(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

* 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.

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).
Several additional results are included in the full version of this paper [Eprint 2022/070]. First, we 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.
Second, we observe that the fact that our constructions only achieve inverse polynomial (statistical) security is inherent. 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.
Third, we 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.

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:

Unconditionally secure constructions via the probabilistic method. [26, 38] 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^{c4}—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 1.4 for further discussion.)

Computationally secure constructions via strong cryptographic assumptions. [12, 28, 29] 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[5]) require the existence of objects that we currently only know how to provably instantiate with random oracles (e.g. [12] uses P-certificates and [28, 29] uses keyless multi-collision resistant hash functions).

Consequently, these works only yield explicit constructions of non-malleable codes under heuristic assumptions.[6] Additionally, these works fall short of providing statistical security guarantees.

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

[4] If (E, D) is ϵ-non-malleable code for n^c-size tampering, then D is hard-on-average for $n^c - O(n)$ size circuits with respect to the distribution $\mathsf{E}(\mathcal{U}_{\{0,1\}})$, encodings of a random bit. In particular if there exists a small circuit C such that $\Pr[C(\mathsf{E}(\mathcal{U})) = \mathsf{D}(\mathsf{E}(\mathcal{U})) = \mathcal{U}] \geq 1/2 + \epsilon$ then consider the C' that on input c outputs a fixed encoding of 0,c_0, if $C(c) = 1$ and a fixed encoding of 1, c_1 otherwise. Then we have $\Pr[\mathsf{D}(C'(\mathsf{E}(\mathcal{U}))) = 1 - \mathcal{U}] \geq 1/2 + \epsilon$, breaking ϵ-non-malleability.

[5] In addition to a variety of subexponentially secure variants of standard cryptographic assumptions, the work of [28, 29] also crucially requires a specific number-theoretic assumption (the non-uniform subexponential hardness of the repeated squaring assumption), while the work of [12] needs the same derandomization assumption in this work.

[6] E.g. [21] 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.

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 the main results of this work, and additional results included in the full version of our paper [16]. Finally in Section 1.4, 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

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 *iff there exists an assignment,* w*, to the non-deterministic input wires such that the circuit, evaluated deterministically on input* (x, w) *outputs* 1*.*

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 *s.t. 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 nondeterminism 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.*[7]

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*.*

[7] 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.

Hardness assumptions against nondeterministic/NP/Σ_i circuits appear in the literature in various contexts of complexity theory and derandomization [19, 32, 39, 41, 45, 51, 59, 61, 62, 63, 64]. As noted in [7], 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.

We summarize the main results of this paper in Section 1.2, and then discuss additional results contained in the full version [16] in Section 1.3. Briefly, included in this work are new constructions of non-malleable codes (Section 1.2) and non-malleable extractors. Additional results contained in the full version include barriers to achieving negligible security guarantees, circumventing these barriers in a manner that has applications to tamper and leakage resilient cryptography (with negligible security guarantees), and an equivalence between non-malleable codes and non-malleable secret sharing in the context of polynomial size circuit tampering.

1.2 Our Results–included in this work

Non-Malleable Codes Our results are as follows:

Theorem 1.3 (Informal). *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.*

The formal statement and proof of this theorem can be found in Section 3.

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 codeword half. In addition to a 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 $G(s)$. The existence of such PRGs follows from Assumption 2 [50, 46, 51, 61, 62, 7].

Given a (leakage-resilient) split-state non-malleable code, with necessary properties and 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 1.4) 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 (as opposed to in the construction where codewords are sampled as $(G(s), c_R)$), every tampering function (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 (this follows from classical results: $\mathsf{IP}[O(1)] \subseteq \mathsf{AM} \subseteq \mathsf{NP/poly}$ [42, 8, 9, 7]).
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[8] and for tampering with bounded polynomial size circuits. We consider two types of computational sources:

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

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 [64]. In 1986, Levin [52] argued that this class reasonably captures sources arising in nature.[9]

A non-malleable extractor for this class yields non-malleable cryptography resilient to 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.[10]

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 [60] 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). *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.*

The formal statement and proof are left to the full version [16]. A detailed construction and proof sketch for a weaker type of non-malleable extractor can be found in the technical overview (see Section 1.4). The construction and proof sketch in the technical overview contain the main ideas needed for the full result.

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

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

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

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

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 $\mathsf{Ext}_{\mathrm{samp}}(X) = s$. 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), 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.

We obtain a similar result for recognizable sources:

Theorem 1.5 (Informal). *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.*

The formal statement and proof are left to the full version [16]. 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 [7] 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 $O(n^c)$.[11] 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. [15] 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 [15] 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.

1.3 Our Results–included in the full version [16]

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. [7]. 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 the full version [16], we extend the results of Applebaum et al. [7] 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.[12] 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

[11] 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.

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

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 and the corresponding results of Applebaum et al. [7]: 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. [7] are unconditional and rule out even inefficient constructions. Please see the full version for further discussion.

Taken together, the results of Applebaum et al. [7] together with our new results for non-malleable codes in the full version [16], 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 $\mathcal{D}_1$,

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

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

Applebaum et al. [7] 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;[13] (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 [18, 31], 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.

Non-malleable secret sharing and non-malleable codes are equivalent under poly-size 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 [43] 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

[13] In fact, the precise leakage class we can handle is slightly more broad.

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).

In the full version [16], 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.

1.4 Technical Overview

We demonstrate our techniques 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). This simplified case will already provides 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.

A Simple Example: (Relaxed) Seedless Non-Malleable "Extractor" for Uniform Sources First, recall that a *relaxed seedless non-malleable extractor* 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. [14]

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

[14] 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

Strong relaxed two-source non-malleable extractor. Loosely speaking, a function $\text{NMExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ 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

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

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

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

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 $2^{-\Omega(n \log\log(n)/\log(n))}$ [56]. Further, [54] 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. 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.

---

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 $\text{NMExt}(s, x) = b$. Note that the resulting non-malleable code will not have perfect correctness because the rejection sampling procedure might fail.

Indeed, we are interested in exactly such PRGs that remain secure even when given the seed, referred to as "seed-extending" PRGs.[15] 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. [50, 53] They observed that PRG constructions based on Nisan and Wigderson's seminal construction [50] can be made seed-extending. Consequently, many constructions of PRGs for nondeterministic circuits can be made seed extending.

Theorem 1.6 ([50, 46, 51, 61, 62, 7]). *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*.*

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

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

[15] We refer the reader to [51] for further discussion.

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 ($\mathsf{IP}[k] \subseteq \mathsf{AM} \subseteq \mathsf{NP/poly}$ [9, 42]) that a small non-deterministic distinguisher exists.[16]

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 [40] 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 $\tilde{z} = 2\text{NMExt}((\tilde{s}, \tilde{y}), \tilde{x})$ 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 $z \leftarrow 2\text{NMExt}((s, y), x)$ 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

[16] 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.

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:

- $\tau_L^{\tilde{s}} : (s, y) \mapsto (\tilde{s}, \tilde{y})$ 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 deficiency 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:

Non-malleable extractors for samplable/recognizable sources. First, we combine the above construction with a seedless extractor for polynomially samplable (resp. recognizable) sources [64, 7] 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 [20, 47].

The full details of our constructions and their analysis can be found in the full version. Similar to above, we first construct an extractor secure against tampering functions without fixed points (this one by Cheraghchi and Guruswami [27] and first construct an extractor secure against tampering functions without fixed points. Then 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.[17]

Non-malleable code. The above non-malleable extractors suggest an natural path to non-malleable codes. Cheraghchi and Guruswami [27] show that *invertible* 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-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

[17] Cheraghchi and Guruswami [27] showed a similar lemma for the case of split-state tampering.

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.

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*.[18]

"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.

The split-state NMC constructions of [2, 3] satisfy the necessary properties. In the full version [16] we show how the leakage-resilience transformation of Ball et al. [17] yields comparable codes with better leakage parameters. Details of our NMC construction and its analysis are in Section 3. The rate of our code inherits the rate of the NMC of Aggarwal et al. [2], which means that to encode a message of length 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.

1.5 Related Work

Non-malleable extractors and codes. There is by now a large body of work on non-malleable extractors (NME) and non-malleable codes (NMC) resilient against various classes of tampering [34, 33, 55, 56, 2, 1, 24, 11, 13, 17, 10, 5, 4, 48]. In the NMC case, some constructions not included in the list above rely on cryptographic assumptions [14, 6], while others require an untamperable common reference string (CRS) [57, 14]. There has also been much work on variants of NME/NMC [23, 37, 30, 49], as well as a relatively new line of work on a related primitive called non-malleable secret sharing [43, 44]. We restrict our attention to constructions most relevant to the current work, namely, the prior constructions of NMC (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, or (3) circuit depth. Existence

[18] 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.

of non-malleable codes under all of the above types of tampering was initially shown via the probabilistic method in [35] and they can also be constructed efficiently in the random oracle model [35]. 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 [36, 25].

NMC against bounded polynomial sized circuits in the CRS model. Faust et al. [38] 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 $\mathsf{poly}(n)$. 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. [12] presented efficient NMC 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, as well as cryptographic assumptions of the existence of sub-exponentially hard trapdoor permutations and the existence of P-certificates with sub-exponential soundness. The only known instantiation of P-certificates requires assuming soundness of a non-trivial argument system (Micali's CS proofs [58]), 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.

NMC against bounded polynomial depth circuits (unbounded polynomial size) in the standard model. Dachman-Soled et al. [28, 29] constructed NMC 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 [64] 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. [7] showed that the $1/\mathsf{poly}$ error

is somewhat inherent by ruling out black-box reductions in this setting. They 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.

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 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. We take circuits to denote circuits over the standard basis $\{\vee, \wedge, \neg\}$. 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 sufficiently large n.

2.2 Non-malleable codes

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* $s \in \{0,1\}^k$*,* $\Pr[\mathsf{Dec}(\mathsf{Enc}(s)) = s] = 1$*.*

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

The 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 $\frac{k}{n}$. We require split-state codes with special properties.

Leakage-resilience: Alice and Bob perform the split-state tampering, but can communicate a bounded amount before tampering.

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.

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.

Theorem 2.4 ([2, 3, 22, 3]). *There is a constant* $\gamma \in (0,1]$ *such that, there exist efficient* n^γ*-leakage-resilient* $\exp(-n^\Omega(1))$*-augmented-split-state non-malleable codes with special encoding. Moreover, the codewords are length* $(3 + o(1))k$ *where* k *is the message length.*

2.3 Seed-extending pseudorandom generators

Definition 2.5 ([50]). *A function $G : \{0,1\}^\ell \to \{0,1\}^n$ is said to be an* ϵ-pseudorandom generator (PRG) *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 PRG, 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 $G' : \{0,1\}^\ell \to \{0,1\}^{n-\ell}$.*

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. [50], 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 2.6 ([50, 46, 51, 61, 62, 7]). *If E requires exponential size circuits of type $X \in \{\text{deterministic}, \text{nondeterministic}, \mathsf{NP}, \Sigma_i\}$, 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 $X \in \{\text{deterministic}, \text{nondeterministic}, \mathsf{NP}, \Sigma_i\}$.*

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 the full version [16].

3 A Non-Malleable Code for Small Circuit Tampering

Lemma 3.1. *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 3.1 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 Theorem 2.4, and with G given in Theorem 2.6, 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 we obtain the following corollary:

Theorem 3.2. *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 $(1.5 + o(1))k$ that is resilient to $\mathsf{SIZE}[s(n)]$-tampering.*

We note that rate 1 seems quite plausible in this setting.

Figure 3.1: 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 $\mathsf{D}(\tilde{s}, \tilde{x}) = \tilde{m}$.

Proof of Lemma 3.1. 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, $\tau(s, x) = (g(s, x), f(s, x))$.

Claim 3.1. There exists a set Π_Y such that

1. Π_Y is noticeably dense in G: $\Pr_{s \xleftarrow{u} \{0,1\}^\ell}[\mathsf{G}(s) \in \Pi_Y] \geq \epsilon/2$
2. When Merlin is honest, Arthur accepts $(s, y) \in \Pi_Y$ with prob. $> \frac{1+\epsilon/2}{2}$.

Figure 3.2: Interactive Proof for distinguishing G from uniform random

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.

Proof. If the protocol in Figure 3.2 is given inputs from $\mathsf{G}(S) = (S, \mathsf{G}'(S))$ (where $S \equiv \mathcal{U}_\ell$), then upon choice of $b = 1$, Arthur's final message is that of the alternate-non-malleability game, $z \sim \mathrm{AltNM}^\tau_{m_0,m_1}(1)$. 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. From this and a simple combinatorial argument (Proposition 4 in the full version [16]), there exists a set Π_Y s.t. 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 a standard argument (Proposition 2 in the full version [16]), this means that for any $(s, y) \in \Pi_Y$ Merlin guesses b correctly and Arthur accepts with probability $\geq \frac{1+\epsilon/2}{2}$. □

Claim 3.2. There exists a set Π_N such that

1. Π_N is large: $\Pr_{(s,y)\xleftarrow{u}\{0,1\}^{\ell+n}}[(s,y) \in \Pi_N] \geq 1 - 8\delta/\epsilon$
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 3.2).

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. We use the existence an 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 and assume s, y are uniformly distributed. We make some observations about the protocol in this case:

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.

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.

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 (see Lemma A.9 in the full version [16]) 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 a simple Markov argument (Proposition 3 in the full version [16]) there exists a set, Π_N such that $\Pr_{(s,y)\xleftarrow{u}\{0,1\}^{\ell+n}}[(s, y) \in \Pi_N] \geq 1-8\delta/\epsilon$ 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 a standard argument (Proposition 2 in the full version [16]), 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 3.1 and Claim 3.2, 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 classical results (Lemma 2.21 in the full version [16]), 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

1. D. Aggarwal, Y. Dodis, T. Kazana, and M. Obremski. Non-malleable reductions and applications. In R. A. Servedio and R. Rubinfeld, editors, *47th Annual ACM Symposium on Theory of Computing*, pages 459–468, Portland, OR, USA, June 14–17, 2015. ACM Press.
2. D. Aggarwal, Y. Dodis, and S. Lovett. Non-malleable codes from additive combinatorics. *SIAM J. Comput.*, 47(2):524–546, 2018.
3. D. Aggarwal, B. Kanukurthi, S. L. B. Obbattu, M. Obremski, and S. Sekar. Rate one-third non-malleable codes. *IACR Cryptol. ePrint Arch.*, page 1042, 2021.
4. D. Aggarwal and M. 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, Nov. 16–19, 2020. IEEE Computer Society Press.
5. S. Agrawal, D. Gupta, H. K. Maji, O. Pandey, and M. Prabhakaran. Explicit non-malleable codes against bit-wise tampering and permutations. In R. Gennaro and M. 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, Aug. 16–20, 2015. Springer, Heidelberg, Germany.
6. S. Agrawal, D. Gupta, H. K. Maji, O. Pandey, and M. Prabhakaran. A rate-optimizing compiler for non-malleable codes against bit-wise tampering and permutations. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part I*, volume 9014 of *Lecture Notes in Computer Science*, pages 375–397, Warsaw, Poland, Mar. 23–25, 2015. Springer, Heidelberg, Germany.
7. B. Applebaum, S. Artemenko, R. Shaltiel, and G. Yang. Incompressible functions, relative-error extractors, and the power of nondeterministic reductions. *Comput. Complex.*, 25(2):349–418, 2016.
8. L. 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.
9. L. Babai and S. Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988.

10. M. Ball, E. Chattopadhyay, J.-J. Liao, T. Malkin, and L.-Y. Tan. Non-malleability against polynomial tampering. In D. Micciancio and T. Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 97–126, Santa Barbara, CA, USA, Aug. 17–21, 2020. Springer, Heidelberg, Germany.
11. M. Ball, D. Dachman-Soled, S. Guo, T. Malkin, and L.-Y. Tan. Non-malleable codes for small-depth circuits. In M. Thorup, editor, *59th Annual Symposium on Foundations of Computer Science*, pages 826–837, Paris, France, Oct. 7–9, 2018. IEEE Computer Society Press.
12. M. Ball, D. Dachman-Soled, M. Kulkarni, H. Lin, and T. Malkin. Non-malleable codes against bounded polynomial time tampering. In Y. Ishai and V. 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.
13. M. Ball, D. Dachman-Soled, M. Kulkarni, and T. Malkin. Non-malleable codes for bounded depth, bounded fan-in circuits. In M. Fischlin and J.-S. 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.
14. M. Ball, D. Dachman-Soled, M. Kulkarni, and T. Malkin. Non-malleable codes from average-case hardness: AC^0, decision trees, and streaming space-bounded tampering. In J. B. Nielsen and V. Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 618–650, Tel Aviv, Israel, Apr. 29 – May 3, 2018. Springer, Heidelberg, Germany.
15. M. Ball, D. Dachman-Soled, M. Kulkarni, and T. Malkin. Limits to non-malleability. In T. Vidick, editor, *ITCS 2020: 11th Innovations in Theoretical Computer Science Conference*, volume 151, pages 80:1–80:32, Seattle, WA, USA, Jan. 12–14, 2020. LIPIcs.
16. M. Ball, D. Dachman-Soled, and J. Loss. (nondeterministic) hardness vs. non-malleability. *IACR Cryptol. ePrint Arch.*, page 70, 2022.
17. M. Ball, S. Guo, and D. Wichs. Non-malleable codes for decision trees. In A. Boldyreva and D. Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 413–434, Santa Barbara, CA, USA, Aug. 18–22, 2019. Springer, Heidelberg, Germany.
18. B. Barak, Y. Dodis, H. Krawczyk, O. Pereira, K. Pietrzak, F.-X. Standaert, and Y. Yu. Leftover hash lemma, revisited. In P. Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 1–20, Santa Barbara, CA, USA, Aug. 14–18, 2011. Springer, Heidelberg, Germany.
19. B. Barak, S. J. Ong, and S. P. Vadhan. Derandomization in cryptography. In D. Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 299–315, Santa Barbara, CA, USA, Aug. 17–21, 2003. Springer, Heidelberg, Germany.
20. M. Bellare, O. Goldreich, and E. Petrank. Uniform generation of np-witnesses using an np-oracle. *Inf. Comput.*, 163(2):510–526, 2000.
21. N. Bitansky, Y. T. Kalai, and O. Paneth. Multi-collision resistance: a paradigm for keyless hash functions. In I. Diakonikolas, D. Kempe, and M. Henzinger, editors, *50th Annual ACM Symposium on Theory of Computing*, pages 671–684, Los Angeles, CA, USA, June 25–29, 2018. ACM Press.

22. G. Brian, A. Faonio, M. Obremski, M. Simkin, and D. Venturi. Non-malleable secret sharing against bounded joint-tampering attacks in the plain model. In D. Micciancio and T. Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 127–155, Santa Barbara, CA, USA, Aug. 17–21, 2020. Springer, Heidelberg, Germany.
23. E. Chattopadhyay, V. Goyal, and X. Li. Non-malleable extractors and codes, with their many tampered extensions. In D. Wichs and Y. Mansour, editors, *48th Annual ACM Symposium on Theory of Computing*, pages 285–298, Cambridge, MA, USA, June 18–21, 2016. ACM Press.
24. E. Chattopadhyay and X. Li. Non-malleable codes and extractors for small-depth circuits, and affine functions. In H. Hatami, P. McKenzie, and V. King, editors, *49th Annual ACM Symposium on Theory of Computing*, pages 1171–1184, Montreal, QC, Canada, June 19–23, 2017. ACM Press.
25. B. Chen, Y. Chen, K. Hostáková, and P. Mukherjee. Continuous space-bounded non-malleable codes from stronger proofs-of-space. In A. Boldyreva and D. Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 467–495, Santa Barbara, CA, USA, Aug. 18–22, 2019. Springer, Heidelberg, Germany.
26. M. Cheraghchi and V. Guruswami. Capacity of non-malleable codes. In M. Naor, editor, *ITCS 2014: 5th Conference on Innovations in Theoretical Computer Science*, pages 155–168, Princeton, NJ, USA, Jan. 12–14, 2014. Association for Computing Machinery.
27. M. Cheraghchi and V. Guruswami. Non-malleable coding against bit-wise and split-state tampering. In Y. Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 440–464, San Diego, CA, USA, Feb. 24–26, 2014. Springer, Heidelberg, Germany.
28. D. Dachman-Soled, I. Komargodski, and R. Pass. Non-malleable codes for bounded polynomial depth tampering. Cryptology ePrint Archive, Report 2020/776, 2020. `https://eprint.iacr.org/2020/776`.
29. D. Dachman-Soled, I. Komargodski, and R. Pass. Non-malleable codes for bounded parallel-time tampering. In *Annual International Cryptology Conference*, pages 535–565. Springer, Cham, 2021.
30. D. Dachman-Soled, F.-H. Liu, E. Shi, and H.-S. Zhou. Locally decodable and updatable non-malleable codes and their applications. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part I*, volume 9014 of *Lecture Notes in Computer Science*, pages 427–450, Warsaw, Poland, Mar. 23–25, 2015. Springer, Heidelberg, Germany.
31. Y. Dodis and Y. Yu. Overcoming weak expectations. In A. Sahai, editor, *TCC 2013: 10th Theory of Cryptography Conference*, volume 7785 of *Lecture Notes in Computer Science*, pages 1–22, Tokyo, Japan, Mar. 3–6, 2013. Springer, Heidelberg, Germany.
32. A. 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, Oct. 26–29, 2013. IEEE Computer Society Press.
33. S. Dziembowski, T. Kazana, and M. Obremski. Non-malleable codes from two-source extractors. In R. Canetti and J. 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, Aug. 18–22, 2013. Springer, Heidelberg, Germany.

34. S. Dziembowski, K. Pietrzak, and D. Wichs. Non-malleable codes. In A. C.-C. Yao, editor, *ICS 2010: 1st Innovations in Computer Science*, pages 434–452, Tsinghua University, Beijing, China, Jan. 5–7, 2010. Tsinghua University Press.
35. S. Dziembowski, K. Pietrzak, and D. Wichs. Non-malleable codes. *J. ACM*, 65(4):20:1–20:32, 2018.
36. S. Faust, K. Hostáková, P. Mukherjee, and D. Venturi. Non-malleable codes for space-bounded tampering. In J. Katz and H. Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 95–126, Santa Barbara, CA, USA, Aug. 20–24, 2017. Springer, Heidelberg, Germany.
37. S. Faust, P. Mukherjee, J. B. Nielsen, and D. Venturi. Continuous non-malleable codes. In Y. Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 465–488, San Diego, CA, USA, Feb. 24–26, 2014. Springer, Heidelberg, Germany.
38. S. Faust, P. Mukherjee, D. Venturi, and D. Wichs. Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In P. Q. Nguyen and E. 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.
39. U. Feige and C. Lund. On the hardness of computing the permanent of random matrices. *Comput. Complex.*, 6(2):101–132, 1997.
40. O. Goldreich, S. Micali, and A. 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.
41. O. Goldreich and A. 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.
42. S. Goldwasser and M. 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.
43. V. Goyal and A. Kumar. Non-malleable secret sharing. In I. Diakonikolas, D. Kempe, and M. Henzinger, editors, *50th Annual ACM Symposium on Theory of Computing*, pages 685–698, Los Angeles, CA, USA, June 25–29, 2018. ACM Press.
44. V. Goyal and A. Kumar. Non-malleable secret sharing for general access structures. In H. Shacham and A. Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 501–530, Santa Barbara, CA, USA, Aug. 19–23, 2018. Springer, Heidelberg, Germany.
45. D. Gutfreund, R. Shaltiel, and A. Ta-Shma. Uniform hardness versus randomness tradeoffs for arthur-merlin games. *Comput. Complex.*, 12(3-4):85–130, 2003.
46. R. Impagliazzo and A. 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.
47. M. Jerrum, L. G. Valiant, and V. V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, 43:169–188, 1986.
48. B. Kanukurthi, S. L. B. Obbattu, and S. Sekar. Four-state non-malleable codes with explicit constant rate. In Y. Kalai and L. 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, Nov. 12–15, 2017. Springer, Heidelberg, Germany.

49. B. Kanukurthi, S. L. B. Obbattu, and S. Sekar. Non-malleable randomness encoders and their applications. In J. B. Nielsen and V. Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 589–617, Tel Aviv, Israel, Apr. 29 – May 3, 2018. Springer, Heidelberg, Germany.
50. J. Kinne, D. van Melkebeek, and R. Shaltiel. Pseudorandom generators, typically-correct derandomization, and circuit lower bounds. *Comput. Complex.*, 21(1):3–61, 2012.
51. A. R. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31(5):1501–1526, 2002.
52. L. A. Levin. Average case complete problems. *SIAM J. Comput.*, 15(1):285–286, 1986.
53. F. Li and D. Zuckerman. Improved extractors for recognizable and algebraic sources. In D. Achlioptas and L. A. Végh, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA*, volume 145 of *LIPIcs*, pages 72:1–72:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
54. X. Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. *Electron. Colloquium Comput. Complex.*, 23:115, 2016.
55. X. Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In H. Hatami, P. McKenzie, and V. King, editors, *49th Annual ACM Symposium on Theory of Computing*, pages 1144–1156, Montreal, QC, Canada, June 19–23, 2017. ACM Press.
56. X. 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.
57. F.-H. Liu and A. Lysyanskaya. Tamper and leakage resilience in the split-state model. In R. Safavi-Naini and R. Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 517–532, Santa Barbara, CA, USA, Aug. 19–23, 2012. Springer, Heidelberg, Germany.
58. S. Micali. CS proofs (extended abstracts). In *35th Annual Symposium on Foundations of Computer Science*, pages 436–453, Santa Fe, NM, USA, Nov. 20–22, 1994. IEEE Computer Society Press.
59. P. B. Miltersen and N. V. Vinodchandran. Derandomizing arthur-merlin games using hitting sets. *Comput. Complex.*, 14(3):256–279, 2005.
60. R. Shaltiel. Weak derandomization of weak algorithms: Explicit versions of yao's lemma. *Comput. Complex.*, 20(1):87–143, 2011.
61. R. Shaltiel and C. Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *J. ACM*, 52(2):172–216, 2005.
62. R. Shaltiel and C. Umans. Pseudorandomness for approximate counting and sampling. *Comput. Complex.*, 15(4):298–341, 2006.
63. R. Shaltiel and C. Umans. Low-end uniform hardness versus randomness tradeoffs for AM. *SIAM J. Comput.*, 39(3):1006–1037, 2009.
64. L. Trevisan and S. P. Vadhan. Extracting randomness from samplable distributions. In *41st Annual Symposium on Foundations of Computer Science*, pages 32–42, Redondo Beach, CA, USA, Nov. 12–14, 2000. IEEE Computer Society Press.