A preliminary version of this work appears in the proceedings of the *The Sixth Information-Theoretic Cryptography (ITC 2025)* Conference. This is the full version.

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

Sandro Coretti[1], Pooya Farshim[2], Patrick Harasser[3], and Karl Southern[4]

[1] IOG, France sandro.coretti@iohk.io
[2] IOG, Switzerland and Durham University, UK pooya.farshim@gmail.com
[3] Technische Universität Darmstadt, Germany patrick.harasser@tu-darmstadt.de
[4] Durham University, UK karl.southern@durham.ac.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 arbitrary 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.

Keywords: Multi-source randomness extraction · Multi-source randomness generation · Compression argument · Convex decomposition

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 [17, 20, 27]. However, while "nature" provides many sources that contain some amounts of entropy (e.g., those 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 [28], 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 [18]. 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: (a) Modeling the entire extractor as a monolithic random oracle; (b) Merkle–Damgård-based constructions where the compression function is treated as a random function; (c) Using the Davies–Meyer compression function with the block cipher modeled as an ideal cipher; (d) 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 [7, 14, 31, 34]). 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 [19,23]). 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, 8, 13, 26]. 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 [25]. 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: (a) 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; (b) To measure the entropy (or unpredictability) of

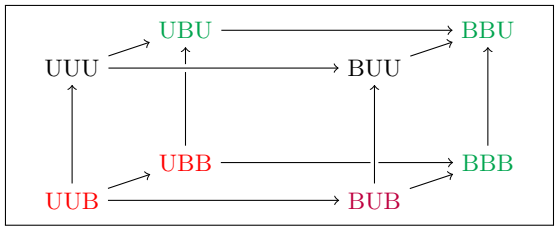

Fig. 1. Multi-source extractor notions in the random-oracle model. Notions are labeled via an "SDP" string (for *S*ource, *D*istinguisher, and *P*redictor), with $S, D, P \in \{\mathrm{B}, \mathrm{U}\}$ depending on whether the corresponding party has bounded-query (B) or unbounded (U) access to the random oracle. An arrow from X to Y means that any construction for X is also a construction for Y. Corners typeset in red or purple are shown not to be achieved by the monolithic construction via attacks presented in Section 4, specifically Proposition 3 (UBB) and Proposition 4 (BUB). Corners typeset in green are achieved in Sections 5 and 6, specifically Theorem 1 (BBB) and Theorem 3 (UBU).

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; (c) 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 Fig. 1). We establish a complete set of generic implications between these notions.

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 1). 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 3).
(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 3), 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 4).

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, namely, that the random oracle itself acts as a multi-source extractor in the UBU setting, to the analogous result 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 under the additional restriction 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, as introduced by Gennaro and Trevisan [24].

Our proof strategy follows that of Dodis et al. [15], which in turn is based on De et al. [12]. We begin by using a compression argument to establish that a random oracle, when applied to inputs $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\ell)$ produced by independent sources $\mathcal{S}_1(H), \ldots, \mathcal{S}_\ell(H)$, satisfies the property of unrecoverability: It is computationally infeasible to recover the input $\boldsymbol{x}$ from the output $H(\boldsymbol{x})$, even given the leakage of the (unpredictable) sources. The proof then transitions to reducing distinguishing $H(\boldsymbol{x})$ from a uniformly 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 $\boldsymbol{x}_i$ are not uniformly random and independent of the oracle, as is the case in [15]. 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 [33] 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: (a) Dodis et al. [21], who investigate extracting randomness from extractor-dependent sources that are capable of making black-box queries to the seeded extractor; (b) 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 [30], but more recent work has shown that extraction is feasible for non-binary SV sources [3]; (c) 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 [16] uses the probabilistic method to show the existence of so-called super-strong extractors (and then provides explicit constructions thereof).[5] 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.

1.4 Future Directions

Our work shows that compression is sufficiently powerful to establish meaningful 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 practical non-monolithic constructions such as Merkle–Damgård or Sponge.[6] 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 [32] 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)

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

[6] 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 a single-stage game [29].

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.[7] Indeed, consider sources $\mathcal{S}_1$ and $\mathcal{S}_2$ that leak $z_1 \coloneqq \sum_{1 \neq x \in [N]} H(x)$ and $z_2 \coloneqq \sum_{x \in [N]} H(x)$. 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 $H(1)$.[8]

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 the notions where the predictor is not as powerful as either the source or the distinguisher. Section 5 contains our security result for distinguishes 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. In Appendix A, we prove that the random oracle is one-way under fully random-oracle dependent challenges generated via multiple sources.

[7] 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.

[8] 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.

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 $[n] \coloneqq \{1, \ldots, n\}$. 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 $x \twoheadleftarrow \mathcal{X}$; when $\mathcal{X}$ is uniformly distributed over a finite set X, we write $x \twoheadleftarrow X$ instead. The support of a random variable $\mathcal{X}$ is denoted $\mathrm{Supp}(\mathcal{X})$. For $\ell, M, N \in \mathbb{N}$, we write $\mathrm{Fun}(N^\ell, M)$ for the set of functions from $[N]^\ell$ to $[M]$. We use boldface notation for vectors, as in $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\ell)$, and we write $|\boldsymbol{x}|$ for the length of the vector $\boldsymbol{x}$. If not stated otherwise, indexes in vectors and strings start from 1. All logarithms are to base 2.

ROM [5, 22]. Let $\ell, M, N \in \mathbb{N}$. We define the random-oracle model $\mathcal{RO}[\ell, N, M]$ as the uniform distribution on $\mathrm{Fun}(N^\ell, M)$. When operating in this model, all algorithms (both honest and adversarial) can query a function $H \in \mathrm{Fun}(N^\ell, M)$ as an oracle. In security games, one first samples H uniformly from $\mathrm{Fun}(N^\ell, M)$. 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 $|\Pr[\mathsf{G}_1] - \Pr[\mathsf{G}_2]| \leq \Pr[\mathsf{Bad}]$.

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

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

Lemma 1 (Deterministic compression [12, 15]). *Let S be a set, and let $(\mathsf{Enc}, \mathsf{Dec})$ be a code for S such that, for every $m \in S$, $\mathsf{Dec}(\mathsf{Enc}(m)) = m$. Then*

$$\mathop{\mathbb{E}}_{m \twoheadleftarrow S}[|\mathsf{Enc}(m)|] \geq \log |S| \,.$$

Lemma 2 (Randomized compression [12, 15]). *Let S be a set, $\delta \in \mathbb{R}_{>0}$, and let $(\mathsf{Enc}, \mathsf{Dec})$ be a randomized encoding for S with recovery probability δ, meaning that for every $m \in S$,*

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

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 [33]). *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. In Appendix A, we complete the picture by presenting the classical notion of one-wayness (OW) in our framework.

SOURCES, ADVERSARIES, DISTINGUISHERS, AND PREDICTORS. Let $\ell, M, N \in \mathbb{N}$. A source is an algorithm $\mathcal{S}$ that takes no input and returns a pair $(x, z) \in [N] \times \{0,1\}^*$. An adversary is an algorithm $\mathcal{A}$ which, on input $y \in [M]$ and a vector $\boldsymbol{z} \in (\{0,1\}^*)^\ell$ of strings, returns $\boldsymbol{x}' \in [N]^\ell$. Likewise, a distinguisher is an algorithm $\mathcal{D}$ which, on input $y \in [M]$ and a vector $\boldsymbol{z} \in (\{0,1\}^*)^\ell$, returns a bit. Finally, we let a predictor be 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 $H \in \mathrm{Fun}(N^\ell, M)$.

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 $\mathrm{Adv}^{\mathrm{pred}}_{\mathcal{S},\mathcal{P}} := \Pr[\mathrm{Pred}^{\mathcal{P}}_{\mathcal{S}}]$, where the game $\mathrm{Pred}^{\mathcal{P}}_{\mathcal{S}}$ is given in Fig. 2 (left). 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 $\mathrm{Adv}^{\mathrm{pred}}_{\mathcal{S},\mathcal{P}} \leq \delta$.

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

Fig. 2. Left: The prediction game for a source $\mathcal{S}$ played by a predictor $\mathcal{P}$. **Center and right**: The unrecoverability and multi-source-extractor games for the monolithic construction Mono. In both games, $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$ is a tuple of sources, and $\mathcal{A}$ (resp., $\mathcal{D}$) is an adversary (resp., a distinguisher).

Remark. 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 [16], 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., [4, 11]. 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.

Remark. 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.

Unrecoverability. Let $\ell, M, N \in \mathbb{N}$, $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$ 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 $\mathrm{Adv}^{\mathrm{ur}}_{\mathcal{S},\mathcal{A}} \coloneqq \Pr[\mathrm{UR}^{\mathcal{S},\mathcal{A}}]$, where the game $\mathrm{UR}^{\mathcal{S},\mathcal{A}}$ is given in Fig. 2 (center). 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 $\mathrm{Adv}^{\mathrm{ur}}_{\mathcal{S},\mathcal{A}} \leq \epsilon$.

Multi-source extraction. Let $\ell, M, N \in \mathbb{N}$, $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$ 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 $\mathrm{Adv}^{\mathrm{mse}}_{\mathcal{S},\mathcal{D}} \coloneqq 2 \cdot \Pr[\mathrm{MSE}^{\mathcal{S},\mathcal{D}}] - 1$, where the game $\mathrm{MSE}^{\mathcal{S},\mathcal{D}}$ is given in Fig. 2 (right). 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 $\mathrm{Adv}^{\mathrm{mse}}_{\mathcal{S},\mathcal{D}} \leq \epsilon$.

REMARK. 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 to the relevant parties), so different techniques (e.g., along the lines of [15]) are necessary in this setting.

REMARK. In this work, we opt for a concrete treatment of security, but a formulation in the asymptotic framework is also possible. In that setting, all idealized models, parties, and security notions depend on a security parameter λ. For security to hold, the 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. Let $\ell, M, N \in \mathbb{N}$, $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$ 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

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

where $\mathcal{A}[r]$ denotes the deterministic adversary $\mathcal{A}$ with hard-coded randomness r. This means that there must exist r such that $\mathrm{Adv}^{\mathrm{ur}}_{\mathcal{S},\mathcal{A}} \leq \mathrm{Adv}^{\mathrm{ur}}_{\mathcal{S},\mathcal{A}[r]}$, showing that for every (possibly randomized) adversary $\mathcal{A}$, there exists a deterministic adversary $\mathcal{B}$ such that $\mathrm{Adv}^{\mathrm{ur}}_{\mathcal{S},\mathcal{A}} \leq \mathrm{Adv}^{\mathrm{ur}}_{\mathcal{S},\mathcal{B}}$. 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 $\mathrm{SEC} \in \{\mathrm{UR}, \mathrm{MSE}\}$. 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 1. *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 $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$ 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 $\mathrm{Adv}^{\mathrm{ur}}_{\mathcal{S},\mathcal{A}} \leq \epsilon \leq \epsilon'$. 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 2. *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 $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$ 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 $(y, \boldsymbol{z})$ to get $\boldsymbol{x}'$, and then returns the bit $(H(\boldsymbol{x}') = y)$. Then clearly $\mathcal{D}$ makes at most $q_{\mathcal{D}}$ oracle calls. Furthermore notice that, when $b = 1$ in $\mathrm{MSE}^{\mathcal{S},\mathcal{D}}$, distinguisher $\mathcal{D}$ will return 1 if $\mathcal{A}$ succeeds in finding a preimage, so that $\Pr[\mathrm{MSE}^{\mathcal{S},\mathcal{D}} \mid b = 1] \geq \mathrm{Adv}^{\mathrm{ur}}_{\mathcal{S},\mathcal{A}}$. On the other hand, when $b = 0$, we have:

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

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

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

Rearranging this inequality, we obtain the claimed bound. □

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

4 Attacks

In this section, we present attacks against the configurations UBB (Proposition 3) and BUB (Proposition 4) in Fig. 1. The former result also rules out UUB, which implies UBB.

Proposition 3. *Let $M, N, q_{\mathcal{P}} \in \mathbb{N}$ with $M \leq N$, and set $\ell \coloneqq 2$. 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*

$$\mathrm{Adv}^{\mathrm{mse}}_{\mathcal{S},\mathcal{D}} \geq 1 - \frac{1}{e} - \frac{1}{M}\,. \tag{1}$$

Src. $\mathcal{S}_i^H$:	Game G_1:
$E_H \leftarrow \{x \in [M] \subseteq [N] \mid H(x,x) = x\}$	$H \twoheadleftarrow E$
if $(E_H \neq \emptyset)$ then $x \leftarrow \min E_H$ else $x \twoheadleftarrow [M]$	$x' \twoheadleftarrow \mathcal{P}^H(\varepsilon)$
$z \leftarrow \varepsilon$; return (x, z)	return $(H(x', x') = x')$
Dist. $\mathcal{D}^H(y, \boldsymbol{z})$:	
return $(H(y, y) = y)$	

Fig. 3. Left: Definition of sources $\mathcal{S}_i$, $i \in [2]$, and distinguisher $\mathcal{D}$ from the proof of Proposition 3. **Right**: Definition of the game G_1 from the proof of Proposition 3.

Proof. For every $i \in [2]$, let $\mathcal{S}_i$ be the source defined in Fig. 3 (left), 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 Fig. 3 (left), 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

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

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

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

where G_1 is the game defined in Fig. 3 (right). Here, the last inequality holds because the uniform distribution on $\mathrm{Fun}(N^2, M)$, conditioned on the outcome being in E, is the uniform distribution on E, and because the winning condition in G_1 is less restrictive than the one in $\mathrm{Pred}^{\mathcal{P}}_{\mathcal{S}_i}$, given that $H \in E$. Indeed, in $\mathrm{Pred}^{\mathcal{P}}_{\mathcal{S}_i}$, predictor $\mathcal{P}$ must guess the exact output x of $\mathcal{S}_i$, which satisfies $H(x,x) = x$, whereas in G_1 it wins if it returns any such value $x' \in [N]$.

To bound $\Pr[\mathsf{G}_1]$, we transition to yet another game G_2 that proceeds as 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 G_1. Then again $\Pr[\mathsf{G}_1] \leq \Pr[\mathsf{G}_2]$, because $\mathcal{P}$ will always win G_2 if it wins G_1. Now notice that, by definition, predictor $\mathcal{P}$ will lose game 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

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

Src. $\mathcal{S}_i^H$:
$x \twoheadleftarrow [N]$; $z \leftarrow H(x,x)$
return (x,z)
 	
Dist. $\mathcal{D}^H(y,\boldsymbol{z})$:
for $i = 1$ to 2 do
  $X_i \leftarrow \{x \in [N] \mid H(x,x) = \boldsymbol{z}_i\}$
return $(\exists x_1 \in X_1)(\exists x_2 \in X_2)$
  $(H(x_1,x_2) = y)$
	Game G_1:
	$T \leftarrow \emptyset$; $z \twoheadleftarrow [M]$; $x' \twoheadleftarrow \mathcal{P}^H(z)$; $x \twoheadleftarrow [N]$
	if $((x,x) \in \mathrm{Dom}(T))$ then return 1
	$T[(x,x)] \leftarrow z$; return $(x = x')$
	
	Game G_2:
	$T \leftarrow \emptyset$; $z \twoheadleftarrow [M]$; $x' \twoheadleftarrow \mathcal{P}^H(z)$; $x \twoheadleftarrow [N]$
	return $(x = x')$
	
	Proc. $H(u,v)$:
	if $((u,v) \notin \mathrm{Dom}(T))$ then
	  $y \twoheadleftarrow [M]$; $T[(u,v)] \leftarrow y$
	return $T[(u,v)]$

Fig. 4. Left: Definition of sources $\mathcal{S}_i$, $i \in [2]$, and distinguisher $\mathcal{D}$ from the proof of Proposition 4. **Right**: Definition of the games G_1 and G_2 from the proof of Proposition 4.

which means

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

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 for Mono 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,

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

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

Proposition 4. *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*

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

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

Proof. For every $i \in [2]$, let $\mathcal{S}_i$ be the source defined in Fig. 4 (left), which returns a random $x \in [N]$ and $z = H(x, x)$ as leakage. Furthermore, let $\mathcal{D}$ be the distinguisher defined in Fig. 4 (left), which finds, for each $i \in [2]$, all inputs of the form (x_i, x_i) satisfying $H(x_i, x_i) = \boldsymbol{z}_i$. 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

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

where games G_1 and G_2 are defined in Fig. 4 (right). 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 $(x, x) \in \mathrm{Dom}(T)$, then G_1 and G_2 are identical until Bad, and $\Pr[\mathsf{Bad}] \le q_{\mathcal{P}}/N$ because x is random in $[N]$ and $\mathrm{Dom}(T)$ 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

$$E \coloneqq \left\{ H \in \mathrm{Fun}(N^2, M) \;\middle|\; \begin{array}{c} (\forall k \in \mathbb{N})\big((\exists x_1, \ldots x_k \in [N])\big) \\ (x_i \neq x_j \wedge H(x_i, x_i) = H(x_j, x_j)) \implies (k < \bar{M})\big) \end{array} \right\},$$

where $\bar{M} \coloneqq \lceil \sqrt[4]{M} \rceil$. In other words, E is the set of all functions $H \in \mathrm{Fun}(N^2, M)$ 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 $\mathrm{MSE}^{\mathcal{S},\mathcal{D}}$ with bit $b = 0$ fixed and inverted winning condition. Then $\Pr[\mathrm{MSE}^{\mathcal{S},\mathcal{D}} \mid b = 0] = 1 - \Pr[\mathsf{G}]$, and by the law of total probability,

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

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

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

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

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 Fig. 1). Observe that this result also implies security in the BBU setting by Proposition 1. Our main result, where sources and predictors can query the entire oracle, is presented in Section 6.

Theorem 1. *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 $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$ 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*

$$\mathrm{Adv}^{\mathrm{mse}}_{\mathcal{S},\mathcal{D}} \leq \frac{\ell q_{\mathcal{S}} + q_{\mathcal{D}}}{2^k} \,. \tag{3}$$

Proof Overview. Let $(\mathcal{S}, \mathcal{D})$ be an adversary in the MSE-game for Mono. We transition to a game where no source $\mathcal{S}_j$ in $\mathcal{S}$ or $\mathcal{D}$ queries H on $\boldsymbol{x}$, the vector containing the outputs of all the sources. For each $\mathcal{S}_j$, querying $\boldsymbol{x}$ means being able to predict the output returned by any other source (say, $\mathcal{S}_{j+1}$), and for $\mathcal{D}$ it means being able to predict the output of any given source (say, $\mathcal{S}_1$). The bound (3) accounts for both these terms.

If no party queries H on $\boldsymbol{x}$, we can move to yet another game where $y = H(\boldsymbol{x})$ is now sampled at random. Then observe that $(\mathcal{S}, \mathcal{D})$ cannot have any advantage in this game, since the challenge y is independent of the experiment bit b. □

Proof. Consider a tuple of sources $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$ 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 Fig. 5 (top):

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, $\Pr[\mathsf{G}_1] = \Pr[\mathsf{G}_0]$.

G_2 proceeds as G_1, but we return 1 if any source $\mathcal{S}_i$ or distinguisher $\mathcal{D}$ query vector $\boldsymbol{x}$ (which is used when computing $y = H(\boldsymbol{x})$) to the random oracle H. Notice that G_2 returns 1 whenever G_1 does, so that $\Pr[\mathsf{G}_2] \geq \Pr[\mathsf{G}_1]$.

G_3 proceeds as G_2, but we no longer compute y as $H(\boldsymbol{x})$, and instead sample it at random. We claim that G_2 and G_3 are indistinguishable. Indeed, if $\boldsymbol{x} \in Q$, both games return 1 by definition. If $\boldsymbol{x} \notin Q$, then $H(\boldsymbol{x})$ is a random value that is unknown to all parties, so computing y as $y = H(\boldsymbol{x})$ is equivalent to sampling it at random. Therefore, $\Pr[\mathsf{G}_3] = \Pr[\mathsf{G}_2]$.

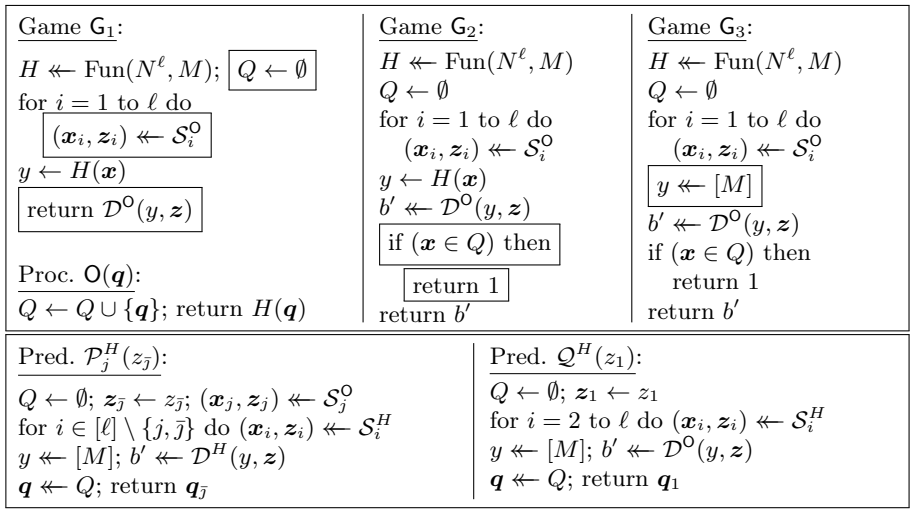

Fig. 5. Top: Code of the intermediate games in the proof of the MSE advantage bound in Theorem 1. **Bottom**: Definition of the predictors $\mathcal{P}_j$ and $\mathcal{Q}$ playing the predictability games for $\mathcal{S}_{\bar{\jmath}}$ and $\mathcal{S}_1$. Oracle O is the same as in the top part of the Figure.

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 $\boldsymbol{x}$ while playing G_3, and Bad_i the event that $\mathcal{S}_i$ queries $\boldsymbol{x}$ in the same game, for $i \in [\ell]$. Then notice that G_3 and G_4 are identical until $\mathsf{Bad} \coloneqq \bigvee_{i=0}^{\ell} \mathsf{Bad}_i$, and by the fundamental lemma of game playing we therefore have

$$\big|\Pr[\mathsf{G}_4] - \Pr[\mathsf{G}_3]\big| = \Pr[\mathsf{Bad}] \le \Pr[\mathsf{Bad}_0] + \sum_{i=1}^{\ell} \Pr[\mathsf{Bad}_i]\,.$$

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

$$2^{-k} \ge \Pr\big[\mathrm{Pred}_{\mathcal{S}_{\bar{\jmath}}}^{\mathcal{P}_j}\big] \ge \Pr\big[\mathrm{Pred}_{\mathcal{S}_{\bar{\jmath}}}^{\mathcal{P}_j} \,\big|\, \mathsf{Bad}_j\big] \Pr[\mathsf{Bad}_j] \ge \frac{1}{q_{\mathcal{S}}} \Pr[\mathsf{Bad}_j]\,.$$

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

Combining the estimates above we obtain

$$\mathrm{Adv}^{\mathrm{mse}}_{\mathcal{S},\mathcal{D}} = \Pr[\mathsf{G}_0] - \Pr[\mathsf{G}_4] \leq \left|\Pr[\mathsf{G}_4] - \Pr[\mathsf{G}_3]\right| \leq \epsilon\,. \qquad \square$$

REMARK. Observe that Theorem 1 is false when $\ell = 1$. Indeed, we bound the probability of $\mathcal{S}_j$ querying $\boldsymbol{x}$ 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 $\boldsymbol{x}$ in this case, because $\boldsymbol{x} = \boldsymbol{x}_1$ 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 $x \twoheadleftarrow [N]$, 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 $(q_{\mathcal{P}}, \delta)$-unpredictable for any $q_{\mathcal{P}}$ and $\delta = 1/\lfloor M/2 \rfloor \approx 2/M$, and yet $\mathrm{Adv}^{\mathrm{mse}}_{\mathcal{S},\mathcal{D}} \leq 1 - \lceil M/2 \rceil / M \approx 1/2$ 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 Fig. 1). In particular, each source can fully depend on the random oracle while producing entropic outputs.

Following Dodis et al. [15], our approach is as follows: To show that the output of H on the values $\boldsymbol{x}_i$ 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 $H(\boldsymbol{x})$, say, the lth bit. Two cases are now possible:

1. $\mathcal{P}$ queries $\boldsymbol{x}$ 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 $H(\boldsymbol{x})$. The advantage of $\mathcal{Q}$ can be bounded by a compression argument similar to the one in [15]. However, special care needs to be taken because unlike in [15], the challenge $\boldsymbol{x}$ 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 $\boldsymbol{x}$ with probability less than $\epsilon/(2 \log M)$, while still having advantage $\epsilon / \log M$ in guessing the lth bit of $H(\boldsymbol{x})$. We can then turn $\mathcal{P}$ into a guesser $\mathcal{Q}$ that never queries H on $\boldsymbol{x}$ but still has advantage $\epsilon/(2 \log M)$ in guessing the lth bit of $H(\boldsymbol{x})$, by simply running $\mathcal{P}$ and giving up whenever $\mathcal{P}$ wants to query $\boldsymbol{x}$. Predictor $\mathcal{Q}$ can then be used in an incompressibility argument to ultimately bound ϵ. For this part of the proof we follow De et al. [12, Lemma 8.4], but once again we must take care of the fact that the $\boldsymbol{x}_i$ are non-uniform and oracle-dependent.

Following the roadmap outlined above, we begin by analyzing unrecoverability.

Theorem 2 (Unrecoverability). *Let $\ell, M, N, q_{\mathcal{A}} \in \mathbb{N}$ and $k \in \mathbb{R}_{>0}$. Then* Mono *is $(N^\ell, 2^{-k}, N^\ell, q_{\mathcal{A}}, \epsilon)$-unrecoverable, where ϵ is given in the right-hand side of (4). More precisely, for every tuple $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$ of sources, each of which is $(N^\ell, 2^{-k})$-unpredictable and has unbounded access to its oracle, and every adversary $\mathcal{A}$ making at most $q_{\mathcal{A}}$ oracle calls, we have*

$$\mathrm{Adv}^{\mathrm{ur}}_{\mathcal{S},\mathcal{A}} = \mathcal{O}\left(\ell \sqrt[\ell+1]{\frac{q_{\mathcal{A}}}{2^{k\ell}}(\sigma + \ell N)}\right). \tag{4}$$

Here, $\sigma \geq \sum_{i=1}^{\ell} \sigma_i$, where $\sigma_i \in \mathbb{N}$ is an upper bound on the length of the leakage $\boldsymbol{z}_i$ returned by $\mathcal{S}_i$.

Proof. As discussed in Section 3, we can assume without loss of generality that $\mathcal{A}$ is deterministic and makes exactly $q_{\mathcal{A}}$ many oracle calls. By making one extra query, we can also assume that $\mathcal{A}$ always queries its output before returning it.

Let $\epsilon \coloneqq \mathrm{Adv}^{\mathrm{ur}}_{\mathcal{S},\mathcal{A}}$, and fix a constant $0 < \alpha < 1$ that we will specify later. We can now immediately distinguish two cases: If $\epsilon \leq \ell/((1-\alpha)2^k)$, then $\epsilon = \mathcal{O}(\ell/2^k)$. Thus, in this case ϵ already satisfies our bound (4), so assume for the remainder of the proof that $\epsilon > \ell/((1-\alpha)2^k)$. Our proof strategy will be to define a randomized encoding for a particular subset of oracles that we call "(α, β)-good." These are oracles $\mathsf{O} \in \mathrm{Fun}(N^\ell, M)$ such that (a) $(\mathcal{S}, \mathcal{A})$ win the UR game with "good" advantage when $H = \mathsf{O}$ and $\boldsymbol{z} = \mathbf{z}_{\mathsf{O}}$ are fixed in the game, where $\mathbf{z}_{\mathsf{O}}$ is some source leakage associated to O, and (b) none of the sources in $\mathcal{S}$ becomes "too predictable" when $H = \mathsf{O}$ and $z = \mathbf{z}_{\mathsf{O},i}$ are fixed in the Pred game. Instead of trivially specifying such an (α, β)-good oracle O in full, our encoding runs the adversary $\mathcal{A}$ to recover O on a selected set of inputs $G_{\mathsf{O}} \subseteq [N]^\ell$, thereby compressing O. However note that, for decoding to work, special care needs to be taken to avoid having to answer a query by $\mathcal{A}$ on an input where O has not yet been recovered. To ensure that this is the case on a large set (and therefore achieve meaningful compression), we randomize G_{O} and show that it is indeed large with high probability. After upper-bounding the length of every codeword, we can apply Lemma 2 to obtain an inequality involving ϵ, which yields (4).

GOOD PAIRS. For a parameter $\beta \geq 1$ that we choose later together with α, we say that a pair $(\mathsf{O}, \mathbf{z}) \in \mathrm{Fun}(N^\ell, M) \times \mathrm{Supp}(\mathcal{S}^{\mathsf{O}}_{1,2}) \times \cdots \times \mathrm{Supp}(\mathcal{S}^{\mathsf{O}}_{\ell,2})$ is (α, β)-*good* if the following two conditions hold:

1. $\Pr[\mathrm{UR}^{\mathcal{S},\mathcal{A}} \mid H = \mathsf{O} \wedge \boldsymbol{z} = \mathbf{z}] \geq \alpha\epsilon$, and
2. For every $i \in [\ell]$ and every predictor $\mathcal{P}$ with unbounded access to its oracle, $\Pr[\mathrm{Pred}^{\mathcal{P}}_{\mathcal{S}_i} \mid H = \mathsf{O} \wedge z = \mathbf{z}_i] \leq \beta \cdot 2^{-k}$.

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

1. Pair $(\mathsf{O}, \mathbf{z})$ is low-win: $(\mathcal{S}, \mathcal{A})$ wins the UR game with $(H, \boldsymbol{z})$ fixed to $(\mathsf{O}, \mathbf{z})$ with probability at most $\alpha\epsilon$, i.e., $\Pr[\mathrm{UR}^{\mathcal{S},\mathcal{A}} \mid H = \mathsf{O} \wedge \boldsymbol{z} = \mathbf{z}] < \alpha\epsilon$.
2. Pair $(\mathsf{O}, \mathbf{z})$ is low-entropy: There exist $i \in [\ell]$ and a predictor $\mathcal{P}_{\mathsf{O},\mathbf{z}}$ with unbounded oracle access such that $\Pr[\mathrm{Pred}^{\mathcal{P}_{\mathsf{O},\mathbf{z}}}_{\mathcal{S}_i} \mid H = \mathsf{O} \wedge z = \mathbf{z}_i] > \beta \cdot 2^{-k}$.

Let $GP_{\alpha,\beta}$ and $BP_{\alpha,\beta}$ be the sets of (α, β)-good and (α, β)-bad pairs, respectively.

Bounding good pairs. As our first step, we choose parameters α and β so that $GP_{\alpha,\beta} \neq \emptyset$, or, equivalently, that the probability $p_{\mathrm{gp}} \coloneqq \Pr[(H, \boldsymbol{z}) \in GP_{\alpha,\beta}]$ of a pair being good is non-zero. Denote by W_α and E_β the sets of low-win and low-entropy pairs, respectively, so that $BP_{\alpha,\beta} = W_\alpha \cup E_\beta$. We begin by estimating the probability $p_{\mathrm{bp}} \coloneqq \Pr[(H, \boldsymbol{z}) \in BP_{\alpha,\beta}]$ of a pair being (α, β)-bad. By the union bound, we have $p_{\mathrm{bp}} \leq \Pr[(H, \boldsymbol{z}) \in W_\alpha] + \Pr[(H, \boldsymbol{z}) \in E_\beta]$. To bound the term $p_{\mathrm{lw}} \coloneqq \Pr[(H, \boldsymbol{z}) \in W_\alpha]$, notice that

$$\begin{aligned}\epsilon = {} & \Pr[\mathrm{UR}^{\mathcal{S},\mathcal{A}} \mid (H, \boldsymbol{z}) \in W_\alpha] \Pr[(H, \boldsymbol{z}) \in W_\alpha] \\ & \qquad + \Pr[\mathrm{UR}^{\mathcal{S},\mathcal{A}} \mid (H, \boldsymbol{z}) \notin W_\alpha] \Pr[(H, \boldsymbol{z}) \notin W_\alpha] \\ & < \alpha\epsilon \cdot 1 + 1 \cdot (1 - p_{\mathrm{lw}}) \,,\end{aligned}$$

from which we obtain $p_{\mathrm{lw}} < 1 - (1 - \alpha)\epsilon$. For the term $p_{\mathrm{le}} \coloneqq \Pr[(H, \boldsymbol{z}) \in E_\beta]$, let $E_{\beta,i}$, $i \in [\ell]$, be the set of low-entropy pairs for the ith source. Then clearly $E_\beta = \bigcup_i E_{\beta,i}$, which means $p_{\mathrm{le}} \leq \sum p_{\mathrm{le},i}$, where $p_{\mathrm{le},i} \coloneqq \Pr[(H, \boldsymbol{z}) \in E_{\beta,i}]$. To bound $p_{\mathrm{le},i}$, let $\mathcal{P}$ be the predictor that queries its oracle H in full and then runs the predictor $\mathcal{P}_{H,\hat{z}}$, where $\hat{z} \coloneqq (0, \ldots, z, \ldots, 0)$, if $(H, \hat{z}) \in E_{\beta,i}$, and returns the constant 1 otherwise. Then we have

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

from which we obtain $p_{\mathrm{le},i} < 1/\beta$ for every $i \in [\ell]$, and thus $p_{\mathrm{le}} < \ell/\beta$. Collecting both terms, we have $p_{\mathrm{bp}} < 1 - (1 - \alpha)\epsilon + \ell/\beta$, meaning that

$$p_{\mathrm{gp}} = 1 - p_{\mathrm{bp}} > (1 - \alpha)\epsilon - \frac{\ell}{\beta} \,.$$

To ensure that good pairs exist (i.e., that $GP_{\alpha,\beta} \neq \emptyset$), we can set $0 < \alpha < 1$ to be any constant, and choose β so that $(1-\alpha)\epsilon - \ell/\beta > 0$, giving $\beta > \ell/((1-\alpha)\epsilon)$. 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), i.e., we can set $\beta = b\ell/(1 - \alpha)\epsilon$ for some $1 < b \leq 2$. Note that our assumption $\epsilon > \ell/((1 - \alpha)2^k)$ means that $2^k > \ell/((1 - \alpha)\epsilon)$, so such a value β always exists.

Good oracles. We now call an oracle $\mathsf{O} \in \mathrm{Fun}(N^\ell, M)$ (α, β)-*good* if $(\mathsf{O}, \mathbf{z}) \in GP_{\alpha,\beta}$ for some leakage vector $\mathbf{z}$. Denote by $G_{\alpha,\beta}$ the set of (α, β)-good oracles, and for each $\mathsf{O} \in G_{\alpha,\beta}$, fix $\mathbf{z}_{\mathsf{O}}$ such that $(\mathsf{O}, \mathbf{z}_{\mathsf{O}}) \in GP_{\alpha,\beta}$. Then notice that

$$p_{\mathrm{go}} \coloneqq \Pr[H \in G_{\alpha,\beta}] = \Pr[H \in G_{\alpha,\beta} \wedge \boldsymbol{z} \in \{0,1\}^*] \geq \Pr[(H, \boldsymbol{z}) \in GP_{\alpha,\beta}] \geq p_{\mathrm{gp}} \,.$$

Decomposing sources. Recall that, for every $\mathsf{O} \in G_{\alpha,\beta}$, we have that for all $i \in [\ell]$ and all predictors $\mathcal{P}$, $\Pr[\mathrm{Pred}^{\mathcal{P}}_{\mathcal{S}_i} \mid H = \mathsf{O} \wedge z = \mathbf{z}_{\mathsf{O},i}] \leq \beta \cdot 2^{-k}$. Denote by $\mathcal{X}_{\mathsf{O},i}$ the conditional distribution of source $\mathcal{S}_i^{\mathsf{O}}$ on $[N]$ conditioned to $z = \mathbf{z}_{\mathsf{O},i}$. Then the above says that each $\mathcal{X}_{\mathsf{O},i}$ is a $(k - \log \beta)$-source, meaning that by

Lemma 3, it can then be written as a convex combination of flat $(k - \log\beta)$-sources, all of which have support size *exactly* $2^{k-\log\beta}$ (which, by our choice of β, is a natural number). Then observe that, for each such O and each $i \in [\ell]$, there must exist a flat source $\mathcal{X}^*_{\mathsf{O},i}$ in the convex combination of $\mathcal{X}_{\mathsf{O},i}$ such that

$$\epsilon_{\mathsf{O}} \coloneqq \Pr[\mathrm{UR}^{\mathcal{X}^*,\mathcal{A}} \mid H = \mathsf{O} \wedge \boldsymbol{z} = \mathbf{z}_{\mathsf{O}}] \geq \alpha\epsilon\,, \tag{5}$$

where $\mathcal{X}^* \coloneqq (\mathcal{X}_{\mathsf{O},1}, \ldots, \mathcal{X}_{\mathsf{O},\ell})$. Indeed, if for all such $\mathcal{X}^*$ the probability in (5) was smaller than $\alpha\epsilon$, then on average (weighted as in the convex combination) it would again be smaller than $\alpha\epsilon$, contradicting that $(\mathsf{O}, \mathbf{z}_{\mathsf{O}})$ is not a low-win pair.

For any $\mathsf{O} \in G_{\alpha,\beta}$ and $i \in [\ell]$, define $\mathcal{S}_{\mathsf{O},i} \coloneqq \mathrm{Supp}(\mathcal{X}^*_{\mathsf{O},i}) \subseteq [N]$, which satisfies $S \coloneqq |\mathcal{S}_{\mathsf{O},i}| = 2^{k-\log\beta}$ independently of O, and let $\mathcal{S}_{\mathsf{O}} \coloneqq \mathcal{S}_{\mathsf{O},1} \times \cdots \times \mathcal{S}_{\mathsf{O},\ell}$, so that $|\mathcal{S}_{\mathsf{O}}| = S^\ell$ independently of O.

Probabilistic claim. Recall that our strategy is to define a source code for the set $G_{\alpha,\beta}$ of (α,β)-good oracles, and then to apply Lemma 2 to obtain a lower bound on the maximal length of any codeword in this encoding. This is an inequality involving ϵ which ultimately gives our bound (4). Our analysis requires the following probabilistic claim, which we establish next.

For any $\mathsf{O} \in G_{\alpha,\beta}$, pick a random subset $R_{\mathsf{O}} \subseteq \mathcal{S}_{\mathsf{O}}$ where every $\boldsymbol{x} \in \mathcal{S}_{\mathsf{O}}$ is placed in R_{O} with probability $1/(100q_{\mathcal{A}})$, and define sets

$$G_{\mathsf{O}} \coloneqq \left\{ \boldsymbol{x} \in R_{\mathsf{O}} \;\middle|\; \begin{array}{l} \mathcal{A}^{\mathsf{O}}(\mathsf{O}(\boldsymbol{x}), \mathbf{z}_{\mathsf{O}}) \text{ queries } \boldsymbol{x} \text{ to } \mathsf{O} \text{ and} \\ \text{does not query any other } \boldsymbol{x}' \in R_{\mathsf{O}} \end{array} \right\} \quad \text{and} \quad V_{\mathsf{O}} \coloneqq \mathsf{O}(G_{\mathsf{O}})\,.$$

Since for $\boldsymbol{x} \in G_{\mathsf{O}}$, $\mathcal{A}^{\mathsf{O}}(\mathsf{O}(\boldsymbol{x}), \mathbf{z}_{\mathsf{O}})$ only queries $\boldsymbol{x} \in R_{\mathsf{O}}$ and no other value in R_{O}, there cannot exist $\boldsymbol{x}, \boldsymbol{x}' \in G_{\mathsf{O}}$ such that $\boldsymbol{x} \neq \boldsymbol{x}'$ and $\mathsf{O}(\boldsymbol{x}) = \mathsf{O}(\boldsymbol{x}')$. In other words, O is injective on G_{O}, and we have $|V_{\mathsf{O}}| = |G_{\mathsf{O}}| \leq S^\ell$.

Claim. For every (α,β)-good oracle $\mathsf{O} \in \mathrm{Fun}(N^\ell, M)$, either $q_{\mathcal{A}} > \epsilon_{\mathsf{O}} S^\ell/2000$, or $\Pr[|G_{\mathsf{O}}| \geq 3\epsilon_{\mathsf{O}} S^\ell/(1000q_{\mathcal{A}})] \geq 0.9$, where the probability is over the choice of R_{O}.

Proof. Fix any $\mathsf{O} \in G_{\alpha,\beta}$. If $q_{\mathcal{A}} > \epsilon_{\mathsf{O}} S^\ell/2000$ there is nothing to prove, so assume from now on that $q_{\mathcal{A}} \leq \epsilon_{\mathsf{O}} S^\ell/2000$. Denote by I the number of elements $\boldsymbol{x} \in \mathcal{S}_{\mathsf{O}}$ such that $\mathcal{A}^{\mathsf{O}}(\mathsf{O}(\boldsymbol{x}), \mathbf{z}_{\mathsf{O}})$ queries $\boldsymbol{x}$ during its execution. Notice that I is well defined because $\mathcal{A}$ is deterministic, and $I \geq \epsilon_{\mathsf{O}} S^\ell$ because $\mathcal{A}$ wins the UR game in (5) on a fraction $\epsilon_{\mathsf{O}} S^\ell$ of all inputs and we assume that it always queries its output to the oracle before returning it.

Now let N_1 be the number of $\boldsymbol{x} \in R_{\mathsf{O}}$ such that $\mathcal{A}^{\mathsf{O}}(\mathsf{O}(\boldsymbol{x}), \mathbf{z}_{\mathsf{O}})$ queries $\boldsymbol{x}$ to O, and N_2 the number of $\boldsymbol{x} \in R_{\mathsf{O}}$ such that $\mathcal{A}^{\mathsf{O}}(\mathsf{O}(\boldsymbol{x}), \mathbf{z}_{\mathsf{O}})$ queries $\boldsymbol{x}$ to O and some other query $\boldsymbol{x}'$ of $\mathcal{A}^{\mathsf{O}}(\mathsf{O}(\boldsymbol{x}), \mathbf{z}_{\mathsf{O}})$ also lies in R_{O}. Then, by definition, $|G_{\mathsf{O}}| = N_1 - N_2$. Our strategy now is to lower-bound N_1 and to upper-bound N_2, in both cases with "high" probability 0.95 over the choice of R_{O}; when both bounds hold, which happens with "large" probability 0.9, we obtain the desired lower bound on $|G_{\mathsf{O}}|$. To establish our bounds on N_1 and N_2, we first estimate their expected values, and then apply an appropriate concentration bound, proving that large deviations from the expectation are unlikely.

For N_1, notice that $\mathbb{E}[N_1] = I/(100q_{\mathcal{A}})$, and using the Chernoff bound gives

$$\Pr\left[N_1 \le \left(1-\frac{1}{2}\right)\cdot\frac{I}{100q_{\mathcal{A}}}\right] \le \left(\frac{e^{-1/2}}{(1-1/2)^{1-1/2}}\right)^{I/(100q_{\mathcal{A}})} = \left(\frac{2}{e}\right)^{I/(200q_{\mathcal{A}})}.$$

Since $q_{\mathcal{A}} \le \epsilon_{\mathsf{O}} S^\ell/2000 \le I/2000$ by assumption, we have $I/(200q_{\mathcal{A}}) \ge 10$, meaning $\Pr[N_1 \le I/(200q_{\mathcal{A}})] \le (2/e)^{10} \le 0.05$, or $\Pr[N_1 > I/(200q_{\mathcal{A}})] \ge 0.95$.

Similarly, for N_2, observe that

$$\begin{aligned}\mathbb{E}[N_2] &= \frac{1}{100q_{\mathcal{A}}}\cdot I\cdot\left(1-\left(1-\frac{1}{100q_{\mathcal{A}}}\right)^{q_{\mathcal{A}}-1}\right)\\ &\lesssim \frac{I}{100q_{\mathcal{A}}}\cdot(1-e^{-1/100}) \le \frac{I}{100q_{\mathcal{A}}}\cdot\frac{1}{100} = \frac{I}{10^4 q_{\mathcal{A}}},\end{aligned}$$

so that by Markov's inequality,

$$\Pr\left[N_2 > 20\cdot\frac{I}{10^4 q_{\mathcal{A}}} = \frac{I}{500q_{\mathcal{A}}}\right] \le \frac{1}{20} = 0.05,$$

or, equivalently, $\Pr[N_2 \le I/(500q_{\mathcal{A}})] \ge 0.95$. Combining these two estimates, we obtain that when both events $(N_1 > I/(200q_{\mathcal{A}}))$ and $(N_2 \le I/(500q_{\mathcal{A}}))$ occur, which happens with probability at least 0.9, then

$$|G_{\mathsf{O}}| = N_1 - N_2 \ge \frac{I}{200q_{\mathcal{A}}} - \frac{I}{500q_{\mathcal{A}}} = \frac{3I}{1000q_{\mathcal{A}}},$$

and since $I \ge \epsilon_{\mathsf{O}} S^\ell$, this also means

$$\Pr\left[|G_{\mathsf{O}}| \ge \frac{3\epsilon_{\mathsf{O}} S^\ell}{1000q_{\mathcal{A}}}\right] \ge 0.9. \qquad \square$$

Encoding of good oracles. We can now define our randomized encoding, and begin with another case distinction. If $q_{\mathcal{A}} > \alpha\epsilon S^\ell/4000e$, then we have

$$\epsilon < \frac{4000eq_{\mathcal{A}}}{\alpha S^\ell} = \frac{4000eq_{\mathcal{A}}\beta^\ell}{\alpha 2^{k\ell}} = \frac{4000eq_{\mathcal{A}}b^\ell\ell^\ell}{\alpha 2^{k\ell}(1-\alpha)^\ell\epsilon^\ell},$$

which in turn yields $\epsilon = \mathcal{O}(\ell\sqrt[\ell+1]{q_{\mathcal{A}}\cdot 2^{-k\ell}})$. In this case, ϵ again satisfies our bound (4), so assume from now on that $q_{\mathcal{A}} \le \alpha\epsilon S^\ell/4000e$. The latter also means $q_{\mathcal{A}} < \epsilon_{\mathsf{O}} S^\ell/2000$ for every (α,β)-good oracle $\mathsf{O} \in G_{\alpha,\beta}$, and therefore $\Pr[|G_{\mathsf{O}}| \ge 3\epsilon_{\mathsf{O}} S^\ell/(1000q_{\mathcal{A}})] \ge 0.9$ for every such O by our claim above. We then define a randomized encoding for the set $G_{\alpha,\beta}$ of (α,β)-good oracles as follows:

Encoding: Given $\mathsf{O} \in G_{\alpha,\beta}$ and random coins r, first form $\mathcal{S}_{\mathsf{O}}$, and generate R_{O} using r. From R_{O}, then compute G_{O} and V_{O}. If $|G_{\mathsf{O}}| < 3\epsilon_{\mathsf{O}} S^\ell/(1000q_{\mathcal{A}})$, the encoder returns ε. Otherwise, encode the advice string z_{O}, create a description of each $\mathcal{S}_{\mathsf{O},i}$ as a subset of $[N]$, and a description of V_{O} as a subset of $[M]$.

Then, for each $y \in V_{\mathsf{O}}$ (in lexicographic order), run $\mathcal{A}^{\mathsf{O}}(y, \mathbf{z}_{\mathsf{O}})$ and include a description of the answers of each query $\mathcal{A}$ makes that has not been included so far, except for the single query that lies in R_{O}, which is the preimage of y. Finally, include $\mathsf{O}(\boldsymbol{x})$ for all $\boldsymbol{x} \in [N]^\ell$ not considered so far, in lexicographic order of $\boldsymbol{x}$. To summarize, an encoding of O consists of the following:

- The advice string $\mathbf{z}_{\mathsf{O}}$, using σ bits;
- For each $i \in [\ell]$, the size $|\mathcal{S}_{\mathsf{O},i}|$, encoded as a string of $\log N$ bits;
- For each $i \in [\ell]$, a description of $\mathcal{S}_{\mathsf{O},i} \subseteq [N]$ of size $\log \binom{N}{|\mathcal{S}_{\mathsf{O},i}|}$ bits;
- The size $|V_{\mathsf{O}}| = |G_{\mathsf{O}}|$, encoded as a string of $\log N^\ell$ bits;
- A description of $V_{\mathsf{O}} \subseteq [M]$ of size $\log \binom{M}{|V_{\mathsf{O}}|}$ bits;
- The answers to the queries of $\mathcal{A}$ that have not been included so far, and $\mathsf{O}(\boldsymbol{x})$ for all remaining $\boldsymbol{x}$, which have combined size $(N^\ell - |G_{\mathsf{O}}|) \log M$.

Decoding: Given a codeword as above and random coins r, we decode it as follows:

- If the codeword was the empty string ε, return the constant function O that always returns 1. Otherwise, initialize an empty table for O.
- Recover the advice string $\mathbf{z}_{\mathsf{O}}$, the sets $\mathcal{S}_{\mathsf{O},i}$ for $i \in [\ell]$, and the set V_{O} from the codeword. Then form $\mathcal{S}_{\mathsf{O}}$ and generate R_{O} using r.
- For each $y \in V_{\mathsf{O}}$ (in lexicographic order), run $\mathcal{A}$ on input $(y, \mathbf{z}_{\mathsf{O}})$. To do so, the decoder answers the oracle queries of $\mathcal{A}$ as follows. For every query $\boldsymbol{x}$ that $\mathcal{A}$ makes to O, if O has already been recovered on $\boldsymbol{x}$, then reply accordingly and continue running $\mathcal{A}$. If not and $\boldsymbol{x} \in R_{\mathsf{O}}$, then the answer must be y, so set $\mathsf{O}(\boldsymbol{x}) \coloneqq y$ and reply with y. Finally, if O is currently undefined on $\boldsymbol{x}$ and $\boldsymbol{x} \notin R_{\mathsf{O}}$, then read the next answer y' from the encoding, set $\mathsf{O}(\boldsymbol{x}) \coloneqq y'$, reply with y', and continue running $\mathcal{A}$.
- Complete the table for O with the list of values at the end of the encoding.

Notice that the decoder described above always manages to correctly recover O, except when $|G_{\mathsf{O}}| < 3\epsilon_{\mathsf{O}} S^\ell/(1000 q_{\mathcal{A}})$, an event happening with probability at most 0.1. We conclude that our encoding scheme has recovery probability $\delta \geq 0.9$.

Bounding codeword length. We next upper-bound the length of any codeword in the encoding above. For any $\mathsf{O} \in G_{\alpha,\beta}$, the length of the corresponding codeword is (either zero or) at most

$$
\begin{aligned}
&\sigma + 2\ell \log N + \sum_{i=1}^{\ell} \log \binom{N}{|\mathcal{S}_{\mathsf{O},i}|} + \log \binom{M}{|V_{\mathsf{O}}|} + (N^\ell - |G_{\mathsf{O}}|) \log M \\
&\quad \leq \sigma + 2\ell \log N + \sum_{i=1}^{\ell} |\mathcal{S}_{\mathsf{O},i}| \log \frac{eN}{|\mathcal{S}_{\mathsf{O},i}|} \\
&\qquad\qquad + |G_{\mathsf{O}}| \log \frac{e}{|G_{\mathsf{O}}|} + |G_{\mathsf{O}}| \log M + N^\ell \log M - |G_{\mathsf{O}}| \log M \\
&\quad = \sigma + 2\ell \log N + \sum_{i=1}^{\ell} |\mathcal{S}_{\mathsf{O},i}| \log \frac{eN}{|\mathcal{S}_{\mathsf{O},i}|} + |G_{\mathsf{O}}| \log \frac{e}{|G_{\mathsf{O}}|} + N^\ell \log M \,,
\end{aligned}
$$

where the inequality follows from $|V_{\mathsf{O}}| = |G_{\mathsf{O}}|$ and Inequality 3 from Lemma 4. We now use the fact that the function $x \mapsto x \log(eN/x)$ has a maximum at $x = N$, where it takes the value $N \log e$. This allows us to upper-bound the term

$$\sum_{i=1}^{\ell} |\mathcal{S}_{i,\mathsf{O}}| \log \frac{eN}{|\mathcal{S}_{i,\mathsf{O}}|} \leq \ell N \log e \,,$$

and our estimate above becomes

$$\sigma + 2\ell \log N + \ell N \log e + |G_{\mathsf{O}}| \log \frac{e}{|G_{\mathsf{O}}|} + N^{\ell} \log M \,.$$

Moreover, since a non-trivial codeword arises only when $|G_{\mathsf{O}}| \geq 3\epsilon_{\mathsf{O}} S^{\ell}/(1000 q_{\mathcal{A}})$, and $3\epsilon_{\mathsf{O}} S^{\ell}/(1000 q_{\mathcal{A}}) > \epsilon_{\mathsf{O}} S^{\ell}/(2000 q_{\mathcal{A}}) \geq \alpha\epsilon S^{\ell}/(2000 q_{\mathcal{A}}) \geq 1$ by our assumption on $q_{\mathcal{A}}$, this upper bound is at least

$$\sigma + 2\ell \log N + \ell N \log e + \frac{\alpha\epsilon S^{\ell}}{2000 q_{\mathcal{A}}} \log \frac{2000 e q_{\mathcal{A}}}{\alpha\epsilon S^{\ell}} + N^{\ell} \log M \,, \tag{6}$$

because the function $x \mapsto x \log(e/x)$ is decreasing for $x \geq 1$.

Analysis. Having given an upper bound on the maximum length of the randomized encoding, we can apply Lemma 2 and compress the set $G_{\alpha,\beta}$ of (α, β)-good oracles with a recovery probability of at least 0.9, giving

$$\begin{aligned} \sigma + 2\ell \log N + \ell N \log e + \frac{\alpha\epsilon S^{\ell}}{2000 q_{\mathcal{A}}} \log \frac{2000 e q_{\mathcal{A}}}{\alpha\epsilon S^{\ell}} + N^{\ell} \log M \\ \geq \log\bigl(p_{\mathrm{go}} M^{N^{\ell}}\bigr) + \log 0.9 = N^{\ell} \log M + \log(0.9 p_{\mathrm{go}}) \,, \end{aligned}$$

which after simplification becomes

$$\frac{\alpha\epsilon S^{\ell}}{2000 q_{\mathcal{A}}} \log \frac{\alpha\epsilon S^{\ell}}{2000 e q_{\mathcal{A}}} \leq \sigma + 2\ell \log N + \ell N \log e - \log(0.9 p_{\mathrm{go}}) \,.$$

Now recall that $\alpha\epsilon S^{\ell}/(2000 e q_{\mathcal{A}}) \geq 2$ by our case distinction on $q_{\mathcal{A}}$, so the above inequality becomes

$$\frac{\alpha\epsilon S^{\ell}}{2000 q_{\mathcal{A}}} \leq \sigma + 2\ell \log N + \ell N \log e - \log(0.9 p_{\mathrm{go}}) \,,$$

and substituting $S = 2^k/\beta$ and $p_{\mathrm{go}} > (1-\alpha)\epsilon - \ell/\beta$ gives

$$\frac{\alpha\epsilon 2^{k\ell}}{2000 q_{\mathcal{A}} \beta^{\ell}} \leq \sigma + 2\ell \log N + \ell N \log e - \log\left(0.9(1-\alpha)\epsilon - \frac{0.9\ell}{\beta}\right) . \tag{7}$$

We now once more distinguish two cases: If $0.9(1-\alpha)\epsilon - 0.9\ell/\beta \leq 1/N$, then

$$\epsilon \leq \frac{10}{9N(1-\alpha)} + \frac{\ell}{\beta(1-\alpha)} = \frac{10}{9N(1-\alpha)} + \frac{\epsilon}{b} \,,$$

and bringing the last term on the right to the left-hand side we obtain

$$\epsilon \leq \left(1 - \frac{1}{b}\right)^{-1} \cdot \frac{10}{9N(1-\alpha)} = \mathcal{O}\left(\frac{1}{N}\right),$$

showing that ϵ again satisfies our bound (4) in this case. If on the other hand $0.9(1-\alpha)\epsilon - 0.9\ell/\beta \geq 1/N$, Inequality (7) becomes

$$\frac{\alpha\epsilon 2^{k\ell}}{2000 q_{\mathcal{A}} \beta^{\ell}} \leq \sigma + (2\ell+1)\log N + \ell N \log e\,,$$

which, after substituting $\beta = b\ell/((1-\alpha)\epsilon)$, reads

$$\frac{\alpha(1-\alpha)^{\ell}\epsilon^{\ell+1}2^{k\ell}}{2000 q_{\mathcal{A}} b^{\ell}\ell^{\ell}} \leq \sigma + (2\ell+1)\log N + \ell N \log e\,.$$

Rearranging for ϵ, we finally obtain

$$\epsilon \leq \sqrt[\ell+1]{\frac{2000 q_{\mathcal{A}} b^{\ell}\ell^{\ell}}{\alpha(1-\alpha)^{\ell}2^{k\ell}}\big(\sigma + (2\ell+1)\log N + \ell N \log e\big)} = \mathcal{O}\left(\ell \sqrt[\ell+1]{\frac{q_{\mathcal{A}}}{2^{k\ell}}\big(\sigma + \ell N\big)}\right),$$

showing that our bound (4) also holds in this final case. This concludes the proof. □

Remark. We briefly consider the setting where every pair is known to be high-entropy, allowing us to set $\beta = 1$ and to simplify the bound on p_{bp}. This is the case, e.g., when all sources are uniformly distributed and independent of O.

If every pair is known to be high-entropy, then a pair is bad if and only if it is low-win, giving $p_{\mathrm{bp}} < 1 - (1-\alpha)\epsilon$. Following the steps in the proof above, we obtain a bound $\epsilon = \mathcal{O}\big(q_{\mathcal{A}}(\sigma + \ell N)/2^{k\ell}\big)$.

To compare our bound with that of Dodis et al. [15], we consider the case of uniform sources (where every oracle is high-entropy), and set $N = 2^k$ and $\ell = 1$. In this case, the encoding does not need to include the description of $\mathcal{S}_{\mathsf{O},i} = [N]$, so the term $\ell N \log e$ disappears from the bound, and $2\ell \log N$ becomes the dominant term in (7). With these parameters, we recover the result from [15].

Having proven our unrecoverability result, we now state and prove an appropriate extension of Lemma 8.4 from the full version of [12].

Lemma 5. *Let $N, \ell, q_{\mathcal{A}}, \sigma, S \in \mathbb{N}$ and $\epsilon \in \mathbb{R}_{>0}$, and assume that $S^{\ell} \geq 40$ and $q_{\mathcal{A}} \leq S^{\ell}/\log S^{\ell}$. Let $\mathcal{A}$ be a deterministic algorithm that makes at most $q_{\mathcal{A}}$ queries to its oracle, takes an advice string z of length at most σ, and is not allowed to query its input. Let $O \subseteq \mathrm{Fun}(N^{\ell}, \{0,1\})$ be a set such that, for each $\mathsf{O} \in O$, there exist $z \in \{0,1\}^{\sigma}$ and sets $\mathcal{S}_i \subseteq [N]$ with $|\mathcal{S}_i| = S$ for all $i \in [\ell]$ such that*

$$\Pr_{\boldsymbol{x}_i \twoheadleftarrow \mathcal{S}_i}[\mathcal{A}^{\mathsf{O}}(\boldsymbol{x}, z) = \mathsf{O}(\boldsymbol{x})] \geq \frac{1}{2} + \epsilon\,.$$

Then either $q_{\mathcal{A}} > \epsilon S^\ell$, or there exists a randomized encoding scheme $(\mathsf{Enc}, \mathsf{Dec})$ for O such that, for every $\mathsf{O} \in O$,

$$\Pr[\mathsf{Dec}(\mathsf{Enc}(\mathsf{O}, r), r) = \mathsf{O}] = \Omega\left(\frac{\epsilon}{q_{\mathcal{A}}}\right),$$

and the length of each codeword is at most

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

Proof. If $q_{\mathcal{A}} > \epsilon S^\ell$, there is nothing to prove, so assume from now on that $q_{\mathcal{A}} \leq \epsilon S^\ell$. Fix any $\mathsf{O} \in O$ and the corresponding $z \in \{0,1\}^\sigma$ and $\mathcal{S}_i \subseteq [N]$ as in the statement of the lemma, and define $\mathcal{S} \coloneqq \mathcal{S}_1 \times \cdots \times \mathcal{S}_\ell$. We now prove a probabilistic claim similar to the one in the proof of Theorem 2.

Pick a random subset $R \subseteq \mathcal{S}$ by independently placing each $\boldsymbol{x} \in \mathcal{S}$ in R with probability $1/(10q_{\mathcal{A}})$. Call $\boldsymbol{x} \in R$ *good* if $\mathcal{A}^{\mathsf{O}}(\boldsymbol{x}, z)$ makes no query inside R, and let $G \subseteq R$ be the set of good elements. Moreover, let $G_0, G_1 \subseteq G$ partition G into the $\boldsymbol{x}$ on which $\mathcal{A}^{\mathsf{O}}(\cdot, z)$ is correct and incorrect, respectively. We then call the set R itself *good* if

$$|G_0| - |G_1| \geq \frac{\epsilon S^\ell}{20 q_{\mathcal{A}}} \qquad \text{and} \qquad |G| = \mathcal{O}\left(\frac{S^\ell}{q_{\mathcal{A}}}\right).$$

Claim. The set R is good with probability at least $\epsilon/(40q_{\mathcal{A}})$, where the probability is taken over the random choice of R.

Proof. Denote by E_1 and E_2 the events that $|G_0| - |G_1| \geq \epsilon S^\ell/(20q_{\mathcal{A}})$ and that $|G| = \mathcal{O}(S^\ell/q_{\mathcal{A}})$, respectively, so that proving the claim amounts to showing that $\Pr[E_1 \cap E_2] \geq \epsilon/(40q_{\mathcal{A}})$. Furthermore, for each $\boldsymbol{x} \in \mathcal{S}$, let $q_{\boldsymbol{x}} \leq q_{\mathcal{A}}$ be the number of queries that $\mathcal{A}^{\mathsf{O}}(\boldsymbol{x}, z)$ makes inside $\mathcal{S}$. By definition of R and G we have that, for every $\boldsymbol{x} \in \mathcal{S}$,

$$\Pr[\boldsymbol{x} \in G] = \left(1 - \frac{1}{10q_{\mathcal{A}}}\right)^{q_{\boldsymbol{x}}} \cdot \frac{1}{10q_{\mathcal{A}}}.$$

This shows $\Pr[\boldsymbol{x} \in G] = \Theta(1/q_{\mathcal{A}})$, because clearly $\Pr[\boldsymbol{x} \in G] \leq 1/(10q_{\mathcal{A}})$, but also

$$\Pr[\boldsymbol{x} \in G] \geq \left(1 - \frac{1}{10q_{\mathcal{A}}}\right)^{q_{\mathcal{A}}} \cdot \frac{1}{10q_{\mathcal{A}}} \geq \frac{1}{20q_{\mathcal{A}}},$$

where the last inequality uses that $(1 - 1/(10q_{\mathcal{A}}))^{q_{\mathcal{A}}} \geq 1/2$ for all $q_{\mathcal{A}} \geq 1$. With these estimates we can now compute

$$\begin{aligned}
\mathbb{E}[|G_0| - |G_1|] &= \mathbb{E}\left[\sum_{\boldsymbol{x} \in \mathcal{S}} (\boldsymbol{x} \in G_0) - (\boldsymbol{x} \in G_1)\right] = \sum_{\boldsymbol{x} \in \mathcal{S}} \Pr[\boldsymbol{x} \in G_0] - \Pr[\boldsymbol{x} \in G_1] \\
&= \sum_{\boldsymbol{x} \in \mathcal{S}} \Pr[\boldsymbol{x} \in G] \cdot \big((\mathcal{A}^{\mathsf{O}}(\boldsymbol{x}, z) = \mathsf{O}(\boldsymbol{x})) - (\mathcal{A}^{\mathsf{O}}(\boldsymbol{x}, z) \neq \mathsf{O}(\boldsymbol{x}))\big) \\
&\geq \frac{1}{20q_{\mathcal{A}}} \cdot 2\epsilon S^\ell = \frac{\epsilon S^\ell}{10q_{\mathcal{A}}}.
\end{aligned}$$

Since for any choice of R it holds that $|G_0| - |G_1| \le S^\ell$, we have

$$\frac{\epsilon S^\ell}{10 q_{\mathcal{A}}} \le \mathbb{E}[|G_0| - |G_1| \mid E_1] \Pr[E_1] + \mathbb{E}[|G_0| - |G_1| \mid E_1^{\complement}] \Pr[E_1^{\complement}] \le S^\ell \Pr[E_1] + \frac{\epsilon S^\ell}{20 q_{\mathcal{A}}} (1 - \Pr[E_1]) ,$$

giving $\Pr[E_1] \ge \epsilon/(20 q_{\mathcal{A}})$. Moreover, note that

$$\mathbb{E}[|G|] = \mathbb{E}\left[\sum_{\boldsymbol{x} \in \mathcal{S}} (\boldsymbol{x} \in G)\right] = \sum_{\boldsymbol{x} \in \mathcal{S}} \Pr[\boldsymbol{x} \in G] = S^\ell \cdot \Theta\left(\frac{1}{q_{\mathcal{A}}}\right) = \Theta\left(\frac{S^\ell}{q_{\mathcal{A}}}\right) .$$

Thus, applying a Chernoff bound, one obtains

$$\Pr[E_2] \ge 1 - e^{-2S^\ell / q_{\mathcal{A}}}$$

for an appropriate constant hidden in the $\mathcal{O}$-notation. Now observe that

$$e^{-2S^\ell / q_{\mathcal{A}}} \le e^{-2 \log S^\ell} = S^{-2\ell} \le \frac{1}{40 S^\ell} \le \frac{\epsilon}{40 q_{\mathcal{A}}} ,$$

where the first inequality uses $q_{\mathcal{A}} \le S^\ell / \log S^\ell$, the second one holds since $S^\ell \ge 40$, and the last one because we are assuming in this part of the proof that $q_{\mathcal{A}} \le \epsilon S^\ell$. We thus conclude that

$$\Pr[E_2] \ge 1 - \frac{\epsilon}{40 q_{\mathcal{A}}} .$$

A union bound now yields the claim:

$$\Pr[E_1 \cap E_2] = 1 - \Pr[E_1^{\complement} \cup E_2^{\complement}] \ge 1 - \Pr[E_1^{\complement}] - \Pr[E_2^{\complement}] = \Pr[E_1] + \Pr[E_2] - 1 \ge \frac{\epsilon}{20 q_{\mathcal{A}}} - \frac{\epsilon}{40 q_{\mathcal{A}}} = \frac{\epsilon}{40 q_{\mathcal{A}}} .$$

□

We now define the randomized encoding for the set O as follows:

Encoding: Given $\mathsf{O} \in O$ and random coins r, first generate R using r. If R is not good, return ε. Otherwise, an encoding of O consists of the following:

- A description of the advice string z, taking σ bits;
- Descriptions of the sets $\mathcal{S}_i$ for $i \in [\ell]$, taking $\ell \log \binom{N}{S}$ bits;
- $\mathsf{O}(\boldsymbol{x})$ for $\boldsymbol{x} \in [N]^\ell \setminus R$ in lexicographic order of $\boldsymbol{x}$, taking $N^\ell - |R|$ bits;
- $\mathsf{O}(\boldsymbol{x})$ for $\boldsymbol{x} \in R \setminus G$ in lexicographic order of $\boldsymbol{x}$, taking $|R| - |G|$ bits;
- The set G_0, taking $\log(S^\ell) + \log \binom{|G|}{|G_0|}$ bits, which is at most

$$\ell \log(S) + |G| \cdot \mathbb{H}\left(\frac{1}{2} + \frac{\epsilon S^\ell}{40 q_{\mathcal{A}} |G|}\right) \le \ell \log(S) + |G| - \frac{\epsilon^2 S^{2\ell}}{\mathcal{O}(q_{\mathcal{A}}^2 |G|)} ,$$

where $\mathbb{H}$ is the binary entropy function. Here, the first statement follows from the fact that since R is good, $|G_0| - |G_1| \ge \epsilon S^\ell/(20 q_{\mathcal{A}})$, so that $|G_0| \ge |G|/2 + \epsilon S^\ell/(40 q_{\mathcal{A}})$, meaning $|G_0|/|G| \ge 1/2 + \epsilon S^\ell/(40 q_{\mathcal{A}} |G|)$. The inequality holds because $\mathbb{H}(1/2 + q) \le 1 - q^2$ for all $-1/2 \le q \le 1/2$.

Decoding: Given a codeword as above and the corresponding random coins r, we decode it as follows:

- If the codeword was the empty string ε, return the constant zero function. Otherwise, initialize an empty table for O.
- Recover the advice string z and the sets $\mathcal{S}_i$ and G_0 from the codeword, and use the shared randomness r to compute the set R.
- Fill in the entries $\mathsf{O}(\boldsymbol{x})$ corresponding to $\boldsymbol{x} \in [N]^\ell \setminus R$.
- Reconstruct the set G as follows: Run $\mathcal{A}$ on z and each $\boldsymbol{x} \in R$, answering its queries using the table for O. If $\mathcal{A}$ makes a query in R, $\boldsymbol{x}$ does not belong to G (then abort the execution of $\mathcal{A}$); otherwise, it does.
- Fill in the entries $\mathsf{O}(\boldsymbol{x})$ corresponding to $\boldsymbol{x} \in R \setminus G$.
- Run $\mathcal{A}$ on z and each $\boldsymbol{x} \in G$, answering its queries based on the table for O generated so far. Notice that it is now possible to do so because, by definition of G, $\mathcal{A}$ will only make queries outside R. The output of $\mathcal{A}^{\mathsf{O}}(\boldsymbol{x}, z)$ equals $\mathsf{O}(\boldsymbol{x})$ (resp., $1 - \mathsf{O}(\boldsymbol{x})$) if $\boldsymbol{x} \in G_0$ (resp., $\boldsymbol{x} \notin G_0$).

Notice that the decoder described above always manages to correctly recover O, except when the set R is not good, an event happening with probability at most $1 - \epsilon/(40q_{\mathcal{A}})$ by our claim above. We conclude that our encoding scheme has recovery probability $\delta \geq \epsilon/(40q_{\mathcal{A}}) = \Omega(\epsilon/q_{\mathcal{A}})$.

Adding up the lengths of the various parts in the encoding, the overall length of a codeword is then (either zero or) at most

$$\begin{aligned}\sigma + \ell \log(S) + \ell S \log\left(\frac{eN}{S}\right) + N^\ell + \mathcal{O}(1) - \frac{\epsilon^2 S^{2\ell}}{\mathcal{O}(q_{\mathcal{A}}^2 |G|)} \\ \leq \sigma + 2\ell S \log\left(\frac{eN}{S}\right) + N^\ell + \mathcal{O}(1) - \Omega\left(\frac{\epsilon^2 S^\ell}{q_{\mathcal{A}}}\right),\end{aligned}$$

where we use $|G| = \mathcal{O}(S^\ell/q_{\mathcal{A}})$, since R is good for every non-trivial codeword. □

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

Theorem 3 (MSE). *Let $\ell, M, N, q_{\mathcal{D}} \in \mathbb{N}$ and $k \in \mathbb{R}_{>0}$. Then* Mono *is $(N^\ell, 2^{-k}, N^\ell, q_{\mathcal{D}}, \epsilon)$-MSE, where ϵ is given in the right-hand side of* (8)*. More precisely, for every tuple $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$ of sources, each of which is $(N^\ell, 2^{-k})$-unpredictable and has unbounded access to its oracle, and every distinguisher $\mathcal{D}$ making at most $q_{\mathcal{D}}$ oracle calls, we have*

$$\mathrm{Adv}^{\mathrm{mse}}_{\mathcal{S},\mathcal{D}} = \mathcal{O}\left(\ell \log M \sqrt[\ell+1]{\frac{q_{\mathcal{D}}}{2^{k\ell}}(\sigma + \ell N)}\right). \tag{8}$$

Here, $\sigma \geq \sum_{i=1}^{\ell} \sigma_i$, where $\sigma_i \in \mathbb{N}$ is an upper bound on the length of the leakage $\boldsymbol{z}_i$ returned by $\mathcal{S}_i$.

Game $\mathrm{nbPred}^{\mathcal{P}}_{\mathcal{S},j}$:	Game $\mathrm{outPred}^{\mathcal{Q}}_{\mathcal{S},j}$:
$H \twoheadleftarrow \mathrm{Fun}(N^\ell, M)$	$H \twoheadleftarrow \mathrm{Fun}(N^\ell, M)$
for $i = 1$ to ℓ do $(\boldsymbol{x}_i, \boldsymbol{z}_i) \twoheadleftarrow \mathcal{S}_i^H$	for $i = 1$ to ℓ do $(\boldsymbol{x}_i, \boldsymbol{z}_i) \twoheadleftarrow \mathcal{S}_i^H$
$y \leftarrow H(\boldsymbol{x})$; $y'_j \twoheadleftarrow \mathcal{P}^H(y_1, \ldots, y_{j-1}, \boldsymbol{z})$	$y \leftarrow H_j(\boldsymbol{x})$; $z \twoheadleftarrow \mathcal{Q}_0^H$; $y' \twoheadleftarrow \mathcal{Q}_1^{H_j}(\boldsymbol{x}, z, \boldsymbol{z})$
return $(y_j = y'_j)$	return $(y = y')$

Fig. 6. Left: The jth next-bit prediction game for $\mathcal{S}$ played by a predictor $\mathcal{P}$. **Right**: The output prediction game for $\mathcal{S}$, played by a predictor $\mathcal{Q} = (\mathcal{Q}_0, \mathcal{Q}_1)$. Here, $\mathcal{Q}_1$ is not allowed to query its input $\boldsymbol{x}$ to its oracle, and H_j is the projection on the jth bit of H.

Proof. Let $\epsilon \coloneqq \mathrm{Adv}^{\mathrm{mse}}_{\mathcal{S},\mathcal{D}}$. By Yao's equivalence principle [35], any $q_{\mathcal{D}}$-query distinguisher can be converted into a predictor $\mathcal{P}$ with the same query complexity, which can predict the lth output bit of the oracle for some $l \in [\log M]$, that is,

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

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

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

$$\epsilon = \mathcal{O}\left(\ell \log M \sqrt[\ell+1]{\frac{q_{\mathcal{D}}}{2^{k\ell}}(\sigma + \ell N)}\right), \tag{9}$$

that is, our bound (8) 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 $H_l(\boldsymbol{x})$ (the lth output bit of $H(\boldsymbol{x})$) without querying its input. Algorithm $\mathcal{Q}_0$ returns a table containing, in lexicographical order of the inputs, all but the lth 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 $(H(\boldsymbol{x})_1, \ldots, H(\boldsymbol{x})_{l-1}, \boldsymbol{z})$, and combines the output of $\mathcal{P}$ with the generated table z, such that

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

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

From here on, we can carry out a compression argument similar to the one in the proof of Theorem 2. We call a pair (O, z) (α, β)-good if the following hold:

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

In other words, (O, z) is (α, β)-bad if either of the following apply:

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

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

$$p_{\mathrm{bp}} \le \Pr[(H,\boldsymbol{z}) \in W_\alpha] + \sum_i \Pr[(H,\boldsymbol{z}) \in E_{\beta,i}]\,.$$

To bound $p_{\mathrm{lw}} \coloneqq \Pr[(H,\boldsymbol{z}) \in W_\alpha]$, notice that

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

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

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

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

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 $\mathsf{O} \in \mathrm{Fun}(N^\ell, M)$ such that $(\mathsf{O},\mathbf{z}) \in GP_{\alpha,\beta}$ for some leakage vector $\mathbf{z}$. For every $\mathsf{O} \in G_{\alpha,\beta}$, fix any leakage vector $\mathbf{z}_{\mathsf{O}}$ such that $(\mathsf{O},\mathbf{z}_{\mathsf{O}}) \in GP_{\alpha,\beta}$. Then

$$p_{\mathrm{go}} \coloneqq \Pr[H \in G_{\alpha,\beta}] \ge \Pr[(H,\boldsymbol{z}) \in GP_{\alpha,\beta}] = p_{\mathrm{gp}}\,,$$

meaning that

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

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

$$\Pr[\mathrm{outPred}^{\mathcal{Q}}_{\mathcal{X}^*_{\mathsf{O},i},l} \mid H = \mathsf{O} \wedge \boldsymbol{z} = \mathbf{z}_{\mathsf{O}}] \ge \frac{1}{2} + \alpha\epsilon'\,.$$

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 (8), 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 5.

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

Encoding: Given $\mathsf{O} \in G_{\alpha,\beta}$, we encode it as follows:
- Generate a table containing, in lexicographical order of $\boldsymbol{x}$, all but the lth bit of the oracle outputs $\mathsf{O}(\boldsymbol{x})$, taking $N^\ell(\log M - 1)$ bits;
- Apply the encoding algorithm from Lemma 5 to obtain a randomized encoding of O_j, which has an encoding length of at most $\sigma + 2\ell S \log(eN/S) + N^\ell - \Omega\big((\alpha\epsilon/\log M)^2 \cdot S^\ell/q_{\mathcal{A}}\big) + \mathcal{O}(1)$ many bits.

Decoding: Given a codeword as above, we decode it as follows:
- Initialize an empty table for O;
- Extract the table of all but the lth bit of outputs from the codeword and use this to fill the oracle table;
- Follow the decoding algorithm in Lemma 5 to extract the lth bit of each oracle output.

From Lemma 5 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 $\epsilon \le q_{\mathcal{D}} \log M/(\alpha N^\ell)$ then again we are done, since we obtain a bound that verifies (8). Indeed,

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

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

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

where the first inequality uses the fact that the function $x \mapsto x \log(eN/x)$ 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

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

Hence,

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

We now once more distinguish two cases: If $\log M/\epsilon > N^\ell$, then $\epsilon < \log M/N^\ell$, and again our bound (8) holds. Otherwise, we obtain

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

which gives

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

and thus

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

This concludes the proof. □

Remark. Compared to the work of Dodis and Oliveira [16], note that our MSE theorem allows extracting many more bits (akin to a PRG security result), which cannot hold in the statistical setting of [16]. 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 prove in Appendix A 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 $\ell \ge 2$.

Acknowledgments

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.

Pooya Farshim was supported in part by EPSRC grant EP/V034065/1. Patrick Harasser was funded by the Deutsche Forschungsgemeinschaft (DFG) – SFB 1119 – 236615297.

References

1. Aggarwal, D., Obremski, M., Ribeiro, J.L., Siniscalchi, L., Visconti, I.: How to extract useful randomness from unreliable sources. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part I. LNCS, vol. 12105, pp. 343–372. Springer, Heidelberg (May 2020). https://doi.org/10.1007/978-3-030-45721-1_13

2. Barak, B., Impagliazzo, R., Wigderson, A.: Extracting randomness using few independent sources. In: 45th FOCS. pp. 384–393. IEEE Computer Society Press (Oct 2004). https://doi.org/10.1109/FOCS.2004.29
3. Beigi, S., Etesami, O., Gohari, A.: Deterministic randomness extraction from generalized and distributed Santha-Vazirani sources. In: Halldórsson, M.M., Iwama, K., Kobayashi, N., Speckmann, B. (eds.) ICALP 2015, Part I. LNCS, vol. 9134, pp. 143–154. Springer, Heidelberg (Jul 2015). https://doi.org/10.1007/978-3-662-47672-7_12
4. Bellare, M., Hoang, V.T., Keelveedhi, S.: Instantiating random oracles via UCEs. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 398–415. Springer, Heidelberg (Aug 2013). https://doi.org/10.1007/978-3-642-40084-1_23
5. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM CCS 93. pp. 62–73. ACM Press (Nov 1993). https://doi.org/10.1145/168588.168596
6. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 409–426. Springer, Heidelberg (May / Jun 2006). https://doi.org/10.1007/11761679_25
7. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Sponge-based pseudo-random number generators. In: Mangard, S., Standaert, F.X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 33–47. Springer, Heidelberg (Aug 2010). https://doi.org/10.1007/978-3-642-15031-9_3
8. Chattopadhyay, E., Zuckerman, D.: Explicit two-source extractors and resilient functions. In: Wichs, D., Mansour, Y. (eds.) 48th ACM STOC. pp. 670–683. ACM Press (Jun 2016). https://doi.org/10.1145/2897518.2897528
9. Chor, B., Goldreich, O.: Unbiased bits from sources of weak randomness and probabilistic communication complexity (extended abstract). In: 26th FOCS. pp. 429–442. IEEE Computer Society Press (Oct 1985). https://doi.org/10.1109/SFCS.1985.62
10. Coretti, S., Dodis, Y., Guo, S., Steinberger, J.P.: Random oracles and non-uniformity. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 227–258. Springer, Heidelberg (Apr / May 2018). https://doi.org/10.1007/978-3-319-78381-9_9
11. Coretti, S., Dodis, Y., Karthikeyan, H., Tessaro, S.: Seedless Fruit is the sweetest: Random number generation, revisited. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part I. LNCS, vol. 11692, pp. 205–234. Springer, Heidelberg (Aug 2019). https://doi.org/10.1007/978-3-030-26948-7_8
12. De, A., Trevisan, L., Tulsiani, M.: Time space tradeoffs for attacks against one-way functions and PRGs. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 649–665. Springer, Heidelberg (Aug 2010). https://doi.org/10.1007/978-3-642-14623-7_35
13. Dodis, Y., Elbaz, A., Oliveira, R., Raz, R.: Improved randomness extraction from two independent sources. In: Ron, D. (ed.) RANDOM 04. LNCS, vol. 3122, pp. 334–344. Springer, Heidelberg (Aug 2004). https://doi.org/10.1007/978-3-540-27821-4_30
14. Dodis, Y., Gennaro, R., Håstad, J., Krawczyk, H., Rabin, T.: Randomness extraction and key derivation using the CBC, cascade and HMAC modes. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 494–510. Springer, Heidelberg (Aug 2004). https://doi.org/10.1007/978-3-540-28628-8_30

15. Dodis, Y., Guo, S., Katz, J.: Fixing cracks in the concrete: Random oracles with auxiliary input, revisited. In: Coron, J.S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part II. LNCS, vol. 10211, pp. 473–495. Springer, Heidelberg (Apr / May 2017). https://doi.org/10.1007/978-3-319-56614-6_16
16. Dodis, Y., Oliveira, R.: On extracting private randomness over a public channel. In: Sahai, A. (ed.) RANDOM 2003. LNCS, vol. 2764, pp. 252–263. Springer, Heidelberg (Aug 2003). https://doi.org/10.1007/978-3-540-45198-3_22
17. Dodis, Y., Ong, S.J., Prabhakaran, M., Sahai, A.: On the (im)possibility of cryptography with imperfect randomness. In: 45th FOCS. pp. 196–205. IEEE Computer Society Press (Oct 2004). https://doi.org/10.1109/FOCS.2004.44
18. Dodis, Y., Ristenpart, T., Vadhan, S.P.: Randomness condensers for efficiently samplable, seed-dependent sources. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 618–635. Springer, Heidelberg (Mar 2012). https://doi.org/10.1007/978-3-642-28914-9_35
19. Dodis, Y., Shamir, A., Stephens-Davidowitz, N., Wichs, D.: How to eat your entropy and have it too - optimal recovery strategies for compromised RNGs. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 37–54. Springer, Heidelberg (Aug 2014). https://doi.org/10.1007/978-3-662-44381-1_3
20. Dodis, Y., Spencer, J.: On the (non)universality of the one-time pad. In: 43rd FOCS. pp. 376–387. IEEE Computer Society Press (Nov 2002). https://doi.org/10.1109/SFCS.2002.1181962
21. Dodis, Y., Vaikuntanathan, V., Wichs, D.: Extracting randomness from extractor-dependent sources. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part I. LNCS, vol. 12105, pp. 313–342. Springer, Heidelberg (May 2020). https://doi.org/10.1007/978-3-030-45721-1_12
22. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (Aug 1987). https://doi.org/10.1007/3-540-47721-7_12
23. Gazi, P., Pietrzak, K., Tessaro, S.: The exact PRF security of truncation: Tight bounds for keyed sponges and truncated CBC. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part I. LNCS, vol. 9215, pp. 368–387. Springer, Heidelberg (Aug 2015). https://doi.org/10.1007/978-3-662-47989-6_18
24. Gennaro, R., Trevisan, L.: Lower bounds on the efficiency of generic cryptographic constructions. In: 41st FOCS. pp. 305–313. IEEE Computer Society Press (Nov 2000). https://doi.org/10.1109/SFCS.2000.892119
25. Intel Corporation: Intel Digital Random Number Generator (DRNG) Software Implementation Guide (2019), https://www.intel.com/content/www/us/en/developer/articles/guide/intel-digital-random-number-generator-drng-software-implementation-guide.html, accessed: 2024-05-19
26. Li, X.: Three-source extractors for polylogarithmic min-entropy. In: Guruswami, V. (ed.) 56th FOCS. pp. 863–882. IEEE Computer Society Press (Oct 2015). https://doi.org/10.1109/FOCS.2015.58
27. McInnes, J.L., Pinkas, B.: On the impossibility of private key cryptography with weakly random keys. In: Menezes, A.J., Vanstone, S.A. (eds.) CRYPTO'90. LNCS, vol. 537, pp. 421–435. Springer, Heidelberg (Aug 1991). https://doi.org/10.1007/3-540-38424-3_31
28. Nisan, N., Zuckerman, D.: Randomness is linear in space. J. Comput. Syst. Sci. **52**(1), 43–52 (1996). https://doi.org/10.1006/JCSS.1996.0004, https://doi.org/10.1006/jcss.1996.0004

29. Ristenpart, T., Shacham, H., Shrimpton, T.: Careful with composition: Limitations of the indifferentiability framework. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 487–506. Springer, Heidelberg (May 2011). https://doi.org/10.1007/978-3-642-20465-4_27
30. Santha, M., Vazirani, U.V.: Generating quasi-random sequences from slightly-random sources (extended abstract). In: 25th FOCS. pp. 434–440. IEEE Computer Society Press (Oct 1984). https://doi.org/10.1109/SFCS.1984.715945
31. Shrimpton, T., Terashima, R.S.: A provable-security analysis of Intel's secure key RNG. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 77–100. Springer, Heidelberg (Apr 2015). https://doi.org/10.1007/978-3-662-46800-5_4
32. Unruh, D.: Random oracles and auxiliary input. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 205–223. Springer, Heidelberg (Aug 2007). https://doi.org/10.1007/978-3-540-74143-5_12
33. Vadhan, S.P.: Pseudorandomness. Now Publishers (2012). https://doi.org/10.1561/0400000010
34. Woodage, J., Shumow, D.: An analysis of NIST SP 800-90A. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part II. LNCS, vol. 11477, pp. 151–180. Springer, Heidelberg (May 2019). https://doi.org/10.1007/978-3-030-17656-3_6
35. Yao, A.C.C.: Theory and applications of trapdoor functions (extended abstract). In: 23rd FOCS. pp. 80–91. IEEE Computer Society Press (Nov 1982). https://doi.org/10.1109/SFCS.1982.45

A One-Wayness

In this appendix, we show that the monolithic random oracle Mono is one-way. Before coming to our result, we formally introduce the notion that we achieve. Indeed, there are at least two possible definitions of one-wayness when the preimage $\boldsymbol{x}$ of the challenge point $y = H(\boldsymbol{x})$ is not sampled uniformly at random: Either, the adversary wins whenever it returns *any* valid preimage $\boldsymbol{x}'$ of y, or the adversary is forced to output a preimage $\boldsymbol{x}'$ that lies *in the support* of the sources.[9] In this appendix, we opt for the latter, because we view the support of the sources as the set of all "valid" hash inputs. Accordingly, to win one-wayness, the adversary should return any "valid" preimage of the challenge point.

ONE-WAYNESS. Let $\ell, M, N \in \mathbb{N}$, $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$ a tuple of sources, and $\mathcal{A}$ an adversary. We define the advantage of $(\mathcal{S}, \mathcal{A})$ in the one-wayness game for Mono as $\mathrm{Adv}^{\mathrm{ow}}_{\mathcal{S},\mathcal{A}} \coloneqq \Pr[\mathrm{OW}^{\mathcal{S},\mathcal{A}}]$, where the game $\mathrm{OW}^{\mathcal{S},\mathcal{A}}$ is given in Fig. 7. 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 $(q_{\mathcal{P}}, \delta, q_{\mathcal{S}}, q_{\mathcal{A}}, \epsilon)$-one-way 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 $\mathrm{Adv}^{\mathrm{ow}}_{\mathcal{S},\mathcal{A}} \leq \epsilon$.

Theorem 4 (One-wayness). *Let $\ell, M, N, q_{\mathcal{A}} \in \mathbb{N}$ and $k \in \mathbb{R}_{>0}$. Then Mono is $(N^\ell, 2^{-k}, N^\ell, q_{\mathcal{A}}, \epsilon)$-one-way, where ϵ is given in the right-hand side of (10). More precisely, for every tuple $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$ of sources, each of which is $(N^\ell, 2^{-k})$-unpredictable and has unbounded access to its oracle, and every adversary $\mathcal{A}$ making at most $q_{\mathcal{A}}$ oracle calls, we have*

$$\mathrm{Adv}^{\mathrm{ow}}_{\mathcal{S},\mathcal{A}} = \mathcal{O}\left(\ell\left(\frac{2^{k\ell}}{M\ell^\ell}\log\left(\frac{2^{k\ell}}{M\ell^\ell}\right)+1\right)\sqrt[\ell+1]{\frac{q_{\mathcal{A}}(\ell N + M + \sigma)}{2^{k\ell}}} + \frac{1}{N}\right). \tag{10}$$

Here, $\sigma \geq \sum_{i=1}^{\ell} \sigma_i$, where $\sigma_i \in \mathbb{N}$ is an upper bound on the length of the leakage $\boldsymbol{z}_i$ returned by $\mathcal{S}_i$.

Proof. As shown in Section 3, we can assume without loss of generality that $\mathcal{A}$ is deterministic. Let $\epsilon \coloneqq \mathrm{Adv}^{\mathrm{ow}}_{\mathcal{S},\mathcal{A}}$ and $\epsilon' \coloneqq \mathrm{Adv}^{\mathrm{ur}}_{\mathcal{S},\mathcal{A}}$.

Call a pair $(\mathsf{O}, \mathsf{z}) \in \mathrm{Fun}(N^\ell, M) \times \mathrm{Supp}(\mathcal{S}^{\mathsf{O}}_{1,2}) \times \cdots \times \mathrm{Supp}(\mathcal{S}^{\mathsf{O}}_{\ell,2})$ (α, β)-good if $\Pr[\mathrm{OW}^{\mathcal{S},\mathcal{A}} \mid H = \mathsf{O} \wedge \boldsymbol{z} = \mathsf{z}] \geq \alpha\epsilon$, and each source $\mathcal{S}_i$ remains $(N^\ell, \beta \cdot 2^{-k})$-unpredictable when $H = \mathsf{O}$ and $z = \mathsf{z}_i$. In other words, a pair (O, z) as above is (α, β)-bad if either of the following conditions hold:

1. Pair (O, z) is low-win: $(\mathcal{S}, \mathcal{A})$ succeeds with probability less than $\alpha\epsilon$ when $(H, \boldsymbol{z})$ is fixed to (O, z), i.e., $\Pr[\mathrm{OW}^{\mathcal{S},\mathcal{A}} \mid H = \mathsf{O} \wedge \boldsymbol{z} = \mathsf{z}] < \alpha\epsilon$.
2. Pair (O, z) is low-entropy: There exist $i \in [\ell]$ and a predictor $\mathcal{P}_{\mathsf{O},\mathsf{z}}$ with unbounded oracle access such that $\Pr[\mathrm{Pred}^{\mathcal{P}_{\mathsf{O},\mathsf{z}}}_{\mathcal{S}_i} \mid H = \mathsf{O} \wedge z = \mathsf{z}_i] > \beta \cdot 2^{-k}$.

[9] Note that the first condition is easier for an adversary to satisfy, meaning that it defines a stronger version of one-wayness.

Game $\mathrm{OW}^{\mathcal{S},\mathcal{A}}$:
$H \twoheadleftarrow \mathrm{Fun}(N^\ell, M)$
for $i = 1$ to ℓ do $(\boldsymbol{x}_i, \boldsymbol{z}_i) \twoheadleftarrow \mathcal{S}_i^H$
$y \leftarrow H(\boldsymbol{x})$; $\boldsymbol{x}' \twoheadleftarrow \mathcal{A}^H(y, \boldsymbol{z})$
return $(H(\boldsymbol{x}') = y) \wedge \bigwedge_{i=1}^{\ell}((\boldsymbol{x}'_i, \cdot) \in \mathrm{Supp}(\mathcal{S}_i^H))$

Fig. 7. The one-wayness game for the monolithic construction Mono. Here, $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$ is a tuple of sources, and $\mathcal{A}$ is an adversary.

Denote by W_α the set of low-win pairs, $E_{\beta,i}$ the set of pairs that are low-entropy for the ith source, $E_\beta \coloneqq \bigcup_i E_{\beta,i}$ the set of pairs that are low-entropy for at least one source, and let $BP_{\alpha,\beta} \coloneqq W_\alpha \cup E_\beta$ and $GP_{\alpha,\beta}$ be the sets of (α,β)-bad and (α,β)-good pairs. We start by bounding the probability $p_{\mathrm{bp}} \coloneqq \Pr[(H,\boldsymbol{z}) \in BP_{\alpha,\beta}]$ of a pair being (α,β)-bad. By the union bound, we have

$$p_{\mathrm{bp}} \le \Pr[(H,\boldsymbol{z}) \in W_\alpha] + \sum_i \Pr[(H,\boldsymbol{z}) \in E_{\beta,i}]\,.$$

To bound the term $p_{\mathrm{lw}} \coloneqq \Pr[(H,\boldsymbol{z}) \in W_\alpha]$, notice that

$$\begin{aligned}\epsilon = {} & \Pr[\mathrm{UR}^{\mathcal{S},\mathcal{A}} \mid (H,\boldsymbol{z}) \in W_\alpha]\Pr[(H,\boldsymbol{z}) \in W_\alpha] \\ & \qquad + \Pr[\mathrm{UR}^{\mathcal{S},\mathcal{A}} \mid (H,\boldsymbol{z}) \notin W_\alpha]\Pr[(H,\boldsymbol{z}) \notin W_\alpha] \\ & < \alpha\epsilon \cdot 1 + 1 \cdot (1 - p_{\mathrm{lw}})\,,\end{aligned}$$

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

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

from which we obtain $p_{\mathrm{le},i} < 1/\beta$ for every $i \in [\ell]$. Collecting all the terms above, we obtain $p_{\mathrm{bp}} < 1 - (1-\alpha)\epsilon + \ell/\beta$, and therefore

$$p_{\mathrm{gp}} > (1-\alpha)\epsilon - \frac{\ell}{\beta}\,.$$

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). Then $p_{\mathrm{gp}} > 0$, and therefore

$$|GP_{\alpha,\beta}| \ge p_{\mathrm{gp}} \cdot |\mathrm{Fun}(N^\ell, M)| \ge \left((1-\alpha)\epsilon - \frac{\ell}{\beta}\right) \cdot M^{N^\ell} > 0\,.$$

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

$$\Pr[\mathrm{OW}^{\mathcal{S}^*_{\mathsf{O},z},\mathcal{A}} \mid H = \mathsf{O} \wedge \boldsymbol{z} = \mathbf{z}] \geq \alpha\epsilon\,,$$

where $\mathcal{S}^*_{\mathsf{O},\mathbf{z}} \coloneqq (\mathcal{S}^*_{\mathsf{O},\mathbf{z},1}, \ldots, \mathcal{S}^*_{\mathsf{O},\mathbf{z},\ell})$. We will use the flat sources $\mathcal{S}^*_{\mathsf{O},\mathbf{z}}$ and $\mathcal{A}$ to compress O. Define the sets

$$\begin{aligned}
Y_{\mathsf{O},\mathbf{z}} &\coloneqq \{y \in [M] \mid \mathsf{O}(\mathcal{A}^{\mathsf{O}}(y, \mathbf{z})) = y\}\,,\\
X_{\mathsf{O},\mathbf{z}} &\coloneqq \{\boldsymbol{x} \in [N]^\ell \mid \mathsf{O}(\boldsymbol{x}) \in Y_{\mathsf{O},\mathbf{z}}\}\,,\\
S_{\mathsf{O},\mathbf{z},i} &\coloneqq \{x \in [N] \mid (\exists r)(\mathcal{S}^*_{\mathsf{O},\mathbf{z},i}(r) = x)\} = \mathrm{Supp}(\mathcal{S}^*_{\mathsf{O},\mathbf{z},i})\,,\\
Z_{\mathsf{O},\mathbf{z}} &\coloneqq (S_{\mathsf{O},\mathbf{z},1} \times \cdots \times S_{\mathsf{O},\mathbf{z},\ell}) \cap X_{\mathsf{O},\mathbf{z}}\,,\\
U_{\mathsf{O},\mathbf{z}} &\coloneqq \mathsf{O}(S_{\mathsf{O},\mathbf{z},1} \times \cdots \times S_{\mathsf{O},\mathbf{z},\ell}) \cap Y_{\mathsf{O},\mathbf{z}}\,.
\end{aligned}$$

Then observe that, for every $(\mathsf{O}, \mathbf{z}) \in GP_{\alpha,\beta}$,

$$\begin{aligned}
\epsilon_{\mathsf{O},\mathbf{z}} &\coloneqq \Pr[\mathrm{OW}^{\mathcal{S}^*,\mathcal{A}} \mid H = \mathsf{O} \wedge \boldsymbol{z} = \mathbf{z}] = \frac{|Z_{\mathsf{O},\mathbf{z}}|}{|S_{\mathsf{O},\mathbf{z},1} \times \cdots \times S_{\mathsf{O},\mathbf{z},\ell}|} = \frac{|Z_{\mathsf{O},\mathbf{z}}|}{S^\ell}\,,\\
\epsilon'_{\mathsf{O},\mathbf{z}} &\coloneqq \Pr[\mathrm{UR}^{\mathcal{S}^*,\mathcal{A}} \mid H = \mathsf{O} \wedge \boldsymbol{z} = \mathbf{z}] = \frac{|U_{\mathsf{O},\mathbf{z}}|}{|S_{\mathsf{O},\mathbf{z},1} \times \cdots \times S_{\mathsf{O},\mathbf{z},\ell}|} = \frac{|U_{\mathsf{O},\mathbf{z}}|}{S^\ell}\,.
\end{aligned} \tag{11}$$

We briefly comment on the two equalities above. For the first one notice that, if $\mathcal{S}^*$ return a point $\boldsymbol{x} \in Z_{\mathsf{O},\mathbf{z}}$, then $\mathsf{O}(\boldsymbol{x})$ will be inverted, and only if such an $\boldsymbol{x}$ is chosen it will be inverted. We conclude by observing that $\mathcal{S}^*$ is a product of independent uniform distributions. As for the second equality, notice that $\mathcal{A}$ will be successful when given a point in $U_{\mathsf{O},\mathbf{z}}$, and it will additionally win the UR game if the game has chosen exactly the preimage returned by $\mathcal{A}$.

We now define an encoding for the set of (α, β)-good pairs $GP_{\alpha,\beta}$:

Encoding: Given any $(\mathsf{O}, \mathbf{z}) \in GP_{\alpha,\beta}$, we encode it as follows:

1. The advice string $\mathbf{z}$;
2. Descriptions of each $S_{\mathsf{O},\mathbf{z},i}$ as a subset of $[N]$;
3. A description of $Z_{\mathsf{O},\mathbf{z}}$ as a subset of $S_{\mathsf{O},\mathbf{z},1} \times \cdots \times S_{\mathsf{O},\mathbf{z},\ell}$;
4. A description of $U_{\mathsf{O},\mathbf{z}}$ as a subset of $[M]$;
5. The mapping O from $Z_{\mathsf{O},\mathbf{z}}$ to $U_{\mathsf{O},\mathbf{z}}$;
6. The mapping O for all other inputs in lexicographic order.

Decoding: Given an encoding of the above form, we decode it as follows. First, read the leakage vector $\mathbf{z}$ off the codeword. Next, for each flat source, we recover the support $S_{\mathsf{O},\mathbf{z},i}$, we then construct $Z_{\mathsf{O},\mathbf{z}}$ using its encoding as a subset of $S_{\mathsf{O},\mathbf{z},1} \times \cdots \times S_{\mathsf{O},\mathbf{z},\ell}$, and $U_{\mathsf{O},\mathbf{z}}$ from its encoding as a subset of $[M]$. We then use the mapping from $Z_{\mathsf{O},\mathbf{z}}$ to $U_{\mathsf{O},\mathbf{z}}$, and the mapping of the remaining inputs to complete the decoding.

The lengths of each component in the encoding is as follows:

1. σ;
2. $\ell \log N + \sum_{i=1}^{\ell} \log \binom{N}{|S_{\mathsf{O},\mathbf{z},i}|} = \ell \log N + \sum_{i=1}^{\ell} \log \binom{N}{S}$;
3. $\sum_{i=1}^{\ell} \log |S_{\mathsf{O},\mathbf{z},i}| + \log \binom{|S_{\mathsf{O},\mathbf{z},1} \times \cdots \times S_{\mathsf{O},\mathbf{z},\ell}|}{|Z_{\mathsf{O},\mathbf{z}}|} = \ell \log S + \log \binom{S^\ell}{|Z_{\mathsf{O},\mathbf{z}}|}$;
4. $\log M + \log \binom{M}{|U_{\mathsf{O},\mathbf{z}}|}$;
5. $|Z_{\mathsf{O},\mathbf{z}}| \log |U_{\mathsf{O},\mathbf{z}}|$;
6. $(N^\ell - |Z_{\mathsf{O},\mathbf{z}}|) \log M$.

We now further upper-bound the codeword length in our encoding. First, applying Inequality 3 from Lemma 4 and $S \leq N$, we obtain:

$$\sigma + 2\ell \log N + \log M + N^\ell \log M + \sum_{i=1}^{\ell} S \log \frac{eN}{S} + |Z_{\mathsf{O},\mathbf{z}}| \log \frac{eS^\ell |U_{\mathsf{O},\mathbf{z}}|}{|Z_{\mathsf{O},\mathbf{z}}| M} + |U_{\mathsf{O},\mathbf{z}}| \log \frac{eM}{|U_{\mathsf{O},\mathbf{z}}|} .$$

Next, recall that the function $x \mapsto x \log eN/x$ has a maximum at $x = N$, where it takes the value $N \log e$. This allows us to give an upper bound of the terms

$$\ell S \log \frac{eN}{S} \leq \ell N \log e \quad \text{and} \quad |U_{\mathsf{O},\mathbf{z}}| \log \frac{eM}{|U_{\mathsf{O},\mathbf{z}}|} \leq M \log e \,;$$

applying these gives an overall bound on the codeword length of

$$\sigma + \log(N^{2\ell} M) + N^\ell \log M + (\ell N + M) \log e + |Z_{\mathsf{O},\mathbf{z}}| \log \frac{eS^\ell |U_{\mathsf{O},\mathbf{z}}|}{|Z_{\mathsf{O},\mathbf{z}}| M} .$$

Finally, using the expressions for $\epsilon_{\mathsf{O},\mathbf{z}}$ and $\epsilon'_{\mathsf{O},\mathbf{z}}$[10] from (11), the bound becomes

$$\sigma + \log(N^{2\ell} M) + N^\ell \log M + (\ell N + M) \log e + \epsilon_{\mathsf{O},\mathbf{z}} S^\ell \log \frac{e\epsilon'_{\mathsf{O},\mathbf{z}} S^\ell}{\epsilon_{\mathsf{O},\mathbf{z}} M} .$$

We can now apply Lemma 1 to compress the set of (α, β)-good pairs and obtain

$$\log(p_{\mathrm{gp}} M^{N^\ell}) \leq \sigma + \log(N^{2\ell} M) + N^\ell \log M + (\ell N + M) \log e + \mathop{\mathbb{E}}_{GP_{\alpha,\beta}} \left[\epsilon_{\mathsf{O},\mathbf{z}} S^\ell \log \frac{e\epsilon'_{\mathsf{O},\mathbf{z}} S^\ell}{\epsilon_{\mathsf{O},\mathbf{z}} M} \right] ,$$

which, after rearranging terms, becomes

$$\mathop{\mathbb{E}}_{GP_{\alpha,\beta}} \left[\epsilon_{\mathsf{O},\mathbf{z}} S^\ell \log \frac{\epsilon_{\mathsf{O},\mathbf{z}} M}{e\epsilon'_{\mathsf{O},\mathbf{z}} S^\ell} \right] \leq \sigma + \log(p_{\mathrm{gp}}^{-1} N^{2\ell} M) + (\ell N + M) \log e \,. \tag{12}$$

[10] Here we are using $\mathcal{A}$ to invert values, rather than encoding the mapping.

We now distinguish two cases: If $p_{\mathrm{gp}}^{-1}N^{2\ell}M > N^{2\ell+1}M$, then $p_{\mathrm{gp}} < 1/N$, and substituting $p_{\mathrm{gp}} > (1-\alpha)\epsilon - \ell/\beta$ and $\beta = b\ell/((1-\alpha)\epsilon)$ gives

$$\epsilon < \left(1 - \frac{1}{b}\right)^{-1}(1-\alpha)^{-1} \cdot \frac{1}{N} = \mathcal{O}\left(\frac{1}{N}\right), \tag{13}$$

showing that ϵ already satisfies our bound (10) in this case. If the above is not the case, our upper bound (12) becomes

$$\mathop{\mathbb{E}}_{GP_{\alpha,\beta}}\left[\epsilon_{\mathsf{O},\mathbf{z}} S^\ell \log \frac{\epsilon_{\mathsf{O},\mathbf{z}} M}{e\epsilon'_{\mathsf{O},\mathbf{z}} S^\ell}\right] \leq \sigma + \log(N^{2\ell+1}M) + (\ell N + M)\log e\,.$$

Now define sets

$$C_1 \coloneqq \left\{(\mathsf{O},\mathbf{z}) \in GP_{\alpha,\beta} \;\middle|\; \frac{\epsilon_{\mathsf{O},\mathbf{z}} M}{e\epsilon'_{\mathsf{O},\mathbf{z}} S^\ell} < 4\right\}, C_2 \coloneqq \left\{(\mathsf{O},\mathbf{z}) \in GP_{\alpha,\beta} \;\middle|\; \frac{\epsilon_{\mathsf{O},\mathbf{z}} M}{e\epsilon'_{\mathsf{O},\mathbf{z}} S^\ell} \geq 4\right\}.$$

We split the expectation over both sets as follows:

$$\begin{aligned}
\sigma + \log(N^{2\ell+1}M) + (\ell N + M)\log e &\geq \mathop{\mathbb{E}}_{GP_{\alpha,\beta}}\left[\epsilon_{\mathsf{O},\mathbf{z}} S^\ell \log \frac{\epsilon_{\mathsf{O},\mathbf{z}} M}{e\epsilon'_{\mathsf{O},\mathbf{z}} S^\ell}\right] \\
&= \mathop{\mathbb{E}}_{C_1}\left[\epsilon_{\mathsf{O},\mathbf{z}} S^\ell \log \frac{\epsilon_{\mathsf{O},\mathbf{z}} M}{e\epsilon'_{\mathsf{O},\mathbf{z}} S^\ell}\right] \Pr_{GP_{\alpha,\beta}}[(\mathsf{O},\mathbf{z}) \in C_1] \\
&\qquad + \mathop{\mathbb{E}}_{C_2}\left[\epsilon_{\mathsf{O},\mathbf{z}} S^\ell \log \frac{\epsilon_{\mathsf{O},\mathbf{z}} M}{e\epsilon'_{\mathsf{O},\mathbf{z}} S^\ell}\right] \Pr_{GP_{\alpha,\beta}}[(\mathsf{O},\mathbf{z}) \in C_2] \\
&\geq \mathop{\mathbb{E}}_{C_1}\left[\epsilon_{\mathsf{O},\mathbf{z}} S^\ell \log \frac{\epsilon_{\mathsf{O},\mathbf{z}} M}{e\epsilon'_{\mathsf{O},\mathbf{z}} S^\ell}\right] \Pr_{GP_{\alpha,\beta}}[(\mathsf{O},\mathbf{z}) \in C_1] + 2S^\ell \mathop{\mathbb{E}}_{C_2}[\epsilon_{\mathsf{O},\mathbf{z}}] \Pr_{GP_{\alpha,\beta}}[(\mathsf{O},\mathbf{z}) \in C_2] \\
&\geq \mathop{\mathbb{E}}_{C_1}\left[\epsilon_{\mathsf{O},\mathbf{z}} S^\ell \log \frac{M}{eS^\ell}\right] \Pr_{GP_{\alpha,\beta}}[(\mathsf{O},\mathbf{z}) \in C_1] + 2S^\ell \mathop{\mathbb{E}}_{C_2}[\epsilon_{\mathsf{O},\mathbf{z}}] \Pr_{GP_{\alpha,\beta}}[(\mathsf{O},\mathbf{z}) \in C_2] \quad (14) \\
&= S^\ell \log \frac{M}{eS^\ell} \mathop{\mathbb{E}}_{C_1}[\epsilon_{\mathsf{O},\mathbf{z}}] \Pr_{GP_{\alpha,\beta}}[(\mathsf{O},\mathbf{z}) \in C_1] + 2S^\ell \mathop{\mathbb{E}}_{C_2}[\epsilon_{\mathsf{O},\mathbf{z}}] \Pr_{GP_{\alpha,\beta}}[(\mathsf{O},\mathbf{z}) \in C_2]\,,
\end{aligned}$$

where (14) follows from $\epsilon_{\mathsf{O}} \geq \epsilon'_{\mathsf{O}}$, as for all O, if an adversary wins the unrecoverability game, it will also win the one-wayness game. We can rearrange this to give the following bound:

$$\begin{aligned}
\mathop{\mathbb{E}}_{C_2}[\epsilon_{\mathsf{O},\mathbf{z}}] \Pr_{GP_{\alpha,\beta}}[(\mathsf{O},\mathbf{z}) \in C_2] \leq{}& \frac{\sigma + \log(N^{2\ell+1}M) + (\ell N + M)\log e}{2S^\ell} \\
&+ \frac{1}{2}\log\left(\frac{eS^\ell}{M}\right) \mathop{\mathbb{E}}_{C_1}[\epsilon_{\mathsf{O},\mathbf{z}}] \Pr_{GP_{\alpha,\beta}}[(\mathsf{O},\mathbf{z}) \in C_1]\,.
\end{aligned} \tag{15}$$

We are now ready to bound ϵ. First, notice that

$$\begin{aligned}
\epsilon = \mathbb{E}[\epsilon_{\mathsf{O},\mathbf{z}}] = \mathop{\mathbb{E}}_{C_1}[\epsilon_{\mathsf{O},\mathbf{z}}] \Pr[(\mathsf{O},\mathbf{z}) \in C_1] + \mathop{\mathbb{E}}_{C_2}[\epsilon_{\mathsf{O},\mathbf{z}}] \Pr[(\mathsf{O},\mathbf{z}) \in C_2] \\
+ \mathop{\mathbb{E}}_{BP_{\alpha,\beta}}[\epsilon_{\mathsf{O},\mathbf{z}}] \Pr[(\mathsf{O},\mathbf{z}) \in BP_{\alpha,\beta}]\,.
\end{aligned}$$

For the last term we have

$$\begin{aligned}\mathop{\mathbb{E}}_{BP_{\alpha,\beta}}[\epsilon_{\mathsf{O},\mathbf{z}}]\Pr[(\mathsf{O},\mathbf{z})\in BP_{\alpha,\beta}] &\le \mathop{\mathbb{E}}_{W_\alpha}[\epsilon_{\mathsf{O},\mathbf{z}}]\Pr[(\mathsf{O},\mathbf{z})\in W_\alpha]\\ &\quad+\sum_{i=1}^{\ell}\mathop{\mathbb{E}}_{E_{\beta,i}}[\epsilon_{\mathsf{O},\mathbf{z}}]\Pr[(\mathsf{O},\mathbf{z})\in E_{\beta,i}]\\ &< \alpha\epsilon\cdot 1+\sum_{i=1}^{\ell}1\cdot\frac{1}{\beta}=\alpha\epsilon+\frac{\ell}{\beta}=\epsilon\left(\alpha+\frac{1-\alpha}{b}\right),\end{aligned}$$

meaning that our equality above becomes

$$\begin{aligned}\epsilon(1-\alpha)\left(1-\frac{1}{b}\right) &\le \mathop{\mathbb{E}}_{C_1}[\epsilon_{\mathsf{O},\mathbf{z}}]\Pr[(\mathsf{O},\mathbf{z})\in C_1]+\mathop{\mathbb{E}}_{C_2}[\epsilon_{\mathsf{O},\mathbf{z}}]\Pr[(\mathsf{O},\mathbf{z})\in C_2]\\ &\le \mathop{\mathbb{E}}_{C_1}[\epsilon_{\mathsf{O},\mathbf{z}}]\Pr[(\mathsf{O},\mathbf{z})\in C_1]+\mathop{\mathbb{E}}_{C_2}[\epsilon_{\mathsf{O},\mathbf{z}}]\Pr_{GP_{\alpha,\beta}}[(\mathsf{O},\mathbf{z})\in C_2]\Pr[(\mathsf{O},\mathbf{z})\in GP_{\alpha,\beta}].\end{aligned}$$

Plugging in Inequality (15) then gives

$$\begin{aligned}&\epsilon(1-\alpha)\left(1-\frac{1}{b}\right)\\ &\quad\le \mathop{\mathbb{E}}_{C_1}[\epsilon_{\mathsf{O},\mathbf{z}}]\Pr[(\mathsf{O},\mathbf{z})\in C_1]+\left[\frac{\sigma+\log(N^{2\ell+1}M)+(\ell N+M)\log e}{2S^\ell}\right.\\ &\qquad\left.+\frac{1}{2}\log\left(\frac{eS^\ell}{M}\right)\mathop{\mathbb{E}}_{C_1}[\epsilon_{\mathsf{O},\mathbf{z}}]\Pr_{GP_{\alpha,\beta}}[(\mathsf{O},\mathbf{z})\in C_1]\right]\Pr[(\mathsf{O},\mathbf{z})\in GP_{\alpha,\beta}]\\ &\quad\le \frac{\sigma+\log(N^{2\ell+1}M)+(\ell N+M)\log e}{2S^\ell}\\ &\qquad+\frac{1}{2}\left(2+\log\left(\frac{eS^\ell}{M}\right)\right)\mathop{\mathbb{E}}_{C_1}[\epsilon_{\mathsf{O},\mathbf{z}}]\Pr[(\mathsf{O},\mathbf{z})\in C_1]\\ &\quad\le \frac{\sigma+\log(N^{2\ell+1}M)+(\ell N+M)\log e}{2S^\ell}\\ &\qquad+\log\left(\frac{4eS^\ell}{M}\right)\left(\frac{2eS^\ell}{M}\right)\mathop{\mathbb{E}}_{C_1}[\epsilon'_{\mathsf{O},\mathbf{z}}]\Pr[(\mathsf{O},\mathbf{z})\in C_1].\end{aligned}\tag{16}$$

We now once more distinguish two cases: If $4eS^\ell/M<1$, then substituting $S=2^k/\beta$ and $\beta=b\ell/((1-\alpha)\epsilon)$ we obtain $\epsilon^\ell<(b^\ell\ell^\ell M)/(4e2^{k\ell}(1-\alpha)^\ell)$, meaning

$$\epsilon<\sqrt[\ell]{\frac{b^\ell\ell^\ell M}{4e2^{k\ell}(1-\alpha)^\ell}}=\frac{b\ell}{1-\alpha}\sqrt[\ell]{\frac{M}{4e2^{k\ell}}}=\mathcal{O}\left(\ell\sqrt[\ell]{\frac{M}{2^{k\ell}}}\right).\tag{17}$$

In particular, ϵ satisfies our bound (10) in this case. If not, then the last term in the chain of inequalities (16) is positive, so that we can continue (16) as

$$\begin{aligned}\epsilon(1-\alpha)\left(1-\frac{1}{b}\right) &\le \frac{\sigma+\log(N^{2\ell+1}M)+(\ell N+M)\log e}{2S^\ell}\\ &\quad+\log\left(\frac{4eS^\ell}{M}\right)\left(\frac{2eS^\ell}{M}\right)\epsilon',\end{aligned}$$

and substituting for S and β gives

$$\begin{aligned}\epsilon(1-\alpha)\left(1-\frac{1}{b}\right) &\leq \frac{b^\ell \ell^\ell}{2\cdot 2^{k\ell}(1-\alpha)^\ell \epsilon^\ell}(\sigma + \log(N^{2\ell+1}M) + (\ell N + M)\log e) \\ &\quad + \log\left(\frac{4e\cdot 2^{k\ell}(1-\alpha)^\ell\epsilon^\ell}{Mb^\ell\ell^\ell}\right)\left(\frac{2e\cdot 2^{k\ell}(1-\alpha)^\ell\epsilon^\ell}{Mb^\ell\ell^\ell}\right)\epsilon' \\ &\leq \frac{b^\ell \ell^\ell}{2\cdot 2^{k\ell}(1-\alpha)^\ell \epsilon^\ell}(\sigma + \log(N^{2\ell+1}M) + (\ell N + M)\log e) \\ &\quad + \log\left(\frac{4e\cdot 2^{k\ell}}{M\ell^\ell}\right)\left(\frac{2e\cdot 2^{k\ell}}{M\ell^\ell}\right)\epsilon',\end{aligned}$$

or, equivalently,

$$\begin{aligned}\epsilon^{\ell+1}(1-\alpha)\left(1-\frac{1}{b}\right) &\leq \frac{b^\ell \ell^\ell}{2\cdot 2^{k\ell}(1-\alpha)^\ell}(\sigma + \log(N^{2\ell+1}M) + (\ell N + M)\log e) \\ &\quad + \log\left(\frac{4e\cdot 2^{k\ell}}{M\ell^\ell}\right)\left(\frac{2e\cdot 2^{k\ell}\epsilon^\ell}{M\ell^\ell}\right)\epsilon'. \quad (18)\end{aligned}$$

Inequality (18) means that one of two cases must hold: Either

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

meaning

$$\begin{aligned}\epsilon &= \mathcal{O}\left(\sqrt[\ell+1]{\frac{b^\ell \ell^\ell}{2\cdot 2^{k\ell}(1-\alpha)^\ell}(\sigma + \log(N^{2\ell+1}M) + (\ell N + M)\log e)}\right) \\ &= \mathcal{O}\left(\ell\sqrt[\ell+1]{\frac{\ell N + M + \sigma}{2^{k\ell}}}\right).\end{aligned} \quad (19)$$

If the above is not the case, then (18) implies that

$$\epsilon^{\ell+1} = \mathcal{O}\left(\log\left(\frac{4e\cdot 2^{k\ell}}{M\ell^\ell}\right)\left(\frac{2e\cdot 2^{k\ell}\epsilon^\ell}{M\ell^\ell}\right)\epsilon'\right),$$

which, using the bound from Theorem 2, yields

$$\epsilon = \mathcal{O}\left(\frac{2^{k\ell}}{M\ell^\ell}\log\left(\frac{2^{k\ell}}{M\ell^\ell}\right)\ell\sqrt[\ell+1]{\frac{q_{\mathcal{A}}}{2^{k\ell}}(\sigma+\ell N)}\right). \quad (20)$$

Our overall bound (10) is then a combination of the terms (13), (17), (19) and (20). □