum. Then, for any $A > 0$, $\Pr[|X-\mu| \geq A] \leq 8\left(\frac{q\mu+q^2}{A^2}\right)^{q/2}$. In particular, for any $\alpha > 0$ and $\mu > q$, we have $\Pr[X \geq (1+\alpha)\mu] \leq 8\left(\frac{2q}{\alpha^2\mu}\right)^{q/2}$.*

3 Defining Extractors for Unpredictability Applications

We start by abstracting out the notion of general unpredictability applications (e.g., one-way functions, signatures, message authentication codes, soundness of an argument, etc.) as follows. The security of such all such primitives is abstractly defined via a security game P which requires that, for all attackers $\mathcal{A}$ (in some complexity class), $\Pr[P^{\mathcal{A}}(U)=1] \leq \delta$ where $P^{\mathcal{A}}(U)$ denotes the execution of the game P with the attacker $\mathcal{A}$, where P uses the uniform randomness U.[5] For example, in the case of a message-authentication code (MAC), the value U is used as secret key for the MAC scheme and the game P is the standard "existential unforgeability against chosen-message attack game" for the given MAC. Next, we will assume that δ is some small (e.g., negligible) value, and ask the question if we can still use the primitive P if, instead of a uniformly random U, we only have some arbitrary (n,k)-source X?

[5] In contrast, for *indistinguishability* games we typically require that $\Pr[P^{\mathcal{A}}(U)=1] \leq \frac{1}{2}+\delta$.

To formally answer this question, we would like a function $\mathsf{UExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ (seeded unpredictability extractor) such that, for all attackers $\mathcal{A}$ (in some complexity class), $\Pr[P^{\mathcal{A}(S)}(\mathsf{UExt}(X;S)) = 1] \le \varepsilon$, where the seed S is chosen uniformly at random and given to the attacker, and ε is not much larger than δ. Since we do not wish to assume much about the application P or the attacker $\mathcal{A}$, we can roll them up into a unified adversarial "distinguisher" defined by $D(R,S) := P^{\mathcal{A}(S)}(R)$. By definition, if $R = U$ is random and independent of S, then $\Pr[D(U,S) = 1] = \Pr[P^{\mathcal{A}(S)}(U) = 1] \le \delta$. On the other hand, we need to ensure that $\Pr[P^{\mathcal{A}(S)}(\mathsf{UExt}(X;S)) = 1] = \Pr[D(\mathsf{UExt}(X;S),S) = 1] \le \varepsilon$ for some ε which is not much larger than δ. This motivates the following definition of unpredictability extractor which ensures that the above holds for *all* distinguishers D.

Definition 3.1 (UExtract)**.** *We say that a function $D : \{0,1\}^m \times \{0,1\}^d \to \{0,1\}$ is a δ-distinguisher if $\Pr[D(U,S) = 1] \le \delta$ where (U,S) is uniform over $\{0,1\}^m \times \{0,1\}^d$. A function $\mathsf{UExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a* (k,δ,ε)-unpredictability extractor (UExtract) *if for any (n,k)-source X and any δ-distinguisher D, we have $\Pr[D(\mathsf{UExt}(X;S),S) = 1] \le \varepsilon$ where S is uniform over $\{0,1\}^d$.*

Notice that the above definition is essentially the same as that of standard extractors except that: (1) we require that the distinguisher has a "small" probability δ of outputting 1 on the uniform distribution, and (2) we only require a one-sided error that the probability of outputting 1 does not increase too much. A similar notion was also proposed by [RTS00] and called a "slice extractor".

Toward the goal of understanding unpredictability extractors, we show tight connections between the above definition and two seemingly unrelated notions. Firstly, we define "condensers for min-entropy" and show that the they yield "good" unpredictability extractors. Second, we define something called "balanced hash functions" and show that they yield good condensers, and therefore also good unpredictability extractors. Lastly, we show that unpredictability extractors also yield balanced hash functions, meaning that all three notions are essentially equivalent up to a small gap in parameters.

Definition 3.2 (Condenser)**.** *A function $\mathsf{Cond} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k,ℓ,ε)-condenser if for all (n,k)-sources X, and a uniformly random and independent seed S over $\{0,1\}^d$, the joint distribution $(S,\mathsf{Cond}(X;S))$ is ε-statistically-close to some joint distribution (S,Y) such that, for all $s \in \{0,1\}^d$, $\mathbf{H}_\infty(Y|S=s) \ge m-\ell$.*

Lemma 3.3 (Condenser $\Rightarrow$ UExtract)**.** *Any (k,ℓ,ε)-condenser is a (k,δ,ε^*)-UExtract where $\varepsilon^* = \varepsilon + 2^\ell\delta$.*

Proof. Let $\mathsf{Cond} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a (k,ℓ,ε)-condenser and let X be an (n,k)-source. Let S be uniform over $\{0,1\}^d$, so that, by definition, there is a joint distribution (S,Y) which has statistical distance at most ε from $(S,\mathsf{Cond}(X;S))$ such that $\mathbf{H}_\infty(Y|S=s) \ge m-\ell$ for all $s \in \{0,1\}^d$. Therefore, for any δ-distinguisher D, we have

$$\begin{aligned}
\Pr[D(\mathsf{Cond}(X;S),S)=1] &\le \varepsilon + \Pr[D(Y,S)=1] \\
&= \varepsilon + \sum_{y,s} \Pr[S=s]\Pr[Y=y|S=s]\Pr[D(y,s)=1] \\
&\le \varepsilon + \sum_{y,s} 2^{-d}2^{-\mathbf{H}_\infty(Y|S=s)}\Pr[D(y,s)=1] \\
&\le \varepsilon + 2^\ell \sum_{y,s} 2^{-(m+d)}\Pr[D(y,s)=1] \le \varepsilon + 2^\ell\delta.
\end{aligned}$$

□

Definition 3.4 (Balanced Hashing)**.** *Let $h := \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^d}$ be a hash function family. For $\mathcal{X} \subseteq \{0,1\}^n, s \in \{0,1\}^d, x \in \mathcal{X}$ we define $\mathsf{Load}_\mathcal{X}(x,s) := |\{x' \in \mathcal{X} : h_s(x') = h_s(x)\}|$.*[6] *We say that*

[6]Note that we allow $x' = x$ and so $\mathsf{Load}_\mathcal{X}(x,s) \ge 1$.

the family h is (k,t,ε)-balanced if for all $\mathcal{X} \subseteq \{0,1\}^n$ of size $|\mathcal{X}| = 2^k$, we have

$$\Pr\left[\mathsf{Load}_{\mathcal{X}}(X,S) > t2^{k-m}\right] \le \varepsilon$$

where S, X are uniformly random and independent over $\{0,1\}^d, \mathcal{X}$ respectively.

Lemma 3.5 (Balanced $\Rightarrow$ Condenser). *Let $\mathcal{H} := \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^d}$ be a (k,t,ε)-balanced hash function family. Then the function $\mathsf{Cond} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ defined by $\mathsf{Cond}(x;s) = h_s(x)$ is a (k,ℓ,ε)-condenser for $\ell = \log(t)$.*

Proof. Without loss of generality, we can restrict ourselves to showing that Cond satisfies the condenser definition for every *flat source* X which is uniformly random over some subset $\mathcal{X} \subseteq \{0,1\}^n$, $|\mathcal{X}| = 2^k$. Let us take such a source X over the set $\mathcal{X}$, and define a *modified* hash family $\tilde{h} = \{\tilde{h}_s : \mathcal{X} \to \{0,1\}^m\}_{s\in\{0,1\}^d}$ which depends on $\mathcal{X}$ and essentially "re-balances" h on the set $\mathcal{X}$. In particular, for every pair (s,x) such that $\mathsf{Load}^h_{\mathcal{X}}(x,s) \le t2^{k-m}$ we set $\tilde{h}_s(x) := h_s(x)$, and for all other pairs (s,x) we define $\tilde{h}_s(x)$ in such a way that $\mathsf{Load}^{\tilde{h}}_{\mathcal{X}}(x,s) \le t2^{k-m}$ (the super-script is used to denote the hash function with respect to which we are computing the load). It is easy to see that this "re-balancing" is always possible. We use the re-balanced hash function $\tilde{h}$ to define a joint distribution (S,Y) by choosing S uniformly at random over $\{0,1\}^d$, choosing X uniformly/independently over $\mathcal{X}$ and setting $Y = \tilde{h}_S(X)$. It's easy to check that the statistical distance between $(S, \mathsf{Cond}(X;S))$ and (S,Y) is at most $\Pr[h_S(X) \neq \tilde{h}_S(X)] \le \Pr[\mathsf{Load}^h_{\mathcal{X}}(X,S) > t2^{k-m}] \le \varepsilon$. Furthermore, for every $s \in \{0,1\}^d$, we have:

$$\begin{aligned}\mathbf{H}_\infty(Y|S=s) &= -\log(\max_y \Pr[Y=y|S=s]) \\ &= -\log(\max_y \Pr[X \in \tilde{h}_s^{-1}(y)]) \ge -\log(t2^{k-m}/2^k) = m - \log t.\end{aligned}$$

Therefore Cond is a $(k, \ell = \log t, \varepsilon)$-condenser. $\square$

Lemma 3.6 (UExtract $\Rightarrow$ Balanced). *Let $\mathsf{UExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a (k,δ,ε)-UExtractor for some, $\varepsilon > \delta > 0$. Then the hash family $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^d}$ defined by $h_s(x) = \mathsf{UExt}(x;s)$ is $(k,\varepsilon/\delta,\varepsilon)$-balanced.*

Proof. Let $t = \varepsilon/\delta$ and assume that $\mathcal{H}$ is *not* (k,t,ε)-balanced. Then there exists some set $\mathcal{X} \subseteq \{0,1\}^n$, $|\mathcal{X}| = 2^k$ such that $\hat{\varepsilon} := \Pr[\mathsf{Load}_{\mathcal{X}}(X,S) > t2^{k-m}] > \varepsilon$ where X is uniform over $\mathcal{X}$ and S is uniform over $\{0,1\}^d$. Let $\mathcal{X}_s \subseteq \mathcal{X}$ be defined by $\mathcal{X}_s := \{x \in \mathcal{X} : \mathsf{Load}_{\mathcal{X}}(x,s) > t2^{k-m}\}$ and let $\varepsilon_s \stackrel{\text{def}}{=} |\mathcal{X}_s|/2^k$. By definition $\hat{\varepsilon} = \sum_s 2^{-d}\varepsilon_s$. Define $\mathcal{Y}_s \subseteq \{0,1\}^m$ via $\mathcal{Y}_s := h_s(\mathcal{X}_s)$. Now by definition, each $y \in \mathcal{Y}_s$ has at least $t2^{k-m}$ pre-images in $\mathcal{X}_s$ and therefore $\delta_s \stackrel{\text{def}}{=} |\mathcal{Y}_s|/2^m \le |\mathcal{X}_s|/(t2^{k-m}2^m) \le \varepsilon_s/t$ and $\delta := \sum_s 2^{-d}\delta_s \le \hat{\varepsilon}/t$.

Define the distinguisher D via $D(y,s) = 1$ iff $y \in \mathcal{Y}_s$. Then D is a δ-distinguisher for $\delta \le \hat{\varepsilon}/t \le \varepsilon/t$ but $\Pr[D(h_S(X),S) = 1] = \hat{\varepsilon} \ge \varepsilon$. Therefore, UExt is not a $(k,\varepsilon/t,\varepsilon)$-UExtractor. $\square$

Summary. Taking all of the above lemmata together, we see that they are close to tight. In particular, for any $\varepsilon > \delta > 0$, we get:

$$(k,\delta,\varepsilon)\text{-UExt} \stackrel{Lem.3.6}{\Rightarrow} (k,\varepsilon/\delta,\varepsilon)\text{-Balanced} \stackrel{Lem.3.5}{\Rightarrow} (k,\log(\varepsilon/\delta),\varepsilon)\text{-Condenser} \stackrel{Lem.3.3}{\Rightarrow} (k,\delta,2\varepsilon)\text{-UExt}$$

4 Constructing Unpredictability Extractors

Given the connections established in the previous section, we have paved the road for constructing unpredictability extractors via balanced hash functions, which is a seemingly simpler property to analyze. Indeed, we will give relatively simple lemmas showing that "sufficiently independent" hash functions are balanced. This will lead to the following parameters (restating Theorem 1.3 from the introduction):

Theorem 4.1. *There exists an efficient (k, δ, ε)-unpredictability extractor* $\mathsf{UExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *for the following parameters:*

1. *When $k = m$ (no entropy loss), we get $\varepsilon = (1 + \log(1/\delta))\delta$.*
2. *When $k \geq m + \log\log 1/\delta + 4$, we get $\varepsilon = 3\delta$.*
3. *In general, $\varepsilon = O(1 + 2^{m-k}\log(1/\delta))\delta$.*

In all cases, the function UExt *is simply a $(\log(1/\delta) + O(1))$-wise independent hash function and the seed length is $d = O(n\log(1/\delta))$.*

Although these constructions may already be practical, the level of independence we will need is $O(\log 1/\delta)$, which will result in a large seed $O(n\log(1/\delta))$. We will show how to achieve similar parameters with a shorter seed $O(n\log k)$ in Section 4.2. We now proceed to prove all of the parts of Theorem 4.1 by constructing "good" balanced hash functions and using our connections between balanced hashing and unpredictability extractors from the previous section.

4.1 Sufficient Independence Provides Balance

First we start with a simple case where the output m is equal to the entropy k.

Lemma 4.2. *Let $\mathcal{H} := \{h_s : \{0,1\}^n \to \{0,1\}^k\}_{s\in\mathcal{S}}$ be $(t+1)$-wise independent. Then it is (k, t, ε)-balanced where $\varepsilon \leq \left(\frac{e}{t}\right)^t$ and e is the base of the natural logarithm.*

Proof. Fix any set $\mathcal{X} \subseteq \{0,1\}^n$ of size $|\mathcal{X}| = 2^k$. Let X be uniform over $\mathcal{X}$ and S be uniform/independent over $\{0,1\}^d$. Then

$$\begin{aligned}
\Pr[\mathsf{Load}_{\mathcal{X}}(X,S) > t] &\leq \Pr[\ \exists \mathcal{C} \subseteq \mathcal{X}, |\mathcal{C}| = t \ \ \forall x' \in \mathcal{C} \ : \ h_S(x') = h_S(X) \wedge x' \neq X] \\
&\leq \sum_{\mathcal{C}\subseteq\mathcal{X}, |\mathcal{C}|=t} \Pr[\forall x' \in \mathcal{C} \ : \ h_S(x') = h_S(X) \wedge x' \neq X] \\
&\leq \binom{2^k}{t} 2^{-tk} \leq \left(\frac{e2^k}{t}\right)^t 2^{-tk} \leq \left(\frac{e}{t}\right)^t .
\end{aligned}$$

□

Corollary 4.3. *For any $0 < \varepsilon < 2^{-2e}$, any $\delta > 0$, a $(\log(1/\varepsilon) + 1)$-wise independent hash family $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^k\}_{s\in\{0,1\}^d}$ is:*

$$(k, \log(1/\varepsilon), \varepsilon)\text{-balanced}, \quad (k, \log\log(1/\varepsilon), \varepsilon)\text{-condenser}, \quad (k, \delta, \log(1/\varepsilon)\delta + \varepsilon)\text{-UExtractor}.$$

Setting $\delta = \varepsilon$, we get a $(k, \delta, (1 + \log(1/\delta))\delta)$-UExtractor.

Proof. Set $t = \log(1/\varepsilon)$ in Lemma 4.2 and notice that $\left(\frac{e}{t}\right)^t \leq 2^{-t} \leq \varepsilon$ as long as $t \geq 2e$. □

This establishes part (1) of Theorem 4.1. Next we look at a more general case where k may be larger than m. This also covers the case $k = m$ but gets a somewhat weaker bound. It also requires a more complex tail bound for q-wise independent variables.

Lemma 4.4. *Let $\mathcal{H} := \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\mathcal{S}}$ be $(q+1)$-wise independent. Then, for any $\alpha > 0$, it is $(k, 1+\alpha, \varepsilon)$-balanced where $\varepsilon \leq 8\left(\frac{q2^{k-m}+q^2}{(\alpha 2^{k-m}-1)^2}\right)^{q/2}$.*

Proof. Let $\mathcal{X} \subseteq \{0,1\}^n$ be a set of size $|\mathcal{X}| = 2^k$, X be uniform over $\mathcal{X}$, and S be uniform/independent over $\{0,1\}^d$. Define the indicator random variables Define $C(x^*, x)$ to be 1 if $h_S(x) = h_S(x^*)$ and 0 otherwise. Then:

$$\begin{aligned}
\Pr[\mathsf{Load}_{\mathcal{X}}(X,S) > (1+\alpha)2^{k-m}] &= \sum_{x^* \in \mathcal{X}} \Pr[X = x^*] \Pr[\mathsf{Load}_{\mathcal{X}}(x^*, S) > (1+\alpha)2^{k-m}] \\
&= 2^{-k} \sum_{x^* \in \mathcal{X}} \Pr\left[\sum_{x \in \mathcal{X} \setminus \{x^*\}} C(x^*, x) + 1 > (1+\alpha)2^{k-m}\right] \\
&\leq 8 \left(\frac{q2^{k-m} + q^2}{(\alpha 2^{k-m} - 1)^2}\right)^{q/2}
\end{aligned}$$

Where the last line follows from the tail inequality Lemma 2.1 with the random variables $\{C(x^*, x)\}_{x \in \mathcal{X} \setminus \{x^*\}}$ which are q-wise independent and have expected value $\mu = \mathbb{E}[\sum_{x \in \mathcal{X} \setminus \{x^*\}} C(x^*, x)] = (2^k - 1)2^{-m} \leq 2^{k-m}$, and by setting $A = (1+\alpha)2^{k-m} - 1 - \mu \geq \alpha 2^{k-m} - 1$; recall that $C(x^*, x^*)$ is always 1 and $C(x^*, x)$ for $x \neq x^*$ is 1 with probability 2^{-m}. □

Corollary 4.5. *For any $0 < \varepsilon < 2^{-7}$, $k \geq m + \log\log(1/\varepsilon) + 4$, a $(\log(1/\varepsilon) + 4)$-wise independent hash function family $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \{0,1\}^d}$ is:*

$(k, 2, \varepsilon)$-balanced, $(k, 1, \varepsilon)$-condenser, $(k, \delta, 2\delta + \varepsilon)$-UExt for any $\delta > 0$.

Setting $\delta = \varepsilon$, it is a $(k, \delta, 3\delta)$-UExt.

Proof. Set $q = \log(1/\varepsilon) + 3$, $\alpha = 1$ and $2^{k-m} = 5q$. Then we apply Lemma 4.4

$$8 \left(\frac{q2^{k-m} + q^2}{(\alpha 2^{k-m} - 1)^2}\right)^{q/2} \leq 8 \left(\frac{6q^2}{(5q-1)^2}\right)^{q/2} \leq 8 \left(\frac{1}{4}\right)^{q/2} \leq 8(2^{-q}) \leq \varepsilon.$$

The second step assumes $q > 10$ meaning that $\varepsilon < 2^{-7}$. □

The above corollary establishes part (2) of Theorem 4.1. The next corollary gives us a general bound which establishes part (3) of the theorem. Asymptotically it implies both Corollary 4.5 and Corollary 4.3 but with worse constants.

Corollary 4.6. *For any $\varepsilon > 0$ and $q := \log(1/\varepsilon) + 3$, a $(q+1)$-wise independent hash function family $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s \in \{0,1\}^d}$ is $(k, 1+\alpha, \varepsilon)$-balanced for*

$$\alpha = 4\sqrt{q2^{m-k} + (q2^{m-k})^2} = O(2^{m-k}\log(1/\varepsilon) + 1).$$

By setting $\delta = \varepsilon$, a $(\log \frac{1}{\delta} + 4)-$wise independent hash function is a $(k, \delta, O(1 + 2^{m-k}\log\frac{1}{\delta})\delta)$-UExtactor.

Proof. The first part follows from Lemma 4.4 by noting that

$$8 \left(\frac{q2^{k-m} + q^2}{(\alpha 2^{k-m} - 1)^2}\right)^{q/2} \leq 8 \left(\frac{q2^{k-m} + q^2}{\frac{1}{4}(\alpha 2^{k-m})^2}\right)^{q/2} \leq 8 \left(\frac{1}{4}\right)^{q/2} \leq \varepsilon.$$

For the second part, we can consider two cases. If $q2^{m-k} \leq 1$ then $\alpha \leq 4\sqrt{2}$ and we are done. Else, $\alpha \leq 4\sqrt{2}(q2^{m-k}) = 4\sqrt{2}(\log(1/\varepsilon) + 3)2^{m-k}$. □

4.2 Minimizing the Seed Length

In both of the above constructions (Corollary 4.3, Corollary 4.5), to get an (k, δ, ε)-UExtractor, we need a $O(\log(1/\varepsilon))$-wise independent hash function $h_s : \{0,1\}^n \to \{0,1\}^m$, which requires a seed-length $d = O(\log(1/\varepsilon) \cdot n)$. Since in many applications, we envision $\varepsilon \approx 2^{-k}$, this gives a seed $d = O(kn)$. We should contrast this with standard extractors constructed using universal hash functions (via the leftover-hash lemma), where the seed is $d = n$. We now show how to optimize the seed length of UExtractors to $O(n \log k)$. We adapt the technique of Celis et al. [CRSW11] which shows how to construct hash functions with a small seed that achieve essentially optimal "max-load" (e.g., minimize the hash value with the most items inside it). We show that a lightly modified analysis can also be used to show that such hash functions are "balanced" with essentially optimal parameters.

We start by recalling the notion of q-wise δ-dependent hash functions.

Definition 4.7 ((Almost) Independent Hashing)**.** *A hash family $\mathcal{H} : \{h : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^d}$ is q-wise δ-dependent if for any distinct $x_1, \ldots, x_q \in \{0,1\}^n$,*

$$(h_S(x_1), \ldots, h_S(x_q)) \approx_\delta (U_1, \ldots, U_q)$$

where S is uniformly random over $\{0,1\}^d$ and U_i are uniformly random/independent over $\{0,1\}^m$.

Such almost independent hash functions can be constructed using ε-biased distributions [NN93, AGHP92]. The following parameters are stated in [CRSW11].

Lemma 4.8. *For any integers n, ℓ, there exists a family of q-wise δ-dependent hash functions from n-bits to ℓ-bits with seed-length $d = O(n + \ell \cdot q + \log(1/\delta))$.*

We will also rely on the following tail-bound from [CRSW11].

Lemma 4.9 ([CRSW11], Lemma 2.2)**.** *Suppose that $X_1, \ldots, X_n$ are q-wise δ-dependent random variables taking values in $[0,1]$. Let $X := X_1 + \cdots + X_n$ and define $\mu := \mathbf{E}[X]$ be the expectation of the sum. Then, for any $\alpha > 0$, $\Pr[X \geq (1+\alpha)\mu] \leq 2\left(\frac{qn}{(\alpha\mu)^2}\right)^{q/2} + \delta\left(\frac{n}{\alpha\mu}\right)^q$.*

Construction. Our goal is to construct a hash function family $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^k\}_{s\in\{0,1\}^d}$ such that $\mathcal{H}$ is (k, t, ε)-balanced for some small $\varepsilon \approx 2^{-t}$. Assume that $n \geq k \geq t$. We will choose h_s to be a concatenation of several hash functions with gradually increasing levels of independence q_i and gradually decreasing output size ℓ_i while keeping the product $q_i\ell_i = O(t)$ essentially constant. More precisely, let $\mathcal{H}_1, \ldots, \mathcal{H}_r, \mathcal{H}_{r+1}$ be hash function families, where each family $\mathcal{H}_i = \{h_{s_i} : \{0,1\}^n \to \{0,1\}^{\ell_i}\}_{s_i\in\{0,1\}^{d_i}}$ is q_i-wise δ_i-dependent with the parameters q_i, ℓ_i and δ_i being chosen as follows:

- For $i = 1, \ldots, r$ (where r will be specified later), set ℓ_i so that $\sum_{j=1}^{i} \ell_i = \lfloor\left(1 - \left(\frac{3}{4}\right)^i\right) k\rfloor$. Note that this means $\ell_i = \frac{1}{4}(\frac{3}{4})^{i-1}k \pm 1$ and $k - \sum_{j=1}^{i} \ell_i = 3\ell_i \pm 4 = 4\ell_{i+1} \pm 5$.
- For $i = 1, \ldots, r$, set $q_i := 4\lceil t/\ell_i\rceil + 1$.
- Set r be the largest integer such that $\ell_r \geq \log t + 2\log\log_{4/3} k + 7$. Note that $r \leq \log_{4/3} k = O(\log k)$.
- Set $\ell_{r+1} := k - \sum_{i=1}^{r} \ell_i$. This gives $\ell_{r+1} = O(\log t + \log\log k)$. Set $q_{r+1} = 4t + 1$
- For $i = 1, \ldots, r$, set $\delta_i := 2^{-18k}$ and set $\delta_{r+1} = 2^{-t\ell_{r+1}-2t} = 2^{-O(k\log k)}$.

Let $\mathcal{H} := \mathcal{H}_1 \circ \ldots \circ \mathcal{H}_{r+1}$ meaning that $\mathcal{H} = \{h_s : \{0,1\}^n \to \{0,1\}^k\}_{s\in\{0,1\}^d}$ is defined by

$$h_s(x) := h_{s_1}(x)||\cdots||h_{s_{r+1}}(x)$$

where $s = (s_1, \ldots, s_r, s_{r+1})$ and $h_{s_i} \in \mathcal{H}_i$ and '$||$' denotes concatenation. Notice that, using the parameters of Lemma 4.8 for the function families $\mathcal{H}_i$, we can get the total seed-length to be $d = |s| = \sum_{i=1}^{r+1} d_i = O(n \log k)$, assuming $n \geq k \geq t$.

Theorem 4.10. *The above family* $\mathcal{H} : \{h_s : \{0,1\}^n \to \{0,1\}^k\}_{s\in\{0,1\}^d}$ *is* $(k, t, 2^{-t})$*-balanced for any*

$$n \geq k \geq t \geq \log\log_{4/3} k + 4 = \log\log k + O(1).$$

The seed length is $d = O(n \log k)$*. In particular,* $\mathcal{H}$ *is also* $(k, \log(t), 2^{-t})$*-condenser and a* $(k, \delta, t\delta + 2^{-t})$*-UExtract for any* $\delta > 0$*.*

We can also consider the family $\mathcal{H}' : \mathcal{H}_1 \circ \ldots \circ \mathcal{H}_r$, defined analogously to the above but excluding $\mathcal{H}_{r+1}$, so that $\mathcal{H}' = \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^{d'}}$ where $d' = \sum_{i=1}^r d_i$ and $m = k - \ell_{r+1} = k - O(\log t + \log\log k)$.

Theorem 4.11. *The above family* $\mathcal{H}' : \{h_s : \{0,1\}^n \to \{0,1\}^m\}_{s\in\{0,1\}^{d'}}$ *is* $(k, (e+1), \varepsilon)$*-balanced for any*

$$n \geq k \geq t \geq \log\log_{4/3} k + 4 = \log\log k + O(1), m = k - \ell_{r+1} = k - O(\log t + \log\log k)$$

with $\varepsilon = 2^{-t}$*. The seed length is* $d = O(n \log k)$*. In particular,* $\mathcal{H}$ *is also* $(k, \log(e+1), \varepsilon)$*-condenser and a* $(k, \delta, (e+1)\delta + \varepsilon)$*-UExtract for any* $\delta > 0$*.*

Proof of Theorem 4.10 and Theorem 4.11. We start with the proof of Theorem 4.10. Let us choose some arbitrary set $\mathcal{X} \subseteq \{0,1\}^n$, $|\mathcal{X}| = 2^k$ and some arbitrary $x \in \mathcal{X}$. For a seed $s = (s_1, \ldots, s_{r+1}) \leftarrow \{0,1\}^{d=\sum_i^{r+1} d_i}$ we will iteratively define $\mathcal{X}_0 = \mathcal{X} \setminus \{x\}$ and for $i > 0$, $\mathcal{X}_i = \{x' \in \mathcal{X}_{i-1} : h_{s_i}(x') = h_{s_i}(x)\}$. We start with the following lemma:

Lemma 4.12. *Let* $\alpha = 1/r$ *and assume that for some* $i \in \{1, \ldots, r\}$*, we have* $|\mathcal{X}_{i-1}| \leq (1+\alpha)^{i-1} 2^{k-\sum_{j=1}^{i-1}\ell_j}$*. Then*

$$\Pr_{s_i \leftarrow \{0,1\}^{d_i}}\left[|\mathcal{X}_i| > (1+\alpha)^i 2^{k-\sum_{j=1}^{i}\ell_j}\right] < 3 \cdot 2^{-2t}$$

Proof. Without loss of generality, assume the worst-case scenario that $|\mathcal{X}_{i-1}| = \lfloor (1+\alpha)^{i-1} 2^{k-\sum_{j=1}^{i-1}\ell_j} \rfloor \geq 2^{k-\sum_{j=1}^{i-1}\ell_j}$. In this case, we can write the above as:

$$\begin{aligned}
\Pr_{s_i \leftarrow \{0,1\}^{d_i}}\left[|\mathcal{X}_i| > (1+\alpha)^i 2^{k-\sum_{j=1}^{i-1}\ell_j}\right] &\leq \Pr_{s_i \leftarrow \{0,1\}^{d_i}}\left[\sum_{x' \in \mathcal{X}_{i-1}} \left\{ \begin{array}{cc} 1 & \text{if } h_{s_i}(x') = h_{s_i}(x) \\ 0 & \text{otherwise} \end{array} \right\} > (1+\alpha)\frac{|\mathcal{X}_{i-1}|}{2^{\ell_i}}\right] \\
&\leq 2\left(\frac{4\lceil t/\ell_i\rceil |\mathcal{X}_{i-1}|}{(\alpha|\mathcal{X}_{i-1}|/2^{\ell_i})^2}\right)^{2\lceil t/\ell_i\rceil} + \delta_i \left(\frac{|\mathcal{X}_{i-1}|}{\alpha|\mathcal{X}_{i-1}|/2^{\ell_i}}\right)^{4\lceil t/\ell_i\rceil} \qquad (5) \\
&\leq 2\left(\frac{4\lceil t/\ell_i\rceil 2^{2\ell_i} r^2}{2^{k-\sum_{j=1}^{i-1}\ell_j}}\right)^{2\lceil t/\ell_i\rceil} + \delta_i (2^{4\lceil t/\ell_i\rceil(\ell_i + \log r)}) \\
&\leq 2\left(\frac{4\lceil t/\ell_i\rceil r^2}{2^{2\ell_i - 5}}\right)^{2\lceil t/\ell_i\rceil} + \delta_i 2^{4(t+\ell_i+t\log r/\ell_i + \log r)} \qquad (6) \\
&\leq 2 \cdot 2^{-2t} + 2^{-2t} \leq 3 \cdot 2^{-2t} \qquad (7)
\end{aligned}$$

Line (5) follows from Lemma 4.9 and the fact that the variables being summed are $(q_i - 1)$-wise δ_i-dependent with mean $\mu = \frac{|\mathcal{X}_{i-1}|}{2^{\ell_i}}$. Line (6) follows from the fact that $k - \sum_{j=1}^{i-1}\ell_j \geq 4\ell_i - 5$. Line (7) follows from the fact that

$$2^{\ell_i} \geq 2^{\ell_r} \geq 2^{\log t + 2\log r + 7} \geq 4tr^2 2^5 \geq 4\lceil t/\ell_i\rceil r^2/2^{-5}$$

which gives the bound for the left-hand summand, and

$$4(t + \ell_i + t\log r/\ell_i + \log r) \leq 4(t + k + t + \log\log_{4/3} k) \leq 16k$$

which gives the bound for the right hand summand as long as $\delta_i \leq 2^{-18k} \leq 2^{-16k-2t}$.

□

By using Lemma 4.12 inductively, we get $\Pr_{s_1,\ldots,s_r}\left[|\mathcal{X}_r| \geq (1+1/r)^r 2^{k-\sum_{j=1}^r \ell_j}\right] \leq (3r)2^{-2t}$. Since $(1+1/r)^r \leq e$ and $\ell_{r+1} = k - \sum_{j=1}^r \ell_j$, we can rewrite the above as:

$$\Pr_{s_1,\ldots,s_r}\left[|\mathcal{X}_r| \geq e2^{\ell_{r+1}}\right] \leq (3r)2^{-2t} \tag{8}$$

Assuming that $|\mathcal{X}_r| \leq e2^{\ell_{r+1}}$. Then

$$\begin{aligned}
\Pr[|\mathcal{X}_{r+1}| \geq t] &= \Pr_{s_{r+1}\leftarrow\{0,1\}^{d_{r+1}}}[\ \exists \mathcal{C} \subseteq \mathcal{X}_r, |\mathcal{C}| = t \ \ \forall x' \in \mathcal{C} \ : \ h_{s_{r+1}}(x') = h_{s_{r+1}}(x)] \\
&\leq \sum_{\mathcal{C}\subseteq\mathcal{X}_r, |\mathcal{C}|=t} \Pr[\forall x' \in \mathcal{C} \ : \ h_{s_{r+1}}(x') = h_{s_{r+1}}(x)] \\
&\leq \binom{|\mathcal{X}_r|}{t}(2^{-t\ell_{r+1}} + \delta_{r+1}) \left(\frac{e^2 2^{\ell_{r+1}}}{t}\right)^t (2^{-t\ell_{r+1}} + \delta_{r+1}) \\
&\leq (e^2/t)^t + \delta_{r+1}2^{t\ell_{r+1}} \leq 2^{-2t} + 2^{-2t} \leq 2 \cdot 2^{-2t}.
\end{aligned}$$

Therefore, altogether, we have

$$\Pr_{s\leftarrow\{0,1\}^s}[\mathcal{X}_{r+1} \geq t] \leq 3(r+1)2^{-2t} \leq 3(\log_{4/3} k + 1)2^{-2t} \leq 2^{-(t+1)}$$

since we chose t so that $2^{t-1} \geq 3(\log_{4/3} k + 1)$. Moreover, we have $\mathsf{Load}^{\mathcal{H}}_{\mathcal{X}}(x,s) = |\mathcal{X}_{r+1}| + 1$ (since we must also include the point x itself). Therefore $\Pr[\mathsf{Load}^{\mathcal{H}}_{\mathcal{X}}(X,S) \geq t+1] \leq 2^{-(t+1)}$ which proves the Theorem 4.10.

To prove Theorem 4.11, we go back to equation (8) and notice that $\Pr_{s_1,\ldots,s_r}\left[|\mathcal{X}_r| \geq e2^{\ell_{r+1}}\right] \leq (3r)2^{-2t} \leq 3(r+1)2^{-2t} \leq 2^{-t}$. Combining this with the fact that $\mathsf{Load}^{\mathcal{H}'}_{\mathcal{X}}(x,s) = |\mathcal{X}_{r+1}| + 1$, we get for every $\mathcal{X}$ and $x \in \mathcal{X}$:

$$\Pr_{s\leftarrow\{0,1\}^{d'}}[\mathsf{Load}^{\mathcal{H}'}_{\mathcal{X}}(x,s) \geq (e+1)2^{k-m}] = \Pr[|\mathcal{X}_{r+1}| \geq e2^{\ell_{r+1}}] \leq 2^{-t}.$$

4.3 A Probabilistic Method Bound

We also give a probabilistic method argument showing the existence of unpredictability extractors with very small seed length $d \approx \log(1/\delta) + \log(n-k)$. In other words, unpredictability extractors with small entropy loss do not, in principle, require a larger seed than standard randomness extractors (with much larger entropy loss).

See e.g., Theorem 4.1 of [MR95], for the following Chernoff tail-bound.

Lemma 4.13 (Multiplicative Chernoff Bound)**.** *Let $X_1, \ldots, X_n$ be independent random variables taking on values in $\{0,1\}$ with $\Pr[\mathcal{X}_i = 1] = \delta_i$ and $\mathbb{E}[\sum_{i=1}^n X_i] = \delta n$. Then, for any $\alpha > 1$, we have*

$$\Pr\left[\sum_{i=1}^n X_i > \alpha\delta n\right] < \left(\frac{e^{(\alpha-1)}}{\alpha^\alpha}\right)^{\delta n} \leq \left(\frac{e}{\alpha}\right)^{\alpha\delta n}.$$

We use the above tail-bound to prove the following theorem.

Theorem 4.14. *There exists a (k,δ,ε)-UExtract with input length n, output length m, seed length d as long as:*

$$\varepsilon \geq 2e\delta + \log(e/\delta)\delta 2^{m-k} \quad , \quad d \geq \log(1/\delta) + \log(n-k) + 1.$$

In particular if $k = m$, we can set $\varepsilon = \delta(3 + \log(1/\delta))$, and if $k \geq m + \log(\log(1/\delta)) + 1$ we can set $\varepsilon = 2\delta$.

Proof. We use the probabilistic method argument. For simplicity of notation, let $N = 2^n, K = 2^k, D = 2^d, M = 2^m$. Let $R : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be chosen uniformly at random from the set $\mathcal{R}$ of all such functions. Then R *fails* to be a (k, δ, ε)-Uextract if there exists some subset $\mathcal{X} \subseteq \{0,1\}^n, |X| = K$ and some (deterministic) δ-distinguisher $\mathcal{D}$ such that $|\{x \in \mathcal{X}, s \in \{0,1\}^d : \mathcal{D}(R(x;s), s) = 1\}| > \varepsilon KD$. For a fixed $\mathcal{X}, \mathcal{D}$ and a uniformly random R, we can define indicator random variables $\{V^{\mathcal{D}}_{x,s}\}_{x\in\mathcal{X},s\in\{0,1\}^d}$ via $V^{\mathcal{D}}_{x,s} = 1$ iff $\mathcal{D}(R(x;s), s) = 1$. These variables are mutually independent (but not identically) distributed with $\Pr[V^{\mathcal{D}}_{x,s} = 1] = \delta_s := \frac{|\{y\in\{0,1\}^m : \mathcal{D}(y,s)=1\}|}{2^m}$ and $\mathbb{E}[\sum_{x,s} V^{\mathcal{D}}_{x,s}] = \sum_{x,s} \delta_s = \delta KD$. Therefore, we have:

$$
\begin{aligned}
\Pr_R[R \text{ is not a } (k,\delta,\varepsilon)\text{-UExtract}] &\leq \Pr_R\left[\exists \mathcal{D}, \mathcal{X} \quad \text{s.t.} \sum_{x\in\mathcal{X}, s\in\{0,1\}^d} V^P_{x,s} > \varepsilon DK\right] \\
&\leq \sum_{\mathcal{D},\mathcal{X}} \Pr_R\left[\sum_{x\in\mathcal{X}, s\in\{0,1\}^d} V^{\mathcal{D}}_{x,s} > (\varepsilon/\delta)\delta KD\right] \\
&< \binom{N}{K}\binom{MD}{\delta MD}\left(\frac{e}{(\varepsilon/\delta)}\right)^{\varepsilon KD} \\
&\leq \left(\frac{eN}{K}\right)^K \left(\frac{e}{\delta}\right)^{\delta MD}\left(\frac{e}{(\varepsilon/\delta)}\right)^{\varepsilon KD}
\end{aligned}
$$

where the third line follow by the Chernoff bound Lemma 4.13. Therefore, the above probability is strictly less than 1 as long as:

$$\left(\frac{(\varepsilon/\delta)}{e}\right)^{\varepsilon KD} \geq \left(\frac{eN}{K}\right)^K \left(\frac{e}{\delta}\right)^{\delta MD}$$

Taking the logarithms (base 2) of both sides, we get the equivalent requirement:

$$\varepsilon(\log(\varepsilon/\delta) - \log e) \geq (\log e + n - k)\frac{1}{D} + (\log e + \log(1/\delta))\delta\frac{M}{K}.$$

Assuming $D \geq (\log e + n - k)/\delta$ and $\varepsilon \geq 2e\delta$ the above is implied by:

$$\varepsilon \geq \delta + (\log e + \log(1/\delta))\delta\frac{M}{K}.$$

□

5 SRT Lower-Bound: Samplability Doesn't Improve Entropy Loss

In this section, we prove the 'SRT' conjecture of Dachman-Soled et al. [DGKM12], showing that randomness extractors need to incur a $2\log 1/\varepsilon$ entropy loss (difference between entropy and output length) even if we only require them to work for *efficiently samplable sources*. The lower-bound even holds if the extractor itself is not required to be efficient. The efficient source for which we show a counter-example is sampled via a 4-wise independent hash function. That is, we define the source $X = h_r(Z)$ where $Z \leftarrow \{0,1\}^k$ is chosen uniformly at random and $h_r : \{0,1\}^k \to \{0,1\}^n$ is chosen from some 4-wise independent hash function family. The choice of the seed r will need to be fixed non-uniformly; we show that for any "candidate extractor" Ext there is some seed r such that the above efficiently sampleable source makes $(\mathsf{Ext}(X;S), S)$ distinguishable from uniform with advantage $\approx 2^{(m-k)/2}$.

5.1 Preliminaries: Anti-Concentration Bounds

Lemma 5.1 ([Ber91], Theorem 2.3)**.** *For any random variable V, we have:* $\mathbb{E}[|V|] \geq \frac{\mathbb{E}[V^2]^{3/2}}{\mathbb{E}[V^4]^{1/2}}$.

Corollary 5.2. *Let $V_1, \ldots, V_q$ be 4-wise independent random variables over $\mathbb{R}$ such that for all $i \in [q]$ we have $\mathbb{E}[V_i] = 0, \mathbb{E}[V_i^2] \in [p/4, p]$, $\mathbb{E}[V_i^4] \leq p$ for some $p \geq 1/q$. Then $\mathbb{E}[|\sum_{i=1}^q V_i|] \geq \frac{1}{16}\sqrt{q \cdot p}$.*

Proof. Let us define $V := \sum_{i=1}^q V_i$. Then

$$\begin{aligned}
\mathbb{E}[V^2] &= \sum_{i,j \in [q]} \mathbb{E}[V_i \cdot V_j] = \sum_{i \in [q]} \mathbb{E}[V_i^2] \geq qp/4 \\
\mathbb{E}[V^4] &= \sum_{i,j,r,t \in [q]} \mathbb{E}[V_i \cdot V_j \cdot V_r \cdot V_t] = \sum_{i \in [q]} \mathbb{E}[V_i^4] + 3 \sum_{i \neq j \in [q]} \mathbb{E}[V_i^2]\,\mathbb{E}[V_j^2] \\
&\leq qp + 3(qp)^2 \leq 4(qp)^2
\end{aligned}$$

Therefore, by Lemma 5.1, we have

$$\mathbb{E}[|V|] \geq \frac{\mathbb{E}[V^2]^{3/2}}{\mathbb{E}[V^4]^{1/2}} \geq \frac{(\frac{1}{4}qp)^{3/2}}{(4(qp)^2)^{1/2}} = \frac{1}{16}\sqrt{qp}$$

□

5.2 Statement of Lower Bound

Let $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a candidate strong extractor, and let X be some random variable over $\{0,1\}^n$. Define the *distinguishability* of Ext on X via:

$$\begin{aligned}
\mathsf{Dist}(X) &\stackrel{\text{def}}{=} \frac{1}{2} \sum_{s \in \{0,1\}^d, y \in \{0,1\}^m} |\Pr[S = s, \mathsf{Ext}(X; s) = y] - \Pr[S = s, Y = y]| \\
&= \frac{1}{2^{d+1}} \sum_{s \in \{0,1\}^d, y \in \{0,1\}^m} \left| \Pr[\mathsf{Ext}(X, s) = y] - \frac{1}{2^m} \right|.
\end{aligned}$$

where S, Y are uniformly and independently distributed over $\{0,1\}^d$,$\{0,1\}^m$ respectively. Note that $\mathsf{Dist}(X)$ is simply the statistical distance between $(S, \mathsf{Ext}(X; S))$ and (S, U_m) where U_m is uniformly random m bit string.

Theorem 5.3. *For any (possibly inefficient) function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, any positive integer $k \geq m + 2$ such that $n > 3k - m + 14$, there exists a distribution X with $\mathbf{H}_\infty(X) \geq k$, which is efficiently samplable by a* poly(n)*-size circuit, such that* $\mathsf{Dist}(X) \geq 2^{(m-k)/2-8}$.

Alternatively, for any positive $k \geq m$ such that $n > k + \log(k) + 11$, there exists some distribution X with $\mathbf{H}_\infty(X) \geq k$, which is efficiently samplable by a poly(n)*-size circuit such that* $\mathsf{Dist}(X) \geq 2^{(m-k-\log(k))/2-9}$.

5.3 Proof of Lower Bound

Let $\mathcal{X} = \{x_1, \ldots, x_{2^k}\} \subseteq \{0,1\}^n$ be a multiset (i.e, we may have $x_i = x_j$ for $i \neq j$) and let X be a random variable distributed uniformly over $\mathcal{X}$ (i.e., to sample $x \leftarrow X$, choose random $i \in [2^k]$ and output x_i).

Define $\mathsf{Dist}(\mathcal{X}) \stackrel{\text{def}}{=} \mathsf{Dist}(X)$. Then we can write:

$$\begin{aligned}
\mathsf{Dist}(\mathcal{X}) &= \frac{1}{2^{d+1}} \sum_{s\in\{0,1\}^d, y\in\{0,1\}^m} \left| \Pr[\mathsf{Ext}(X,s)=y] - \frac{1}{2^m} \right| \\
&= \frac{1}{2^{d+1}} \sum_{s\in\{0,1\}^d, y\in\{0,1\}^m} \left| \frac{\sum_{i\in[2^k]} b_{i,s,y}}{2^k} - \frac{1}{2^m} \right| \\
&= \frac{1}{2^{d+k+1}} \sum_{s\in\{0,1\}^d, y\in\{0,1\}^m} \left| \sum_{i\in[2^k]} \left(b_{i,s,y} - 2^{-m}\right) \right|
\end{aligned}$$

where $b_{i,s,y}=1$ if $\mathsf{Ext}(x_i;s)=y$ and 0 otherwise.

Now let us choose the multiset $\mathcal{X}$ randomly via $\mathcal{X}=\{X_1,\ldots,X_{2^k}\}$ where the X_i are *4-wise independent* random variables, each of which is uniform over $\{0,1\}^n$. For example, let $\mathcal{H}$ be a 4-wise independent family of hash functions $h : \{0,1\}^k \to \{0,1\}^n$ and define $X_i = h(i)$ where the randomness is over the choice of $h \leftarrow \mathcal{H}$. Such hash functions can be efficient so that we can compute $h(i)$ in poly(n)-time. In that case, taking an expectation over the choice of $\mathcal{X}$, we can write:

$$\mathbb{E}[\mathsf{Dist}(\mathcal{X})] = \frac{1}{2^{d+k+1}} \sum_{s\in\{0,1\}^d, y\in\{0,1\}^m} \mathbb{E}\left[\left| \sum_{i\in[2^k]} \left(B_{i,s,y} - 2^{-m}\right) \right| \right]$$

where $B_{i,s,y}$ is an indicator random variable which is 1 if $\mathsf{Ext}(X_i;s)=y$ and 0 otherwise. We now prove the following:

Claim 5.4. *For all $s\in\{0,1\}^d, y\in\{0,1\}^m$, we have* $\mathbb{E}\left[\left|\sum_{i\in[2^k]}\left(B_{i,s,y}-2^{-m}\right)\right|\right] \geq \frac{1}{64}2^{(k-m)/2}$

Proof. Let us fix some arbitrary $s\in\{0,1\}^d, y\in\{0,1\}^m$. Let $p \stackrel{\text{def}}{=} \frac{|\{x\in\{0,1\}^n \,:\, \mathsf{Ext}(x;s)=y\}|}{2^n}$, so that for all $i\in[2^k]$ we have, $\mathbb{E}[B_{i,s,y}] = \Pr[B_{i,s,y}=1] = p$. First, let us consider the case that $|p-2^{-m}| \geq \frac{1}{64}\cdot 2^{-(k+m)/2}$. In the case, by Jensen's inequality and the linearity of expectation, we have:

$$\begin{aligned}
\mathbb{E}\left[\left| \sum_{i\in[2^k]} \left(B_{i,s,y} - 2^{-m}\right) \right| \right] &\geq \left| \left(\sum_{i\in[2^k]} \mathbb{E}[B_{i,s,y}] \right) - 2^{k-m} \right| \\
&\geq 2^k|p-2^{-m}| \geq \frac{1}{64}2^{(k-m)/2}
\end{aligned}$$

which matches the claim.

Therefore, we are left to consider the alternate case, where $|p-2^{-m}| < \frac{1}{64}2^{-(k+m)/2}$. In this case, we have the bounds:

$$\begin{aligned}
p &\leq 2^{-m} + 2^{-(k+m)/2} \leq 2^{-m} + 2^{-m-1} \leq \frac{3}{4} \\
2^k p &\geq 2^k(2^{-m} - 2^{-(k+m)/2}) \geq 2^k(2^{-m} - 2^{-m-1}) \geq 2^{k-m-1}
\end{aligned}$$

where the latter also implies $p \geq 2^{-k}$ since $k \geq m+1$. Let us define the random variables $V_i := (B_{i,s,y} - p)$. Then these variables are 4-wise independent, and for all $i\in[2^k]$ we have $\mathbb{E}[V_i]=0$ and:

$$\begin{aligned}
\mathbb{E}[V_i^2] &= p(1-p)^2 + p^2(1-p) = p(1-p) \in [p/4, p] \\
\mathbb{E}[V_i^4] &= p(1-p)^4 + p^4(1-p) \leq p.
\end{aligned}$$

Let use define $V := \sum_{i\in[2^k]} V_i$. Then, by applying Corollary 5.2 with $q = 2^k$, we have

$$\mathbb{E}[|V|] \geq \frac{1}{16}(2^k p)^{1/2} \geq \frac{1}{32} 2^{(k-m)/2}.$$

Finally, we have:

$$\begin{aligned}
\mathbb{E}\left[\left|\sum_{i\in[2^k]} \left(B_{i,s,y} - 2^{-m}\right)\right|\right] &= \mathbb{E}\left[\ \left|\sum_{i\in[2^k]} (B_{i,s,y} - p) + \sum_{i\in[2^k]} (p - 2^{-m})\right|\ \right] \\
&\geq \mathbb{E}[|V|] - 2^k|p - 2^{-m}| \geq \frac{1}{64} 2^{(k-m)/2}
\end{aligned}$$

which concludes the proof of the claim.

□

Using the above claim, we get a bound for the expected distinguishing advantage as:

$$\begin{aligned}
\mathbb{E}[\mathsf{Dist}(\mathcal{X})] &= \frac{1}{2^{d+k+1}} \sum_{s\in\{0,1\}^d, y\in\{0,1\}^m} \mathbb{E}\left[\left|\sum_{i\in[2^k]} \left(B_{i,s,y} - 2^{-m}\right)\right|\right] \\
&\geq \frac{1}{2}\cdot\frac{2^m}{2^k}\cdot\frac{1}{64}\cdot 2^{(k-m)/2} = \frac{1}{128} 2^{(m-k)/2} = 2^{(m-k)/2-7}.
\end{aligned}$$

This already shows the *expected* distinguishing advantage for $\mathcal{X}$ is sufficiently high. We now want to show that the distinguishing advantage is high with "good" probability:

$$\begin{aligned}
\Pr\left[\mathsf{Dist}(\mathcal{X}) \leq \frac{1}{2}\,\mathbb{E}[\mathsf{Dist}(\mathcal{X})]\right] &= \Pr\left[1 - \mathsf{Dist}(\mathcal{X}) \geq 1 - \frac{1}{2}\,\mathbb{E}[\mathsf{Dist}(\mathcal{X})]\right] \\
&\leq \frac{1 - \mathbb{E}[\mathsf{Dist}(\mathcal{X})]}{1 - \frac{1}{2}\,\mathbb{E}[\mathsf{Dist}(\mathcal{X})]} \leq 1 - \mathbb{E}[\mathsf{Dist}(\mathcal{X})]^2 \leq 1 - 2^{m-k-14}
\end{aligned}$$

where the second line follows by Markov inequality and the fact that $\mathsf{Dist}(\mathcal{X}) \in [0,1]$. Therefore, we get:

$$\Pr[\mathsf{Dist}(\mathcal{X}) > 2^{(m-k)/2-8}] > 2^{m-k-14}.$$

Next, we want to show that, if X is uniform over $\mathcal{X}$, then $\mathbf{H}_\infty(X) \geq k$ with overwhelming probability over the choice of $\mathcal{X}$. This happens as long as $X_1, \ldots, X_{2^k}$ are all distinct, which happens with probability $\geq 1 - \Pr[\exists i \neq j \text{ s.t. } X_i = X_j] \geq 1 - 2^{2k-n}$. Therefore, we get:

$$\Pr[(\mathsf{Dist}(\mathcal{X}) > 2^{(m-k)/2-8}) \quad\wedge\quad (\mathbf{H}_\infty(X) \geq k)] > 2^{(m-k)-14} - 2^{2k-n} > 0$$

as long as $n > 3k - m + 14$. This means that, as long as the above inequality is satisfied, there *exits* some choice of $\mathcal{X} = \{x_1, \ldots, x_{2^k}\}$ from the 4-wise independent family (e.g., some hash function $h \in \mathcal{H}$ with $x_i = h(i)$) such that, if X is uniform over $\mathcal{X}$, we have $\mathbf{H}_\infty(X) \geq k$, and $\mathsf{Dist}(X) \geq 2^{(m-k)/2-8}$. As long as the has function h is efficiently computable, we can sample from X efficiently in poly(n)-time. Therefore, this proves the first part of theorem.

For the second part of the theorem, we first generalize our bound on entropy by choosing the elements $X_1, \ldots, X_{2^k}$ of $\mathcal{X}$ via a $(t+1)$-wise independent distribution. In that case, we get $\mathbf{H}_\infty(X) \geq k - \log(t)$ as long as the multiplicity of any element of $\mathcal{X}$ is at most t, which happens with probability

$$\Pr[\mathbf{H}_\infty(X) \geq k - \log(t)] \geq 1 - \binom{2^k}{(t+1)} 2^{-tn} \geq 2^{(t+1)k-tn}.$$

Therefore, we get:

$$\Pr[(\mathsf{Dist}(\mathcal{X}) > 2^{(m-k)/2-8}) \quad \wedge \quad (\mathbf{H}_\infty(X) \geq k - \log(t))] > 2^{(m-k)-14} - 2^{(t+1)k-tn} > 0$$

as long as $n > k + 2k/t - (m-14)/t$.

Finally, to prove the second part of the theorem, for any $k' \geq m$, $n > k' + \log(k') + 11$ let us apply the generalized bound on $k = k' + \log(k') + 2$ and $t = 2k'$. Note, $k \geq m+2$ and $n > k + 2k/t - (m-14)/t$. Therefore, we get:

$$\Pr[(\mathsf{Dist}(\mathcal{X}) > 2^{(m-k'-\log(k'))/2-9}) \quad \wedge \quad (\mathbf{H}_\infty(X) \geq k')] > 0.$$

This proves the second part of the theorem, by noting that we can ($t = 2k'$)-wise independent hash functions can be efficiently computable in poly(n)-time.

6 Lower Bound: Square-Friendly Applications

In this section we prove Theorem 1.2. We define an application P for which we show that it is δ-square secure in Claim 6.1, but a single run can be broken with advantage $\Omega(\sqrt{\delta \cdot 2^{m-k}})$. The two claims imply Theorem 1.2.

We consider the following (artificial) indistinguishability application P between a distinguisher D and a challenger $C(r)$, which is initialized with a key $r \in \{0,1\}^m$ and a bit $b \in \{0,1\}$ (where $b = 0$ means we're playing the random, and $b = 1$ the real game.)

- $C(r)$ flips a biased coin α where $\Pr[\alpha = 1] = \sqrt{\delta}$.
- If $\alpha = 0$, $C(r)$ sends $\perp$ to D.
- If $\alpha = 1$ and $b = 1$ then $C(r)$ sends r to D.
- If $\alpha = 1$ and $b = 0$ then $C(r)$ samples a random $r' \leftarrow \{0,1\}^m$ and sends r' to D.
- D outputs its guess b'.

Let $f_D(r)$ denote the advantage of D (over the choice of b) in the above game

$$f_D(r) = \Pr_{b \leftarrow \{0,1\}}[b = b'] - 1/2$$

By the following claim P is $\delta/4$-square secure (against computationally unbounded distinguishers and any distribution of keys).

Claim 6.1. *For any D and any possible key r, $|f_D(r)| \leq \sqrt{\delta}/2$ (and thus also $\mathbb{E}[f_D(U_m)^2] \leq \delta/4$)*

Proof.

$$\begin{aligned} |\Pr[b = b'] - 1/2| &= |\underbrace{\Pr[\alpha = 1]}_{\sqrt{\delta}} \Pr[b = b'|\alpha = 1] + \underbrace{\Pr[\alpha = 0]}_{1-\sqrt{\delta}} \underbrace{\Pr[b = b'|\alpha = 0]}_{1/2} - 1/2| && (9) \\ &\leq |\sqrt{\delta}\Pr[b = b'|\alpha = 1] - \sqrt{\delta}/2| \leq \sqrt{\delta}/2 && (10) \end{aligned}$$

In the last step we used that any probability is between 0 and 1, in the second step we used that conditioned on $\alpha = 0$, D gets no information about the uniformly random bit b and thus any guess b' will be equal to b with probability exactly 1/2. ☐

Claim 6.2. *For any family $\mathcal{H} = \{h_s\}$ of functions $\{0,1\}^n \to \{0,1\}^m$, there exists an (even efficiently samplable) (n,k)-source X and a (generally inefficient) distinguisher $D(.)$ such that*

$$\mathop{\mathbb{E}}_S[f_{D(S)}(h_S(X))] = \Omega(\sqrt{\delta \cdot 2^{m-k}})$$

Proof. By Theorem 1.1 there exists an efficiently samplable X such that, for a random S, the statistical distance of the derived key $h_S(X)$ from uniform is

$$\Delta((U_m, S)\ ,\ (h_S(X), S)) = \Omega(\sqrt{2^{m-k}})$$

And thus, we can define a (potentially inefficient) distinguisher $D(S)$ that can distinguish $(h_S(X), S)$ from (U_m, S) with advantage $\Omega(\sqrt{2^{m-k}})$, and thus guess b with the same advantage whenever he gets to see the key (i.e. $\alpha = 1$).

Concretely, the distinguisher $D(s)$ is defined as follows. If it receives $\perp$ (i.e. $\alpha = 0$), it simply outputs a random guess $b' \leftarrow \{0,1\}$. If it receives some $r \in \{0,1\}^m$ (i.e. $\alpha = 1$), then it outputs 1 if $\Pr_X(h_s(X) = r) \geq 2^{-m}$ and 0 otherwise. We have

$$\begin{aligned}
\mathop{\mathbb{E}}_S[f_{D(S)}(h_S(X))] &= \underbrace{\Pr[\alpha = 0]}_{1-\sqrt{\delta}}\underbrace{\Pr[b = b'|\alpha = 0]}_{1/2} + \underbrace{\Pr[\alpha = 1]}_{\sqrt{\delta}}\underbrace{\Pr[b = b'|\alpha = 1]}_{1/2+\Omega(\sqrt{2^{m-k}})} - 1/2 \\
&= \Omega(\sqrt{\delta \cdot 2^{m-k}})
\end{aligned}$$

□

References

[AGHP92] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple construction of almost k-wise independent random variables. *Random Struct. Algorithms*, 3(3):289–304, 1992.

[BDK+05] Xavier Boyen, Yevgeniy Dodis, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Secure remote authentication using biometric data. In Ronald Cramer, editor, *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 147–163. Springer-Verlag, 2005.

[BDK+11] Boaz Barak, Yevgeniy Dodis, Hugo Krawczyk, Olivier Pereira, Krzysztof Pietrzak, François-Xavier Standaert, and Yu Yu. Leftover hash lemma, revisited. In Phillip Rogaway, editor, *CRYPTO*, LNCS, pages 1–20. Springer, 2011.

[Ber91] Bonnie Berger. The fourth moment method. In Alok Aggarwal, editor, *SODA*, pages 373–383. ACM/SIAM, 1991.

[BH05] Boaz Barak and Shai Halevi. A model and architecture for pseudo-random generation with applications to /dev/random. In *Proceedings of the 12th ACM Conference on Computer and Communication Security*, pages 203–212, 2005.

[BR94] M. Bellare and J. Rompel. Randomness-efficient oblivious sampling. In *35th Annual Symposium on Foundations of Computer Science*, pages 276–287. IEEE, 1994.

[BST03] Boaz Barak, Ronen Shaltiel, and Eran Tromer. True random number generators secure in a changing environment. In *Proceedings of the 5th Cryptographic Hardware and Embedded Systems*, pages 166–180, 2003.

[CDH+00] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In Bart Preneel, editor, *Advances in Cryptology—EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 453–469. Springer-Verlag, 2000.

[CG89] Benny Chor and Oded Goldreich. On the power of two-point based sampling. *Journal of Complexity*, 5:96–106, 1989.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pages 209–218, Dallas, Texas, 23–26 May 1998.

[CRSW11] L. Elisa Celis, Omer Reingold, Gil Segev, and Udi Wieder. Balls and bins: Smaller hash families and faster evaluation. In Rafail Ostrovsky, editor, *FOCS*, pages 599–608. IEEE, 2011.

[DGH+04] Yevgeniy Dodis, Rosario Gennaro, Johan Håstad, Hugo Krawczyk, and Tal Rabin. Randomness extraction and key derivation using the cbc, cascade and hmac modes. In Matt Franklin, editor, *Advances in Cryptology—CRYPTO 2004*, volume 3152 of *LNCS*, pages 494–510. Springer-Verlag, 15–19 August 2004.

[DGKM12] Dana Dachman-Soled, Rosario Gennaro, Hugo Krawczyk, and Tal Malkin. Computational extractors and pseudorandomness. In Ronald Cramer, editor, *TCC*, volume 7194 of *Lecture Notes in Computer Science*, pages 383–403. Springer, 2012.

[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.

[DRV12] Yevgeniy Dodis, Thomas Ristenpart, and Salil P. Vadhan. Randomness condensers for efficiently samplable, seed-dependent sources. In *9th Theory of Cryptography Conference*, pages 618–635, 2012.

[DTT10] Anindya De, Luca Trevisan, and Madhur Tulsiani. Time space tradeoffs for attacks against one-way functions and prgs. In *CRYPTO*, pages 649–665, 2010.

[DY13] Yevgeniy Dodis and Yu Yu. Overcoming weak expectations. In *TCC*, pages 1–22, 2013.

[GKR04] Rosario Gennaro, Hugo Krawczyk, and Tal Rabin. Secure hashed diffie-hellman over non-ddh groups. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 361–381. Springer-Verlag, 2004.

[HILL99] J. Håstad, R. Impagliazzo, L.A. Levin, and M. Luby. Construction of pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[Kra10] Hugo Krawczyk. Cryptographic Extraction and Key Derivation: The HKDF Scheme. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *LNCS*, pages 631–648. Springer-Verlag, 2010.

[KZ03] Jess Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. In *44th Annual Symposium on Foundations of Computer Science*, pages 92–101, Cambridge, Massachusetts, October 2003. IEEE.

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[NN93] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–53, 1996.

[RR99] Ran Raz and Omer Reingold. On recycling the randomness of states in space bounded computation. In *Proceedings of the 31st ACM Symposium on the Theory of Computing*, pages 159–168, 1999.

[RSW06] Omer Reingold, Ronen Shaltiel, and Avi Wigderson. Extracting randomness via repeated condensing. *SIAM J. Comput.*, 35(5):1185–1209, 2006.

[RTS00] Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Computing*, 13(1):2–24, 2000.

[Sie89] Alan Siegel. On universal classes of fast high performance hash functions, their time-space tradeoff, and their applications (extended abstract). In *FOCS*, pages 20–25, 1989.

[TV00] Luca Trevisan and Salil Vadhan. Extracting randomness from samplable distributions. In *41st Annual Symposium on Foundations of Computer Science*, pages 32–42, Redondo Beach, California, November 2000. IEEE.