Electronic Colloquium on Computational Complexity, Report No. 10 (2022)

(Nondeterministic) Hardness vs. Non-Malleability

Marshall Ball[1] [⋆], Dana Dachman-Soled[2] [⋆⋆], and Julian Loss[3] [⋆⋆⋆]

[1] New York University
marshall@cs.nyu.edu
[2] University of Maryland
danadach@umd.edu
[3] CISPA Helmholtz Center for Information Security
lossjulian@gmail.com

Abstract. We present the first truly explicit constructions of *non-malleable codes* against tampering by bounded polynomial size circuits. These objects imply unproven circuit lower bounds and our construction is secure provided E requires exponential size nondeterministic circuits, an assumption from the derandomization literature.

Prior works on NMC for polysize circuits, either required an untamperable CRS [Cheraghchi, Guruswami ITCS'14; Faust, Mukherjee, Venturi, Wichs EUROCRYPT'14] or very strong cryptographic assumptions [Ball, Dachman-Soled, Kulkarni, Lin, Malkin EUROCRYPT'18; Dachman-Soled, Komargodski, Pass CRYPTO'21]. Both of works in the latter category only achieve non-malleability with respect to efficient distinguishers and, more importantly, utilize cryptographic objects for which no provably secure instantiations are known outside the random oracle model. In this sense, none of the prior yields fully explicit codes from non-heuristic assumptions. Our assumption is not known to imply the existence of one-way functions, which suggests that cryptography is unnecessary for non-malleability against this class.

Technically, security is shown by *non-deterministically* reducing polynomial size tampering to split-state tampering. The technique is general enough that it allows us to to construct the first *seedless non-malleable extractors* [Cheraghchi, Guruswami TCC'14] for sources sampled by polynomial size circuits [Trevisan, Vadhan FOCS'00] (resp. recognized by polynomial size circuits [Shaltiel CC'11]) and tampered by polynomial size circuits. Our construction is secure assuming E requires exponential size Σ_4-circuits (resp. Σ_3-circuits), this assumption is the state-of-the-art for extracting randomness from such sources (without non-malleability).

We additionally observe that non-malleable codes and non-malleable secret sharing [Goyal, Kumar STOC'18] are essentially equivalent with respect to polynomial size tampering. In more detail, assuming E is hard for exponential size nondeterministic circuits, any efficient secret sharing scheme can be made non-malleable against polynomial size circuit tampering.

Unfortunately, all of our constructions only achieve inverse polynomial (statistical) security. Extending a result from [Applebaum, Artemenko, Shaltiel, Yang CC'16] we show it is impossible to do better using black-box reductions. However, we extend the notion of relative error from [Applebaum, Artemenko, Shaltiel, Yang CC'16] to non-malleable extractors and show that they can be constructed from similar assumptions. We additionally observe that relative-error non-malleable extractors can be utilized to render a broad class of cryptographic primitives tamper and leakage resilient, while preserving negligible security guarantees.

[⋆] Part of this work was done while the author was a student at Columbia University and a postdoc at University of Washington. This material is based upon work supported by the National Science Foundation under Grant #2030859 to the Computing Research Association for the CIFellows Project. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation nor the Computing Research Association.

[⋆⋆] Supported in part by NSF grants #CNS-1933033, #CNS-1453045 (CAREER), and by financial assistance awards 70NANB15H328 and 70NANB19H126 from the U.S. Department of Commerce, National Institute of Standards and Technology.

[⋆⋆⋆] Part of this work was done while the author was a postdoc at the University of Maryland and Carnegie Mellon University.

ISSN 1433-8092

Table of Contents

1 Introduction

This work focuses on mitigating polynomial size circuit tampering attacks via constructing two kinds of fundamental objects: *non-malleable* codes (NMC) and seedless *non-malleable* extractors (NME) . In the coding setting, non-malleability (roughly) guarantees that the output of the decoding algorithm on a codeword is independent of the output of the decoding algorithm on a tampered version of the codeword. Similarly in the seedless extractor setting, non-malleability guarantees that the output of the extractor on a sample drawn from a high min-entropy source remains uniform random, even conditioned on the output of the extractor on a tampered version of the sample.

A recent thrust of research has focused on constructing explicit (efficient) NMC and NME for broad and natural classes of tampering. Perhaps the most natural class of tampering functions, is tampering by polynomial size circuits. Unfortunately, a simple argument shows that any (seedless) non-malleable code (resp. extractor) resilient to arbitrary polynomial size circuit tampering cannot be decoded (resp. evaluated) in polynomial time. The next best thing would be a non-malleable code (resp. seedless extractor) that can be encoded/decoded (resp. evaluated) in polynomial time that is resilient to *bounded* polynomial size circuit tampering—tampering by circuits of size at most n^c where c is a constant fixed a priori. In this work, we are interested in constructing *explicit* (i.e. computable by polynomial time Turing machines) objects that are resilient to such tampering attacks.

This tampering class has been studied extensively in the non-malleable code literature and prior work constructing NMC for bounded polynomial size circuit tampering can be collected into two categories, both of which fail to provide explicit constructions:

1. *Unconditionally secure constructions via the probabilistic method.* [CG14a, FMVW14] show that efficiently computable non-malleable codes for bounded polynomial size circuit tampering exist. These constructions can alternately be cast as explicit codes in an (untamperable) common reference string (CRS) model, or as codes with efficient Monte Carlo style constructions.

   Computational assumptions are *needed* for any explicit construction (without a CRS) since security of the non-malleable code implies circuit lower bounds—existence of an explicit hard-on-average problem for circuits of size n^c—a question that is still wide open in the complexity literature.

   Unfortunately even under strong assumptions, it is unclear how to derandomize these constructions completely. (See beginning of Section 1.3 for further discussion.)

2. *Computationally secure constructions via strong cryptographic assumptions.* [BDK$^+$19, DKP20, DSKP21] leverage a variety of non-standard cryptographic assumptions to construct non-malleable codes for bounded polynomial size circuit tampering (no CRS) with computational security guarantees.

   While some assumptions are necessary (as mentioned above), these works utilize very powerful computational assumptions. Most importantly, these works (among other assumptions[4]) require the existence of objects that we currently only know how to provably instantiate with random oracles (e.g. [BDK$^+$19] uses P-certificates and [DKP20, DSKP21] uses keyless multi-collision resistant hash functions).

   Consequently, these works only yield explicit constructions of non-malleable codes under heuristic assumptions.[5] Additionally, these works fall short of providing statistical security guarantees.

In summary, none of the prior constructions are fully explicit.

In this work, we employ an assumption from the derandomization literature to construct *explicit* non-malleable codes and seedless non-malleable extractors resilient to bounded polynomial tampering. Our non-malleable codes in particular are secure under a hardness conjecture introduced in the context of derandomizing AM: there is a language that can be computed in exponential deterministic time that requires exponential size nondeterministic circuits.

In Section 1.1, we describe the hardness assumptions we use to construct our codes and extractors. In Section 1.2, we discuss our main results, barriers to improving them, and applications. Finally in Section 1.3, we illustrate our primary technique through a simple yet illuminating example and describe how the ideas can be extended to prove our main results.

1.1 Hardness assumptions for nondeterministic and Σ_i-circuits

We begin with a formal definition of nondeterministic circuits and then introduce the corresponding assumption about their limitations.

Definition 1.1 (Nondeterministic circuit)**.** *A* nondeterministic circuit *C is a circuit with "non-deterministic" inputs, in addition to the usual inputs. We say C evaluates to* 1 *on x if and only if there exists an assignment, w, to the non-deterministic input wires such that the circuit, evaluated deterministically on input (x, w) outputs* 1*.*

[4] In addition to a variety of subexponentially secure variants of standard cryptographic assumptions, the work of [DKP20, DSKP21] also crucially requires a specific number-theoretic assumption (the non-uniform subexponential hardness of the repeated squaring assumption), while the work of [BDK$^+$19] needs the same derandomization assumption in this work.

[5] E.g. [BKP18] suggests possibly instantiating keyless multi-collision resistant hash with an unstructured hash, such as SHA-2 (extended to arbitrarily large keys), with keys chosen according to digits of π. Establishing the security of any such candidate is well beyond our current techniques, as we cannot even base the security of (extended) SHA-2 with randomly chosen keys to a natural computational problem.

Assumption 1 (E requires exponential size nondeterministic circuits)**.** *There is a language $L \in \mathsf{E} = \mathsf{DTIME}(2^{O(n)})$ and a constant γ such that for sufficiently large n nondeterministic circuits of size $2^{\gamma n}$ fail to decide L on inputs of length n.*

Informally, the above assumption says that non-uniformity and non-determinism do not always imply significant speed-ups of uniform deterministic computations. For some of the results in this work, we require assumptions that hold even for (non-deterministic) NP circuits or Σ_i circuits. Before we state the assumption, we provide a formal definition of these objects.

Definition 1.2. *An oracle circuit $C^{(\cdot)}$ is a circuit which in addition to the standard gates uses an additional gate (which may have large fan in). When instantiated with a specific boolean function A, C^A is the circuit in which the additional gate is A. Given a boolean function $A(x)$, an A-circuit is a circuit that is allowed to use A gates (in addition to the standard gates). An NP-circuit is a SAT-circuit (where SAT is the satisfiability function) a Σ_i-circuit is an A-circuit where A is the canonical Σ_i^P-complete language. We take the size of a circuit to be the total number of wires and gates.*[6]

We now state the corresponding set of assumptions:

Assumption 2 (E requires exponential size NP (resp. Σ_i) circuits)**.** *There is a language $L \in \mathsf{E} = \mathsf{DTIME}(2^{O(n)})$ and a constant γ such that for sufficiently large n, NP (resp. Σ_i) circuits of size $2^{\gamma n}$ fail to compute the characteristic function of L on inputs of length n.*

Hardness assumptions against nondeterministic/NP/Σ_i circuits appear in the literature in various contexts of complexity theory and derandomization [BOV03, Dru13, FL97, GW02, GST03, KvM02, MV05, SU05, SU06, SU09, TV00]. As noted in [AASY16], such assumptions can be seen as the nonuniform and scaled-up versions of assumptions of the form $\mathsf{EXP} \neq \mathsf{NP}$ or $\mathsf{EXP} \neq \Sigma_2^{\mathsf{P}}$. While very strong, falsification of one of these assumptions would yield surprising implications on the relationship between standard complexity classes, thus creating a win-win situation: Either the construction based on these assumptions is secure, or a breakthrough result has been achieved that changes our current understanding of the power of nonuniformity and nondeterminism. Further, since assumptions of the above type on the strength of E are *worst-case assumptions*, we can directly instantiate constructions based on these assumptions with any E-complete problem.

Finally, we highlight that, so far as we know, this assumption is orthogonal to standard cryptographic assumptions such as one-way functions and, consequently, may hold even if cryptography does not exist.

1.2 Our Results

In this section, we summarize the main results of this paper. We present our results for non-malleable codes (Section 1.2) and non-malleable extractors before elaborating on various aspects of the results: necessary assumptions for such constructions, and barriers to achieving negligible security guarantees. We then discuss how to circumvent these barriers in a manner that ultimately has applications to tamper and leakage resilient cryptography (with negligible security guarantees). Finally, we state an equivalence between non-malleable codes and non-malleable secret sharing in the context of polynomial size circuit tampering.

Non-Malleable Codes We begin by stating our results for non-malleable codes:

Theorem 1.3 (Informal (See Lemma 4.1 and Theorem 4.2 for formal versions)). *If E requires exponential size nondeterministic circuits, then for every constant c, and for sufficiently large k, there is an explicit, efficient, n^{-c}-secure non-malleable code for k-bit messages, with codeword length $n = \mathsf{poly}(k)$, resilient to tampering by n^c-size circuits.*

We construct our codes by "fooling" non-malleable codes for *split-state tampering* (with special properties).

Split-state tampering functions may manipulate the left and right halves of a codeword arbitrarily, but independently (i.e. functions such that $(c_L, c_R) \mapsto (f_L(c_L), f_R(c_R))$ for some f_L, f_R). Leakage-resilient split-state tampering allows each tampered codeword half to depend on bounded leakage from the opposite

[6] Note that an NP-circuit is different than a nondeterministic circuit. The former is a nonuniform analogue of P^{NP} (which contains coNP) while the latter is an analogue of NP.

codeword half. In addition to split-state NMC, we also use a pseudorandom generator (PRG) for nondeterministic circuits, where $c' > c$ is a constant. In particular, we require that the PRG, G, is secure even when given the seed (seed extending), i.e. no nondeterministic circuit of bounded polynomial size can distinguish $G(s)$ from uniform *and* s is a prefix of $G(s)$.

Given a (leakage-resilient) split-state non-malleable code, with necessary properties and a a seed-extending pseudorandom PRG for nondeterministic circuits, G, we encode a message x by sampling the following:

$$(s, c_R) \text{ such that } (G(s), c_R) \text{is a split-state encoding of } x.$$

While we refer the reader to the technical overview (Section 1.3) for a more detailed sketch, we provide here some intuition for security:

1. We assume towards contradiction that (s, c_R) is *malleable* and fix the corresponding poly-size tampering function g which is *not* split-state and violates non-malleability.
2. We transform g into a split-state tampering function f_L, f_R on (c_L, c_R), where (1) f_L is *unbounded*, relies on $|s|$ bits of leakage from c_R and returns some c'_L, (2) f_R is efficient, relies on $|s|$ bits of leakage from c_L and returns c'_R. Crucially, split-state tampering function (f_L, f_R) is guaranteed to break non-malleability when $c_L = (s||y) = G(s)$.
3. Since (c_L, c_R) is a leakage-resilient split-state non-malleable code when c_L is uniform random, then when c_L is random, every tampering functon (f'_L, f_R) *fails* to break non-malleability, even when f'_L is unbounded and chooses its output c'_L in the "optimal" way.
4. We construct an Arthur-Merlin protocol (with bounded poly-size Arthur), that distinguishes between input c_L being random or pseudorandom. Such a protocol can then be transformed into a non-deterministic polynomial bounded circuit.
5. Intuitively, Arthur can efficiently compute all the values needed to simulate the tampering experiment except for c'_L, which is obtained from Merlin. Specifically, on input c_L, Arthur samples c_R, and computes $c'_R = f_R(c_R)$, as well as the leakage on c_R. Arthur sends c_L and the leakage on c_R to Merlin who responds with c'_L. If c_L is pseudorandom, then an honest Merlin will return $c'_L = f_L(c_L)$, and, with Merlin's help, Arthur can check that non-malleability is violated with this c'_L. If c_L is random, then despite any response $c'_L = f'_L(c_L)$ from Merlin, non-malleability will *not* be violated, and a dishonest Merlin cannot convince Arthur otherwise.

Non-Malleable Extractors We next shift our focus to the case of seedless non-malleable extractors for computational sources with sufficient min-entropy[7] and for tampering with bounded polynomial size circuits. We consider two types of computational sources:

- **Samplable sources:** These are distributions that can be generated by bounded polynomial size circuits that are given uniform random coins as input. Specifically, the source distribution X is equivalent to $C(U_r)$, the distribution generated by some circuit C of size n^c on input uniform randomness of length r bits.

  Extracting from this class of sources was first considered by Trevisan and Vadhan [TV00]. In 1986, Levin [Lev86] argued that this class reasonably captures sources arising in nature.[8]

  A non-malleable extractor for this class yields non-malleable cryptography resilient tampering attacks on the very entropy sources used for key generation.

  As an alternate motivation, one can consider a natural, albeit restricted, online extraction setting: imagine a natural source over a time interval as (X_1, X_2) where X_1 is efficiently (and randomly) transformed to X_2 with the promise that X_1 and X_2 have entropy independent of the other. Then any non-malleable extractor for samplable sources with respect to polynomial size tampering, Ext, can extract from such as source online, i.e. $\mathsf{Ext}(X_1), \mathsf{Ext}(X_2)$ is approximately uniform.[9]

[7] Min-entropy measures the unpredictability of a random variable. In particular, X has min-entropy k if x in the support of X, $\Pr[X = x] \leq 2^{-k}$.

[8] Sources sampled by polynomial size *quantum* circuits seem a more appropriate model for physical sources of randomness. Nonetheless, (classical) sampable sources are an interesting and important subclass.

[9] Note that with a random seed it is easy to extract from say X_1 conditioned on X_2.

- **Recognizable sources:** These are uniform distributions over the set of inputs accepted by some polynomial sized circuit. Specifically, the source distribution X is uniform over $\{x : C(x) = 1\}$, where C is a circuit of size n^c.

  Extracting from this class of sources was first considered by Shaltiel [Sha11] in the context of derandomization. This class corresponds with sources about which some efficiently computable leakage is known.

  As we will see, *non-malleable* extractors for recognizable sources and polynomial size tampering provide a natural, generic means constructing non-malleable, leakage-resilient cryptography.

Theorem 1.4 (Informal (See Theorem 3.5 for formal version)). *If E requires exponential size Σ_4-circuits, then for every constant c, there is an explicit n^{-c}-secure seedless non-malleable extractor for sources $X \in \{0,1\}^n$ samplable by n^c size circuits with linear min-entropy, that outputs $\Omega(\frac{n \log\log(n)}{\log(n)})$ bits and is resilient to tampering by n^c-size circuits.*

Similarly to our non-malleable codes, we construct our non-malleable extractors by "fooling" *(seedless) two-source non-malleable extractors.*

Roughly, a two-source non-malleable extractor, 2NMExt, can extract randomness from two-independent sources (with sufficient min-entropy) even after seeing the output of the extractor invoked on input generated by independently (and arbitrarily) tampering each source.

Our construction of a non-malleable extractor for samplable sources and polynomial size tampering follows. Let $\mathsf{Ext}_{\text{samp}}$ be an extractor for samplable sources, 2NMExt an (efficient) two-source non-malleable extractor, and G a PRG for nondeterministic NP-circuits, then given a samplable source X. The idea is to extract a seed with the samplable extractor and then use the seed to "fool" the two-source non-malleable extractor in a similar manner to the non-malleable code construction above.

- Extract a seed $s = \mathsf{Ext}_{\text{samp}}(X)$.
- Output $\text{2NMExt}(G(s), X)$.

The high-level idea of the proof is similar to the outline for the non-malleable code proof. An added difficulty here over our non-malleable code analysis (responsible for the stronger assumption on the PRG) is that Arthur again receives either pseudorandom $(s||y) = G(s)$ or random $(s||y)$ as input, but now must sample a source, X, that is consistent with its input, i.e. sample X such that $\mathsf{Ext}_{\text{samp}}(X) = s$. Arthur can do this with a bounded poly-size circuit, given an added level of non-determinism.

The above result is obtained by first constructing "relaxed" seedless non-malleable extractors for $n^{c'}$ samplable sources and n^c tampering (by "relaxed" we mean restricting the tampering function to have no fixed points) in Section 3.1, and then presenting a generic transformation from relaxed seedless non-malleable extractors for $n^{c'}$ samplable sources and n^c tampering to seedless non-malleable extractors for n^c samplable sources and n^c tampering in Section 3.3.

We obtain a similar result for recognizable sources:

Theorem 1.5 (Informal (See Theorem 3.6 for formal version)). *If E requires exponential size Σ_3-circuits, then for every constant c, there is an explicit n^{-c}-secure seedless non-malleable extractor for sources $X \in \{0,1\}^n$ recognizable by n^c size circuits with linear min-entropy, that outputs $\Omega(\frac{n \log\log(n)}{\log(n)})$ bits and is resilient to tampering by n^c-size circuits.*

We note that the assumption that E requires exponential size Σ_4-circuits (resp. E requires exponential size Σ_3-circuits) is inherited from the seedless extractor for samplable (resp. recognizable) sources of [AASY16] that is used as a building block in our construction. Assuming the existence of a seedless extractor for samplable (resp. recognizable) sources, our construction requires only the weaker assumption that E requires exponential size nondeterministic NP circuits.

Before presenting a technical overview of the main ideas of our constructions, we discuss the relationship between our positive results and known negative results from the literature.

On the feasibility of explicit codes from minimal assumptions. It is known that explicit non-malleable codes for circuits of size $O(n^c)$ imply explicit languages that are hard on average for circuits of size $O(n^c)$.[10] Due to the limitations in current techniques for proving unconditional circuit lower bounds, it is therefore unlikely to construct explicit codes for such a tampering class, unconditionally. Yet, one might still hope to construct codes by assuming minimal circuit lower bounds (i.e. assuming there exists a language computable in time n^d, for some $d > c$, that is hard on average for $O(n^c)$-size circuits). Unfortunately, Ball et al. [BDKM20] showed a barrier to proving such a theorem. In particular, they ruled out constructions of non-malleable codes where the *security proof*–which is a *reduction* from breaking the above assumption to breaking the non-malleable code— makes *black box* usage of the tampering adversary. This implies that either radically different proof approaches are necessary (that make use of non-black box methods) or stronger assumptions (beyond the minimal one discussed above) are needed.

Our present result skirts this lower bound by taking the second approach of stronger assumptions. Specifically, the techniques of [BDKM20] rule out non-black box reductions when the constructed non-malleable code is resilient against some class $\mathcal{C}$ and the underlying assumption is hard for the *same* class $\mathcal{C}$ of circuits. In this work, our tampering class consists of small *deterministic* circuits, but our assumption is stronger and requires hardness for small *nondeterministic* circuits.

On the necessity of $1/\mathsf{poly}$*-indistinguishability.* One could hope to construct non-malleable extractors and non-malleable codes with *negligible* error from the types of assumptions we consider in this work–i.e. that E requires exponential size Σ_i-circuits. Unfortunately, for the case of non-malleable extractors for samplable or recognizable distributions, barriers to achieving such a result were already shown in the work of Applebaum et al. [AASY16]. Specifically, they rule out certain types of black-box reductions from functions that are $(1/2 + \delta)$-hard (where δ is a small constant) for n^d-size Σ_i-circuits to extractors for distributions that are samplable or recognizable by size n^c circuits (where $c \leq d$ are constants), and that achieve negligible error. As a consequence, their results rule out reductions from the assumption that E requires exponential size Σ_i-circuits In Appendix B, we extend the results of Applebaum et al. [AASY16] to rule out black-box reductions from any function f that is $(1/2 + \delta)$-hard for n^d-size Σ_i-circuits to efficient, 1-bit non-malleable codes resilient to tampering by by size n^c circuits (where $c \leq d$ are constants), and that achieve negligible error.[11] Since f as above can be constructed from the scaled down and padded characteristic function of some (average case hard) language in E, it means that if one can compute the characteristic function of an E-complete language on all inputs (i.e. break the worst-case hardness of an E-complete language), then one can compute f on average (with probability $1/2 + \delta$). Thus, our results also rule out reductions from the assumption that E is (worst-case) hard for exponential size Σ_i-circuits.

We note that there are differences in the class of reductions ruled out by our result in Theorem B.2 and Corollary B.3 and the corresponding results of Applebaum et al. [AASY16]: Our result allows *function-specific* and *non*-security parameter-preserving reductions. On the other hand, our results require the assumption that there is a function that is hard for n^d-size Σ_i-circuits and rule out only efficient constructions of non-malleable codes (where encode/decode are polynomial time), while the results of Applebaum et al. [AASY16] are unconditional and rule out even inefficient constructions. Please see Remarks 1, and 2 for further discussion.

Taken together, the results of Applebaum et al. [AASY16] together with our new results for non-malleable codes in Appendix B, indicate that significantly new proof techniques are necessary to construct non-malleable extractors and non-malleable codes with *negligible* error from the assumption that E requires exponential size Σ_i-circuits.

Partially bypassing the impossibility via "relative error." The above results indicate that it is inherently difficult to construct non-malleable extractors with negligible error under non-deterministic reductions, where error is measured in terms of *statistical distance*. Another measure of closeness between distributions is known as *relative error*. Specifically, relative error α between a pair of distributions $\mathcal{D}_1$, $\mathcal{D}_2$ requires that for every element x in the support of $\mathcal{D}_1$,

$$(1 - \alpha)Pr_{\mathcal{D}_2}[x] \leq Pr_{\mathcal{D}_1}[x] \leq (1 + \alpha)Pr_{\mathcal{D}_2}[x].$$

[10] In particular, the Decode function is hard with respect to the distribution formed by encoding a random bit. If this wasn't the case, one could attack by computing the encoded value and outputting a fixed encoding of the opposite bit.

[11] Note that ruling out reductions to 1-bit non-malleable codes also rules out reductions to k-bit non-malleable codes.

In this case, even if α is *non-negligible*, the above guarantee is still useful for achieving negligible security.

Applebaum et al. [AASY16] introduced a notion of *relative-error* extractors, observing that if the output of the extractor is $1/\mathsf{poly}$-close to uniform with relative error, then every event occurs w.r.t. the output distribution with probability at most $(1+1/\mathsf{poly})$ times the probability it occurs w.r.t. the uniform distribution. In particular, events that are negligible under the uniform distributions cannot become noticeable under the distribution outputted by the extractor. This was then sufficient for obtaining leakage resilient cryptosystems with negligible security guarantees.

In this work, we consider applying the relative error notion to the setting of *seedless, non-malleable* extractors. Our notion differs in two ways: First, we need to extend the notion to the case where neither the real nor simulated distribution is uniform. This is because the guarantee of the non-malleable extractor holds with respect to a pair of output values (a, b), where a should be uniform random, but b can come from an arbitrary distribution. Second, due to the above, we slightly relax the notion and incorporate a small additive term, $\beta \ll 2^{-2m}$, where m is the output length of the extractor.

We now parametrize the relative extractor notion by α and β and require that the probability of any untampered/tampered output pair (a, b) under the real distribution is at most $(1 + \alpha)p_I(a, b) + \beta$, where $p_I(a, b)$ denotes the probability of output pair (a, b) under the ideal distribution.

Applications to leakage and tamper resilience with negligible security. A non-malleable extractor $\mathsf{E} : \{0,1\}^n \to \{0,1\}^m$ with *relative error (α, β)* for a class of recognizable sources $\mathcal{X}$ and tampering family $\mathcal{T}$, can be used to obtain leakage and *tamper* resilient cryptosystems with *negligible* security guarantees. To achieve this, one can store a uniformly random R on a device and use $a = \mathsf{E}(R)$ as the secret key for a symmetric key cryptosystem Π. The attacker is allowed (1) leakage on R with leakage function ℓ from the class of bounded polynomial-size circuits with bounded output length;[12] (2) tampering on R with tampering function t from the class of bounded polynomial-size circuits; (3) oracle access to *both* Π_a, and Π_b, where $b = \mathsf{E}(t(R))$ is the tampered version of the key (Π_a, Π_b denote fixing the secret key of Π to a or b respectively). We show that in several cases, we can still guarantee the *negligible* security of the cryptosystem with respect to the *original* key a, despite this stronger adversarial model.

We consider two types of applications. First, for cryptosystems Π that have an associated *unpredictability* game (such as MAC's), negligible security in the leakage and tampering game described above can be proved from the properties of the relative error non-malleable extractor, assuming the original cryptosystem Π satisfies the standard security notion. Second, for cryptosystems Π that have an associated *indistinguishability* game (such as CPA secure symmetric key encryption), negligible security in the leakage and tampering game described above can be proved in the case that the original cryptosystem Π satisfies a type of "square-security" notion (see for example [BDK+11, DY13], for a discussion of the square-security notion). We note that there are natural examples of cryptosystems that achieve this required notion. For example CPA-secure symmetric key encryption satisfies the "square-security" notion needed for our result.

We emphasize that, for both the unpredictability and indistinguishability applications discussed above, by using *relative error* non-malleable extractors, we are able to prove that the attacker's advantage is *negligible* in the leakage and tampering game. See Appendix C.4 for further details.

Non-malleable secret sharing and non-malleable codes are equivalent under polysize circuit tampering. Secret sharing schemes allow a user with a secret to send "shares" to a set of parties such that any "authorized" subset of parties can recover the secret from their collective shares, but "unauthorized" subsets of parties learn nothing about the secret from their collective shares. This relatively simple object, about which many foundational questions remain unanswered, is a critical tool in modern cryptography.

In 2018, Goyal and Kumar [GK18a] introduced the notion of *non-malleable secret sharing.* To understand what it means for a secret sharing scheme to be non-malleable, consider the following experiment: share a secret, jointly tamper all the shares, reconstruct the tampered shares of some authorized subset of parties. Loosely, a secret sharing scheme is non-malleable if the outcome of this experiment returns the original secret or some value independent of the original secret (and which case occurs should also be independent of the original secret).

Goyal and Kumar constructed non-malleable *threshold* secret sharing schemes (where any t parties are authorized and can recover the secret) for tampering that is independent on each share. They also constructed

[12] In fact, the precise leakage class we can handle is more broad and is discussed in Section C.4.

a scheme resilient to joint tampering on small sets of $< t$ shares (shares are partitioned into small sets and each set of shares is tampered independently).[13]

Subsequent work has constructed schemes for similar settings with improved parameters [BS19, GSZ21], schemes with additional features such as leakage resilience [ADN+19], schemes for more exotic access structures (the predicate specifying authorized sets of parties) [GK18b, BS19, ADN+19], schemes that can tolerate joint tampering of all the shares by "simple" tampering such as affine tampering or tampering by low degree polynomials [LCG+19a, BCL+20], and more.

In Appendix D, we construct non-malleable secret sharing schemes that are resilient to joint tampering of the shares by polynomial size circuits for a wide variety of access structures, any access structure for which an explicit (efficiently computable) secret sharing scheme exists.

In fact, we observe that non-malleable secret sharing and non-malleable codes for polynomial size circuit tampering are effectively equivalent. This is a testament to the richness of this tampering class. More precisely, to construct such a non-malleable secret sharing scheme from a non-malleable code, one simply encodes the secret with the non-malleable code and shares the codeword according to a polysize computable secret sharing scheme (to reconstruct the secret, simply reconstruct the codeword and decode). This is safe because composing sharing, tampering, and reconstructing can in turn be performed by a polynomial size circuit, because the secret sharing scheme is efficient. (The reverse direction is immediate.)

We go on to construct *adaptive* non-malleable secret sharing schemes resilient to polynomial size circuit tampering for a wide variety of access structures, including any access structure admitting an efficient *linear* secret sharing scheme. In adaptive non-malleable secret sharing, the tampering function can be chosen arbitrarily as a function of any unauthorized set of shares.

1.3 Technical Overview

We begin by discussing difficulties in a strawman approach: directly derandomizing probabilistic method constructions. Then, we demonstrate our technique, which avoids the strawman's issues, by presenting a construction and proofsketch for a simplified case: Constructing "relaxed" non-malleable extractors (where the tampering function is guaranteed to have no fixed points) for uniformly random sources and bounded polynomial tampering (i.e. size n^c circuits for some constant c). While this is a simplified case, it will already give most of the key ideas of our main results. We conclude the section by discussing how to extend this example and its analysis to achieve our main results.

Aside: On Derandomizing Randomized Constructions [CG14a, FMVW14] (A reader only interested in how are solution works can safely skip this aside.)

Given the existence of Monte Carlo style[14] constructions of efficient non-malleable codes for polynomial-size tampering, intuitively one might try to derandomize these constructions to arrive at a single, explicit non-malleable code. Unfortunately, while it is straightforward to arrive at an enumerable family of candidate code, nearly all of which are non-malleable (but not all), it is unclear how to *combine* these candidate codes (unlike when solving a decision problem, one cannot simply try them all and take the majority).

Indeed, it may be instructive to walk through this argument for encoding just a single bit. There is a particularly simple monte carlo style construction of a non-malleable code for single bit messages [Bal21]. Let $\mathcal{H}$ to be an $t(n) = \tilde{O}(n^c)$-wise independent hash family.[15] Then sample $h \leftarrow \mathcal{H}$ and set $\mathsf{D}_h \equiv h$ and take $\mathsf{E}_h(b)$ to simply sample a uniformly random x such that $\mathsf{D}_h(x) = b$. This gives us a large, $2^{\tilde{O}(n^c)}$, family of candidate codes $\{(\mathsf{E}_h, \mathsf{D}_h)\}_{h \in \mathcal{H}}$, most of which are non-malleable against n^c-size tampering.

Next, we remark that one-bit non-malleability admits a (relatively) simple test. In particular, there is a *nondeterministic* circuit of size $t(n)^{O(1)}$ that outputs 1 if and only if the input, two circuits (E, D) (with inputs for randomness) of size $t(n)$, represents a non-malleable code against n^c-size tampering [Bal21].[16]

[13] Note that there is no hope achieve non-malleability if the tampering function can recover the secret and output based on that.

[14] By Monte Carlo, we mean their is an efficient *randomized* algorithm that outputs succinct description of (E, D) such that with high probability (E, D) represent a non-malleable code.

[15] Recall, $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}\}$ is *t-wise independent* if for any distinct strings $x_1, \ldots, x_t \in \{0,1\}^n$, $h(x_1), \ldots, h(x_t)$ is a uniformly random string when $h \leftarrow \mathcal{H}$.

[16] Recall that for single bit messages, (E, D) is ϵ-non-malleable if and only if for every tampering function f $\Pr_{r,b}[\mathsf{D}(f(\mathsf{E}(b); r)) = 1 - b] \leq 1/2 + \epsilon$. Now consider the MA-style proof that a E, D is *not* $1/n^c$-non-malleable:

Because we can test non-malleability for n^c-size tampering with a polysize nondeterministic circuit, it follows that $(\mathsf{E}_h, \mathsf{D}_h)$ should be non-malleable with high probability when h is sampled *pseudorandomly*. In particular, if G is a pseudorandom generator for nondeterministic circuits, then $\{(\mathsf{E}_{G(s)}, \mathsf{D}_{G(s)}\}_s$ is a small family of candidate codes that are mostly non-malleable. Moreover, if G has exponential stretch (which follows from our assumption "E is hard for exponential size nondeterministic circuits"), then this family can be enumerated in polynomial time.

Unfortunately, at this point we are stuck. It is unclear how to *combine* $\{(\mathsf{E}_{G(s)}, \mathsf{D}_{G(s)}\}_s$ to arrive at a single non-malleable code. While we have yet to rule out the existence of *generic* combiners (black box procedures that construct non-malleable codes from families of candidate codes where most are non-malleable) entirely, most intuitive constructions admit counterexamples.[17] While this particular family has some specific structure (beyond the promise that most candidate codes are non-malleable), it is not obvious how to exploit this.

It may be helpful to compare the situation to that of *error correcting codes*. It is well known that a random matrix generates a linear code on the Gilbert-Varshamov (GV) bound with high probability. Because minimal distance can be tested by small nondeterministic circuits, we can even efficiently enumerate a set of mostly good candidate codes [CSW06]. If we could combine such an ensemble, we would have an explicit construction a code on the GV bound.

However, after 70 years, explicitly constructing a code with this distance, even under appropriate hardness assumptions, remains an open problem. In contrast, while it remains unclear how to *directly* derandomize ensembles of non-malleable codes for polysize circuit tampering, we can use the same assumption used to partially derandomize such an ensemble to indirectly arrive at an explicit construction.

A Simple Example: (Relaxed) Seedless Non-Malleable "Extractor" for Uniform Sources First, recall that a *relaxed seedless non-malleable extractor* (Def. 2.9) for sources of the form (S, X) is a deterministic function NMExt such that for any n^c size circuit, C *without fixed points* we have

$$(\mathrm{NMExt}(S, X), \mathrm{NMExt}(C(S, X))) \approx (\mathcal{U}, \mathrm{NMExt}(C(S, X))).$$

We reiterate that here we simplify by assuming that the source (S, X) is uniform random. While this trivializes the task of randomness extraction, the question of non-malleable extraction remains interesting for such sources, e.g. it already implies the existence of non-malleable codes for 1-bit messages. [18]

Before describing our construction, we give a brief overview of the necessary building blocks:

Strong relaxed two-source non-malleable extractor. Loosely speaking, a function $\mathrm{NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ is a relaxed two-source non-malleable extractor for sources (X, Y) if for every split-state tampering function (τ_L, τ_R) for which either τ_L or τ_R has no fixed points, we have

$$(\mathrm{NMExt}(X, Y), \mathrm{NMExt}(\tau_L(X), \tau_R(Y))) \overset{s}{\approx} (\mathcal{U}_m, \mathrm{NMExt}(\tau_L(X), \tau_R(Y))).$$

We say NMExt is a strong two-source non-malleable extractor for no-fixed points tampering if we further have that

$$(X, \mathrm{NMExt}(X, Y), \mathrm{NMExt}(\tau_L(X), \tau_R(Y))) \overset{s}{\approx} (X, \mathcal{U}_m, \mathrm{NMExt}(\tau_L(X), \tau_R(Y))).$$

Merlin sends a n^c-size circuit, f, (the witness/proof) and Arthur accepts if $\mathsf{D}(f(\mathsf{E}(b) = 1 - b$ for random b. If E, D is not $1/n^c$ non-malleable, then there exists an f to make Arthur accept with probability $\geq 1/2 + 1/n^c$. On the other hand, if E, D is $1/2n^c$-non-malleable, then for every f Arthur accepts with probability $\leq 1/2 + 1/2n^c$. This can be derandomized with nonuniform advice using standard techniques.

[17] In more detail, the main issue is that a "bad" candidate code can behave maliciously. For example, imagine candidate codes which output codewords with long "dummy" prefixes. "Good" codes can simply the prefix entirely, but a tampering function could conspire with "bad" decoders by feeding all the other non-dummy codeword parts with some instructions to the bad decoders. So long as the bad decoder has some influence on the output, which seems necessary, tampering is possible.

[18] To see this, recall the characterization of non-malleability for a single bit (see previous footnote). Note that for any tampering function f of size n^c, one can define a function f' of size $n^c + O(n)$ that has no fixed points and behaves identically to f on every x that is not a fixed point of f. Because, $\Pr[D(f(\mathsf{E}(b) = 1 - b] \leq \Pr[D(f'(\mathsf{E}(b)) = 1 - b]$ we can deduce that E, D is non-malleable with respect to circuits of size $n^c - O(n)$, where D is NMExt and E simply performs rejection sampling to find a random (s, x) such that $\mathrm{NMExt}(s, x) = b$. Note that the resulting non-malleable code will not have perfect correctness because the rejection sampling procedure might fail.

Two source non-malleable extractors are well-studied in the literature with the current state-of-the-art being extractors for sources $(X, Y) \in \{0,1\}^n \times \{0,1\}^n$ with min-entropy $(1-\gamma)n$ for some constant γ and error $2^{-\Omega(n \log\log(n)/\log(n))}$ [Li19]. Further, [Li16a] showed that every two source non-malleable extractor is also a strong two source non-malleable extractor for sources with some loss in parameters.

Recalling the notion of a nondeterministic circuit from the introduction, we now introduce a type of pseudorandom generator (PRG) with security against non-deterministic circuits of bounded polynomial size.

Seed-extending pseudorandom generators. A *pseudorandom generator (PRG) for nondeterministic circuits of size* n^d, $\mathsf{G} : \{0,1\}^\ell \to \{0,1\}^n$, allows one to extend a short random seed into a long string that is indistinguishable from random to nondeterministic circuits of size n^d (for constant d). More precisely, for every nondeterministic circuit, C, of size at most n^d,

$$|\Pr[C(\mathsf{G}(\mathcal{U}_\ell)) = 1] - \Pr[C(\mathcal{U}_n) = 1]| \leq \frac{1}{n^d},$$

where $\mathcal{U}_m$ denotes a random variable uniformly distributed over $\{0,1\}^m$.

The above type of PRG are different from cryptographic PRG's since the computation time of the PRG is larger than the size of the adversary. Specifically, these PRG's are secure against nondeterministic circuits of size n^d, but take larger polynomial time to compute. Cryptographic PRG's are computable in some fixed polynomial time but secure against adversaries of arbitrary polynomial size. In the case of seed-extending pseudorandom generators, this gap between honest and adversarial computational resources allows for unintuitive behavior, where the seed of the PRG itself is included as part of the output and the output remains pseudorandom, which is impossible in the cryptographic case.

Indeed, we are interested in exactly such PRGs that remain secure even when given the seed, referred to as “seed-extending” PRGs.[19] A PRG, $\mathsf{G} : \{0,1\}^\ell \to \{0,1\}^n$, is said to be seed-extending if $\mathsf{G}(s) = (s, \mathsf{G}'(s))$ (where G' is the function corresponding to the $n - \ell$ bit suffix). This particular name was introduced by Kinne et al. in the context of derandomizing randomized algorithms on random inputs. [KvMS12, LZ19] They observed that PRG constructions based on Nisan and Wigderson's seminal construction [KvMS12] can be made seed-extending. Consequently, many constructions of PRGs for nondeterministic circuits can be made seed extending.

Theorem 1.6 ([KvMS12, IW97, KvM02, SU05, SU06, AASY16]). *If* E *requires exponential size nondeterministic circuits, then for every constant* $c > 1$ *there exists a constant* $\alpha > 1$ *such that for every sufficiently large* n*, and every* ℓ *such that* $\alpha \log n \leq \ell \leq n$ *there is a seed-extending PRG,* $\mathsf{G} : \{0,1\}^\ell \to \{0,1\}^n$*, for nondeterministic circuits of size* n^c*.*

We are now ready to present our construction for our simplified setting.

Construction of a Seedless Relaxed Non-Malleable Extractor. Our construction of a (relaxed) seedless non-malleable extractor for uniform sources and n^c-size circuit tampering is exceedingly simple. Let 2NMExt be a relaxed, two-source non-malleable extractor (NME). Our seedless relaxed non-malleable extractor, NMExt, is defined as

$$\text{NMExt} : (s, x) \mapsto \text{2NMExt}(\mathsf{G}(s), x)$$

where G is a *seed-extending PRG for nondeterministic circuits of size* n^d for some constant $d > c$.

Sketch of the Security Proof. To prove security of the construction, we need to show that the existence of a size n^c tampering function with no fixed points that breaks the security of the NME, implies the existence of a nondeterministic circuit of size n^d that distinguishes outputs of G from random.

Suppose for the sake of contradiction that there exists a successful tampering function, $\tau : (s, x) \mapsto (\tilde{s}, \tilde{x})$ of circuit size n^c with no fixed points. We will define f to denote the function that computes $(s, x) \mapsto \tilde{x}$ according to τ, and g to denote the function that computes $(s, x) \mapsto \tilde{s}$ according to τ. In other words, $\tau(s, x) = (g(s, x), f(s, x))$ and moreover, for each (s, x) either $g(s, x) \neq s$ or $f(s, x) \neq x$. Note that there is *no* split-state assumption on the tampering function $\tau(s, x) = (g(s, x), f(s, x))$, as both f and g can depend on the entire input (s, x).

[19] We refer the reader to [KvM02] for further discussion.

Now, our assumption on τ (and hence f, g) breaking the NME can be restated as

$$\Delta((\text{2NMExt}(\mathsf{G}(S), X), \text{2NMExt}(\mathsf{G}(g(S, X)), f(S, X))); (\mathcal{U}_m, \text{2NMExt}(\mathsf{G}(g(S, X)), f(S, X)))) \geq \epsilon. \tag{1}$$

We will use this assumption to "distinguish" the seed-extending PRG, G, from the uniform distribution via a private constant round interactive proof (i.e. Arthur Merlin protocol). In particular, (private-coin) Arthur will accept pseudorandom inputs (completeness) with polynomially higher probability than he accepts random inputs, regardless of how Merlin behaves (soundness). Then, we can deduce from standard transformations ($\mathsf{IP}[k] \subseteq \mathsf{AM} \subseteq \mathsf{NP/poly}$ [BM88, GS86]) that a small non-deterministic distinguisher exists.[20]

Looking ahead, (1) which asserts the *malleability* of the constructed extractor when provided pseudorandom inputs will enable us to prove the protocol is complete, i.e. Arthur accepts pseudorandom inputs with high probability. Soundness, i.e. Arthur rejects random inputs with high probability, will ultimately follow from security of the 2-source non-malleable extractor. Furthermore, what ultimately will enable our soundness argument to go through is the fact that to achieve completeness Arthur communicates very little about random variable X and thus X remains entropic, even after conditioning on this communication. We use a standard private coin technique, where Arthur forces Merlin to guess between two samplable distributions [GMW91] to handle the fact that our extractor has relatively long outputs (even though our hardness assumption only holds for boolean distinguishers in a relatively high error regime).

Arthur Merlin Protocol. We next describe the interactive proof for distinguishing G from uniformly random bits. **Both Arthur and Merlin receive (s, y) as input.** Our protocol aims to accept strings from $G(\mathcal{U}_\ell)$ when Merlin plays according to below (completeness) and reject strings from $\mathcal{U}_n$ regardless of the strategy Merlin utilizes (soundness). Because we can amplify by repetition, it suffices for there to be small gap between the two.

Arthur Sample $x \leftarrow \mathcal{U}_n$. Send Merlin $\tilde{s} = g(s, x)$.
Merlin If $(s, y) = \mathsf{G}(s)$, respond $\tilde{y}$ such that $(\tilde{s}, \tilde{y}) = \mathsf{G}(\tilde{s})$. Otherwise, respond arbitrary $\tilde{y}$.
Arthur Sample a random coin $b \leftarrow \mathcal{U}$ and set $\tilde{z} = \text{2NMExt}((\tilde{s}, \tilde{y}), \tilde{x})$ where $\tilde{x} = f(s, x)$.
– *If* $b = 0$*:* Sample $z \leftarrow \mathcal{U}_m$ and send $z, \tilde{z}$.
– *Else if* $b = 1$*:* Sample $z \leftarrow \text{2NMExt}((s, y), x)$ and send $z, \tilde{z}$.
Merlin Guess Arthur's bit by guessing whether $(z, \tilde{z})$ was drawn from the first or second distribution.
Arthur Accept if $b = b'$, and reject otherwise.

Completeness: accepting pseudorandom inputs. We first argue that Arthur, when playing with Merlin as specified above, accepts pseudorandom inputs, drawn from $\mathsf{G}(S)$, with probability significantly greater than 1/2. Indeed, if the protocol above is given inputs from $\mathsf{G}(S)$ (i.e. legitimate outputs of G), then if Arthur chooses $b = 1$, his final message is sampled as:

$$(z, \tilde{z}) \sim (\text{2NMExt}(\mathsf{G}(S), X), \text{2NMExt}(\mathsf{G}(g(S, X)), f(S, X))).$$

On the other hand, if $b = 0$, Arthur's final message is sampled according to:

$$(z, \tilde{z}) \sim (\mathcal{U}_m, \text{2NMExt}(\mathsf{G}(g(S, X)), f(S, X))).$$

By our malleability assumption towards contradiction (1), these two distributions are ϵ-far from each other.

Soundness: rejecting random inputs. We must now show that when given uniformly random inputs, Arthur accepts with significantly lower probability than the case above. This case is harder than the previous case, since here Merlin can behave arbitrarily, and we must show that Arthur still rejects w.h.p.

At a high-level, we get around this by observing that although Merlin is computationally unbounded, the fact that the information sent to him by Arthur is limited, essentially constrains Merlin to *split-state* strategies. Specifically, let $G^* : (s, y, \tilde{s}) \mapsto \tilde{y}$ be the function that given Merlin's input (s, y) and the transcript thus far, outputs Merlin's first message. Conditioned on $s, \tilde{s}$, we have that $G^*(s, y, \tilde{s}) = \tilde{y}$ is independent of x (as is $\tilde{s}$). And similarly, $\tilde{x} = f(s, x)$ is independent of (s, y). So conditioned on $s, \tilde{s}$ we can define a *split-state* tampering function as follows:

[20] In actuality, this is too naive because these transformations only hold for worst-case notions of soundness and completeness. Thus in the body, we will instead show that there exists a constant round interactive proof for a *promise problem* (Π_Y, Π_N) such that Π_Y is dense in the pseudorandom distribution and Π_N is dense in the uniform distribution, and not vice-versa.

- $\tau_L^{\tilde{s}} : (s, y) \mapsto (\tilde{s}, \tilde{y})$ where $\tilde{y} = G^*(s, y, \tilde{s})$
- $\tau_R^{s} : x \mapsto \tilde{x}$ where $\tilde{x} = f(s, x)$

Note that because τ has no fixed points, either $f(s, x) \neq x$ or $g(s, x) \neq s$. So, either $\tau_L^{\tilde{s}}$ or τ_R^{s} contains no fixed points. Thus, conditioned on $s, \tilde{s}$ and Arthur's coin $b = 0$, Merlin's view is simply

$$T_0^{s,\tilde{s}} \equiv \left((s, y), \mathcal{U}, \text{2NMExt}(\tau_L^{\tilde{s}}(s, y), \tau_R^{s}(x))\right).$$

On the other hand, if Arthur's coin is $b = 1$, Merlin's view is

$$T_1^{s,\tilde{s}} \equiv \left((s, y), \text{2NMExt}((s, y), x), \text{2NMExt}(\tau_L^{\tilde{s}}(s, y), \tau_R^{s}(x))\right).$$

Recall that the input (s, y) (left source) and x (right source) are both uniform. Thus, after conditioning on the transcript (or equivalently $s, \tilde{s}$) nearly all the entropy remains in each source (in fact, we can take $s, \tilde{s}$ short enough that the entropy deficiencly is just $O(\log(n))$). Then because 2NMExt is a *strong* two-source non-malleable extractor for sources with linear min-entropy, it follows from the security property that:

$$T_0^{s,\tilde{s}} \overset{s}{\approx} T_1^{s,\tilde{s}}.$$

Obtaining our Main Results We extend the above technique in several ways to obtain our main results.

Non-malleable extractors for samplable/recognizable sources. First, we combine the above construction with a seedless extractor for polynomially samplable (resp. recognizable) sources [TV00, AASY16] to obtain a *relaxed* seedless *non-malleable* extractor for polynomially samplable (resp. recognizable) sources and polynomially bounded tampering.

In brief, we use a seedless extractor to sample the uniform seed, s, for the PRG in the simple construction above. The main difference relative to the proof above, is that now Arthur must sample the samplable/recognizable source to be consistent with the pseudorandom challenge, i.e. conditioned on the seedless extractor outputting s. This is resolved in both cases by equipping Arthur with an NP-oracle, so he can efficiently sample random satisfying assignments to small circuits [BGP00, JVV86].

The full details of our constructions and their analysis can be found in Section 3. Similar to above, we first construct an extractor secure against tampering functions without fixed points (this out by Cheraghchi and Guruswami [CG14b] and first construct an extractor secure against tampering functions without fixed points. Then in Section 3.3, we show how to remove the requirement of no fixed-points in the tampering functions to obtain seedless *non-malleable* extractor for polynomially samplable sources and polynomially bounded tampering.[21]

Non-malleable code. The above non-malleable extractors suggest an natural path to non-malleable codes. Cheraghchi and Guruswami [CG14b] show that *invertable* non-malleable extractors for a tampering class C imply non-malleable codes for that C. However, there are two obstacles to applying their approach here. First, it is unclear how to efficiently invert our extractors. Secondly, this transformation has 2^k security loss, where k is the bit length of the messages to be encoded. Given the polynomial security, this means the resulting construction would have exponential length codewords and would not actually be explicit.

We therefore take the route of directly constructing non-malleable codes, with the added benefit that we reduce our hardness assumptions from "E requires exponential size Σ_3-circuits" (required for our non-malleable extractors) to "E requires exponential size nondeterministic circuits."

Our result is obtained by replacing the two-source non-malleable extractor in the simple example above with a split-state non-malleable code: to encode a message m, sample a split-split state encoding of the form $(\mathsf{G}(S), y)$ and output s, y. To make a similar Arthur Merlin distinguisher work for this construction, we need the split-state code to have some special properties:

- **Special Encoding:** We need to be able to sample pseudorandom split-state code words efficiently in order to encode efficiently at all. To do this we introduce a notion of *special encoding*:

  There is an alternate encoding algorithm that receives the value of the first split state along with a message m and samples the second split-state so that the resulting encoding decodes to m. Critically, if the value of the first split-state is sampled uniformly at random, then the outputted encoding is distributed identically to a random encoding of m.

[21] Cheraghchi and Guruswami [CG14b] showed a similar lemma for the case of split-state tampering.

- **Leakage Resilience:** The soundness argument above relied on the fact that two-source extractors remain secure even if there is small amount of leakage on the states (corresponding to the transcript). Note that this leakage is both to the independent components of the split-state tampering function *and* the (possibly inefficient) distinguisher of the non-malleability game. If this is the case, we say a such split-state code is *leakage-resilient*.[22]
- **"Augmented" NMC:** Finally, our soundness argument above additionally required that Merlin could not distinguish the real and ideal experiments even when given the left source in its entirety. For this we relied on the fact that 2NMExt was a *strong* two-source non-malleable extractor. The corresponding notion for split-state non-malleable codes is the *augmented* property: security of the NMC holds even when one half of the codeword is revealed at the end of the experiment to a (possibly inefficient) distinguisher.

An NMC with the necessary properties is constructed by (1) observing that the NMC of Aggarwal et al. [ADL18] satisfies both the augmented NMC and special encoding properties and (2) applying the leakage resilience transformation of Ball et al. [BGW19] and proving that it preserves the augmented NMC and special encoding properties. See Appendix A for more details.

The details of our construction of a non-malleable code for polysize tampering and its analysis can be found in Section 4. Note that the rate of our code inherits the rate of the NMC of Aggarwal et al. [ADL18], which means that to encode a message of length k (for sufficiently long k), one needs a codeword length of $n = O(k^7)$. A better split-state NMC with the above properties will yield a better NMC for polysize tampering, but rate is not our focus here.

1.4 Related Work

Non-malleable extractors and codes. There is by now a large body of work on non-malleable extractors and non-malleable codes resilient against various classes of tampering [DPW10, DKO13, Li17, Li19, ADL18, ADKO15, CL17, BDG+18, BDKM16, BGW19, BCL+20, AGM+15a, AO20, KOS17]. In the non-malleable codes case, some constructions not included in the list above rely on cryptographic assumptions [BDKM18, AGM+15b], while others require an untamperable common reference string (CRS) [LL12, BDKM18]. There has also been much work on variants of non-malleable extractors and non-malleable codes [CGL16, FMNV14, DLSZ15, KOS18], as well as a relatively new line of work on a related primitive called non-malleable secret sharing [GK18a, GK18b]. We restrict our attention to constructions most relevant to the current work, namely, the prior constructions of non-malleable codes (in the CRS and standard models) resilient to bounded polynomial tampering, where "bounded polynomial" can refer to a restriction on (1) circuit size, (2) uniform computation time, (3) circuit depth. Existence of non-malleable codes under all of the above types of tampering was initially shown via the probabilistic method in [DPW18] and they can also be constructed efficiently in the random oracle model [DPW18]. In the following, we additionally restrict our attention to explicit, efficient constructions *without* random oracles. We also mention a somewhat related line of work on variants of non-malleable codes resilient to polynomially *space-bounded* tampering in the random oracle model [FHMV17, CCHM19].

NMC against bounded polynomial sized circuits in the CRS model. Faust et al. [FMVW14] presented efficient information theoretically secure NMC with negligible error in the CRS model, resilient against tampering function classes F which can be represented as circuits of size poly(n). We note that the CRS in their construction is a seed s for a p(n)-wise independent hash function, where $p(n)$ is a polynomial that is larger than the bound on the tampering circuit size.

NMC against uniform, bounded polynomial time in the standard model. Ball et al. [BDK+19] presented efficient non-malleable codes resilient against tampering by functions computable in uniform bounded polynomial time. Their construction is in the standard, no-CRS model and achieves error of $1/\mathsf{poly}$. They require a similar assumption as those used in the current work (that E requires exponential size NP circuits), as well as cryptographic assumptions of the existence of sub-exponentially hard trapdoor permutations and the existence of P-certificates with sub-exponential soundness. We note that the only known instantiation of P-certificates requires assuming soundness of a non-trivial argument system (Micalis CS proofs [Mic94]), which is true in the Random Oracle model. Due to the use of cryptographic techniques in the construction and proof, the final non-malleable code achieves computational indistinguishability.

[22] In the literature, leakage-resilient has been alternately used to refer to codes that handle leakage only to the distinguisher as well as code that handle leakage only between the tampering of each state.

NMC against bounded polynomial depth circuits (unbounded polynomial size) in the standard model. Dachman-Soled et al. [DKP20, DSKP21] constructed non-malleable codes resilient to all polynomial size tampering functions that have bounded polynomial depth. This tampering class contains all bounded polynomial size functions and contains non-uniform NC. Their construction is in the standard, no-CRS model and achieves negligible error. They require the cryptographic assumptions of the existence of keyless multi-collision resistant hash function, injective one-way function, and non-interactive witness-indistinguishable proofs, as well as the repeated squaring assumption. Keyless multi-collision resistant hash function are known to exist in the auxiliary input random oracle model. Due to the use of cryptographic techniques in the construction and proof, the final non-malleable code achieves computational indistinguishability.

Seedless extractors for samplable and recognizable sources. Trevisan and Vadhan [TV00] considered seedless extractors for the class of distributions samplable by bounded polynomial sized circuits. Under the assumption that E requires exponential size Σ_4 circuits, they presented constructions of seedless extractors for linear min-entropy, samplable sources over n bits, that output $\Omega(n)$ bits that are $1/\mathsf{poly}$-close to uniform. Applebaum et al. [] showed that the $1/\mathsf{poly}$ error is at least somewhat inherent by ruling out black-box reductions in this setting. They therefore introduced a notion of *relative-error* extractors and showed that if the output of the extractor is $1/\mathsf{poly}$-close to uniform with relative error, then every event occurs w.r.t. the output distribution with probability at most $(1+1/\mathsf{poly})$ times the probability it occurs w.r.t. the uniform distribution. In particular, events that are negligible under the uniform distributions cannot become noticeable under the distribution outputted by the extractor. Under the assumption that E requires exponential size Σ_4 circuits, they constructed relative-error seedless extractors whose outputs are $1/\mathsf{poly}$-close to uniform with relative error for linear min-entropy, samplable sources. Under the assumption that E requires exponential size Σ_3 circuits, they constructed relative-error seedless extractors whose outputs are $1/\mathsf{poly}$-close to uniform with relative error for linear min-entropy, recognizable sources.

1.5 Organization

In Section 2 we present notation and preliminaries. All other sections and appendices are self-contained and can be read in any order.

In Section 3 we present our construction of seedless non-malleable extractors for samplable and recognizable sources. Specifically, in Sections 3.1 and 3.2, we present constructions of *relaxed* seedless non-malleable extractors for samplable and recognizable sources, where *relaxed* means that the tampering function is guaranteed to have no fixed points. In Section 3.3 we show how to remove the "relaxed" assumption. In Section 3.4 we combine the results to obtain our final theorem statements.

In Section 4 we present our construction of the non-malleable code, based on a construction of a new, enhanced type of split-state non-malleable code which we call, NMC with "augmented leakage-resilient split-state and special encoding," which can be found in Appendix A.

In Appendix B, we present our new lower bound, which rules out black-box reductions from assumptions of the form "E requires exponential size Σ_i-circuits" to non-malleable codes resilient against bounded polynomial tampering.

In Appendix C, we present our construction of relative error non-malleable extractors for recognizable sources, and we discuss cryptographic applications of such extractors, which achieve negligible security, in Appendix C.4.

In Appendix D, we present our results on non-malleable secret sharing.

Finally, Appendix E contains some additional proofs.

2 Preliminaries

For $S \subseteq N$, where $S = \{i_1, \ldots, i_\ell : i_1 < \cdots < i_\ell\}$ and any n-ary string of values $x_1, \ldots, x_n$, let x_S denote the string $(x_{i_1}, \ldots, x_{i_\ell})$.

For any two random variables X, Y, we write $\Delta(X; Y) \leq \epsilon$ or $X \approx_\epsilon Y$ if the total variation distance between their distributions is at most ϵ.

2.1 Complexity classes and assumptions

We take E to denote $\mathsf{DTIME}[2^{O(n)}]$ the class of languages decidable by deterministic Turing machines in 2^{cn}-time for some constant c.

In this work, we take circuits to denote circuits over the standard basis $\{\vee, \wedge, \not\}$. For any language O, an O-oracle aided circuit is a circuit that has special gates that decide O, in addition to the standard-basis.

For any circuit, we say it has size s if it contains at most s gates. We say it has depth d if the longest path from any input to any output gate is of size d. A circuit family, $\{C_n\}_{n\in\mathbb{N}}$, is a collection of circuits such that C_n takes inputs of length n.

We take the $\mathsf{SIZE}[s(n)]$ to denote the function families computable by a circuit family $\{C_n\}_{n\in\mathbb{N}}$ such that C_n has size at most $s(n)$, for large enough n. Similarly, we take $\mathsf{SIZE}^O[s(n)]$ to denote the function families computable by an O-oracle aided circuit family $\{C_n\}_{n\in\mathbb{N}}$ such that C_n has size at most $s(n)$, for large enough n.

2.2 Non-malleable codes and seedless non-malleable extractors

Definition 2.1 (Coding schemes)**.** *A pair of functions* $(\mathsf{Enc}, \mathsf{Dec})$*, where* $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^n$ *is a randomized function and* $\mathsf{Dec} : \{0,1\}^n \to \{0,1\}^k \cup \{\bot\}$ *is a deterministic function, is defined to be a coding scheme with block length n and message length k if for all* $z \in \{0,1\}^k$, $\Pr[\mathsf{Dec}(\mathsf{Enc}(s)) = s] = 1$.

Definition 2.2 (Tampering functions)**.** *For any* $n > 0$*, let* $\mathcal{H}_n$ *denote the set of all functions* $h : \{0,1\}^n \to \{0,1\}^n$*. Any subset* $\mathcal{G} \subseteq \mathcal{H}_n$ *is a family of tampering functions.*

For any class of boolean functions $\mathcal{F} = \{f : \{0,1\}^n \to \{0,1\}\}$*, we take* $\mathcal{F}^n$ *denote to denote the class of n-output functions where each output is computed by some function in* $\mathcal{F}$*, i.e.* $\mathcal{F}^n = \{f_{i_1,\ldots,i_n} : x \mapsto f_{i_1}(x), \ldots, f_{i_n}(x) \mid f_{i_1}, \ldots, f_{i_n} \in \mathcal{F}\}$.

Two particular classes of tampering functions we consider in this work:

- Tampering where each output is computable by an $s(n)$-size circuit, $\mathsf{SIZE}^n[s(n)]$.
- Split-state tampering where two halves of an input are tampered independently and arbitrarily: $\{(\tau_L, \tau_R) : x_1, \ldots, x_{2n} \mapsto \tau_L(x_1, \ldots, x_n), \tau_R(x_{n+1}, \ldots, x_{2n}) | \tau_L, \tau_R \in \mathcal{H}_n\}$.

We define a function that will be useful in defining non-malleable codes:

$$\mathrm{Copy}(x, y) = \begin{cases} x & \text{if } x \neq \texttt{same} \\ y & \text{if } x = \texttt{same}. \end{cases}$$

Definition 2.3 (Non-malleable codes)**.** *A coding scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *on alphabet* $\{0,1\}$ *with block length n and message length k is a* ϵ*-non-malleable code with respect to a tampering family* $\mathcal{F} \subset \mathcal{H}_n$ *if for every* $f \in \mathcal{F}$ *there is a random variable* D_f *supported on* $\{0,1\}^k \cup \{\texttt{same}\}$ *that is independent of the randomness in* Enc*, and for any message* $z \in \{0,1\}^k$*, we have*

$$\Delta\left(\mathsf{Dec}(f(\mathsf{Enc}(z))); \mathrm{Copy}(D_f, z)\right) \leq \epsilon.$$

We refer to the parameter ϵ *as the "error" of the non-malleable code.*

We define the rate of a non-malleable code $\mathcal{C}$ to be the quantity $\frac{k}{n}$.

Dziembowski, Pietrzak, and Wichs [DPW10] also provided the following alternate definition of Non-Malleable Codes and proved it to be equivalent, so long as the message domain is large enough. Roughly, this definition simply requires that the outcome of the tampering experiment with m_0 cannot be distinguished from the outcome of the tampering experiment with m_1, unless the outcome is either m_0 or m_1.

Definition 2.4 (Alternative-Non-Malleability [DPW10])**.** *Let* $\mathcal{F}$ *be a family of tampering functions. We say that a coding scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *is* ϵ*-alternative-non-malleable with respect to* $\mathcal{F}$ *if for any* $m_0, m_1 \in \{0,1\}^k$ *and any* $f \in \mathcal{F}$*, we have:*

$$\mathrm{AltNM}^{f,\mathsf{Enc},\mathsf{Dec}}_{m_0,m_1}(0) \approx_\epsilon \mathrm{AltNM}^{f,\mathsf{Enc},\mathsf{Dec}}_{m_0,m_1}(1)$$

where we define the two experiments by

$$\mathrm{AltNM}^{f,\mathsf{Enc},\mathsf{Dec}}_{m_0,m_1}(b) := \left\{ \begin{array}{c} c \leftarrow \mathsf{Enc}(m_b), \tilde{c} \leftarrow f(c), \tilde{m} = \mathsf{Dec}(\tilde{c}) \\ \textit{Output } \texttt{same} \textit{ if } \tilde{m} \in \{m_0, m_1\}, \textit{ and } \tilde{m} \textit{ otherwise.} \end{array} \right\}$$

Lemma 2.5 ([DPW10]). *If* $(\mathsf{Enc}, \mathsf{Dec})$ *is* ϵ*-alternatively-non-malleable with respect to* $\mathcal{F}$ *for* k*-bit inputs, then* $(\mathsf{Enc}, \mathsf{Dec})$ *is* $(\epsilon + 2^{-k})$*-non-malleable with respect to* $\mathcal{F}$*. If* $(\mathsf{Enc}, \mathsf{Dec})$ *is* ϵ*-non-malleable with respect to* $\mathcal{F}$ *for* k*-bit inputs, then* $(\mathsf{Enc}, \mathsf{Dec})$ *is* 2ϵ*-alternatively-non-malleable with respect to* $\mathcal{F}$*.*

Definition 2.6 (Strong Seeded Extractors). *A function* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k, ϵ)*-strong extractor if for every source* X *over* $\{0,1\}^n$ *with min entropy at least* k *and uniform* Y *over* $\{0,1\}^d$*,* $(Y, \mathsf{Ext}(X, Y)) \approx_\epsilon (Y, U_m)$*, where* U_m *is uniformly distributed over* $\{0,1\}^m$*. Moreover, we require* Ext *to be computable in polynomial time.*

Definition 2.7 (Strong Two-Source Extractors). *A function* $2\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k, ϵ)*-extractor if for every pair of sources* X, Y *over* $\{0,1\}^n$ *with combined min entropy at least* k*,* $(Y, 2\mathsf{Ext}(X, Y)) \approx_\epsilon (Y, U_m)$*, and* $(X, 2\mathsf{Ext}(X, Y)) \approx_\epsilon (X, U_m)$ *where* U_m *is uniformly distributed over* $\{0,1\}^m$*. Moreover, we require* Ext *to be computable in polynomial time.*

Definition 2.8 (Seedless non-malleable extractors). *Let* $\mathcal{G}$ *be a class of tampering functions* $\{0,1\}^n \to \{0,1\}^n$ *and* $\mathcal{X}$ *be a class of distributions over* $\{0,1\}^n$*. A function* $\mathrm{NMExt} : \{0,1\}^n \to \{0,1\}^m$ *is called an* ϵ*-seedless non-malleable extractor for source* $\mathcal{X}$ *with respect to tampering class* $\mathcal{G}$ *if for every distribution* $X \in \mathcal{X}$ *and every tampering function* $g \in \mathcal{G}$*, there exists a random variable* D_g *on* $\{0,1\}^m \cup \{\mathit{same}\}$ *that is independent of* X*, such that*

$$\Delta\left((\mathrm{NMExt}(X), \mathrm{NMExt}(g(X))); (\mathcal{U}_m, \mathrm{Copy}(D_g, \mathcal{U}_m))\right) \leq \epsilon.$$

We refer to the parameter ϵ *as the "error" of the seedless non-malleable extractor.*

Moreover, if $\mathcal{G}$ *is the class of split-state functions, we say* NMExt *is an* ϵ*-strong two-source non-malleable extractor for independent sources* X, Y *if for every pair of split-state tampering function* τ_L, τ_R*, there exists a random variable* D_{τ_L,τ_R} *on* $\{0,1\}^m \cup \{\mathit{same}\}$ *that is independent of* X, Y*, such that*

$$\Delta((X, \mathrm{NMExt}(X, Y), \mathrm{NMExt}(f(X), g(Y))); (X, \mathcal{U}_m, \mathrm{Copy}(D_{\tau_L,\tau_R}, \mathcal{U}_m))) \leq \epsilon.$$

Definition 2.9 (Relaxed Seedless non-malleable extractor). *Let* $\mathcal{X}$ *be a family of sources on* $\{0,1\}^n$ *and* $\mathcal{F}$ *be a class of tampering functions acting on* $\{0,1\}^n$*. Further assume that all* $f \in \mathcal{F}$ *does not have any fixed points. A function* $\mathrm{NMExt} : \{0,1\}^n \to \{0,1\}^m$ *is defined to be an* ϵ*-relaxed non-malleable extractor with respect to* $\mathcal{X}$ *and* $\mathcal{F}$ *if the following hold: for any* $X \in \mathcal{X}$ *and* $f \in \mathcal{F}$*, we have*

$$\Delta(\mathrm{NMExt}(X), \mathrm{NMExt}(f(X)); \mathcal{U}_m, \mathrm{NMExt}(f(X))) \leq \epsilon.$$

Theorem 2.10 (**[Li16b]**,**[Li19]**). *There exists a constant* $0 < \gamma < 1$ *and a strong non-malleable two-source extractor for* $(n, (1-\gamma)n)$ *sources with error* $2^{-\Omega(\frac{n \log\log(n)}{\log(n)})}$ *and output length* $\Omega(\frac{n \log\log(n)}{\log(n)})$*.*

The above is obtained by combining Theorem 1.1 in [Li18] (this is the ePrint version of [Li19] and the theorem states the existence of a non-malleable two source extractor for linear min-entropy sources and error $2^{-\Omega(\frac{n \log\log(n)}{\log(n)})}$) with Theorem 8.1 in [Li16a] (this is the ECCC version of [Li16b] and the theorem states that a non-malleable two-source extractor is itself a strong non-malleable two-source extractor with some loss in entropy and error).

2.3 Seed-extending pseudorandom generators

Definition 2.11 ([KvMS12]). *A function* $G : \{0,1\}^\ell \to \{0,1\}^n$ *is said to be an* ϵ-pseudorandom generator *for a class* $\mathcal{C}$*, if for all* $C \in \mathcal{C}$*,*

$$\Delta(C(G(\mathcal{U}_\ell)); C(\mathcal{U}_n)) \leq \epsilon$$

A pseudorandom generator, G*, is said to be* seed-extending *if the prefix of its output is its input, i.e.* $G(s) = s, G'(s)$ *for some function* $G' : \{0,1\}^\ell \to \{0,1\}^{n-\ell}$*.*

In this work, we are principally concerned with seed-extending PRGs against various types of circuits of a given size: non-deterministic circuits, non-deterministic NP-circuits, etc. Throughout this paper, we take a PRG for a class of circuits of size s to mean a $1/s$-PRG for that class of circuits.

Note that because we are interested in both seed-extending PRGs, as well as PRGs for non-deterministic circuits, so-called "cryptographic" PRGs which can be easily evaluated by the classes they are constructed to fool do not suffice: a distinguisher given the seed, or nondeterminism, can easily determine if a string is in the PRG's image.

Thankfully, as observed by Kinne et al. [KvMS12], Nisan and Wigderson's seminal construction yields a seed extending PRG, provided one starts with an appropriately hard function. We conclude with the formal theorem statement.

Theorem 2.12 **([KvMS12, IW97, KvM02, SU05, SU06, AASY16]).** *If E requires exponential size circuits of type $X \in \{deterministic, nondeterministic, \mathsf{NP}, \Sigma_i\}$, then for every constant $c > 1$ there exists a constant $\alpha > 1$ such that for every sufficiently large n, and every r such that $\alpha \log n \leq \ell \leq n$ there is a seed-extending PRG, $\mathsf{G} : \{0,1\}^\ell \to \{0,1\}^n$, for size n^c circuits of type $X \in \{deterministic, nondeterministic, \mathsf{NP}, \Sigma_i\}$.*

2.4 Samplable and Recognizable Distributions

Definition 2.13 (Samplable distribution [TV00, AASY16].)**.** *We say that a distribution X on n bits is samplable by a class $\mathcal{C}$ of functions $C : \{0,1\}^r \to \{0,1\}^n$ if there exists a function C in the class such that X is distributed as $C(U_r)$.*

Definition 2.14 (Recognizable distribution [AASY16].)**.** *We say that a distribution X on n bits is recognizable by a class $\mathcal{C}$ of functions $C : \{0,1\}^n \to \{0,1\}$ if there exists a function C in the class such that X is uniform over $\{x : C(x) = 1\}$.*

2.5 Seedless Extractors and Witness Sampling

Applebaum et al. [AASY16], building on work by Trevisan and Vadhan [TV00], construct extractors for samplable and recognizable sources from derandomization-type assumptions.

Theorem 2.15 **([AASY16]).** *If E requires exponential size Σ_3-circuits, then there exists a constant $\alpha > 0$ such that for every constant $c > 1$ and sufficiently large n, and there is a $((1-\alpha)n, n^{-c})$-extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^{\alpha n}$ for $\mathsf{SIZE}[n^c]$-recognizable sources. Moreover, Ext is computable in time $\mathsf{poly}(n^c)$.*

Theorem 2.16 **([AASY16]).** *If E requires exponential size Σ_4-circuits, then there exists a constant $\alpha > 0$ such that for every constant $c > 1$ and sufficiently large n, and there is a $((1-\alpha)n, n^{-c})$-extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^{\alpha n}$ for $\mathsf{SIZE}[n^c]$-sample sources. Moreover, Ext is computable in time $\mathsf{poly}(n^c)$.*

We note that the following lemma due to Li and Zuckerman [LZ19] gives extractors for sources recognized by small circuits in the very high min entropy regime from better assumptions.

Lemma 2.17 ([LZ19])**.** *Let $\mathcal{C}$ be a flip-invariant family of boolean functions over n bits. For any $\Delta = \Delta(n) > 0$ If G is a seed-extending ϵ-pseudorandom generator $G : \{0,1\}^d \to \{0,1\}^n$ for $\mathcal{C}$, then there exists an $(n - \Delta, 2^\Delta \epsilon)$-extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^{n-d}$ for $\mathcal{C}$-recognizable sources.*

The following is an immediate corollary of Lemma 2.17 and Theorem 2.12.

Corollary 2.18. *If E requires exponential size circuits, then for any constant c, there exists an $(n - c\log n, n^{-c})$-extractor for $\mathsf{SIZE}[n^c]$-recognizable sources.*

We also require the following classical results on approximate counting and sampling NP-witnesses.

Theorem 2.19 (Approximate Counting with an NP-oracle [JVV86]). *For every $i \geq 0$, every sufficiently large s and every $\epsilon > 0$, there is a Σ_{i+1}-circuit A of size $\mathsf{poly}(s/\epsilon)$ that given a Σ_i-circuit $C : \{0,1\}^n \to \{0,1\}$ of size s outputs a value $\widehat{M}$ such that*

$$\widehat{M} \in (1 \pm \epsilon)|\{x : C(x) = 1\}|$$

Theorem 2.20 (Sampling Witnesses with an NP-oracle [BGP00, JVV86]). *For every $i \geq 0$, every sufficiently large s and every $\delta > 0$, there is a randomized Σ_{i+1}-circuit A of size $\mathsf{poly}(s/\log(1/\delta))$ that given a Σ_i-circuit $C : \{0,1\}^n \to \{0,1\}$ of size s outputs a value in $\{0,1\}^n \cup \bot$ such that for every size s Σ_i-circuit, $\Pr[A(C) = \bot] \leq \delta$ and the distribution $(A(C)|A(C) \neq \bot)$ is uniform over $\{x : C(x) = 1\}$.*

Proposition 1. *Let X be a random variable and f a function. Define $Y = f(X)$. For any ϵ and any random variable Y',*

$$\Delta(X; (X|f(X) = Y')) = \Delta(Y; Y').$$

The proof of Proposition 1 can be found in Appendix E.1.

2.6 Other useful facts

Proposition 2 (Implicit in Lemma 3.8.1 in [Vad99]). *Let X_0, X_1 be random variables such that $\Delta(X_0; X_1) = \epsilon$.Consider the following game:*

- *Arthur samples a coin $b \leftarrow \mathcal{U}$ and gives Merlin $x \leftarrow X_b$.*
- *Merlin responds with b'. If $b' = b$, Merlin wins. Otherwise, Merlin loses.*

Merlin wins with probability $\frac{1+\epsilon}{2}$ by outputting b' such that $\Pr[X_{b'} = x] \geq \Pr[X_{1-b'} = x]$. Moreover, this strategy is optimal.

Promise problems generalize the concept of languages that give a better handle on semantic complexity classes. A promise problem, Π, consists of a set of Yes instances, Π_Y, and a disjoint set of No instances, Π_N. A machine is considered to decide Π if on input x *promised* to be in $\Pi_Y \cup \Pi_N$ it accepts x if and only if $x \in \Pi_Y$. In other words, the machine should accept Π_Y and reject Π_N, but can behave arbitrarily elsewhere.

Lemma 2.21 ([GS86, Bab85, BM88, AASY16])**.** *For any polynomial $s(n)$there exists a polynomial $s'(n)$ such that the following holds.*

For any $n \in \mathbb{N}$, if $\Pi = (\Pi_Y, \Pi_N)$ is a promise problem with a constant-round private coin interactive proof where the verifier is an O-oracle-aided machine that runs in deterministic time $s(n)$ with $s(n)$ bits of non-uniform advice with a gap between soundness and completeness of $1/s(n)$ on inputs of length n, then Π is decided by a nondeterministic O-oracle-aided circuit[23] *of size $s'(n)$ on inputs of length n.*

We also use the following simple combinatorial propositions.

Proposition 3. *Let $c > 1$. Let $(XY), (XZ)$ be two joint random variables supported on any space $\Sigma_X \times \Sigma$ such that $\Delta(X, Y; X, Z) \leq \epsilon$, then there exists an event $S \subseteq \Sigma_X$ such that*

1. $\Pr[X \in S] \geq 1 - 1/c$
2. $\forall x \in S, \Delta(XY|X = x; XZ|X = x) \leq c\epsilon$*, where $(XY|X = x)$ denotes the random variable XY conditioned on $X = x$ and, similarly, $(XZ|X = x)$ denotes the random variable XZ conditioned on $X = x$.*

Proof. Let A the variable distributed according to the procedure where $x \leftarrow X$ and then $\Delta(XY|X = x; XZ|X = x)$ is output. By definition, $\mathbb{E}[A] \leq \epsilon$.[24] Thus, by Markov's inequality we have

$$\Pr[A \geq c\epsilon] \leq \frac{\mathbb{E}[A]}{c\epsilon} \leq \frac{1}{c}$$

It follows that there exists a set S with the desired properties. In particular, S is the set of x such that A conditioned on $X = x$ is *not* greater than $c\epsilon$. □

Proposition 4. *Let $\beta \in (0, 1)$ Let $(XY), (XZ)$ be two joint random variables supported on any space $\Sigma_X \times \Sigma$ such that $\Delta(X, Y; X, Z) > \epsilon$, then exists an event $S \subseteq \Sigma_X$ such that*

1. $\Pr[X \in S] \geq \epsilon - \beta$

[23] [AASY16] observed this lemma and made use of it in a related context. The lemma is a result of composing a variety of classical transformations. [GS86] show that any constant round IP can be transformed into a constant round *public coin* interactive proof (or, a constant round arthur merlin proof system). [Bab85, BM88] show that any constant round public coin interactive proof can be transformed into an *AM* protocol. Finally, applying Adleman's trick [Adl78] to the AM protocol yields a *non-uniform*, non-deterministic circuit at most polynomially larger than Arthur's complexity. Here, we additionally observe that all of these transformations relativize.

[24] $\mathbb{E}[A] = \sum_{x \in \Sigma_X} \Pr[X = x]\Delta(XY|X = x; XZ|X = x) = \Delta(XY; XZ)$.

2. *$\forall x \in S, \Delta(XY|X = x; XZ|X = x) \geq \beta$, where $(XY|X = x)$ denotes the random variable XY conditioned on $X = x$ and, similarly, $(XZ|X = x)$ denotes the random variable XZ conditioned on $X = x$.*

Proof. Let A the variable distributed according to the procedure where $x \leftarrow X$ and then $\Delta(XY|X = x; XZ|X = x)$ is output. Let $A' = 1 - A$. By definition, $\mathbb{E}[A'] < 1 - \epsilon$. Thus, by Markov's inequality we have,

$$\Pr[A' \geq 1 - \beta] < \frac{1-\epsilon}{1-\beta}.$$

So, it follows that $\Pr[A \leq \beta] < \frac{1-\epsilon}{1-\beta}$. Thus

$$\Pr[A > \beta] \geq 1 - \frac{1-\epsilon}{1-\beta} = \frac{\epsilon-\beta}{1-\beta} \geq \epsilon - \beta.$$

It follows that there exists a set S with the desired properties. In particular, S is the set of x such that A conditioned on $X = x$ is greater than β. □

3 Non-Malleable Extractors for Samplable and Recognizable Sources

In this section, we present our seedless non-malleable extractors for samplable and recognizable sources resistant to tampering by polynomial size circuits. In Sections 3.1 and 3.2, we will prove they are *relaxed* seedless non-malleable extractors (where we only consider the case of tampering functions with no fixed points). We conclude by observing a connection first articulated by Cheraghchi and Guruswami [CG14b] for both the case of (a) sources recognizable by and tampered by n^c-size circuits and (b) sources samplable by and tampered by n^c-size circuits, relaxed seedless non-malleable extractors are strong seedless non-malleable extractors, albeit with a small hit to entropy tolerance (see Section 3.3). Finally, in Section 3.4 we combine the results from Sections 3.1, 3.2, and 3.3 to obtain our main theorem statements.

The starting point of our construction is the non-malleable "extractor" for uniform sources and n^c tampering sketched in the introduction: $\mathsf{Ext}(s, x) \mapsto \text{2NMExt}(\mathsf{G}(s), x)$ where 2NMExt is a two-source non-malleable extractor and G is a seed-extending PRG for nondeterministic circuits of size n^d, for some constant $d \geq c$. One may hope that this construction yields a non-malleable extractor for a richer class of sources[25], however it seems critical for any reduction to the security of a seed-extending PRG that the seed s is at least somewhat random (which is may not be the case if we only promise linear min-entropy in the entire (s, x) string).

Instead, to build non-malleable extractors for sources recognizable by n^c-size circuits or samplable n^c-size circuits, we show it suffices to simply extract the seed using an appropriate "computational" extractor (not non-malleable). A caveat though, is we now require a stronger PRG that guarantees pseudorandomness against nondeterministic NP-circuits of polynomial size. This is because in our reduction breaking the pseudorandomness of the PRG, Arthur needs to sample from an arbitrary samplable or recognizable source X, with the additional condition that the output of the underlying computational extractor on this source is equal to some fixed seed σ that Arthur receives as input. To do this conditional sampling, we rely on a classical result of Bellare et al. [BGP00] that shows how to efficiently sample uniform witnesses given an NP-oracle, and we give Arthur access to an NP-oracle. Thus, Arthur is now a bounded polynomial size NP-circuit and, ultimately, after collapsing the AM protocol down, our PRG distinguisher is a nondeterministic NP-circuit of bounded polynomial size.

[25] Indeed, the argument sketched in the intro should extend to sources of the form S, X where S is uniform and independent of X, any source with min entropy $(1-\gamma)n$ for some small constant γ.

3.1 Relaxed Non-Malleable Extractors for Samplable Sources

Figure 3.1: Non-Malleable Extractor for Samplable Sources

Let $k(n), s(n), s'(n), \gamma$ be as in Lemma 3.1. Let $\mathsf{Ext}_{\text{samp}}$ be an extractor with error $\gamma(n)$ for n-bit sources samplable by size $s(n)$ circuits, computable in time $\mathsf{poly}(s(n))$. Let 2NMExt be a strong two-source non-malleable extractor with error $\delta(n)$ for independent sources of length n where the left has min-entropy at least $n - \ell(n)$ and the right has min-entropy at least $k(n) - 2\ell(n) - 2\log(s(n)) - 10$, computable in time $\mathsf{poly}(s)$. Let G be a seed-extending PRG for nondeterministic NP circuits of size $s'(n)$.

$$\mathsf{NMExt}_{\text{samp}} : x \mapsto \text{2NMExt}(\mathsf{G}(\mathsf{Ext}_{\text{samp}}(x)), x)$$

Lemma 3.1. *For any polynomial $s(n)$ and function $k(n)$ such that $0 \leq k(n) \leq n$, there exists polynomial $s'(n) = \Omega(s(n))$ such that the following is true.*

If

- *$\mathsf{G} : \{0,1\}^{\ell(n)} \to \{0,1\}^n$ is a seed-extending PRG for nondeterministic NP-circuits of size $s'(n)$ with seed length $\ell(n)$.*
- *$\mathsf{Ext}_{samp} : \{0,1\}^n \to \{0,1\}^{\ell}(n)$ is a γ-extractor for (n,k) sources samplable by $s(n)$-size circuits computable in time $\mathsf{poly}(s(n))$, where $\gamma \leq 1/6s(n)$.*
- *$\text{2NMExt} : \{0,1\}^{2n} \to \{0,1\}^m$ is a strong two-source non-malleable extractor with error $\delta(n) < 1/1000(s(n))^2$ for two independent n-bit sources where the left source has min-entropy at least $n - \ell(n)$ and the right has min-entropy at least $k(n) - 2\ell(n) - 2\log(s(n)) - 10)$. Moreover, 2NMExt should be computable in time $\mathsf{poly}(s(n))$.*

then the construction, $\text{NMExt} : \{0,1\}^n \to \{0,1\}^m$, in Figure 3.1 is a relaxed seedless non-malleable extractor for n-bit sources with $k(n)$-min entropy samplable by size $s(n)$ circuits with respect to $\mathsf{SIZE}[s(n)]$-tampering and error $1/s(n)$.

Proof. Let $\epsilon = 1/s(n)$.

Suppose for the sake of contradiction that there exists a $s(n)$-samplable (n,k)-source X and a tampering function, $\tau : x \mapsto \tilde{x}$ in $\mathsf{SIZE}[s(n)]$ with no fixed points, that breaks the non-malleability guarantee.

Now, our assumption on τ can be restated as

$$\Delta(\text{2NMExt}(\mathsf{G}(\mathsf{Ext}_{\text{samp}}(X)), X), \text{2NMExt}(\mathsf{G}(\mathsf{Ext}_{\text{samp}}(\tau(X))), \tau(X)); \mathcal{U}_m, \text{2NMExt}(\mathsf{G}(\mathsf{Ext}_{\text{samp}}(\tau(X))), \tau(X)) \geq \epsilon. \tag{2}$$

We will use this assumption to distinguish the seed-extending PRG, G, from the uniform distribution via an interactive proof. In more detail, recall that the guarantee of $\mathsf{G} : \{0,1\}^\ell \to \{0,1\}^n$ says that for any non-deterministic NP circuit, C, of size $s'(n)$,

$$\Delta(C(\mathsf{G}(\mathcal{U}_\ell)); C(\mathcal{U}_n)) < 1/s'(n).$$

We show that there exists a circuit C of size at most $s'(n)$ that does not obey this inequality. We do this by following the approach of [AASY16] and constructing a private coin, constant round interactive proof protocol (see Figure 3.2) where Arthur is an NP-circuit of size at most $\mathsf{poly}(s(n))$ for a promise problem, $\Pi = (\Pi_Y, \Pi_N)$ where Π_Y is dense under G and Π_N is dense under the uniform distribution. By standard transformations [BM88, GS86] (Lemma 2.21), this results in a nondeterministic NP-circuit of size $s'(n) = \mathsf{poly}(s(n))$ that decides the same problem, and hence breaks the PRG.

Looking ahead, Assumption (2) above on *malleability* of the resulting extractor when provided pseudorandom inputs will enable us to prove the protocol is complete, i.e. Arthur accepts pseudorandom inputs with high probability. Soundness, i.e. Arthur rejects random inputs with high probability, will ultimately follow from security of the 2-source non-malleable extractor.

To do this, Arthur will run the non-malleability experiment himself, using Merlin to evaluate the PRG. In order to render the experiment consistent with the PRG seed in the PRG security game, s, Arthur uses his NP-oracle to efficiently sample X conditioned on the samplable-source extractor outputting s. Furthermore, what ultimately will enable our soundness argument to go through is the fact that to achieve completeness Arthur needs to communicate very little about each sample X and thus X remains entropic, even after conditioning on this communication.

We then conclude by using a standard private coin technique [GMW91], to distinguish between the real and ideal relaxed non-malleable extractor experiments, if and only if the inputs are pseudorandom.

However, giving our interactive proof we need the following claim.

Claim 3.1. For any α, if X is samplable in time $s(n)$ and $\mathsf{Ext}_{\mathrm{samp}}$ is computable in time $\mathsf{poly}(s(n))$, then for any σ, the there is a $\mathsf{poly}(s(n), \log(1/\alpha))$ time procedure that uses an NP oracle that with probability $1-\alpha$ outputs identically to $(X|\mathsf{Ext}_{\mathrm{samp}}(X) = \sigma)$ and otherwise outputs $\bot$.

Proof of Claim 3.1. Because X is samplable there exists a size $s(n)$ circuit C_X such that $C_X(\mathcal{U}) \equiv X$. Consider the circuit C' that outputs 1 on an input u if and only if $\mathsf{Ext}_{\mathrm{samp}}(C_X(u)) = \sigma$. Let R denote the uniform distribution on $\{u : C'(u) = 1\}$. Note that $C_X(R) \equiv (X|\mathsf{Ext}_{\mathrm{samp}}(X) = \sigma)$. Thus, Theorem 2.19 says we can sample R with probability $1-\alpha$ (and $\bot$ otherwise) in time $\mathsf{poly}(s(n))$ using an NP-oracle. Thus, we can simply output $C_X(R)$ if the procedure from Theorem 2.19 does not output $\bot$ (with an additional $s(n)$ complexity), and output $\bot$ otherwise. □

Figure 3.2: Interactive Proof for distinguishing G from uniformly random bits

Let $\mathsf{Ext}_{\mathrm{samp}}$ be an extractor with error $\gamma(n)$ for n-bit sources samplable by size $s(n)$ circuits, computable in time $\mathsf{poly}(s(n))$. Let 2NMExt be a strong two-source non-malleable extractor with error $\delta(n)$ for independent sources of length n where the left has min-entropy at least $n - \ell(n)$ and the right has min-entropy at least $k - 2\ell(n) - 2\log(s(n)) - 10$, computable in time $\mathsf{poly}(s(n))$. Let G be a seed-extending PRG for nondeterministic NP circuits of size $s(n)$.
Recall that X is the $s(n)$-samplable source and τ the tampering attack from our assumption.
Our protocol aims to accept strings from $G(\mathcal{U}_\ell)$ when Merlin plays according to below (completeness) and reject strings from $\mathcal{U}_n$ regardless of the strategy Merlin utilizes (soundness).

On input (σ, y)**,**

Arthur Sample $x \leftarrow (X|\mathsf{Ext}_{\mathrm{samp}}(X) = \sigma)$ with probability at least $1 - \bar{\alpha}$, where $\bar{\alpha} = 1/2$, using procedure from Claim 3.1). If procedure outputs $\bot$, immediately accept or reject at random. Otherwise, set $\tilde{x} = \tau(x)$ and send Merlin $\tilde{\sigma} = \mathsf{Ext}_{\mathrm{samp}}(\tilde{x})$.

Merlin If $(\sigma, y) = \mathsf{G}(\sigma)$, respond $\tilde{y}$ such that $(\tilde{\sigma}, \tilde{y}) = \mathsf{G}(\tilde{\sigma})$. Otherwise, respond arbitrary $\tilde{y}$.

Arthur Sample a random coin $b \leftarrow \mathcal{U}$ and set $\tilde{z} = \mathrm{2NMExt}((\tilde{\sigma}, \tilde{y}), \tilde{x})$.

- *If* $b = 0$*:* Sample $z \leftarrow \mathcal{U}_m$ and send $z, \tilde{z}$.
- *Else if* $b = 1$*:* Sample $z \leftarrow \mathrm{2NMExt}((\sigma, y), x)$ and send $z, \tilde{z}$.

Merlin (Guess Arthur's bit.) If

$$\Pr_{\mathcal{U}_m, X}[(\mathcal{U}_m, \mathrm{2NMExt}((\tilde{\sigma}, \tilde{y}), \tau(X))) = (z, \tilde{z}) | \mathsf{Ext}_{\mathrm{samp}}(X) = \sigma, \mathsf{Ext}_{\mathrm{samp}}(\tau(X)) = \tilde{\sigma}]$$

is upper bounded by

$$\Pr_{X}[(\mathrm{2NMExt}((\sigma, y), X), \mathrm{2NMExt}((\tilde{s}, \tilde{y}), \tau(X))) = (z, \tilde{z}) | \mathsf{Ext}_{\mathrm{samp}}(X) = \sigma, \mathsf{Ext}_{\mathrm{samp}}(\tau(X)) = \tilde{\sigma}],$$

set $b' = 1$. Otherwise, set $b' = 0$. Respond b'.

Arthur Accept if $b = b'$, and reject otherwise.

Claim 3.2. For any $\beta \in (0,1)$, there exists a set Π_Y^β such that

1. Π_Y is noticeably dense in G: $\Pr_{\sigma \overset{u}{\leftarrow} \{0,1\}^\ell}[\mathsf{G}(\sigma) \in \Pi_Y^\beta] \geq \epsilon - \gamma - \beta$
2. Arthur accepts inputs in Π_Y^β with probability $> \frac{1+(1-\bar{\alpha})\beta}{2}$ when playing with (honest) Merlin (as prescribed in Figure 3.2).

Proof. We begin by considering how the protocol behaves on *random* inputs distributed according to $(\Sigma, \mathsf{G}(\Sigma))$ where Σ is uniform ($\Sigma \equiv \mathcal{U}_\ell$).

In particular, by the guarantee of $\mathsf{Ext}_{\text{samp}}$, we have that $\mathsf{Ext}_{\text{samp}}(X) \approx_\gamma \mathcal{U}_\ell \equiv \Sigma$. It follows from Proposition 1 that

$$(\Sigma, X | \mathsf{Ext}_{\text{samp}}(X) = \Sigma) \approx_\gamma (\mathsf{Ext}_{\text{samp}}(X), X).$$

Therefore, if we let X' denote $(X|\mathsf{Ext}_{\text{samp}}(X) = \Sigma)$ (recall that $\mathsf{Ext}_{\text{samp}}(X') \equiv \Sigma$) it follows from postprocessing that

$$\begin{gathered}\text{2NMExt}(\mathsf{G}(\mathsf{Ext}_{\text{samp}}(X)), X), \text{2NMExt}(\mathsf{G}(\mathsf{Ext}_{\text{samp}}(\tau(X))), \tau(X)) \approx_\gamma \\ \text{2NMExt}(\mathsf{G}(\mathsf{Ext}_{\text{samp}}(X')), X'), \text{2NMExt}(\mathsf{G}(\mathsf{Ext}_{\text{samp}}(\tau(X'))), \tau(X'))\end{gathered}$$

Thus if we temporarily condition on Arthur's initial sampling procedure not outputting $\perp$, observe that if Arthur chooses $b = 1$, it follows from the above that Arthur's last message is γ-close to

$$\text{2NMExt}(\mathsf{G}(\mathsf{Ext}(X)), X), \text{2NMExt}(\mathsf{G}(\tau(X)), \tau(X)) \equiv \mathsf{NMExt}_{\text{samp}}(X), \mathsf{NMExt}_{\text{samp}}(\tau(X)).$$

On the other hand (still conditioning on successful sampling), if $b = 0$, Arthur's last message is γ-close to

$$\mathcal{U}_m, \text{2NMExt}(\mathsf{G}(\tau(X)), \tau(X)) \equiv \mathcal{U}_m, \mathsf{NMExt}_{\text{samp}}(\tau(X)).$$

By our assumption, these two distributions are ϵ-far from each other.

By Proposition 4 and triangle inequality, this implies there exists a set Π_Y^β such that for any $(\sigma, y) \in \Pi_Y^\beta$ the distributions of Arthur's last message in the case that $b = 1$ is β-far from the distribution when $b = 0$, and moreover $\Pr[\mathsf{G}(\Sigma) \in \Pi_Y^\beta] \geq \epsilon - \gamma - \beta$.

So by Proposition 2.6, for any $\mathsf{G}(\sigma) \in \Pi_Y^\beta$, Merlin guesses correctly with probability $\geq \frac{1+\beta}{2}$.

Thus relaxing the condition on Arthur's sampling (which fails with probability at most $\bar{\alpha}$), we can bound Arthur's acceptance probability as follows. The 3rd inequality follows because the line above is minimized when Pr[sampling fails] $= \bar{\alpha}$.

$$\begin{aligned}\forall (s, y) \in \Pi_Y^\beta, \Pr[\text{Arthur accepts}(s, y)] &\geq \Pr[\text{sampling fails}]\frac{1}{2} + \Pr[\text{sampling succeeds}]\frac{\beta}{2} \\ &\geq \bar{\alpha}\frac{1}{2} + (1 - \bar{\alpha})\frac{1+\beta}{2} \\ &= \frac{1 + (1-\bar{\alpha})\beta}{2}.\end{aligned}$$

□

Claim 3.3. For any $c > 1$ and $\zeta(n) \in (0, 1)$ such that $k'(n) \leq k(n) - 2\ell(n) - \log(1/\zeta(n))$ (where $k'(n)$ is the min-entropy requirement of the right source for 2NMExt), there exists a set Π_N^c such that

1. Π_N^c is large: $\Pr_{(\sigma,y) \overset{u}{\leftarrow} \{0,1\}^n}[(\sigma, y) \in \Pi_N^c] \geq 1 - 1/c$
2. Arthur accepts inputs in Π_N^c with probability $\leq \frac{1+c(\delta+\zeta)}{2}$ when playing with any (cheating) Merlin (as prescribed in Figure 3.2).

Proof. As in Claim 3.2, we will analyze the view of Merlin (up to guessing) on a random input and deduce that their exists a large Π_N which Arthur rejects with probability close to $1/2$. The important difference is that, here, Merlin can behave arbitrarily.

Observe that, that if we condition on success in Arthur's initial sampling, Arthur accepts if and only if Merlin guesses his bit, b, correctly ($b' = b$). It follows by Proposition 2.6 that there is an optimal (for any specific input, not just with respect to uniform inputs) Merlin strategy, M^*, that chooses messages to maximize the distance between his view when Arthur chooses $b = 0$ versus his view when $b = 1$. By the optimality of such a strategy, it suffices to consider just this M^*.

So, suppose the protocol in Figure 3.2 is given uniformly random inputs $(\sigma, y) \leftarrow \mathcal{U}_n$. Fix an optimal strategy, M^*. In particular, let $G^* : (\sigma, y, \tilde{\sigma}) \mapsto \tilde{y}$ be the function that given the transcript thus far, outputs Merlin's first message.

Now, note that if we condition on $\sigma, \tilde{\sigma}$, then $G^*(\sigma, y, \tilde{\sigma}) = \tilde{y}$ is independent of x, the string sampled by Arthur initially. And similarly, after conditioning on $\sigma, \tilde{\sigma}$, $\tilde{x} = \tau$ is independent of (σ, y). In other words, we can sample $(\sigma, y, x, \tilde{\sigma}, \tilde{y}, \tilde{x})$ identically as follows:

1. Sample σ uniformly at random and $\tilde{\sigma} \leftarrow \mathsf{Ext}_{\mathrm{samp}}(\tau(X_\sigma))$, where $X_\sigma \equiv X|\mathsf{Ext}_{\mathrm{samp}}(X) = \sigma$. (This is identically distributed to Figure 3.2.) Let $\Sigma, \tilde{\Sigma}$ denote these random variables.
2. Sample y uniformly at random, and sample x from $X_\sigma|\mathsf{Ext}_{\mathrm{samp}}(\tau(X_\sigma)) = \tilde{\sigma}$. Note that conditioned on $\sigma, \tilde{\sigma}$, x and y are independent. Let Y, $X_{\sigma,\tilde{\sigma}}$ denote the random variables corresponding to how y and x are sampled here, respectively.
3. Apply the tampering:
   - $\tau_L^{\sigma,\tilde{\sigma}} : (\sigma, y) \mapsto \tilde{\sigma}, \tilde{y}$ where $\tilde{y} = G^*(\sigma, y, \tilde{\sigma})$
   - $\tau_R^{\sigma,\tilde{\sigma}} : x \mapsto \tilde{x} = \tau(x)$

   Thus, conditioned on $\sigma, \tilde{\sigma}$, $(\tau_L^{\sigma,\tilde{\sigma}}, \tau_R^{\sigma,\tilde{\sigma}})$ is a split-state tampering. Moreover, because τ has no fixed points $\tau_R^{\sigma,\tilde{\sigma}}$ doesn't either.

Thus, for any (valid) fixed choice of $\sigma, \tilde{\sigma}$, $Y, X_{\sigma,\tilde{\sigma}}$ are independent, and $(\tau_L^{\sigma,\tilde{\sigma}}, \tau_R^{\sigma,\tilde{\sigma}})$ is a split-state tampering function with no fixed points. Clearly, Y is always uniformly distributed, so Y has min-entropy $n - \ell$ for any fixed choice of s. Intuitively, $X_{\sigma,\tilde{\sigma}}$ should have not lost much min-entropy on average relative to X because $\sigma, \tilde{\sigma}$ are short. We next formalize this intuition.

For any $\zeta \in (0, 1)$, let T_ζ denote the set of $(\sigma, \tilde{\sigma})$ that occur with probability at least $\zeta \cdot 2^{-2\ell}$. Note that $\Pr[(\Sigma, \tilde{\Sigma}) \in T] \geq 1 - \zeta$, because

$$\Pr[(\Sigma, \tilde{\Sigma}) \notin T] \leq \sum_{(\sigma,\tilde{\sigma})} \zeta 2^{-2\ell} = 2^{2\ell}\zeta 2^{-2\ell} = \zeta.$$

Now, for any $(\sigma, \tilde{\sigma}) \in T_\zeta$ and any $x \in \{0, 1\}^n$, we have

$$\begin{aligned}
\Pr[X_{\sigma,\tilde{\sigma}} = x] &= \Pr[X = x|(\Sigma, \tilde{\Sigma}) = (\sigma, \tilde{\sigma})] \\
&\leq \frac{\Pr[X = x \wedge (\Sigma, \tilde{\Sigma}) = (\sigma, \tilde{\sigma})]}{\Pr[(\Sigma, \tilde{\Sigma}) = (\sigma, \tilde{\sigma})]} \\
&\leq \frac{\Pr[X = x]}{\Pr[(\Sigma, \tilde{\Sigma}) = (\sigma, \tilde{\sigma})]} \\
&\leq \frac{2^{-k}}{\zeta 2^{-2\ell}}
\end{aligned}$$

Thus, for any $(\sigma, \tilde{\sigma}) \in T_\zeta$ (which happens with probability at least $1 - \zeta$), $H_\infty(X_{\sigma,\tilde{\sigma}}) \geq k - 2\ell - \log(1/\zeta)$. Thus, conditioned on $\sigma, \tilde{\sigma} \in T_\zeta$ and Arthur's coin $b = 0$, Merlin's view is simply

$$D_0^{\sigma,\tilde{\sigma}} \equiv (\sigma, y), \mathcal{U}_m, 2\mathrm{NMExt}(\tau_L^{\sigma,\tilde{\sigma}}(\sigma, y), \tau_R^{\sigma,\tilde{\sigma}}(x)).$$

On the other hand, if Arthur's coin is $b = 1$, Merlin's view is

$$D_1^{\sigma,\tilde{\sigma}} \equiv (\sigma, y), 2\mathrm{NMExt}((\sigma, y), x), 2\mathrm{NMExt}(\tau_L^{\sigma,\tilde{\sigma}}(\sigma, y), \tau_R^{\sigma,\tilde{\sigma}}(x)).$$

Because 2NMExt is a strong two-source non-malleable extractor with error δ for independent sources where the left has min entropy at least $n - \ell$ and the right has min entropy at least $k - 2\ell + \log(1/\zeta)$, we have that for any (worst-case) choice of $\sigma, \tilde{\sigma} \in T$,

$$\Delta(D_0^{\sigma,\tilde{\sigma}}; D_1^{\sigma,\tilde{\sigma}}) \leq \delta.$$

From the fact that $\Pr[(\Sigma,\tilde{\Sigma}) \in T] \geq 1-\delta$, it follows that Merlin's views are at most $\zeta+\delta$ distinguishable:

$$\begin{aligned}
\Delta(D_0^{\Sigma,\tilde{\Sigma}}; D_1^{\Sigma,\tilde{\Sigma}}) &= \sum_{\sigma,\tilde{\sigma}} \Pr[(\Sigma,\tilde{\Sigma}) = (\sigma,\tilde{\sigma})] \Delta(D_0^{\sigma,\tilde{\sigma}}; D_1^{\sigma,\tilde{\sigma}}) \\
&= \sum_{\sigma,\tilde{\sigma} \notin T_\zeta} \Pr[(\Sigma,\tilde{\Sigma}) = (\sigma,\tilde{\sigma})] \Delta(D_0^{\sigma,\tilde{\sigma}}; D_1^{\sigma,\tilde{\sigma}}) + \sum_{\sigma,\tilde{\sigma} \in T_\zeta} \Pr[(\Sigma,\tilde{\Sigma}) = (\sigma,\tilde{\sigma})] \Delta(D_0^{\sigma,\tilde{\sigma}}; D_1^{\sigma,\tilde{\sigma}}) \\
&\leq \zeta + \sum_{(\sigma,\tilde{\sigma}) \in T_\zeta} \Pr[(\Sigma,\tilde{\Sigma}) = (\sigma,\tilde{\sigma})]\delta \\
&\leq \zeta + \delta
\end{aligned}$$

Thus, by Proposition 3 there exists a set Π_N^c such that $\Pr_{(\sigma,y)\xleftarrow{u}\{0,1\}^n}[(\sigma,y) \in \Pi_N^c] \geq 1-1/c$ and for any $(\sigma,y) \in \Pi_N^c$, Merlin's views are at most $c(\delta+\zeta)$ distinguishable.

It follows from Proposition 2.6 that for any strategy of Merlin and any input $(\sigma,y) \in \Pi_N$, $\Pr[b' = b] \leq \frac{1+c(\delta+\zeta)}{2}$.

So if we relax the condition on Arthur's sampling (if sampling fails, Arthur accepts with probability $\frac{1}{2} \leq \frac{1+c(\delta+\zeta)}{2}$) we can bound Arthur's acceptance probability as follows. The second inequality follows because the line above is maximized when $\alpha = 0$ (because $1/2 + c(\delta+\zeta)/2 > 1/2$).

$$\begin{aligned}
\forall(\sigma,y) \in \Pi_N^c,\ \Pr[\text{Arthur accepts } (\sigma,y)] &\leq \Pr[\text{sampling succeeds}]\frac{1}{2} + \Pr[\text{sampling fails}]\frac{1+c(\delta+\zeta)}{2} \\
&\leq \frac{1+c(\delta+\zeta)}{2}.
\end{aligned}$$

□

We conclude from Claim 3.2 and Claim 3.3, that for any $c > 1$, $\beta \in (0,1)$, and $\zeta \in (0,1)$ such that $d - 2\ell \geq \log(1/\zeta)$ there is a constant round IP protocol where Arthur can be represented by NP-circuit of size $\mathsf{poly}(s(n))$ that recognizes $\Pi = (\Pi_Y^\beta, \Pi_N^c)$ with completeness/soundness gap

$$|\frac{1+(1-\bar{\alpha})\beta}{2} - \frac{1+c(\delta+\zeta)}{2} = |\frac{(1-\bar{\alpha})\beta - c(\delta+\zeta)}{2}|$$

Now we choose $\bar{\alpha} = 1/2$, $\beta = \epsilon/6$, $c = 6/\epsilon$, and $\zeta = \epsilon^2/1000$ (and $\delta \leq \epsilon^2/1000$). By our choice of ζ and our assumption on 2NMExt, we have $d - 2\ell \geq \log(1/\zeta) \geq 10 + 2\log(s(n))$ so that we can lower bound the completeness/soundness gap by

$$\frac{(1-\bar{\alpha})\beta - c(\delta+\zeta)}{2} \geq \frac{\epsilon}{24} - \frac{3(\delta+\zeta)}{\epsilon} \geq \frac{\epsilon}{24} - \frac{3\epsilon}{500} > \frac{\epsilon}{100} = \frac{1}{100s(n)}$$

Thus by Lemma 2.21, this implies the existence of an $s'(n)$-size nondeterministic NP circuit, $\mathcal{C}$, (where $s'(n) = \mathsf{poly}(s(n)) \geq s(n)$) that decides the promise problem, Π. Because $\Pi_Y^{\epsilon/6}$ is $5\epsilon/6 - \gamma$-dense under G (i.e. $\Pr_s[\mathsf{G}(s) \in \Pi_Y^{\epsilon/6}] \geq 5\epsilon/6 - \gamma$) and $\Pi_N^{6/\epsilon}$ is $1-\epsilon/6$-dense under the uniform distribution (i.e. $\Pr_z[z \notin \Pi_N^{6/\epsilon}] \leq \epsilon/6$), the nondeterministic NP circuit $\mathcal{C}$ can distinguish with advantage at least (by our assumption that $\gamma \leq \epsilon/6$)

$$|(5\epsilon/6 - \gamma) - \epsilon/6| \geq \epsilon/2 \geq 1/s'(n).$$

The first inequality follows from our assumption that $\gamma \leq \epsilon/6$ and the second follows from the fact that $2/\epsilon = 2s(n) \leq s'(n)$. In conclusion, our initial assumption towards contradiction must be false.

□

3.2 Relaxed Non-Malleable Extractors for Recognizable Sources

Figure 3.3: Non-Malleable Extractor for Recognizable Sources

Let $k(n), s(n), s'(n), \gamma$ be as in Lemma 3.1. Let $\mathsf{Ext}_{\text{rec}}$ be an extractor with error $\gamma(n)$ for n-bit sources samplable by size $s(n)$ circuits, computable in time $\mathsf{poly}(s(n))$. Let 2NMExt be a strong two-source non-malleable extractor with error $\delta(n)$ for independent sources of length n where the left has min-entropy at least $n - \ell(n)$ and the right has min-entropy at least $k(n) - 2\ell(n) - 2\log(s(n)) - 10$, computable in time $\mathsf{poly}(s)$. Let G be a seed-extending PRG for nondeterministic NP circuits of size $s'(n)$.

$$\mathsf{NMExt}_{\text{rec}} : x \mapsto \text{2NMExt}(\mathsf{G}(\mathsf{Ext}_{\text{rec}}(x)), x)$$

Lemma 3.2. *For any polynomial $s(n)$ and function $k(n)$ such that $0 \leq k(n) \leq n$, there exists polynomial $s'(n) = \Omega(s(n))$ such that the following is true.*

If

- $\mathsf{G} : \{0,1\}^{\ell(n)} \to \{0,1\}^n$ *is a seed-extending PRG for nondeterministic* NP*-circuits of size $s'(n)$ with seed length $\ell(n)$.*
- $\mathsf{Ext}_{rec} : \{0,1\}^n \to \{0,1\}^\ell(n)$ *is a γ-extractor for (n,k) sources recognized by $s(n)$-size circuits that is computable in time* $\mathsf{poly}(s(n))$*, where $\gamma \leq 1/6s(n)$.*
- $\text{2NMExt} : \{0,1\}^{2n} \to \{0,1\}^m$ *is a strong two-source non-malleable extractor with error $\delta(n) < 1/1000(s(n))^2$ for two independent n-bit sources where the left source has min-entropy at least $n - \ell(n)$ and the right has min-entropy at least $k(n) - 2\ell(n) - 2\log(s(n)) - 10$). Moreover,* 2NMExt *should be computable in time* $\mathsf{poly}(s(n))$.

then the construction, $\text{NMExt} : \{0,1\}^n \to \{0,1\}^m$*, in Figure 3.1 is a relaxed seedless non-malleable extractor for n-bit sources with $k(n)$ min-entropy recognized by size $s(n)$ circuits that is resilient to* $\mathsf{SIZE}[s(n)]$*-tampering with error $1/s(n)$.*

The proof of this Lemma is nearly identical to that of Lemma 3.1. The only significant difference is that, here, in the Arthur Merlin Protocol we need to sample the recognizable source X conditioned on $\mathsf{Ext}_{\text{rec}}(X) = \sigma$ (compare to Claim 3.1). We provide the entire modified Arthur Merlin Proof System as well, for reference.

Claim 3.4. For any α, if X is recognizable by a size $s(n)$ circuit and $\mathsf{Ext}_{\text{rec}}$ is computable in time $\mathsf{poly}(s(n))$, then for any σ, the there is a $\mathsf{poly}(s(n), \log(1/\alpha))$ time procedure that uses an NP oracle that with probability $1 - \alpha$ outputs identically to $(X | \mathsf{Ext}_{\text{rec}}(X) = \sigma)$ and otherwise outputs $\perp$.

Proof of Claim 3.4. Because X is rec there exists a size $s(n)$ circuit C_X such that X is uniform on the set $\{x \in \{0,1\}^n : C_X(x) = 1\}$. Consider the circuit C' of size $\mathsf{poly}(s(n))$ that outputs 1 on input x if and only if $C_X(x) = 1$ and $\mathsf{Ext}_{\text{rec}}(x) = \sigma$. Clearly $(X | \mathsf{Ext}_{\text{rec}}(X) = \sigma)$ is uniform on the set of satisfying assignments for C', $\{x : C'(X) = 1\}$. There we can sample this set using procedure from Theorem 2.19 with probability $1 - \alpha$ in time $\mathsf{poly}(s(n), 1/\alpha)$ using an NP oracle. □

Figure 3.4: Interactive Proof for distinguishing G from uniformly random bits

Let $\mathsf{Ext}_{\text{rec}}$ be an extractor with error $\gamma(n)$ for n-bit sources recognizable by size $s(n)$ circuits, computable in time $\mathsf{poly}(s(n))$. Let 2NMExt be a strong two-source non-malleable extractor with error $\delta(n)$ for independent sources of length n where the left has min-entropy at least $n - \ell(n)$ and the right has min-entropy at least $k - 2\ell(n) - 2\log(s(n)) - 10$, computable in time $\mathsf{poly}(s(n))$. Let G be a seed-extending PRG for nondeterministic NP circuits of size $s(n)$.
Recall that X is the $s(n)$-size recognizable source and τ the tampering attack from our assumption.

On input (σ, y)**,**

Arthur Sample $x \leftarrow (X | \mathsf{Ext}_{\text{rec}}(X) = \sigma)$ with probability at least $1 - \bar{\alpha}$, where $\bar{\alpha} = 1/2$, using procedure from Claim 3.4). If procedure outputs $\perp$, immediately accept or reject at random. Otherwise, set $\tilde{x} = \tau(x)$ and send Merlin $\tilde{\sigma} = \mathsf{Ext}_{\text{rec}}(\tilde{x})$.

Merlin If $(\sigma, y) = \mathsf{G}(\sigma)$, respond $\tilde{y}$ such that $(\tilde{\sigma}, \tilde{y}) = \mathsf{G}(\tilde{\sigma})$. Otherwise, respond arbitrary $\tilde{y}$.

Arthur Sample a random coin $b \leftarrow \mathcal{U}$ and set $\tilde{z} = \text{2NMExt}((\tilde{\sigma}, \tilde{y}), \tilde{x})$.

- *If* $b = 0$*:* Sample $z \leftarrow \mathcal{U}_m$ and send $z, \tilde{z}$.
- *Else if* $b = 1$*:* Sample $z \leftarrow \text{2NMExt}((\sigma, y), x)$ and send $z, \tilde{z}$.

Merlin (Guess Arthur's bit.) If

$$\Pr_{\mathcal{U}_m, X}[(\mathcal{U}_m, \text{2NMExt}((\tilde{\sigma}, \tilde{y}), \tau(X))) = (z, \tilde{z}) | \mathsf{Ext}_{\text{rec}}(X) = \sigma, \mathsf{Ext}_{\text{rec}}(\tau(X)) = \tilde{\sigma}]$$

is upper bounded by

$$\Pr_{X}[(\text{2NMExt}((\sigma, y), X), \text{2NMExt}((\tilde{s}, \tilde{y}), \tau(X))) = (z, \tilde{z}) | \mathsf{Ext}_{\text{rec}}(X) = \sigma, \mathsf{Ext}_{\text{rec}}(\tau(X)) = \tilde{\sigma}],$$

set $b' = 1$. Otherwise, set $b' = 0$. Respond b'.

Arthur Accept if $b = b'$, and reject otherwise.

3.3 Removing the No-Fixed Points Assumption

Theorem 3.3. *Define $\mathcal{X}[k, s(n)]$ be the family of k-min-entropy sources on $\{0,1\}^n$ that are samplable by the class $\mathsf{SIZE}^n[s(n)]$. Assume* NMExt : $\{0,1\}^n \to \{0,1\}^m$ *is a relaxed, seedless ϵ-non-malleable extractor with respect to sources in $\mathcal{X}[k, s_s(n)]$ and tampering functions in $\mathsf{SIZE}^n[s_t(n)]$. Then* NMExt *is an ϵ'-seedless non-malleable extractor with respect to sources in $\mathcal{X}[k', s'_s(n)]$ and tampering functions in $\mathsf{SIZE}^n[s_t(n)]$, where*

- $k' := k + \log(1/\epsilon) + 1$
- $s'_s(n) := \frac{\epsilon}{k' \ln(2)} \cdot s_s(n) - s_t(n) - c \cdot n$ *for some constant c.*
- $\epsilon' := 3\epsilon$.

Proof. Fix tampering function $f \in \mathsf{SIZE}^n[s_t(n)]$ and $X \in \mathcal{X}[k', s'_s(n)]$. Let $\alpha := \Pr_{x \sim X}[f(x) = x]$. Note that WLOG $\epsilon \geq 1/2^k > 1/2^{k'}$. We consider two cases.

Case 1: $\alpha \geq 1 - \epsilon$. Since $f \in \mathsf{SIZE}^n[s_t(n)]$ and $X \in \mathcal{X}[k', s'_s(n)] \subseteq \mathcal{X}[k, s_s(n)]$, and since NMExt is also an ϵ-seedless extractor with respect to sources in $\mathcal{X}[k, s_s(n)]$ and tampering functions in $\mathsf{SIZE}^n[s_t(n)]$ we have that

$$\Delta\left((\text{NMExt}(X), \text{NMExt}(f(X))); (\mathcal{U}_m, \mathcal{U}_m)\right) \leq 2\epsilon.$$

Case 2: $\alpha \leq 1 - \epsilon$. Note that X is equivalent to a convex combination of distributions that draws from X_1 with probability α and X_2 with probability $1 - \alpha$, where X_1 is the distribution X conditioned on $f(X) = X$

and X_2 is the distribution X conditioned on $f(X) \neq X$. By the assumption on α, X_2 has min-entropy at least $k' - \log(1/\epsilon)$, and is $1/2^{k'}$-close to the following distribution X'.

Algorithm 1: Samplable distribution X' with no fixed points w.r.t. f

```
*A draw from $X'$ given randomness $r_1||\cdots||r_z$, where $z := \frac{k' \ln(2)}{\epsilon}$ is performed as follows:*;
**for** $i \in [z]$ **do**
    Use randomness $r_i$ to sample $x_i$ from $X$ using the $\mathsf{SIZE}^n[s'_s(n)]$ circuit;
    **if** $f(x_i) \neq x_i$ **then**
        Output $x_i$ and terminate
    **end**
**end**
**if** $f(x_i) = x_i \forall i \in [z]$ **then**
    Output a fixed value $x_f$ in the support of $X$ for which $f(x_f) \neq x_f$
**end**
```

The distribution X' is $1/2^{k'}$-close to X_2 since

$$\Pr[\forall i \in [z], f(x_i) = x_i] \leq (1-\epsilon)^z \leq \exp(-\epsilon \cdot z) = 2^{-k'},$$

where the final equality follows from the definition of z.

Further, by inspection of the sampling algorithm given for X', the fact that X_2 has min-entropy $k' - \log(1/\epsilon)$ for $\epsilon \leq 1$, and the fact that X' and X_2 are $1/2^{k'}$-close, we have that $X' \in \mathcal{X}[k' - \log(1/\epsilon) - 1, z \cdot (s'_s(n) + s_t(n) + c \cdot n)]$ for some constant c. By definition of $k', s'_s(n)$ we conclude that $X' \in \mathcal{X}[k, s_s(n)]$. Finally, note that by inspection of the sampling algorithm given for X', it can be seen that f has no fixed points with respect to X'.

Let D_f be the distribution that outputs `same` with probability α and outputs $\mathrm{NMExt}(f(X_2))$ with probability $1-\alpha$.

Then we have

$$\begin{align*}
&\Delta\left((\mathrm{NMExt}(X), \mathrm{NMExt}(f(X))); (\mathcal{U}_m, \mathrm{Copy}(D_f, \mathcal{U}_m))\right) \\
&\quad \leq \alpha \Delta\left((\mathrm{NMExt}(X_1), \mathrm{NMExt}(f(X_1))); (\mathcal{U}_m, \mathcal{U}_m)\right) \\
&\qquad + (1-\alpha)\Delta\left((\mathrm{NMExt}(X_2), \mathrm{NMExt}(f(X_2))); (\mathcal{U}_m, \mathrm{NMExt}(f(X_2)))\right) \tag{3} \\
&\quad \leq \alpha \Delta\left((\mathrm{NMExt}(X_1), \mathrm{NMExt}(f(X_1))); (\mathcal{U}_m, \mathcal{U}_m)\right) \\
&\qquad + (1-\alpha)\Delta\left((\mathrm{NMExt}(X'), \mathrm{NMExt}(f(X'))); (\mathcal{U}_m, \mathrm{NMExt}(f(X')))\right) \\
&\qquad + \Delta\left((\mathrm{NMExt}(X_2), \mathrm{NMExt}(f(X_2))); (\mathrm{NMExt}(X'), \mathrm{NMExt}(f(X')))\right) \\
&\qquad + \Delta\left((\mathcal{U}_m, \mathrm{NMExt}(f(X'))); (\mathcal{U}_m, \mathrm{NMExt}(f(X_2)))\right) \tag{4} \\
&\quad \leq \alpha \cdot \epsilon + (1-\alpha)3\epsilon \tag{5} \\
&\quad \leq 3\epsilon = \epsilon',
\end{align*}$$

where (3) and (4) follow from the triangle inequality, and (5) follows from the fact that X_2 and X' are $1/2^{k'}$-close (with $1/2^{k'} \leq \epsilon$) and the assumed properties of NMExt with respect to $X' \in \mathcal{X}[k, s_s(n)]$ and $f \in \mathsf{SIZE}^n[s_t(n)]$ with no fixed points. □

Theorem 3.4. *Define $\mathcal{X}[k, s(n)]$ be the family of k-min-entropy sources on $\{0,1\}^n$ that are recognizable by the class $\mathsf{SIZE}^n[s(n)]$. Assume $\mathrm{NMExt} : \{0,1\}^n \to \{0,1\}^m$ is a relaxed, seedless ϵ-non-malleable extractor with respect to sources in $\mathcal{X}[k, s_s(n)]$ and tampering functions in $\mathsf{SIZE}^n[s_t(n)]$. Then NMExt is an ϵ'-seedless non-malleable extractor with respect to sources in $\mathcal{X}[k', s'_s(n)]$ and tampering functions in $\mathsf{SIZE}^n[s_t(n)]$, where*

- $k' := k + \log(1/\epsilon)$
- $s'_s(n) := s_s(n) - s_t(n) - c \cdot n$*, for some constant c.*
- $\epsilon' := 2\epsilon$*.*

Sketch. The proof proceeds identically to the previous case, with the exception that it is actually easy to show that the source X_2 is recognizable by the class $\mathsf{SIZE}^n[s(n) = s'_s(n) + s_t(n) + c \cdot n]$. Specifically, X is recognizable by some $C \in \mathsf{SIZE}^n[s'(n)]$ which means that X is uniform over $\{x : C(x) = 1\}$. Let C' be the circuit that on input x outputs 1 iff $C(x) = 1 \wedge f(x) \neq x$. Then X_2 is uniform over $\{x : C'(x) = 1\}$, where $C' \in \mathsf{SIZE}^n[s(n) = s'_s(n) + s_t(n) + c \cdot n]$. Thus, X_2 is recognizable by the class $\mathsf{SIZE}^n[s(n)]$. □

3.4 Combining the Results

Non-malleable extractors for ***samplable sources****.* Combining Lemma 3.1 and Theorem 3.3, together with the computational extractor for samplable sources referenced in Theorem 2.16, and the strong two-source non-malleable extractor referenced in Theorem 2.10, we obtain the following:

Theorem 3.5. *If* E *requires exponential size* Σ_4*-circuits, then for any polynomial* $s(n)$ *there exists a construction* $\mathrm{NMExt} : \{0,1\}^n \to \{0,1\}^m$ *of a non-malleable seedless extractor for* n*-bit sources with* $c \cdot n$ *min-entropy (for some constant* $c < 1$*) samplable by size* $s(n)$ *circuits, that is resilient to* $\mathsf{SIZE}[s(n)]$*-tampering with error* $1/s(n)$*. Further, the number of extracted bits is* $m \in \Omega(\frac{n \log\log(n)}{\log(n)})$*, and the extractor runs in time* $s'(n) \in \mathsf{poly}(s(n))$*.*

Non-malleable extractors for ***recognizable sources****.* Combining Lemma 3.2 and Theorem 3.4, together with the computational extractor for recognizable sources referenced in Theorem 2.15, and the strong two-source non-malleable extractor referenced in Theorem 2.10, we obtain the following:

Theorem 3.6. *For any polynomial* $s(n)$ *there exists a construction* $\mathrm{NMExt} : \{0,1\}^n \to \{0,1\}^m$ *of a non-malleable seedless extractor for* n*-bit sources with* $c \cdot n$ *min-entropy (for some constant* $c < 1$*) recognizable by size* $s(n)$ *circuits, that is resilient to* $\mathsf{SIZE}[s(n)]$*-tampering with error* $1/s(n)$*. Further, the number of extracted bits is* $m \in \Omega(\frac{n \log\log(n)}{\log(n)})$ *and the extractor runs in time* $s'(n) \in \mathsf{poly}(s(n))$*.*

4 A Non-Malleable Code for Small Circuit Tampering

Lemma 4.1. *For any polynomial* $s(n)$*, there exists a polynomial* $s'(n) > s(n)$ *such that the following is true. Let* $\ell(n) = O(\log n)$ *be the function from Theorem 1.6 for* $\mathsf{G} : \{0,1\}^{\ell(n)} \to \{0,1\}^n$*. If* $\mathsf{alrssEnc} : \{0,1\}^{k'} \to \{0,1\}^{2n}, \mathsf{alrssDec} : \{0,1\}^{2n} \to \{0,1\}^{k'}$ *is an augmented* α*-leakage-resilient split-state* δ*-non-malleable code with special encoding, computable in time* $o(s(n))$*, and* $\mathsf{G} : \{0,1\}^{\ell(n)} \to \{0,1\}^n$ *is a seed-extending PRG for nondeterministic circuits of size* $O(s(n)^c)$ *such that* $\ell(n) \leq \alpha(n)$ *and* $\delta < (s'(n))^2/32$*, then the construction,* (E, D) *in Figure 4.1 is a* $4/s'(n)$*-alternate-non-malleable code for* k'*-bit messages with codeword length* $O(n)$*, resilient to* $\mathsf{SIZE}[s(n)]$*-tampering with error* $4/s'(n)$*.*

Instantiating the above lemma with the $\mathsf{alrssEnc}$ presented in Section A (which in turn uses the NMC of [ADL18] along with the leakage compiler of [BGW19]), and with G given in Theorem 2.12, and using the fact that a $4/s'(n)$-alternate-non-malleable code for k'-bit messages is a $4/s'(n) + 2^{-k'}$-non-malleable code for k'-bit messages (see Lemma 2.5) we obtain the following corollary:

Theorem 4.2. *If* E *requires exponential size nondeterministic circuits then for any polynomial* $s(n)$*, and for sufficiently large* k*, there exists a* $1/s(n)$*-non-malleable code for* k*-bit messages with codeword length* $O(k^7)$ *that is resilient to* $\mathsf{SIZE}[s(n)]$*-tampering.*

We note that the rate of the code is polynomial but quite large. This rate is inherited from the NMC construction of [ADL18].

Figure 4.1: Non-Malleable Code

Let $(\mathsf{alrssEnc}, \mathsf{alrssDec})$ be an augmented $\alpha(n)$-leakage-resilient δ-split-state non-malleable code with special encoding. Recall that special encoding means that there exists an efficient algorithm $\mathsf{alrssEnc}^*$ that takes a pattern $p := y||*^n$ as input, in addition to the message m, and outputs $\mathsf{alrssEnc}^*(m, p) = (y, X)$ with the property that $(\mathsf{alrssEnc}^*(\cdot, \mathcal{U}), \mathsf{alrssDec})$ is an augmented leakage-resilient split-state non-malleable code.
Let G be a PRG for nondeterministic circuits of size $O(s(n))$.

```
Encoding (E) : On input $m$, do the following
    Sample $s \leftarrow \mathcal{U}_\ell$. Sample $(\mathsf{G}(s), x) \leftarrow \mathsf{alrssEnc}^*(m; p = \mathsf{G}(s)||*^n)$.
    Output $\mathsf{E}(m) = (s, x)$.
Decoding (D) : On input $(\tilde{s}, \tilde{x})$, do the following
    Compute $\tilde{m} = \mathsf{alrssDec}(\mathsf{G}(\tilde{s}), \tilde{x})$.
    Output $\mathsf{D}(\tilde{s}, \tilde{x}) = \tilde{m}$.
```

We now present the proof of Lemma 4.1.

Proof of Lemma 4.1. Let $\epsilon(n) = 4/s'(n)$ (the target error of our non-malleable code). Recall that $1/s'(n)$ is the advantage bound of the PRG, G. And $(\mathsf{alrssEnc}, \mathsf{alrssDec})$ is δ-non-malleable (with additional properties).

For the sake contradiction, assume (E, D) does not satisfy ϵ-alternate-non-malleability: namely, there exists $m_0, m_1 \in \{0,1\}^k$ and tampering function τ of size $s(n)$ such that

$$\mathrm{AltNM}^{\tau,\mathsf{E},\mathsf{D}}_{m_0,m_1}(0) \not\approx_{4/\epsilon} \mathrm{AltNM}^{\tau,\mathsf{E},\mathsf{D}}_{m_0,m_1}(1)$$

As before, we will use this fact (as well as the security of the underlying leakage-resilient augmented-split-state non-malleable code) to break the pseudorandomness guarantee of G by designing a constant-round private coin interactive proof that distinguishes with some non-trivial soundness/completeness gap.

Fix any $\tau : (s, x) \mapsto (\tilde{s}, \tilde{x})$ in $\mathsf{SIZE}^{\Sigma_k}[s(n)]$. Define f to denote the function that computes $(s, x) \mapsto \tilde{x}$ according to τ, and g to denote the function that computes $(s, x) \mapsto \tilde{s}$ according to τ. In other words, $\tau(s, x) = (g(s, x), f(s, x))$.

Figure 4.2: Interactive Proof for distinguishing G from uniformly random bits

Recall that $(\mathsf{alrssEnc}, \mathsf{alrssDec})$ is an augmented leakage-resilient split-state non-malleable code with special encoding, $\mathsf{alrssEnc}^*$. Define $\mathsf{alrssEnc}^*_R$ to be the $\mathsf{alrssEnc}^*$ that just outputs the right state, i.e. if $\mathsf{alrssEnc}^*(m, p = y||*^n; r) \mapsto (y, x)$ then $\mathsf{alrssEnc}^*_R : (m, p = y||*^n; r) \mapsto x$.
Recall that G is a PRG for nondeterministic circuits of size $O(s(n))$. Finally, recall that f, g correspond to the tampering attack.
Our protocol aims to accept strings from $\mathcal{U}_\ell, G(\mathcal{U}_\ell)$ when Merlin plays according to below (completeness) and reject strings from $\mathcal{U}_{\ell+n}$ regardless of the strategy Merlin utilizes (soundness). Because we can amplify by repetition, it suffices for there to be small gap between the two.
Hardcoded into Arthur as non-uniform advice are f, g and m_0, m_1.

On input s, y:

Arthur Sample coin $b \leftarrow \mathcal{U}$. Sample encoding $(y, x) \leftarrow \mathsf{alrssEnc}^*(m_b, p = y||*^n)$. Send Merlin $\tilde{s} = g(s, x)$.
Merlin If $(s, y) = \mathsf{G}(s)$, respond $\tilde{y}$ such that $(\tilde{s}, \tilde{y}) = \mathsf{G}(\tilde{s})$. Otherwise, respond arbitrary $\tilde{y}$.
Arthur Set $z' = \mathsf{alrssDec}(\tilde{y}, \tilde{x})$ where $\tilde{x} = f(s, x)$. If $z' \in \{m_0, m_1\}$, set $z = \texttt{same}$. Otherwise, set $z = z'$. Send z to merlin.
Merlin (Guess Arthur's bit.) If

$$\Pr[\mathsf{alrssDec}(\tilde{y}, f(s, \mathsf{alrssEnc}^*_R(m_0, y))) = z | g(s, \mathsf{alrssEnc}^*_R(m_0, y)) = \tilde{s}]$$

is upper bounded by

$$\Pr[\mathsf{alrssDec}(\tilde{y}, f(s, \mathsf{alrssEnc}^*_R(m_1, y))) = z | g(s, \mathsf{alrssEnc}^*_R(m_1, y)) = \tilde{s}]$$

set $b' = 1$. Otherwise, set $b' = 0$. Respond b'.
Arthur Accept if $b = b'$, and reject otherwise.

Claim 4.1. There exists a set Π_Y such that

1. Π_Y is noticeably dense in G: $\Pr_{s \overset{u}{\leftarrow} \{0,1\}^\ell}[\mathsf{G}(s) \in \Pi_Y] \geq \epsilon/2$
2. Arthur accepts inputs in Π_Y with probability $> \frac{1+\epsilon/2}{2}$ when playing with (honest) Merlin (as prescribed in Figure 4.2).

Proof. If the protocol in Figure 4.2 is given inputs from $\mathsf{G}(S) = (S, \mathsf{G}'(S))$ (where $S \equiv \mathcal{U}_\ell$), then upon the choice of $b = 1$, Arthur's final message is exactly that of the alternate-non-malleability game :

$$z \sim \mathrm{AltNM}^{\tau}_{m_0,m_1}(1).$$

Similarly, if $b = 0$, Arthur's final message is sampled according to:

$$(z, \tilde{z}) \sim \mathrm{AltNM}^{\tau}_{m_0,m_1}(0).$$

By our assumption, these two distributions are ϵ-far from each other.

By Proposition 4, this implies there exists a set Π_Y such that for any $(s, y) \in \Pi_Y$ these distributions are $\epsilon/2$-far, and moreover $\Pr[\mathsf{G}(S) \in \Pi_Y] \geq \epsilon/2$.

By Proposition 2.6, for any $(s, y) \in \Pi_Y$ Merlin guesses b correctly and Arthur accepts with probability $\geq \frac{1+\epsilon/2}{2}$. □

Claim 4.2. There exists a set Π_N such that

1. Π_N is large: $\Pr_{(s,y)\overset{u}{\leftarrow}\{0,1\}^{\ell+n}}[(s, y) \in \Pi_N] \geq 1 - 8\delta/\epsilon$
2. Arthur accepts inputs in Π_N with probability $\leq \frac{1+\epsilon/4}{2}$ when playing with any (cheating) Merlin (as prescribed in Figure 4.2).

Proof. Soundness follows from first observing that any Merlin strategy corresponds to some α-leaky split-state tampering on the augmented-leakage resilient split state-code. We conclude soundness because Merlin's view is that of the alternate leakage-resilient augmented-split-state game. As with we the case of the non-malleable extractor (see argument in the proof of Claim 3.3), we use the existence the optimality of some optimal strategy M^* (who, for any input (s, y), chooses messages to maximize the distance of his view when Arthur chooses $b = 0$ versus his view when Arthur chooses $b = 1$) to apply the Markov argument to a single distribution.

Fix an optimal Merlin strategy M^* as described above and assume s, y are uniformly distributed. We make some observations about the protocol in this case:

1. **Well-formed augmented leakage-resilient split-state encodings.**
   Uniform $y \sim \mathcal{U}$ means our leakage-resilient augmented-split-state codewords are properly distributed, namely for $b = 0, 1$ it is the case that $\mathsf{alrssEnc}^*(m_b, p = \mathcal{U}||*^n) \equiv \mathsf{alrssEnc}(m_b)$. Moreover, s is independent of the split-state codeword (x, y) sampled by Arthur at the beginning.
2. **ℓ-leaky split-state tampering.**
   Arthur's first message to Merlin, corresponding to the random variable $\tilde{s} = g(s, x)$, can be viewed as ℓ-bits of leakage from the right codeword state (to the left tampering function).
   Thus, we have $\tilde{x} = f(s, x)$ and $\tilde{y} = M^*(s, y, g(s, x))$ which for any fixed choice of s is an ℓ-leaky split-state tampering, Π^s. Thus when s is random, Π^s is a distribution over ℓ-leaky split-state tampering functions.
3. **Merlin's view is identical to augmented alternate-non-malleable game.**
   Recall that Merlin's view corresponds to the variables $(s, y, \tilde{s}, z) = \mathsf{View}^{M^*}(b)$, where b is Arthur's initial coin. Observe that $(y, \tilde{s}, z)$ is sampled identically to $\mathrm{AltANM}^{\Pi^s,\mathsf{alrssEnc},\mathsf{alrssDec}}(b)$, where b is Arthur's initial coin toss. And s is independent of the initial encoding in the AltANM game, which has worst case guarantees that apply to Π^s for any choice of s.

Putting these observations together, we have by that, because $(\mathsf{alrssEnc}, \mathsf{alrssDec})$ is an ℓ-leakage-resilient δ-augmented-split-state non-malleable code, and since, by Lemma A.9, this implies that it is also a 2δ-augmented-split-state alternate non-malleable code,

$$\mathsf{View}^{M^*}(0) \approx_{2\delta} \mathsf{View}^{M^*}(1).$$

Observe that if there existed a strategy M' and input (s, y) such that the distance between the view of M' on $b = 0$ vs $b = 1$ was greater than that of M^*, this would contradict the optimality of M^*. Thus, by Proposition 3 there exists a set, Π_N such that $\Pr_{(s,y)\overset{u}{\leftarrow}\{0,1\}^{\ell+n}}[(s, y) \in \Pi_N] \geq 1 - 8\delta/\epsilon$ and for each $(s, y) \in \Pi_N$ and any Merlin strategy M', the view when $b = 0$ is $\epsilon/4$-far from the view when $b = 1$. Thus, by Proposition 2.6, this means for any $(s, y) \in \Pi_N$, any Merlin strategy outputs b' such that $b' = b$ with probability at most $\frac{1+\epsilon/4}{2}$. □

We conclude from Claim 4.1 and Claim 4.2, that there is a constant round IP protocol where Arthur can be represented by circuit of size $O(s(n))$ that recognizes $\Pi = (\Pi_Y, \Pi_N)$ with completeness/soundness gap $\epsilon/2$. By Lemma 2.21, this implies the existence of an $s'(n)$-size nondeterministic circuit,$\mathcal{C}$, that decides the promise problem, Π. Because Π_Y is $\epsilon/2$-dense under G (i.e. $\Pr_s[\mathsf{G}(s) \in \Pi_Y]$) and Π_N is $1 - 8\delta/\epsilon$ dense under the uniform distribution (i.e. $\Pr_z[z \in \Pi_Y] \leq 4\delta/\epsilon$). The nondeterministic circuit $\mathcal{C}$ can distinguish with advantage $|\epsilon/2 - 8\delta/\epsilon| \geq \epsilon/4 = 1/s'(n)$. So, our initial assumption must be false. □

References

AASY16. Benny Applebaum, Sergei Artemenko, Ronen Shaltiel, and Guang Yang. Incompressible functions, relative-error extractors, and the power of nondeterministic reductions. *Comput. Complex.*, 25(2):349–418, 2016.

ADKO15. Divesh Aggarwal, Yevgeniy Dodis, Tomasz Kazana, and Maciej Obremski. Non-malleable reductions and applications. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th Annual ACM Symposium on Theory of Computing*, pages 459–468, Portland, OR, USA, June 14–17, 2015. ACM Press.

Adl78. Leonard M. Adleman. Two theorems on random polynomial time. In *FOCS*, pages 75–83. IEEE Computer Society, 1978.

ADL18. Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. *SIAM J. Comput.*, 47(2):524–546, 2018.

ADN$^+$19. Divesh Aggarwal, Ivan Damgård, Jesper Buus Nielsen, Maciej Obremski, Erick Purwanto, João Ribeiro, and Mark Simkin. Stronger leakage-resilient and non-malleable secret sharing schemes for general access structures. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 510–539, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

AGM$^+$15a. Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Explicit non-malleable codes against bit-wise tampering and permutations. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 538–557, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

AGM$^+$15b. Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. A rate-optimizing compiler for non-malleable codes against bit-wise tampering and permutations. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part I*, volume 9014 of *Lecture Notes in Computer Science*, pages 375–397, Warsaw, Poland, March 23–25, 2015. Springer, Heidelberg, Germany.

AO20. Divesh Aggarwal and Maciej Obremski. A constant rate non-malleable code in the split-state model. In *61st Annual Symposium on Foundations of Computer Science*, pages 1285–1294, Durham, NC, USA, November 16–19, 2020. IEEE Computer Society Press.

Bab85. László Babai. Trading group theory for randomness. In *17th Annual ACM Symposium on Theory of Computing*, pages 421–429, Providence, RI, USA, May 6–8, 1985. ACM Press.

Bal21. Marshall Ball. *On Resilience to Computable Tampering*. PhD thesis, Columbia University, 2021.

BCL$^+$20. Marshall Ball, Eshan Chattopadhyay, Jyun-Jie Liao, Tal Malkin, and Li-Yang Tan. Non-malleability against polynomial tampering. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 97–126, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Heidelberg, Germany.

BDG$^+$18. Marshall Ball, Dana Dachman-Soled, Siyao Guo, Tal Malkin, and Li-Yang Tan. Non-malleable codes for small-depth circuits. In Mikkel Thorup, editor, *59th Annual Symposium on Foundations of Computer Science*, pages 826–837, Paris, France, October 7–9, 2018. IEEE Computer Society Press.

BDK$^+$11. Boaz Barak, Yevgeniy Dodis, Hugo Krawczyk, Olivier Pereira, Krzysztof Pietrzak, François-Xavier Standaert, and Yu Yu. Leftover hash lemma, revisited. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 1–20, Santa Barbara, CA, USA, August 14–18, 2011. Springer, Heidelberg, Germany.

BDK$^+$19. Marshall Ball, Dana Dachman-Soled, Mukul Kulkarni, Huijia Lin, and Tal Malkin. Non-malleable codes against bounded polynomial time tampering. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 501–530, Darmstadt, Germany, May 19–23, 2019. Springer, Heidelberg, Germany.

BDKM16. Marshall Ball, Dana Dachman-Soled, Mukul Kulkarni, and Tal Malkin. Non-malleable codes for bounded depth, bounded fan-in circuits. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 881–908, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.

BDKM18. Marshall Ball, Dana Dachman-Soled, Mukul Kulkarni, and Tal Malkin. Non-malleable codes from average-case hardness: AC^0, decision trees, and streaming space-bounded tampering. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 618–650, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany.

BDKM20. Marshall Ball, Dana Dachman-Soled, Mukul Kulkarni, and Tal Malkin. Limits to non-malleability. In Thomas Vidick, editor, *ITCS 2020: 11th Innovations in Theoretical Computer Science Conference*, volume 151, pages 80:1–80:32, Seattle, WA, USA, January 12–14, 2020. LIPIcs.

Bei11. Amos Beimel. Secret-sharing schemes: A survey. In *IWCC*, volume 6639 of *Lecture Notes in Computer Science*, pages 11–46. Springer, 2011.

BGP00. Mihir Bellare, Oded Goldreich, and Erez Petrank. Uniform generation of np-witnesses using an np-oracle. *Inf. Comput.*, 163(2):510–526, 2000.

BGW19. Marshall Ball, Siyao Guo, and Daniel Wichs. Non-malleable codes for decision trees. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 413–434, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

BKP18. Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. Multi-collision resistance: a paradigm for keyless hash functions. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th Annual ACM Symposium on Theory of Computing*, pages 671–684, Los Angeles, CA, USA, June 25–29, 2018. ACM Press.

BM88. László Babai and Shlomo Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988.

BOV03. Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 299–315, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Heidelberg, Germany.

BS19. Saikrishna Badrinarayanan and Akshayaram Srinivasan. Revisiting non-malleable secret sharing. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Part I*, volume 11476 of *Lecture Notes in Computer Science*, pages 593–622, Darmstadt, Germany, May 19–23, 2019. Springer, Heidelberg, Germany.

CCHM19. Binyi Chen, Yilei Chen, Kristina Hostáková, and Pratyay Mukherjee. Continuous space-bounded non-malleable codes from stronger proofs-of-space. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 467–495, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

CG14a. Mahdi Cheraghchi and Venkatesan Guruswami. Capacity of non-malleable codes. In Moni Naor, editor, *ITCS 2014: 5th Conference on Innovations in Theoretical Computer Science*, pages 155–168, Princeton, NJ, USA, January 12–14, 2014. Association for Computing Machinery.

CG14b. Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 440–464, San Diego, CA, USA, February 24–26, 2014. Springer, Heidelberg, Germany.

CGL16. Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In Daniel Wichs and Yishay Mansour, editors, *48th Annual ACM Symposium on Theory of Computing*, pages 285–298, Cambridge, MA, USA, June 18–21, 2016. ACM Press.

CL17. Eshan Chattopadhyay and Xin Li. Non-malleable codes and extractors for small-depth circuits, and affine functions. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th Annual ACM Symposium on Theory of Computing*, pages 1171–1184, Montreal, QC, Canada, June 19–23, 2017. ACM Press.

CSW06. Mahdi Cheraghchi, Amin Shokrollahi, and Avi Wigderson. Computational hardness and explicit constructions of error correcting codes. Technical report, 2006.

DKO13. Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Non-malleable codes from two-source extractors. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 239–257, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.

DKP20. Dana Dachman-Soled, Ilan Komargodski, and Rafael Pass. Non-malleable codes for bounded polynomial depth tampering. Cryptology ePrint Archive, Report 2020/776, 2020. https://eprint.iacr.org/2020/776.

DLSZ15. Dana Dachman-Soled, Feng-Hao Liu, Elaine Shi, and Hong-Sheng Zhou. Locally decodable and updatable non-malleable codes and their applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part I*, volume 9014 of *Lecture Notes in Computer Science*, pages 427–450, Warsaw, Poland, March 23–25, 2015. Springer, Heidelberg, Germany.

DPW10. Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In Andrew Chi-Chih Yao, editor, *ICS 2010: 1st Innovations in Computer Science*, pages 434–452, Tsinghua University, Beijing, China, January 5–7, 2010. Tsinghua University Press.

DPW18. Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. *J. ACM*, 65(4):20:1–20:32, 2018.

Dru13. Andrew Drucker. Nondeterministic direct product reductions and the success probability of SAT solvers. In *54th Annual Symposium on Foundations of Computer Science*, pages 736–745, Berkeley, CA, USA, October 26–29, 2013. IEEE Computer Society Press.

DSKP21. Dana Dachman-Soled, Ilan Komargodski, and Rafael Pass. Non-malleable codes for bounded parallel-time tampering. In *Annual International Cryptology Conference*, pages 535–565. Springer, Cham, 2021.

DY13. Yevgeniy Dodis and Yu Yu. Overcoming weak expectations. In Amit Sahai, editor, *TCC 2013: 10th Theory of Cryptography Conference*, volume 7785 of *Lecture Notes in Computer Science*, pages 1–22, Tokyo, Japan, March 3–6, 2013. Springer, Heidelberg, Germany.

FHMV17. Sebastian Faust, Kristina Hostáková, Pratyay Mukherjee, and Daniele Venturi. Non-malleable codes for space-bounded tampering. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 95–126, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany.

FL97. Uriel Feige and Carsten Lund. On the hardness of computing the permanent of random matrices. *Comput. Complex.*, 6(2):101–132, 1997.

FMNV14. Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. Continuous non-malleable codes. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 465–488, San Diego, CA, USA, February 24–26, 2014. Springer, Heidelberg, Germany.

FMVW14. Sebastian Faust, Pratyay Mukherjee, Daniele Venturi, and Daniel Wichs. Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 111–128, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany.

GK18a. Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th Annual ACM Symposium on Theory of Computing*, pages 685–698, Los Angeles, CA, USA, June 25–29, 2018. ACM Press.

GK18b. Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing for general access structures. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 501–530, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany.

GMW91. Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.

GS86. Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *18th Annual ACM Symposium on Theory of Computing*, pages 59–68, Berkeley, CA, USA, May 28–30, 1986. ACM Press.

GST03. Dan Gutfreund, Ronen Shaltiel, and Amnon Ta-Shma. Uniform hardness versus randomness tradeoffs for arthur-merlin games. *Comput. Complex.*, 12(3-4):85–130, 2003.

GSZ21. Vipul Goyal, Akshayaram Srinivasan, and Chenzhi Zhu. Multi-source non-malleable extractors and applications. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II*, volume 12697 of *Lecture Notes in Computer Science*, pages 468–497. Springer, 2021.

GW02. Oded Goldreich and Avi Wigderson. Derandomization that is rarely wrong from short advice that is typically good. In *RANDOM*, volume 2483 of *Lecture Notes in Computer Science*, pages 209–223. Springer, 2002.

IW97. Russell Impagliazzo and Avi Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *29th Annual ACM Symposium on Theory of Computing*, pages 220–229, El Paso, TX, USA, May 4–6, 1997. ACM Press.

JVV86. Mark Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, 43:169–188, 1986.

KOS17. Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, and Sruthi Sekar. Four-state non-malleable codes with explicit constant rate. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part II*, volume 10678 of *Lecture Notes in Computer Science*, pages 344–375, Baltimore, MD, USA, November 12–15, 2017. Springer, Heidelberg, Germany.

KOS18. Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, and Sruthi Sekar. Non-malleable randomness encoders and their applications. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 589–617, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany.

KvM02. Adam R. Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31(5):1501–1526, 2002.

KvMS12. Jeff Kinne, Dieter van Melkebeek, and Ronen Shaltiel. Pseudorandom generators, typically-correct derandomization, and circuit lower bounds. *Comput. Complex.*, 21(1):3–61, 2012.

LCG$^+$19a. Fuchun Lin, Mahdi Cheraghchi, Venkatesan Guruswami, Reihaneh Safavi-Naini, and Huaxiong Wang. Non-malleable secret sharing against affine tampering. *CoRR*, abs/1902.06195, 2019.

LCG$^+$19b. Fuchun Lin, Mahdi Cheraghchi, Venkatesan Guruswami, Reihaneh Safavi-Naini, and Huaxiong Wang. Non-malleable secret sharing against affine tampering. *CoRR*, abs/1902.06195, 2019.

Lev86. Leonid A. Levin. Average case complete problems. *SIAM J. Comput.*, 15(1):285–286, 1986.

Li16a. Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. *Electron. Colloquium Comput. Complex.*, 23:115, 2016.

Li16b. Xin Li. Improved two-source extractors, and affine extractors for polylogarithmic entropy. In Irit Dinur, editor, *57th Annual Symposium on Foundations of Computer Science*, pages 168–177, New Brunswick, NJ, USA, October 9–11, 2016. IEEE Computer Society Press.

Li17. Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th Annual ACM Symposium on Theory of Computing*, pages 1144–1156, Montreal, QC, Canada, June 19–23, 2017. ACM Press.

Li18. Xin Li. Non-malleable extractors and non-malleable codes: Partially optimal constructions. Cryptology ePrint Archive, Report 2018/353, 2018. https://eprint.iacr.org/2018/353.

Li19. Xin Li. Non-malleable extractors and non-malleable codes: Partially optimal constructions. In *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, pages 28:1–28:49, 2019.

LL12. Feng-Hao Liu and Anna Lysyanskaya. Tamper and leakage resilience in the split-state model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 517–532, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Heidelberg, Germany.

LZ19. Fu Li and David Zuckerman. Improved extractors for recognizable and algebraic sources. In *APPROX-RANDOM*, volume 145 of *LIPIcs*, pages 72:1–72:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

Mic94. Silvio Micali. CS proofs (extended abstracts). In *35th Annual Symposium on Foundations of Computer Science*, pages 436–453, Santa Fe, NM, USA, November 20–22, 1994. IEEE Computer Society Press.

MV05. Peter Bro Miltersen and N. V. Vinodchandran. Derandomizing arthur-merlin games using hitting sets. *Comput. Complex.*, 14(3):256–279, 2005.

Sha11. Ronen Shaltiel. Weak derandomization of weak algorithms: Explicit versions of yao's lemma. *Comput. Complex.*, 20(1):87–143, 2011.

Sha21. Personal communication with r. shaltiel, 2021.

SU05. Ronen Shaltiel and Christopher Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *J. ACM*, 52(2):172–216, 2005.

SU06. Ronen Shaltiel and Christopher Umans. Pseudorandomness for approximate counting and sampling. *Comput. Complex.*, 15(4):298–341, 2006.

SU09. Ronen Shaltiel and Christopher Umans. Low-end uniform hardness versus randomness tradeoffs for AM. *SIAM J. Comput.*, 39(3):1006–1037, 2009.

SV10. Ronen Shaltiel and Emanuele Viola. Hardness amplification proofs require majority. *SIAM J. Comput.*, 39(7):3122–3154, 2010.

TV00. Luca Trevisan and Salil P. Vadhan. Extracting randomness from samplable distributions. In *41st Annual Symposium on Foundations of Computer Science*, pages 32–42, Redondo Beach, CA, USA, November 12–14, 2000. IEEE Computer Society Press.

Vad99. Salil Pravin Vadhan. *A study of statistical zero-knowledge proofs*. PhD thesis, Massachusetts Institute of Technology, 1999.

Vio06. Emanuele Viola. The complexity of hardness amplification and derandomization, 2006.

A Augmented Leakage-Resilient Split-State and Special Encoding

Ball, Guo, and Wichs [BGW19] presented a construction of a "one-time information-theoretic leakage-resilient encryption" scheme (see Definition A.1) and showed that it could be composed with a split-state non-malleable code to achieve a non-malleable code against "leaky split-state tampering." In this section we observe that the leakage-resilient compiler of Ball, Guo, and Wichs [BGW19] discussed above preserves the following two two properties of the underlying split-state non-malleable code that are of interest to us:

- **Augmented split-state non-malleability:** there exists a simulator which can simulate the joint distribution of the left (or right) codeword states in addition to the outcome of non-malleability experiment. Because in the leaky split-state tampering setting the tampered left (or right) codeword additionally depends on the leakage, we require the simulator to output this as well.
- **Special encoding:** there exists a special encoding procedure that given a desired left (or right) codeword state and message, outputs a valid encoding of the message. Importantly, if the special encoder is given uniform left codeword states, its output is identically distributed to real encodings of the message.

In the following, we denote by $(c_{\tilde{L}}, c_{\tilde{R}}, \mathsf{trans}) \leftarrow \langle f(c_L), g(c_R) \rangle$ an interactive protocol where the left side gets input c_L, the right side gets input c_R, the left and right side communicate by generating a transcript trans of bounded length. Finally, the left side outputs $c_{\tilde{L}}$, the right side outputs $c_{\tilde{R}}$. Formally, the properties are defined as follows:

Definition A.1 ($(2n, k, \epsilon)$-Augmented Non-Malleable Code Resilient to Leaky Spit-State Tampering)**.** *Let* $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^{2n}$ *and* $\mathsf{Dec} : \{0,1\}^{2n} \to \{0,1\}^k \cup \{\perp\}$ *be a coding scheme. For any pair of α-leaky split state tampering functions (f, g) (each acting on n bits of the codeword) and $m \in \{0,1\}^k$, define the augmented tampering experiment*

$$\mathrm{ANM}_m^{f,g,\mathsf{Enc},\mathsf{Dec}} := \left\{ \begin{array}{c} (c_L, c_R) \leftarrow \mathsf{Enc}(m), (c_{\tilde{L}}, c_{\tilde{R}}, \mathsf{trans}) \leftarrow \langle f(c_L), g(c_R) \rangle, \tilde{m} = \mathsf{Dec}(c_{\tilde{L}}, c_{\tilde{R}}) \\ \textit{Output: } (c_L, \mathsf{trans}, \tilde{m}), \end{array} \right\}$$

where trans *has length $\alpha \cdot n$. We say that* $(\mathsf{Enc}, \mathsf{Dec})$ *is a* $(2n, k, \epsilon)$-augmented non-malleable code resilient to α-leaky split state tampering *if there exists a simulator* $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ *s.t. for all f, g as above and all $m \in \{0,1\}^k$,* $(\mathsf{Sim}_1(f,g), \mathrm{Copy}(\mathsf{Sim}_2(f,g), m)) \approx_\epsilon \mathrm{ANM}_m^{f,g,\mathsf{Enc},\mathsf{Dec}}$.

We observe that the definition and equivalence of alternative-non-malleability (Definition 2.4 and Lemma 2.5) extends to augmented non-malleability.

Definition A.2 (Special Encoding)**.** *Let* $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^{2n}$ *and* $\mathsf{Dec} : \{0,1\}^{2n} \to \{0,1\}^k \cup \{\perp\}$ *be a $(2n, k, \epsilon)$-non-malleable code. We say that* $(\mathsf{Enc}, \mathsf{Dec})$ has a special encoding *if there is an efficient algorithm* Enc^* *that takes as input a message m and a pattern string $p \in \{0,1,*\}^{2n}$. It outputs an encoding x of m such that x matches p in all positions not set to $*$.*

A.1 Leakage-resilient split-state compiler of [BGW19]

Ball, Guo, and Wichs [BGW19] introduced a novel notion of one-time information-theoretic leakage-resilient encryption. We recall their definition and construction next:

Definition A.3 (Leakage-Resilient Encryption [BGW19])**.** *Consider a (randomized) encryption scheme* $(\mathsf{Encrypt}, \mathsf{Decrypt})$ *which encrypts message x of length $|x| = k$ using a* key *of size* $|\mathsf{key}| = m$*. For some message $x \in \{0,1\}^k$ consider the following randomized experiment* $\mathrm{Game}^{\mathsf{LRENC}}(x)$*:*

- *Choose* $\mathsf{key} \leftarrow \{0,1\}^m$, $\mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{key}, x)$.
- *Alice gets* ct *and Bob gets* key*. They can run an arbitrary protocol with each other subject to the total communication being at most ℓ_1 bits. Let* $\mathsf{trans} \in \{0,1\}^{\ell_1}$ *be the transcript.*
- *At the end of the protocol, Alice also outputs an additional value* $\mathsf{aux} \in \{0,1\}^{\ell_2}$.
- *The output of the game is* $\mathsf{key}, \mathsf{trans}, \mathsf{aux}$.

We say that an encryption scheme is $(\ell_1, \ell_2, \varepsilon)$-leakage-resilient if for any adversarial strategy of Alice and Bob and for any x_0, x_1 the outputs of $\mathrm{Game}^{\mathsf{LRENC}}(x_0)$ *and* $\mathrm{Game}^{\mathsf{LRENC}}(x_1)$ *have statistical distance at most ε.*

Let $\mathsf{Ext} : \{0,1\}^r \times \{0,1\}^d \to \{0,1\}^k$ be a strong seeded extractor and let $2\mathsf{Ext} : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}^d$ be a strong two-source extractor. Define the scheme $(\mathsf{Encrypt}, \mathsf{Decrypt})$ as follows:

- $\mathsf{Encrypt}(\mathsf{key}, x)$: Choose $u \leftarrow \{0,1\}^r, y \leftarrow \{0,1\}^m, s = 2\mathsf{Ext}(\mathsf{key}, y), z = \mathsf{Ext}(u; s) \oplus x$. Output $\mathsf{ct} = (u, y, z)$.

- $\mathsf{Decrypt}(\mathsf{key}, \mathsf{ct} = (u, y, z))$ to compute $s = \mathsf{2Ext}(\mathsf{key}, y)$ and output $z \oplus \mathsf{Ext}(u; s)$.

Lemma A.4 ([BGW19])**.** *For any $\alpha \in (0, 1/4)$, $\epsilon \in \exp(-\Omega(k))$ there exist explicit extractors such that the following holds. The scheme* $(\mathsf{Encrypt}, \mathsf{Decrypt})$ *is $(\alpha \cdot n, m, \epsilon/2)$-leakage-resilient, where $n = r + 2m + k$ and $n \in \Theta(k)$.*

Leakage-resilient split-state non-malleability. Let $\mathcal{E} = (\mathsf{Encrypt}, \mathsf{Decrypt})$ be a leakage-resilient encryption with message size k and key length m. Ball, Guo, and Wich's encoding $(\mathsf{Enc}, \mathsf{Dec})$ is below:

- $\mathsf{Enc}(x_L, x_R)$: Sample $\mathsf{key}_L \leftarrow \{0,1\}^m, \mathsf{ct}_L \leftarrow \mathsf{Encrypt}(x_L)$, $\mathsf{key}_R \leftarrow \{0,1\}^m, \mathsf{ct}_R \leftarrow \mathsf{Encrypt}(x_R)$. Output (c_L, c_R) where $c_L = (\mathsf{ct}_L, \mathsf{key}_R), c_R = (\mathsf{ct}_R, \mathsf{key}_L)$.
- $\mathsf{Dec}(c_L, c_R)$ Parse $c_L = (\mathsf{ct}_L, \mathsf{key}_R), c_R = (\mathsf{ct}_R, \mathsf{key}_L)$ and output $x_L = \mathsf{Decrypt}(\mathsf{key}_L, \mathsf{ct}_L)$ and $x_R = \mathsf{Decrypt}(\mathsf{key}_R, \mathsf{ct}_R)$.

We can immediately observe that this reduction preserves special encoding when the leakage-resilient encryption scheme above is used.

Proposition 5. *For any constant $\alpha \in (0, 1/4)$ there exists a setting of parameters such that the following holds.*

If $(\mathsf{E}', \mathsf{D}')$ *is an efficient split-state non-malleable code with special encoding for the pattern string p that is set to any constant on the left hand side and set to $*$ on the right hand side and each side is of length k, then* $(\mathsf{Enc} \circ \mathsf{E}', \mathsf{D}' \circ \mathsf{Dec})$ *is an efficient α-leakage-resilient split-state non-malleable code with special encoding for the pattern string p that is set to any constant on the left hand side and set to $*$ on the right hand side and each side is of length n, when* $(\mathsf{Enc}, \mathsf{Dec})$ *is instantiated with the leakage-resilient encryption* $(\mathsf{Encrypt}, \mathsf{Decrypt})$.

Let E^* denote the special encoder for $(\mathsf{E}', \mathsf{D}')$. Then, we can define a special encoder Enc^* for the composed code as follows. On input $p = (u_L, y_L, z_L, \mathsf{key}_R)||*^n, x$, where $m \in \{0,1\}^{k'}$ is the message and $(u_L, y_L, z_L, \mathsf{key}_R) \in \{0,1\}^n$, where $n = r + 2m + k$ is the desired left codeword state, do the following:

1. Sample key_L uniformly at random.
2. Compute $s_L = \mathsf{2Ext}(\mathsf{key}_L, y_L)$ and set $x_L = \mathsf{Ext}(u_L; s_L) \oplus z_L$.
3. Invoke $\mathsf{E}^*(p = x_L||*^k, m)$ to get the split-state encoding of m, (x_L, x_R).
4. Sample $u_R, y_R \in \{0,1\}^{r+m}$ uniformly at random.
5. Compute $s_R = \mathsf{2Ext}(\mathsf{key}_R, y_R)$ and set $z_R = \mathsf{Ext}(u_R; s_R) \oplus x_R$.
6. Output $(u_L, y_L, z_L, \mathsf{key}_R), (u_R, y_R, z_R, \mathsf{key}_L)$

Notice that if $(u_L, y_L, z_L, \mathsf{key}_R)$ is uniform, the output of the above is identically distributed to $\mathsf{Enc} \circ \mathsf{E}'(m)$.

A.2 Augmenting the construction of [BGW19]

Here we sketch how to prove that the construction above preserves augmented non-malleability. In particular, we show how to extend the analysis of [BGW19].

Lemma A.5. *For any $\alpha \in (0, 1/4)$, if* $(\mathsf{E}', \mathsf{D}')$ *is an efficient ϵ-augmented-split-state non-malleable code, then* $(\mathsf{Enc} \circ \mathsf{E}', \mathsf{D}' \circ \mathsf{Dec})$ *is an efficient α-leakage-resilient 2ϵ-augmented split-state non-malleable code, when* $(\mathsf{Enc}, \mathsf{Dec})$ *is instantiated with the leakage-resilient encryption* $(\mathsf{Encrypt}, \mathsf{Decrypt})$*, with an appropriate setting of parameters that achieves $(\alpha \cdot n, m, \epsilon/2)$-leakage-resilience, and $n = r + 2m + k$.*

Sketch. We adapt the proof strategy from [BGW19]. Fix some (possibly interactive) split-state tampering functions f, g, some message m, and consider an outcome $(c_L, \mathsf{trans}, \tilde{m})$ of the experiment $\mathrm{ANM}_m^{f,g,\mathsf{Enc}\circ\mathsf{E}',\mathsf{D}'\circ\mathsf{Dec}}$. We show a sequence of hybrids which lead from $\mathrm{ANM}_m^{f,g,\mathsf{Enc}\circ\mathsf{E}',\mathsf{D}'\circ\mathsf{Dec}}$ to a game in which an interactive tampering can be reduced to a non-interactive, split-state one.

More formally, in the following, recall that codewords outputted by Enc have the form $c_L = (\mathsf{key}_R, \mathsf{ct}_L)$ and $c_R = (\mathsf{key}_L, \mathsf{ct}_R)$. We consider the distribution of $z = (\mathsf{key}_L, \mathsf{key}_R, \mathsf{trans}, \tilde{\mathsf{key}}_L, \tilde{\mathsf{key}}_R)$ in Hybrids H_0, H_1, H_2 and show that it is indistinguishable in each consecutive pair of hybrids.

Hybrid H_0: This is the original game $\mathrm{ANM}_m^{f,g,\mathsf{Enc}\circ\mathsf{E}',\mathsf{D}'\circ\mathsf{Dec}}$.

Hybrid H_1: In this game, we form x as first running $x \leftarrow \mathsf{E}'(m)$ and then setting the first k bits of x to zero. For readability, we refer to the resulting string as x_0. We show that H_1 is statistically ϵ-close to H_0. This follows directly from the ϵ-leakage resilience property of $(\mathsf{Enc}, \mathsf{Dec})$. We now show that the resulting distributions of z in these two hybrids (with x and x_0 fixed) can be seen as being induced by corresponding variables in games $\mathrm{Game}^{\mathsf{LRENC}}(x)$ or $\mathrm{Game}^{\mathsf{LRENC}}(x_0)$, respectively.

To this end, suppose that in game $\mathrm{Game}^{\mathsf{LRENC}}(x)$ or $\mathrm{Game}^{\mathsf{LRENC}}(x_0)$, respectively, the game correctly samples key_L, c_L and hands Alice $c_L = \mathsf{Enc}(x)$ (or $\mathsf{Enc}(x_0)$, respectively), and Bob key_L. Alice and Bob can now pick now pick (as part of their strategy in these games), values key_R, c_R according to any worst case distribution, and have Alice output $\mathsf{aux} = \widetilde{\mathsf{key}}_R$ (the tampered outcome in the output location of key_R) at the end of the game. This is a valid output of Alice, since $|\widetilde{\mathsf{key}}_R| = m$. By the $\epsilon/2$-leakage resilience property of $(\mathsf{Enc}, \mathsf{Dec})$, we have that the outputs of $\mathrm{Game}^{\mathsf{LRENC}}(x)$ or $\mathrm{Game}^{\mathsf{LRENC}}(x_0)$ are statistically $\epsilon/2$-close for any x, x_0 as above. This implies that the distribution of the tuple $(\mathsf{key}_L, \mathsf{trans}, \widetilde{\mathsf{key}}_R)$ is $\epsilon/2$-close in the above two experiments. Note that $\widetilde{\mathsf{key}}_L$ is fully determined by the outcomes of c_R, key_L, and trans, and that key_R, c_R are fixed to the worst case choice. Hence, for worst-case choice of key_R, c_R, the distribution of the tuple $z = (\mathsf{key}_L, \mathsf{key}_R, \mathsf{trans}, \widetilde{\mathsf{key}}_L, \widetilde{\mathsf{key}}_R)$ is also $\epsilon/2$-close in the above two experiments. Reintroducing the randomness over the sampling of these values yields the claim.

Hybrid H_2: In this game, we encode 0^{2k} rather than m_0. By a symmetric argument as above, H_1 is statistically $\epsilon/2$-close to H_2.

Consider the tuple of random variables $z = (\mathsf{key}_L, \mathsf{key}_R, \mathsf{trans}, \widetilde{\mathsf{key}}_L, \widetilde{\mathsf{key}}_R)$ in H_2. Suppose that we sample c_L, c_R, and $\tilde{m}$ according to their proper distributions in H_2 conditioned on z and then output c_L and $\tilde{m}$. Further, conditioned on z, the output of the tampering function on c_L is independent of the output of the tampering function on c_R. We will use this fact to construct a (possibly inefficient) split-state tampering function $f^{z,\rho}, g^{z,\rho}$ with the corresponding values hardcoded. Denote $\mathsf{Sim}' = (\mathsf{Sim}'_1, \mathsf{Sim}'_2)$ the simulator for the underlying split-state encoding $(\mathsf{E}', \mathsf{D}')$. We use the above observation to construct a simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ as follows.

1. Sim_1 samples $z = (\mathsf{key}_L, \mathsf{key}_R, \mathsf{trans}, \widetilde{\mathsf{key}}_L, \widetilde{\mathsf{key}}_R)$ according to its distribution in H_2 and random coins ρ and derives the resulting split-state tampering functions $f^{z,\rho}, g^{z,\rho}$, defined as follows:

   On input x_L (resp. x_R), $f^{z,\rho}(x_L)$ (resp. $g^{z,\rho}$) samples ct_L (resp. ct_R) uniformly at random conditioned on z and x_L (resp. x_R) using random coins ρ. This fixes $c_L = (\mathsf{key}_R, \mathsf{ct}_L)$ (resp. $c_R = (\mathsf{key}_L, \mathsf{ct}_R)$). It then executes the role of the left (resp. right) player in the leaky split-state protocol (which we assume WLOG is deterministic given input c_L (resp. c_R) and transcript trans) and outputs $c_{\tilde{L}}$ (resp. $c_{\tilde{R}}$) It then applies $\mathsf{Decrypt}(\widetilde{\mathsf{key}}_L, c_{\tilde{L}})$ (resp. $\mathsf{Decrypt}(\widetilde{\mathsf{key}}_R, c_{\tilde{R}})$) and outputs $\tilde{x}_L$ (resp. $\tilde{x}_R$).
2. Invoke Sim' on split-state functions $f^{z,\rho}, g^{z,\rho}$, as above. This gives $(x_L, s) \leftarrow \mathsf{Sim}'(f^{z,\rho}, g^{z,\rho})$ (where $s \in \{0,1\}^k \cup \{\mathtt{same}\}$).
3. Sample ct_L uniformly at random conditioned on z and x_L, using the same random coins ρ that are hardwired into $f^{z,\rho}$.
4. Output $c_L = (\mathsf{key}_R, \mathsf{ct}_L), \mathsf{trans}, s$.

□

Now, we conclude by recalling that the construction of [ADL18] is an augmented split-state non-malleable code with special encoding.[26]

Lemma A.6 ([ADL18])**.** *There exist efficient $n^{-\omega(1)}$-augmented-split-state non-malleable codes with special encoding.*

Our main theorem of this section is a corollary of Lemma A.5, Lemma A.6 and Proposition 5.

Theorem A.7. *For any constant $\alpha \in (0, 1/4)$, there exist efficient α-leakage-resilient $n^{-\omega(1)}$-augmented-split-state non-malleable codes with special encoding.*

[26] The special encoding is not noted explicitly in either [ADL18] but easy to observe because to encode a message x, first it is encoded as $y \in \mathbb{F}_p$ (via an affine evasive encoding scheme), and then the encoder simply chooses A, B in $\mathbb{F}_p^n$ uniformly at random such that $\langle A, B \rangle = y$.

A.3 Augmenting alternate-non-malleability

In this section, we observe that the alternative definition of non-malleability of [DPW10] (Def. A.8) can be extended to handle this notion of augmented non-malleability. As before these definitions are equivalent up to an additive factor of 2^{-k} in the security parameter.

We give a definition for the specific setting of leakage-resilient split-state.

Definition A.8 (Alternate Definition of Augmented Non-Malleability)**.** *Let* $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^{2n}$ *and* $\mathsf{Dec} : \{0,1\}^{2n} \to \{0,1\}^k \cup \{\perp\}$ *be a coding scheme. We say that* $(\mathsf{Enc}, \mathsf{Dec})$ *is an* α*-leakage-resilient augmented-split-state* ϵ*-alternative-non-malleable if for any* $m_0, m_1 \in \{0,1\}^k$ *and any pair of* α*-leaky split-state tampering functions* f, g*, we have:*

$$\mathrm{AltANM}^f_{m_0,m_1}(0) \approx_\epsilon \mathrm{AltANM}^f_{m_0,m_1}(1)$$

where we define the two experiments by

$$\mathrm{AltANM}^f_{m_0,m_1}(b) := \left\{ \begin{array}{c} c_L, c_R \leftarrow \mathsf{Enc}(m_b), (\tilde{c}_L, \tilde{c}_R, \mathsf{trans}) \leftarrow \langle f(c_L), g(c_R) \rangle, \tilde{m} = \mathsf{Dec}(\tilde{c}) \\ \textit{Output } (c_L, \mathsf{trans}, \textsf{\textit{same}}) \textit{ if } \tilde{m} \in \{m_0, m_1\}, \textit{ and } (c_L, \mathsf{trans}, \tilde{m}) \textit{ otherwise.} \end{array} \right\}$$

Lemma A.9. *If* $(\mathsf{Enc}, \mathsf{Dec})$ *is an* α*-leakage-resilient augmented-split-state* $(2n, k, \epsilon)$*-non-malleable code, then it is an* α*-leakage-resilient augmented-split-state* $(2n, k, 2\epsilon)$*-alternate-non-malleable code. If* $(\mathsf{Enc}, \mathsf{Dec})$ *is an* α*-leakage-resilient augmented-split-state* $(2n, k, \epsilon)$*-alternate-non-malleable code, then it is an* α*-leakage-resilient augmented-split-state* $(2n, k, \epsilon + 2^{-k})$*-non-malleable code.*

See Appendix E.2 for the proof.

B Impossibility of Negligible Error via Non-Deterministic Reductions

In this section, we rule out black-box reductions from any function f that is $(1/2 + \delta)$-hard (where δ is a small constant) for n^d-size i-nondeterministic circuits to an efficiently computable function F that is $(1/2 + \epsilon)$-hard relative to some distribution Y for non-uniform circuits of size n^c for constant $c < d$ and negligible ϵ (see Theorem B.2). We then show that this implies that there is no black-box reduction from any function f that is $(1/2 + \delta)$-hard (where δ is a small constant) for n^d-size i-nondeterministic circuits to efficient non-malleable codes with negligible error that are resilient to tampering by non-uniform circuits of size n^c for constant $c < d$ (see Corollary B.3). Since f as above can be constructed from the scaled down and padded characteristic function of some (average case hard) language in E, it means that if one can compute the characteristic function of an E-complete language on all inputs (i.e. break the worst-case hardness of an E-complete language), then one can compute f on average (with probability $1/2 + \delta$). Thus, the above results also rule out reductions from the assumption that E is (worst-case) hard for exponential size i-nondeterministic circuits.

We begin with a definition of a black-box reduction:

Definition B.1. *A black-box reduction* Red *from a function* $f : \{0,1\}^k \to \{0,1\}$ *that is* $(1/2 + \delta)$*-hard over the uniform distribution for* n^d*-size* i*-nondeterministic circuits to a function* $F : \{0,1\}^n \to \{0,1\}$ *that is* $(1/2 + \epsilon)$*-hard over a distribution* Y *for non-uniform circuits of size* n^c *for constant* c *has the following properties:*

- *The reduction* Red *must be an* i*-nondeterministic circuit that makes oracle queries, represented by "oracle gates". Let* $q(n) < n^d$ *be the number of oracle gates in the circuit that represents* Red*. Then* Red *can have size at most* $(n^d - q(n) \cdot n^c)$*. This captures the fact that* Red *is only useful in obtaining a contradiction to the hardness of* f *if the composition of* Red *and* C *(denoted* $\mathsf{Red}^{C(\cdot)}$*) is contained in the class of circuits for which* f *is assumed to be hard, whenever* C *has size at most* n^c*.*
- *There exists a constant* $\tilde{c}$ *such that for any adversary* C *(even inefficient) such that*

$$\Pr[C(\mathsf{Y}) = F(\mathsf{Y})] \geq 1/2 + \epsilon,$$

we have that

$$\Pr_{x \overset{u}{\leftarrow} \{0,1\}^k}[\mathsf{Red}^{C(\cdot)}(x) = f(x)] \geq 1/2 + \delta + 1/n^{\tilde{c}}.$$

This captures the fact that Red *must break the underlying assumption in the case that it interacts with an adversary breaking* F*. We call an adversary* C *for which* $\Pr[C(\mathsf{Y}) = F(\mathsf{Y})] \geq 1/2 + \epsilon$*, a* valid *adversary.*

The following theorem is similar in spirit to Theorem 6.5 in [AASY16], which ruled out certain types of black-box hardness amplification that employ reductions that are i-nondeterministic circuits. The key difference in terms of the types of constructed functions F that are ruled out by the respective results is that Theorem 6.5 in [AASY16] ruled out only functions F that are hard with respect to the uniform distribution, whereas we rule out functions F that are hard with respect to an arbitrary distribution Y. Ruling out such F is necessary for us to obtain Corollary B.3 below, which rules out black-box reductions from f as above to non-malleable codes. There are several further differences between our theorem and that of [AASY16]. We discuss those in Remarks 1 and 2 following the theorem statement.

Theorem B.2. *Assume there exists a function $f : \{0,1\}^k \rightarrow \{0,1\}$ that is $1/2+\delta$-hard over the uniform distribution for n^d-size i-nondeterministic circuits, where $n := n(k)$, d is constant and $\delta \in \Omega(1)$ is constant. Then there is no black-box reduction Red from f to a function $F : \{0,1\}^n \rightarrow \{0,1\}$ with the following properties:*

- *F is computable using polynomial size circuits of size n^a for constant a.*
- *F is $1/2+\epsilon$-hard over a distribution Y for non-uniform circuits of size n^c for constant c and negligible $\epsilon \in n^{-\omega(1)}$.*

Remark 1. While Theorem 6.5 of [AASY16] is unconditional, we require $f : \{0,1\}^k \rightarrow \{0,1\}$ that is $1/2+\delta$-hard over the uniform distribution for n^d-size i-nondeterministic circuits. Further, our notion of black-box reduction requires that Red succeed in computing $f(x)$ with probability $1/2+\delta+1/\mathsf{poly}(n)$ in the case that it interacts with a valid adversary. However, unlike Theorem 6.5 of [AASY16], we allow Red to depend on f in an arbitrary way, whereas [AASY16] required a universal reduction that succeeds on every function f. Personal communication with R. Shaltiel [Sha21] indicated that the authors were aware of the limitation in their result of Red being independent of f and had sketched a proof for the case where Red could depend on f, but extending the proof in this way required a similar assumption to ours that the function $f : \{0,1\}^k \rightarrow \{0,1\}$ is $1/2+\delta-1/\mathsf{poly}(n)$-hard over the uniform distribution for n^d-size i-nondeterministic circuits.

Remark 2. The above theorem requires that the function F be computable with poly size circuits, unlike Theorem 6.5 of [AASY16]. However, our result rules out reductions Red that are *not* security-parameter preserving, meaning that if the reduction receives as input $x \in \{0,1\}^k$, it is not restricted to query its oracle on input length $n := n(k)$, but may query its oracle on any input length n'. We note that the fact that the reduction is security-parameter preserving (i.e. on input $x \in \{0,1\}^k$ it may only query its oracle on inputs of length $n := n(k)$) was assumed in the prior work of [AASY16] although they did not explicitly state it, a $1/2+\epsilon$-hard F exists unconditionally for circuits of size n^c. Further, for applications, such as for construction of non-malleable codes discussed below, F must be explicit and poly-time computable in order to be useful, since e.g. F essentially corresponds to the "decode" algorithm of the NMC. Thus, ruling out efficient F is the case of interest.

Remark 3. While we do not place restrictions on the function $n(k)$, note that since the size of Red is at most n^d (by Property 1 of Definition B.1) and since $x \in \{0,1\}^k$ is the input to the circuit Red, we must have that $k \in O(n^d)$.

Before proving the theorem, we present the following corollary to rule out black-box reductions for NMC.

Corollary B.3. *Let $\delta \in \Omega(1)$ and let $\epsilon \in n^{-\omega(1)}$. Assume there exists a function $f : \{0,1\}^k \rightarrow \{0,1\}$ that is $1/2+\delta$-hard over the uniform distribution for n^d-size i-nondeterministic circuits, where $n = \mathsf{poly}(k)$ and d is constant. Then there is no black-box reduction Red from f to a 1-bit non-malleable code $(\mathsf{Enc}, \mathsf{Dec})$ with codeword length n against circuits of size $n^{c'}$ for constant c', with ϵ error.*

To obtain Corollary B.3 from Theorem B.2, consider the task of constructing a 1-bit NMC with codeword length n and negligible error $\epsilon = n^{-\omega(1)}$ for $\mathsf{SIZE}[n^{c'}]$, where c' is a constant, from a function f over domain $x \in \{0,1\}^k$ (where $n \in \mathsf{poly}(k)$) that is $1/2+\delta$-hard for n^d-size i-nondeterministic circuits.

Recall that the following is an equivalent security definition for ϵ-non-malleable codes that encode a single bit:

Definition B.4. *An encoding scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *for one bit messages is* ϵ*-non-malleable against a tampering class* $\mathsf{SIZE}[n^{c'}]$*, if for all* $g \in \mathsf{SIZE}[n^{c'}]$*:*

$$\Pr_{b \overset{u}{\leftarrow} \{0,1\}}[\mathsf{Dec}(g(\mathsf{Enc}(b))) = 1 - b] \leq \frac{1}{2} + \epsilon$$

where the randomness is over E*, in addition to the uniform message* b*.*

The above definition implies that for any 1-bit ε-NMC against $n^{c'}$-size circuits, and any $C \in \mathsf{SIZE}[n^c]$ for constant c such that $\mathsf{SIZE}[n^c] \subseteq \mathsf{SIZE}[n^{c'} - c_1 \cdot n]$, for some constant c_1,

$$\Pr_{b \overset{u}{\leftarrow} \{0,1\}}[\mathsf{Dec}(\mathsf{Enc}(b)) = C(\mathsf{Enc}(b))] \leq \frac{1}{2} + \epsilon,$$

since otherwise there is a simple tampering attack with a circuit $C' \in \mathsf{SIZE}[n^{c'}]$. Specifically, on input $\mathsf{Y} = \mathsf{Enc}(b), b \overset{u}{\leftarrow} \{0,1\}$, C' runs $C(\mathsf{Y})$, outputs a hardcoded encoding of 0 if C outputs 1 and outputs a hardcoded encoding of 1 if C outputs 0. It is straightforward to see that $\Pr_{b \overset{u}{\leftarrow} \{0,1\}}[\mathsf{Dec}(C'(\mathsf{Enc}(b))) = 1 - b] > \frac{1}{2} + \epsilon$ and so C' breaks ϵ-non malleability. Note that the size of C' is at most the size of C plus a linear in n circuit of size $c_1 \cdot n$ for constant c_1. So $C' \in \mathsf{SIZE}[n^c + c_1 \cdot n] \subseteq \mathsf{SIZE}[n^{c'} - c_1 \cdot n + c_1 \cdot n] = \mathsf{SIZE}[n^{c'}]$.

Thus, if we have a black-box reduction $\mathsf{Red}^{C'}$ from f to $(\mathsf{Enc}, \mathsf{Dec})$ with ϵ error, where Red has size $n^d - q(n) \cdot n^{c'}$, then we can re-write it as $\mathsf{Red}^{C'^C}$. Setting $\mathsf{Red}' = \mathsf{Red}^{C'}$, we obtain a black-box reduction of size at most $n^d - q(n) \cdot n^{c'} + q \cdot c_1 \cdot n \leq n^d - q \cdot n^c$ from f to $F = \mathsf{Dec} : \{0,1\}^n \to \{0,1\}$, which is ϵ-hard for circuits of size n^c with respect to the distribution $\mathsf{Y} \leftarrow \mathsf{Enc}(b), b \overset{u}{\leftarrow} \{0,1\}$. However, this yields a contradiction to Theorem B.2 and so a black-box reduction $\mathsf{Red}^{C'}$ from f to $(\mathsf{Enc}, \mathsf{Dec})$ with ϵ error cannot exist.

We now proceed to prove Theorem B.2.

Proof of Theorem B.2. We will first give a proof overview and present the full proof for the case that Red is security parameter preserving. We then discuss how to extend the result to non-security parameter preserving reductions.

Proof overview. Our high level structure is similar to the structure of the proof of Applebaum et al. [AASY16]. Our goal is to create two adversaries C, C' against F. Both adversaries get some input y in the support of Y and will output an answer by doing a lookup on a large string (N or N') indexed by $H_s(\mathsf{y})$, where H_s is an n^d-wise independent hash function. Note that this differs at a high level from Applebaum et al. [AASY16] due to our use of the hash function. More specifically, C (resp. C') will respond to input y with the output $F(\mathsf{y}) \oplus N[H_s(\mathsf{y})]$ (resp. $F(\mathsf{y}) \oplus N'[H_s(\mathsf{y})]$). The idea will be to sample each index of N from a distribution over $\{0,1\}$ that is negligibly biased towards 0, and sample each index of N' from the uniform distribution over $\{0,1\}$. We will then argue that:

1. C is a *valid* adversary w.h.p. over choice of N. This implies that w.h.p. over choice of N, $\mathsf{Red}^{C(\cdot)}$ agrees with f with probability at least $1/2 + \delta + 1/\mathsf{poly}(n)$, since this is one of the requirements of a black-box reduction Red.
2. C' is a *useless* adversary w.h.p. over choice of N. I.e. w.h.p. over choice of s, the view of $\mathsf{Red}^{C(\cdot)}$ over random choice of N' can be simulated by returning 0 or 1 uniformly at random in response to each query (specifically, this is a good simulation as long as no collisions occur). Further, this simulation can be achieved via sampling from a distribution $\mathcal{Z}$ over size n^d circuits, where the random responses of C are sampled and then hardcoded into a circuit that runs Red. Thus, the success probability of $\mathsf{Red}^{C(\cdot)}$ over choice of s, N must be negligibly close to $1/2 + \delta$, since otherwise there must be a particular circuit in the support of the distribution $\mathcal{Z}$ that agrees with f with probability more than $1/2 + \delta$ (a contradiction to the assumption on the hardness of f).

To argue (1), we note that Y must have min-entropy at least $\log(1/\epsilon)$. We choose the parameters of H_s (i.e. output length) appropriately to ensure that w.h.p. over choice of s, $H_s(\mathsf{Y})$ is close to uniform random. This implies that as long as N has a sufficiently high (but still $1/2 + \mathrm{negl}(n)$) fraction of 0's, then $\Pr[C(\mathsf{Y}) =$

$F(\mathsf{Y})] = \Pr[N(H_s(\mathsf{Y})) = 0] \approx \frac{\text{number of 0's in } N}{\text{length of } N}$ is at least $1/2 + \epsilon$. Using standard Chernoff/Hoeffding bounds, we can show that w.h.p. over choice of N, N has a sufficiently high fraction of 0's.

Now, given that (1) and (2) hold, we construct a constant depth circuit that distinguishes between the two distributions that N and N' are drawn from. This circuit will be of size $2^{\mathsf{poly}(n)}$ and will proceed by running the reduction on each input x to f and all possible non-uniform and non-deterministic advice strings. The reduction will then approximately count the number of inputs x for which the output of the reduction agrees with f. This can be done with constant depth, since the reduction need only distinguish between agreement of $1/2 + \delta$ versus $1/2 + \delta + 1/\mathsf{poly}(n)$. This will contradict the fact that to distinguish strings N, N' drawn from the distributions above, constant depth circuits require size $2^{n^{\omega(1)}}$. We now proceed with the formal proof.

Formal proof for security parameter preserving reductions. Let Y be the assumed hard distribution for F. Note that Y must have min-entropy at least $\log(1/\epsilon)$, since otherwise there is some y^* that occurs with probability greater than ϵ and C can agree with F with probability at least $1/2 + \epsilon$ by simply hardcoding $b^* = F(\mathsf{y}^*)$, and $b' = \mathsf{argmax}_{b \in \{0,1\}} \rho(b)$, where $\rho(b) := \Pr[F(\mathsf{Y}) = b \mid \mathsf{Y} \neq \mathsf{y}^*]$, outputting b^* if $\mathsf{Y} = \mathsf{y}^*$ and outputting b' otherwise. Further, let H_s be an n^d-wise independent hash with input length n and output length $\ell_H := \log(1/\epsilon)/3$ and seed s chosen uniformly at random from set $\mathcal{S}$. We also define $\mathsf{L_H} := 2^{\ell_H}$ and $\ell := 8n \cdot n^{2\tilde{c}}$.

Fact B.5. *H_s is a strong extractor for sources of min-entropy $\log(1/\epsilon)$ and statistical distance $2^{-\log(1/\epsilon)/3} = \epsilon^{1/3}$ from uniform. This implies that, with probability $1 - 1/(2\ell)$ over choice of $s \sim \mathcal{S}$, $H_s(\mathsf{Y})$ has negligible statistical distance at most $2\ell\epsilon^{1/3}$ from the uniform distribution over $\{0,1\}^{\ell_H}$.*

We also define the following distribution:

Definition B.6. *For $0 \leq p \leq 1$ and a natural number t, we use N_p^t to denote the distribution of t i.i.d. bits where each of them has probability p to evaluate to 1.*

We will consider two distributions: The first consists of ℓ independent draws from $N^{\mathsf{L_H}}_{1/2-3\epsilon^{1/8}}$, which is equivalent to a single draw from $N^{\ell\cdot\mathsf{L_H}}_{1/2-3\epsilon^{1/8}}$. The second consists of ℓ independent draws from $N^{\mathsf{L_H}}_{1/2}$, which is equivalent to a single draw from $N^{\ell\cdot\mathsf{L_H}}_{1/2}$.

For $j \in [\ell]$ and $N_j \sim N^{\mathsf{L_H}}_{1/2-3\epsilon^{1/8}}$, let C_{s_j,N_j} be the adversary that on input Y, outputs $F(\mathsf{Y}) \oplus N_j[H_{s_j}(\mathsf{Y})]$. Let $I_{s_j,N_j,x}$ be the indicator variable that is set to 1 when $\mathsf{Red}^{C_{s_j,N_j}(\cdot)}(x) = f(x)$ and is set to 0 otherwise.

Lemma B.7. *With probability at least $1/2$ over choice of $s_1, \ldots, s_\ell \sim \mathcal{S}$, we have that with all but negligible probability over $N_1, \ldots, N_\ell$ drawn independently from $N^{\mathsf{L_H}}_{1/2-3\epsilon^{1/8}}$,*

$$\frac{\sum_{j \in \ell, x \in \{0,1\}^k} [I_{s_j,N_j,x}]}{\ell \cdot 2^k} \geq 1/2 + \delta + 1/n^{\tilde{c}}.$$

Proof. First, note that if N_j of length $\mathsf{L_H}$ has at most $1/2 - 2\epsilon^{1/8}$ number of 1's, and at least $1/2 + 2\epsilon^{1/8}$ number of 0's then

$$\Pr[N_j[U_{\ell_H}] = 0] \geq 1/2 + 2\epsilon^{1/8},$$

where U_{ℓ_H} is a uniform random string of length ℓ_H.

Since by Fact B.5, we have that with probability $1 - 1/(2\ell)$ over choice of s_j, $H_{s_j}(\mathsf{Y})$ is statistically $2\ell \cdot \epsilon^{1/3}$-close to uniform random, we therefore have that if N_j of length $\mathsf{L_H}$ has at most $1/2 - 2\epsilon^{1/8}$ number of 1's, and at least $1/2 + 2\epsilon^{1/8}$ number of 0's then

$$\Pr[F(\mathsf{Y}) = C_{s_j,N_j}(\mathsf{Y})] = \Pr[N_j[H_s(\mathsf{Y})] = 0] \geq 1/2 + \epsilon^{1/8} > 1/2 + \epsilon,$$

for sufficiently large n, since $\epsilon \in o(2\ell \cdot \epsilon^{1/3}) \in o(\epsilon^{1/8})$. Thus, with probability $1 - 1/(2\ell)$ over choice of s_j, we have (by standard Chernoff/Hoeffding bounds) that with probability at least $1 - \exp(-\mathsf{L_H} \cdot (\epsilon^{1/8})^2) = 1 - \exp(-(1/\epsilon)^{1/12})$ (which is $1 - \mathsf{negl}(n)$) over choice of N_j, C_{s_j,N_j} is a valid adversary. Moreover, by a union bound, with probability $1/2$ over choice of $s_1, \ldots, s_\ell$, we have that with probability $1 - \mathsf{negl}(n)$ over choice of $N_1, \ldots, N_\ell$, all C_{s_j,N_j} are valid adversaries.

Recall $I_{s_j,N_j,x}$ is the indicator variable that is set to 1 when $\mathsf{Red}^{C_{s_j,N_j}(\cdot)}(x) = f(x)$ and is set to 0 otherwise. Then, by Property 2 of Definition B.1 (which says Red succeeds when given oracle access to a valid adversary), with probability 1/2 over choice of $s_1, \dots, s_\ell$, we have that with all but negligible probability over $N_1, \dots, N_\ell$ drawn independently from $N^{\mathsf{L_H}}_{1/2-3\epsilon^{1/8}}$,

$$\frac{\sum_{j \in \ell, x \in \{0,1\}^k} [I_{s_j,N_j,x}]}{\ell \cdot 2^k} \geq 1/2 + \delta + 1/n^{\tilde{c}}.$$

□

For $N'_j \sim N_{1/2}$, we now consider an adversary C'_{s_j,N'_j} that on input Y, outputs $F(\mathsf{Y}) \oplus N'_j[H_{s_j}(\mathsf{Y})]$. Let $I'_{s_j,N'_j,x}$ be the indicator variable that is set to 1 when $\mathsf{Red}^{C'_{s_j,N'_j}(\cdot)}(x) = f(x)$ and is set to 0 otherwise.

Lemma B.8. *With all but negligible probability over choice of $s_1, \dots, s_j \sim \mathcal{S}$, we have that with all but negligible probability over $N'_1, \dots, N'_\ell$ drawn independently from $N^{\mathsf{L_H}}_{1/2}$,*

$$\frac{\sum_{j \in \ell, x \in \{0,1\}^k} [I'_{s_j,N'_j,x}]}{\ell \cdot 2^k} \leq 1/2 + \delta + 1/(2n^{\tilde{c}}).$$

Proof. Note that C'_{s_j,N'_j} simply outputs an independent, random 0/1 bit for each query Y from Red, unless a hash collision occurs on Y.

Since H_{s_j} is n^d-wise independent and Red makes at most n^d queries per run, the probability over random choice of $s_j \sim \mathcal{S}$ that a collision occurs in a single run of Red on random input x and random N'_j is:

$$\Pr_{s_j \sim \mathcal{S}, N'_j \sim N_{1/2}, x \sim U_k}[H_{s_j}(\mathsf{y}_u) = H_s(\mathsf{y}_w) \text{ for two distinct queries of } \mathsf{Red}] \leq \frac{n^{2d}}{\mathsf{L_H}} = n^{2d} \cdot \epsilon^{1/3}.$$

Let Ev be the event that a collision occurs during a run of Red. Conditioned on $\overline{\mathsf{Ev}}$, the output of $\mathsf{Red}^{C'_{s_j,N'_j}}$ is identically distributed to the output of a circuit drawn from the following distribution $\mathcal{Z}$:

- Draw a string r of length $q(n) < n^d$, where $q(n)$ is an upperbound on the number of distinct queries made by Red on any input $x \in \{0,1\}^k$ (recall that we assume WLOG that Red never queries on the same string twice).
- Construct the circuit Z_r that hardwires r and whose computation proceeds on input x by running Red on input x and responding to the i-th query of Red with $r[i]$. Z_r then returns whatever Red returns.

Since Red has size at most $n^d - q(n) \cdot n^c$, the circuits in the above distribution $\mathcal{Z}$ all have size $O(n^d)$, and so by our assumption on n^d-hardness of f we have that

$$\Pr_{Z_r \sim \mathcal{Z}, x \sim U_k}[Z_r(x) = f(x)] \leq 1/2 + \delta,$$

since otherwise there must exist a particular r such that

$$\Pr_{x \sim U_k}[Z_r(x) = f(x)] > 1/2 + \delta,$$

a contradiction to the $1/2 + \delta$-hardness of f. Further,

$$\Pr_{s_j \sim \mathcal{S}, N'_j \sim N_{1/2}, x \sim U_k}[\mathsf{Red}^{C'_{s_j,N'_j}(\cdot)}(x) = f(x) \mid \overline{\mathsf{Ev}}] = \Pr_{Z_r \sim \mathcal{Z}, x \sim U_k}[Z_r(x) = f(x) \mid \overline{\mathsf{Ev}}],$$

which implies that

$$\Pr_{s_j \sim \mathcal{S}, N'_j \sim N_{1/2}, x \sim U_k}[\mathsf{Red}^{C'_{s_j,N'_j}(\cdot)}(x) = f(x)] \leq 1/2 + \delta + 2 \Pr_{s_j \sim \mathcal{S}, N'_j \sim N_{1/2}, x \sim U_k}[\mathsf{Ev}] \leq 1/2 + \delta + 2n^{2d} \cdot \epsilon^{1/3}.$$

Recall $I'_{s_j,N'_j,x}$ is the indicator variable that is set to 1 when $\mathsf{Red}^{C_{s_j,N'_j}(\cdot)}(x) = f(x)$ and is set to 0 otherwise. Then the above can be re-written as

$$\mathsf{E}_{s_j \sim \mathcal{S}, N'_j \sim N_{1/2}} \left[\frac{\sum_{j \in \ell, x \in \{0,1\}^k} [I'_{s_j,N'_j,x}]}{\ell \cdot 2^k} \right] \leq 1/2 + \delta + 2n^{2d} \cdot \epsilon^{1/3}.$$

Now consider $\ell := 8n \cdot n^{2\tilde{c}}$ independent runs, where in the j-th run we execute Red with independently chosen N'_j, s_j on all inputs x. Then by standard Chernoff/Hoeffding bounds, with probability at least $1 - \exp(-2\ell/(16n^{2\tilde{c}})) = 1 - \exp(-n))$ (which is $1 - \mathsf{negl}(n)$) over $s_1, \dots, s_\ell$ drawn from $\mathcal{S}$, and $N'_1, \dots, N'_\ell$ drawn from $N^{\mathsf{L_H}}_{1/2}$,

$$\frac{\sum_{j \in \ell, x \in \{0,1\}^k} [I'_{s_j,N'_j,x}]}{\ell \cdot 2^k} \leq 1/2 + \delta + 1/(2n^{\tilde{c}}).$$

By Markov's inequality, with all but negligible probability over $s_1, \dots, s_\ell$ drawn from $\mathcal{S}$, we have that with all but negligible probability over $N'_1, \dots, N'_\ell$ drawn from $N^{\mathsf{L_H}}_{1/2}$,

$$\frac{\sum_{j \in \ell, x \in \{0,1\}^k} [I'_{s_j,N'_j,x}]}{\ell \cdot 2^k} \leq 1/2 + \delta + 1/(2n^{\tilde{c}}),$$

which completes the proof of the lemma. □

Combining Lemmas B.7 and B.8, there must exist seeds $s^*_1, \dots, s^*_j \in \mathcal{S}$ such that *both*:

1. With all but negligible probability over $N_1, \dots, N_\ell$ drawn independently from $N^{\mathsf{L_H}}_{1/2 - 3\epsilon^{1/8}}$,

$$\frac{\sum_{j \in \ell, x \in \{0,1\}^k} [I_{s^*_j,N_j,x}]}{\ell \cdot 2^k} \geq 1/2 + \delta + 1/n^{\tilde{c}}.$$

2. With all but negligible probability over $N'_1, \dots, N'_\ell$ drawn independently from $N^{\mathsf{L_H}}_{1/2}$,

$$\frac{\sum_{j \in \ell, x \in \{0,1\}^k} [I'_{s^*_j,N'_j,x}]}{\ell \cdot 2^k} \leq 1/2 + \delta + 1/(2n^{\tilde{c}}).$$

Recall that Red is a non-uniform Σ_i circuit of size n^d with oracle gates. We instantiate the oracle with $F(\cdot) \oplus \tilde{N}_j(H_{s^*_j}(\cdot))$, where $\tilde{N}_j$ is drawn from either $N_{1/2 - 3\epsilon^{1/8}}$ or $N_{1/2}$. For each $j \in [\ell]$, let $B_{x,\alpha,z_1,\dots,z_i}(\tilde{N}_j)$ denote the output of $\mathsf{Red}^{F(\cdot) \oplus \tilde{N}_j(H_{s^*_j}(\cdot))}$ for a fixed input x, non-uniform advice string α and non-deterministic inputs $z_1, \dots, z_i$. Note that the input to $B_{x,\alpha,z_1,\dots,z_i}$ is the length $\mathsf{L_H}$ string $\tilde{N}_j$. Since for a fixed input x, non-uniform advice string α and non-deterministic inputs $z_1, \dots, z_i$, $\mathsf{Red}^{F(\cdot) \oplus \tilde{N}_j(H_{s^*_j}(\cdot))}$ can be viewed as a depth n^d decision tree that makes queries to $F(\cdot) \oplus \tilde{N}_j(H_{s^*_j}(\cdot))$, $B_{x,\alpha,z_1,\dots,z_i}(\tilde{N}_j)$ can be implemented by a depth-2 circuit of size $2^{O(n^d)}$.

We now consider the function $B_{x,\alpha}(\tilde{N}_j)$ defined to be one iff $\exists z_1 \forall z_2 \dots Q z_i : B_{x,\alpha,z_1,\dots,z_i}(\tilde{N}_j) = 1$. Note that this function can be implemented by a circuit of depth i + 2 and size $2^{O(n^d)}$ times the size of a circuit $B_{x,\alpha,z_1,\dots,z_i}$. Overall, we get a depth $i + 2$, size $2^{O(n^d)}$ circuit.

Thus, we can construct the following circuit A_f of size $2^{O(n^d)}$ and depth $i + O(1)$, that gets as input either $\tilde{N}_1, \dots, \tilde{N}_\ell$ that is drawn from either $[N^{\mathsf{L_H}}_{1/2 - 3\epsilon^{1/8}}]^\ell$ or $[N^{\mathsf{L_H}}_{1/2}]^\ell$ and distinguishes the two with advantage more than 0.99. A_f has the characteristic vector of f hardwired—i.e. a vector v_f such that $v_f[x] = f(x)$ for all $x \in \{0,1\}^k$. A_f does the following: In parallel, for every $\alpha \in \{0,1\}^\alpha$, $j \in [\ell]$, $x \in \{0,1\}^k$ evaluate $B_{\alpha,x}(N'_j)$ in depth 2. Obtain an output vector $v_{\alpha,j}$ of size 2^k such that $v_{j,\alpha}[x] = B_{\alpha,x}(N'_j)$. Note that $2^k \cdot 2^\alpha$ must be contained in $2^{O(n^d)}$ since the input and non-uniform advice to Red must be smaller than the total size of the circuit, n^d. For every $\alpha \in \{0,1\}^\alpha$, A_f now approximately counts the number of positions x such that the vector $v_{\alpha,1} || \dots || v_{\alpha,\ell}$ agrees with $v_f || \dots || v_f$ (i.e. ℓ concatenations of v_f). If for some α, the fraction of positions is at least $1/2 + \delta + 1/n^{\tilde{c}}$, A_f outputs 1. If for all α, the fraction of positions is at most $1/2 + \delta + 1/(2n^{\tilde{c}})$, A_f outputs 0. Since the difference is $1/(2n^{\tilde{c}})$, which is $1/\mathsf{poly}(n)$, given outputs

$v_{\alpha,1}, \dots, v_{\alpha,\ell}$ from the previous stage and the hardcoded v_f, the approximate counting can be done by a constant depth circuit of size $2^{\mathsf{poly}(n)}$. Thus, in total A_f has constant depth and size $2^{\mathsf{poly}(n)}$. However, since $3\epsilon^{1/8} \in n^{-\omega(1)}$ is negligible, setting $\epsilon' = 3\epsilon^{1/8}$ and $t = \ell \cdot \mathsf{L_H} = \ell \cdot (1/\epsilon)^{1/3} \in 2^{\mathsf{poly}(n)}$ in the following theorem implies that circuits of constant depth and size $2^{\mathsf{poly}(n)}$ cannot distinguish between $[N_{1/2-3\epsilon^{1/8}}^{\mathsf{L_H}}]^\ell = N_{1/2-3\epsilon^{1/8}}^{\ell \cdot \mathsf{L_H}}$ or $[N_{1/2}^{\mathsf{L_H}}]^\ell = N_{1/2}^{\ell \cdot \mathsf{L_H}}$ with advantage 0.99.

Theorem B.9 ([Vio06, SV10])**.** *There exists a constant $a > 1$, such that for every sufficiently small $\epsilon' > 0$, circuits of depth k and size $s = \exp((1/\epsilon')^{\frac{1}{k+a}})$ cannot distinguish $N_{1/2-\epsilon'}^t$ and $N_{1/2}^t$ with advantage 0.99 for any $t \leq s$.*

We thus obtain a contradiction to the existence of a black-box reduction Red from a function $f : \{0,1\}^k \to \{0,1\}$ that is $(1/2+\delta)$-hard over the uniform distribution for n^d-size i-nondeterministic circuits to a function $F : \{0,1\}^n \to \{0,1\}$ that is $(1/2+\epsilon)$-hard over a distribution Y for non-uniform circuits of size n^c for constant c.

Extending the proof to non-security parameter preserving reductions. Here, we crucially rely on the fact that the construction F is efficiently computable by circuits of size n^a for constant a. We begin by explaining how to modify the adversaries $C_{s_j,N_j}(\cdot)$ and $C'_{s_j,N'_j}(\cdot)$ for the non-security parameter preserving setting. For any fixed input length k, Red may now query its oracle on arbitrary input lengths n'. Further, we now view $\epsilon(\cdot)$ as a negligible function of its input, n'. We therefore index the strings N_j and N'_j by input length $n' \in \{1, \dots, n^d\}$, so $N_j = N_j^1 || \cdots || N_j^{n^d}$ (resp. $N'_j = {N'}_j^1 || \cdots || {N'}_j^{n^d}$), where $N_j^{n'}$ (resp.${N'}_j^{n'}$) has length $\epsilon(n')^{1/3}$. Note that Red cannot query its oracle on input length larger than n^d since Red has size at most n^d, so we must have $n' \in \{1, \dots, n^d\}$. The adversaries $C_{s_j,N_j}(\cdot)$ (resp. $C'_{s_j,N'_j}(\cdot)$) will, similarly to before, return $F(\mathsf{y}) \oplus N_j^{|\mathsf{y}|}(H_{s_j}(\mathsf{y}))$ on input y, where H_{s_j} now corresponds to a set of n^d-wise independent hashes, one for each possible input length $n' \in \{1, \dots, n^d\}$. The only difference will be in how we sample N_j and N'_j. Specifically, For n' such that $(n')^a \leq n^c$, $N_j^{n'}$ (resp. ${N'}_j^{n'}$) are now set to the all 0 string. This represents that the adversaries C_{s_j,N_j}, C'_{s_j,N'_j} always return the correct answer of F evaluated on their input. For n' such that $(n')^a > n^c$, ${N'}_j^{n'}$ is sampled exactly as before, while $N_j^{n'}$ is now sampled as in the security preserving case described above but with bias $3\epsilon(n')^{1/8} < 3\epsilon(n^{c/a})^{1/8}$, Since c and a are both constants, $\epsilon(n^{c/a})$ (and hence also $3\epsilon(n^{c/a})^{1/8}$) is still negligible in n. By defining the adversaries in this way, we preserve all the properties needed for the proof above to go through. Specifically, with probability $1/2$ over $s_1, \dots, s_\ell$, all C_{s_j,N_j} are still valid adversaries with all but negligible probability, since for n' such that $(n')^a \leq n^c$ they always returns the correct answer and for n' such that $(n')^a > n^c$, $\epsilon(n')$ is still negligible in n, which was the only property of ϵ needed for Lemma B.7 to go through. Further, on input length n' such that $(n')^a \leq n^c$, F can be evaluated in size $(n')^a \leq n^c$, whereas on input length n' such that $(n')^a > n^c$, the response is random. So $\mathsf{Red}^{C'_{s_j,N'_j}}$ can still be evaluated in size at most n^d and again $\epsilon(n')$ is still negligible in n, which was the only property of ϵ needed for Lemma B.8 to go through. Finally, the strings N_j and N'_j differ by a negligible fraction, and so the final construction of the distinguishing adversary A_f still leads to contradiction. □

C A Non-Malleable Extractor for Recognizable Sources with Relative Error

Applebaum et al. [AASY16] introduced the notion of a relative error extractor to circumvent some of the shortcomings of $1/poly$ error that is inherent using current techniques. While these extractors fail to achieve negligible error, they do effectively preserve the probability of low preserve the probability of low probability events.

Definition C.1 (Relative Error Extractor [AASY16])**.** *We say that $\mathsf{E} : \{0,1\}^n \to \{0,1\}^m$ is an extractor for a class of sources $\mathcal{X}$ with* relative error α *if for all $X \in \mathcal{X}$ and any event $A \subseteq \{0,1\}^m$*

$$\Pr_X[\mathsf{E}(X) \in A] \in (1 \pm \alpha) \Pr_U[U \in A],$$

where U is the uniform distribution over $\{0,1\}^m$.

In particular, consider constant $\alpha = 1/2$. This does not give meaningful bounds on the total variation distance of the output of the extractor from uniform. However, it does guarantee that if some event A occurs with negligible probability under the uniform distribution (e.g. the event of an adversary breaking a cryptographic scheme), then A still occurs with negligible probability in the output of the extractor.

In this section, we extend this notion to the non-malleable setting. Consequently, our relative error extractors give a generic means of achieving tamper and leakage resilient cryptography (with strong security guarantees).

Definition C.2 (Relative Error Non-Malleable Extractor)**.** *We say $\mathsf{E} : \{0,1\}^n \to \{0,1\}^m$ is a non-malleable extractor with* relative error (α, β) *for a class of sources $\mathcal{X}$ and tampering family $\mathcal{T}$, if for all $X \in \mathcal{X}$ and all $t \in \mathcal{T}$ there exists a simulator $\mathsf{Sim}_{X,t}$ (supported on $\{0,1\}^m \cup \{\mathit{same}\}$) such that for any event $A \subseteq \{0,1\}^{2m}$,*

$$(1-\alpha)\Pr[U\text{Copy}(S_{X,t}, U) \in A] - \beta \leq \Pr[\mathsf{E}(X)\mathsf{E}(t(X)) \in A] \leq (1+\alpha)\Pr[U\text{Copy}(S_{X,t}, U) \in A] + \beta,$$

where $S_{X,t}$ denotes the random variable representing the output of $\mathsf{Sim}_{X,t}$.

To handle small errors on zero probability events in the ideal model, we introduce the additive error term β. We encourage the reader to think of α as "large," $1/\mathsf{poly}$, and β as "small," $2^{-\Omega(m)}$, as this is the parameter range we are aiming for.

C.1 Non-Malleable Extractor Construction

Our construction builds on the conceptual approach of Applebaum et al. [AASY16]. So before delving into our construction we recall their approach.

Theorem C.3 (Relative Error Extractors for Recognizable Sources [AASY16]). *If E is hard for exponential size Σ_3-circuits then there exists a constant $\alpha > 0$ such that for every constant $c > 1$ and sufficiently large n, and every $m \leq \alpha n$ there is an extractor $\mathsf{E} : \{0,1\}^n \to \{0,1\}^m$ for sources with min-entropy $(1-\alpha)n$ that are recognizable by size n^c circuits with relative error n^{-c}. Furthermore, E is computable in time $\mathsf{poly}(n^c)$.*

Their starting point is an ϵ-incomputable function, $f : \{0,1\}^\ell \to \{0,1\}^n$ with long output, where ϵ bounds any small circuits advantage over random guessing in computing f on uniformly random inputs. Because the output is long, it is feasible to construct such function from strong derandomization assumptions where ϵ is negligible, in fact $\epsilon = 2^{-\Omega(n)}$ [TV00].

Definition C.4 (Incomputable Functions [AASY16])**.** *We say a function $f : \{0,1\}^\ell \to \{0,1\}^n$ is ϵ-incomputable by a class $\mathcal{C}$, if for all $C \in \mathcal{C}$, $\Pr_{x \leftarrow \mathcal{U}_\ell}[C(x) = f(x)] \leq \frac{1}{2^n} + \epsilon$.*

Theorem C.5 (Incomputable Functions [AASY16]). *For all natural numbers i, if E is hard for exponential size Σ_{3+i}-circuits then there exists a constant $\alpha > 0$ such that for every constant $c > 1$ and sufficiently large ℓ, and every $n \leq \alpha\ell$ there is a function $f : \{0,1\}^\ell \to \{0,1\}^n$ that is $\ell^{-c}2^{-n}$-incomputable by Σ_i-circuits of size ℓ^c. Furthermore, f is computable in time $\mathsf{poly}(\ell^c)$.*

Given such an f, their construction, $\mathsf{reExt} : \{0,1\}^{\ell+n} \to \{0,1\}^m$ views parses a source recognizable by an n^c size circuit, $R = (R_1, R_2)$, and outputs

$$\mathsf{reExt}(X) := \mathsf{2Ext}(f(R_1), R_2)$$

where $\mathsf{2Ext}$ is an explicit 2-source extractor and f is $\epsilon = 2^{-\Omega(n)}$-incomputable by σ_2-circuits of size n^d, for some constant $d > c$.

They then reduce the relative error property to the incomputability of f. In particular, they observe that if there exists some z such that $\Pr[\mathsf{reExt}(f(R_1), R_2) = z] > (1+\Omega(\epsilon))2^{-m}$, then there must be many "useful" inputs x to the incomputable function f such that conditioning on $R_1 = x$ both preserves the high likelihood of z, $\Pr[\mathsf{reExt}(f(R_1), R_2) = z | R_1 = x] > (1 + \Omega(\epsilon))2^{-m}$, and $R_2 | R_1 = x$ maintains high min-entropy. They then observe that 2-source extractors have the following list-decoding-like guarantee: there aren't too many inputs y that can "explain" the probability of z relative to $(X_2|X_1 = x)$, i.e. for any particular "useful" x

there cannot be not too many y such that $\Pr[\mathsf{2Ext}(y, (X_2|X_1 = x)) = z] > (1 + \Omega(\epsilon))2^{-m}$.[27] Note that, since x is useful, $f(x) = y^*$ is necessarily in this set. So their circuit to compute $f(x)$ simply hopes the input to f, x, is useful, "list-decodes" possible y's that could "explain" z relative to $(X_2|X_1 = x)$, and uniformly samples a y from the list. If x is indeed useful, than $f(x)$ will be in this list and the list will be sufficiently small that $f(x)$ is sampled with high enough probability to break the incomputability of f.

That said, even in this good case the list is still of exponential size, so there is not space in circuit to write down this list. Thankfully, due to the fact that R is recognizable and two-source extractor is efficient, membership in this list can be decided by a small NP-circuit (the NP-oracle is used to approximate the conditional probability z occurs for any fixed y, see Theorem 2.20). Thus, again by classical techniques (see Theorem 2.19), members of this list can be sampled uniformly with a Σ_2 circuit.

This paradigm is our starting point, but tampering introduces some new hurdles. One might hope that it suffices to replace the 2-source extractor in the construction of [AASY16] with a *non-malleable* 2-source extractor. While 2-source non-malleable extractors enjoy a similar list-decoding-like property, it is unclear how to perform a similar reduction without compressing the source before feeding it to f and the 2-source non-malleable extractor in order to reduce the tampering to a split-state tampering.[28] In more detail, we will condition on "useful" inputs x as well as "helpful" tampered inputs $\tilde{x}$. If these inputs are too long, no entropy will remain in the source after conditioning on such inputs and we won't be able to utilize the list-decoding property. Moreover, due to the strict accounting required here, it is important that any compressions of the source have high min-entropy, and are not merely close to high min-entropy variables (especially if close just means $1/\mathsf{poly}$!). For this reason, we compress using relative-error extractors for samplable sources.

Figure C.1: Non-Malleable Extractor with Relative Error for $(N, (1-\gamma)N)$-Sources Recognized by and Tampered by Size N^c circuits

Ingredients:

- $\mathsf{Ext}^1_{\mathsf{samp}} : \{0,1\}^N \to \{0,1\}^\ell$ an extractor for sources with min-entropy $(1-\gamma)N$ recognized by size N^c circuits with 1 relative error, computable in time N^{c_1} for some constant $c_1 > c$.
- $\mathsf{Ext}^2_{\mathsf{samp}} : \{0,1\}^N \to \{0,1\}^n$ an extractor for sources with min-entropy $(1-\gamma)N - 2\ell - 3 - \log(1/\beta)$ recognized by size $c'N^{c_1}$ circuits with 1 relative error, computable in time N^{c_2} for some constants $c_2 > c_1$ and c'.
- $f : \{0,1\}^\ell \to \{0,1\}^n$ a function that is ϵ-incomputable by Σ_3-circuits of size $c''N^{c_3}$
- $2\text{NMExt} : \{0,1\}^{2n} \to \{0,1\}^{n'}$ a 2-source non-malleable extractor with error $\beta/12$ for independent $(n, (1-\gamma)n)$-source and $(n, n-1)$-source that is computable in size N^{c_1}.
- $\mathsf{Trunc}_m : \{0,1\}^{n'} \to \{0,1\}^m$ truncates an n'-bit input to m bits, $x_1, \ldots, x_{n'} \mapsto x_1, \ldots, x_m$.

Construction:

$$\mathsf{E}(R) := \mathsf{Trunc}_m(2\text{NMExt}(f(\mathsf{Ext}^1_{\mathsf{samp}}(R)), \mathsf{Ext}^2_{\mathsf{samp}}(R)))$$

Theorem C.6. *The construction, E, in Figure C.1 is a non-malleable extractor for sources recognized by N^c size circuits with min-entropy $(1-\gamma)N$ and tampered by N^c circuits with $(1/N^c, 2^{2m} \cdot \beta)$-relative error.*

If we instantiate the ingredients in Figure C.1 with Theorem C.3, Theorem C.5, and Theorem 2.10, then we can take $n = O(\ell)$, $\ell = O(N)$, and $m = O(N \log\log N / \log N)$ to derive the following corollary:

Corollary C.7. *If E is hard for exponential size Σ_6-circuits, then for every constant $d > 1$ there exists constants $\gamma, \zeta > 0$ such that for all constants $c > 1$ and sufficiently large n and $m \leq \zeta n \log\log n / \log n$ there is a non-malleable extractor for sources recognized by n^c size circuits with min-entropy $(1-\gamma)n$ and tampered by n^c circuits with $(1/n^c, 2^{-dm})$-relative error.*

[27] For those familiar, this is similar to the fact that every two source extractor is "strong," up to a loss in parameters.

[28] Additionally, at this time, constructions of 2-source non-malleable extractors in the literature could only handle sources with comparatively high min-entropy, relative to standard 2-source extractors.

C.2 Canonical Simulation for Non-Malleable Extractors

Before proving the main technical lemma in this section, we need the following observation that a non-malleable extractor always admits a canonical simulator.

Critically for us, if sampling the source, tampering the source, and evaluating the extractor can be done efficiently (according to some notion of efficiency) then the canonical simulator is also efficient.

Lemma C.8. *Let $\mathsf{E}' : \{0,1\}^n \to \{0,1\}^{n'}$ be an ϵ-non-malleable seedless extractor for source class $\mathcal{X}$ and tampering family $\mathcal{T}$. Then, then for any $X \in \mathcal{X}$ and $t \in \mathcal{T}$, the following is a "canonical" simulator with error at most $2\epsilon + 2^{-(n'+1)}$.*

(We define, $S_{X,t}$, the random variable corresponding to its output.)

$$S_{X,t} := \left\{ \begin{array}{c} \textit{Sample } X \textit{and compute} (z', \tilde{z}') = \mathsf{E}'(X), \mathsf{E}'(t(X)) \\ \textit{if } z' = \tilde{z}', \textit{ output } \texttt{same} \\ \textit{otherwise, output } \tilde{z}' \end{array} \right\}.$$

In other words,

$$\forall X \in \mathcal{X}, \forall t \in \mathcal{T}, \quad \varDelta(\mathsf{E}'(X), \mathsf{E}'(t(X)); U_{n'}, \mathrm{Copy}(S_{X,t}, U_{n'})) \leq 2\epsilon + 2^{-(n'+1)},$$

where $U_{n'}$ is a uniformly distributed random variable over n' bits.

Proof. First, because E' is ϵ-non-malleable, for any $X \in \mathcal{X}$ and $t \in \mathcal{T}$, there exists $S^{E'}_{X,t}$ such that

$$\varDelta(\mathsf{E}'(X), \mathsf{E}'(t(X); U_{n'}, \mathrm{Copy}(S^{E'}_{X,t}, U_{n'}) \leq \epsilon.$$

From this it follows that

$$\begin{aligned} 2\epsilon &\geq 2\varDelta(\mathsf{E}'(X), \mathsf{E}'(t(X); U_{n'}, \mathrm{Copy}(S^{E'}_{X,t}, U_{n'}) \\ &= \sum_{z'} |\Pr[\mathsf{E}'(X) = \mathsf{E}'(t(X)) = z'] - 2^{-n'} \Pr[S^{E'}_{X,t} = \texttt{same}] - 2^{-n'} \Pr[S^{E'}_{X,t} = z']| \\ &\qquad + \sum_{z' \neq \tilde{z}'} |\Pr[\mathsf{E}'(X)\mathsf{E}'(t(X)) = z'\tilde{z}'] - \frac{1}{2^{n'}-1} \Pr[S^{E'}_{X,t} = \tilde{z}']| \\ &\geq \sum_{z'} |\Pr[\mathsf{E}'(X) = \mathsf{E}'(t(X)) = z'] - 2^{-n'} \Pr[S^{E'}_{X,t} = \texttt{same}]| - \sum_{z'} 2^{-n'} \Pr[S^{E}_{X,t} = z'] \\ &\qquad + \sum_{z' \neq \tilde{z}'} |\Pr[\mathsf{E}'(X)\mathsf{E}'(t(X)) = z'\tilde{z}'] - \frac{1}{2^{n'}-1} \Pr[S^{E'}_{X,t} = \tilde{z}']| \\ &\geq \sum_{z'} |\Pr[\mathsf{E}'(X) = \mathsf{E}'(t(X)) = z'] - 2^{-n'} \Pr[S^{E'}_{X,t} = \texttt{same}]| \\ &\qquad + \sum_{z' \neq \tilde{z}'} |\Pr[\mathsf{E}'(X)\mathsf{E}'(t(X)) = z'\tilde{z}'] - \frac{1}{2^{n'}-1} \Pr[S^{E'}_{X,t} = \tilde{z}']| - 2^{-n'}. \end{aligned}$$

Or, in other words:

$$2\epsilon + \frac{1}{2^{n'}} \geq \sum_{z'} |\Pr[\mathsf{E}'(X) = \mathsf{E}'(t(X)) = z'] - 2^{-n'} \Pr[S^{E'}_{X,t} = \texttt{same}]| + \sum_{z' \neq \tilde{z}'} |\Pr[\mathsf{E}'(X)\mathsf{E}'(t(X)) = z'\tilde{z}'] - \frac{1}{2^{n'}-1} \Pr[S^{E'}_{X,t} = \tilde{z}']|$$

Next, observe that we can bound the distance between our simulators as follows:

$$
\begin{aligned}
2\varDelta(S_{X,t}; S^{E'}_{X,t}) &= |\Pr[S_{X,t} = \mathtt{same}] - \Pr[S^{E'}_{X,t} = \mathtt{same}]| + \sum_{\tilde{z}'} |\Pr[S_{X,t} = \tilde{z}'] - \Pr[S^{E'}_{X,t} = \tilde{z}']| \\
&= |\Pr[\mathsf{E}'(X) = \mathsf{E}'(t(X))] - \Pr[S^{E'}_{X,t} = \mathtt{same}]| \\
&\quad + \sum_{\tilde{z}'} |\Pr[\mathsf{E}'(X) \neq \mathsf{E}'(t(X)) \wedge \mathsf{E}'(t(X)) = \tilde{z}'] - \Pr[S^{E'}_{X,t} = \tilde{z}']| \\
&= |\sum_{z'} \Pr[\mathsf{E}'(X) = \mathsf{E}'(t(X)) = z'] - \sum_{z'} 2^{-n'} \Pr[S^{E'}_{X,t} = \mathtt{same}]| \\
&\quad + \sum_{\tilde{z}'} |\sum_{z' \neq \tilde{z}'} \Pr[\mathsf{E}'(X)\mathsf{E}'(t(X)) = z'\tilde{z}'] - \sum_{z' \neq \tilde{z}'} \frac{1}{2^{n'}-1} \Pr[S^{E'}_{X,t} = \tilde{z}']| \\
&\leq \sum_{z'} |\Pr[\mathsf{E}'(X) = \mathsf{E}'(t(X)) = z'] - 2^{-n'} \Pr[S^{E'}_{X,t} = \mathtt{same}]| \\
&\quad \sum_{z' \neq \tilde{z}'} |\Pr[\mathsf{E}'(X)\mathsf{E}'(t(X)) = z'\tilde{z}'] - \frac{1}{2^{n'}-1} \Pr[S^{E'}_{X,t} = \tilde{z}']| \\
&\leq 2\epsilon + \frac{1}{2^{n'}}.
\end{aligned}
$$

Thus, we can conclude by post-processing that

$$
\begin{aligned}
\varDelta(\mathsf{E}'(X)\mathsf{E}'(t(X)); U_{n'}\mathrm{Copy}(S_{X,t}, U_{n'})) &\leq \varDelta(\mathsf{E}'(X)\mathsf{E}'(t(X)); U_{n'}\mathrm{Copy}(S^{E'}_{X,t}, U_{n'})) \\
&\quad + \varDelta(U_{n'}\mathrm{Copy}(S^{E'}_{X,t}, U_{n'}); U_{n'}\mathrm{Copy}(S_{X,t}, U_{n'})) \\
&\leq \varDelta(\mathsf{E}'(X)\mathsf{E}'(t(X)); U_{n'}\mathrm{Copy}(S^{E'}_{X,t}, U_{n'})) + \varDelta(S^{E'}_{X,t}; S_{X,t}) \\
&\leq \epsilon + (\epsilon + \frac{1}{2^{n'+1}})
\end{aligned}
$$

□

Note that since post-processing cannot increase statistical distance, Lemma C.8 implies that

$$\forall X \in \mathcal{X}, \forall t \in \mathcal{T}, \quad \varDelta(\mathsf{Trunc}_m(\mathsf{E}'(X)), \mathsf{Trunc}_m(\mathsf{E}'(t(X))); \mathsf{Trunc}_m(U_{n'}), \mathsf{Trunc}_m(\mathrm{Copy}(S_{X,t}, U_{n'})) \leq 2\epsilon + 2^{-(n'+1)},$$

In particular, we will use the fact that for any $z, \tilde{z} \in \{0,1\}^m$

$$|\Pr[\mathsf{Trunc}_m(\mathsf{E}'(X)), \mathsf{Trunc}_m(\mathsf{E}'(t(X))) = z\tilde{z}] - \Pr[\mathsf{Trunc}_m(U_{n'}), \mathsf{Trunc}_m(\mathrm{Copy}(S_{X,t}, U_{n'})) = z\tilde{z}]| \leq 4\epsilon + 2^{-(n')}. \tag{6}$$

C.3 Analysis

Lemma C.9. *For all size N^c recognizable N bit sources, R, with min-entropy $k \geq (1-\gamma)N$ and all size N^c tampering functions t, there exists a simulator S such that for all $z, \tilde{z}$*

$$(1-N^{-c}) \Pr[U_m \mathrm{Copy}(S, U_m)] - 2^{-2m-dm} \leq \Pr[\mathsf{E}(X)\mathsf{E}(t(X)) = z\tilde{z}] \leq (1+N^c) \Pr[U_m \mathrm{Copy}(S, U_m)] + 2^{-2m-dm}.$$

Theorem C.6 is an immediate corollary of this Lemma.

The simulator S is quite simple. Let E' be the same as E except without truncation, namely $\mathsf{E}'(R) := \mathrm{2NMExt}(f(\mathsf{Ext}^1_{\mathrm{samp}}(R)), \mathsf{Ext}^2_{\mathrm{samp}}(R))$. S is simply the canonical simulator for E' (see Lemma C.8) except with any non-$\mathtt{same}$ output truncated to m bits. In other words,

$$S := \left\{ \begin{array}{c} \text{Sample } R \text{ and compute} (z', \tilde{z}') = \mathsf{E}'(R), \mathsf{E}'(t(R)) \\ \text{if } z' = \tilde{z}', \text{ output } \mathtt{same} \\ \text{otherwise, output } \tilde{z} = \mathsf{Trunc}_m(\tilde{z}') \end{array} \right\}.$$

Note that even though S cannot necessarily be sampled by a polysize circuit (because R is a recognizable source), it can be sampled by a polysize NP-circuit.

Proof. Assume for the sake of contradiction that there exists $R, t, z, \tilde{z}$ such that

$$\Pr[\mathsf{E}(R)\mathsf{E}(t(R) = z\tilde{z}] > (1+\alpha)\Pr[U_m \mathrm{Copy}(S, U_m) = z\tilde{z}] + \beta,$$

where $\alpha = N^{-c}$ and $\beta = 2^{-2m-dm}$. Let C be some circuit of size at most N^c such that R is uniform on $\{r : C(r) = 1\}$.

The case that the lower bound is violated follows in the same manner as presented below. Our goal is to construct a Σ_3-circuit of size $O(N^{c_3})$ that has $> \epsilon$ advantage over random guessing in computing f.

Before we continue, introduce some notation for the intermediate random variables involved in evaluating the extractor on the source and the tampered source.

- R is the source and $\tilde{R} := t(R)$, the tampered source.
- $X := \mathsf{Ext}^1_{\text{samp}}(R)$, $\tilde{X} := \mathsf{Ext}^1_{\text{samp}}(R)$.
- $W := \mathsf{Ext}^2_{\text{samp}}(R)$, $\tilde{W} := \mathsf{Ext}^2_{\text{samp}}(R)$.
- $W_x := W | X = x$, $\tilde{W}_x := \tilde{W} | X = x$
- $W_{x,\tilde{x}} := W | X = x \wedge \tilde{X} = \tilde{x}$, $\tilde{W}_{x,\tilde{x}} := W | X = x \wedge \tilde{X} = \tilde{x}$
- $R_x := R | X = x$, $\tilde{R}_x := R | X = x$
- $R_{x,\tilde{x}} := R | X = x, \tilde{X} = x$, $\tilde{R}_{x,\tilde{x}} := R | X = x, \tilde{X} = x$
-

$$S_x := \left\{ \begin{array}{c} \text{Sample } (z', \tilde{z}') \leftarrow \mathsf{E}'(R_x), \mathsf{E}'(\tilde{R}_x) \\ \text{if } z' = \tilde{z}', \text{ output } \texttt{same} \\ \text{otherwise, output } \tilde{z} = \mathsf{Trunc}_m(\tilde{z}') \end{array} \right\} \qquad S_{x,\tilde{x}} := \left\{ \begin{array}{c} \text{Sample } (z', \tilde{z}') \leftarrow \mathsf{E}'(R_{x,\tilde{x}}), \mathsf{E}'(\tilde{R}_{x,\tilde{x}}) \\ \text{if } z' = \tilde{z}', \text{ output } \texttt{same} \\ \text{otherwise, output } \tilde{z} = \mathsf{Trunc}_m(\tilde{z}') \end{array} \right\}.$$

We begin by proving a sequence of simple claims. Loosely, these claims say that there are many inputs, x, to the incomputable function, f, such that conditioning on x in the non-malleable extractor experiment preserves the violation of the relative error guarantee in addition to some other properties. The primary technical work of the proof, the small circuit that computes f and its analysis, can be found after these elementary claims.

We say $x \in \{0,1\}^\ell$ is *useful* if

$$\Pr[\mathsf{E}(R_x)\mathsf{E}(\tilde{R}_x) = z\tilde{z}] > (1+\alpha)\Pr[U_m \mathrm{Copy}(S_x, U_m) = z\tilde{z}] + \beta/2$$

We will show that useful x's are not too sparse. We do so by observing that useful x's are not too sparse relative to X (which is close to uniform in relative error).

Claim. $\Pr[X \text{ is useful}] \geq \beta/2$

$$\begin{aligned}
\beta &< \Pr[\mathsf{E}(R)\mathsf{E}(\tilde{R}) = z\tilde{z}] - (1+\alpha)\Pr[U_m \mathrm{Copy}(S, U_m) = z\tilde{z}] \\
&= \sum_x \Pr[X = x](\Pr[\mathsf{E}(R_x)\mathsf{E}(\tilde{R}_x = z\tilde{z}] - \Pr[U_m \mathrm{Copy}(S_x, U_m) = z\tilde{x}]) \\
&= \sum_{x \text{ useful}} \Pr[X = x](\Pr[\mathsf{E}(R_x)\mathsf{E}(\tilde{R}_x) = z\tilde{z}] - \Pr[U_m \mathrm{Copy}(S_x, U_m) = z\tilde{z}]) \\
&\quad + \sum_{x \text{ not useful}} \Pr[X = x](\Pr[\mathsf{E}(R_x)\mathsf{E}(\tilde{R}_x) = z\tilde{z}] - \Pr[U_m \mathrm{Copy}(S_x, U_m) = z\tilde{z}]) \\
&\leq \Pr[X \text{ is useful}] + \beta/2
\end{aligned}$$

The claim follows.

Claim. $\Pr[U_\ell \text{ is useful}] > \beta/4$

Because $\mathsf{Ext}^1_{\text{samp}}(R)$ is a samplable source extractor for size N^c recognizable sources with relative error α, we have

$$\Pr[X \text{ is useful}] \leq (1+\alpha)\Pr[U_\ell \text{ is useful}].$$

Claim follows by our choice of $\alpha < 1$ and $\Pr[X \text{ is useful}] \geq \beta/2$.

Now, we observe that conditioning on x doesn't effect the entropy of of R too much.

Claim. For all $x \in \{0,1\}^\ell$, $H_\infty(R|X = x) \geq k - (\ell + 1)$.

Notice that for each $r \in \mathsf{Supp}(R|X = x)$, we have

$$\Pr[R = r|X = x] \leq \frac{\Pr[R = r]}{\Pr[X = x]} \leq \frac{2^{-(1-\gamma)N}}{(1+\alpha)2^{-\ell}} < 2^{\ell+1-k}.$$

We say $\tilde{x}$ is *helpful for x* if

1. $\Pr[\mathsf{E}(R)\mathsf{E}(\tilde{R}) = z\tilde{z}|X = x \wedge \tilde{X} = \tilde{x}] \geq (1+\alpha)\Pr[U_m\mathrm{Copy}(S_{x,\tilde{x}}, U_m) = z\tilde{z}] + \beta/8$
2. $H_\infty(R|X = x \wedge \tilde{X} = \tilde{x}) \geq k - 2\ell - 3 - \log(1/\beta)$

Claim. $\Pr[\tilde{X}$ is helpful for $x|X = x$ is useful$] \geq \beta/8$.

Fix any x that is useful and let $H = \{\tilde{x} : \Pr[\tilde{X} = \tilde{x}|X = x] > \beta/2^{\ell+2}\}$. Then we have $\Pr[\tilde{X} \notin H|X = x] \leq 2^\ell \cdot \beta/2^{\ell+3} = \beta/4$, and moreover for any $\tilde{x} \in H$,

$$\Pr[R = r|X = x \wedge \tilde{X} = \tilde{x}] \leq \frac{\Pr[R = r|X = x]}{\Pr[\tilde{X} = \tilde{x}|X = x]} \leq \frac{2^{\ell+1-k}}{\beta/2^{\ell+2}} = 2^{-(k-2\ell-3-\log(1/\beta))}$$

Next, let $G = \{\tilde{x} : \Pr[\mathsf{E}(R_{x,\tilde{x}})\mathsf{E}(\tilde{R}_{x,\tilde{x}}) = z\tilde{z}] > (1+\alpha)\Pr[U_m\mathrm{Copy}(S_{x,\tilde{x}}, U_m) = z\tilde{z}] + \beta/4\}$. Notice that, we can bound

$$\begin{aligned}
\beta/2 &> \Pr[\mathsf{E}(R_x)\mathsf{E}(\tilde{R}_x) = z\tilde{z}] - (1+\alpha)\Pr[U_m\mathrm{Copy}(S_x, U_m) = z\tilde{z}] \\
&= \sum_{\tilde{x} \in G} \Pr[\tilde{X} = x|X = x](\Pr[\mathsf{E}(R_{x,\tilde{x}})\mathsf{E}(\tilde{R}_{x,\tilde{x}}) = z\tilde{z}] - (1+\alpha)\Pr[U\mathrm{Copy}(S_{x,\tilde{x}}, U_m) = z\tilde{z}]) \\
&\quad + \sum_{\tilde{x} \notin G} \Pr[\tilde{X} = x|X = x](\Pr[\mathsf{E}'(R_{x,\tilde{x}})\mathsf{E}'(\tilde{R}_{x,\tilde{x}}) = z\tilde{z}] - (1+\alpha)\Pr[U_m\mathrm{Copy}(S_{x,\tilde{x}}, U_m) = z\tilde{z}]) \\
&\geq \Pr[\tilde{X} \in G|X = x] + \beta/4
\end{aligned}$$

Therefore, $\Pr[\tilde{X} \in G|X = x] > \beta/4$. Finally, the claim follows from

$$\Pr[\tilde{X} \in G \cap H|X = x] \geq \Pr[\tilde{X} \in G|X = x] - \Pr[\tilde{X} \notin H|X = x] \geq \beta/4 - \beta/8.$$

Next we deduce that the "right" input to the 2-source non-malleable extractor maintains high entropy, even after conditioning on any useful "left" inputs and helpful "left" tampered inputs. Looking ahead, this (and the fact that useful and helpful inputs bias the 2-source non-malleable extractor) will allow us to apply the list-decoding guarantee of the 2-source non-malleable extractor.

Claim. For all useful x and $\tilde{x}$ helpful for x, $H_\infty(W_{x,\tilde{x}}) \geq n - 1$.

Recall that $W_{x,\tilde{x}} := (\mathsf{Ext}^2_{\mathrm{samp}}(R)|X = x \wedge \tilde{X} = \tilde{x})$. Or equivalently, $W_{x,\tilde{x}} \equiv \mathsf{Ext}^2_{\mathrm{samp}}(R_{x,\tilde{x}})$, where $R_{x,\tilde{x}} := U|C(U) = 1 \wedge \mathsf{Ext}^1_{\mathrm{samp}}(U) = x \wedge \mathsf{Ext}^1_{\mathrm{samp}}(t(U)) = \tilde{x}$. In particular, this means $R_{x,\tilde{x}}$ is recognizable by a circuit of size $c'N^{c_1}$) for some constant c'. By our previous claims we have that $H_\infty(R_{x,\tilde{x}}) \geq k - 2\ell - 3\log(1/\beta)$. By our choice of parameters, this means $H_\infty(R_{x,\tilde{x}}) \geq (1-\gamma_2)N$ and it follows from the relative error property that for any $w \in \{0,1\}^n$

$$\Pr[\mathsf{Ext}^2_{\mathrm{samp}}(R_{x,\tilde{x}}) = w] \leq (1+\alpha)2^{-n} < 2^{-n+1}.$$

We are finally ready to describe a (randomized) circuit A that attempts to compute $f(x)$, given x:

1. Using NP-oracle sample $\tilde{x} \leftarrow \tilde{X}|X = x$. (If the sampler outputs $\perp$, which happens with probability at most $1/4$, output $y^* = \mathrm{argmax}_y \Pr[f(U_n) = y]$.)
2. Construct the following nondeterministic NP-circuit, $C_{x,\tilde{x}}$, to recognize y's that "explain" $z, \tilde{z}$ well:
   - **Input:** y
   - **Witness:** $\tilde{y}$

- **Computation:** use NP oracle to δ-approximate the following quantities

$$\begin{aligned}
M_E &= |\{r : C(r) = 1, \mathsf{Ext}^1_{\text{samp}}(r) = x, \mathsf{Ext}^1_{\text{samp}}(t(r)) = \tilde{x}, \\
&\qquad \mathsf{Trunc}_m(\text{2NMExt}(y, \mathsf{Ext}^2_{\text{samp}}(r))) = z, \mathsf{Trunc}_m(\text{2NMExt}(\tilde{y}, \mathsf{Ext}^2_{\text{samp}}(t(r)))) = \tilde{z}\}| \\
M_S^{\neq} &= |\{r : C(r) = 1, \mathsf{Ext}^1_{\text{samp}}(r) = x, \mathsf{Ext}^1_{\text{samp}}(t(r)) = \tilde{x}, \\
&\qquad \text{2NMExt}(y, \mathsf{Ext}^2_{\text{samp}}(r)) \neq \text{2NMExt}(\tilde{y}, \mathsf{Ext}^2_{\text{samp}}(t(r))), \mathsf{Trunc}_m(\text{2NMExt}(\tilde{y}, \mathsf{Ext}^2_{\text{samp}}(t(r)))) = \tilde{z}\}| \\
M_S^{=} &= |\{r : C(r) = 1, \mathsf{Ext}^1_{\text{samp}}(r) = x, \mathsf{Ext}^1_{\text{samp}}(t(r)) = \tilde{x}, \text{2NMExt}(y, \mathsf{Ext}^2_{\text{samp}}(r)) = \text{2NMExt}(\tilde{y}, \mathsf{Ext}^2_{\text{samp}}(t(r)))\}| \\
Q &= |\{r : C(r) = 1, \mathsf{Ext}^1_{\text{samp}}(r) = x, \mathsf{Ext}^1_{\text{samp}}(t(r)) = \tilde{x}\}|
\end{aligned}$$

- **Output:** Let $\widehat{M_E}, \widehat{M_S^{\neq}}, \widehat{M_S^{=}}, \widehat{Q}$ denote the approximations of $M_E, M_S^{\neq}, M_S^{=}, Q$ (respectively).
  - If $z \neq \tilde{z}$, accept if and only if
  $$\frac{\widehat{M_E}}{\widehat{Q}} \geq (1 + 5\delta)2^{-m}\frac{\widehat{M_S^{\neq}}}{\widehat{Q}} + \beta/10$$
  - If $z = \tilde{z}$, accept if and only if
  $$\frac{\widehat{M_E}}{\widehat{Q}} \geq (1 + 5\delta)2^{-m}\frac{\widehat{M_S^{\neq}} + \widehat{M_S^{=}}}{\widehat{Q}} + \beta/10$$

3. Use a Σ_3-oracle to sample $y \overset{u}{\leftarrow} \{y : C_{x,\tilde{x}}(y) = 1\}$. (If the sampler outputs $\perp$, which happens with probability at most $1/4$, output $y^* = \operatorname{argmax}_y \Pr[f(U_n) = y]$.)

Claim. If x is useful and $\tilde{x}$ is helpful for x, then $C_{x,\tilde{x}}(f(x)) = 1$ and $|\{y : C_{x,\tilde{x}}(y) = 1\}| \leq 2^{(1-\gamma)n}$.

Before proving this claim, we show it implies our desired contradiction with the incomputability of f: $\Pr[A(x) = f(x)] > -n^{-c} \cdot 2^{-n}$. By Theorem 2.20 and Theorem 2.19, A can be written as a Σ_3-circuit of size $\mathsf{poly}(\ell/\delta) \leq c'' N^{c_3}$ for some constants c'', c_3.

$$\begin{aligned}
\Pr_{x \leftarrow U_\ell}[A(x) = f(x)] &\geq \Pr[\text{sampling doesn't fail}] \cdot \Pr[U_\ell \text{ is useful}] \cdot \Pr[\tilde{X} \text{ is helpful for } x | X = x] \cdot 2^{-(1-\gamma)n} \\
&\geq 1/2 \cdot \beta/4 \cdot \beta/8 \cdot 2^{-(1-\gamma)n} \\
&= 2^{-(1-\gamma)n - 6 - 2\log(1/\beta)}
\end{aligned}$$

By our choice of $6 + 2\log(1/\beta) < \gamma n/2$, it follows that $\Pr[A(x) = f(x)] > 2^{(1-\gamma/2)n}$.

We now prove the claim.

1. We begin by observing that because $\widehat{M_E}, \widehat{M_S^{\neq}}, \widehat{M_S^{=}}, \widehat{Q}$ are δ-approximations of $M_E, M_S^{\neq}, M_S^{=}, Q$ (respectively),
$$\frac{\widehat{M_E}}{\widehat{Q}} \in (1 \pm 2\delta) \Pr[\text{2NMExt}(y, W_{x,\tilde{x}})\text{2NMExt}(\tilde{y}, \tilde{W}_{x,\tilde{x}}) = z\tilde{z}].$$
Let $S'_{x,y,\tilde{x}\tilde{y}}$ denote the random variable distributed according to the following:
$$S'_{x,y,\tilde{x}\tilde{y}} = \left\{ \begin{array}{c} \text{Sample } W_{x,\tilde{x}}, \tilde{W}_{x,\tilde{x}} \text{ and compute}(z', \tilde{z}') = \text{2NMExt}(y, W_{x,\tilde{x}}), \text{2NMExt}(\tilde{y}, \tilde{W}_{x,\tilde{x}}) \\ \text{if } z' = \tilde{z}', \text{ output } \mathtt{same} \\ \text{otherwise, output } \tilde{z} = \mathsf{Trunc}_m(\tilde{z}') \end{array} \right\}.$$
Then similarly,
$$\begin{aligned}
z \neq \tilde{z} &\implies 2^{-m}\frac{\widehat{M_S^{\neq}}}{\widehat{Q}} \in (1 \pm 2\delta) \Pr[U_m \text{Copy}(S'_{x,y,\tilde{x},\tilde{y}}, U_m) = z\tilde{z}] \\
z = \tilde{z} &\implies 2^{-m}\frac{\widehat{M_S^{\neq}} + \widehat{M_S^{=}}}{\widehat{Q}} \in (1 \pm 2\delta) \Pr[U_m \text{Copy}(S'_{x,y,\tilde{x},\tilde{y}}, U_m) = z\tilde{z}]
\end{aligned}$$

So, $C_{x,\tilde{x}}(y) = 1$ if there exists $\tilde{y}$ such that

$$\begin{aligned}(1-2\delta)\Pr[\mathsf{Trunc}_m(\text{2NMExt}(y, W_{x,\tilde{x}}))\mathsf{Trunc}_m(\text{2NMExt}(\tilde{y}, \tilde{W}_{x,\tilde{x}})) = z\tilde{z}] \\ \geq (1+5\delta)(1+2\delta)\Pr[U_m\text{Copy}(S'_{x,y,\tilde{x},\tilde{y}}, U_m) = z\tilde{z}] + \beta/10\end{aligned}$$

Or in other words,

$$\begin{aligned}\Pr[\mathsf{Trunc}_m(\text{2NMExt}(y, W_{x,\tilde{x}}))\mathsf{Trunc}_m(\text{2NMExt}(\tilde{y}, \tilde{W}_{x,\tilde{x}})) = z\tilde{z}] \\ \geq \frac{(1+5\delta)(1+2\delta)}{1-2\delta}\Pr[U_m\text{Copy}(S'_{x,y,\tilde{x},\tilde{y}}, U_m) = z\tilde{z}] + \frac{\beta}{10(1-2\delta)}\end{aligned}$$

Then, for $\delta < 1/10$ we have $\frac{(1+2\delta)(1+5\delta)}{1-2\delta} \leq 1 + 10\delta$[29] and $\frac{\beta}{10(1-2\delta)} \leq \frac{\beta}{8}$. It follows that if there exists $\tilde{y}$ such that

$$\Pr[\mathsf{Trunc}_m(\text{2NMExt}(y, W_{x,\tilde{x}}))\mathsf{Trunc}_m(\text{2NMExt}(\tilde{y}, \tilde{W}_{x,\tilde{x}})) = z\tilde{z}] \geq (1+10\delta)\Pr[U_m\text{Copy}(S'_{x,y,\tilde{x},\tilde{y}}, U_m) = z\tilde{z}] + \frac{\beta}{8}$$

then $C_{x,\tilde{x}}(y) = 1$.
In the other direction,$C_{x,\tilde{x}}(y) = 0$ if for all $\tilde{y}$,

$$\begin{aligned}(1+2\delta)\Pr[\mathsf{Trunc}_m(\text{2NMExt}(y, W_{x,\tilde{x}}))\mathsf{Trunc}_m(\text{2NMExt}(\tilde{y}, \tilde{W}_{x,\tilde{x}})) = z\tilde{z}] \\ < (1+5\delta)(1-2\delta)\Pr[U_m\text{Copy}(S'_{x,y,\tilde{x},\tilde{y}}, U_m) = z\tilde{z}] + \beta/8\end{aligned}$$

Because $\frac{(1+5\delta)(1-2\delta)}{1+2\delta} \geq 1$ for $\delta < 1/10$[30] and $\frac{\beta}{10(1+2\delta)} \geq \beta/12$, we can deduce that if for all $\tilde{y}$ it is the case that

$$\Pr[\mathsf{Trunc}_m(\text{2NMExt}(y, W_{x,\tilde{x}}))\mathsf{Trunc}_m(\text{2NMExt}(\tilde{y}, \tilde{W}_{x,\tilde{x}})) = z\tilde{z}] < \Pr[U\text{Copy}(S'_{x,y,\tilde{x},\tilde{y}}, U) = z\tilde{z}] + \frac{\beta}{12},$$

then $C_{x,\tilde{x}}(y) = 0$.

2. Now, we show that $C_{x,\tilde{x}}(y) = 1$. Because x is useful and $\tilde{x}$ is helpful for x, we have

$$\Pr[\mathsf{E}(R)\mathsf{E}(\tilde{R}) = z\tilde{z} | X = x \wedge \tilde{X} = \tilde{x}] \geq (1+\alpha)\Pr[U_m\text{Copy}(S_{x,\tilde{x}}, U_m) = z\tilde{z}] + \beta/8.$$

   Thus,

$$\begin{aligned}\Pr[\mathsf{Trunc}_m(\text{2NMExt}(f(x), W_{x,\tilde{x}}))\mathsf{Trunc}_m(\text{2NMExt}(f(\tilde{x}), \tilde{W}_{x,\tilde{x}})) = z\tilde{z}] = Pr[\mathsf{E}(R)\mathsf{E}(\tilde{R}) = z\tilde{z} | X = x \wedge \tilde{X} = \tilde{x}] \\ \geq (1+\alpha)\Pr[U_m\text{Copy}(S_{x,\tilde{x}}, U_m) = z\tilde{z}] + \beta/8 \\ = (1+10\delta)\Pr[U_m\text{Copy}(S'_{x,f(x),\tilde{x},f(\tilde{x})}, U_m) = z\tilde{z}] + \frac{\beta}{8}.\end{aligned}$$

3. Finally, we show that $|T| \leq 2^{(1-\gamma)n}$ where $T = \{y : C_{x,\tilde{x}}(y) = 1\}$. Suppose this isn't true for the sake of contradiction. Note that by the above we have that for any such $y \in T$ there must be some $\tilde{y}$ such that

$$\Pr[\mathsf{Trunc}_m(\text{2NMExt}(y, W_{x,\tilde{x}}))\mathsf{Trunc}_m(\text{2NMExt}(\tilde{y}, \tilde{W}_{x,\tilde{x}})) = z\tilde{z}] \geq \Pr[U_m\text{Copy}(S'_{x,y,\tilde{x},\tilde{y}}, U_m) = z\tilde{z}] + \frac{\beta}{12},$$

   Let τ_L be the function that maps y to y^* that maximizes

$$\Pr[\mathsf{Trunc}_m(\text{2NMExt}(y, W_{x,\tilde{x}}))\mathsf{Trunc}_m(\text{2NMExt}(y^*, \tilde{W}_{x,\tilde{x}})) = z\tilde{z}] - \Pr[U_m\text{Copy}(S'_{x,y,\tilde{x},y^*}, U_m) = z\tilde{z}].$$

   Let τ_R be the (randomized) function that maps w to $W^* \equiv (\tilde{W}_{x,\tilde{x}} | W_{x,\tilde{x}} = w)$. Note that the distribution of (τ_L, τ_R) can be written as a convex combination of split-state functions (τ_L, τ_R^i), where i is the randomness

[29] $(1+2\delta)(1+5\delta) < (1+10\delta)(1-2\delta) \iff 1 + 7\delta + 10\delta^2 < 1 + 8\delta - 20\delta^2 \iff 0 < \delta - 10\delta^2 = \delta(1-10\delta)$.

[30] $(1-2\delta)(1+5\delta) \geq 1 + 2\delta \iff 1 + 3\delta - 10\delta^2 \geq 1 + 2\delta \iff 0 \leq \delta - 10\delta^2 = \delta(1-10\delta)$

used for sampling W^*. Let I be the random variable representing this randomness, i.e. $(\tau_L, \tau_R) \equiv (\tau_L, \tau_R^I)$. Moreover, for any $y \in T$, our guarantee above translates to

$$\Pr[\mathsf{Trunc}_m(\text{2NMExt}(y, W_{x,\tilde{x}}))\mathsf{Trunc}_m(\text{2NMExt}(\tau_L(y), \tau_R(W_{x,\tilde{x}}))) = z\tilde{z}] \geq \Pr[U_m\text{Copy}(S'_{x,y,\tilde{x},\tau_L(y)}, U_m) = z\tilde{z}]+\frac{\beta}{12}.$$

Then let $S'_{x,Y,\tilde{x},\tau_L(Y),i}$ denote $S'_{x,y,\tilde{x},\tilde{y}}$ where $\tilde{W}_{x,\tilde{x}}$ is sampled using the randomness i and y is sampled from Y, taken to be uniform over T. We have

$$\begin{aligned}\beta/12 &\leq \Pr[\mathsf{Trunc}_m(\text{2NMExt}(Y, W_{x,\tilde{x}}))\mathsf{Trunc}_m(\text{2NMExt}(\tau_L(Y), \tau_R(W_{x,\tilde{x}}))) = z\tilde{z}] - \Pr[U_m\text{Copy}(S'_{x,Y,\tilde{x},\tau_L(Y)}, U_m) = z\tilde{z}]\\ &= \Pr[I = i](\Pr[\mathsf{Trunc}_m(\text{2NMExt}(Y, W_{x,\tilde{x}}))\mathsf{Trunc}_m(\text{2NMExt}(\tau_L(Y), \tau_R^{i^*}(W_{x,\tilde{x}}))) = z\tilde{z}]\\ &\qquad - \Pr[U_m\text{Copy}(S'_{x,Y,\tilde{x},\tau_L(Y),i}, U_m) = z\tilde{z}])\end{aligned}$$

By an averaging argument, there must some choice of randomness i^* such that

$$\begin{aligned}&\Pr[\mathsf{Trunc}_m(\text{2NMExt}(Y, W_{x,\tilde{x}}))\mathsf{Trunc}_m(\text{2NMExt}(\tau_L(Y), \tau_R^{i^*}(W_{x,\tilde{x}}))) = z\tilde{z}]\\ &\qquad \geq \Pr[U_m\text{Copy}(S'_{x,Y,\tilde{x},\tau_L(Y),i^*}, U_m) = z\tilde{z}] + \beta/12.\end{aligned}$$

However, we can observe that

$$(U_m\text{Copy}(S'_{x,Y,\tilde{x},\tau_L(Y),i^*}, U_m) \equiv (\mathsf{Trunc}_m(U_{n'})\mathsf{Trunc}_m(\text{Copy}(S''_{x,Y,\tilde{x},\tau_L(Y),i^*}, U_{n'})),$$

where $S''_{x,Y,\tilde{x},\tau_L(Y),i^*}$ is the canonical simulator for $(Y, W_{x,\tilde{x}})$ with respect to $(\tau_L, \tau_R^{i^*})$ tampering:

$$S''_{x,Y,\tilde{x}\tau_L(Y)),i^*} = \left\{\begin{array}{c}\text{Sample } y \leftarrow Y, w \leftarrow W_{x,\tilde{x}}\\ \text{Compute}(z', \tilde{z}') = \text{2NMExt}(y, w), \text{2NMExt}(\tau_L(y), \tau_R^{i^*}(w))\\ \text{If } z' = \tilde{z}', \text{ output } \texttt{same}\\ \text{Otherwise, output } \tilde{z}'\end{array}\right\}.$$

However, because x is useful and $\tilde{x}$ is helpful we have that $H_\infty(W_{x,\tilde{x}}) \geq n - 1$. Additionally, because Y is a random variable that is uniformly distributed over T, we have that $H_\infty(Y) \geq (1 - \gamma)n$. Thus, we have contradicted the guarantee of 2NMExt which says, via Lemma C.8 and via (6), that for any $z, \tilde{z} \in \{0, 1\}^m$,

$$\begin{aligned}&|\Pr[\mathsf{Trunc}_m(\text{2NMExt}(Y, W_{x,\tilde{x}}))\mathsf{Trunc}_m(\text{2NMExt}(\tau_L(Y), \tau_R^{i^*}(W_{x,\tilde{x}}))) = z\tilde{z}]\\ &\qquad - \Pr[(\mathsf{Trunc}_m(U_{n'})\mathsf{Trunc}_m(\text{Copy}(S''_{x,Y,\tilde{x},\tau_L(Y),i^*}, U_{n'})) = z\tilde{z}]|\\ &\leq 2 \cdot \beta/30 + 2^{-n'} < \beta/12.\end{aligned}$$

□

C.4 Applications of Relative Error Non-Malleable Extractors

We consider using a non-malleable extractor $\mathsf{E} : \{0, 1\}^n \to \{0, 1\}^m$ with *relative error (α, β)* for a class of recognizable sources $\mathcal{X}$ and tampering family $\mathcal{T}$, to obtain leakage and *tamper* resilient cryptosystems with *negligible* security guarantees.

The high level idea is to store a uniformly random R on a device and use $a = \mathsf{E}(R)$ as the secret key for some symmetric key cryptosystem Π. The attacker is allowed (1) leakage on R with leakage function ℓ, for which the source $X \mid \ell(X) = v$ is contained in $\mathcal{X}$, with overwhelming probability over choice of v;[31] (2) tampering on R with tampering function $t \in \mathcal{T}$; (3) oracle access to *both* Π_a, and Π_b, where $b = \mathsf{E}(t(R))$ is the tampered version of the key (Π_a, Π_b denote fixing the secret key of Π to a or b respectively). The goal is to still guarantee security of the cryptosystem with respect to the *original* key a, despite this stronger adversarial model.

[31] In particular, the class of leakage functions can be circuits of bounded polynomial size with bounded output length. In this case, the source $X \mid \ell(X) = v$ is recognizable by polynomial size circuits and, further, the min-entropy requirement is satisfied with overwhelming probability over choice of v.

More formally, for a particular cryptosystem Π, define $\mathrm{Game}_{(a,b)}$ to be the security game for Π, with an added interface that allows the attacker to make oracle queries to Π under *both* secret keys a and b. For unpredictability games, we define $f(a,b)$ to be the probability the adversary wins $\mathrm{Game}_{(a,b)}$ on input (a,b). For indistinguishability games, we define $f(a,b)$ to be probability the adversary wins $\mathrm{Game}_{(a,b)}$ minus $1/2$ (so $f(a,b)$ can be negative).
We consider a two-phase experiment with an adversary $A = (A_1, A_2)$:

Phase 1:

- R is chosen uniformly at random
- The attacker A_1 chooses leakage function ℓ and receives back $v = \ell(R)$.
- A_1 chooses a tampering function t.
- A_1 outputs some additional state st.
- The output of Phase 1 is $(\ell, t, v, \mathsf{st})$.

Phase 2:

- Phase 2 takes as input $(\ell, t, v, \mathsf{st})$ from Phase 1.
- In the Ideal execution, the experiment samples $(a,b) \sim \mathsf{Ideal}_{\ell,t,v}$. In the Real execution, the experiment samples $(a,b) \sim \mathsf{Real}_{\ell,t,v}$.
- The attacker A_2 takes as input st and participates in security game $\mathrm{Game}_{(a,b)}$
- If the attacker wins in $\mathrm{Game}_{(a,b)}$, Phase 2 outputs 1. Otherwise, Phase 2 outputs 0.

The distributions Ideal and Real are defined as follows:
$\mathsf{Real}_{\ell,t,v}$: Sample R uniformly at random conditioned on $\ell(R) = v$. Output $(a = \mathsf{E}(R), b = \mathsf{E}(t(R)))$. For any pair (a,b), let $p_R(a,b)$ denote the probability of (a,b) under distribution Real.
$\mathsf{Ideal}_{\ell,t,v}$: Let $R(v)$ denote the distribution $R|\ell(R) = v$. Let $S_{R(v),t}$ be the random variable corresponding to the output of the simulator $\mathsf{Sim}_{R(v),t}$ for E, which may depend on $R(v)$ and t. Sample a uniformly at random. Output $(a, \mathrm{Copy}(S_{R(v),t}, a))$. For any pair (a,b), let $p_I(a,b)$ denote the probability of (a,b) under distribution Ideal.

Claim. Asume Π is secure against non-uniform, ppt adversaries (i.e. the standard security notion) then for every (non-uniform) ppt (A_1, A_2) and for every fixed $(\ell, t, v, \mathsf{st})$ outputted in Phase 1, $|E_{(a,b)\sim\mathsf{Ideal}_{\ell,t,v}}[f^{A_2(\mathsf{st})}(a,b)]| \leq \mathrm{negl}(m)$.

Proof Sketch. Assume towards contradiction that there is some $(\ell, t, v, \mathsf{st})$ outputted in Phase 1 such that

$$|E_{(a,b)\sim\mathsf{Ideal}_{\ell,t,v}}[f^{A_2(\mathsf{st})}(a,b)]|$$

is non-negligible. Then we will define an adversary A' and a distribution $\mathcal{D}$ over non-uniform advice ζ, such that the expected advantage of $A'[\zeta]$ in the security game is non-negligible, where the expectation is taken over a random draw from the distribution, random choice of secret key a, and the random coins of the adversary.

The non-uniform advice is $\zeta = (\mathsf{st}, \hat{b})$, where $\hat{b}$ is sampled from the distribution $S_{R(v),t}$.

$A'[\zeta = (\mathsf{st}, \hat{b})]$ behaves as follows:

- Begin the execution of $A_2(\mathsf{st})$.
- If A_2 queries its first oracle (corresponding to Π_a), forward the query to the external challenger and return the response to A_2.
- If A_2 queries its second oracle (corresponding to Π_b), then if $\hat{b} = \mathtt{same}$, forward the query externally and return the response to A_2. Otherwise, evaluate $\Pi_{\hat{b}}$ on the query and return the response to A_2.
- Output whatever A_2 outputs.

Existence of a distribution as above implies that there is a particular setting of the non-uniform advice ζ such that $|\mathbb{E}_{a\sim U_m}[f^{A'[\zeta]}(a)]| \geq |\mathbb{E}_{a\sim U_m,\zeta\sim\mathcal{D}}[f^{A'[\zeta]}(a)]|$, which is, in turn, non-negligible. This contradicts the security of Π against non-uniform adversaries. □

We would now like to switch to the Real distribution and show that for all ppt A_1, A_2, with all but negligible probability over $(\ell, t, v, \mathsf{st})$ outputted in Phase 1, $|\mathbb{E}_{(a,b)\sim\mathsf{Real}_{\ell,t,v}}[f^{A_2(\mathsf{st})}(a,b)]| \leq \mathrm{negl}(m)$. Since with all but negligible probability over the coins of Phase 1, R has high min-entropy conditioned on $\ell(R) = v$, it is sufficient to show that for any $(\ell, t, v, \mathsf{st})$ outputted in Phase 1 for which R has high min-entropy conditioned on $\ell(R) = v$, it is the case that $|\mathbb{E}_{(a,b)\sim\mathsf{Real}}[f^{A_2(\mathsf{st})}(a,b)]|$ is negligible.

Unpredictability Games: For unpredictability games (such as MAC's), security in the Real game follows immediately from the properties of the relative error non-malleable extractor given in Definition C.2, by defining the Event A as the event that the adversary wins $\mathrm{Game}_{(a,b)}$.

Indistinguishability Games: For indistinguishability games we can only show that $|\mathbb{E}_{(a,b)\sim\mathsf{Real}}[f^{A_2(\mathsf{st})}(a,b)]|$ is negligible in the case that the cryptosystem Π satisfies a stronger property. Specifically, we require a type of "square-security" notion [BDK+11, DY13] that says that $\mathbb{E}_{(a,b)\sim\mathsf{Ideal}}[(f^{A_2(\mathsf{st})}(a,b))^2]$ is negligible.

We note that there are natural cryptosystems that achieve this notion, such as CPA-secure symmetric key encryption.

Claim. Asume Π is a CPA-secure symmetric key encryption scheme that is secure against non-uniform, ppt adversaries (i.e. the standard security notion) then for every (non-uniform) ppt (A_1, A_2) and for every fixed $(\ell, t, v, \mathsf{st})$ outputted in Phase 1, $\mathbb{E}_{(a,b)\sim\mathsf{Ideal}_{\ell,t,v}}[(f^{A_2(\mathsf{st})}(a,b))^2] \leq \mathrm{negl}(m)$.

We provide a proof sketch below. See e.g. [DY13] for more details.

Proof Sketch. Towards contradiction, assume that for a fixed setting of $(\ell, t, v, \mathsf{st})$, outputted in Phase 1, $\mathbb{E}_{(a,b)\sim\mathsf{Ideal}_{\ell,t,v}}[(f^{A_2(\mathsf{st})}(a,b))^2]$ is non-negligible. We can redefine $A'[\zeta = (\mathsf{st}, \hat{b})]$ from the proof of the previous claim to run the CPA experiment with $A_2(\mathsf{st})$ *twice* with independent random coins, once with a challenge ciphertext for which A' knows the corresponding message (which it can construct by querying its external encryption oracle) and once on the real challenge (which it received from its external challenger). If A_2 answers correctly in the first run, A' returns its answer in the second run to its external challenger. Otherwise, A' returns the complement of A_2's answer in the second run to its external challenger. The expected advantage of A' is equal to twice the expected *squared* advantage of A_2. Thus, the advantage of A' is non-negligible, which contradicts the security of Π. □

From now on, we replace the notation $f^{A_2(\mathsf{st})}(a,b)$ with $f(a,b)$, $\mathsf{Ideal}_{\ell,t,v}$ with Ideal, and $\mathsf{Real}_{\ell,t,v}$ with Real. In the following, we upperbound $|\mathbb{E}_{(a,b)\sim\mathsf{Real}}[f(a,b)]|$ by an expression that involves $E_{(a,b)\sim\mathsf{Ideal}}[f^2(a,b)]$:

$$
\begin{aligned}
\left|\mathbb{E}_{(a,b)\sim\mathsf{Real}}[f(a,b)]\right| &\le \left|\sum_{(a,b):p_I(a,b)<\beta} p_R(a,b)\cdot f(a,b)\right| + \left|\sum_{(a,b):p_I(a,b)\ge\beta} p_R(a,b)\cdot f(a,b)\right| && (7)\\
&\le \sum_{(a,b):p_I(a,b)<\beta} |p_R(a,b)\cdot f(a,b)| + \left|\sum_{(a,b):p_I(a,b)\ge\beta} p_R(a,b)\cdot f(a,b)\right| && (8)\\
&\le \sum_{(a,b):p_I(a,b)<\beta} p_R(a,b) + \left|\sum_{(a,b):p_I(a,b)\ge\beta} p_R(a,b)\cdot f(a,b)\right| \\
&\le \sum_{(a,b):p_I(a,b)<\beta} ((1+\alpha)p_I(a,b)+\beta) + \left|\sum_{(a,b):p_I(a,b)\ge\beta} p_R(a,b)\cdot f(a,b)\right| && (9)\\
&\le 2^{2m}\cdot(2+\alpha)\beta + \left|\sum_{(a,b):p_I(a,b)\ge\beta} p_R(a,b)\cdot f(a,b)\right| \\
&\le 2^{2m}\cdot(2+\alpha)\beta + \left|\sum_{(a,b):p_I(a,b)\ge\beta} \left(\frac{p_R(a,b)}{\sqrt{p_I(a,b)}}\right)\cdot\left(\sqrt{p_I(a,b)}\cdot f(a,b)\right)\right| \\
&\le 2^{2m}\cdot(2+\alpha)\beta + \sqrt{\sum_{(a,b):p_I(a,b)\ge\beta} \frac{p_R^2(a,b)}{p_I(a,b)}}\cdot\sqrt{\sum_{(a,b):p_I(a,b)\ge\beta} p_I(a,b)\cdot f^2(a,b)} && (10)\\
&\le 2^{2m}\cdot(2+\alpha)\beta + \sqrt{\sum_{(a,b):p_I(a,b)\ge\beta} \frac{(1+\alpha)^2 p_I^2(a,b)+2\beta(1+\alpha)p_I(a,b)+\beta^2}{p_I(a,b)}}\cdot\sqrt{\sum_{(a,b):p_I(a,b)\ge\beta} p_I(a,b)\cdot f^2(a,b)} \\
& && (11)\\
&\le 2^{2m}\cdot(2+\alpha)\beta + \sqrt{(1+\alpha)^2+2^{2m+1}\beta(1+\alpha)+2^m\cdot\beta}\sqrt{\mathbb{E}_{(a,b)\sim\mathsf{Ideal}}[f^2(a,b)]},
\end{aligned}
$$

where (7) and (8) follow from the triangle inequality, (10) follows from Cauchy-Schwartz, (9) and (11) follow from the properties of the relative error non-malleable extractor and non-negativity of $p_I(a,b)$ and $p_R(a,b)$. Finally, setting $\alpha = 1/\mathsf{poly}$, β to be sufficiently small, and under the "square-security" assumption that $E_{(a,b)\sim\mathsf{Ideal}}[f^2(a,b)]$ is negligible, we achieve the desired result.

D Non-Malleable Secret Sharing

In 2018, Goyal and Kumar introduced non-malleable secret-sharing. Recall that a secret sharing scheme for an access structure $\mathcal{A}$, allows one to share a secret to n parties such that any "authorized" set of parties (according to $\mathcal{A}$) can recover the secret, and any unauthorized set of parties learns nothing. To understand what it means for a secret sharing scheme to be non-malleable, consider the following experiment: share a secret, jointly tamper all the shares, reconstruct from the tampered shares of some authorized subset of parties. Loosely speaking, a secret sharing scheme is said to be non-malleable if for any secret, any tampering function, and any authorized set of parties: the output of this tampering experiment should either return the original secret or a value unrelated to the original secret (and which of these two cases occurs should also be independent of the original secret). (See Definition D.3 for details.)

In this section, we observe non-malleable codes for polynomial size circuit tampering imply non-malleable secret sharing for polynomial size circuit tampering for a wide variety of access structures: any access structure that an efficient secret sharing scheme. We additionally show that this connection holds even when the tampering attack and reconstructing parties in the above experiment can be chosen as an arbitrary function (not necessarily efficiently computable) of unauthorized shares, provided the efficient secret sharing scheme satisfies an additional property which holds for any linear secret sharing scheme.

D.1 Secret Sharing and Non-Malleable Secret Sharing

We recall the definition of secret sharing and establish some notation for this section. For more details on secret sharing, see the classic survey due to Amos Beimel [Bei11].

Definition D.1 (Access Structure)**.** *Let $[n]$ be a set of parties. An* access structure *is a monotone collection $\mathcal{A} \subseteq 2^{[n]}$ of non-empty subsets of $[n]$. We say sets in $\mathcal{A}$ are* authorized *and sets not in $\mathcal{A}$ are* unauthorized*.*

Definition D.2 (Secret Sharing)**.** *Let K be a finite set of secrets, $[n]$ a set of parties, and $R_1, \ldots, R_n$ sets of possible shares. A* secret sharing scheme *for an access structure $\mathcal{A}$ is a pair of algorithms* $(\mathsf{Share}, \mathsf{Rec})$ *such that* $\mathsf{Share} : K \to R_1 \times \cdots \times R_n$ *is a randomized algorithm that maps secrets to n-tuples of shares and* $\mathsf{Rec} : \mathcal{A} \times R^* \mapsto K$ *maps authorized sets of shares to secrets.*

- **Correctness:** *For any authorized set $B = \{i_1, \ldots, i_\ell\} \in \mathcal{A}$ and any secret $k \in K$,*

$$\Pr[\mathsf{Rec}(B, s_{i_1}, \ldots, s_{i_\ell}) = k; (s_1, \ldots, s_n) \leftarrow \mathsf{Share}(k)] = 1$$

- **Perfect Privacy:** *For any unauthorized set $T = \{i_1, \ldots, i_\ell\} \notin \mathcal{A}$ and any two secrets $a, b \in K$ the following two distributions are identical:*

$$\{(s_{i_1}, \ldots, s_{i_\ell}) : (s_1, \ldots, s_n) \leftarrow \mathsf{Share}(a)\} \equiv \{(s_{i_1}, \ldots, s_{i_\ell}) : (s_1, \ldots, s_n) \leftarrow \mathsf{Share}(b)\}$$

We say a secret sharing scheme has complexity $s(n)$, if both Share *and* Rec *can be computed by (randomized) circuits of size $s(n)$.*

Let $\mathbb{F}$ be a finite field. A secret sharing scheme is said to be $\mathbb{F}$-linear *if $K = \mathbb{F}$, R_i's are $\mathbb{F}$-vector spaces and* Share *can be expressed as a $\mathbb{F}$-linear function that takes as input (k, r) where k is the secret and r is a uniformly random vector of field elements from $\mathbb{F}$.*

Definition D.3 (Non-Malleable Secret Sharing [GK18a])**.** *A secret sharing scheme for an access structure $\mathcal{A}$ is said to be* ϵ-non-malleable *with respect to a tamper class $\mathcal{F} \subseteq \{R^n \to R^n\}$ if for every $f \in \mathcal{F}$ and every authorized set $B \in \mathcal{A}$, there is a distribution,* $\mathsf{SSim}_{f,T}$ *over $K \cup \{\mathtt{same}\}$ such that for any secret $k \in K$ and every authorized set $B = i_1, \ldots, i_\ell \in \mathcal{A}$ the two distributions are ϵ-close:*

$$\left\{\begin{array}{c} (s_1, \ldots, s_n) \leftarrow \mathsf{Share}(k) \\ (\tilde{s}_1, \ldots, \tilde{s}_n) = f(s_1, \ldots, s_n) \\ \mathsf{Rec}(B, s_{i_1}, \ldots, s_{i_\ell}) = \tilde{k} \\ \textit{output } \tilde{k} \end{array}\right\} \approx_\epsilon \left\{\begin{array}{c} x \leftarrow \mathsf{SSim}_{f,B} \\ \textit{if } x = \mathtt{same} \textit{ output } k, \\ \textit{otherwise output } x \end{array}\right\}$$

D.2 Non-Malleable secret sharing from non-malleable codes

We begin by observing that, in contrast to many tampering classes studied in the literature, non-malleable codes and non-malleable secret sharing schemes for poly-size circuit tampering are effectively equivalent. In particular, if an access structure admits a secret sharing scheme such that sharing and reconstruction are computable by size n^c circuits, then non-malleable codes for size $O(n^c)$ circuit tampering imply non-malleable secret sharing for size $O(n^c)$ circuit tampering! (Conversely, non-malleable secret sharing trivially implies non-malleable codes.) The idea is that if the secret sharing scheme is efficient, then the non-malleable code for this expressive tampering class can in fact handle "tampering attacks" that share, tamper, and then reconstruct. Thus we can simply compose the non-malleable code with the secret sharing scheme to inherit the security properties of the secret sharing scheme *and* the non-malleable code (Theorem D.4). Because efficient secret sharing schemes are known for a wide array of access structures (in particular, any access structure that can be represented by a monotone formula of size n^c), efficient non-malleable secret sharing schemes resilient to polynomial size circuit tampering for all such access structures follow as an immediate corollary.

Theorem D.4. *If* $(\mathsf{Enc}, \mathsf{Dec})$ *is an ϵ-non-malleable code for $s(n)+2s'(n)$-size circuit tampering, and* $(\mathsf{Share}, \mathsf{Rec})$ *is a secret sharing scheme for access structure $\mathcal{A}$ where* Share *and* Rec *have complexity at most $s'(n)$, then* $(\mathsf{Share} \circ \mathsf{Enc}, \mathsf{Dec} \circ \mathsf{Rec})$ *is an ϵ-non-malleable secret sharing scheme for access structure $\mathcal{A}$ with respect to $s(n)$-size circuit tampering.*

Sketch. Privacy follows immediately from the privacy of $(\mathsf{Share}, \mathsf{Rec})$. Correctness follows from the correctness of $(\mathsf{Share}, \mathsf{Rec})$ and $(\mathsf{Enc}, \mathsf{Dec})$.

To see non-malleability, fix any size $s(n)$ tampering attack on the non-malleable code and authorized set $B = i_1, \ldots, i_\ell \in \mathcal{A}$. Now consider the function $\phi_{f,B}(c; r)$ that on input c with random coins r, (1) computes shares $(s_1, \ldots, s_n) = \mathsf{Share}(c; r)$, (2) tampers the shares $(\tilde{s}_1, \ldots, \tilde{s}_n) = f(s_1, \ldots, s_n)$, and (3) outputs $\tilde{c} = \mathsf{Rec}(B, s_{i_1}, \ldots, s_{i_\ell})$.

Because $\mathsf{Share}, \mathsf{Rec}$ admit size $s'(n)$ representations and f admits a size $s(n)$ representation, for any coins r, $\phi_{f,B}(\cdot; r)$ can be computed by circuits of size $s(n) + 2s'(n)$. Moreover, by inspection the non-malleable secret sharing experiment with respect to f, B is identically distributed to the non-malleable code experiment with respect to $\phi_{f,B}(\cdot; r)$ where r is chosen uniformly at random.

Because the non-malleable code experiment with respect to $\phi_{f,B}(\cdot; r)$ for any coins r can be simulated up to distance ϵ with some distribution D_r. It follows that the convex combination of such experiments can be simulated via D', the convex combination of distributions D_r. And hence the non-malleable secret sharing scheme can be simulated by D' up to distance ϵ as well. □

The following is an immediate corollary of the above and Theorem 4.2.

Corollary D.5. *Assuming* E *is hard nondeterministic circuits of exponential size, for any polynomial* $p(n)$ *and any class of access structures* $\mathcal{A}$ *admitting an efficiently computable family of secret sharing schemes, there exists an explicit* $1/p(n)$*-non-malleable secret sharing scheme for access structures* $\mathcal{A}$ *robust against tampering by* $p(n)$*-size circuits.*

D.3 Adaptive Non-Malleable Secret Sharing

We go on to show that by asking slightly more of the underlying secret sharing scheme, this simple construction satisfies a stronger definition of *adaptive* non-malleability [LCG+19b, BCL+20]. In adaptive non-malleability, a computationally unbounded adversary may choose the tampering attack and reconstructing parties as an arbitrary function of the shares of some unauthorized set of parties. (See Definition D.6 for details.) We show that so long the secret sharing scheme admits an efficient procedure for "completing" adversarial views (sampling a secret sharing that is consistent with any specific values for some unauthorized set of shares). Note that any linear secret sharing scheme indeed admits such an efficient "completion" procedure. Therefore, as an immediate corollary we get adaptive non-malleable secret schemes resilient to polynomial size circuit tampering for any access structure with an efficient linear secret scheme (which includes any access structure that can be represented by a monotone formula of size n^c).

We begin by formalizing the definition of non-malleable secret sharing for adversaries that may choose a tampering attack and reconstruction set arbitrarily as a function of some unauthorized set of shares. (Note that this allows such an adversary to tamper the unauthorized shares arbitrarily.)

Definition D.6 (Non-Malleable Secret Sharing for Adaptive Adversaries [LCG+19b, BCL+20])**.** *A secret sharing scheme* $(\mathsf{Share}, \mathsf{Rec})$ *for access structure* $\mathcal{A}$ *is* ϵ-non-malleable with respect to adaptive $\mathcal{F}$ tampering *if for any unauthorized set* $T = \{i_1, \ldots, i_\ell\} \notin \mathcal{A}$ *and any function* $\alpha : R_{i_1} \times \cdots \times R_{i_\ell} \to \mathcal{F} \times \mathcal{A}$*, there exists a distribution* $\mathsf{SSim}_{T,\alpha}$ *such that*

$$\left\{\begin{array}{c} (s_1, \ldots, s_n) \leftarrow \mathsf{Share}(k) \\ (f, B) \leftarrow \alpha(s_{i_1}, \ldots, s_{i_\ell}) \\ (\tilde{s}_1, \ldots, \tilde{s}_n) = f(s_1, \ldots, s_n) \\ \mathsf{Rec}(B, s_{i_1}, \ldots, s_{i_\ell}) = \tilde{k} \\ \textit{output } \tilde{k} \end{array}\right\} \approx_\epsilon \left\{\begin{array}{c} x \leftarrow \mathsf{SSim}_{T,\alpha} \\ \textit{if } x = \mathtt{same} \textit{ output } k, \\ \textit{otherwise output } x \end{array}\right\}$$

Before stating our main lemma we need to define an additional property need from the underlying secret sharing scheme: an efficient procedure for sampling a sharing of a secret consistent with any valid setting of unauthorized shares.

Definition D.7. *A secret sharing scheme* $(\mathsf{Share}, \mathsf{Rec})$ *for access structure* $\mathcal{A}$ *is said to admit* completion with complexity $s(n)$ *if for every unauthorized set of shares* $B = \{i_1, \ldots, i_\ell\} \notin \mathcal{A}$ *and any values* $s'_{i_1}, \ldots, s'_{i_\ell} = s'_B$ *consistent with the support of* Share *(there exists* s'_j *for each* $j \notin B$ *and* $k \in K$ *such that* $s'_1, \ldots, s'_n \in$

$\mathsf{Supp}(\mathsf{Share}(k))$), *there is a distribution* $\mathcal{C}_{s'_B}$ *over circuits of size at most* $s(n)$ *such that for every* $k \in K$ *that identically samples sharing of* k *consistent with the unauthorized values:*

$$\mathcal{C}_{s'_B}(k) \equiv \{(s_1, \ldots, s_n) \leftarrow \mathsf{Share}(k) | \forall i \in B, s_i = s'_i\}$$

We observe the following naive bound on the complexity of completion for any linear secret sharing scheme.

Proposition 6. *Any* $\mathbb{F}$*-linear secret sharing scheme admits completion with complexity* $O(N^2 \mathsf{poly}\log(|\mathbb{F}|))$, *where* N *is the total share size of the scheme (measured in number of field elements).*

Sketch. Recall that due to linearity, Share can be written as Mx where $x = (k, r_1, \ldots, r_m)^\top$ for some $m \times n$ matrix, M, where $r_1, \ldots, r_m$ are independently and uniformly drawn from $\mathbb{F}$.

Thus, any set of unauthorized shares yields an underdetermined linear system. To sample a random consistent sharing, it suffices to find a solution x to this system and then add a random vector from the null space of M.

By preprocessing the linear system into reduced row eschelon form, we can, given any secret k, compute a solution, $x = (k, r)^\top$, to this system with at most N non-zero entries with at most $O(N^2)$ arithmetic operations. Because x is N-sparse (with non-zero locations determined at preprocessing time), we can compute $y = Mx$ in time $O(N^2)$.

Finally, we simply output $y + z$ where z is uniform vector from the nullspace of M (sampled during preprocessing phase). The addition takes $O(N)$ operations.

This gives us a distribution over circuits of size $O(N^2 \mathsf{poly}\log(\mathbb{F}))$, $\mathcal{C}$. Such that for any k, the random variable $\mathcal{C}(k)$ formed by sampling a circuit $C \leftarrow \mathcal{C}$ and outputting $C(k)$ is identically distributed to $\mathsf{Share}(k)$ conditioned on the values of the specified unauthorized shares. □

Now we are ready to prove the main theorem in this section.

Theorem D.8. *If* $(\mathsf{Enc}, \mathsf{Dec})$ *is an* ϵ*-non-malleable code for* $s(n)+2s'(n)$*-size circuit tampering, and* $(\mathsf{Share}, \mathsf{Rec})$ *is a secret sharing scheme for access structure* $\mathcal{A}$ *that admits completion with complexity* $s'(n)$ *and additionally* $(\mathsf{Share}, \mathsf{Rec})$ *have complexity at most* $s'(n)$, *then* $(\mathsf{Share} \circ \mathsf{Enc}, \mathsf{Dec} \circ \mathsf{Rec})$ *is an* ϵ*-non-malleable secret sharing scheme for access structure* $\mathcal{A}$ *with respect to* $s(n)$*-size circuit tampering.*

Before proving we observe the following immediate Corollary of this Theorem, Proposition 6, and Theorem 4.2.

Corollary D.9. *Assuming* E *is hard nondeterministic circuits of exponential size, for any polynomial* $p(n)$ *and any class of access structures* $\mathcal{A}$ *admitting an efficiently computable family of linear secret sharing schemes, there exists an explicit* $1/p(n)$*-non-malleable secret sharing scheme for access structures* $\mathcal{A}$ *robust against tampering by* $p(n)$*-size circuits.*

Proof. Correctness and Privacy follow from Theorem D.4. It suffices to show adaptive non-malleability.

Fix any unauthorized set $T = \{i_1, \ldots, i_\ell\} \subset [n]$ and corresponding adversarial strategy $\alpha : R_{i_1} \times \cdots \times R_{i_\ell} \to \mathcal{F} \times \mathcal{A}$.

By perfect secrecy, we know that the residual distribution of the shares in T, for every secret, is identically distributed to some fixed distribution D_T. So, for any secret k, the output of $\mathsf{Share}(k)$ is identically distributed to first sampling D_T (independently of k) to get share values s_T and only then completing the shares using k by sampling $s_1, \ldots, s_n \leftarrow \mathcal{C}_{s_T}(k)$.

Thus, the non-malleability experiment (sample shares, give unauthorized shares to adversary, α, to select reconstruction set and tampering, then perform perform specied tampering and reconstruction) is identical to the experiment where this alternative sharing procedure is used. Critically this alternative sharing, allows us to sample the adversaries view (when choosing a tamepring and reconstruction set) indepedently of the secret. We will use this fact to generate a distribution over tampering small circuit tampering attacks independently of the non-malleable code experiment.

In particular consider the simulator S' that works as follows (where Sim_τ is the simulator for the NMC $(\mathsf{Enc}, \mathsf{Dec})$ with respect to tampering function τ):

- Sample $s_T \leftarrow D_T$

- Sample $C \leftarrow \mathcal{C}_{s_T}$
- Let $f, B = \alpha(s_T)$
- Define $\tau : x \mapsto \mathsf{Rec}_B(f(C(x)))$
- Sample and output $y \leftarrow \mathsf{Sim}_\tau$.

Because Rec_B and $C(x)$ can always be computed by size $s'(n)$ circuits and f can be computed by a size $s(n)$ circuit, τ can be computed by a circuit of size $2s'(n) + s(n)$ and hence the Sim_τ is well-defined.

We are now prepared to define a sequence of hybrid experiments:

- **Real Experiment** H_0**:**

$$\begin{aligned}
&(s_1, \ldots, s_n) \leftarrow \mathsf{Share}(\mathsf{Enc}(k)) \\
&\alpha((s_T) = (f, B) \\
&\tilde{s}_1, \ldots, \tilde{s}_n = f(s_1, \ldots, s_n) \\
&\tilde{k} = \mathsf{Dec}(\mathsf{Rec}_B(\tilde{s}_B))
\end{aligned}$$

- **Hybrid** H_1**:**

$$\begin{aligned}
&s_T \leftarrow D_T \\
&C \leftarrow \mathcal{C}_{s_T} \\
&(s_1, \ldots, s_n) = C(\mathsf{Enc}(k)) \\
&\alpha((s_T) = (f, B) \\
&\tilde{s}_1, \ldots, \tilde{s}_n = f(s_1, \ldots, s_n) \\
&\tilde{k} = \mathsf{Dec}(\mathsf{Rec}_B(\tilde{s}_B))
\end{aligned}$$

- **Hybrid** H_2**:**

$$\begin{aligned}
&s_T \leftarrow D_T \\
&\alpha((s_T) = (f, B) \\
&C \leftarrow \mathcal{C}_{s_T} \\
&(s_1, \ldots, s_n) = C(\mathsf{Enc}(k)) \\
&\tilde{s}_1, \ldots, \tilde{s}_n = f(s_1, \ldots, s_n) \\
&\tilde{k} = \mathsf{Dec}(\mathsf{Rec}_B(\tilde{s}_B))
\end{aligned}$$

- **Hybrid** H_3**:**

$$\left.\begin{array}{l}
s_T \leftarrow D_T \\
\alpha((s_T) = (f, B) \\
C \leftarrow \mathcal{C}_{s_T} \\
\tau(x) \overset{\text{def}}{=} \mathsf{Rec}_B(f(C(x))_B)
\end{array}\right\} \text{Presampling } \tau$$

$$\left.\begin{array}{l}
c \leftarrow Enc(k) \\
\tilde{c} = \tau(c) \\
\tilde{k} = \mathsf{Dec}(\tilde{c})
\end{array}\right\} \text{NMC Experiment w.r.t. } \tau$$

- **Hybrid** H_4**:**

$$\left.\begin{array}{l}
s_T \leftarrow D_T \\
\alpha((s_T) = (f, B) \\
C \leftarrow \mathcal{C}_{s_T} \\
\tau(x) \overset{\text{def}}{=} \mathsf{Rec}_B(f(C(x))_B)
\end{array}\right\} \text{Presampling } \tau$$

$$\text{Copy}(k, \mathsf{Sim}_\tau) \big\} \text{ NMC Simulation w.r.t. } \tau$$

– **Simulation H_5:**

$$\mathrm{Copy}(k, S')$$

As argued above, $H_0 \equiv H_1 \equiv H_2$. Similarly, one can observe $H_4 \equiv H_5$ (from the definition of S'). The security of the NMC says that if we condition on the presampling of any specific τ^* (and any message k), $\mathrm{Copy}(k, \mathsf{Sim}_{\tau^*})$ is ϵ-close to $\mathsf{Dec}(\tau^*(\mathsf{Enc}(k)))$. It follows from the triangle inequality that

$$H_3 \approx_\epsilon H_4.$$

This completes the proof. □

E Missing Proofs

E.1 Proof of Proposition 1

We restate the proposition for completeness and then present the proof.

Proposition 2. *Let X be a random variable and f a function. Define $Y = f(X)$. For any ϵ and any random variable Y',*

$$\Delta(X; (X|f(X) = Y')) = \Delta(Y; Y').$$

Proof. Let $X' \equiv (X|f(X) = Y')$.

$$\begin{aligned}
\Delta(X; X') = \Delta(XY; X'Y') &= \frac{1}{2}\sum_{x,y} |\Pr[Y = y]\Pr[X = x|Y = y] - \Pr[Y' = y]\Pr[X' = x|Y' = y]| \\
&= \frac{1}{2}\sum_{x,y} |\Pr[Y = y]\Pr[X = x|f(X) = y] - \Pr[Y' = y]\Pr[X' = x|f(X') = y]| \\
&= \frac{1}{2}\sum_{x,y} |\Pr[Y = y]\Pr[X = x|f(X) = y] - \Pr[Y' = y]\Pr[X' = x|f(X) = y]| \\
&= \frac{1}{2}\sum_{x,y} |\Pr[Y = y] - \Pr[Y' = y]|\Pr[X = x|f(X) = y] \\
&= \frac{1}{2}\sum_{y} |\Pr[Y = y] - \Pr[Y' = y]|\sum_{x}\Pr[X = x|f(X) = y] \\
&= \frac{1}{2}\sum_{y} |\Pr[Y = y] - \Pr[Y' = y]| \\
&= \Delta(Y; Y')
\end{aligned}$$

□

E.2 Proof of Lemma A.9

We restate the lemma for completeness and then present the proof.

Lemma 4.9. *If* $(\mathsf{Enc}, \mathsf{Dec})$ *is an α-leakage-resilient augmented-split-state $(2n, k, \epsilon)$-non-malleable code, then it is an α-leakage-resilient augmented-split-state $(2n, k, 2\epsilon)$-alternate-non-malleable code. If* $(\mathsf{Enc}, \mathsf{Dec})$ *is an α-leakage-resilient augmented-split-state $(2n, k, \epsilon)$-alternate-non-malleable code, then it is an α-leakage-resilient augmented-split-state $(2n, k, \epsilon + 2^{-k})$-non-malleable code.*

Proof. Our proof follows closely along the lines of the proof given in the work of Dziembowski et al. [DPW10]. We prove the statements in the presented sequence. Thus, assume that $(\mathsf{Enc}, \mathsf{Dec})$ is an α-leakage-resilient augmented-split-state $(2n, k, \epsilon)$-non-malleable code. By definition, there exists a simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$

s.t. for all α-leaky split-state tampering functions f, g and all $m \in \{0,1\}^k$, $(\mathsf{Sim}_1(f,g), \mathrm{Copy}(\mathsf{Sim}_2(f,g), m)) \approx_\epsilon \mathrm{ANM}_m^{f,g,\mathsf{Enc},\mathsf{Dec}}$, where $\mathrm{ANM}_m^{f,g,\mathsf{Enc},\mathsf{Dec}}$ is defined in Definition A.1. Hence,

$$
\begin{aligned}
\mathrm{AltANM}^{f,g}_{m_0,m_1}(0) &\equiv \left\{\begin{array}{c} c_L, c_R \leftarrow \mathsf{Enc}(m_0), (\tilde{c}_L, \tilde{c}_R, \mathsf{trans}) \leftarrow \langle f(c_L), g(c_R)\rangle, \tilde{m} = \mathsf{Dec}(\tilde{c}) \\ \text{Output } (c_L, \mathsf{trans}, \mathtt{same}) \text{ if } \tilde{m} \in \{m_0, m_1\}, \text{ and } (c_L, \mathsf{trans}, \tilde{m}) \text{ otherwise.} \end{array}\right\} \\
&\approx_\epsilon \left\{\begin{array}{c} (c_L, \mathsf{trans}, \tilde{m}) \leftarrow \mathsf{Sim}(f,g) \\ \text{Output } (c_L, \mathsf{trans}, \mathtt{same}) \text{ if } \tilde{m} \in \{m_0, m_1, \mathtt{same}\}, \text{ and } (c_L, \mathsf{trans}, \tilde{m}) \text{ otherwise.} \end{array}\right\} \\
&\approx_\epsilon \left\{\begin{array}{c} c_L, c_R \leftarrow \mathsf{Enc}(m_1), (\tilde{c}_L, \tilde{c}_R, \mathsf{trans}) \leftarrow \langle f(c_L), g(c_R)\rangle, \tilde{m} = \mathsf{Dec}(\tilde{c}) \\ \text{Output } (c_L, \mathsf{trans}, \mathtt{same}) \text{ if } \tilde{m} \in \{m_0, m_1\}, \text{ and } (c_L, \mathsf{trans}, \tilde{m}) \text{ otherwise.} \end{array}\right\} \\
&\equiv \mathrm{AltANM}^{f,g}_{m_0,m_1}(1).
\end{aligned}
$$

Overall, we get that $\mathrm{AltANM}^{f}_{m_0,m_1}(0) \approx_{2\epsilon} \mathrm{AltANM}^{f}_{m_0,m_1}(1)$, as required. This proves the first part of the statement. For the second part, assume that $(\mathsf{Enc}, \mathsf{Dec})$ is an α-leakage-resilient augmented-split-state $(2n, k, \epsilon)$-alternate-non-malleable code. We construct a simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ s.t.

$$(\mathsf{Sim}_1(f,g), \mathrm{Copy}(\mathsf{Sim}_2(f,g), m)) \approx_{\epsilon + 2^{-k}} \mathrm{ANM}_m^{f,g,\mathsf{Enc},\mathsf{Dec}}.$$

Sim works as follows:

- It samples $m^* \in \{0,1\}^k$ uniformly at random.
- It computes $c_L, c_R \leftarrow \mathsf{Enc}(m^*)$, $(\tilde{c}_L, \tilde{c}_R, \mathsf{trans}) \leftarrow \langle f(c_L), g(c_R)\rangle$, and $\tilde{m} = \mathsf{Dec}(\tilde{c}_L, \tilde{c}_R)$
- It outputs $(c_L, \mathsf{trans}, \mathtt{same})$ if $\tilde{m} = m^*$ and $(\mathsf{trans}, \tilde{m})$ otherwise. (Here, $\mathtt{same}$ or $\tilde{m}$, respectively, corresponds to the output of Sim_2 and trans to the output of Sim_1).

For all m, we have that

$$
\begin{aligned}
(\mathsf{Sim}_1(f,g), \mathrm{Copy}(\mathsf{Sim}_2(f,g), m)) &\equiv \begin{cases} \mathsf{Sim}_1(f,g), \mathsf{Sim}_2(f,g) & \text{if } \mathsf{Sim}_2(f,g) \neq \mathtt{same} \\ \mathsf{Sim}_1(f,g), m & \text{if } \mathsf{Sim}_2(f,g) = \mathtt{same}. \end{cases} \\
&\equiv \left\{\begin{array}{c} m^* \in \{0,1\}^k, (c_L, \mathsf{trans}, \tilde{m}) \leftarrow \mathrm{AltANM}^{f,g}_{m,m^*}(1) \\ \text{Output } (c_L, \mathsf{trans}, m) \text{ if } \tilde{m} = \mathtt{same}, \text{ and } (c_L, \mathsf{trans}, \tilde{m}) \text{ otherwise.} \end{array}\right\} \\
&\approx_\epsilon \left\{\begin{array}{c} m^* \in \{0,1\}^k, (c_L, \mathsf{trans}, \tilde{m}) \leftarrow \mathrm{AltANM}^{f,g}_{m,m^*}(0) \\ \text{Output } (c_L, \mathsf{trans}, m) \text{ if } \tilde{m} = \mathtt{same}, \text{ and } (c_L, \mathsf{trans}, \tilde{m}) \text{ otherwise.} \end{array}\right\} \\
&\approx_{2^{-k}} \left\{\begin{array}{c} (c_L, c_R) \leftarrow \mathsf{Enc}(m), (\tilde{c}_L, \tilde{c}_R, \mathsf{trans}) \leftarrow \langle f(c_L), g(c_R)\rangle, \tilde{m} = \mathsf{Dec}(\tilde{c}_L, \tilde{c}_R) \\ \text{Output } (c_L, \mathsf{trans}, \tilde{m}). \end{array}\right\} \\
&\equiv \mathrm{ANM}_m^{f,g,\mathsf{Enc},\mathsf{Dec}}.
\end{aligned}
$$

The $\approx_{2^{-k}}$ is true because the two experiments in this case are equivalent unless $\tilde{m} = m^*$, where $m^* \in \{0,1\}^k$ is chosen uniformly at random. □