2018 IEEE 59th Annual Symposium on Foundations of Computer Science

Non-Malleable Codes for Small-Depth Circuits

Marshall Ball*, Dana Dachman-Soled†, Siyao Guo‡, Tal Malkin§, Li-Yang Tan¶
*Columbia University, New York, NY, USA & IDC, Herzliya, Israel
Email: marshallball@cs.columbia.edu
†University of Maryland, College Park, MD, USA
Email: danadach@ece.umd.edu
‡Northeastern University, Boston, MA, USA
Email: s.guo@neu.edu
§Columbia University, New York, NY, USA
Email: tal@cs.columbia.edu
¶Stanford University, Stanford, CA, USA
Email: liyang@cs.stanford.edu

***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.

***Keywords*-Non-malleable codes, Small-depth circuits, Switching lemma**

I. Introduction

Non-malleable codes (NMC) were introduced in the seminal work of Dziembowski, Pietrzak, and Wichs as a natural generalization of error correcting codes [1], [2]. 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 $\alpha < 1$) [1], [3]. (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 [3]–[9], and tampering by flipping bits and permuting the result [10].

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-

2575-8454/18/$31.00 ©2018 IEEE
DOI 10.1109/FOCS.2018.00083

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 [11], 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 [12], Chattopadhyay and Li, using new constructions of non-malleable 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.

A. Efficient NMC for small-depth circuits

In this work, we address the main open problem from [12]: 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 I-B. 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 [9].

The flurry of work on non-malleable codes has yielded many surprising connections to other areas of theoretical computer science, including additive combinatorics [13], two-source extractors [14]–[16], and non-malleable encryption/commitment [17]–[19]. As we discuss in Section I-B, 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 the full version [20] for a discussion on the possible applicability of our techniques to other classes).

Remark 1 (On the efficiency of non-malleable codes)**.** A few prior 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 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

[1] In this paper we focus on constructing explicit, unconditional codes; see Section I-C 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.

[3] Although [12] 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 $2^{O(k/\log(k))}$.

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 encode 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. [21], [22]), 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.

B. Our Techniques

At a high level, we use the *non-malleable reduction* framework of Aggarwal et al. [13]. 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 $\mathcal{G}$). [13] 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 3 for a comparison of our approach to that of [12].)

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[4], which we use in conjunction with the pseudorandom switching lemma of Trevisan and Xue [23] 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 [11]. 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 [24]–[27], 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).[5] 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

[4]Recall that random restriction of a circuit randomly selects certain input wires of the circuit to fix to randomly chosen values. If the restriction is a pseudorandom pseudorandom, the random choices are performed pseudorandomly via a short seed (much smaller than the number of input wires to the circuit). Informally, we say the seed of pseudorandom restriction is *extractable* if there is an efficient procedure that recovers the seed from string formed by the values the restriction fixes to the chosen input wires and *any* setting of the wires unfixed by the restriction.

[5]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.

needed for classical switching lemmas. In particular, we invoke a recent pseudorandom switching lemma of Trevisan and Xue [23], which reduces DNFs to local functions (with parameters matching those of [27]) while only requiring that randomness specifying survivors and fixed values be σ-wise independent[6]. This allows us to avoid problems with independence arising in the naive solution above: We 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.

Leaky Local Functions to Split State.: Ball et al. [11] 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 [11], 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 the only significant difference with the construction in [11].) The presence of the seed allows us to determine the embedding locations in the absence of tampering and to simulate the 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 (Relation to the techniques of [12])**.** Although Chattopadhyay and Li [12] also use the switching lemma in their work, our overall approach is essentially orthogonal to theirs. At a high level, [12] uses a framework of Cheraghchi and Guruswami [3] 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 [3] framework) into

[6]Although this is not stated explicitly in [23], as we show, it follows immediately by combining their main lemma with results on bounded independence fooling CNF formulas [28], [29].

codes with at best exponentially small rate (see pg. 10 of [12] 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 [12] uses the classic switching lemma of Håstad [27] for fully random restrictions, whereas our work employs a recent extension of this switching lemma to pseudorandom restrictions [23].)

C. Related Work

Non-malleable codes were introduced by Dziembowski, Pietrzak, and Wichs [1], [2]. Various subsequent works re-formulated the definition [13], or considered extensions of the notion [30]–[33]. The original work of [1] 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 [34] presented constructions of non-malleable codes secure against split-state tampering. The original work and some subsequent works [35], [36] 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 [6], [8], [9], [13]. Among these works, the construction of Li [9] 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 [5], [37].

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. [38] 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 [1]. 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 [3] 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. [39] 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. [40] 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.

II. Preliminaries

A. 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 $i \le j \in [n]$, 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 $\varepsilon > 0$.

B. Non-malleable Reductions and Codes

Definition 1 (Coding Scheme)**.** [1] 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 [1]. Here we use a simpler, but equivalent, definition based on the following notion of non-malleable reduction by Aggarwal et al. [13].

Definition 2 (Non-Malleable Reduction)**.** [13] 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)**.** [13] 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)**.** *[13] If $(\mathcal{F} \Rightarrow \mathcal{G}, \varepsilon_1)$ and $(\mathcal{G} \Rightarrow \mathcal{H}, \varepsilon_2)$, then $(\mathcal{F} \Rightarrow \mathcal{H}, \varepsilon_1 + \varepsilon_2)$.*

C. Tampering Function Families

1) Split-State and Local Functions:

Definition 4 (Split-State Model)**.** [1] 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)**.** *[9] 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) 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 $i \in [m]$.

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 $(\boldsymbol{F}, h_1, \cdots, h_i, h)$.

Definition 7 (Binary Reconstructable Probabilistic Encoding)**.** [41], [42] We say a triple $(\mathrm{E}, \mathrm{D}, \mathrm{R})$ is a binary reconstructable probabilistic encoding scheme with parameters $(k, n, c_{\text{err}}, c_{\text{sec}})$, where $k, n \in \mathbb{N}$, $0 \leq c_{\text{err}}, c_{\text{sec}} < 1$, if it satisfies the following properties:

1) **Error correction.** E: $\{0,1\}^k \to \{0,1\}^n$ is an efficient probabilistic procedure, which maps a message $x \in \{0,1\}^k$ to a distribution over $\{0,1\}^n$. D corrects up to c_{err}-fraction of errors in codeword.
2) **Secrecy of partial views.** For all $x \in \{0,1\}^k$ and any non-empty set $S \subset [n]$ of size $\leq \lfloor c_{\text{sec}} \cdot n \rfloor$, $\mathrm{E}(x)_S$ 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 $\leq \lfloor c_{\text{sec}} \cdot n \rfloor$, 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. *[41], [42] 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 $(k, c_{rate}k, c_{err}, c_{sec})$.*

D. The Pseudorandom Switching Lemma of [23]

Definition 8. 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 9. 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 $(s,y) \leftarrow \mathcal{D}$, 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 5 (Polylogarithmic independence fools CNF formulas [28], [29])**.** *The class of M-clause CNF formulas is ε-fooled by $O((\log(M/\varepsilon))^2)$-wise independence.*

Theorem 6 (A Pseudorandom version of Håstad's switching lemma [23])**.** *Fix $p, \delta \in (0,1)$ and $w, S, t \in \mathbb{N}$. There exists a value $r \in \mathbb{N}$,*

$$r = \mathrm{poly}(t, w, \log(S), \log(1/\delta), \log(1/p)),^7$$

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

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

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

Proof: By Lemma 7 of [23], 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

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

where the probability is taken with respect to a pseudorandom restriction $\boldsymbol{\rho} \leftarrow \mathcal{R}(\mathcal{D}')$. By Theorem 5, the class of $M := (S \cdot 2^{w(\log(1/p)+1)})$-clause CNF formulas is fooled by r-wise independence where

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

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

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 $r \in \mathbb{N}$,*

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

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

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

III. NMC for Small-Depth Circuits

A. NM-Reducing Small-Depth Circuits to Leaky Local Functions

Lemma 2. *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

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

We define a simple encoding and decoding scheme (See Figure 1 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 8 & 9).

Lemma 3. *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

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

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

To prove Lemma 2, we instantiate Lemma 3 using the pseudorandom switching lemma of Theorem 6 (in fact, Corollary 1) and iteratively reduce $\mathrm{AC}_d(S)$ to leaky

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

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. We proceed to the formal proof.

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.
Let $(\mathrm{E}_R, \mathrm{D}_R, \mathrm{R}_R)$ denote the RPE from lemma 1 with codewords of length $m(s) \geq \sigma / c_{\text{sec}}$.
Let $\zeta^* \in \{0,1\}^{s(\sigma)}$ be some fixed string such that $\|L(G(\zeta^*))_{n-m+1,\ldots,n}\|_0 \geq k$. ($L$ is defined in Definition 8. 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, $\rho = (\rho^{(1)}, \rho^{(2)})$:
   Set $\rho^{(1)} = L(G(\zeta))$.
   $(*)$ If $\|L(G(\zeta))_{n-m+1:n}\|_0 < k$, set $\zeta = \zeta^*$.
   Set $\rho^{(2)} = \mathrm{E}_R(\zeta) \| U$.
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 $\|L(G(\tilde{\zeta}))_{n-m+1:n}\|_0 \leq k$, 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, $\rho = (\rho^{(1)}, \rho^{(2)})$:
   Set $\rho^{(1)} = L(G(\zeta))$.
   $(*)$ If $\|L(G(\zeta))_{n-m+1:n}\|_0 < k$, set $\zeta = \zeta^*$.
   Set $\rho^{(2)} = \mathrm{E}_R(\zeta) \| R$.
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) $h'(y'_1, \ldots, y'_{i+1}) := h(y'_1, \ldots, y'_i)_T$, where $T := \mathrm{ExtIndices}(L(G(\mathrm{D}_R(y'_{i+1}))))$.
7) Finally, output $\tau' = (\boldsymbol{F}', h'_1, \ldots, h'_{i+1}, h')$.

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

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

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

Proof: The second step of $\mathrm{E}^\star$ 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 $\|L(G(\zeta)\|_0 \geq k$, $\mathrm{D}^\star(\mathrm{E}^\star(x)) = \mathrm{Extract}(c, L(G(\zeta))) = x$ 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* $\delta \leq \Pr[\boldsymbol{F} \circ \mathrm{E}^\star$ *is not in* $\mathcal{G}]$.

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., $\Pr[\boldsymbol{F} \circ \mathrm{E}^\star$ is not in $\mathcal{G}]$.

∎

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

Proof: We first show $\mathcal{D} = G(\zeta) \| \mathrm{E}_R(\zeta) \| U$ is σ-wise independent when ζ and U are uniform. 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 $|S| \le \sigma$. By the secrecy property of the RPE and $m \cdot c_{\text{sec}} \ge \sigma$, 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 $\mathrm{E}^\star$ is distributed identically to $\mathcal{R}(\mathcal{D})$ (See Definition 9), except when ζ^* is used. Hence

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

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 $L(G(\zeta))_{n-m+1:n}$ 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 $L(G(\zeta))_{n-m+1:n}$. By linearity of expectation $\mu = (n-m)p$. For $k \le \mu/2$ and $\frac{\sigma}{\log(1/p)} \le \mu/8$, we can use Theorem 5 from [43] to conclude that $\|L(G(\zeta))_{n-m+1:n}\|_0 < k$ happens with probability at most $\exp(-\frac{\sigma}{2\log(1/p)})$. The desired conclusion follows. ∎

We now prove Lemma 2. 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. $(\mathrm{E}^\star, \mathrm{D}^\star)$ *non-malleably reduces* $(\mathrm{AC}_d(S)$ *to* $\mathrm{LL}^{1,m,n}[\mathrm{AC}_{d-2}(S) \circ t\text{-}\mathrm{AC}_2(2^t S)]$ *with error at most* ε.

Proof: Let $F \in \mathrm{AC}_d(S)$ be a boolean function. Note that Theorem 6 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 $n(S\left(2^{t+2}(5p)^t + \delta\right)) + \exp(-\Omega(\frac{\sigma}{\log(1/p)})) \le \varepsilon$.

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. $(\mathrm{E}^\star, \mathrm{D}^\star)$ *non-malleably reduces* $\mathrm{LL}^{i,m,n}[\mathrm{AC}_{d-i-1}(S) \circ t\text{-}\mathrm{AC}_2(2^t S)]$ *to* $\mathrm{LL}^{i+1,m,n}[\mathrm{AC}_{d-i-2}(S) \circ t\text{-}\mathrm{AC}_2(2^t S)]$ *with error at most* ε

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 3, $(\mathrm{E}^\star, \mathrm{D}^\star)$ 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. $(\mathrm{E}^\star, \mathrm{D}^\star)$ *non-malleably reduces* $\mathrm{LL}^{d-1,m,n}[t\text{-}\mathrm{AC}_2(2^t S)]$ *to* $\mathrm{LL}^{d,m,n}[\mathrm{Local}^{2^t}]$ *with error at most* ε.

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 3, $(\mathrm{E}^\star, \mathrm{D}^\star)$ 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 $n' \ge k \ge 2m$, Lemma 3 holds for messages of length $(n'-m)p/2 \ge n'(p/4)$. Therefore, the composed reduction works for any $2m \le k \le n(p/4)^d$.

B. NM-Reducing Leaky Local to Split State

Simple modifications to construction from the appendix of [11] yield a $(\mathrm{LL}^{d,s,N}[\mathrm{Local}^\ell], \mathrm{SS}_k, \mathrm{negl}(k))$-non-malleable reduction.

Lemma 4. *There exists a constant* $c \in (0,1)$*, such that for any* m, q, ℓ *satisfying* $mq\ell^3 \le cn$ *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. The construction and the proof of Lemma 4 can be found in the full version [20].

C. Putting It All Together

In this section, we show our main results. By composing the non-malleable reductions from Lemma 2 and Lemma 4, we obtain a non-malleable reduction which reduces small-depth circuits to split state.

Lemma 5. *For $S, d, n, \ell \in \mathbb{N}$, $p, \delta \in (0,1)$, there exists $\sigma = \mathrm{poly}(\log \ell, \log(\ell S), \log(1/\delta), \log(1/p))$ such that for k that $k \geq O(\sigma \log n)$ and $k = \Omega(n(p/4)^d/\ell^2)$,*

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

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. AC^0), we obtain the following corollary by setting $\ell = n^{1/\log\log\log n}$, $\delta = n^{-\log\log n}$ and $p = (\log \ell \cdot \log n)^{-1} = \frac{\log\log\log n}{\log^2 n}$,

Corollary 2. $\left(\mathrm{AC0} \implies \mathrm{SS}_k, n^{-(\log\log n)^{1-o(1)}}\right)$ *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 7. *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 $n = k^{1+c}$.*

Acknowledgment

The first and fourth authors are 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. 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. The first author is additionally supported by ISF grant no. 1790/13 and the Check Point Institute for Information Security. The second author is 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. The third author is supported by NSF grants CNS1314722 and CNS-1413964. Part of this research was done while the third author was visiting the FACT Center at IDC Herzliya. The fifth author is supported by NSF grant CCF 1563122.

References

[1] S. Dziembowski, K. Pietrzak, and D. Wichs, "Non-malleable codes," in *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, A. C. Yao, Ed. Tsinghua University Press, 2010, pp. 434–452.

[2] ——, "Non-Malleable Codes," *Journal of the ACM*, 2018.

[3] M. Cheraghchi and V. Guruswami, "Capacity of non-malleable codes," *IEEE Trans. Information Theory*, vol. 62, no. 3, pp. 1097–1118, 2016.

[4] S. Dziembowski, T. Kazana, and M. Obremski, "Non-malleable codes from two-source extractors," in *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, ser. Lecture Notes in Computer Science, R. Canetti and J. A. Garay, Eds., vol. 8043. Springer, 2013, pp. 239–257.

[5] E. Chattopadhyay and D. 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*. IEEE Computer Society, 2014, pp. 306–315.

[6] D. Aggarwal, Y. Dodis, and S. Lovett, "Non-malleable codes from additive combinatorics," in *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, D. B. Shmoys, Ed. ACM, 2014, pp. 774–783.

[7] D. Aggarwal, "Affine-evasive sets modulo a prime," *Inf. Process. Lett.*, vol. 115, no. 2, pp. 382–385, 2015.

[8] X. Li, "Improved non-malleable extractors, non-malleable codes and independent source extractors," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, H. Hatami, P. McKenzie, and V. King, Eds. ACM, 2017, pp. 1144–1156.

[9] ——, "Pseudorandom correlation breakers, independence preserving mergers and their applications," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 25, p. 28, 2018. [Online]. Available: https://eccc.weizmann.ac.il/report/2018/028

[10] S. Agrawal, D. Gupta, H. K. Maji, O. Pandey, and M. 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*, 2015, pp. 538–557.

[11] M. Ball, D. Dachman-Soled, M. Kulkarni, and T. 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*, 2016, pp. 881–908.

[12] E. Chattopadhyay and X. 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*, 2017, pp. 1171–1184.

[13] D. Aggarwal, Y. Dodis, T. Kazana, and M. Obremski, "Non-malleable reductions and applications," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, R. A. Servedio and R. Rubinfeld, Eds. ACM, 2015, pp. 459–468.

[14] X. 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*. IEEE Computer Society, 2012, pp. 688–697.

[15] ——, "New independent source extractors with exponential improvement," in *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, D. Boneh, T. Roughgarden, and J. Feigenbaum, Eds. ACM, 2013, pp. 783–792.

[16] E. Chattopadhyay and D. Zuckerman, "Explicit two-source extractors and resilient functions," in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, D. Wichs and Y. Mansour, Eds. ACM, 2016, pp. 670–683.

[17] S. Coretti, U. Maurer, B. Tackmann, and D. 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*, 2015, pp. 532–560.

[18] S. Coretti, Y. Dodis, B. Tackmann, and D. 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*, 2016, pp. 306–335.

[19] V. Goyal, O. Pandey, and S. 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*, 2016, pp. 1128–1141.

[20] M. Ball, D. Dachman-Soled, S. Guo, T. Malkin, and L. Tan, "Non-malleable codes for small-depth circuits," *IACR Cryptology ePrint Archive*, vol. 2018, p. 207, 2018. [Online]. Available: http://eprint.iacr.org/2018/207

[21] M. M. Klawe, W. J. Paul, N. Pippenger, and M. Yannakakis, "On monotone formulae with restricted depth (preliminary version)," in *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, R. A. DeMillo, Ed. ACM, 1984, pp. 480–487.

[22] L. G. Valiant, "Exponential lower bounds for restricted monotone circuits," in *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, D. S. Johnson, R. Fagin, M. L. Fredman, D. Harel, R. M. Karp, N. A. Lynch, C. H. Papadimitriou, R. L. Rivest, W. L. Ruzzo, and J. I. Seiferas, Eds. ACM, 1983, pp. 110–117.

[23] L. Trevisan and T. 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*. IEEE Computer Society, 2013, pp. 242–247.

[24] M. L. Furst, J. B. Saxe, and M. Sipser, "Parity, circuits, and the polynomial-time hierarchy," *Mathematical Systems Theory*, vol. 17, no. 1, pp. 13–27, 1984. [Online]. Available: https://doi.org/10.1007/BF01744431

[25] M. Ajtai, "First-order definability on finite structures," *Ann. Pure Appl. Logic*, vol. 45, no. 3, pp. 211–225, 1989. [Online]. Available: https://doi.org/10.1016/0168-0072(89)90036-5

[26] A. C. 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*, 1985, pp. 1–10. [Online]. Available: https://doi.org/10.1109/SFCS.1985.49

[27] J. 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*, 1986, pp. 6–20. [Online]. Available: http://doi.acm.org/10.1145/12130.12132

[28] L. M. J. Bazzi, "Polylogarithmic independence can fool DNF formulas," *SIAM J. Comput.*, vol. 38, no. 6, pp. 2220–2272, 2009.

[29] A. Razborov, "A simple proof of bazzis theorem," *ACM Transactions on Computation Theory (TOCT)*, vol. 1, no. 1, p. 3, 2009.

[30] S. Faust, P. Mukherjee, J. B. Nielsen, and D. 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*, 2014, pp. 465–488. [Online]. Available: https://doi.org/10.1007/978-3-642-54242-8_20

[31] D. Dachman-Soled, F. Liu, E. Shi, and H. 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*, 2015, pp. 427–450. [Online]. Available: https://doi.org/10.1007/978-3-662-46494-6_18

[32] E. Chattopadhyay, V. Goyal, and X. 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*, 2016, pp. 285–298. [Online]. Available: http://doi.acm.org/10.1145/2897518.2897547

[33] N. Chandran, V. Goyal, P. Mukherjee, O. Pandey, and J. Upadhyay, "Block-wise non-malleable codes," in *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, 2016, pp. 31:1–31:14. [Online]. Available: https://doi.org/10.4230/LIPIcs.ICALP.2016.31

[34] F. Liu and A. 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*, 2012, pp. 517–532.

[35] D. Aggarwal, S. Agrawal, D. Gupta, H. K. Maji, O. Pandey, and M. 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*, 2016, pp. 393–417. [Online]. Available: https://doi.org/10.1007/978-3-662-49099-0_15

[36] A. Kiayias, F. Liu, and Y. 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*, 2016, pp. 1317–1328. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978352

[37] B. Kanukurthi, L. Obbattu, and S. 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.

[38] S. Faust, P. Mukherjee, D. Venturi, and D. 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*, 2014, pp. 111–128. [Online]. Available: https://doi.org/10.1007/978-3-642-55220-5_7

[39] S. Faust, K. Hostáková, P. Mukherjee, and D. 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*, 2017, pp. 95–126. [Online]. Available: https://doi.org/10.1007/978-3-319-63715-0_4

[40] M. Ball, D. Dachman-Soled, M. Kulkarni, and T. Malkin, "Non-malleable codes from average-case hardness: $${\mathsf {A}}{\mathsf {C}}^0$$, decision trees, and streaming space-bounded tampering," in *EUROCRYPT (3)*, ser. Lecture Notes in Computer Science, vol. 10822. Springer, 2018, pp. 618–650.

[41] S. G. Choi, D. Dachman-Soled, T. Malkin, and H. Wee, "A black-box construction of non-malleable encryption from semantically secure encryption," *J. Cryptology*, vol. 31, no. 1, pp. 172–201, 2018, extended abstract appeared in TCC 2008.

[42] ——, "Improved, black-box, non-malleable encryption from semantic security," *Des. Codes Cryptography*, vol. 86, no. 3, pp. 641–663, 2018.

[43] J. P. Schmidt, A. Siegel, and A. Srinivasan, "Chernoff-hoeffding bounds for applications with limited independence," *SIAM J. Discrete Math.*, vol. 8, no. 2, pp. 223–250, 1995.