Electronic Colloquium on Computational Complexity, Report No. 40 (2018)

Non-Malleable Codes for Small-Depth Circuits

Marshall Ball* Dana Dachman-Soled † Siyao Guo‡ Tal Malkin§ Li-Yang Tan¶

February 20, 2018

Abstract

We construct efficient, unconditional non-malleable codes that are secure against tampering functions computed by small-depth circuits. For constant-depth circuits of polynomial size (i.e. AC^0 tampering functions), our codes have codeword length $n = k^{1+o(1)}$ for a k-bit message. This is an exponential improvement of the previous best construction due to Chattopadhyay and Li (STOC 2017), which had codeword length $2^{O(\sqrt{k})}$. Our construction remains efficient for circuit depths as large as $\Theta(\log(n)/\log\log(n))$ (indeed, our codeword length remains $n \leq k^{1+\varepsilon}$), and extending our result beyond this would require separating P from NC^1.

We obtain our codes via a new efficient non-malleable reduction from small-depth tampering to split-state tampering. A novel aspect of our work is the incorporation of techniques from unconditional derandomization into the framework of non-malleable reductions. In particular, a key ingredient in our analysis is a recent pseudorandom switching lemma of Trevisan and Xue (CCC 2013), a derandomization of the influential switching lemma from circuit complexity; the randomness-efficiency of this switching lemma translates into the rate-efficiency of our codes via our non-malleable reduction.

*marshall@cs.columbia.edu, Columbia University. Supported in part by the Defense Advanced Research Project Agency (DARPA) and Army Research Office (ARO) under Contract W911NF-15-C-0236, NSF grants CNS1445424 and CCF-1423306, ISF grant no. 1790/13, the Leona M. & Harry B. Helmsley Charitable Trust, and the Check Point Institute for Information Security. Part of this research was done while visiting the FACT Center at IDC Herzliya. Any opinions, findings and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency, Army Research Office, the National Science Foundation, or the U.S. Government.

†danadach@ece.umd.edu, University of Maryland. Supported in part by an NSF CAREER Award #CNS-1453045, by a research partnership award from Cisco and by financial assistance award 70NANB15H328 from the U.S. Department of Commerce, National Institute of Standards and Technology.

‡s.guo@neu.edu, Northeastern University. Supported by NSF grants CNS1314722 and CNS-1413964

§tal@cs.columbia.edu, Columbia University. Supported in part by the Defense Advanced Research Project Agency (DARPA) and Army Research Office (ARO) under Contract W911NF-15-C-0236, NSF grants CNS1445424 and CCF-1423306, and the Leona M. & Harry B. Helmsley Charitable Trust.

¶liyang@cs.columbia.edu, Toyota Technological Institute. Supported by NSF grant CCF 1563122.

ISSN 1433-8092

1 Introduction

Non-malleable codes were introduced in the seminal work of Dziembowski, Pietrzak, and Wichs as a natural generalization of error correcting codes [DPW10, DPW18]. Non-malleability against a class T is defined via the following "tampering" experiment:

Let $t \in T$ denote an "adversarial channel," i.e. the channel modifies the transmitted bits via the application of t.

1. Encode message m using a (public) randomized encoding algorithm: $c \leftarrow \mathrm{E}(m)$,
2. Tamper the codeword: $\tilde{c} = t(c)$,
3. Decode the tampered codeword (with public decoder): $\tilde{m} = \mathrm{D}(\tilde{c})$.

Roughly, the encoding scheme, (E, D), is non-malleable against a class T, if for any $t \in T$ the result of the above experiment, $\tilde{m}$, is either identical to the original message, or completely unrelated. More precisely, the outcome of a t-tampering experiment should be simulatable without knowledge of the message m (using a special flag "same" to capture the case of unchanged message).

In contrast to error correcting codes, the original message m is only guaranteed to be recovered if no tampering occurs. On the other hand, non-malleability can be achieved against a much wider variety of adversarial channels than those that support error detection/correction. As an example, a channel implementing a constant function (overwriting the codeword with some fixed codeword) is impossible to error correct (or even detect) over, but is non-malleable with respect to any encoding scheme.

Any construction of non-malleable codes must make *some* restriction on the adversarial channel, or else the channel that decodes, modifies the message to a related one, and re-encodes, will break the non-malleability requirement. Using the probabilistic method, non-malleable codes have been shown to exist against any class of functions that is not too large ($|T| \leq 2^{2^{\alpha n}}$ for $\alpha < 1$) [DPW10, CG16]. (Here, and throughout the paper, we use k to denote the length of the message, and n to denote the length of the codeword.) A large body of work has been dedicated to the *explicit* construction of codes for a variety of tampering classes: for example, functions that tamper each half (or smaller portions) of the codeword arbitrarily but independently [DKO13, CG16, CZ14, ADL14, Agg15, Li17, Li18], and tampering by flipping bits and permuting the result [AGM+15].

In this paper, we extend a recent line of work that focuses on explicit constructions of non-malleable codes that are secure against adversaries whose computational strength correspond to well-studied complexity-theoretic classes. Since non-malleable codes for a tampering class T yields lower bounds against T (see Remark 2), a broad goal in this line of work is to construct efficient non-malleable codes whose security (in terms of computational strength of the adversary) matches the current state of the art in computational lower bounds.[1]

Prior work on complexity-theoretic tampering classes. In [BDKM16], Ball et al. constructed efficient non-malleable codes against the class of ℓ-local functions, where each output bit is a function of ℓ input bits, and ℓ can be as large as $\Omega(n^{1-\varepsilon})$ for constant $\varepsilon > 0$.[2] This class can be thought of as NC (circuits of fan-in 2) of almost logarithmic depth, $< (1-\varepsilon)\log n$, and in particular, contains NC^0. In [CL17], Chattopadhyay and Li, using new constructions of non-malleable

[1] In this paper we focus on constructing explicit, unconditional codes; see Section 1.3 for a discussion on a different line of work on *conditional* constructions in various models: access to common reference strings, random oracles, or under cryptographic/computational assumptions.

[2] They give constructions even for $o(n/\log n)$-local tampering, but the code rate is inversely proportional to locality, so the codes become inefficient for this locality.

extractors, gave explicit constructions of non-malleable codes against AC^0 and affine tampering functions. These are the first constructions of information-theoretic non-malleable codes in the standard model where each tampered bit may depend on *all* the input bits. However, their construction for AC^0 circuits has exponentially small rate $\Omega(k/2^{\sqrt{k}})$ (equivalently, codeword length $2^{O(\sqrt{k})}$ for a k-bit message), yielding an encoding procedure that is not efficient.

1.1 This work: Efficient non-malleable codes for small-depth circuits

In this work, we address the main open problem from [CL17]: we give the first explicit construction of non-malleable codes for small-depth circuits achieving polynomial rate:

Theorem 1 (Non-malleable codes for small-depth circuits; informal version)**.** *For any $\delta \in (0,1)$, there is a constant $c \in (0,1)$ such that there is an explicit and efficient non-malleable code that is unconditionally secure against polynomial-size unbounded fan-in circuits of depth $c\log(n)/\log\log(n)$ with codeword length $n = k^{1+\delta}$ for a k-bit message and negligible error.*

Extending Theorem 1 to circuits of depth $\omega(\log(n)/\log\log(n))$ would require separating P from NC^1; see Remark 2. Therefore, in this respect the parameters that we achieve in Theorem 1 bring the security of our codes (in terms of computational strength of the adversary) into alignment with the current state of the art in circuit lower bounds.[3]

For the special case of AC^0 circuits, our techniques lead to a non-malleable code with sub-polynomial rate (indeed, we achieve this for all depths $o(\log(n)/\log\log(n))$):

Theorem 2 (Non-malleable codes for AC^0 circuits; informal version)**.** *There is an explicit and efficient non-malleable code that is unconditionally secure against AC^0 circuits with codeword length $n = k^{1+o(1)}$ for a k-bit message and negligible error.*

Prior to our work, there were no known constructions of polynomial-rate non-malleable codes even for depth-2 circuits (i.e. polynomial-size DNF and CNF formulas).

We describe our proof and the new ideas underlying it in Section 1.2. At a high level, we proceed by designing a new efficient *non-malleable reduction* from small-depth tampering to split-state tampering. Our main theorem thus follows by combining this non-malleable reduction with the best known construction of split-state non-malleable codes [Li18].

The flurry of work on non-malleable codes has yielded many surprising connections to other areas of theoretical computer science, including additive combinatorics [ADKO15], two-source extractors [Li12, Li13, CZ16], and non-malleable encryption/commitment [CMTV15, CDTV16, GPR16]. As we discuss in Section 1.2, our work establishes yet another connection—to techniques in unconditional derandomization. While we focus exclusively on small-depth adversaries in this work, we are optimistic that the techniques we develop will lead to further work on non-malleable codes against other complexity-theoretic tampering classes (see Remark 3 for a discussion on the possible applicability of our techniques to other classes).

Remark 1 (On the efficiency of non-malleable codes)**.** A few previous works on non-malleable codes use a non-standard definition of efficiency, only requiring encoding/decoding to take time that is polynomial in the length of the codeword (namely, the output of the encoding algorithm), thus allowing a codeword and computational complexity that is super-polynomial in the message length. In contrast, we use the standard definition of efficiency—running time that is polynomial

[3] Although [CL17] state their results in terms of AC^0 circuits, an inspection of their proof shows that their construction also extends to handle circuits of depth as large as $\Theta(\log(n)/\log\log(n))$. However, for such circuits their codeword length becomes $2^{O(k/\log(k))}$.

in the length of the input. While the non-standard definition is appropriate in some settings, we argue that the standard definition is the right one in the context of non-malleable codes. Indeed, many error-correcting codes in the literature fall under the category of *block codes*—codes that act on a block of k bits of input data to produce n bits of output data, where n is known as the block size. To encode messages m with length greater than k, m is split into blocks of length k and the error-correcting code is applied to each block at a time, yielding a code of rate k/n. For block codes, the block size n can be fixed first and then k can be set as a function of n. A non-malleable code, however, cannot be a block code: If m is encoded block-by-block, the tampering function can simply "destroy" some blocks while leaving the other blocks untouched, thus breaking non-malleability. Instead, non-malleable codes take the entire message m as input and encodes it in a single shot. So in the non-malleable codes setting, we must assume that k is fixed first and that n is set as a function of k. Thus, in order to obtain efficient codes, the parameters of the code must be polynomial in terms of k.

Remark 2 (On the limits of extending our result)**.** Because any function in NC^1 can be computed by a polynomial-size unbounded fan-in circuit of depth $O(\log(n)/\log\log(n))$ (see e.g. [KPPY84, Val83]), any non-trivial non-malleable code for larger depth circuits would yield a separation of NC^1 from P. Here, we take non-trivial to mean that error is bounded away from 1 and encoding/decoding run in time polynomial in the *codeword* length (namely, even an inefficient code, as per the discussion above, can be non-trivial). This follows from the fact (noted in many previous works) that any explicit, non-trivial code is vulnerable to the simple P-tampering attack: decode, flip a bit, re-encode. Hence, in this respect Theorem 1 is the limit of what we can hope to establish given the current state of the art in circuit and complexity theory.

1.2 Our Techniques

At a high level, we use the *non-malleable reduction* framework introduced by Aggarwal et al. [ADKO15]. Loosely speaking, an encoding scheme (E, D) non-malleably reduces a "complex" tampering class, $\mathcal{F}$, to a "simpler" tampering class, $\mathcal{G}$, if the tampering experiment (encode, tamper, decode) behaves like the "simple" tampering (for any $f \in \mathcal{F}$, $\mathrm{D}(f(\mathrm{E}(\cdot))) \approx G_f$, a distribution over $\mathcal{G}$). [ADKO15] showed that a non-malleable code for the simpler $\mathcal{G}$, when concatenated with an (inner) non-malleable reduction (E, D) from $\mathcal{F}$ to $\mathcal{G}$, yields a non-malleable code for the more "complex" $\mathcal{F}$. (See Remark 4 for a comparison of our approach to that of [CL17].)

Our main technical lemma is a new non-malleable reduction from small-depth tampering to *split-state* tampering, where left and right halves of a codeword may be tampered arbitrarily, but independently. We achieve this reduction in two main conceptual steps. We first design a non-malleable reduction from small-depth tampering to a variant of local tampering that we call leaky local, where the choice of local tampering may depend on leakage from the codeword. This step involves a careful design of pseudorandom restrictions with extractable seeds, which we use in conjunction with the pseudorandom switching lemma of Trevisan and Xue [TX13] to show that small-depth circuits "collapse" to local functions under such restrictions. In the second (and more straightforward) step, we reduce leaky-local tampering to split-state tampering using techniques from [BDKM16]. We now describe both steps in more detail.

Small-Depth Circuits to Leaky Local Functions. To highlight some of the new ideas underlying our non-malleable reduction, we first consider the simpler case of reducing w-DNFs (each clause contains at most w literals) to the family of leaky local functions. The reduction for general small-depth circuits will follow from a recursive composition of this reduction.

A non-malleable reduction (E, D) reducing DNF-tampering to (leaky) local-tampering needs to satisfy two conditions (i) $\Pr[\mathrm{D}(\mathrm{E}(x)) = x] = 1$ for any x and, (ii) $\mathrm{D} \circ f \circ \mathrm{E}$ is a distribution over (leaky) local functions for any width-w DNF f. A classic result from circuit complexity, the switching lemma [FSS84, Ajt89, Yao85, Hås86], states that DNFs collapse to local functions under fully random restrictions ("killing" input variables by independently fixing them to a random value with some probability).[4] Thus a natural choice of E for satisfying (ii) is to simply sample from the generating distribution of restrictions and embed the message in the surviving variable locations (fixing the rest according to restriction). However, although $f \circ \mathrm{E}$ becomes local, it is not at all clear how to decode and fails even (i). To satisfy (i), a naive idea is to simply append the "survivor" location information to the encoding. However, this is now far from a fully random restriction (which requires among other things that the surviving variables are chosen independently of the random values used to fix the killed variables) is no longer guaranteed to "switch" the DNFs to Local functions with overwhelming probability.

To overcome these challenges, we employ *pseudorandom switching lemmas*, usually arising in the context of unconditional derandomization, to relax the stringent properties of the distribution of random restrictions needed for classical switching lemmas. In particular, we invoke a recent pseudorandom switching lemma of Trevisan and Xue [TX13], which reduces DNFs to local functions (with parameters matching those of [Hås86]) while only requiring that randomness specifying survivors and fixed values be σ-wise independent[5]. This allows us to avoid problems with independence arising in the naive solution above. Now, we can append a σ-wise independent encoding of the (short) random seed that specifies the surviving variables. This gives us a generating distribution of random restrictions such that (a) DNFs are switched to Local functions, and (b) the seed can be decoded and used to extract the input locations.

At this point, we can satisfy (i) easily: D decodes the seed (whose encoding is always in, say, the first m coordinates), then uses the seed to specify the surviving variable locations and extract the original message. In addition to correctness, $f \circ \mathrm{E}$ becomes a distribution over local functions where the distribution only depends on f (not the message). However, composing D with $f \circ \mathrm{E}$ induces dependence on underlying message: tampered encoding of the seed, may depend on the message in the survivor locations. The encoded seed is comparatively small and thus (assuming the restricted DNF collapses to a local function) requires a comparatively small number of bits to be leaked from the message in order to simulate the tampering of the encoded seed. Given a well simulated seed we can accurately specify the local functions that will tamper the input (the restricted DNFs whose output locations coincide with the survivors specified by the tampered seed). This is the intermediate leaky local tampering class we reduce to, which can be described via the following adversarial game: (1) the adversary commits to N local functions, (2) the adversary can select m of the functions to get leakage from, (3) the adversary then selects the actual tampering function to apply from the remaining local functions.

To deal with depth d circuits, we recursively apply this restriction-embedding scheme d times. Each recursive application allows us to trade a layer of gates for another (adaptive) round of m bits of leakage in the leaky local game. One can think of the recursively composed simulator as applying the composed random restrictions to collapse the circuit to local functions and then, working inwardly, sampling all the seeds and the corresponding survivor locations until the final survivor locations can be used to specify the local tampering.

[4]The switching lemma actually shows that DNFs become *small-depth decision trees* under random restrictions. However, it is this (straightforward) consequence of the switching lemma that we will use in our reduction.

[5]Although this is not stated explicitly in [TX13], as we show, it follows immediately by combining their main lemma with results on bounded independence fooling CNF formulas [Baz09, Raz09].

Leaky Local Functions to Split State. Ball et al. [BDKM16] gave non-malleable codes for local functions via a non-malleable reduction to split state. We make a simple modification to a construction with deterministic decoding from the appendix of the paper to show leaky local functions (the class specified by the above game) can be reduced to split state.

Loosely, we can think of the reduction in the following manner.

First, the left and right states are given leakage-resilient properties via σ_L-wise and σ_R-wise independent encodings. These encodings have the property that any small set (here, a constant fraction of the length of the encoding) of bits will be uniformly distributed, regardless of the message inside. This will allow us, in some sense, to leak bits from the underlying encoding to (a) specify the local tampering functions, and (b) aid in subsequent stages of the reduction.

Second, we take the right encoding to be much longer than the left encoding. Because the tampering will be local, this means that the values of the bits on the right used to tamper the left encoding will be uniformly distributed, regardless of the message. This follows from the fact that there aren't too many such bits relative to the length of the right, given that there significantly fewer output bits on the left and these outputs are each dependent on relatively few bits in general.

Third, we embed the left encoding pseudorandomly in a string that is much longer than the right encoding. This means that with overwhelming probability the bits of the left encoding that affect the tampering of the right will be uniformly distributed. (The rest we can take to be uniformly distributed as well.) Note that although here we use a σ-wise independent generator, an unconditional PRG for small space, as is used in [BDKM16], would have worked as well.

Finally, we prepend to the embedding itself, the short seed used to generate the embedding, after encoding it in a leakage resilient manner (as above). (This is in fact the only significant difference with construction in [BDKM16].) The presence of the seed allows us to determine the embedding locations in the absence of tampering and simulate the embedding locations in the presence of tampering without violating the leakage-resilient properties of the left and right state encodings. The leakage-resilience of the seeds encoding allows a simulator to sample the seed after leaking bits to specify a local tampering.

Remark 3 (On the possible applicability of our techniques to other tampering classes)**.** While we focus exclusively on on small-depth adversaries in this work, we remark that analogous pseudorandom switching lemmas have been developed for many other function classes in the context of unconditional derandomization: various types of formulas and branching programs [IMZ12], low sensitivity functions [HT18], read-once branching programs [RSV13, CHRT18] and CNF formulas [GMR$^+$12], sparse $\mathbb{F}_2$ formulas [ST18], etc. In addition to being of fundamental interest in complexity theory, these function classes are also natural tampering classes to consider in the context of non-malleable codes, as they capture basic types of computationally-bounded adversaries. We are optimistic that the techniques we develop in this paper—specifically, the connection between pseudorandom switching lemmas and non-malleable reductions, and the new notion of pseudorandom restrictions with extractable seeds—will lead to constructions of efficient non-malleable codes against other tampering classes, and we leave this is an interesting avenue for future work.

Remark 4 (Relation to the techniques of [CL17].)**.** Although Chattopadhyay and Li [CL17] also use the switching lemma in their work, our overall approach is essentially orthogonal to theirs. At a high level, [CL17] uses a framework of Cheraghchi and Guruswami [CG16] to derive non-malleable codes from *non-malleable extractors*. In this framework, the rate of the code is directly tied to the error of the extractor; roughly speaking, as the parameters of the switching lemma can be at best inverse-quasipolynomial when reducing to local functions, this unfortunately translates (via the [CG16] framework) into codes with at best exponentially small rate (see pg. 10 of [CL17] for a discussion of this issue). Circumventing this limitation therefore necessitates a significantly

different approach, and indeed, as discussed above we construct our non-malleable codes without using extractors as an intermediary. (On a more technical level, we remark that [CL17] uses the classic switching lemma of Håstad [Hås86] for fully random restrictions, whereas our work employs a recent extension of this switching lemma to pseudorandom restrictions [TX13].)

1.3 Related Work

Non-malleable codes were introduced by Dziembowski, Pietrzak, and Wichs [DPW10, DPW18]. Various subsequent works re-formulated the definition [ADKO15], or considered extensions of the notion [FMNV14, DLSZ15, CGL16, CGM+16]. The original work of [DPW10] presented a construction of non-malleable codes against bit-wise tampering, and used the probabilistic method to prove the existence of non-malleable codes against tampering classes $\mathcal{F}$ of bounded size (this result gives rise to constructions for the same tampering classes $\mathcal{F}$ in the random oracle model). A sequence of works starting from the work of Liu and Lysyanskaya [LL12] presented constructions of non-malleable codes secure against split-state tampering. The original work and some subsequent works [AAG+16, KLT16] required an untamperable common reference string (CRS) and/or computational assumptions. Other works removed these restrictions and achieved unconditionally non-malleable codes against split-state tampering with no CRS [ADL14, ADKO15, Li17, Li18]. Among these works, the construction of Li [Li18] currently achieves the best rate of $\Omega(\log\log n/\log n)$ for two states. Constructions requiring more than two split-states, and which achieve constant rate, were also given in [CZ14, KOS14].

Conditional results on complexity-based tampering. In this paper we work within the standard model and focus on explicit, *unconditional* non-malleable codes. A variety of non-malleable codes against complexity-based tampering classes have been constructed in other models. These constructions require either common randomness (CRS), access to a public random oracle, and/or computational/cryptographic assumptions.

Faust et al. [FMVW14] presented an efficient non-malleable code, in the CRS model, against tampering function families $\mathcal{F}$ of bounded size, improving upon the original work of [DPW10]. Since the size of the CRS grows with the size of the function family, this approach cannot be used to obtain efficient constructions of non-malleable codes against tampering classes that contain circuits of unbounded polynomial size (e.g., AC^0 circuits). Cheraghchi and Guruswami [CG16] in an independent work showed the existence of unconditionally secure non-malleable codes (with no CRS) against tampering families $\mathcal{F}$ of bounded size via a randomized construction. However their construction is inefficient for negligible error (and also does not apply to AC^0 due to the requirement of bounded size).

Faust et al. [FHMV17] gave constructions of (a weaker notion of) non-malleable codes against space-bounded tampering in the random oracle model.

In very recent work, Ball et al. [BDKM17] presented a general framework for converting average-case bounds for a class C into efficient non-malleable codes against the same class C in the CRS model and under cryptographic assumptions. Among several applications of their framework, they give a construction of non-malleable codes against AC^0 tampering circuits in the CRS model under these assumptions (in fact, circuits of depth up to $\Theta(\log(n)/\log\log(n))$, like in our work). In contrast, our constructions are unconditional.

2 Preliminaries

2.1 Basic Notation

For a positive integer n, let $[n]$ to denote $\{1, \ldots, n\}$. For $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, $\|x\|_0$ denotes the number of 1's in x. For $i \leq j \in [n]$, we define $x_{i:j} := (x_i, \ldots, x_j)$. For a set $S \subseteq [n]$, x_S denotes the projection of x to S. For $S \in [n]^m$, $x_S := (x_{S_1}, \ldots, x_{S_m})$. For $x, y \in \{0,1\}^n$, if they disagree on at least $\varepsilon \cdot n$ indices, we say they are ε-far, otherwise, they are ε-close to each other.

For a set Σ, we use Σ^Σ to denote the set of all functions from Σ to Σ. Given a distribution $\mathcal{D}$, $z \leftarrow \mathcal{D}$ denotes sample z according to $\mathcal{D}$. For two distributions $\mathcal{D}_1, \mathcal{D}_2$ over Σ, their statistical distance is defined as $\Delta(\mathcal{D}_1, \mathcal{D}_2) := \frac{1}{2} \sum_{z \in \Sigma} |\mathcal{D}_1(z) - \mathcal{D}_2(z)|$.

We say $g(n) = \tilde{O}(f(n))$ if $g(n) = O(n^\varepsilon f(n))$ for all $\varepsilon > 0$.

2.2 Non-malleable Reductions and Codes

Definition 1 (Coding Scheme)**.** [DPW10] A *Coding scheme*, (E, D), consists of a randomized encoding function $\mathrm{E}\colon \{0,1\}^k \mapsto \{0,1\}^n$ and a decoding function $\mathrm{D}\colon \{0,1\}^n \mapsto \{0,1\}^k \cup \{\perp\}$ such that $\forall x \in \{0,1\}^k, \Pr[\mathrm{D}(\mathrm{E}(x)) = x] = 1$ (over randomness of E).

Non-malleable codes were first defined in [DPW10]. Here we use a simpler, but equivalent, definition based on the following notion of non-malleable reduction by Aggarwal et al. [ADKO15].

Definition 2 (Non-Malleable Reduction)**.** [ADKO15] Let $\mathcal{F} \subset A^A$ and $\mathcal{G} \subset B^B$ be some classes of functions. We say $\mathcal{F}$ *reduces to* $\mathcal{G}$, $(\mathcal{F} \Rightarrow \mathcal{G}, \varepsilon)$, if there exists an efficient (randomized) encoding function $\mathrm{E} : B \to A$, and an efficient decoding function $\mathrm{D} : A \to B$, such that

(a) $\forall x \in B, \Pr[\mathrm{D}(\mathrm{E}(x)) = x] = 1$ (over the randomness of E).

(b) $\forall f \in \mathcal{F}, \exists G$ s.t. $\forall x \in B$, $\Delta(\mathrm{D}(f(\mathrm{E}(x))); G(x)) \leq \varepsilon$, where G is a distribution over $\mathcal{G}$ and $G(x)$ denotes the distribution $g(x)$, where $g \leftarrow G$.

If the above holds, then (E, D) is an $(\mathcal{F}, \mathcal{G}, \varepsilon)$*-non-malleable reduction.*

Definition 3 (Non-Malleable Code)**.** [ADKO15] Let NM_k denote the set of *trivial manipulation functions* on k-bit strings, consisting of the identity function $\mathrm{id}(x) = x$ and all constant functions $f_c(x) = c$, where $c \in \{0,1\}^k$.

A coding scheme (E, D) defines an $(\mathcal{F}_{n(k)}, k, \varepsilon)$*-non-malleable code*, if it defines an $(\mathcal{F}_{n(k)}, \mathrm{NM}_k, \varepsilon)$-non-malleable reduction.

Moreover, the rate of such a code is taken to be $k/n(k)$.

The following useful theorem allows us to compose non-malleable reductions.

Theorem 3 (Composition)**.** *[ADKO15] If* $(\mathcal{F} \Rightarrow \mathcal{G}, \varepsilon_1)$ *and* $(\mathcal{G} \Rightarrow \mathcal{H}, \varepsilon_2)$*, then* $(\mathcal{F} \Rightarrow \mathcal{H}, \varepsilon_1 + \varepsilon_2)$*.*

2.3 Tampering Function Families

2.3.1 Split-State and Local Functions

Definition 4 (Split-State Model)**.** [DPW10] The *split-state model*, SS_k, denotes the set of all functions:

$$\{f = (f_1, f_2) : \ f(x) = (f_1(x_{1:k}) \in \{0,1\}^k, f_2(x_{k+1:2k}) \in \{0,1\}^k) \text{ for } x \in \{0,1\}^{2k}\}.$$

Theorem 4 (Split-State NMC)**.** *[Li18] For any $n \in \mathbb{N}$, there exists an explicit, efficient non-malleable code in the 2-split-state model (SS_n) with rate $k/n = \Omega(\log\log n/\log n)$ and error $2^{-\Omega(k)}$*

Definition 5 (Local Functions)**.** Let $f : \{0,1\}^n \to \{0,1\}^m$ be a function. We say output j of f *depends* on input i if there exists $x, x' \in \{0,1\}^n$ that differ only in the ith coordinate such that $f(x)_j \neq f(x')_j$. We say f is ℓ-local or in the class Local^ℓ, if every output bit f_j depends on at most ℓ input bits.

2.3.2 Small-Depth Circuits and Decision Trees

Let $\mathrm{AC}_d(S)$ denote alternating depth d circuits of size at most S with unbounded fan-in. Let w-$\mathrm{AC}_d(S)$ denote alternating depth d circuits of size at most S with fan-in at most w at the first level and unbounded fan-in elsewhere. For depth 2 circuits, a DNF is an OR of ANDs (terms) and a CNF is an AND or ORs (clauses). The *width* of a DNF (respectively, CNF) is the maximum number of variables that occur in any of its terms (respectively, clauses). We use w-DNF to denote the set of DNFs with width at most w. Let $\mathrm{DT}(t)$ denote decision trees with depth at most t. We say that a multiple output function $f = (f_1, \ldots, f_m)$ is in $\mathcal{C}$ if $f_i \in \mathcal{C}$ for any $i \in [m]$.

2.3.3 Leaky Function Families

Given an arbitrary class of tampering functions, we consider a variant of the class of tampering functions which may depend in some limited way on limited leakage from the underlying code word.

Definition 6 (Leaky Function Families)**.** Let $\mathrm{LL}^{i,m,N}[\mathcal{C}]$ denote tampering functions generated via the following game:

1. The adversary first commits to N functions from a class $\mathcal{C}$, $F_1, \ldots, F_N = \boldsymbol{F}$.
   (Note: $F_j : \{0,1\}^N \to \{0,1\}$ for all $j \in [N]$.)
2. The adversary then has i-adaptive rounds of leakage. In each round $j \in [i]$,
   - the adversary selects s indices from $[N]$, denoted S_j,
   - the adversary receives $\boldsymbol{F}(x)_{S_j}$.

   Formally, we take $h_j : \{0,1\}^{m(j-1)} \to [N]^m$ to be the selection function such that
   $$h_j(F(X)_{S_1}, \ldots, F(X)_{S_{j-1}}) = S_j.$$
   Let h_1 be the constant function that outputs S_1.
3. Finally, selects a sequence of n functions $(F_{t_1}, \ldots, F_{t_n})$ ($T = \{t_1, \ldots, t_n\} \subseteq [N]$ such that $t_1 < t_2 < \cdots < t_n$) to tamper with.
   Formally, we take $h : \{0,1\}^{mi} \to [N]^n$ such that $h(F(X)_{S_1}, \ldots, F(X)_{S_i}) = T$.

Thus, any $\tau \in \mathrm{LL}^{i,m,N}[\mathcal{C}]$ can be described via $(\boldsymbol{F}, h_1, \cdots, h_i, h)$. In particular, we take $\tau = \mathrm{Eval}(\boldsymbol{F}, h_1, \cdots, h_i, h)$ to denote the function whose output given input X is $T(X)$, where T is, in turn, outputted by the above game given input X and adversarial strategy $(\boldsymbol{F}, h_1, \cdots, h_i, h)$.

2.4 Pseudorandom Ingredients

2.4.1 A Binary Reconstructible Probabilistic Encoding Scheme

Reconstructable Probabilistic Encoding (RPE) schemes were first introduced by Choi et al. [CDMW08, CDMW16]. Informally, RPE is a combination of error correcting code and secret sharing, in particular, it is an error correcting code with an additional secrecy property and reconstruction property.

Definition 7 (Binary Reconstructable Probabilistic Encoding)**.** [CDMW08, CDMW16] We say a triple $(\mathrm{E}, \mathrm{D}, \mathrm{R})$ is a binary reconstructable probabilistic encoding scheme with parameters $(k, n, c_{\mathrm{err}}, c_{\mathrm{sec}})$, where $k, n \in \mathbb{N}$, $0 \leq c_{\mathrm{err}}, c_{\mathrm{sec}} < 1$, if it satisfies the following properties:

1. **Error correction.** $\mathrm{E}\colon \{0,1\}^k \to \{0,1\}^n$ is an efficient probabilistic procedure, which maps a message $x \in \{0,1\}^k$ to a distribution over $\{0,1\}^n$. If we let $\mathcal{C}$ denote the support of E, any two strings in $\mathcal{C}$ are $2c_{\mathrm{err}}$-far. Moreover, D is an efficient procedure that given any $w' \in \{0,1\}^n$ that is ε-close to some string w in $\mathcal{C}$ for any $\varepsilon \leq c_{\mathrm{err}}$, outputs w along with a consistent x.

2. **Secrecy of partial views.** For all $x \in \{0,1\}^k$ and any non-empty set $S \subset [n]$ of size $\leq \lfloor c_{\mathrm{sec}} \cdot n \rfloor$, $\mathrm{E}(x)_S$ is identically distributed to the uniform distribution over $\{0,1\}^{|S|}$.

3. **Reconstruction from partial views.** R is an efficient procedure that given any set $S \subset [n]$ of size $\leq \lfloor c_{\mathrm{sec}} \cdot n \rfloor$, any $\hat{c} \in \{0,1\}^n$, and any $x \in \{0,1\}^k$, samples from the distribution $\mathrm{E}(x)$ with the constraint $\mathrm{E}(x)_S = \hat{c}_S$.

Lemma 1. *[CDMW08, CDMW16] For any $k \in \mathbb{N}$, there exist constants $0 < c_{rate}, c_{err}, c_{sec} < 1$ such that there is a binary RPE scheme with parameters $(k, c_{rate}k, c_{err}, c_{sec})$.*

To achieve longer encoding lengths n, with the same $c_{\mathrm{err}}, c_{\mathrm{sec}}$ parameters, one can simply pad the message to an appropriate length.

RPE was been used in Ball et al.[BDKM16] for building non-malleable reductions from local functions to split state functions. However, for all reductions in our paper, error correction property is not necessary (RPE $c_{\mathrm{err}} = 0$ is adequate). In addition, we observe that RPEs with parameters $(k, n, 0, c_{\mathrm{sec}})$ are implied by any linear error correcting code with parameters (k, n, d) where k is the message length, n is the codeword length, $d := c_{\mathrm{sec}} \cdot n + 1$ is the minimal distance.

Lemma 2. *Suppose there exists a binary linear error correcting code with parameters (k, n, d), then there is a binary RPE scheme with parameters $(k, n, 0, (d-1)/n)$.*

Proof. For a linear error correcting code with (k, n, d), let A denote its encoding matrix, H denote its parity check matrix. Let B be a matrix so that $BA = I$ where I is the $k \times k$ identity matrix (such B exists because A has rank k and can be found efficiently). By property of parity check matrix, $HA = \mathbf{0}$ and $Hs \neq 0$ for any $0 < ||s||_0 < d$ where $\mathbf{0}$ is the $(n-k) \times k$ all 0 matrix.

We define $(\mathrm{E}, \mathrm{D}, \mathrm{R})$ as follows: for $x \in \{0,1\}^k$ and randomness $r \in \{0,1\}^{n-k}$, $\mathrm{E}(x; r) := B^T x + H^T r$, for $c \in \{0,1\}^n$; $\mathrm{D}(c) := A^T c$; given $S \subset [n]$ of size $\leq d-1$, $\hat{c} \in \{0,1\}^n$, $x \in \{0,1\}^k$, R samples r uniformly from the set of solutions to $(H^T r)_S = (\hat{c} - B^T x)_S$ then outputs $\mathrm{E}(x; r)$.

(E, D) is an encoding scheme because $\mathrm{D} \circ \mathrm{E} = A^T B^T = I^T = I$. For secrecy property, note that for any non-empty $S \subseteq [n]$ of size at most $d-1$, $(Hr)_S$ is distributed uniformly over $\{0,1\}^{|S|}$, because for any $a \in \{0,1\}^{|S|}$,

$$\Pr_r[(H^T r)_S = a] = \mathrm{E}[\Pi_{i \in S} \frac{1 + (-1)^{(H^T r)_i + a_i}}{2}] = 2^{-|S|} \sum_{S' \subseteq S} \mathrm{E}[\Pi_{i \in S'} (-1)^{(H^T r)_i + a_i}] = 2^{-|S|},$$

where the last equality is because the only surviving term is $S' = \emptyset$ and for other S', $\sum_{i \in S'} H_i^T \neq 0$ so $\mathrm{E}[\Pi_{i \in S'}(-1)^{(H^T r)_i}] = 0$. It implies $\mathrm{E}(x)_S$ is also distributed uniformly over $\{0,1\}^S$. By definition, R satisfies reconstruction property. Hence $(\mathrm{E}, \mathrm{D}, \mathrm{R})$ is a binary RPE with parameters $(k, n, 0, (d-1)/n)$.

□

2.4.2 A Simple σ-wise Independent Generator

Definition 8 (Bounded-independent Generator)**.** We say a distribution $\mathcal{D}$ over $\{0,1\}^n$ is *σ-wise independent* if for any $S \subseteq [n]$ of size at most σ, $\mathcal{D}_S$ distributes identically to the uniform distribution over $\{0,1\}^{|S|}$. We say function $G\colon \{0,1\}^s \to \{0,1\}^n$ is a σ-wise independent generator if $G(\zeta)$ is σ-wise independent where ζ is uniformly distributed over $\{0,1\}^s$.

The simple Carter-Wegman hashing construction based on random polynomials suffices for our purposes.

Lemma 3. *[WC81] There exists an (explicit) σ-wise independent generator: $G : \{0,1\}^{(\sigma+1)\log n} \to \{0,1\}^n$, computable in time $\tilde{O}(\sigma n)$.*

Moreover, $G : \{0,1\}^{(\sigma+1)m} \to \{0,1\}^n$ can be constructed such that for an subset $S \subseteq [n]$ of size σ, $G(\zeta)_S \equiv X_1, \ldots, X_\sigma$ (for uniformly chosen ζ) where (1) X_i's are independent Bernoullis with $\Pr[X_i = 1] = q/d$ for $q \in \{0, \ldots, d\}$, and (2) $m = \max\{\log n, \log d\}$.

Let $G_{\sigma,p}$ denote a σ-wise independent Carter-Wegman generator with bias p, and G_σ such a generator with $p = 1/2$

The following useful theorem gives Chernoff-type concentration bounds for σ-wise independent distributions.

Theorem 5 ([SSS95])**.** *If X is a sum of σ-wise independent random indicator variables with $\mu = \mathbb{E}[X]$, then $\forall \varepsilon : 0 < \varepsilon \leq 1, \sigma \leq \varepsilon^2 \mu e^{-1/3}$, $\Pr[|X - \mu| > \varepsilon\mu] < \exp(-\lfloor \sigma/2 \rfloor)$.*

2.4.3 The Pseudorandom Switching Lemma of Trevisan and Xue

Definition 9. Fix $p \in (0,1)$. A string $s \in \{0,1\}^{n \times \log(1/p)}$ encodes a subset $L(s) \subseteq [n]$ as follows: for each $i \in [n]$,

$$i \in L(s) \iff s_{i,1} = \cdots = s_{i,\log(1/p)} = 1.$$

Definition 10. Let $\mathcal{D}$ be a distribution over $\{0,1\}^{n \log(1/p)} \times \{0,1\}^n$. This distribution defines a distribution $\mathcal{R}(\mathcal{D})$ over restrictions $\{0,1,*\}^n$, where a draw $\rho \leftarrow \mathcal{R}(\mathcal{D})$ is sampled as follows:

1. Sample $(s, y) \leftarrow \mathcal{R}(\mathcal{D})$, where $s \in \{0,1\}^{n \log(1/p)}, y \in \{0,1\}^n$.
2. Output ρ where

$$\rho_i := \begin{cases} y_i & \text{if } i \notin L(s) \\ * & \text{otherwise} \end{cases}$$

Theorem 6 (Polylogarithmic independence fools CNF formulas [Baz09, Raz09])**.** *The class of M-clause CNF formulas is ε-fooled by $O((\log(M/\varepsilon))^2)$-wise independence.*

Theorem 7 (A Pseudorandom version of Håstad's switching lemma [TX13])**.** *Fix $p, \delta \in (0,1)$ and $w, S, t \in \mathbb{N}$. There exists a value $r \in \mathbb{N}$,*

$$r = \text{poly}(t, w, \log(S), \log(1/\delta), \log(1/p)),^{6}$$

such that the following holds. Let $\mathcal{D}$ be any r-wise independent distribution over $\{0,1\}^{n \times \log(1/p)} \times \{0,1\}^n$ If $F : \{0,1\}^n \to \{0,1\}$ is a size-S depth-2 circuit with bottom fan-in w, then

$$\Pr\left[\,\text{DT}(F \restriction \boldsymbol{\rho}) \geq t\,\right] \leq 2^{w+t+1}(5pw)^t + \delta,$$

where the probability is taken with respect to a pseudorandom restriction $\boldsymbol{\rho} \leftarrow \mathcal{R}(\mathcal{D})$.

Proof. By Lemma 7 of [TX13], any distribution $\mathcal{D}'$ over $\{0,1\}^{n \times \log(1/p)} \times \{0,1\}^n$ that ε-fools the class of all $(S \cdot 2^{w(\log(1/p)+1)})$-clause CNFs satisfies

$$\Pr\left[\,\text{DT}(F \restriction \boldsymbol{\rho}) \geq t\,\right] \leq 2^{w+t+1}(5pw)^t + \varepsilon \cdot 2^{(t+1)(2w+\log S)},$$

where the probability is taken with respect to a pseudorandom restriction $\boldsymbol{\rho} \leftarrow \mathcal{R}(\mathcal{D}')$. By Theorem 6, the class of $M := (S \cdot 2^{w(\log(1/p)+1)})$-clause CNF formulas is

$$\varepsilon := \delta \cdot 2^{-(t+1)(2w+\log S)}$$

fooled by r-wise independence where

$$r = O((\log(M/\varepsilon))^2) = \text{poly}(t, w, \log(S), \log(1/\delta), \log(1/p)),$$

and the proof is complete. □

Taking a union bound we get the following corollary.

Corollary 1. *Fix $p, \delta \in (0,1)$ and $w, S, t \in \mathbb{N}$. There exists a value $r \in \mathbb{N}$,*

$$r = \text{poly}(t, w, \log(S), \log(1/\delta), \log(1/p)),$$

such that the following holds. Let $\mathcal{D}$ be any r-wise independent distribution over $\{0,1\}^{n \times \log(1/p)} \times \{0,1\}^n$. Let $F_1, \ldots, F_M$ be M many size-S depth-2 circuits with bottom fan-in w. Then

$$\Pr_{\boldsymbol{\rho} \leftarrow \mathcal{R}_p}\left[\exists\, j \in [M] \text{ such that } \text{DT}(F_j \restriction \boldsymbol{\rho}) \geq t\right] \leq M \cdot \left(2^{w+t+1}(5pw)^t + \delta\right). \tag{1}$$

2.4.4 Helpful Functions.

Lastly, we define some convient functions. For a random restriction $\rho = (\rho^{(1)}, \rho^{(2)}) \in \{0,1\}^n \times \{0,1\}^n$, $\text{ExtIndices}(\rho^{(1)}) := (i_1, \ldots, i_k) \in [n+1]^k$ are the last k indices of 1s in $\rho^{(1)}$ where $i_1 \leq i_2 \cdots \leq i_k$ and $i_j = n+1$ for $j \in [k]$ if such index doesn't exist (k should be obvious from context unless otherwise noted).

We define a pair of functions for embedding and extracting a string x according to a random restriction, ρ. Let $\text{Embed} : \{0,1\}^{k+2n} \to \{0,1\}^n$, such that for $\rho = (\rho^{(1)}, \rho^{(2)}) \in \{0,1\}^n \times \{0,1\}^n$ and $x \in \{0,1\}^k$, and $i \in [n]$,

$$\text{Embed}(x, \rho)_i = \begin{cases} x_j & \text{if } \exists j \in [k] : i = \text{ExtIndices}(\rho^{(1)})_j \\ \rho^{(2)}_i & \text{otherwise} \end{cases}$$

And, let $\text{Extract} : \{0,1\}^{2n} \to \{0,1\}^k \times \{\perp\}$ be such that if $c \in \{0,1\}^n, \rho^{(1)} \in \{0,1\}^n$, and $\|\rho^{(1)}\|_0 \geq k$, then $\text{Extract}(c, \rho^{(1)}) = c_{\text{ExtIndices}(\rho^{(1)})}$. Otherwise, $\text{Extract}(c, \rho^{(1)}) = \perp$.

Note that, for any ρ such that $\|\rho^{(1)}\|_0 \geq k$, $\text{Extract}(\text{Embed}(x, (\rho^{(1)}, \rho^{(2)})), \rho^{(1)}) = x$.

[6] The exponent of this polynomial is a fixed absolute constant independent of all other parameters.

3 Non-Malleable Codes for Small-Depth Circuits

3.1 NM-Reducing Small-Depth Circuits to Leaky Local Functions

Lemma 4. *For $S, d, n, \ell \in \mathbb{N}, p, \delta \in (0,1)$, there exist $\sigma = \mathrm{poly}(\log \ell, \log(\ell S), \log(1/\delta), \log(1/p))$ and $m = O(\sigma \log n)$ such that, for any $2m \leq k \leq n(p/4)^d$,*

$$(\mathrm{AC}_d(S) \implies \mathrm{LL}^{d,m,n}[\mathrm{Local}^{\ell}], d\varepsilon)$$

where

$$\varepsilon = nS\left(2^{2\log \ell + 1}(5p\log \ell)^{\log \ell} + \delta\right) + \exp(-\frac{\sigma}{2\log(1/p)}).$$

We define a simple encoding and decoding scheme (See Figure 1 in below) and show this scheme is a non-malleable reduction from (leaky) class $\mathcal{F}$ to (leaky) class $\mathcal{G}$ with an additional round of leakage if functions in $\mathcal{F}$ reduce to $\mathcal{G}$ under a suitable notion pseudorandom restrictions (recall definitions 9 & 10).

Lemma 5. *Let $\mathcal{F}$ and $\mathcal{G}$ be two classes of functions. Suppose for $n \in \mathbb{N}$, $p \in (0,1)$ and any σ-wise independent distribution $\mathcal{D}$ over $\{0,1\}^{n\log(1/p)} \times \{0,1\}^n$, it holds that for any $F\colon \{0,1\}^n \to \{0,1\} \in \mathcal{F}$,*

$$\Pr_{\rho \leftarrow \mathcal{R}(\mathcal{D})}[F_\rho \text{ is not in } \mathcal{G}] \leq \varepsilon.$$

Then for $i, N, k \in \mathbb{N}$, $(\mathrm{E}^{\star}_{k,n,p,\sigma}, \mathrm{D}^{\star}_{k,n,p,\sigma})$ defined in Figure 1 is an

$$(\mathrm{LL}^{i,m,N}[\mathcal{F}] \implies \mathrm{LL}^{i+1,m,N}[\mathcal{G}], N\varepsilon + \exp(-\frac{\sigma}{2\log(1/p)}))$$

non-malleable reduction when $(4\sigma/\log(1/p)) \leq k \leq (n-m)p/2$.

To prove Lemma 4, we instantiate Lemma 5 using the pseudorandom switching lemma of Theorem 7 (in fact, Corollary 1) and iteratively reduce $\mathrm{AC}_d(S)$ to leaky local functions. Each application of the reduction, after the first, will allow us to trade a level of depth in the circuit for an additional round of leakage until we are left with a depth-2 circuit. The final application of the reduction will allow us to convert this circuit to local functions at the expense of a final round of leakage.

3.1.1 Proof of Lemma 5

The simple encoding and decoding scheme based on the pseudorandom switching lemma is defined in Figure 1.

The Lemma follows immediately from Claims 1, 2, and 3 below.

Claim 1. *For any $x \in \{0,1\}^k$, $\Pr[\mathrm{D}^*(\mathrm{E}^*(x)) = x] = 1$.*

Proof. The second step of E^* guarantees that $\mathrm{ExtIndices}(L(G(\zeta)))_1 > m$ and $\|L(G(\zeta)\|_0 \geq k$. Therefore, $\mathrm{E}_R(\zeta)$ is located in the first m bits of c and the entire x is embedded inside the remaining $n-m$ bits of c according to $L(G(\zeta))$. By the decoding property of RPE from lemma 1, $\Pr[\mathrm{D}_R(c, \ldots, c_m) = \zeta] = 1$, namely, $\Pr[\tilde{\zeta} = \zeta] = 1$. Conditioned on $\tilde{\zeta} = \zeta$, because $\|L(G(\zeta)\|_0 \geq k$, $\mathrm{D}^*(\mathrm{E}^*(x)) = \mathrm{Extract}(c, L(G(\zeta))) = x$ holds. The desired conclusion follows. □

Claim 2. *Given any $\tau = \mathrm{Eval}(\boldsymbol{F}, h_1, \ldots, h_i, h) \in \mathrm{LL}^{i,m,N}[\mathcal{F}]$, there is a distribution S_τ over $\tau' \in \mathrm{LL}^{i+1,m,N}[\mathcal{G}]$, such that for any $x \in \{0,1\}^k$, $\mathrm{D}^\star \circ \tau \circ \mathrm{E}^\star(x)$ is δ-close to $\tau'(x)$ where $\tau' \leftarrow \mathrm{S}_\tau$ and $\delta \leq \Pr[\boldsymbol{F} \circ \mathrm{E}^* \text{ is not in } \mathcal{G}]$.*

Take k, n, p, σ to be parameters.
Let $G = G_\sigma \colon \{0,1\}^{s(\sigma)} \to \{0,1\}^{n \log 1/p}$ be an σ-wise independent generator from Lemma 3.
Let $(\mathrm{E}_R, \mathrm{D}_R, \mathrm{R}_R)$ denote the RPE from lemma 1 with codewords of length $m(s) \geq \sigma/c_{\mathrm{sec}}$.
Let $\zeta^* \in \{0,1\}^{s(\sigma)}$ be some fixed string such that $\|L(G(\zeta^*))_{n-m+1,\ldots,n}\|_0 \geq k$. (For our choice of G, such a ζ^* can be found efficiently via interpolation.)

$\mathrm{E}^\star(x)$**:**

1. Draw (uniformly) random seed $\zeta \leftarrow \{0,1\}^s$ and (uniformly) random string $U \leftarrow \{0,1\}^{n-m}$.
2. Generate pseudorandom restriction, $\rho = (\rho^{(1)}, \rho^{(2)})$:
   $\rho^{(1)} \leftarrow L(G(\zeta))$; $(*)$ If $\|L(G(\zeta))_{n-m+1,\ldots,n}\|_0 < k$, set $\zeta = \zeta^*$.
   $\rho^{(2)} \leftarrow \mathrm{E}_R(\zeta) \| U$.
3. Output $c = \mathrm{Embed}(x, \rho)$.

$\mathrm{D}^\star(\tilde{c})$**:**

1. Recover tampered seed: $\tilde{\zeta} \leftarrow \mathrm{D}_R(\tilde{c}_1, \ldots, \tilde{c}_m)$.
   If $\|L(G(\tilde{\zeta}))_{n-m+1:n}\|_0 < k$, output $\perp$ and halt.
2. Output $\mathrm{Extract}\big(\tilde{c}, L(G(\tilde{\zeta}))\big)$.

Figure 1: A Pseudorandom Restriction Based Non-Malleable Reduction, $(\mathrm{E}^\star_{k,n,p,\sigma}, \mathrm{D}^\star_{k,n,p,\sigma})$

given $\mathrm{LL}^{i,m,N}[\mathcal{F}]$ tampering $\tau = (\boldsymbol{F}, h_1, \ldots, h_i, h)$ output $\tau' = (\boldsymbol{F}', h'_1, \ldots, h'_{i+1}, h')$:

1. Draw (uniformly) random seed $\zeta \leftarrow \{0,1\}^s$ and (uniformly) random string $R \leftarrow \{0,1\}^{n-m}$.
2. Generate pseudorandom restriction, $\rho = (\rho^{(1)}, \rho^{(2)})$:
   $\rho^{(1)} \leftarrow L(G(\zeta))$. $(*)$ If $\|L(G(\zeta))_{n-m+1,\ldots,n}\|_0 < k$, set $\zeta = \zeta^*$.
   $\rho^{(2)} \leftarrow \mathrm{E}_R(\zeta) \| R$
3. Apply (constructive) switching lemma with pseudorandom restriction to get function $\boldsymbol{F}' \equiv \boldsymbol{F}|_\rho$ (n-bit output).
   If $\boldsymbol{F}$ is not in $\mathcal{G}$, halt and output some constant function.
4. For $j \in [i]$, $h'_j \equiv h_j$.
5. $h'_{i+1}(y'_1, \ldots, y'_i) := h(y'_1, \ldots, y'_i)_{[m]}$.
6. $h'(y'_1, \ldots, y'_{i+1}) := h(y'_1, \ldots, y'_i)_{\mathrm{ExtIndices}(L(G(\mathrm{D}_R(y'_{i+1}))))}$.
7. Finally, output $\tau' = (\boldsymbol{F}', h'_1, \ldots, h'_{i+1}, h')$.

Figure 2: Simulator, S, for $(\mathrm{E}^\star, \mathrm{D}^\star)$

Proof. Recall that a function τ in $\mathrm{LL}^{i,m,N}[\mathcal{F}]$ can be described via $(\boldsymbol{F}, h_1, \ldots, h_i, h)$ where $\boldsymbol{F}$ is a function in $\mathcal{F}$ from $\{0,1\}^k$ to $\{0,1\}^N$ and for every $x \in \{0,1\}^k$, h takes $\boldsymbol{F}(x)_{S_1}, \ldots, \boldsymbol{F}(x)_{S_i}$ (where S_j are sets adaptively chosen by h_j for $j \in [i]$) as input and outputs a set T of size k. And the evaluation of τ on x is $\boldsymbol{F}(x)_T$.

Let S_τ be defined in Figure 2. We call a chocie of randomness ζ, U, r "good for $\boldsymbol{F} = (F_1, \cdots, F_N)$" (where r is the randomness for E_R) if $\boldsymbol{F} \circ \mathrm{E}^\star(\cdot; \zeta, U, r)$ is in $\mathcal{G}$. We will show for any good ζ, U, r for $\boldsymbol{F}$, $\mathrm{D}^\star \circ \tau \circ \mathrm{E}^\star(\cdot; \zeta, U, r) \equiv \tau'(\cdot)$, where $\tau' = S_\tau(\zeta, U, r)$.

For good ζ, U, r, note that (1) $\boldsymbol{F}' \equiv \boldsymbol{F}|_\rho$ and (2) ρ was used in both $\mathrm{E}^\star$ and S_τ. It follows that for all x, $\boldsymbol{F}'(x) = \boldsymbol{F}|_\rho(x) = \boldsymbol{F}(\mathrm{E}^\star(x; \zeta, R, r))$. Because $h'_j \equiv h_j$ for $j \in [i]$, it follows by induction that

$y'_j = y_j$ (the output of each h'_j and h_j respectively, $j \in [i]$). Therefore, $h(y_1, \cdots, y_i) = h(y'_1, \cdots, y'_i)$. It follows that $\tilde{c}_{[m]} = y'_{i+1}$ and $L(G(\mathrm{D}_R(y'_{i+1}))) = L(G(\tilde{\zeta}))$. Consequently, $h'(y'_1, \cdots, y'_{i+1})$ outputs that exact same indices that the decoding algorithm, $\mathrm{D}^\star$, will extract its output from. Thus, $\tau'(x) = \mathrm{D}^\star \circ \tau \circ \mathrm{E}^\star(x; \zeta, R, r)$ for any x.

Because S and $\mathrm{E}^\star$ sample their randomness identically, the distributions are identical, conditioned on the randomness being "good." Hence δ is at most the probability that ζ, U, r are not "good for $\boldsymbol{F}$", i.e., $\Pr[\boldsymbol{F} \circ \mathrm{E}^* \text{ is not in } \mathcal{G}]$.

□

Claim 3. $\Pr[\boldsymbol{F} \circ \mathrm{E}^* \textit{ is not in } \mathcal{G}] \leq N\varepsilon + \exp(-\sigma/2\log(1/p))$.

Proof. We first show $\mathcal{D} = G(\zeta) \| \mathrm{E}_R(\zeta) \| U$ is σ-wise independent when $\zeta \leftarrow \{0,1\}^s$ and $U \leftarrow \{0,1\}^{n-m}$. As U is uniform and independent of the rest, it suffices to simply consider $Z = G(\zeta) \| \mathrm{E}_R(\zeta)$. Fix some $S \subseteq [n\log(1/p) + m]$ such that $|S| \leq \sigma$. By the secrecy property of the RPE and $m \cdot c_{\mathrm{sec}} \geq \sigma$, conditioned on any fixed ζ, $Z_{S \cap \{n\log(1/p)+1, \ldots, n\log(1/p)+m\}}$is distributed uniformly. Therefore, ζ is independent of $Z_{S \cap \{n\log(1/p)+1, \ldots, n\log(1/p)+m\}}$, so G guarantees that $Z_{S \cap \{1, \ldots, n\log(1/p)\}}$ is independently of $S \cap \{n\log(1/p) + 1, \ldots, n\log(1/p) + m\}$ and also distributed uniformly. Therefore, Z_S is distributed uniformly.

Note that ρ in E^* is distributed identically to $\mathcal{R}(\mathcal{D})$, except when ζ^* is used. Hence

$$\Pr[\boldsymbol{F} \circ \mathrm{E}^* \text{ is not in } \mathcal{G}] \leq \Pr_{\rho \leftarrow \mathcal{R}(\mathcal{D})}[\boldsymbol{F}_\rho \text{ is not in } \mathcal{G}] + \Pr[\|L(G(\zeta))_{n-m+1,\ldots,n}\|_0 < k].$$

By our assumption and a union bound over the N boolean functions, $\boldsymbol{F}_\rho \notin \mathcal{G}$ happens with probability at most $N\varepsilon$ when $\rho \leftarrow \mathcal{R}(\mathcal{D})$. Observe that $L(G(\zeta))_{n-m+1,\ldots,n}$ is a $\frac{\sigma}{\log(1/p)}$-wise independent distribution over $\{0,1\}^{n-m}$ and each coordinate is 1 with probability p. Let $\mu = (n-m)p$ denote the expected number of 1's in $L(G(\zeta))_{n-m+1,\ldots,n}$. By linearity of expectation $\mu = (n-m)p$. For $k \leq \mu/2$ and $\frac{\sigma}{\log(1/p)} \leq \mu/8$, we can use the concentration bound from Theorem 5 to conclude that $\|L(G(\zeta))_{n-m+1,\ldots,n}\|_0 < k$ happens with probability at most $\exp(-\frac{\sigma}{2\log(1/p)})$. The desired conclusion follows. □

3.1.2 Proof of Lemma 4

To prove Lemma 4, we instantiate Lemma 5 using the pseudorandom switching lemma of Theorem 7 (in fact, Corollary 1) and iteratively reduce $\mathrm{AC}_d(S)$ to leaky local functions. Each application of the reduction, after the first, will allow us to trade a level of depth in the circuit for an additional round of leakage until we are left with a depth-2 circuit. The final application of the reduction will allow us to convert this circuit to local functions at the expense of a final round of leakage.

Let $t := \log(\ell)$ and let $\sigma := \mathrm{poly}(t, \log(2^t S), \log(1/\delta), \log(1/p))$ as in Corollary 1 so that any depth-2 circuits with bottom fan-in t become depth t decision trees with probability at least $1 - (2^{2t+1}(5pt)^t + \delta)$ under pseudorandom restrictions drawn from σ-wise independent distribution.

We use $\mathrm{AC}_d(S) \circ \mathrm{DT}(t)$ to denote alternating (unbounded fan-in) circuits of depth d, size S that take the output of depth t decision trees as input. (Note may contain up to S decision trees.) Similarly it is helpful to decompose an alternating circuit (from w-AC_d) into a base layer of CNFs or DNFs and the rest of the circuit, $\mathrm{AC}_{d-2}(S) \circ w\text{-}\mathrm{AC}_2(S')$. (Again, the base may contain up to S CNFs/DNFs of size S'.)

Claim 4. $(\mathrm{AC}_d(S) \implies \mathrm{LL}^{1,m,n}[\mathrm{AC}_{d-2}(S) \circ t\text{-}\mathrm{AC}_2(2^t S)], \varepsilon)$.

Proof. Let $F \in \mathrm{AC}_d(S)$ be a boolean function. Note that Theorem 7 and Corollary 1 are only useful for bounded width DNF and CNF. So, we view F as having an additional layer of fan-in 1 AND/OR gates, namely, as a function in 1-$\mathrm{AC}_{d+1}(S)$. Because there are at most S DNFs (or CNFs) of size S at the bottom layers of F, by Corollary 1, the probability that F is not in $\mathrm{AC}_{d-1}(S) \circ \mathrm{DT}(t)$ is at most $S\left(2^{t+2}(5p)^t + \delta\right)$ under the pseudorandom switching lemma with parameters p, δ, σ. So by Corollary 1, $(\mathrm{E}^\star, \mathrm{D}^\star)$ reduces $\mathrm{AC}_d(S)$ to $\mathrm{LL}^{1,m,n}[\mathrm{AC}_{d-1}(S) \circ \mathrm{DT}(t)]$ with error $n(S\left(2^{t+2}(5p)^t + \delta\right)) + \exp(-\Omega(\frac{\sigma}{\log(1/p)})) \leq \varepsilon$.

By the fact that $\mathrm{DT}(t)$ can be computed either by width-t DNFs or width-t CNFs of size at most 2^t, any circuit in $\mathrm{AC}_{d-1}(S) \circ \mathrm{DT}(t)$ is equivalent to a circuit in $\mathrm{AC}_{d-2}(S) \circ t\text{-}\mathrm{AC}_2(2^t S)$, in other words, a depth d circuit with at most S width-t size-$S2^t$ DNFs or CNFs at the bottom. Hence, $\mathrm{AC}_{d-1}(S) \circ \mathrm{DT}(t)$ is a subclass of $\mathrm{AC}_{d-2}(S) \circ t\text{-}\mathrm{AC}_2(2^t S)$ and the claim follows. □

Claim 5. $(\mathrm{LL}^{i,m,n}[\mathrm{AC}_{d-i-1}(S) \circ t\text{-}\mathrm{AC}_2(2^t S)] \implies \mathrm{LL}^{i+1,m,n}[\mathrm{AC}_{d-i-2}(S) \circ t\text{-}\mathrm{AC}_2(2^t S)], \varepsilon)$.

Proof. For a boolean function $F \in \mathrm{AC}_{d-i-1}(S) \circ t\text{-}\mathrm{AC}_2(2^t S)$, because there are at most S DNFs (or CNFs) of size $2^t S$ at the bottom layers of F, Corollary 1 shows F is not in $\mathrm{AC}_{d-i-1}(S) \circ \mathrm{DT}(t)$ with probability at most $S\left(2^{2t+2}(5pt)^t + \delta\right)$ under a pseudorandom switching lemma with parameters p, δ, σ. So by Lemma 5, $(\mathrm{E}^\star, \mathrm{D}^\star)$ reduces $(\mathrm{LL}^{i,m,n}[\mathrm{AC}_{d-i-1}(S) \circ t\text{-}\mathrm{AC}_2(2^t S)]$ to $\mathrm{LL}^{i+1,m,n}[\mathrm{AC}_{d-i-2}(S) \circ \mathrm{DT}(t)]$ with error at most ε. Similarly as the previous proof, because $\mathrm{AC}_{d-i-1}(S) \circ \mathrm{DT}(t)$ is a subclass of $\mathrm{AC}_{d-i-2}(S) \circ t\text{-}\mathrm{AC}_2(2^t S)$, the claim follows. □

Claim 6. $(\mathrm{LL}^{d-1,m,n}[t\text{-}\mathrm{AC}_2(2^t S)] \implies \mathrm{LL}^{d,m,n}[\mathrm{Local}^{2^t}], \varepsilon)$

Proof. Finally, for a boolean function $F \in t\text{-}\mathrm{AC}_2(2^t S)$, Corollary 1 shows F is not in $\mathrm{DT}(t)$ with probability at most $S\left(2^{2t+2}(5pt)^t + \delta\right)$. So by Lemma 5, $(\mathrm{E}^\star, \mathrm{D}^\star)$ reduces $\mathrm{LL}^{d-1,m,n}[t\text{-}\mathrm{AC}_2(2^t S)]$ to $\mathrm{LL}^{d,m,n}[\mathrm{DT}(t)]$ with error at most ε. The desired conclusion follows from the fact that $\mathrm{DT}(t)$ is a subclass of Local^{2^t}. □

By applying Claim 4 once, then Claim 5 $(d-2)$ times and Claim 6 once, $\mathrm{AC}_d(S)$ reduces to $\mathrm{LL}^{d,m,n}[\mathrm{Local}^{2^t}]$ with error at most $d\varepsilon$. Note that $m = O(\sigma \log n)$ throughout, and during each application of above claims, given a codeword of length $n' \geq k \geq 2m$, Lemma 5 holds for messages of length $(n'-m)p/2 \geq n'(p/4)$. Therefore, the composed reduction works for any $2m \leq k \leq n(p/4)^d$.

3.2 NM-Reducing Leaky Local to Split State

Simple modifications to construction from the appendix of [BDKM16] yield a $(\mathrm{LL}^{d,s,N}[\mathrm{Local}^\ell], \mathrm{SS}_k, \mathrm{negl}(k))$-non-malleable reduction.

Lemma 6. *There exists a constant $c \in (0,1)$, such that for any m, q, ℓ satisfying $mq\ell^3 \leq cn$ there is a $(\mathrm{LL}^{q,m,N}[\mathrm{Local}^\ell] \implies \mathrm{SS}_k, \exp(-\Omega(k/\log n)))$-non-malleable reduction with rate $\Omega(1/\ell^2)$.*

Note that we do not actually require any restrictions on N.

We construct an encoding scheme (E, D), summarized in Figure 3, adapted from the appendix of [BDKM16]. We then show that the pair (E, D) is a $(\mathrm{LL}^{d,s,N}[\mathrm{Local}^\ell], \mathrm{SS}_k, \mathrm{negl}(k))$-non-malleable reduction.

Let $\mathrm{G} = \mathrm{G}_{p,\sigma} : \{0,1\}^{s(\sigma)} \to \{0,1\}^{\tau}$ be a σ-wise independent generator with bias $p = \frac{3n_L}{2\tau}$(see Lemma 3, $s = s(\sigma) = \sigma \log(2\tau) = O(\sigma \log n)$), with inputs of length s and outputs of length τ.
Let $(\mathrm{E}_L, \mathrm{D}_L)$, $(\mathrm{E}_Z, \mathrm{D}_Z)$, $(\mathrm{E}_R, \mathrm{D}_R)$ be RPEs with parameters $(k, n_L, c_{\mathrm{sec}}, c_{\mathrm{err}})$, $(s, n_Z, c_{\mathrm{sec}}, c_{\mathrm{err}})$, and $(k, n_R, c_{\mathrm{sec}}, c_{\mathrm{err}})$ respectively.
Assume $\ell > 1/c_{\mathrm{sec}}$. Define feasible parameters according to the following:
$n_Z \geq \max\{mq\ell/c_{\mathrm{sec}}, s(\sigma)c_{\mathrm{rate}}\}$ (Take $n_Z = \theta(mq\ell + s(\sigma))$),
$n_L \geq kc_{\mathrm{rate}}$ (Take $n_L = \theta(k)$),
$n_R \geq \frac{\ell}{c_{\mathrm{sec}}}(n_L + n_Z + mq)$ (Take $n_R = \theta(\ell(k + mq\ell + s(\sigma)))$),
$\tau \geq \frac{9\ell}{4c_{\mathrm{sec}}}(n_R + n_Z + mq)$ (Take $= \theta(\ell^2(k + mq\ell + s(\sigma)))$),
$n := n_Z + \tau + n_R$ ($n = \theta(\ell^2(k + mq\ell + s(\sigma)))$.

$\mathrm{E}(x^L := x_1^L, \ldots, x_k^L, x^R := x_1^R, \ldots, x_k^R)$:

1. Let $s_L := \mathrm{E}_L(x^L), S_R := \mathrm{E}_R(x^R)$.
2. Choose $\zeta \leftarrow \{0,1\}^s$ uniformly at random. Compute $\rho^{(1)} := \mathrm{G}(\zeta)$. Choose $\rho^{(2)} \leftarrow \{0,1\}^{\tau}$ uniformly at random. Let $\rho := (\rho^{(1)}, \rho^{(2)})$; $(*)$ If $\rho^{(1)}$ has less than n_L 1s, take $\rho^{(1)} := \mathrm{G}(\zeta^*)$ for some ζ^* such that $\mathrm{G}(\zeta^*)$ has n_L ones.
3. Let $X_L := \mathrm{Embed}(s_L, \rho)$.
4. Let $Z \leftarrow \mathrm{E}_L(\zeta)$; Output the encoding (Z, X_L, S_R).

$\mathrm{D}(\widetilde{Z}, \widetilde{X_L}, \widetilde{S_R})$:

1. Let $\widetilde{\rho} := G(\mathrm{D}_L(\widetilde{Z}_L))$.
   $(*)$ If $\widetilde{\rho}$ contains less than n_L 1s, output $\perp$.
2. $\widetilde{s_L} := \mathrm{Extract}(X_L, \widetilde{\rho})$.
3. Let $\widetilde{x^L} = \mathrm{D}_L(\widetilde{s_L})$, $\widetilde{x^R} = \mathrm{D}_R(\widetilde{S_R})$; Output $(\widetilde{x^L}, \widetilde{x^R})$.

Figure 3: A NON-MALLEABLE REDUCTION OF $\mathrm{LL}^{q,m,N}[\mathrm{Local}^{\ell}]$ TO SPLIT STATE SS_k WITH DETERMINISTIC DECODING

Security. Before formalizing, we will briefly describe why the construction works. We will reduce the leaky local tampering to split-state tampering using the encoding and decoding algorithms. Given that encoding/decoding on the left (x_L and Z, X_L, respectively) is independent of encoding/decoding on the right (x_R and S_R, respectively), all of non-split state behavior is derived from the tampering function. We will show how to essentially sample all of the information necessary to tamper independently on each side without looking at the inputs. Then, using the reconstruction properties of the RPEs we will be able to generate encodings on each side consistent with these common random bits that we have sampled. Conditioned on a simple event happening, composition of these modified encoding, tampering, and decoding algorithms will be identical to the normal tampering experiment.

The key observation, is that all of the leakage is under the privacy threshold of any of the RPEs. In particular, this means that after calculating all of the leakage to define which functions will be applied to the codeword, both left and right inputs remain private, *as well as* the seed, ζ. Moreover, the leakage is far enough below the privacy thresholds on the inputs that we may leak more bits.

Given the local functions that will be applied to the codeword, the bits that will affect either the tampered seed, or the right side, or were used to calculate the leakage from X_L have all been defined (and there aren't too many relative to the length of X_L). As the seed, ζ, is still uniformly distributed at this point we can sample it and apply a pseudorandom chernoff bound to show that, with overwhelming probability, relatively few of these locations will overlap with embedding locations for the (RPE encoding of the) left side input, this is the "simple event" mention above. (We additionally require that the embedding has enough space for the RPE encoding.) Consequently, we can safely sample all these locations in X_L. Additionally, at this point we have totally defined the RPE of the seed, Z.

Now, because the RPE of the seed is significantly shorter than RPE of the right input, we can safely sample all the locations in S_R that affect $\tilde{Z}$. Moreover, the tampering resulting in $\tilde{Z}$ is now a constant function (given all the sampled bits), which allows us to simulate the tampered seed, $\tilde{\zeta}$. Then, we can use the tampered seed to determine decoding locations for extracting the RPE of the left input from X_L. As these are the only locations that the output of decoding depends on, we only are concerned with the bits that affect these (few) locations. As there are less than security threshold bits in S_R that affect these locations (in conjunction with bits that affect $\tilde{Z}$), we can sample all of these locations uniformly at random. At this point we can now output the left-side tampering function: given left input x_L, reconstruct an RPE to be consistent with the bits of s_L sampled above, apply tampering function (given by simulated exacted locations and restricted according to *all* the sampled bits above that is only dependent on s_L), decode the result.

Also note that at this point the tampered RPE of the right input, $\tilde{S}_R$, is simply a function of random bits (sampled independently of the input) and the RPE S_R. Thus, we can similarly output the right-side tampering function: given right input x_R, reconstruct an RPE to be consistent with the bits of S_R sampled above, apply tampering function (restricted according to *all* the sampled bits above: only depends on S_R), decode the result of tampering.

Proof of Lemma 6. We begin by formally defining a simulator in Figure 4.

We additionally consider the following "bad events":

1. $G(\zeta)$ contains at least n_L 1s. (Condition $*$ does not occur.)
2. Given $(h_1, \ldots, h_k)$-leakage, denoted $(y_1, \ldots, y_k)$, the resulting tampering function $\boldsymbol{F}_{h(y_1,\ldots,y_k)}$ is such that the intersection of the set V' (defined as in figure 4) with $\{i + n_Z : i \in \text{ExtIndices}(\text{G}(\zeta))\}$ is less than $c_{\text{sec}} \cdot n_L$. (Condition $**$ does not happen.)

Given $\mathrm{LL}^{q,m,N}[\mathrm{Local}^{\ell}]$ tampering $t = (\boldsymbol{F}, h_1, \ldots, h_q, h)$ output (f_L, f_R):
Let $A_F(S)$ denote the indices (in $[n]$) of inputs that affect $\boldsymbol{F}_S$ for $S \subseteq [N]$. Let $I_Z = \{1, \ldots, n_Z\}$, $I_L = \{n_Z + 1, \ldots, n_Z + \tau\}$, and $I_R = \{n_Z + \tau + 1, \ldots, n_Z + \tau + n_R\}$.

1. Sample uniform $r \in \{0,1\}^n$.
2. (Sample leakage) Let $S_1 = h_1$. Let $U_1 = (r_j)_{j \in S_1}$
   For $i = 1$ to q:
   Let $S_i := h_i(Y_1, \ldots, Y_{i-1})$, $U_i := A(S_i)$, $Y_i := \boldsymbol{F}_{S_i}(r)$.
3. Let $(T_Z, T_X, T_R) := h(Y_1, \ldots, Y_q)$.
4. Let $V_Z := A_F(T_Z) \cap (I_L \cup I_R)$, $V_R := A_F(T_R) \cap I_L$, and $V' := V_Z \cup V_R \cup U$ where $U = \bigcup U_i$.
5. Let $\mu' \in \{0, 1, \star\}^n$ denote the string where $\forall i \in V' : \mu'_i = r_i$, and $\forall i \notin V' : \mu'_i = \star$.
6. (Sample seed) $\zeta \leftarrow \{0,1\}^s$ uniformly at random. Compute $\rho^{(1)} := \mathrm{G}_{p,q}(\zeta)$. Let $\rho := (\rho^{(1)}, r_{\{n_Z+1,\ldots,n_Z+\tau\}})$. For $i \in [\tau]$, let $\rho^{(1)}_i$ denote the i-th bit of $\rho^{(1)}$.
   $(*)$ If $\rho^{(1)}$ has less than n_L 1s, take $\rho^{(1)} := \mathrm{G}(\zeta^*)$ for some ζ^* such that $\mathrm{G}(\zeta^*)$ has n_L ones.
   $(**)$ If $\sum_{i \in V'} \rho^{(1)}_{i - n_Z} > c_{\mathrm{sec}} n_L$ (if $\rho^{(1)}$ has too many 1s with indices in V', after shifting), output some constant function and halt.
   Let $I'_L := (i_1 + n_Z, \ldots, i_{n_L} + n_Z)$, where $(i_1, \ldots, i_{n_L}) := \mathrm{ExtIndices}(\rho^{(1)})$. Let $B := I'_L \cap V'$ (i.e. the embedding locations that are also in V').
7. (Reconstruct encodings consistent with μ') Let $C := I_Z \cap V'$. $Z \leftarrow \mathrm{R}_Z(C, \mu'_{I_Z}, \zeta)$.
8. (Recover tampered seed) $\tilde{\zeta} := \mathrm{D}_Z \circ \boldsymbol{F}_{T_Z}|_{\mu'}(Z)$, and $\tilde{\rho} := \mathrm{G}(\tilde{\zeta})$.
   (By definition, $T_Z|_{\mu'}$ is only a function of the variables in I_Z.)
9. (Recover tampered extraction locations) Let $J = (j_1, \cdots, j_{n_L})$ denote the set of elements in $\mathrm{ExtIndices}(\tilde{\rho})$. If n_L elements cannot be recovered, output $\perp$ and halt.
   Let $T_L := T_{X,j_1}, \cdots, T_{X,j_{n_L}}$ (where $T_{X,v}$ denotes the v-th element in T_X), and $V_L := A_F(T_L) \cap (I_Z \cup I_R)$.
10. (Extend μ' to μ) Let $V := V' \cup V_L \cup I_Z$ and $\forall i \in V \setminus I_Z : \mu_i = r_i$, $\forall i \in I_Z : \mu_i = Z_i$, and $\forall i \notin V : \mu_i = \star$.
    (Note that $\forall i \in V', \mu_i = \mu'_i$, and, consequently, $\boldsymbol{F}_{T_Z}|_{\mu'}(Z) \equiv \boldsymbol{F}_{T_Z}|_{\mu}(Z)$.)
11. Output:

    f_L: On input x,

    (a) (Reconstruct encodings consistent with μ) $s_L \leftarrow \mathrm{R}_L(B, \mu_{I_L}, x_L)$,
    (b) (Embed reconstructed encoding) $X_L := \mathrm{Embed}(s_L, \rho)$
    (c) (Tamper) $\tilde{c}_L := T_L|_{\mu}(X_L)$
    (d) (Decode) Output $\tilde{x}_L := \mathrm{D}_L(\tilde{c}_L)$.

    f_R: On input y

    (a) (Reconstruct encodings consistent with μ) Let $A := \{i - (\tau + n_Z) : i \in V \cap I_R\}$ $S_R \leftarrow \mathrm{R}_R(A, \mu_{I_R}, x_R)$.
    (b) (Tamper) $\tilde{c}_R := T_R|_{\mu}(S_R)$.
    (c) (Decode) Output $\tilde{x}_R := \mathrm{D}_R(\tilde{c}_R)$.

Figure 4: Simulator, S, for (E, D)

Next we argue, via a sequence of hybrids, that for any fixed input (x_L, x_R) and tampering function t, $\Delta(\mathrm{D}(f(\mathrm{E}(x_L, x_R))); G(x_L, x_R)) \leq \exp(-\sigma/2 + 1)$, where G denotes the distribution over split-state functions (f_L, f_R) induced by the simulator S.

Hybrid H_0 (The real experiment): Outputs $\mathrm{D} \circ t \circ \mathrm{E}(x_L, x_R)$.

Hybrid H_1 (Alternate RPE encoding):

In this hybrid, we change the order of sampling in the encoding procedure: We first sample ζ, ρ, μ', Z as in S and then reconstruct RPEs s_L and S_R to be consistent with μ'.

Specifically, replace the encoding procedure E with the following: On input (x_L, x_R), first execute S steps 1-7, then sample $s_L \leftarrow \mathrm{R}_L(B, \mu'_{I_L}, x_L)$, $X_L := \mathrm{Embed}(s_L, \rho)$, and $S_R \leftarrow \mathrm{R}_R(A, \mu'_{I_R}, x_R)$ and output (Z, X_L, S_R).

Hybrid H_2 (Simulate tampered seed/"Alternate" Left-side decoding):

In this hybrid, we modify the decoding procedure to simulate tampered seed, $\tilde{\zeta}$ using Z, μ' sampled as in the previous experiment. We then use its extracted locations $\tilde{\rho}$ to extract the embedded RPE encoding s_L.

Specifically, on input $(\widetilde{Z}, \widetilde{X_L}, \widetilde{S_R})$, we replace steps 1 and 2 in decoding procedure D with steps 8,9 in S.

Hybrid H_3 (Alternate Alternate RPE encoding):

In this hybrid we again change the order of sampling in the encoding procedure. This time we sample $Z, \mu', \tilde{\rho}$ as in the previous hybrid, and then sample μ and reconstruct s_L, S_R as in S.

Specifically, on input (x_L, x_R), sample $Z, \mu', \tilde{\rho}$ as before and sample μ as in Step 10 of S. Then, set $s_L \leftarrow \mathrm{R}_L(B, \mu_{I_L}, x_L)$, $X_L := \mathrm{Embed}(s_L, \rho)$, $S_R \leftarrow \mathrm{R}_R(A, \mu_{I_R}, x_R)$ and output (Z, X_L, S_R) as the output of the encoding procedure.

Hybrid H_4 (The split-state simulation):

In this hybrid, instead of applying the actual tampering function $t = (\boldsymbol{F}, h_1, \ldots, h_q, h)$ to the output of the encoding procedure from H_4 and then applying the decoding procedure from H_4, we instead simply output $f_L(x^L), f_R(x^R)$, where f_L, f_R are defined as in S.

We will show that, $H_0 \approx_{\mathrm{negl}(n)} H_1$, and, in fact, $H_1 \equiv H_2 \equiv H_3 \equiv H_4$.

$\|H_0 - H_1\| \leq \exp(-\sigma/2 + 1)$: It is sufficient to show that: (1) conditioned on $*$ and $**$ not occurring, experiments H_0 and H_1 are identical (2) the probability of $*$ or $**$ occurring is at most $\exp(-\sigma/2 + 1)$.

Notation. For every variable x that is set during experiments H_0, H_1, let $\boldsymbol{x}$ denote the corresponding random variable. Given a string μ' of length n and a set $S \subseteq [n]$, define the n-bit string $\mu'(S)$ as $\mu'(S)_i = \mu'_i$, $i \in S$ and $\mu'(S)_i = 0$, $i \notin S$.

For (1), it is sufficient to show that $(\boldsymbol{\zeta}, \boldsymbol{\mu'(V')})$ are identically distributed in H_0, H_1, conditioned on $*$ and $**$ not occurring, where $\boldsymbol{\mu'}$ is the random variable denoting the outcome of (Z, X_L, S_R). In order to compute steps $2-4, 6, 7$ of S, we need only (adaptively) fix the bits $(Z, X_L, S_R)_U$ corresponding to the set $U = \bigcup U_i$. Since $|U| \leq \ell m q \leq c_{\mathrm{sec}} \min\{n_L, n_R, n_Z\}$, by the properties of the RPE, this means that $(\boldsymbol{\zeta}, \boldsymbol{\mu'(U)})$ are identically distributed in H_0 and H_1.

Since $*$ and $**$ depend only on ζ and $\mu'(U)$, it is sufficient to show that, for every $\zeta, \mu'(U)$ for which $*$ and $**$ do not occur, the distributions over $\boldsymbol{\mu'(V')}$, conditioned on $(\boldsymbol{\zeta} = \zeta) \wedge (\boldsymbol{\mu'(U)} = \mu'(U))$ are identical in H_1 and H_2. Due to independence of $\boldsymbol{\zeta}, \boldsymbol{s_L}, \boldsymbol{S_R}$, the fact that $\boldsymbol{\mu'_{V' \cap (I_L \setminus I'_L)}}$ is uniform random in both experiments, and since $V' \cap I_Z = U \cap I_Z$, it remains to show that each of $(\boldsymbol{\mu'(V' \cap I'_L)} \mid \boldsymbol{\mu'(U)} = \mu'(U))$ and $(\boldsymbol{\mu'(V' \cap I_R)} \mid \boldsymbol{\mu'(U)} = \mu'(U))$ are identically distributed in both experiments.

Since $**$ does not occur, the total size of $I'_L \cap V' = B$ is at most $c_{\text{sec}} \cdot n_L$ and the total size of $I_R \cap V'$ is at most $|V_Z| + |U| \leq \ell \cdot (n_Z + mq) \leq c_{\text{sec}} \cdot n_R$. Therefore, by the properties of the RPE, the corresponding distributions in H_0 and H_1 are identical.

We now turn to proving (2). To bound the probability of $*$, note that the expected number of 1's in $G(\zeta)$ is $\tau \cdot p = \frac{3\tau n_L}{2\tau} = \frac{3n_L}{2}$. Invoking Theorem 5, item (1) with $k = \sigma$, $\mu = \frac{3n_L}{2}$ and $\varepsilon = \frac{1}{2}$, it follows that $\Pr[*] \leq \exp(-\sigma/2)$.

To bound the probability of $**$, note that given fixed set V', the expected size of $V' \cap \{i + n_Z : i \in \text{ExtIndices}(\text{G}(\zeta))\}$ is at most $|V'| \cdot p = \frac{3|V'|n_L}{2\tau} = \frac{2c_{\text{sec}}|V'|n_L}{3\ell(n_R+n_Z+mq)}$. Now, $|V'| \leq \ell(n_R + n_Z + mq)$. So $\frac{2c_{\text{sec}}|V'|n_L}{3\ell(n_R+n_Z+mq)} \leq \frac{2c_{\text{sec}}n_L}{3}$. Invoking Theorem 5, item (1) with $k = \sigma$, $\mu = \frac{2c_{\text{sec}}n_L}{3}$ and $\varepsilon = \frac{1}{3}$, it follows that $\Pr[**] \leq \exp(-\sigma/2)$.

The conclusion follows from a union bound.

$\|H_1 - H_2\| = 0$**:** By inspection, it can be seen that the two experiments are, in fact, identical.

$\|H_2 - H_3\| = 0$**:** Note that the distribution over Z, X_L does not change from the previous hybrid (since all of Z is sampled based on μ' and since μ does not fix additional bits from I_L). The total number of bits of S_R fixed by μ is at most $|V_L| + |V_Z| + |U| \leq \ell \cdot (n_L + n_Z + mq) \leq c_{\text{sec}} \cdot n_R$, where the last inequality is by choice of parameters. Therefore, by the properties of the RPE (E_R, D_R), the distribution over (Z, X_L, S_R) is identical in H_1 and H_2.

$\|H_3 - H_4\| = 0$**:** By inspection, these experiments are also identical.

Correctness. By the definitions of (Embed, Extract) and RPE, $\Pr[\text{D}(\text{E}(x)) = x] = 1$. □

3.3 Putting It All Together

In this section, we put things together and show our main results. By composing the non-malleable reductions from Lemma 4 and Lemma 6, we obtain a non-malleable reduction which reduces small-depth circuits to split state.

Lemma 7. *For $S, d, n, \ell \in \mathbb{N}$, $p, \delta \in (0,1)$, there exists $\sigma = \text{poly}(\log \ell, \log(\ell S), \log(1/\delta), \log(1/p))$ such that for k that $k \geq O(\sigma \log n)$ and $k = \Omega(n(p/4)^d/\ell^2)$,*

$$(\text{AC}_d(S) \implies \text{SS}_k, d\varepsilon + \exp(-\sigma/2))$$

where

$$\varepsilon = nS\left(2^{2\log \ell + 1}(5p \log \ell)^{\log \ell} + \delta\right) + \exp(-\frac{\sigma}{2\log(1/p)}).$$

For constant-depth polynomial-size circuits (i.e. $\mathsf{AC^0}$), we obtain the following corollary by setting $\ell = n^{1/\log\log\log(n)}$, $\delta = n^{-\log\log(n)}$ and $p = \frac{1}{\log \ell} \cdot \frac{1}{\log n} = \frac{\log\log\log(n)}{\log^2 n}$,

Corollary 2. $\left(\text{AC0} \implies \text{SS}_k, n^{-(\log\log n)^{1-o(1)}}\right)$ *for $n = k^{1+o(1)}$.*

The same setting of parameters works for depth as large as $\Theta(\log(n)/\log\log(n))$ with $n = k^{1+c}$ where constant $0 < c < 1$ can be arbitrary small. We remark that one can improve the error to $n^{-\Omega(\log(n))}$ by using a smaller p (e.g. $p = n^{-1/100d}$) thus a worse rate (but still $n = k^{1+\varepsilon}$).

Combining the non-malleable code for split state from Theorem 4 with rate $\Omega(\log\log n/\log(n))$, we obtain our main theorem.

Theorem 8. *There exists an explicit, efficient, information theoretic non-malleable code for any polynomial-size, constant-depth circuits with error* $\mathrm{negl}(n)$ *and encoding length* $n = k^{1+o(1)}$.

Moreover, for any constant $c \in (0,1)$*, there exists another constant* $c' \in (0,1)$ *and an explicit, efficient, information theoretic non-malleable code for any polynomial-size,* $(c' \log n / \log\log n)$*-depth circuits with error* $\mathrm{negl}(n)$ *and encoding length* $n = k^{1+c}$.

References

[AAG+16] Divesh Aggarwal, Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Optimal computational split-state non-malleable codes. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, pages 393–417, 2016.

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

[ADL14] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 774–783. ACM, 2014.

[Agg15] Divesh Aggarwal. Affine-evasive sets modulo a prime. *Inf. Process. Lett.*, 115(2):382–385, 2015.

[AGM+15] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Explicit non-malleable codes against bit-wise tampering and permutations. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, pages 538–557, 2015.

[Ajt89] Miklós Ajtai. First-order definability on finite structures. *Ann. Pure Appl. Logic*, 45(3):211–225, 1989.

[Baz09] Louay M. J. Bazzi. Polylogarithmic independence can fool DNF formulas. *SIAM J. Comput.*, 38(6):2220–2272, 2009.

[BDKM16] Marshall Ball, Dana Dachman-Soled, Mukul Kulkarni, and Tal Malkin. Non-malleable codes for bounded depth, bounded fan-in circuits. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 881–908, 2016.

[BDKM17] Marshall Ball, Dana Dachman-Soled, Mukul Kulkarni, and Tal Malkin. Non-malleable codes from average-case hardness: AC0, decision trees, and streaming space-bounded tampering. Cryptology ePrint Archive, Report 2017/1061, 2017. `http://eprint.iacr.org/2017/1061`.

[CDMW08] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Black-box construction of a non-malleable encryption scheme from any semantically secure one. In Ran Canetti, editor, *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008.*, volume 4948 of *Lecture Notes in Computer Science*, pages 427–444. Springer, 2008.

[CDMW16] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. A black-box construction of non-malleable encryption from semantically secure encryption. *IACR Cryptology ePrint Archive*, 2016:720, 2016.

[CDTV16] Sandro Coretti, Yevgeniy Dodis, Björn Tackmann, and Daniele Venturi. Non-malleable encryption: Simpler, shorter, stronger. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 306–335, 2016.

[CG16] Mahdi Cheraghchi and Venkatesan Guruswami. Capacity of non-malleable codes. *IEEE Trans. Information Theory*, 62(3):1097–1118, 2016.

[CGL16] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 285–298, 2016.

[CGM+16] Nishanth Chandran, Vipul Goyal, Pratyay Mukherjee, Omkant Pandey, and Jalaj Upadhyay. Block-wise non-malleable codes. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 31:1–31:14, 2016.

[CHRT18] Eshan Chattopadhyay, Pooya Hatami, Omer Reingold, and Avishay Tal. Improved pseudorandomness for unordered branching programs through local monotonicity. In *Proceedings of the 50th Annual ACM Symposium on the Theory of Computing (STOC)*, 2018.

[CL17] Eshan Chattopadhyay and Xin Li. Non-malleable codes and extractors for small-depth circuits, and affine functions. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1171–1184, 2017.

[CMTV15] Sandro Coretti, Ueli Maurer, Björn Tackmann, and Daniele Venturi. From single-bit to multi-bit public-key encryption via non-malleable codes. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 532–560, 2015.

[CZ14] Eshan Chattopadhyay and David Zuckerman. Non-malleable codes against constant split-state tampering. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 306–315. IEEE Computer Society, 2014.

[CZ16] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 670–683. ACM, 2016.

[DKO13] Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Non-malleable codes from two-source extractors. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 239–257. Springer, 2013.

[DLSZ15] Dana Dachman-Soled, Feng-Hao Liu, Elaine Shi, and Hong-Sheng Zhou. Locally decodable and updatable non-malleable codes and their applications. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 427–450, 2015.

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

[DPW18] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-Malleable Codes. *Journal of the ACM*, 2018.

[FHMV17] Sebastian Faust, Kristina Hostáková, Pratyay Mukherjee, and Daniele Venturi. Non-malleable codes for space-bounded tampering. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, pages 95–126, 2017.

[FMNV14] Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. Continuous non-malleable codes. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, pages 465–488, 2014.

[FMVW14] Sebastian Faust, Pratyay Mukherjee, Daniele Venturi, and Daniel Wichs. Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 111–128, 2014.

[FSS84] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[GMR+12] Parikshit Gopalan, Raghu Meka, Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Better pseudorandom generators from milder pseudorandom restrictions. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 120–129, 2012.

[GPR16] Vipul Goyal, Omkant Pandey, and Silas Richelson. Textbook non-malleable commitments. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 1128–1141, 2016.

[Hås86] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 6–20, 1986.

[HT18] Pooya Hatami and Avishay Tal. Pseudorandom generators for low sensitivity functions. In *9th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 29:1–29:13, 2018.

[IMZ12] Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from shrinkage. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 111–119, 2012.

[KLT16] Aggelos Kiayias, Feng-Hao Liu, and Yiannis Tselekounis. Practical non-malleable codes from l-more extractable hash functions. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1317–1328, 2016.

[KOS14] Bhavana Kanukurthi, Lakshmibhavana Obbattu, and Sruthi Sekar. Four-state non-malleable codes with explicit constant rate. Theory of Cryptography - 15th Theory of Cryptography Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, to appear, 2014.

[KPPY84] Maria M. Klawe, Wolfgang J. Paul, Nicholas Pippenger, and Mihalis Yannakakis. On monotone formulae with restricted depth (preliminary version). In Richard A. DeMillo, editor, *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, pages 480–487. ACM, 1984.

[Li12] Xin Li. Non-malleable extractors, two-source extractors and privacy amplification. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 688–697. IEEE Computer Society, 2012.

[Li13] Xin Li. New independent source extractors with exponential improvement. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 783–792. ACM, 2013.

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

[Li18] Xin Li. Pseudorandom correlation breakers, independence preserving mergers and their applications. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:28, 2018.

[LL12] Feng-Hao Liu and Anna Lysyanskaya. Tamper and leakage resilience in the split-state model. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 517–532, 2012.

[Raz09] Alexander Razborov. A simple proof of bazzis theorem. *ACM Transactions on Computation Theory (TOCT)*, 1(1):3, 2009.

[RSV13] Omer Reingold, Thomas Steinke, and Salil Vadhan. Pseudorandomness for regular branching programs via Fourier analysis. In *Proceedings of the 17th International Workshop on Randomization and Computation (RANDOM)*, pages 655–670, 2013.

[SSS95] Jeanette P. Schmidt, Alan Siegel, and Aravind Srinivasan. Chernoff-hoeffding bounds for applications with limited independence. *SIAM J. Discrete Math.*, 8(2):223–250, 1995.

[ST18] Rocco A. Servedio and Li-Yang Tan. Improved pseudorandom generators from pseudorandom multi-switching lemmas. *CoRR*, abs/1801.03590, 2018.

[TX13] Luca Trevisan and Tongke Xue. A derandomized switching lemma and an improved derandomization of AC0. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 242–247. IEEE Computer Society, 2013.

[Val83] Leslie G. Valiant. Exponential lower bounds for restricted monotone circuits. In David S. Johnson, Ronald Fagin, Michael L. Fredman, David Harel, Richard M. Karp, Nancy A. Lynch, Christos H. Papadimitriou, Ronald L. Rivest, Walter L. Ruzzo, and Joel I. Seiferas, editors, *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 110–117. ACM, 1983.

[WC81] Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.

[Yao85] Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles (preliminary version). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 1–10, 1985.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il