

Multi-Source Randomness Extraction and Generation in the Random-Oracle Model

Sandro Coretti  

IOG, Paris, France



Pooya Farshim  

IOG, Zürich, Switzerland

Durham University, UK

Patrick Harasser  

Technische Universität Darmstadt, Germany

Karl Southern  

Durham University, UK

Abstract

We study the multi-source randomness extraction and generation properties of the monolithic random oracle (RO), whereby one is tasked with extracting or generating uniform random bits from multiple unpredictable sources. We formalize this problem according to the query complexities of the involved parties – sources, distinguishers, and predictors, where the latter are used to define unpredictability.

We show both positive and negative results. On the negative side, we rule out definitions where the predictor is not at least as powerful as the source or the distinguisher. On the positive side, we show that the RO is a multi-source extractor when the query complexity of the distinguisher is bounded. Our main positive result in this setting is with respect to arbitrary unpredictable sources, which we establish via a combination of a compression argument (Dodis, Guo, and Katz, EUROCRYPT’17) and the decomposition of high min-entropy sources into flat sources.

Our work opens up a rich set of problems, ranging from statistical multi-source extraction with respect to unbounded distinguishers to novel decomposition techniques (Unruh, CRYPTO’07; Coretti et al., EUROCRYPT’18) and multi-source extraction for non-monolithic constructions.

2012 ACM Subject Classification Security and privacy → Cryptography

Keywords and phrases Multi-source randomness extraction, Multi-source randomness generation, Compression argument, Convex decomposition

Digital Object Identifier 10.4230/LIPIcs.ITC.2025.10

Related Version *Full Version:* <https://eprint.iacr.org/2025/1258> [12]

Funding *Pooya Farshim:* Supported in part by EPSRC grant EP/V034065/1.

Patrick Harasser: Funded by the Deutsche Forschungsgemeinschaft (DFG) – SFB 1119 – 236615297.

Acknowledgements We thank Balthazar Bauer for many fruitful discussions and for participating in the early stages of this work. We also thank the anonymous reviewers who helped improve the presentation of our results.

1 Introduction

1.1 Overview

Randomness Extraction. High-quality, close-to-uniform randomness is indispensable in cryptography for many applications (encryption, authentication, zero-knowledge proofs, and multi-party computation, to name but a few). In fact, there are a number of cryptographic tasks that simply cannot be achieved without perfect randomness [18, 21, 28]. However, while “nature” provides many sources that contain some amounts of entropy (e.g., those



© Sandro Coretti, Pooya Farshim, Patrick Harasser, and Karl Southern;
licensed under Creative Commons License CC-BY 4.0

6th Conference on Information-Theoretic Cryptography (ITC 2025).

Editor: Niv Gilboa; Article No. 10; pp. 10:1–10:23



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

based on keystrokes, mouse movements, the timing of interrupts, or passwords), sources that output perfectly uniform random bits are practically non-existent. A fundamental task is therefore so-called *randomness extraction*, which involves obtaining uniform randomness from high-entropy sources. One of the key principles in randomness extraction is that no assumption is made about the sources, except that they have high min-entropy; doing otherwise would entail the need for a very deep understanding of the sources, which may be impossible in practice.

An important hurdle to randomness extraction is that deterministic extractors do not exist [9], i.e., there is no function f whose output is close to uniform on *all* high-entropy sources. A simple counterexample is the so-called “extractor-fixing” attack, where one considers the sources \mathcal{S}_0 and \mathcal{S}_1 that sample an input uniformly among those on which the output of f starts with a zero and a one, respectively. At least one of these sources, say, \mathcal{S}_0 , has close to maximal min-entropy, but the output $f(\mathcal{S}_0)$ is certainly not uniformly random (it always starts with 0).

One way to circumvent this issue involves seeded extractors [29], where f receives an additional uniformly random value s – the seed – as input, and the requirement is that for all *seed-independent* high-entropy sources \mathcal{S} , $(f(s, \mathcal{S}), s)$ be close to uniform. Such seeded extractors may be useful for certain applications (e.g., privacy amplification), but in the context of randomness generation, their drawback is that they need a uniform seed in the first place. Moreover, even if one was somehow able to produce a “once-and-for-all” random seed, it is unrealistic to assume that sources would not depend on this (publicly known) value.

For these reasons, one often uses cryptographic hash functions (CHFs) instead of seeded extractors in practice. However, the only CHF-based extractor constructions known in the standard model (achieving in fact only the weaker notion of randomness condensing) require non-standard assumptions [19]. Therefore, an approach commonly taken is to analyze constructions in an idealized model, where the extractor or parts of it are modeled as an idealized primitive. Instances of this approach include:

1. Modeling the entire extractor as a monolithic random oracle;
2. Merkle–Damgård-based constructions where the compression function is treated as a random function;
3. The Davies–Meyer compression function with the block cipher modeled as an ideal cipher;
4. Sponge-based constructions with the round function treated as a random permutation.

The analysis in an idealized model can serve as heuristic justification for the use of these constructions in practice. Another benefit of the CHF-based approach in idealized models is that the final constructions are often very efficient.

Extraction in Idealized Models. When studying extraction in idealized models, there are two possibilities. One is to consider only oracle-independent sources (as done, e.g., in [15, 7, 32, 35]). This approach provides limited realism, since the oracle abstracts a CHF, and it is unreasonable to assume that sources are completely independent of it. For example, the timing of an interrupt may well depend on computations related to the hash function in use. Furthermore, if one does not allow oracle-dependent sources, results in the idealized model do not match those in the standard model: For example, the extractor-fixing attack can in general not be performed without oracle access, as the idealized primitive essentially acts as a (very large) seed.

The other option is to consider oracle-dependent sources. However, here the extractor-fixing attack reappears, and at a first glance, the only solution seems to be using a seed (as done, e.g., in [24, 20]). Fortunately, there are other alternatives: Work by Coretti et al. [11]

weakens the notion of oracle-dependence by requiring that the output of a source have entropy even given the queries the source made to the idealized primitive and the corresponding answers. This essentially amounts to modeling that “natural” sources occurring in practice add extra entropy on top of the CHF evaluations occurring around them.

Independent Sources of Entropy. Another way of avoiding a seed is to consider split entropy sources, i.e., sources that produce independent values satisfying certain min-entropy conditions. The study of such “multi-source” extractors began with the work of Chor and Goldreich [9], who introduced the problem of extracting randomness from two or more independent weak sources. Since then, many works have improved the entropy requirements, output length, and error parameters of extractors for both the two-source and the multi-source settings [2, 14, 27, 8]. These developments have led to increasingly efficient constructions, with broad applications in derandomization, cryptography, and complexity theory.

When studying the practicality of such extractors, it is pertinent to first examine whether sources of entropy that can be assumed to be independent actually exist. The answer appears to be affirmative when considering sources used in practice: For instance, Intel Secure Key, a hardware-based random number generator integrated into most modern Intel processors, uses thermal noise as entropy source [26]. Another source of entropy is sensor data measuring different physical properties, such as temperature and light intensity. Yet other sources use user-driven events such as keystrokes, mouse movements, and touchscreen interactions. All these measurements are influenced by distinct physical processes and environmental factors, and can therefore reasonably be considered independent.

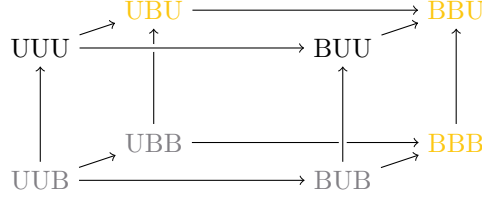
In spite of the theoretical advancements mentioned above, for practically relevant constructions of extractors one still resorts to CHFs and, to avoid non-standard assumptions, to idealized models. As a result, in this work we initiate the study of CHF-based multi-source extractors by providing suitable ideal-model definitions and by analyzing, as a first step, the security of the random oracle itself as a multi-source extractor wrt. several definitional variants. Investigating the security of non-monolithic constructions (e.g., Merkle–Damgård or Sponges) is left as an exciting open problem.

1.2 Our Contributions

Multi-Source Extractors in Idealized Models. A first contribution of this work is to put forth a definitional framework that captures extraction in idealized models, allowing the source to depend on the idealized primitive. When defining oracle-dependent extraction in an idealized model, the following definitional choices are possible:

1. The source can either fully depend on the ideal primitive (IP) or be computationally restricted by limiting the number of queries it makes to the IP;
2. To measure the entropy (or unpredictability) of the source, one can require unpredictability conditioned on the entire IP or only with respect to a predictor with bounded-query access to the IP;
3. For the distinguisher trying to tell the extractor output from a uniform string, one can either consider an unbounded party that gets the entire oracle as input or one that is only allowed a limited number of queries to the IP.

These choices result in $2^3 = 8$ possible notions, captured by a simple “source-distinguisher-predictor” (*SDP*) notation, where each entity can be “bounded” (B) or “unbounded” (U) depending on the type of access the corresponding party has to the IP (see Figure 1). We establish a complete set of generic implications between these notions.



■ **Figure 1** Multi-source extractor notions in the random-oracle model. Notions are labeled via an “ SDP ” string (for *Source*, *Distinguisher*, and *Predictor*), with $S, D, P \in \{B, U\}$ depending on whether the corresponding party has bounded-query (B) or unbounded (U) access to the random oracle. An arrow from X to Y means that any construction for X is also a construction for Y . Corners typeset in gray are shown not to be achieved by the monolithic construction via attacks presented in Section 4, specifically Proposition 17 (UBB) and Proposition 18 (BUB). Corners typeset in yellow are achieved in Sections 5 and 6, specifically Theorem 19 (BBB) and Theorem 23 (UBU).

Our definitions are captured by two games: In the first game, the source interacts with the IP to produce values x and z , where x is the input to the extractor and z is some leakage about x . A predictor, also with access to the IP, is then given z and tries to predict x . The second game is the extraction game, in which $\ell \geq 1$ sources interact with the oracle to produce otherwise independently sampled (x_i, z_i) , and a distinguisher attempts to tell apart the extractor output on the x_i from a uniform string, given the leakage strings z_i .

Note that this definitional framework also captures as a special case the entropy notion from [11], in which there is a single bounded source, and the predictor as well as the distinguisher are bounded. This is done by considering canonical bounded sources, which output the query/answer list in the leakage.

Results. We then go on to investigate whether the monolithic random oracle is a good multi-source extractor. Specifically, we make the following findings:

1. We first establish a positive result for one of the easiest cases, BBB (see Theorem 19). This result also implies the case BBU, as making the predictor unbounded reduces the class of sources.
2. We then turn to the main focus of this work, involving unbounded sources. This is a crucial setting, as it minimizes the assumptions on the source.
 - a. We first observe that the random oracle does not work as an extractor when the predictor is bounded (i.e., the UBB and UUB configurations). This is due to the fact that sources with unbounded oracle access can return a fixed point in the RO. While such sources are unpredictable for a bounded-query predictor, a bounded distinguisher can easily check if the given input is fixed under the RO (see Proposition 17).
 - b. The main result of our paper is to show that the monolithic random oracle is an effective extractor in the UBU setting (see Theorem 23), i.e., with unbounded source and predictor, but with a bounded distinguisher. This is an important case as, in practice, one can reasonably assume distinguishers to be bounded.
3. Finally, we show an attack against BUB, where sources can leak some information about their output to the distinguisher, which can use its unbounded oracle access to check if the challenge it is given has a preimage compatible with the leakage (see Proposition 18).

Finding efficient constructions when the distinguisher is unbounded (i.e., for UUU and BUU) is left for future work.

Comparison to Coretti et al. [11]. Comparing our main result to the analogous one in [11], we observe that the two statements are formally incomparable. Indeed, our notion assumes at least two *independent, unbounded* sources, while their setting assumes a *single, bounded* source in the BBU setting and that the source leaks its random oracle queries.

Techniques. In our main result, the source can depend arbitrarily on the random oracle. This dependency allows the source to create arbitrary correlations in the distribution of the oracle, conditioned on the source’s output and potential leakage. One way of dealing with such correlations is to use the compression technique of Gennaro and Trevisan [25].

Our proof strategy follows that of Dodis et al. [16], which in turn is based on De et al. [13]. We begin by using a compression argument to establish that a random oracle, when applied to inputs $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_\ell)$ produced by independent sources $\mathcal{S}_1(H), \dots, \mathcal{S}_\ell(H)$, satisfies the property of unrecoverability: It is computationally infeasible to recover \mathbf{x} from $H(\mathbf{x})$, even given the leakage of the (unpredictable) sources. The proof then transitions to reducing distinguishing $H(\mathbf{x})$ from a random output to a next-bit guessing problem. The difficulty of next-bit guessing is in turn established via reduction to the unrecoverability property and via an incompressibility argument that uses a next-bit guesser to compress oracles.

Throughout these steps, it is crucial to account for the fact that the inputs \mathbf{x}_i are not uniformly random and independent of the oracle, as is the case in [16]. Instead, they can fully depend on the oracle and are only guaranteed to have entropy conditioned on the oracle. To handle this obstacle, we decompose high-entropy sources into convex combinations of flat sources [34] at the necessary steps in the proof.

1.3 Related Work

There are numerous other models in which one can study randomness extraction. Some examples include:

- Dodis et al. [22], who investigate extracting randomness from extractor-dependent sources that are capable of making black-box queries to the seeded extractor;
- The study of Santha–Vazirani (SV) sources, which consist of sequences of random bits where an adversary has partial control over the conditional distribution of each bit given the previous bits – extraction from such sources is known to be impossible [31], but more recent work has shown that extraction is feasible for non-binary SV sources [3];
- Aggarwal et al. [1], who consider a weaker type of sources called SHELA sources, which do not allow for uniform extraction but for a weaker notion of “somewhere extraction” that is sufficient for some applications.

The main difference between our work and the majority of existing results is that they focus on providing explicit constructions in the standard model. For instance, the work of Dodis and Oliveira [17] uses the probabilistic method to show the existence of so-called super-strong extractors (and then provides explicit constructions thereof).¹ Note the qualitative difference between proving the existence of a not necessarily efficient extractor in the standard model via the probabilistic method versus our treatment in the random-oracle model, where the focus is on validating cryptographic constructions of extractors in an idealized model, with the monolithic extractor being the first object of study. We further comment on the differences between these results and our work in the relevant sections of the paper.

¹ In our terminology, super-strong extractors are two-source extractors that work even if one of the two inputs is made public.

1.4 Future Directions

Our work shows that compression is sufficiently powerful to establish multi-source extraction and generation bounds for the monolithic random oracle. However, compression is not straightforward to apply when the idealized object is not fully random, as is the case for many non-monolithic constructions such as Merkle–Damgård or Sponge.² This raises the question of whether other techniques can be extended to work in the multi-source setting.

The decomposition techniques of Coretti et al. (CDGS) [10] and Unruh [33] are alternative techniques that can lead to simpler proofs, and can therefore be potentially applied to non-monolithic constructions in the multi-source setting. In CDGS, for a parameter $P \in \mathbb{N}$, one decomposes the random oracle conditioned on some RO-dependent leakage z into a convex combination of sources, each of which is fixed on at most P points and dense otherwise (plus some small error distribution). CDGS use such a decomposition to show that the RO model with auxiliary information (AI-ROM) is indistinguishable from a P -bit-fixed (BF) RO model, where some inputs to the RO are fixed to values computed during decomposition on leakage z , and the other values are uniformly random. The statistical difference between the two settings is $\mathcal{O}(ST/P)$, where S is an upper bound on the size of the leakage z , T is the query complexity of the distinguisher, and P is the number of fixed points. In applications, P cannot be chosen too large, as then the probability of certain bad events (e.g., a challenge point lying in the set of fixed points) would be too large.

Unruh decomposition is conceptually simpler in that it decomposes the original random oracle and the set of fixed points is compatible with the given random oracle. The distinguishing advantage, however, is now $\mathcal{O}(\sqrt{ST/P})$.

A natural question is whether decomposition techniques like CDGS or Unruh can be used in the multi-source setting. This turns out not to be the case, at least when these techniques are applied directly in *parallel* to fix points for each source separately, and then fixing the points in the union of these sets.³ Indeed, consider sources \mathcal{S}_1 and \mathcal{S}_2 that leak $z_1 := \sum_{1 \neq x \in [N]} H(x)$ and $z_2 := \sum_{x \in [N]} H(x)$. Clearly, given (z_1, z_2) , the value $H(1)$ is revealed. That is, an AI-ROM adversary can determine $H(1)$ given leakage (z_1, z_2) . On the other hand, z_1 and z_2 *individually* leak very little information about the hash of any particular point: $H(1)$ is random and independent of the hashes of all other points. This means that running the CDGS decomposition in parallel on z_1 and z_2 will not fix $H(1)$. Consequently, a BF-ROM adversary will not be able to predict $H(1)$.⁴

Despite these attacks, we conjecture that parallel decomposition *is* possible under appropriate restrictions. Such parallel decomposition techniques would be helpful in studying the multi-source extraction properties of non-monolithic constructions. We leave the development of these techniques to future work.

1.5 Structure of the Paper

We start by recalling the preliminaries, including the compression lemmas, in Section 2. In Section 3, we formally define multi-source extractors with respect to different source classes, unpredictability notions, and indistinguishability metrics. In Section 4, we rule out

² Note that such a result does not follow from the indifferentiability of these constructions, as the multi-source extractor game, due to the unpredictability requirement on the sources, is not single-stage [30].

³ Another approach would be to combine the sources and apply decomposition to the resulting source. Multi-source extraction, however, does not hold with respect to a single source, and accordingly it is unclear how one would upper-bound the probability of sampled points lying in the set of fixed points.

⁴ We note that this is an attack on directly applying decomposition as a proof technique. Moreover, z_1 and z_2 can be made unpredictable by simply appending random bits.

the notions where the predictor is not as powerful as either the source or the distinguisher. Section 5 contains our security result for distinguishers and sources with a bounded number of queries. In Section 6, we present our main contribution on the multi-source randomness extraction with respect to arbitrary unpredictable sources. This leaves two notions, where both the distinguisher and predictor are unbounded, open for future work.

2 Preliminaries

Basic Notation. We denote by \mathbb{N} and \mathbb{R} the sets of natural and of real numbers, and for $n \in \mathbb{N}$ we define $[n] := \{1, \dots, n\}$. The cardinality of a set S is denoted $|S|$, and we let S^* be the set of finite-length tuples over S . As usual, the empty tuple is denoted ε . If \mathcal{X} is a random variable, then sampling from \mathcal{X} is denoted $x \leftarrow \mathcal{X}$; when \mathcal{X} is uniformly distributed over a finite set X , we write $x \leftarrow X$ instead. The support of a random variable \mathcal{X} is denoted $\text{Supp}(\mathcal{X})$. For $\ell, M, N \in \mathbb{N}$, we write $\text{Fun}(N^\ell, M)$ for the set of functions from $[N]^\ell$ to $[M]$. We use boldface notation for vectors, as in $\mathbf{x} = (x_1, \dots, x_\ell)$, and we write $|\mathbf{x}|$ for the length of the vector \mathbf{x} . If not stated otherwise, indexes in vectors and strings start from 1. All logarithms are to base 2.

ROM [23, 5]. Let $\ell, M, N \in \mathbb{N}$. We define the random-oracle model $\mathcal{RO}[\ell, N, M]$ as the uniform distribution on $\text{Fun}(N^\ell, M)$. When operating in this model, all algorithms (both honest and adversarial) can query a function $H \in \text{Fun}(N^\ell, M)$ as an oracle. In security games, one first samples H uniformly from $\text{Fun}(N^\ell, M)$. Afterwards, the game is run with all parties having oracle access to H . For the remainder of this paper, we will always work in this setting.

Fundamental Lemma of Game Playing [6]. We use the code-based game-playing framework of Bellare and Rogaway. Let G_1 and G_2 be two games whose code is identical except for the consequent inside one if-branch, and let **Bad** be the event that the if-statement is triggered. Then $|\Pr[G_1] - \Pr[G_2]| \leq \Pr[\text{Bad}]$.

Monolithic Construction. Let $\ell, M, N \in \mathbb{N}$. The monolithic construction is the algorithm $\text{Mono} := \text{Mono}[\ell, N, M]$ which takes a vector $\mathbf{x} \in [N]^\ell$ as input, has oracle access to $H \in \text{Fun}(N^\ell, M)$, and returns $y := H(\mathbf{x}) \in [M]$.

Source Codes. Let S be a set. A (deterministic) code for S is a pair (Enc, Dec) , where $\text{Enc}: S \rightarrow \{0, 1\}^l$ and $\text{Dec}: \{0, 1\}^l \rightarrow S$ are functions, and $l \in \mathbb{N}$. A randomized encoding for S is a pair (Enc, Dec) , where $\text{Enc}: S \times R \rightarrow \{0, 1\}^l$ and $\text{Dec}: \{0, 1\}^l \times R \rightarrow S$ are functions, R is a set, and $l \in \mathbb{N}$.

► **Lemma 1** (Deterministic compression [13, 16]). *Let S be a set, and let (Enc, Dec) be a code for S such that, for every $m \in S$, $\text{Dec}(\text{Enc}(m)) = m$. Then*

$$\mathbb{E}_{m \leftarrow S}[|\text{Enc}(m)|] \geq \log |S|.$$

► **Lemma 2** (Randomized compression [13, 16]). *Let S be a set, $\delta \in \mathbb{R}_{>0}$, and let (Enc, Dec) be a randomized encoding for S with recovery probability δ , meaning that for every $m \in S$,*

$$\Pr_{r \leftarrow R}[\text{Dec}(\text{Enc}(m, r), r) = m] \geq \delta.$$

Then $l \geq \log |S| - \log 1/\delta$.

Flat Sources. Let X be a random variable over a finite set S and $k \in \mathbb{R}_{>0}$.

1. We call X a k -source if, for every $x \in S$, $\Pr[X = x] \leq 2^{-k}$.
2. We say that X is flat if it is uniformly distributed over a subset $T \subseteq S$.
3. Assume that $2^k \in \mathbb{N}$. We call X a flat k -source if it is uniformly distributed over a subset $T \subseteq S$ with $|T| = 2^k$.

► **Lemma 3** (Flat decomposition of sources [34]). *Let $k \in \mathbb{R}_{>0}$ with $2^k \in \mathbb{N}$. Then every k -source X is a convex combination of flat k -sources, i.e., $X = \sum p_i X_i$ with $0 \leq p_i \leq 1$ for all i , $\sum p_i = 1$, and all the X_i are flat k -sources.*

Numeric Inequalities. We conclude by recalling three inequalities for real numbers and their powers that we will use in this work.

► **Lemma 4** (Numeric inequalities). *Let $e \in \mathbb{R}$ denote Euler's number. Then:*

1. *For all $x \in \mathbb{R}$ with $x \geq 1$, it holds that $(1 - 1/x)^x \leq e^{-1}$.*
2. *For all $x \in \mathbb{R}$ with $x \geq -1$, and all $r \in \mathbb{N}$, it holds that $(1 + x)^r \geq 1 + rx$.*
3. *For all $k, n \in \mathbb{N}$ with $0 < k \leq n$, it holds that $(n/k)^k \leq \binom{n}{k} \leq (en/k)^k$.*

3 Multi-Source Extractors

In this section, we formally introduce multi-source extractors (MSEs) in the random-oracle model. We first formalize unpredictability (Pred) of sources, the property that forms the underlying assumption in all results that we prove. We then go on to defining unrecoverability (UR) and MSE for Mono. Our notions are parameterized by the query complexity of the various parties in the Pred, UR, and MSE games, and by the advantages they achieve in these games. We present our definitions for the monolithic construction Mono, but generalizations to other constructions and idealized models are straightforward. Finally, we prove implications between these notions and between different parameter sets for the same notion.

► **Definition 5** (Sources, Adversaries, Distinguishers, and Predictors). *Let $\ell, M, N \in \mathbb{N}$.*

1. *A source is an algorithm \mathcal{S} that takes no input and returns a pair $(x, z) \in [N] \times \{0, 1\}^*$.*
2. *An adversary is an algorithm \mathcal{A} which, on input $y \in [M]$ and a vector $\mathbf{z} \in (\{0, 1\}^*)^\ell$ of strings, returns $\mathbf{x}' \in [N]^\ell$.*
3. *A distinguisher is an algorithm \mathcal{D} which, on input $y \in [M]$ and a vector $\mathbf{z} \in (\{0, 1\}^*)^\ell$, returns a bit.*
4. *A predictor is an algorithm \mathcal{P} which, on input $z \in \{0, 1\}^*$, returns $x' \in [N]$.*

All these algorithms are probabilistic and unbounded, and additionally receive oracle access to a function $H \in \text{Fun}(N^\ell, M)$.

► **Definition 6** (Unpredictability). *Let $\ell, M, N \in \mathbb{N}$, \mathcal{S} be a source, and \mathcal{P} a predictor. We define the advantage of \mathcal{P} in the prediction game for \mathcal{S} as $\text{Adv}_{\mathcal{S}, \mathcal{P}}^{\text{pred}} := \Pr[\text{Pred}_{\mathcal{S}}^{\mathcal{P}}]$, where the game $\text{Pred}_{\mathcal{S}}^{\mathcal{P}}$ is given in Figure 2. For $q \in \mathbb{N}$ and $\delta \in \mathbb{R}_{\geq 0}$, we say that \mathcal{S} is (q, δ) -unpredictable if for every predictor \mathcal{P} making at most q oracle calls, we have $\text{Adv}_{\mathcal{S}, \mathcal{P}}^{\text{pred}} \leq \delta$.*

► **Remark 7.** The notion of unpredictability that we consider represents the first main point of difference with several prior works. For example, the class of sources that we consider is *larger* than the one in the work of Dodis and Oliveira [17], who essentially require high min-entropy for every oracle. We on the other hand require unpredictability to hold only on average over the choice of the RO. This approach has been used in other prior works, e.g., [11, 4]. In particular, the class of sources that we consider contains the one of Coretti et al. [11] as a subclass, by setting the leakage z to be the query-answer pairs of the source.

Game $\text{Pred}_S^{\mathcal{P}}$: $H \leftarrow \text{Fun}(N^\ell, M)$ $(x, z) \leftarrow \mathcal{S}^H$ $x' \leftarrow \mathcal{P}^H(z)$ return $(x = x')$	Game $\text{UR}^{\mathcal{S}, \mathcal{A}}$: $H \leftarrow \text{Fun}(N^\ell, M)$ for $i = 1$ to ℓ do $(x_i, z_i) \leftarrow \mathcal{S}_i^H$ $y \leftarrow H(x)$; $x' \leftarrow \mathcal{A}^H(y, z)$ return $(x = x')$	Game $\text{MSE}^{\mathcal{S}, \mathcal{D}}$: $H \leftarrow \text{Fun}(N^\ell, M)$; $b \leftarrow \{0, 1\}$ for $i = 1$ to ℓ do $(x_i, z_i) \leftarrow \mathcal{S}_i^H$ $y_0 \leftarrow [M]$; $y_1 \leftarrow H(x)$ $b' \leftarrow \mathcal{D}^H(y_b, z)$; return $(b = b')$
--	--	--

■ **Figure 2** The prediction game for a source \mathcal{S} , and the unrecoverability and multi-source-extractor games for the monolithic construction Mono.

► **Remark 8.** The class of sources that we consider can leak information to the distinguisher or predictor via z . One benefit of this definitional choice is that it allows modeling of auxiliary input via *preprocessing*. Indeed, we can assume without loss of generality that such auxiliary information is computed deterministically, and thus every source can recompute it and include it as part of its leakage. Unpredictability is preserved under this transformation as long as predictors are unbounded and can also recompute the auxiliary information.

► **Definition 9 (Unrecoverability).** Let $\ell, M, N \in \mathbb{N}$, $\mathcal{S} = (\mathcal{S}_1, \dots, \mathcal{S}_\ell)$ be a tuple of sources, and \mathcal{A} an adversary. We define the advantage of $(\mathcal{S}, \mathcal{A})$ in the unrecoverability game for Mono as $\text{Adv}_{\mathcal{S}, \mathcal{A}}^{\text{ur}} := \Pr[\text{UR}^{\mathcal{S}, \mathcal{A}}]$, where the game $\text{UR}^{\mathcal{S}, \mathcal{A}}$ is given in Figure 2. For $q_{\mathcal{S}}, q_{\mathcal{A}}, q_{\mathcal{P}} \in \mathbb{N}$ and $\delta, \epsilon \in \mathbb{R}_{\geq 0}$, we say that Mono is $(q_{\mathcal{P}}, \delta, q_{\mathcal{S}}, q_{\mathcal{A}}, \epsilon)$ -unrecoverable if for every $(\mathcal{S}, \mathcal{A})$ as above such that, for every $i \in [\ell]$, \mathcal{S}_i is $(q_{\mathcal{P}}, \delta)$ -unpredictable and makes at most $q_{\mathcal{S}}$ oracle calls, and such that \mathcal{A} makes at most $q_{\mathcal{A}}$ oracle calls, we have $\text{Adv}_{\mathcal{S}, \mathcal{A}}^{\text{ur}} \leq \epsilon$.

► **Definition 10 (Multi-Source Extraction).** Let $\ell, M, N \in \mathbb{N}$, $\mathcal{S} = (\mathcal{S}_1, \dots, \mathcal{S}_\ell)$ be a tuple of sources, and \mathcal{D} a distinguisher. We define the advantage of $(\mathcal{S}, \mathcal{D})$ in the multi-source-extraction game for Mono as $\text{Adv}_{\mathcal{S}, \mathcal{D}}^{\text{mse}} := 2 \cdot \Pr[\text{MSE}^{\mathcal{S}, \mathcal{D}}] - 1$, where the game $\text{MSE}^{\mathcal{S}, \mathcal{D}}$ is given in Figure 2. For $q_{\mathcal{S}}, q_{\mathcal{D}}, q_{\mathcal{P}} \in \mathbb{N}$ and $\delta, \epsilon \in \mathbb{R}_{\geq 0}$, we say that Mono is a $(q_{\mathcal{P}}, \delta, q_{\mathcal{S}}, q_{\mathcal{D}}, \epsilon)$ -multi-source-extractor (MSE for short) if for every $(\mathcal{S}, \mathcal{D})$ as above such that, for every $i \in [\ell]$, \mathcal{S}_i is $(q_{\mathcal{P}}, \delta)$ -unpredictable and makes at most $q_{\mathcal{S}}$ oracle calls, and such that \mathcal{D} makes at most $q_{\mathcal{D}}$ oracle calls, we have $\text{Adv}_{\mathcal{S}, \mathcal{D}}^{\text{mse}} \leq \epsilon$.

► **Remark 11.** Notice that different restrictions on the query complexity q of the parties above allow to formulate qualitatively different results. When q is “small,” one can use well-known techniques (e.g., lazy sampling) to prove security of a given construction. Usually, bounds obtained in this way become meaningless in the regime where q is “large,” e.g., large enough to query H on all possible inputs (which is equivalent to giving H in full as input), so different techniques (e.g., along the lines of [16]) are necessary in this setting.

► **Remark 12.** In this work, we opt for a concrete treatment of security, but an asymptotic formulation is also possible. In that setting, all idealized models, parties, and security notions depend on a security parameter λ . For security to hold, advantages must be negligible in λ . Parties are probabilistic and can be either polynomial-time in λ or unbounded, and in the latter case either have polynomially bounded oracle access to H_λ , or get H_λ in full as input.

► **Remark 13.** Let $\ell, M, N \in \mathbb{N}$, $\mathcal{S} = (\mathcal{S}_1, \dots, \mathcal{S}_\ell)$ a tuple of sources, and \mathcal{A} an adversary in the UR game for Mono. Without loss of generality, we can assume \mathcal{A} to be deterministic. Indeed, if R denotes the random variable representing the randomness of (the possibly randomized adversary) \mathcal{A} , we have

$$\begin{aligned} \text{Adv}_{\mathcal{S}, \mathcal{A}}^{\text{ur}} &= \Pr[\text{UR}^{\mathcal{S}, \mathcal{A}}] = \sum_r \Pr[\text{UR}^{\mathcal{S}, \mathcal{A}} \mid R = r] \Pr[R = r] \\ &= \sum_r \Pr[\text{UR}^{\mathcal{S}, \mathcal{A}[r]}] \Pr[R = r] = \sum_r \text{Adv}_{\mathcal{S}, \mathcal{A}[r]}^{\text{ur}} \Pr[R = r], \end{aligned}$$

where $\mathcal{A}[r]$ is the deterministic adversary \mathcal{A} with hard-coded randomness r . This means that there must exist r such that $\text{Adv}_{\mathcal{S},\mathcal{A}}^{\text{ur}} \leq \text{Adv}_{\mathcal{S},\mathcal{A}[r]}^{\text{ur}}$, showing that for every (possibly randomized) adversary \mathcal{A} , there exists a deterministic adversary \mathcal{B} such that $\text{Adv}_{\mathcal{S},\mathcal{A}}^{\text{ur}} \leq \text{Adv}_{\mathcal{S},\mathcal{B}}^{\text{ur}}$. It is therefore sufficient to upper-bound the advantage of $(\mathcal{S}, \mathcal{A})$ in the UR game for Mono when \mathcal{A} is deterministic. A similar remark holds for distinguishers \mathcal{D} in the MSE game for Mono.

Generic Implications. We begin by studying the relative strength of the security notions $\text{SEC} \in \{\text{UR}, \text{MSE}\}$. Clearly, if Mono is SEC-secure for a given set of parameters, it remains secure if the sources and the adversary (resp., the distinguisher) are allowed a smaller number of queries, and the predictor can make more queries. Moreover, if Mono is MSE-secure for a given set of parameters, it is also UR-secure in certain parameter ranges. In the following, fix $\ell, M, N \in \mathbb{N}$.

► **Proposition 14.** *Let $q_{\mathcal{S}}, q'_{\mathcal{S}}, q_{\mathcal{A}}, q'_{\mathcal{A}}, q_{\mathcal{D}}, q'_{\mathcal{D}}, q_{\mathcal{P}}, q'_{\mathcal{P}} \in \mathbb{N}$ and $\delta, \delta', \epsilon, \epsilon' \in \mathbb{R}_{\geq 0}$, such that $q'_{\mathcal{S}} \leq q_{\mathcal{S}}, q'_{\mathcal{A}} \leq q_{\mathcal{A}}, q'_{\mathcal{D}} \leq q_{\mathcal{D}}, q'_{\mathcal{P}} \geq q_{\mathcal{P}}, \delta' \leq \delta$, and $\epsilon' \geq \epsilon$. (1) If Mono is $(q_{\mathcal{P}}, \delta, q_{\mathcal{S}}, q_{\mathcal{A}}, \epsilon)$ -unrecoverable, then it is also $(q'_{\mathcal{P}}, \delta', q'_{\mathcal{S}}, q'_{\mathcal{A}}, \epsilon')$ -unrecoverable. (2) If Mono is a $(q_{\mathcal{P}}, \delta, q_{\mathcal{S}}, q_{\mathcal{D}}, \epsilon)$ -MSE, then it is also a $(q'_{\mathcal{P}}, \delta', q'_{\mathcal{S}}, q'_{\mathcal{D}}, \epsilon')$ -MSE.*

Proof. We only prove (1), as the proof of (2) is identical (except that \mathcal{A} is replaced with \mathcal{D} in the following). Let $\mathcal{S} = (\mathcal{S}_1, \dots, \mathcal{S}_{\ell})$ be any tuple of sources such that, for every $i \in [\ell]$, \mathcal{S}_i is $(q'_{\mathcal{P}}, \delta')$ -unpredictable and makes at most $q'_{\mathcal{S}}$ oracle calls, and let \mathcal{A} be any adversary making at most $q'_{\mathcal{A}}$ oracle calls. Then clearly, for each $i \in [\ell]$, \mathcal{S}_i is also $(q_{\mathcal{P}}, \delta)$ -unpredictable and makes at most $q_{\mathcal{S}}$ oracle calls, and \mathcal{A} makes at most $q_{\mathcal{A}}$ oracle calls. By assumption, this means that $\text{Adv}_{\mathcal{S},\mathcal{A}}^{\text{ur}} \leq \epsilon \leq \epsilon'$. Since this inequality holds for every pair $(\mathcal{S}, \mathcal{A})$ as above, by definition Mono is $(q'_{\mathcal{P}}, \delta', q'_{\mathcal{S}}, q'_{\mathcal{A}}, \epsilon')$ -unrecoverable. ◀

► **Proposition 15.** *Let $q_{\mathcal{S}}, q_{\mathcal{D}}, q_{\mathcal{P}} \in \mathbb{N}$ with $q_{\mathcal{D}} \geq 1$, and $\delta, \epsilon \in \mathbb{R}_{\geq 0}$. If Mono is a $(q_{\mathcal{P}}, \delta, q_{\mathcal{S}}, q_{\mathcal{D}}, \epsilon)$ -MSE, then it is $(q_{\mathcal{P}}, \delta, q_{\mathcal{S}}, q_{\mathcal{D}} - 1, \epsilon + N^{\ell}/M)$ -unrecoverable.*

Proof. Let $\mathcal{S} = (\mathcal{S}_1, \dots, \mathcal{S}_{\ell})$ be a tuple of sources such that, for every $i \in [\ell]$, \mathcal{S}_i is $(q_{\mathcal{P}}, \delta)$ -unpredictable and makes at most $q_{\mathcal{S}}$ oracle calls. Let \mathcal{A} be an adversary making at most $q_{\mathcal{D}} - 1$ oracle calls, and consider the distinguisher \mathcal{D} which runs \mathcal{A} on its input (y, z) to get \mathbf{x}' , and then returns the bit $(H(\mathbf{x}') = y)$. Then clearly \mathcal{D} makes at most $q_{\mathcal{D}}$ oracle calls. Furthermore notice that, when $b = 1$ in $\text{MSE}^{\mathcal{S}, \mathcal{D}}$, distinguisher \mathcal{D} will return 1 if \mathcal{A} succeeds in finding a preimage, so that $\Pr[\text{MSE}^{\mathcal{S}, \mathcal{D}} \mid b = 1] \geq \text{Adv}_{\mathcal{S},\mathcal{A}}^{\text{ur}}$. On the other hand, when $b = 0$, we have:

$$\begin{aligned} \Pr[\text{MSE}^{\mathcal{S}, \mathcal{D}} \mid b = 0] &\geq \Pr[\text{MSE}^{\mathcal{S}, \mathcal{D}} \mid (b = 0) \wedge (y \notin \text{Rng}(H))] \Pr[y \notin \text{Rng}(H)] \\ &= \Pr[y \notin \text{Rng}(H)], \end{aligned}$$

where the last equality holds because when $y \notin \text{Rng}(H)$, \mathcal{D} will always return 0. Now notice that $\Pr[y \notin \text{Rng}(H)] = (1 - 1/M)^{N^{\ell}} \geq 1 - N^{\ell}/M$, which means

$$\epsilon \geq \text{Adv}_{\mathcal{S},\mathcal{A}}^{\text{mse}} \geq \text{Adv}_{\mathcal{S},\mathcal{A}}^{\text{ur}} + 1 - \frac{N^{\ell}}{M} - 1 = \text{Adv}_{\mathcal{S},\mathcal{A}}^{\text{ur}} - \frac{N^{\ell}}{M}. \quad \blacktriangleleft$$

► **Remark 16.** If bounds on the support sizes $|\text{Supp}(\mathcal{S}_i)|$ are known, we can give a tighter lower bound of $\Pr[y \notin \text{Rng}(H)] \geq 1 - |\text{Supp}(\mathcal{S}_1)| \cdots |\text{Supp}(\mathcal{S}_{\ell})|/M$, showing that Mono is $(q_{\mathcal{P}}, \delta, q_{\mathcal{S}}, q_{\mathcal{D}} - 1, \epsilon + |\text{Supp}(\mathcal{S}_1)| \cdots |\text{Supp}(\mathcal{S}_{\ell})|/M)$ -unrecoverable. Also, notice that this result holds for *any* construction, not just for the monolithic random oracle Mono.

<p>Src. \mathcal{S}_i^H:</p> <p>$E_H \leftarrow \{x \in [M] \subseteq [N] \mid H(x, x) = x\}$</p> <p>if $(E_H \neq \emptyset)$ then $x \leftarrow \min E_H$</p> <p> else $x \leftarrow [M]$</p> <p>$z \leftarrow \varepsilon$; return (x, z)</p>	<p>Dist. $\mathcal{D}^H(y, z)$:</p> <p>return $(H(y, y) = y)$</p>	<p>Game \mathbf{G}_1:</p> <p>$H \leftarrow E$</p> <p>$x' \leftarrow \mathcal{P}^H(\varepsilon)$</p> <p>return $(H(x', x') = x')$</p>
--	--	---

■ **Figure 3** Definition of sources \mathcal{S}_i , $i \in [2]$, of the distinguisher \mathcal{D} , and of the game \mathbf{G}_1 from the proof of Proposition 17.

4 Attacks

In this section, we present attacks against the configurations UBB (Proposition 17) and BUB (Proposition 18) in Figure 1. The former result also rules out UUB, which implies UBB.

► **Proposition 17.** *Let $M, N, q_{\mathcal{P}} \in \mathbb{N}$ with $M \leq N$, and set $\ell := 2$. Then there exist a pair of sources $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$, each of which is $(q_{\mathcal{P}}, 2(q_{\mathcal{P}} + 2)/M)$ -unpredictable and makes $q_{\mathcal{S}} = M$ oracle calls, and a distinguisher \mathcal{D} making $q_{\mathcal{D}} = 1$ oracle calls, such that*

$$\text{Adv}_{\mathcal{S}, \mathcal{D}}^{\text{mse}} \geq 1 - \frac{1}{e} - \frac{1}{M}. \quad (1)$$

Proof. For every $i \in [2]$, let \mathcal{S}_i be the source defined in Figure 3, which returns the smallest $x \in [M] \subseteq [N]$ such that $H(x, x) = x$ if such an x exists, and a random value $x \in [M]$ otherwise. Notice that this definition of \mathcal{S}_i requires it to be allowed up to M oracle queries. Furthermore, let \mathcal{D} be the distinguisher defined in Figure 3, which checks if its input $y \in [M]$ satisfies $H(y, y) = y$.

To show that \mathcal{S}_i is $(q_{\mathcal{P}}, 2(q_{\mathcal{P}} + 2)/M)$ -unpredictable, let \mathcal{P} be any predictor in the prediction game for \mathcal{S}_i making at most $q_{\mathcal{P}}$ oracle calls, and let

$$E := \{H \in \text{Fun}(N^2, M) \mid (\exists x \in [M] \subseteq [N])(H(x, x) = x)\}$$

be the set of functions having a collision as in the definition of \mathcal{S}_i . A direct counting argument gives $|E^c| = (M - 1)^M \cdot M^{N^2 - M}$, from which we infer that $|E| = M^{N^2} - (M - 1)^M \cdot M^{N^2 - M}$. The law of total probability now gives

$$\begin{aligned} \Pr[\text{Pred}_{\mathcal{S}_i}^{\mathcal{P}}] &\leq \Pr[\text{Pred}_{\mathcal{S}_i}^{\mathcal{P}} \mid H \in E] + \Pr[\text{Pred}_{\mathcal{S}_i}^{\mathcal{P}} \mid H \notin E] \\ &= \Pr[\text{Pred}_{\mathcal{S}_i}^{\mathcal{P}} \mid H \in E] + \frac{1}{M} \leq \Pr[\mathbf{G}_1] + \frac{1}{M}, \end{aligned}$$

where \mathbf{G}_1 is the game defined in Figure 3. Here, the last inequality holds because the uniform distribution on $\text{Fun}(N^2, M)$, conditioned on the outcome being in E , is the uniform distribution on E , and because the winning condition in \mathbf{G}_1 is less restrictive than the one in $\text{Pred}_{\mathcal{S}_i}^{\mathcal{P}}$, given that $H \in E$. Indeed, in $\text{Pred}_{\mathcal{S}_i}^{\mathcal{P}}$, predictor \mathcal{P} must guess the exact output x of \mathcal{S}_i , which satisfies $H(x, x) = x$, whereas in \mathbf{G}_1 it wins if it returns any such value $x' \in [N]$.

To bound $\Pr[\mathbf{G}_1]$, we transition to yet another game \mathbf{G}_2 that proceeds as \mathbf{G}_1 , but we let \mathcal{P} win if any of its oracle queries of the form $(z, z) \in [M]^2$ (there are at most $q_{\mathcal{P}}$ of them), or its final output x' , satisfy the winning condition of \mathbf{G}_1 . Then again $\Pr[\mathbf{G}_1] \leq \Pr[\mathbf{G}_2]$, because \mathcal{P} will always win \mathbf{G}_2 if it wins \mathbf{G}_1 . Now notice that, by definition, predictor \mathcal{P} will lose game \mathbf{G}_2 if for all queries of the form $(z, z) \in [M]^2$ it makes, it holds $H(z, z) \neq z$, and the same is true for its output x' . The probability of this happening is at least

$$\frac{(M - 1)^{q_{\mathcal{P}} + 1} \cdot M^{N^2 - (q_{\mathcal{P}} + 1)} - |E^c|}{|E|} = \frac{(1 - 1/M)^{q_{\mathcal{P}} + 1} - (1 - 1/M)^M}{1 - (1 - 1/M)^M},$$

<p><u>Src. \mathcal{S}_i^H:</u> $x \leftarrow [N]; z \leftarrow H(x, x)$ return (x, z)</p> <p><u>Dist. $\mathcal{D}^H(y, z)$:</u> for $i = 1$ to 2 do $X_i \leftarrow \{x \in [N] \mid H(x, x) = z_i\}$ return $(\exists x_1 \in X_1)(\exists x_2 \in X_2)$ $(H(x_1, x_2) = y)$</p>	<p><u>Game \mathbf{G}_1:</u> $T \leftarrow \emptyset; z \leftarrow [M]; x' \leftarrow \mathcal{P}^H(z); x \leftarrow [N]$ if $((x, x) \in \text{Dom}(T))$ then return 1 $T[(x, x)] \leftarrow z; \textbf{return } (x = x')$</p> <p><u>Game \mathbf{G}_2:</u> $T \leftarrow \emptyset; z \leftarrow [M]; x' \leftarrow \mathcal{P}^H(z); x \leftarrow [N]$ return $(x = x')$</p> <p><u>Proc. $H(u, v)$:</u> if $((u, v) \notin \text{Dom}(T))$ then $T[(u, v)] \leftarrow [M]$ return $T[(u, v)]$</p>
---	---

■ **Figure 4** Definition of sources \mathcal{S}_i , $i \in [2]$, of the distinguisher \mathcal{D} , and of the games \mathbf{G}_1 and \mathbf{G}_2 from the proof of Proposition 18.

which means

$$\Pr[\mathbf{G}_2] \leq 1 - \frac{(1 - 1/M)^{q_{\mathcal{P}}+1} - (1 - 1/M)^M}{1 - (1 - 1/M)^M} \leq \frac{q_{\mathcal{P}} + 1}{M} \cdot \frac{e}{e - 1} \leq \frac{2(q_{\mathcal{P}} + 1)}{M}.$$

Here, the second step uses Inequality 2 from Lemma 4. Collecting all terms above, we obtain the claimed unpredictability bound for sources \mathcal{S}_i .

We are now left with bounding the MSE-advantage of $(\mathcal{S}, \mathcal{D})$. To that end, observe that the probability of $(\mathcal{S}, \mathcal{D})$ winning the MSE game when $b = 0$ is $1 - 1/M$, because for any value $y \in [M] \subseteq [N]$, H returns y on any input with probability $1/M$. On the other hand,

$$\begin{aligned} \Pr[\text{MSE}^{\mathcal{S}, \mathcal{D}} \mid b = 1] &\geq \Pr[\text{MSE}^{\mathcal{S}, \mathcal{D}} \mid (H \in E) \wedge (b = 1)] \Pr[H \in E \mid b = 1] \\ &= \Pr[H \in E] = \frac{M^{N^2} - (M - 1)^M \cdot M^{N^2 - M}}{M^{N^2}} = 1 - \left(1 - \frac{1}{M}\right)^M \geq 1 - \frac{1}{e}, \end{aligned}$$

where the last inequality follows from Inequality 1 from Lemma 4. Collecting the two terms above, we obtain (1). ◀

► **Proposition 18.** *Let $M, N, q_{\mathcal{P}} \in \mathbb{N}$, and set $\ell := 2$. Then there exist a pair of sources $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$, each of which is $(q_{\mathcal{P}}, (q_{\mathcal{P}} + 1)/N)$ -unpredictable and makes $q_{\mathcal{S}} = 1$ oracle calls, and a distinguisher \mathcal{D} making $q_{\mathcal{D}} = N^2$ oracle calls, such that*

$$\text{Adv}_{\mathcal{S}, \mathcal{D}}^{\text{mse}} \geq 1 - \frac{\bar{M}^2}{M} - M \cdot \left(\frac{eN}{M\bar{M}}\right)^{\bar{M}}, \quad (2)$$

where $\bar{M} := \lceil \sqrt[4]{M} \rceil$.

Proof. For every $i \in [2]$, let \mathcal{S}_i be the source defined in Figure 4, which returns a random $x \in [N]$ and $z = H(x, x)$ as leakage. Furthermore, let \mathcal{D} be the distinguisher defined in Figure 4, which finds, for each $i \in [2]$, all inputs of the form (x_i, x_i) satisfying $H(x_i, x_i) = z_i$. Notice that this definition of \mathcal{D} requires it to be allowed up to N^2 oracle queries.

To show that \mathcal{S}_i is $(q_{\mathcal{P}}, (q_{\mathcal{P}} + 1)/N)$ -unpredictable, let \mathcal{P} be any predictor in the prediction game for \mathcal{S}_i . Then we have

$$\Pr[\text{Pred}_{\mathcal{S}_i}^{\mathcal{P}}] \leq \Pr[\mathbf{G}_1] \leq \Pr[\mathbf{G}_2] + \frac{q_{\mathcal{P}}}{N} = \frac{1}{N} + \frac{q_{\mathcal{P}}}{N} = \frac{q_{\mathcal{P}} + 1}{N},$$

where games G_1 and G_2 are defined in Figure 4. Here, the first inequality holds because \mathcal{P} will always succeed in G_1 if it wins the prediction game for \mathcal{S}_i , and the second inequality follows from the fundamental lemma of game playing. Indeed, if we let **Bad** be the event in game G_1 that $(x, x) \in \text{Dom}(T)$, then G_1 and G_2 are identical until **Bad**, and $\Pr[\text{Bad}] \leq q_{\mathcal{P}}/N$ because x is random in $[N]$ and $\text{Dom}(T)$ contains at most $q_{\mathcal{P}}$ entries. Finally, the first equality in the chain above holds because x is uniformly random in $[N]$, and will thus be equal to x' with probability $1/N$.

We are now left with bounding the MSE-advantage of $(\mathcal{S}, \mathcal{D})$. To that end, observe that the probability of $(\mathcal{S}, \mathcal{D})$ winning the MSE game for **Mono** when $b = 1$ is 1, because when y is the real hash value, then a preimage of y in $X_1 \times X_2$ exists by construction. On the other hand, for the case $b = 0$, define

$$E := \left\{ H \in \text{Fun}(N^2, M) \mid \begin{array}{l} (\forall k \in \mathbb{N}) ((\exists x_1, \dots, x_k \in [N]) \\ (x_i \neq x_j \wedge H(x_i, x_i) = H(x_j, x_j)) \implies (k < \bar{M})) \end{array} \right\},$$

where $\bar{M} := \lceil \sqrt[4]{M} \rceil$. In other words, E is the set of all functions $H \in \text{Fun}(N^2, M)$ such that there are less than \bar{M} pairwise distinct elements $(x_i, x_i) \in [N] \times [N]$, all mapping to the same value under H . Denote by G the game $\text{MSE}^{\mathcal{S}, \mathcal{D}}$ with bit $b = 0$ fixed and inverted winning condition. Then $\Pr[\text{MSE}^{\mathcal{S}, \mathcal{D}} \mid b = 0] = 1 - \Pr[G]$, and by the law of total probability,

$$\Pr[G] \leq \Pr[G \mid H \in E] + \Pr[H \notin E] \leq \frac{\bar{M}^2}{M} + \binom{N}{\bar{M}} \cdot M^{1-2\bar{M}}.$$

Here, the bound on the first term follows from the fact that, since $H \in E$, both X_1 and X_2 contain at most $\bar{M} - 1$ elements, and therefore $|H(X_1 \times X_2)| < \bar{M}^2$. Since $y \in [M]$ is random, it will hit an element in this set with probability at most \bar{M}^2/M . For the second term, observe that if $H \notin E$, then there exist at least \bar{M} elements $x_i \in [N]$ such that $H(x_i, x_i)$ collide, an event happening with probability

$$\frac{\binom{N}{\bar{M}} \cdot M \cdot M^{N^2 - \bar{M}}}{M^{N^2}} = \binom{N}{\bar{M}} \cdot M^{1-\bar{M}} \leq \left(\frac{eN}{\bar{M}} \right)^{\bar{M}} \cdot M^{1-\bar{M}} = M \cdot \left(\frac{eN}{M\bar{M}} \right)^{\bar{M}}.$$

Here, the inequality follows from Inequality 3 in Lemma 4. Collecting the terms above, we obtain (2). \blacktriangleleft

5 The Bounded Case

Having covered attacks against the MSE-notion in the previous section, we now turn to positive results. In this section, we prove that the monolithic construction **Mono** is a multi-source extractor in the setting where all parties have bounded-query access to the random oracle (i.e., the BBB corner in Figure 1). Observe that this result also implies security in the BBU setting by Proposition 14. Our main result, where sources and predictors can query the entire oracle, is presented in Section 6.

► **Theorem 19.** *Let $\ell, M, N, q_{\mathcal{S}}, q_{\mathcal{D}}, q_{\mathcal{P}} \in \mathbb{N}$ with $\ell \geq 2$ and $q_{\mathcal{P}} \geq (\ell - 1)q_{\mathcal{S}} + q_{\mathcal{D}}$, and $k \in \mathbb{R}_{\geq 0}$. Then **Mono** is a $(q_{\mathcal{P}}, 2^{-k}, q_{\mathcal{S}}, q_{\mathcal{D}}, \epsilon)$ -MSE, where ϵ is given in the right-hand side of (3). More precisely, for every tuple $\mathcal{S} = (\mathcal{S}_1, \dots, \mathcal{S}_{\ell})$ of sources, each of which is $(q_{\mathcal{P}}, 2^{-k})$ -unpredictable and makes at most $q_{\mathcal{S}}$ oracle calls, and every distinguisher \mathcal{D} making at most $q_{\mathcal{D}}$ oracle calls, we have*

$$\text{Adv}_{\mathcal{S}, \mathcal{D}}^{\text{mse}} \leq \frac{\ell q_{\mathcal{S}} + q_{\mathcal{D}}}{2^k}. \quad (3)$$

Game G_1: $H \leftarrow \text{Fun}(N^\ell, M); Q \leftarrow \emptyset$ for $i = 1$ to ℓ do $(x_i, z_i) \leftarrow \mathcal{S}_i^O$ $y \leftarrow H(x);$ return $\mathcal{D}^O(y, z)$ Proc. $O(q)$: $Q \leftarrow Q \cup \{q\};$ return $H(q)$	Game G_2: $H \leftarrow \text{Fun}(N^\ell, M); Q \leftarrow \emptyset$ for $i = 1$ to ℓ do $(x_i, z_i) \leftarrow \mathcal{S}_i^O$ $y \leftarrow H(x); b' \leftarrow \mathcal{D}^O(y, z)$ if $(x \in Q)$ then return 1 return b'	Game G_3: $H \leftarrow \text{Fun}(N^\ell, M); Q \leftarrow \emptyset$ for $i = 1$ to ℓ do $(x_i, z_i) \leftarrow \mathcal{S}_i^O$ $y \leftarrow [M]; b' \leftarrow \mathcal{D}^O(y, z)$ if $(x \in Q)$ then return 1 return b'
Pred. $\mathcal{P}_j^H(z_j)$: $Q \leftarrow \emptyset; z_j \leftarrow z_j; (x_j, z_j) \leftarrow \mathcal{S}_j^O$ for $i \in [\ell] \setminus \{j, \bar{j}\}$ do $(x_i, z_i) \leftarrow \mathcal{S}_i^H$ $y \leftarrow [M]; b' \leftarrow \mathcal{D}^H(y, z)$ $q \leftarrow Q; \text{return } q_j$	Pred. $\mathcal{Q}^H(z_1)$: $Q \leftarrow \emptyset; z_1 \leftarrow z_1$ for $i = 2$ to ℓ do $(x_i, z_i) \leftarrow \mathcal{S}_i^H$ $y \leftarrow [M]; b' \leftarrow \mathcal{D}^O(y, z)$ $q \leftarrow Q; \text{return } q_1$	

■ **Figure 5** Code of the intermediate games, and definition of the predictors \mathcal{P}_j and \mathcal{Q} , from the proof of Theorem 19. Oracle O is the same throughout the figure.

Proof. Consider a tuple of sources $\mathcal{S} = (\mathcal{S}_1, \dots, \mathcal{S}_\ell)$ such that, for every $i \in [\ell]$, \mathcal{S}_i is $(q_{\mathcal{P}}, 2^{-k})$ -unpredictable and makes at most $q_{\mathcal{S}}$ oracle calls, and a distinguisher \mathcal{D} making at most $q_{\mathcal{D}}$ oracle calls. We prove bound (3) via a sequence of games, whose formal description can be found in Figure 5:

- G_0 is the MSE-game for Mono played by $(\mathcal{S}, \mathcal{D})$, with bit $b = 1$ fixed.
- G_1 proceeds as G_0 , but we use a set Q to track all the queries made by the sources \mathcal{S}_i and the distinguisher \mathcal{D} . Clearly, G_0 and G_1 are indistinguishable, since the only difference is that G_1 records the queries made by the parties, and otherwise all distributions are the identical. Therefore, $\Pr[G_1] = \Pr[G_0]$.
- G_2 proceeds as G_1 , but we return 1 if any source \mathcal{S}_i or distinguisher \mathcal{D} query vector x (which is used when computing $y = H(x)$) to the random oracle H . Notice that G_2 returns 1 whenever G_1 does, so that $\Pr[G_2] \geq \Pr[G_1]$.
- G_3 proceeds as G_2 , but we no longer compute y as $H(x)$, and instead sample it at random. We claim that G_2 and G_3 are indistinguishable. Indeed, if $x \in Q$, both games return 1 by definition. If $x \notin Q$, then $H(x)$ is a random value that is unknown to all parties, so computing y as $y = H(x)$ is equivalent to sampling it at random. Thus, $\Pr[G_3] = \Pr[G_2]$.
- G_4 proceeds as G_3 , but we remove the if-statement added in G_2 . Since we no longer need to track the oracle queries of \mathcal{S}_i and \mathcal{D} , we replace O with H , and do not initialize Q . The resulting game coincides with the MSE-game for Mono played by $(\mathcal{S}, \mathcal{D})$ with bit $b = 0$ fixed and inverted winning condition.

To study the difference in success probability between G_3 and G_4 , let Bad_0 be the event that \mathcal{D} queries x while playing G_3 , and Bad_i the event that \mathcal{S}_i queries x in the same game, for $i \in [\ell]$. Then notice that G_3 and G_4 are identical until $\text{Bad} := \bigvee_{i=0}^{\ell} \text{Bad}_i$, and by the fundamental lemma of game playing we therefore have

$$|\Pr[G_4] - \Pr[G_3]| = \Pr[\text{Bad}] \leq \Pr[\text{Bad}_0] + \sum_{i=1}^{\ell} \Pr[\text{Bad}_i].$$

We study these probabilities separately. For $j \in [\ell]$, let $\bar{j} := (j \bmod \ell) + 1$, and consider the predictor \mathcal{P}_j playing the predictability game for $\mathcal{S}_{\bar{j}}$ defined in Figure 5. Notice that \mathcal{P}_j makes at most $(\ell - 1)q_{\mathcal{S}} + q_{\mathcal{D}}$ oracle calls, which means that

$$2^{-k} \geq \Pr[\text{Pred}_{\mathcal{S}_{\bar{j}}}^{\mathcal{P}_j}] \geq \Pr[\text{Pred}_{\mathcal{S}_{\bar{j}}}^{\mathcal{P}_j} \mid \text{Bad}_j] \Pr[\text{Bad}_j] \geq \frac{1}{q_{\mathcal{S}}} \Pr[\text{Bad}_j].$$

From here we get $\Pr[\text{Bad}_j] \leq q_S \cdot 2^{-k}$ for every $j \in [\ell]$. Similarly, let \mathcal{Q} be the predictor playing the predictability game for \mathcal{S}_1 defined in Figure 5. As before, \mathcal{Q} queries H at most $(\ell - 1)q_S + q_D$ many times, and therefore $\Pr[\text{Bad}_0] \leq q_D \cdot 2^{-k}$.

Combining the estimates above we obtain

$$\text{Adv}_{\mathcal{S}, \mathcal{D}}^{\text{mse}} = \Pr[\mathbf{G}_0] - \Pr[\mathbf{G}_4] \leq |\Pr[\mathbf{G}_4] - \Pr[\mathbf{G}_3]| \leq \epsilon. \quad \blacktriangleleft$$

► **Remark 20.** Observe that Theorem 19 is false when $\ell = 1$. Indeed, we bound the probability of \mathcal{S}_j querying \mathbf{x} by constructing a predictor against *another* source in \mathcal{S} (e.g., \mathcal{S}_{j+1}). When $\ell = 1$, there is no “other” source to do that, and this step in the proof breaks down. Also notice that it is actually impossible to prevent \mathcal{S}_1 from querying \mathbf{x} in this case, because $\mathbf{x} = \mathbf{x}_1$ is known to \mathcal{S}_1 .

More importantly, we can present an attack which relies on this fact: Consider the source \mathcal{S}_1 which samples $x \leftarrow [N]$, sets $z \leftarrow (H(x) \leq M/2)$ and returns (x, z) , and the distinguisher \mathcal{D} who returns the bit $((y \leq M/2) = z)$. Then it is easy to see that \mathcal{S}_1 is (q_P, δ) -unpredictable for any q_P and $\delta = 1/\lfloor M/2 \rfloor \approx 2/M$, and yet $\text{Adv}_{\mathcal{S}, \mathcal{D}}^{\text{mse}} \leq 1 - \lfloor M/2 \rfloor / M \approx 1/2$ for $\mathcal{S} = (\mathcal{S}_1)$. The issue is that when $\ell = 1$, \mathcal{S}_1 knows the entire input of H and can therefore leak information about the output $H(x)$, which then helps \mathcal{D} .

6 The Unbounded Case

In this section, we prove our main positive result, showing that the monolithic construction **Mono** is a multi-source extractor in the setting where the distinguisher is the only party with bounded-query access to the random oracle (i.e., the UBU corner in Figure 1). In particular, each source can fully depend on the random oracle.

Following Dodis et al. [16], our approach is as follows: To show that the output of H on the values \mathbf{x}_i returned by the ℓ sources is pseudorandom, we first use Yao’s equivalence between distinguishing and next-bit prediction to turn any distinguisher with advantage ϵ into a next-bit guesser \mathcal{P} with advantage $\epsilon/\log M$ for some bit in $H(\mathbf{x})$, say, the l th bit. Two cases are now possible:

1. \mathcal{P} queries \mathbf{x} with “good” probability $\epsilon/(2 \log M)$. We can then turn \mathcal{P} into an adversary \mathcal{Q} in the UR game with advantage $\epsilon/(2 \log M)$, by letting \mathcal{Q} inspect the queries of \mathcal{P} and returning the one mapping to $H(\mathbf{x})$. The advantage of \mathcal{Q} can be bounded by a compression argument similar to the one in [16]. However, special care needs to be taken because unlike in [16], the challenge \mathbf{x} is not chosen uniformly and independently of H , but is the output of sources that have full oracle access. We deal with this issue by decomposing the high-entropy sources into convex combinations of flat sources on large sets, and then proceed similarly to the uniform case.
2. \mathcal{P} queries \mathbf{x} with probability less than $\epsilon/(2 \log M)$, while still having advantage $\epsilon/\log M$ in guessing the l th bit of $H(\mathbf{x})$. We can then turn \mathcal{P} into a guesser \mathcal{Q} that never queries H on \mathbf{x} but still has advantage $\epsilon/(2 \log M)$ in guessing the l th bit of $H(\mathbf{x})$, by simply running \mathcal{P} and giving up whenever \mathcal{P} wants to query \mathbf{x} . Predictor \mathcal{Q} can then be used in an incompressibility argument to ultimately bound ϵ . For this part we follow De et al. [13, Lemma 8.4], bearing in mind that the \mathbf{x}_i are non-uniform and oracle-dependent.

Following the roadmap outlined above, we begin by stating our unrecoverability result and a preparatory lemma for our main theorem. In the interest of space, we present the proofs of these statements in the full version of the paper [12].

► **Theorem 21** (Unrecoverability). *Let $\ell, M, N, q_A \in \mathbb{N}$ and $k \in \mathbb{R}_{>0}$. Then Mono is $(N^\ell, 2^{-k}, N^\ell, q_A, \epsilon)$ -unrecoverable, where ϵ is given in the right-hand side of (4). More precisely, for every tuple $\mathcal{S} = (S_1, \dots, S_\ell)$ of sources, each of which is $(N^\ell, 2^{-k})$ -unpredictable and has unbounded oracle access, and every adversary \mathcal{A} making at most q_A oracle calls, we have*

$$\text{Adv}_{\mathcal{S}, \mathcal{A}}^{\text{ur}} = \mathcal{O}\left(\ell^{\ell+1} \sqrt{\frac{q_A}{2^{k\ell}}} (\sigma + \ell N)\right). \quad (4)$$

Here, $\sigma \geq \sum_{i=1}^{\ell} \sigma_i$, where $\sigma_i \in \mathbb{N}$ is a bound on the length of the leakage z_i returned by S_i .

► **Lemma 22.** *Let $N, \ell, q_A, \sigma, S \in \mathbb{N}$ and $\epsilon \in \mathbb{R}_{>0}$, so that $S^\ell \geq 40$ and $q_A \leq S^\ell / \log S^\ell$. Let \mathcal{A} be a deterministic algorithm that makes at most q_A queries to its oracle, takes an advice string z of length at most σ , and is not allowed to query its input. Let $O \subseteq \text{Fun}(N^\ell, \{0, 1\})$ be a set such that, for each $O \in O$, there exist $z \in \{0, 1\}^\sigma$ and sets $S_i \subseteq [N]$ with $|S_i| = S$ for all $i \in [\ell]$ such that*

$$\Pr_{\mathbf{x} \leftarrow S_i} [\mathcal{A}^O(\mathbf{x}, z) = O(\mathbf{x})] \geq \frac{1}{2} + \epsilon.$$

Then either $q_A > \epsilon S^\ell$, or there exists a randomized encoding scheme (Enc, Dec) for O such that, for every $O \in O$,

$$\Pr[\text{Dec}(\text{Enc}(O, r), r) = O] = \Omega\left(\frac{\epsilon}{q_A}\right),$$

and the length of each codeword is at most

$$\sigma + 2\ell S \log\left(\frac{eN}{S}\right) + N^\ell - \Omega\left(\frac{\epsilon^2 S^\ell}{q_A}\right) + \mathcal{O}(1).$$

Finally, we come to the main theorem of this section, establishing that the random oracle is a good multi-source extractor.

► **Theorem 23** (MSE). *Let $\ell, M, N, q_D \in \mathbb{N}$ and $k \in \mathbb{R}_{>0}$. Then Mono is $(N^\ell, 2^{-k}, N^\ell, q_D, \epsilon)$ -MSE, where ϵ is given in the right-hand side of (5). More precisely, for every tuple of sources $\mathcal{S} = (S_1, \dots, S_\ell)$, each of which is $(N^\ell, 2^{-k})$ -unpredictable and has unbounded access to its oracle, and every distinguisher \mathcal{D} making at most q_D oracle calls, we have*

$$\text{Adv}_{\mathcal{S}, \mathcal{D}}^{\text{mse}} = \mathcal{O}\left(\ell \log M^{\ell+1} \sqrt{\frac{q_D}{2^{k\ell}}} (\sigma + \ell N)\right). \quad (5)$$

Here, $\sigma \geq \sum_{i=1}^{\ell} \sigma_i$, where $\sigma_i \in \mathbb{N}$ is a bound on the length of the leakage z_i returned by S_i .

Proof. Let $\epsilon := \text{Adv}_{\mathcal{S}, \mathcal{D}}^{\text{mse}}$. By Yao's equivalence principle [36], any q_D -query distinguisher can be converted into a predictor \mathcal{P} with the same query complexity, which can predict the l th output bit of the oracle for some $l \in [\log M]$, that is,

$$\Pr[\text{nbPred}_{\mathcal{S}, l}^{\mathcal{P}}] \geq \frac{1}{2} + \epsilon',$$

where $\epsilon' := \epsilon / \log M$ and $\text{nbPred}_{\mathcal{S}, l}^{\mathcal{P}}$ is the game defined in Figure 6 (left).

If \mathcal{P} queries \mathbf{x} with probability at least $\epsilon' / (2 \log M)$, then Theorem 21 gives

$$\epsilon = \mathcal{O}\left(\ell \log M^{\ell+1} \sqrt{\frac{q_D}{2^{k\ell}}} (\sigma + \ell N)\right), \quad (6)$$

<p>Game nbPred$_{\mathcal{S},j}^{\mathcal{P}}$:</p> <p>$H \leftarrow \text{Fun}(N^\ell, M)$</p> <p>for $i = 1$ to ℓ do $(x_i, z_i) \leftarrow S_i^H$</p> <p>$y \leftarrow H(\mathbf{x}); y'_j \leftarrow \mathcal{P}^H(y_1, \dots, y_{j-1}, z)$</p> <p>return $(y_j = y'_j)$</p>	<p>Game outPred$_{\mathcal{S},j}^{\mathcal{Q}}$:</p> <p>$H \leftarrow \text{Fun}(N^\ell, M)$</p> <p>for $i = 1$ to ℓ do $(x_i, z_i) \leftarrow S_i^H$</p> <p>$y \leftarrow H_j(\mathbf{x}); z \leftarrow \mathcal{Q}_0^H; y' \leftarrow \mathcal{Q}_1^{H_j}(\mathbf{x}, z, z)$</p> <p>return $(y = y')$</p>
--	---

■ **Figure 6** Definition of the j th next-bit prediction game, and of the output prediction game, both for a tuple of sources \mathcal{S} . In the latter game, \mathcal{Q}_1 is not allowed to query its input \mathbf{x} to its oracle, and H_j is the projection on the j th bit of H .

that is, our bound (5) is already satisfied in this case. So assume for the remainder of the proof that this is not the case. We then construct a (single-bit) predictor $\mathcal{Q} = (\mathcal{Q}_0, \mathcal{Q}_1)$ that tries to predict $H_l(\mathbf{x})$ (the l th output bit of $H(\mathbf{x})$) without querying its input. Algorithm \mathcal{Q}_0 returns a table containing, in lexicographical order of the inputs, all but the l th bit of the corresponding oracle outputs. This table is of size $N^\ell(\log M - 1)$. Algorithm \mathcal{Q}_1 is a $(q_{\mathcal{Q}} = q_{\mathcal{P}})$ -query predictor that runs \mathcal{P} on $(H(\mathbf{x})_1, \dots, H(\mathbf{x})_{l-1}, z)$, and combines the output of \mathcal{P} with the generated table z , such that

$$\Pr[\text{outPred}_{\mathcal{S},l}^{\mathcal{Q}}] \geq \frac{1}{2} + \epsilon',$$

where $\text{outPred}_{\mathcal{S},l}^{\mathcal{Q}}$ is the output prediction game defined in Figure 6 (right).

Call a pair $(\mathbf{O}, \mathbf{z}) \in \text{Fun}(N^\ell, M) \times \text{Supp}(\mathcal{S}_{1,2}^{\mathbf{O}}) \times \dots \times \text{Supp}(\mathcal{S}_{\ell,2}^{\mathbf{O}})$ (α, β) -good if:

1. $\Pr[\text{outPred}_{\mathcal{S},l}^{\mathcal{Q}} \mid H = \mathbf{O} \wedge \mathbf{z} = \mathbf{z}] \geq 1/2 + \alpha\epsilon'$, and
2. For every $i \in [\ell]$ and predictor \mathcal{P} with unbounded access to its oracle, it holds that $\Pr[\text{Pred}_{\mathcal{S}_i}^{\mathcal{P}} \mid H = \mathbf{O} \wedge \mathbf{z} = \mathbf{z}_i] \leq \beta \cdot 2^{-k}$.

In other words, a pair (\mathbf{O}, \mathbf{z}) as above is (α, β) -bad if either of the following conditions apply:

1. Pair (\mathbf{O}, \mathbf{z}) is low-win: $(\mathcal{S}, \mathcal{Q})$ succeeds with probability at most $1/2 + \alpha\epsilon'$ when (H, \mathbf{z}) is fixed to (\mathbf{O}, \mathbf{z}) , i.e., $\Pr[\text{outPred}_{\mathcal{S},l}^{\mathcal{Q}} \mid H = \mathbf{O} \wedge \mathbf{z} = \mathbf{z}] < 1/2 + \alpha\epsilon'$.
2. Pair (\mathbf{O}, \mathbf{z}) is low-entropy: There exist $i \in [\ell]$ and a predictor $\mathcal{P}_{\mathbf{O},\mathbf{z}}$ with unbounded oracle access such that $\Pr[\text{Pred}_{\mathcal{S}_i}^{\mathcal{P}_{\mathbf{O},\mathbf{z}}} \mid H = \mathbf{O} \wedge \mathbf{z} = \mathbf{z}_i] > \beta \cdot 2^{-k}$.

Denote by W_α the set of low-win pairs, $E_{\beta,i}$ set of pairs that are low-entropy for the i th source, and $E_\beta := \bigcup_{i=1}^\ell E_{\beta,i}$. Let $BP_{\alpha,\beta} := W_\alpha \cup E_\beta$ and $GP_{\alpha,\beta}$ be the sets of (α, β) -bad and (α, β) -good pairs, respectively. We start by bounding $p_{\text{bp}} := \Pr[(H, \mathbf{z}) \in BP_{\alpha,\beta}]$. By the union bound we have

$$p_{\text{bp}} \leq \Pr[(H, \mathbf{z}) \in W_\alpha] + \sum_{i=1}^\ell \Pr[(H, \mathbf{z}) \in E_{\beta,i}].$$

To bound $p_{\text{lw}} := \Pr[(H, \mathbf{z}) \in W_\alpha]$, notice that

$$\begin{aligned} \frac{1}{2} + \epsilon' &\leq \Pr[\text{outPred}_{\mathcal{S},l}^{\mathcal{Q}} \mid (H, \mathbf{z}) \in W_\alpha] \Pr[(H, \mathbf{z}) \in W_\alpha] \\ &\quad + \Pr[\text{outPred}_{\mathcal{S},l}^{\mathcal{Q}} \mid (H, \mathbf{z}) \notin W_\alpha] \Pr[(H, \mathbf{z}) \notin W_\alpha] < \left(\frac{1}{2} + \alpha\epsilon'\right) \cdot 1 + 1 \cdot (1 - p_{\text{lw}}), \end{aligned}$$

from which we obtain $p_{\text{lw}} < 1 - (1 - \alpha)\epsilon'$. To bound $p_{\text{le},i} := \Pr[(H, \mathbf{z}) \in E_{\beta,i}]$, let \mathcal{P} be the predictor that queries its oracle H in full and then runs $\mathcal{P}_{H,\hat{z}}$, where $\hat{z} := (0, \dots, z, \dots, 0)$, if $(H, \hat{z}) \in E_{\beta,i}$, and returns the constant 1 otherwise. For this choice of \mathcal{P} we have

$$\begin{aligned} 2^{-k} &\geq \Pr[\text{Pred}_{\mathcal{S}_i}^{\mathcal{P}}] \geq \Pr[\text{Pred}_{\mathcal{S}_i}^{\mathcal{P}} \mid (H, \hat{z}) \in E_{\beta,i}] \Pr[(H, \hat{z}) \in E_{\beta,i}] \\ &= \Pr[\text{Pred}_{\mathcal{S}_i}^{\mathcal{P}_{H,\hat{z}}} \mid (H, \hat{z}) \in E_{\beta,i}] \cdot p_{\text{le},i} \geq \beta \cdot 2^{-k} \cdot p_{\text{le},i}, \end{aligned}$$

from which we get $p_{le,i} < 1/\beta$ for every $i \in [\ell]$. Collecting all terms, we obtain a bound $p_{bp} < 1 - (1 - \alpha)\epsilon' + \ell/\beta$, and thus $p_{gp} = 1 - p_{bp} > (1 - \alpha)\epsilon' - \ell/\beta$.

To ensure that $GP_{\alpha,\beta} \neq \emptyset$, we must pick α and β so that $(1 - \alpha)\epsilon' - \ell/\beta > 0$. In particular, we can choose $0 < \alpha < 1$ to be a constant, and $\beta > \ell/((1 - \alpha)\epsilon')$, i.e., $\beta = b\ell/((1 - \alpha)\epsilon')$ for some constant $b > 1$. By at most doubling β if necessary, we can choose β so that additionally $2^{k-\log \beta} \in \mathbb{N}$ (as required by Lemma 3).

Now let $G_{\alpha,\beta}$ be the set of (α, β) -good oracles, defined as the $\mathbf{O} \in \text{Fun}(N^\ell, M)$ such that $(\mathbf{O}, \mathbf{z}) \in GP_{\alpha,\beta}$ for some leakage vector \mathbf{z} . For every $\mathbf{O} \in G_{\alpha,\beta}$, fix any leakage vector $\mathbf{z}_{\mathbf{O}}$ such that $(\mathbf{O}, \mathbf{z}_{\mathbf{O}}) \in GP_{\alpha,\beta}$. Then

$$p_{go} := \Pr[H \in G_{\alpha,\beta}] \geq \Pr[(H, \mathbf{z}) \in GP_{\alpha,\beta}] = p_{gp},$$

meaning that

$$|G_{\alpha,\beta}| = p_{go} \cdot |\text{Fun}(N^\ell, M)| \geq \left((1 - \alpha)\epsilon' - \frac{\ell}{\beta} \right) \cdot M^{N^\ell} > 0.$$

For each (α, β) -good \mathbf{O} and every $i \in [\ell]$, let $\mathcal{X}_{\mathbf{O},i}$ be the conditional distribution of $\mathcal{S}_i^{\mathbf{O}}$ on $[N]$ given that $z = z_{\mathbf{O},i}$. We can then use Lemma 3 to decompose each $\mathcal{X}_{\mathbf{O},i}$ into a convex combination of flat sources $\mathcal{X}_{\mathbf{O},i,j}$, each having support size *exactly* $S := 2^{k-\log \beta}$. By an averaging argument, there exists a flat source $\mathcal{X}_{\mathbf{O},i}^*$ within each convex combination such that

$$\Pr[\text{outPred}_{\mathcal{X}_{\mathbf{O},i}^*}^{\mathbf{O}} \mid H = \mathbf{O} \wedge \mathbf{z} = \mathbf{z}_{\mathbf{O}}] \geq \frac{1}{2} + \alpha\epsilon'.$$

Before coming to the actual compression argument, we distinguish four cases: If $S < 40$, then $2^k/\beta < 40$, and substituting for β and therein for ϵ' we obtain

$$\epsilon < \log M \cdot \frac{40b\ell}{(1 - \alpha)2^k} = \mathcal{O}\left(\frac{\ell \log M}{2^k}\right).$$

Similarly, if $q_{\mathcal{D}} > S^\ell / \log S^\ell$, then $q_{\mathcal{D}} > 2^{k\ell}/(\beta^\ell \ell(k - \log \beta)) > 2^{k\ell}/(\beta^\ell \ell k)$, so that after the appropriate substitutions we get

$$\epsilon < \sqrt[\ell]{\frac{(\log M)^\ell q_{\mathcal{D}} b^\ell \ell^{\ell+1} k}{2^{k\ell}(1 - \alpha)^\ell}} = \mathcal{O}\left(\ell \log M \sqrt[\ell]{\frac{q_{\mathcal{D}} \ell k}{2^{k\ell}}}\right) = \mathcal{O}\left(\ell \log M \sqrt[\ell]{\frac{q_{\mathcal{D}} \ell N}{2^{k\ell}}}\right).$$

Finally, if $q_{\mathcal{D}} > \epsilon' S^\ell$, then again

$$\epsilon < \sqrt[\ell+1]{\frac{(\log M)^{\ell+1} q_{\mathcal{D}} b^\ell \ell^\ell}{(1 - \alpha)^\ell 2^{k\ell}}} = \mathcal{O}\left(\ell \log M \sqrt[\ell+1]{\frac{q_{\mathcal{D}}}{2^{k\ell}}}\right).$$

This shows that in all cases above, ϵ already satisfies our bound (5), so for the remainder of the analysis assume that $S \geq 40$, that $q_{\mathcal{D}} \leq S^\ell / \log S^\ell$, and that $q_{\mathcal{D}} \leq \epsilon' S^\ell$. Note that this clears all the “trivial” cases from Lemma 22.

We now define our randomized encoding of $G_{\alpha,\beta}$ as follows:

Encoding: Given $\mathbf{O} \in G_{\alpha,\beta}$, we encode it as follows:

- Generate a table containing, in lexicographical order of \mathbf{x} , all but the l th bit of the oracle outputs $\mathbf{O}(\mathbf{x})$, taking $N^\ell(\log M - 1)$ bits;
- Apply the encoding algorithm from Lemma 22 to obtain a randomized encoding of \mathbf{O}_j , which has an encoding length of at most $\sigma + 2\ell S \log(eN/S) + N^\ell - \Omega((\alpha\epsilon/\log M)^2 \cdot S^\ell/q_A) + \mathcal{O}(1)$ many bits.

Decoding: Given a codeword as above, we decode it as follows:

- Initialize an empty table for \mathcal{O} ;
- Extract the table of all but the l th bit of outputs from the codeword and use this to fill the oracle table;
- Follow the decoding algorithm in Lemma 22 to extract the l th bit of each oracle output.

From Lemma 22 we also know that the recovery probability of this encoding scheme is at least $\delta = \Omega(\alpha\epsilon/q_{\mathcal{D}} \log M)$.

Adding up the lengths of the different parts in the encoding, and applying Lemma 2 with δ as above gives

$$\begin{aligned} N^\ell(\log M - 1) + \sigma + 2\ell S \log\left(\frac{eN}{S}\right) + N^\ell - \Omega\left(\left(\frac{\alpha\epsilon}{\log M}\right)^2 \cdot \frac{S^\ell}{q_{\mathcal{D}}}\right) + \mathcal{O}(1) \\ \geq \log\left(\left((1-\alpha)\epsilon' - \frac{\ell}{\beta}\right) \cdot M^{N^\ell}\right) + \log\left(\frac{\alpha\epsilon}{q_{\mathcal{D}} \log M}\right) \\ = N^\ell \log M + \log\left((1-\alpha)\epsilon' - \frac{\ell}{\beta}\right) + \log\left(\frac{\alpha\epsilon}{q_{\mathcal{D}} \log M}\right). \end{aligned}$$

We now distinguish two more cases: If $\epsilon \leq q_{\mathcal{D}} \log M / (\alpha N^\ell)$ then again we are done, since we obtain a bound that verifies (5). Indeed,

$$\epsilon \leq \frac{q_{\mathcal{D}} \log M}{\alpha N^\ell} \leq \frac{q_{\mathcal{D}} \log M}{\alpha S^\ell} = \frac{q_{\mathcal{D}}(\log M)^{\ell+1} b^\ell \ell^\ell}{\alpha 2^{k\ell} (1-\alpha)^\ell \epsilon^\ell},$$

giving

$$\epsilon = \mathcal{O}\left(\ell \log M \sqrt[\ell+1]{\frac{q_{\mathcal{D}}}{2^{k\ell}}}\right).$$

If the above is not the case then, rearranging, we have

$$\begin{aligned} \Omega\left(\left(\frac{\alpha\epsilon}{\log M}\right)^2 \cdot \frac{S^\ell}{q_{\mathcal{D}}}\right) &\leq \sigma + 2\ell S \log\left(\frac{eN}{S}\right) + \log\left(\frac{q_{\mathcal{D}} \log M}{\alpha\epsilon}\right) \\ &\quad - \log\left((1-\alpha)\epsilon' - \frac{\ell}{\beta}\right) + \mathcal{O}(1) \\ &\leq \sigma + 2\ell N \log e + \mathcal{O}(\ell \log N) - \log\left((1-\alpha)\epsilon' - \frac{\ell}{\beta}\right) \\ &= \sigma + \mathcal{O}(\ell N) - \log\left((1-\alpha)\epsilon' - \frac{\ell}{\beta}\right) = \mathcal{O}(\sigma + \ell N) - \log\left((1-\alpha)\epsilon' - \frac{\ell}{\beta}\right), \end{aligned}$$

where the inequality uses the fact that $x \mapsto x \log(eN/x)$ has a maximum at $x = N$, where it takes the value $N \log e$. Now recall that $S = 2^k/\beta$ and $\beta = b\ell/(1-\alpha)\epsilon'$, giving

$$\begin{aligned} \Omega\left(\frac{\epsilon^2}{(\log M)^2}\right) &= \mathcal{O}\left(\frac{q_{\mathcal{D}}}{S^\ell}(\sigma + \ell N)\right) - \frac{q_{\mathcal{D}}}{S^\ell} \log\left((1-\alpha)\epsilon' - \frac{\ell}{\beta}\right) \\ &= \mathcal{O}\left(\frac{q_{\mathcal{D}} \beta^\ell}{2^{k\ell}}(\sigma + \ell N)\right) - \frac{q_{\mathcal{D}} \beta^\ell}{2^{k\ell}} \log\left((1-\alpha)\epsilon' - \frac{\ell}{\beta}\right) \\ &\leq \mathcal{O}\left(\frac{q_{\mathcal{D}} (b\ell)^\ell}{2^{k\ell} ((1-\alpha)\epsilon')^\ell}(\sigma + \ell N)\right) - \frac{q_{\mathcal{D}} (b\ell)^\ell}{2^{k\ell} ((1-\alpha)\epsilon')^\ell} \log\left((1-\alpha)\epsilon' - \frac{(1-\alpha)\epsilon'}{b}\right) \\ &\leq \mathcal{O}\left(\frac{q_{\mathcal{D}} (b\ell)^\ell}{2^{k\ell} ((1-\alpha)\epsilon')^\ell}(\sigma + \ell N)\right) - \frac{q_{\mathcal{D}} (b\ell)^\ell}{2^{k\ell} ((1-\alpha)\epsilon')^\ell} \log\left(\frac{(1-\alpha)\epsilon'(b-1)}{b}\right). \end{aligned}$$

Hence,

$$\Omega\left(\frac{\epsilon^{\ell+2}}{(\log M)^{\ell+2}}\right) \leq \mathcal{O}\left(\frac{q_{\mathcal{D}}(b\ell)^\ell}{2^{k\ell}(1-\alpha)^\ell}(\sigma + \ell N)\right) + \mathcal{O}\left(\frac{q_{\mathcal{D}}(b\ell)^\ell}{2^{k\ell}(1-\alpha)^\ell} \log\left(\frac{\log M}{\epsilon}\right)\right).$$

We now once more distinguish two cases: If $\log M/\epsilon > N^\ell$, then $\epsilon < \log M/N^\ell$, and again our bound (5) holds. Otherwise, we obtain

$$\begin{aligned} \Omega\left(\frac{\epsilon^{\ell+2}}{(\log M)^{\ell+2}}\right) &\leq \mathcal{O}\left(\frac{q_{\mathcal{D}}(b\ell)^\ell}{2^{k\ell}(1-\alpha)^\ell}(\sigma + \ell N)\right) + \mathcal{O}\left(\frac{q_{\mathcal{D}}(b\ell)^\ell}{2^{k\ell}(1-\alpha)^\ell} \log(N^\ell)\right) \\ &\leq \mathcal{O}\left(\frac{q_{\mathcal{D}}(b\ell)^\ell}{2^{k\ell}(1-\alpha)^\ell}(\sigma + \ell N)\right), \end{aligned}$$

which gives

$$\epsilon^{\ell+2} \leq \mathcal{O}\left(\frac{q_{\mathcal{D}}(\log M)^{\ell+2}(b\ell)^{\ell+2}}{2^{k\ell}(1-\alpha)^{\ell+2}}(\sigma + \ell N)\right),$$

and thus

$$\begin{aligned} \epsilon &\leq \mathcal{O}\left(\sqrt[\ell+2]{\frac{q_{\mathcal{D}}(\log M)^{\ell+2}(b\ell)^{\ell+2}}{2^{k\ell}(1-\alpha)^{\ell+2}}(\sigma + \ell N)}\right) \leq \mathcal{O}\left(\frac{b\ell \log M}{1-\alpha} \sqrt[\ell+2]{\frac{q_{\mathcal{D}}}{2^{k\ell}}(\sigma + \ell N)}\right) \\ &\leq \mathcal{O}\left(\ell \log M \sqrt[\ell+2]{\frac{q_{\mathcal{D}}}{2^{k\ell}}(\sigma + \ell N)}\right). \end{aligned}$$

This concludes the proof. \blacktriangleleft

► **Remark 24.** Compared to the work of Dodis and Oliveira [17], note that our MSE theorem allows extracting many more bits (akin to a PRG security result), which cannot hold in the statistical setting of [17]. This allows directly generating random bits from multiple entropy sources, without further processing, using a standard hash function, as opposed to ad-hoc constructions.

As a contribution of independent interest, we show in the full version of our paper [12] that Mono is one-way in the UBU setting with respect to split sources that fully depend on the random oracle. Note that the split structure is necessary for one-wayness, as otherwise the source can output a fixed point of the hash function. And indeed, the bound that we achieve is meaningful only for $\ell \geq 2$.

References

- 1 Divesh Aggarwal, Maciej Obremski, João L. Ribeiro, Luisa Siniscalchi, and Ivan Visconti. How to extract useful randomness from unreliable sources. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 343–372. Springer, Cham, May 2020. doi:10.1007/978-3-030-45721-1_13.
- 2 Boaz Barak, Russell Impagliazzo, and Avi Wigderson. Extracting randomness using few independent sources. In *45th FOCS*, pages 384–393. IEEE Computer Society Press, October 2004. doi:10.1109/FOCS.2004.29.
- 3 Salman Beigi, Omid Etesami, and Amin Gohari. Deterministic randomness extraction from generalized and distributed Santha–Vazirani sources. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *ICALP 2015, Part I*, volume 9134 of *LNCS*, pages 143–154. Springer, Berlin, Heidelberg, July 2015. doi:10.1007/978-3-662-47672-7_12.

- 4 Mihir Bellare, Viet Tung Hoang, and Sriram Keelveedhi. Instantiating random oracles via UCEs. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 398–415. Springer, Berlin, Heidelberg, August 2013. doi:10.1007/978-3-642-40084-1_23.
- 5 Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993. doi:10.1145/168588.168596.
- 6 Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, Berlin, Heidelberg, May / June 2006. doi:10.1007/11761679_25.
- 7 Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge-based pseudo-random number generators. In Stefan Mangard and François-Xavier Standaert, editors, *CHES 2010*, volume 6225 of *LNCS*, pages 33–47. Springer, Berlin, Heidelberg, August 2010. doi:10.1007/978-3-642-15031-9_3.
- 8 Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. In Daniel Wichs and Yishay Mansour, editors, *48th ACM STOC*, pages 670–683. ACM Press, June 2016. doi:10.1145/2897518.2897528.
- 9 Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity (extended abstract). In *26th FOCS*, pages 429–442. IEEE Computer Society Press, October 1985. doi:10.1109/SFCS.1985.62.
- 10 Sandro Coretti, Yevgeniy Dodis, Siyao Guo, and John P. Steinberger. Random oracles and non-uniformity. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 227–258. Springer, Cham, April / May 2018. doi:10.1007/978-3-319-78381-9_9.
- 11 Sandro Coretti, Yevgeniy Dodis, Harish Karthikeyan, and Stefano Tessaro. Seedless fruit is the sweetest: Random number generation, revisited. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 205–234. Springer, Cham, August 2019. doi:10.1007/978-3-030-26948-7_8.
- 12 Sandro Coretti, Pooya Farshim, Patrick Harasser, and Karl Southern. Multi-source randomness extraction and generation in the random-oracle model. Cryptology ePrint Archive, Report 2025/1258, 2025. URL: <https://eprint.iacr.org/2025/1258>.
- 13 Anindya De, Luca Trevisan, and Madhur Tulsiani. Time space tradeoffs for attacks against one-way functions and PRGs. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 649–665. Springer, Berlin, Heidelberg, August 2010. doi:10.1007/978-3-642-14623-7_35.
- 14 Yevgeniy Dodis, Ariel Elbaz, Roberto Oliveira, and Ran Raz. Improved randomness extraction from two independent sources. In Dana Ron, editor, *RANDOM 04*, volume 3122 of *LNCS*, pages 334–344. Springer, August 2004. doi:10.1007/978-3-540-27821-4_30.
- 15 Yevgeniy Dodis, Rosario Gennaro, Johan Håstad, Hugo Krawczyk, and Tal Rabin. Randomness extraction and key derivation using the CBC, cascade and HMAC modes. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 494–510. Springer, Berlin, Heidelberg, August 2004. doi:10.1007/978-3-540-28628-8_30.
- 16 Yevgeniy Dodis, Siyao Guo, and Jonathan Katz. Fixing cracks in the concrete: Random oracles with auxiliary input, revisited. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, pages 473–495. Springer, Cham, April / May 2017. doi:10.1007/978-3-319-56614-6_16.
- 17 Yevgeniy Dodis and Roberto Oliveira. On extracting private randomness over a public channel. In Amit Sahai, editor, *RANDOM 2003*, volume 2764 of *LNCS*, pages 252–263. Springer, August 2003. doi:10.1007/978-3-540-45198-3_22.

- 18 Yevgeniy Dodis, Shien Jin Ong, Manoj Prabhakaran, and Amit Sahai. On the (im)possibility of cryptography with imperfect randomness. In *45th FOCS*, pages 196–205. IEEE Computer Society Press, October 2004. doi:10.1109/FOCS.2004.44.
- 19 Yevgeniy Dodis, Thomas Ristenpart, and Salil P. Vadhan. Randomness condensers for efficiently samplable, seed-dependent sources. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 618–635. Springer, Berlin, Heidelberg, March 2012. doi:10.1007/978-3-642-28914-9_35.
- 20 Yevgeniy Dodis, Adi Shamir, Noah Stephens-Davidowitz, and Daniel Wichs. How to eat your entropy and have it too – Optimal recovery strategies for compromised RNGs. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 37–54. Springer, Berlin, Heidelberg, August 2014. doi:10.1007/978-3-662-44381-1_3.
- 21 Yevgeniy Dodis and Joel Spencer. On the (non)universality of the one-time pad. In *43rd FOCS*, pages 376–387. IEEE Computer Society Press, November 2002. doi:10.1109/SFCS.2002.1181962.
- 22 Yevgeniy Dodis, Vinod Vaikuntanathan, and Daniel Wichs. Extracting randomness from extractor-dependent sources. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 313–342. Springer, Cham, May 2020. doi:10.1007/978-3-030-45721-1_12.
- 23 Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO’86*, volume 263 of *LNCS*, pages 186–194. Springer, Berlin, Heidelberg, August 1987. doi:10.1007/3-540-47721-7_12.
- 24 Peter Gazi, Krzysztof Pietrzak, and Stefano Tessaro. The exact PRF security of truncation: Tight bounds for keyed sponges and truncated CBC. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 368–387. Springer, Berlin, Heidelberg, August 2015. doi:10.1007/978-3-662-47989-6_18.
- 25 Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st FOCS*, pages 305–313. IEEE Computer Society Press, November 2000. doi:10.1109/SFCS.2000.892119.
- 26 Intel Corporation. *Intel Digital Random Number Generator (DRNG) Software Implementation Guide*, 2019. URL: <https://www.intel.com/content/www/us/en/developer/articles/guide/intel-digital-random-number-generator-drng-software-implementation-guide.html>.
- 27 Xin Li. Three-source extractors for polylogarithmic min-entropy. In Venkatesan Guruswami, editor, *56th FOCS*, pages 863–882. IEEE Computer Society Press, October 2015. doi:10.1109/FOCS.2015.58.
- 28 James L. McInnes and Benny Pinkas. On the impossibility of private key cryptography with weakly random keys. In Alfred J. Menezes and Scott A. Vanstone, editors, *CRYPTO’90*, volume 537 of *LNCS*, pages 421–435. Springer, Berlin, Heidelberg, August 1991. doi:10.1007/3-540-38424-3_31.
- 29 Noam Nisan and David Zuckerman. Randomness is linear in space. *J. Comput. Syst. Sci.*, 52(1):43–52, 1996. doi:10.1006/JCSS.1996.0004.
- 30 Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with composition: Limitations of the indistinguishability framework. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 487–506. Springer, Berlin, Heidelberg, May 2011. doi:10.1007/978-3-642-20465-4_27.
- 31 Miklos Santha and Umesh V. Vazirani. Generating quasi-random sequences from slightly-random sources (extended abstract). In *25th FOCS*, pages 434–440. IEEE Computer Society Press, October 1984. doi:10.1109/SFCS.1984.715945.
- 32 Thomas Shrimpton and R. Seth Terashima. A provable-security analysis of Intel’s secure key RNG. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 77–100. Springer, Berlin, Heidelberg, April 2015. doi:10.1007/978-3-662-46800-5_4.

- 33 Dominique Unruh. Random oracles and auxiliary input. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 205–223. Springer, Berlin, Heidelberg, August 2007. doi:10.1007/978-3-540-74143-5_12.
- 34 Salil P. Vadhan. *Pseudorandomness*, volume 7 of *Foundations and Trends in Theoretical Computer Science*. Now Publishers, Boston-Delft, 2012. doi:10.1561/04000000010.
- 35 Joanne Woodage and Dan Shumow. An analysis of NIST SP 800-90A. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 151–180. Springer, Cham, May 2019. doi:10.1007/978-3-030-17656-3_6.
- 36 Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd FOCS*, pages 80–91. IEEE Computer Society Press, November 1982. doi:10.1109/SFCS.1982.45.