Non-Malleable Codes, Extractors and Secret Sharing for Interleaved Tampering and Composition of Tampering

Eshan Chattopadhyay[1] and Xin Li[2]

[1] Cornell University, Ithaca NY 14853, USA
eshan@cs.cornell.edu
[2] Johns Hopkins University, Baltimore MD 21218, USA
lixints@cs.jhu.edu

Abstract. Non-malleable codes were introduced by Dziembowski, Pietrzak, and Wichs (JACM 2018) as a generalization of standard error correcting codes to handle severe forms of tampering on codewords. This notion has attracted a lot of recent research, resulting in various explicit constructions, which have found applications in tamper-resilient cryptography and connections to other pseudorandom objects in theoretical computer science. We continue the line of investigation on explicit constructions of non-malleable codes in the information theoretic setting, and give explicit constructions for several new classes of tampering functions. These classes strictly generalize several previously studied classes of tampering functions, and in particular extend the well studied split-state model which is a "compartmentalized" model in the sense that the codeword is partitioned *a prior* into disjoint intervals for tampering. Specifically, we give explicit non-malleable codes for the following classes of tampering functions.

- Interleaved split-state tampering: Here the codeword is partitioned in an unknown way by an adversary, and then tampered with by a split-state tampering function.
- Affine tampering composed with split-state tampering: In this model, the codeword is first tampered with by a split-state adversary, and then the whole tampered codeword is further tampered with by an affine function. In fact our results are stronger, and we can handle affine tampering composed with interleaved split-state tampering.

Our results are the first explicit constructions of non-malleable codes in any of these tampering models. As applications, they also directly give non-malleable secret-sharing schemes with *binary shares* in the split-state joint tampering model and the stronger model of affine tampering composed with split-state joint tampering. We derive all these results from explicit constructions of seedless non-malleable extractors, which we believe are of independent interest.

Using our techniques, we also give an improved seedless extractor for an unknown interleaving of two independent sources.

Keywords: non-malleable code · tamper-resilient cryptography · extractor

1 Introduction

1.1 Non-malleable Codes

Non-malleable codes were introduced by Dziembowski, Pietrzak, and Wichs [36] as an elegant relaxation and generalization of standard error correcting codes, where the motivation is to handle much larger classes of tampering functions on the codeword. Traditionally, error correcting codes only provide meaningful guarantees (e.g., unique decoding or list-decoding) when *part* of the codeword is modified (i.e., the modified codeword is close in Hamming distance to an actual codeword), whereas in practice an adversary can possibly use much more complicated functions to modify the entire codeword. In the latter case, it is easy to see that error correction or even error detection becomes generally impossible, for example an adversary can simply change all codewords into a fixed string. On the other hand, non-malleable codes can still provide useful guarantees here, and thus partially bridge this gap. Informally, a non-malleable code guarantees that after tampering, the decoding either correctly gives the original message or gives a message that is completely unrelated and independent of the original message. This captures the notion of non-malleability: that an adversary cannot modify the codeword in a way such that the tampered codeword decodes back to a related but different message.

The original intended application of non-malleable codes is in tamper-resilient cryptography [36], where they can be used generally to prevent an adversary from learning secret information by observing the input/output behavior of modified ciphertexts. Subsequently, non-malleable codes have found applications in non-malleable commitments [40], non-malleable encryption [30], public-key encryptions [31], non-malleable secret-sharing schemes [38], and privacy amplification protocols [19]. Furthermore, interesting connections were found to non-malleable extractors [27], and very recently to spectral expanders [54]. Along the way, the constructions of non-malleable codes used various components and sophisticated ideas from additive combinatorics [5, 22] and randomness extraction [18], and some of these techniques have also found applications in constructing extractors for independent sources [46]. As such, non-malleable codes have become fundamental objects at the intersection of coding theory and cryptography. They are well deserved to be studied in more depth in their own right, as well as to find more connections to other well studied objects in theoretical computer science.

We first introduce some notation before formally defining non-malleable codes. For a function $f : S \to S$, we say $s \in S$ is a fixed point (of f) if $f(s) = s$.

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

We use the statistical distance to measure the distance between distributions.

Definition 2 *The statistical distance between two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ over some universal set Ω is defined as $|\mathcal{D}_1 - \mathcal{D}_2| = \frac{1}{2}\sum_{d\in\Omega} |\mathbf{Pr}[\mathcal{D}_1 = d] - \mathbf{Pr}[\mathcal{D}_2 = d]|$. We say $\mathcal{D}_1$ is ϵ-close to $\mathcal{D}_2$ if $|\mathcal{D}_1 - \mathcal{D}_2| \leq \epsilon$ and denote it by $\mathcal{D}_1 \approx_\epsilon \mathcal{D}_2$.*

To introduce non-malleable codes, we need to define a function called copy that takes in two inputs. If the first input is the special symbol "$same^\star$", the copy function just outputs its second input. Else it outputs its first input. This is useful in defining non-malleable codes where one wants to model the situation that the decoding of the tampered codeword is either the original message or a distribution independent of the message. Thus, we define a distribution on the message space and the special symbol $same^\star$, where the probability that the distribution takes on the value $same^\star$ corresponds to the probability that the tampered codeword is decoded back to the original message. More formally, we have

$$\text{copy}(x, y) = \begin{cases} x & \text{if } x \neq same^\star \\ y & \text{if } x = same^\star \end{cases}$$

Following the treatment in [36], we first define coding schemes.

Definition 3 (Coding schemes) *Let* Enc $: \{0,1\}^k \to \{0,1\}^n$ *and* Dec $: \{0,1\}^n \to \{0,1\}^k \cup \{\perp\}$ *be functions such that* Enc *is a randomized function (i.e., it has access to private randomness) and* Dec *is a deterministic function. We say that* (Enc, Dec) *is a coding scheme with block length n and message length k if for all $s \in \{0,1\}^k$,* $\Pr[\text{Dec}(Enc(s)) = s] = 1$*, where the probability is taken over the randomness in* Enc*.*

We can now define non-malleable codes.

Definition 4 (Non-malleable codes) *A coding scheme $\mathcal{C} =$* (Enc, Dec) *with block length n and message length k is a non-malleable code with respect to a family of tampering functions $\mathcal{F} \subset \mathcal{F}_n$ and error ϵ if for every $f \in \mathcal{F}$ there exists a random variable D_f on $\{0,1\}^k \cup \{same^\star\}$ which is independent of the randomness in* Enc *and is efficiently samplable given oracle access to $f(.)$, such that for all messages $s \in \{0,1\}^k$, it holds that*

$$|\text{Dec}(f(\text{Enc}(s))) - \text{copy}(D_f, s)| \leq \epsilon.$$

We say the code is explicit if both the encoding and decoding can be done in polynomial time. The rate of $\mathcal{C}$ is given by k/n.

Relevant prior work on non-malleable codes in the information theoretic setting. There has been a lot of exciting research on non-malleable codes, and it is beyond the scope of this paper to provide a comprehensive survey of them. Instead we focus on relevant explicit (unconditional) constructions in the information theoretic setting, which is also the focus of this paper. One of the most studied classes of tampering functions is the so called *split-state* tampering, where the codeword is divided into (at least two) disjoint intervals and

the adversary can tamper with each interval arbitrarily but independently. This model arises naturally in situations where the codeword may be stored in different parts of memory or different devices. Following a very successful line of work [1, 2, 4, 5, 7, 18, 22, 27, 34, 41, 43, 44, 46, 47], we now have explicit constructions of non-malleable codes in the 2-split state model with constant rate and negligible error.

The split state model is a "compartmentalized" model, where the codeword is partitioned *a priori* into disjoint intervals for tampering. Recently, there has been progress towards handling non-compartmentalized tampering functions. A work of Agrawal, Gupta, Maji, Pandey and Prabhakaran [8] gave explicit constructions of non-malleable codes with respect to tampering functions that permute or flip the bits of the codeword. Ball, Dachman-Soled, Kulkarni and Malkin [12] gave explicit constructions of non-malleable codes against t-local functions for $t \leq n^{1-\epsilon}$. However in all these models, each bit of the tampering function only depends on part of the codeword. A recent work of Chattopadhyay and Li [21] gave the first explicit constructions of non-malleable codes where each bit of the tampering function may depend on all bits of the codeword. Specifically, they gave constructions for the classes of affine functions and small-depth (unbounded fain-in) circuits. The rate of the non-malleable code with respect to small-depth circuits was exponentially improved by a subsequent work of Ball, Dachman-Soled, Guo, Malkin, and Tan [11]. In a recent work, Ball, Guo and Wichs [13] constructed non-malleable codes with respect to bounded depth decision trees.

Given all these exciting results, a major goal of the research on non-malleable codes remains to give explicit constructions for broader classes of tampering functions, as one can use the probabilistic method to show the existence of non-malleable codes with rate close to $1 - \delta$ for any class $\mathcal{F}$ of tampering functions with $|\mathcal{F}| \leq 2^{2^{\delta n}}$ [26].

Our results. We continue the line of investigation on explicit constructions of non-malleable codes, and give explicit constructions for several new classes of non-compartmentalized tampering functions, where in some classes each bit of the tampering function can depend on all the bits of the codeword. In Section 1.2, we discuss motivations and applications of our new non-malleable codes in cryptography. The new classes strictly generalize several previous studied classes of tampering functions. In particular, we consider the following classes.

1. *Interleaved* 2*-split-state tampering*, where the adversary can divide the codeword into two arbitrary disjoint intervals and tamper with each interval arbitrarily but independently. This model generalizes the split-state model and captures the situation where the codeword is partitioned into two blocks (not necessarily of the same length) in an unknown way by the adversary before applying a 2-split-state tampering function. Constructing non-malleable codes for this class of tampering functions was left as an open problem by Cheraghchi and Guruswami [27].
2. *Composition of tampering*, where the adversary takes two tampering functions and composes them together to get a new tampering function. We note

that function composition is a natural strategy for an adversary to achieve more powerful tampering, and it has been studied widely in other fields (e.g., computational complexity and communication complexity). We believe that studying non-malleable codes for the composition of different classes of tampering functions is also a natural and important direction.

We now formally define these classes and some related classes below. For notation, given any permutation $\pi : [n] \to [n]$ and any string x of length n, we let $y = x_\pi$ denote the length n string such that $y_{\pi(i)} = x_i$.

- The family of 2-split-state functions 2SS $\subset \mathcal{F}_{2n}$: Any $f \in$ 2SS comprises of two functions $f_1 : \{0,1\}^n \to \{0,1\}^n$ and $f_2 : \{0,1\}^n \to \{0,1\}^n$, and for any $x, y \in \{0,1\}^n$, $f(x,y) = (f_1(x), f_2(y))$. This family of tampering functions has been extensively studied, with a long line of work achieving near optimal explicit constructions of non-malleable codes.
- The family of affine functions Lin $\subset \mathcal{F}_n$: Any $f \in$ Lin is an affine function from $\{0,1\}^n$ to $\{0,1\}^n$ (viewing $\{0,1\}^n$ as $\mathbb{F}_2^n$), i.e., $f(x) = Mx + v$, for some $n \times n$ matrix M on $\mathbb{F}_2$ and $v \in \mathbb{F}_2^n$.
- The family of interleaved 2-split-state functions $(2,t)$-ISS $\subset \mathcal{F}_n$: Any $f \in (2,t)$-ISS comprises of two functions $f_1 : \{0,1\}^{n_1} \to \{0,1\}^{n_1}$, $f_2 : \{0,1\}^{n_2} \to \{0,1\}^{n_2}$ such that $n_1 + n_2 = n$ and $\min\{n_1, n_2\} \geq t$ (i.e both partitions are of length at least t), and a permutation $\pi : [n] \to [n]$. For any $z = (x,y)_\pi \in \{0,1\}^n$, where $x \in \{0,1\}^{n_1}, y \in \{0,1\}^{n_2}$, let $f(z) = (f_1(x), f_2(y))_\pi$. In this paper we require that $t \geq n^\beta$ for some fixed constant $0 < \beta < 1$. Note this includes as a special case the situation where the two states have the same size, which we denote by 2ISS, and in particular 2SS.
- For any tampering function families $\mathcal{F}, \mathcal{G} \subset \mathcal{F}_n$, define the family $\mathcal{F} \circ \mathcal{G} \subset \mathcal{F}_n$ to be the set of all functions of the form $f \circ g$, where $f \in \mathcal{F}$, $g \in \mathcal{G}$ and $\circ$ denotes function composition.

We now formally state our results. Our most general result is an explicit non-malleable code with respect to the tampering class of Lin $\circ$ $(2, n^\beta)$-ISS, i.e, an affine function composed with an interleaved 2-split-state tampering function. Specifically, we have the following theorem.

Theorem 5 *There exist constants $\beta, \delta > 0$ such that for all integers $n > 0$ there exists an explicit non-malleable code with respect to* Lin $\circ$ $(2, n^\beta)$-ISS *with rate $1/n^\delta$ and error 2^{-n^δ}.*

We immediately have the following corollary, which records the classes of functions for which no explicit non-malleable codes were known (for any rate) prior to this work.

Corollary 1. *There exist constants $\beta, \delta > 0$ such that for all integers $n > 0$ there exists an explicit non-malleable code with respect to the following classes of functions with rate $1/n^\delta$ and error 2^{-n^δ}:*

- 2ISS, $(2, n^\beta)$-ISS, Lin $\circ$ 2ISS *and* Lin $\circ$ 2SS.

1.2 Motivations and applications in cryptography

Just as standard non-malleable codes for split-state tampering arise from natural cryptographic applications, our non-malleable codes for interleaved 2-split-state tampering and affine tampering composed with interleaved split-state tampering also have natural cryptographic motivations and applications.

It is known that any non-malleable code in the 2-split-state model gives a 2 out of 2 secret-sharing scheme, if one views the two split states as two shares [6]. We show that any non-malleable code in the interleaved 2-split state model gives a *non-malleable secret-sharing* scheme with *binary shares*. Secret-sharing schemes [14, 58] are fundamental objects in cryptography, and building blocks for many other more advanced applications such as secure multiparty computation. In short, a secret-sharing scheme shares a message secretly among n parties, such that any qualified subset can reconstruct the message, while any unqualified subset reveals nothing (or almost nothing) about the message. Equivalently, one can view this as saying that any leakage function which leaks the shares in an unqualified subset reveals nothing. In the standard threshold or t out of n secret-sharing, any subset of size at most t is an unqualified subset while any subset of size larger than t is a qualified subset. However, it is known that in such a scheme, the share size has to be at least as large as the message size. Thus, a natural and interesting question is whether the share size can be smaller under some relaxed notion of secret-sharing. This is indeed possible when one considers the notion of (r, t)-*ramp* secret-sharing, where $r > t + 1$. In this setting, any subset of size at most t reveals nothing about the message, while any subset of size at least r can reconstruct message. Thus t is called the privacy threshold and r is called the reconstruction threshold. Subsets of size between $t + 1$ and $r - 1$ may reveal some partial information about the message. Again, it is not hard to see that the share size in this case has to be at least as large as $m/(r - t)$, where m is the message length. Thus, if one allows a sufficiently large gap between r and t, then it is possible to achieve a secret-sharing scheme even with binary shares.

Secret-sharing schemes are also closely related to error correcting codes. For example, the celebrated Shamir's scheme [58] is based on Reed-Solomon codes. Similarly, binary secret-sharing schemes are largely based on binary error correcting codes, and they are studied in a series of recent works [15, 16, 25, 48] in terms of the tradeoff between the message length, the privacy threshold t, the reconstruction threshold r, and the complexity of the sharing and reconstruction functions.

However, standard secret-sharing schemes only allow an adversary to passively observe some shares, thus one can ask the natural question of whether it is possible to protect against even active adversaries who can tamper with the shares. In this context, the notion of *robust* secret-sharing schemes (e.g., [17,51]) allows qualified subsets to recover the message even if the adversary can modify *part* of the shares. More recently, by generalizing non-malleable codes, Goyal and Kumar [38] introduced non-malleable secret-sharing schemes, where the adversary can tamper with *all* shares in some restricted manner. Naturally, the

guarantee is that if tampering happens, then the reconstructed message is either the original message or something completely unrelated. In particular, they constructed t out of n non-malleable secret-sharing schemes in the following two tampering models. In the independent tampering model, the adversary can tamper with each share independently. In the joint tampering model, the adversary can divide any subset of $t+1$ shares arbitrarily into two sets of *different size*, and tamper with the shares in each set jointly, but independently across the two sets. Note that the adversary in the second model is strictly stronger than the adversary in the first one, since for reconstruction one only considers subsets of size $t+1$. Several follow up works [3, 9, 39] studied different models such as non-malleable secret-sharing schemes for general access structures, and achieved improvements in various parameters.

However, in all known constructions of non-malleable secret-sharing schemes the share size is always larger than 1 bit. In other words, no known non-malleable secret-sharing scheme can achieve binary shares. This is an obstacle that results from the techniques in all known constructions. Indeed, even if one allows (r,t)-*ramp* non-malleable secret-sharing with an arbitrarily large gap between r and t, no known constructions can achieve binary shares, because they all need to put at least *two* shares of some standard secret-sharing schemes together to form a single share in the non-malleable scheme. Thus it is a natural question to see if one can construct non-malleable secret-sharing schemes with binary shares using different techniques.

Our non-malleable codes for interleaved 2-split-state tampering directly give non-malleable secret-sharing schemes with binary shares that protect against joint tampering. We have the following theorem.

Theorem 6 *There exist constants $0 < \alpha < \beta < 1$ such that for all integers $n > 0$ there exists an explicit (r,t)-ramp non-malleable secret-sharing scheme with binary shares, where $r = n$, $t = n - n^{\beta}$ and the message length is n^{α}. The scheme has statistical privacy with error $2^{-n^{\Omega(1)}}$, and is resilient with error $2^{-n^{\Omega(1)}}$ to joint tampering where the adversary arbitrarily partitions the r shares into two blocks, each with at most t shares, and tampers with each block independently using an arbitrary function.*

Intuitively, any n-bit non-malleable code for interleaved 2-split-state tampering gives a ramp non-malleable secret-sharing scheme with reconstruction threshold $r = n$, as follows. If the code protects against an adversary who can partition the codeword into two disjoint sets and tamper with each set arbitrarily but independently, then each set must reveal (almost) nothing about the secret message. Otherwise, the adversary can simply look at one set and use the leaked information to modify the shares in this set, and make the reconstructed message become a different but related message. In particular, the same proof in [6] for the standard 2-split state model also works for the interleaved 2-split state model. Since our code works for interleaved 2-split-state tampering and the size of one set can be as large as $n - n^{\beta}$, this implies privacy threshold at least $n - n^{\beta}$,

with the small error in privacy coming from the error of the non-malleable code. We refer the reader to the full version of our paper for more details.

It is an interesting open question to construct explicit non-malleable secret-sharing schemes with binary shares where the reconstruction threshold $r < n$. We note that this question is closely related to constructing non-malleable codes for the tampering class 2SS$\circ$Lin or 2ISS$\circ$Lin (i.e., reverse the order of composition). This is because to get such a scheme, one natural idea is to apply another secret-sharing scheme on top of our non-malleable code. If one uses a linear secret-sharing scheme as in many standard schemes, then the tampering function on the codeword becomes 2SS $\circ$ Lin or 2ISS $\circ$ Lin.

We also note that in an (r, t)-ramp secret-sharing scheme with binary shares, unless the message has only one bit, we must have $r > t + 1$. Thus in the joint tampering model, instead of allowing the adversary to divide r shares arbitrarily into two sets, one must put an upper bound t on the size of each set as in our theorem. For example, one cannot allow an adversary to look at a set of shares with size $r - 1$, because $r - 1 > t$ and this set of shares may already leak some information about the secret message.

In both standard secret-sharing and non-malleable secret-sharing, in addition to looking at sets of shares, researchers have also studied other classes of leakage function or tampering function. For example, the work of Goyal et al. [37] studied secret-sharing schemes that are resilient to affine leakage functions on all shares, and used them to construct parity resilient circuits and bounded communication leakage resilient protocols. A recent work of Lin et. al [49] also studied non-malleable secret-sharing schemes where the adversary can tamper with *all* shares jointly using some restricted classes of functions. Specifically, [49] considered the model of "adaptive" affine tampering, where the adversary is allowed to first observe the shares in some unqualified subset, and then choose an affine function based on this to tamper with all shares. In this sense, our non-malleable codes for affine tampering composed with interleaved 2-split-state tampering also directly give non-malleable secret-sharing schemes with binary shares that protect against affine tampering composed with joint tampering, which is strictly stronger than both the joint tampering model and the affine tampering model (although our affine tampering is non-adaptive compared to [49]). Specifically, we have the following theorem (which strictly generalizes Theorem 6).

Theorem 7 *There exist constants $0 < \alpha < \beta < 1$ such that for all integers $n > 0$ there exists an explicit (r, t)-ramp non-malleable secret-sharing scheme with binary shares, where $r = n$, $t = n - n^{\beta}$ and the message length is n^{α}. The scheme has statistical privacy with error $2^{-n^{\Omega(1)}}$, and is resilient with error $2^{-n^{\Omega(1)}}$ to an adversary that tampers in two stages: In the first stage, the adversary partitions the r shares arbitrarily into two blocks, each with at most t shares, and tampers with each block independently using an arbitrary function. In the second stage, the adversary applies an arbitrary affine tampering function jointly on all the already tampered (from the first stage) r shares.*

We provide a formal proof of the above theorem in the full version of our paper.

Again, it is an interesting open question to construct explicit non-malleable secret-sharing schemes where the order of tampering is reversed.

1.3 Seedless non-malleable extractors

Our results on non-malleable codes are based on new constructions of seedless non-malleable extractors, which we believe are of independent interest. Before defining seedless non-malleable extractors formally, we first recall some basic notation from the area of randomness extraction.

Randomness extraction is motivated by the problem of purifying imperfect (or defective) sources of randomness. The concern stems from the fact that natural random sources often have poor quality, while most applications require high quality (e.g., uniform) random bits. We use the standard notion of min-entropy to measure the amount of randomness in a distribution.

Definition 8 *The min-entropy $H_\infty(\mathbf{X})$ of a probability distribution $\mathbf{X}$ on $\{0,1\}^n$ is defined to be $\min_x(-\log(\Pr[\mathbf{X}=x]))$. We say $\mathbf{X}$ is an $(n, H_\infty(\mathbf{X}))$-source and the min-entropy rate is $H_\infty(\mathbf{X})/n$.*

It turns out that it is impossible to extract from a single general weak random source even for min-entropy $n-1$. There are two possible ways to bypass this barrier. The first one is to relax the extractor to be a *seeded extractor*, which takes an additional independent short random seed to extract from a weak random source. The second one is to construct deterministic extractors for special classes of weak random sources.

Both kinds of extractors have been studied extensively. Recently, they have also been generalized to stronger notions where the inputs to the extractor can be tampered with by an adversary. Specifically, Dodis and Wichs [33] introduced the notion of *seeded non-malleable extractor* in the context of privacy amplification against an active adversary. Informally, such an extractor satisfies the stronger property that the output of the extractor is independent of the output of the extractor on a tampered seed. Similarly, and more relevant to this paper, a seedless variant of non-malleable extractors was introduced by Cheraghchi and Guruswami [27] as a way to construct non-malleable codes. Apart from their original applications, both kinds of non-malleable extractors are of independent interest. They are also related to each other and have applications in constructions of extractors for independent sources [46].

We now define seedless non-malleable extractors.

Definition 9 (Seedless non-malleable extractors) *Let $\mathcal{F} \subset \mathcal{F}_n$ be a family of tampering functions such that no function in $\mathcal{F}$ has any fixed points. A function $\mathrm{nmExt} : \{0,1\}^n \to \{0,1\}^m$ is a seedless (n, m, ϵ)-non-malleable extractor with respect to $\mathcal{F}$ and a class of sources $\mathcal{X}$ if for every distribution $\mathbf{X} \in \mathcal{X}$ and*

every tampering function $f \in \mathcal{F}$, there exists a random variable that is $D_{f,X}$ on $\{0,1\}^m \cup \{same^\star\}$ that is independent of $\mathbf{X}$, such that

$$|\text{nmExt}(\mathbf{X}), \text{nmExt}(f(\mathbf{X})) - \mathbf{U}_m, \text{copy}(D_{f,X}, \mathbf{U}_m)| \leq \epsilon.$$

Further, we say that nmExt *is ϵ'-invertible, if there exists a polynomial time sampling algorithm $\mathcal{A}$ that takes as input $y \in \{0,1\}^m$, and outputs a sample from a distribution that is ϵ'-close to the uniform distribution on the set* $\text{nmExt}^{-1}(y)$.

In the above definition, when the class of sources $\mathcal{X}$ is the distribution $\mathbf{U}_n$, we simply say that nmExt is a seedless (n, m, ϵ)-non-malleable extractor with respect to $\mathcal{F}$.

Relevant prior work on seedless non-malleable extractors. The first construction of seedless non-malleable extractors was given by Chattopadhyay and Zuckerman [22] with respect to the class of 10-split-state tampering. Subsequently, a series of works starting with the work of Chattopadhyay, Goyal and Li [18] gave explicit seedless non-malleable extractors for 2-split-state tampering. The only known constructions with respect to a class of tampering functions different from split state tampering is from the work of Chattopadhyay and Li [21], which gave explicit seedless non-malleable extractors with respect to the tampering class Lin and small depth circuits, and a subsequent follow-up work of Ball et al. [10] where they constructed non-malleable extractors against tampering functions that are low-degree polynomials over large fields. We note that constructing explicit seedless non-malleable extractors with respect to 2ISS was also posed as an open problem in [27].

Our results. As our most general result, we give the first explicit constructions of seedless non-malleable extractors with respect to the tampering class Lin $\circ$ $(2, n^\beta)$-ISS.

Theorem 10 *There exists a constant $\beta > 0$ such that for all $n > 0$ there exists an efficiently computable seedless $(n, n^{\Omega(1)}, 2^{-n^{\Omega(1)}})$-non-malleable extractor with respect to* Lin $\circ$ $(2, n^\beta)$-ISS*, that is $2^{-n^{\Omega(1)}}$-invertible.*

This immediately yields the first explicit non-malleable extractors against the following classes of tampering functions.

Corollary 2. *For all $n > 0$ there exists an efficiently computable seedless $(n, n^{\Omega(1)}, 2^{-n^{\Omega(1)}})$-non-malleable extractor with respect to the following classes of tampering functions:*

- 2ISS*,* $(2, n^\beta)$-ISS*,* Lin $\circ$ 2ISS*, and* Lin $\circ$ 2SS*.*

We derive our results on non-malleable codes using the above explicit constructions of non-malleable extractors based on a beautiful connection discovered by Cheraghchi and Gurswami [27] (see Theorem 25 for more details).

1.4 Extractors for interleaved sources

Our techniques also yield improved explicit constructions of extractors for interleaved sources, which generalize extractors for independent sources in the following way: the inputs to the extractor are samples from a few independent sources mixed (interleaved) in an unknown (but fixed) way. Raz and Yehudayoff [57] showed that such extractors have applications in communication complexity and proving lower bounds for arithmetic circuits. In a subsequent work, Chattopadhyay and Zuckerman [24] showed that such extractors can also be used to construct extractors for certain samplable sources, extending a line of work initiated by Trevisan and Vadhan [60]. We now define interleaved sources formally.

Definition 11 (Interleaved Sources) *Let $\mathbf{X}_1, \ldots, \mathbf{X}_r$ be arbitrary independent sources on $\{0,1\}^n$ and let $\pi : [rn] \to [rn]$ be any permutation. Then $\mathbf{Z} = (\mathbf{X}_1, \ldots, \mathbf{X}_r)_\pi$ is an r-interleaved source.*

Relevant prior work on interleaved extractors. Raz and Yehudayoff [57] gave explicit extractors for 2-interleaved sources when both the sources have min-entropy at least $(1-\delta)n$ for a tiny constant $\delta > 0$. Their construction is based on techniques from additive combinatorics and can output $\Omega(n)$ bits with exponentially small error. Subsequently, Chattopadhyay and Zuckerman [24] constructed extractors for 2-interleaved sources where one source has entropy $(1-\gamma)n$ for a small constant $\gamma > 0$ and the other source has entropy $\Omega(\log n)$. They achieve output length $O(\log n)$ bits with error $n^{-\Omega(1)}$.

A much better result (in terms of the min-entropy) is known if the extractor has access to an interleaving of more sources. For a large enough constant C, Chattopadhyay and Li [20] gave an explicit extractor for C-interleaved sources where each source has entropy $k \geq \text{poly}(\log n)$. They achieve output length $k^{\Omega(1)}$ and error $n^{-\Omega(1)}$.

Our results. Our main result is an explicit extractor for 2-interleaved sources where each source has min-entropy at least $2n/3$. The extractor outputs $\Omega(n)$ bits with error $2^{-n^{\Omega(1)}}$.

Theorem 12 *For any constant $\delta > 0$ and all integers $n > 0$, there exists an efficiently computable function $\text{i}\ell\text{Ext} : \{0,1\}^{2n} \to \{0,1\}^m$, $m = \Omega(n)$, such that for any two independent sources $\mathbf{X}$ and $\mathbf{Y}$, each on n bits with min-entropy at least $(2/3+\delta)n$, and any permutation $\pi : [2n] \to [2n]$,*

$$|\text{i}\ell\text{Ext}((\mathbf{X},\mathbf{Y})_\pi) - \mathbf{U}_m| \leq 2^{-n^{\Omega(1)}}.$$

2 Overview of constructions and techniques

Our results on non-malleable codes are derived from explicit constructions of invertible seedless non-malleable extractors (see Theorem 25). In this section,

we illustrate our main ideas used to give explicit constructions of seedless non-malleable extractors with respect to the relevant classes of tampering functions, and explicit extractors for interleaved sources.

We first focus on the main ideas involved in constructing non-malleable extractors against 2-split-state adversaries when the partition are of equal length (we denote this by 2ISS). This serves to illustrate the important ideas that go into all our explicit non-malleable extractor constructions. We refer the reader to the full version of our paper for complete details of our non-malleable extractor and code constructions.

2.1 Seedless non-malleable extractors with respect to interleaved 2-split-state tampering

We discuss the construction of a non-malleable extractor with respect to 2ISS. In such settings, it was shown in [27] that it is enough to construct non-malleable extractors assuming that at least one of f and g does not have any fixed points, assuming that the sources $\mathbf{X}$ and $\mathbf{Y}$ have entropy at least $n - n^{\delta}$. Thus, we construct a seedless non-malleable extractor $\text{nmExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$, $m = n^{\Omega(1)}$ such that the following hold: let $\mathbf{X}$ and $\mathbf{Y}$ be independent $(n, n - n^{\delta})$-sources, for some small $\delta > 0$. Let $f : \{0,1\}^n \to \{0,1\}^n$, $g : \{0,1\}^n \to \{0,1\}^n$ be arbitrary functions such that at least one of them has not fixed points, and $\pi : [2n] \to [2n]$ be an arbitrary permutation. Then,

$$\text{nmExt}((\mathbf{X},\mathbf{Y})_\pi), \text{nmExt}((f(\mathbf{X}), g(\mathbf{Y}))_\pi)) \approx_\epsilon \mathbf{U}_m, \text{nmExt}((f(\mathbf{X}), g(\mathbf{Y}))_\pi) \quad (1)$$

where $\epsilon = 2^{-n^{\Omega(1)}}$.

Our construction is based on the framework of advice generators and correlation breakers set up in the work [18], and used in various follow-up works on non-malleable extractors and codes. Before explaining this framework, we introduce some notation for ease of presentation. Let $\mathbf{Z} = (\mathbf{X},\mathbf{Y})_\pi$. We use the notation that if $\mathbf{W} = h((\mathbf{X},\mathbf{Y})_\pi)$ (for some function h), then $\mathbf{W}'$ or $(\mathbf{W})'$ stands for the corresponding random variable $h((f(\mathbf{X}), g(\mathbf{Y}))_\pi)$. Thus, $\mathbf{Z}' = (f(\mathbf{X}), g(\mathbf{Y}))_\pi$.

On a very high level, the task of constructing a non-malleable extractor can be broken down into the following two steps:

1. Generating advice: the task here is to construct a function $\text{advGen} : \{0,1\}^{2n} \to \{0,1\}^a$, $a \leq n^{\delta}$, such that $\text{advGen}(\mathbf{Z}) \neq \text{advGen}(\mathbf{Z}')$ with high probability.
2. Breaking correlation: here we construct an object that can be seen as a relaxation of a non-malleable extractor, in the sense that we supply the non-malleable extractor with a short advice string. This object is called an advice correlation breaker. We require that for all distinct strings $s, s' \in \{0,1\}^a$,

$$\text{ACB}(\mathbf{Z}, s), \text{ACB}(\mathbf{Z}', s') \approx \mathbf{U}_m, \text{ACB}(\mathbf{Z}', s').$$

Given the above components, the non-malleable extractor is defined as:

$$\text{nmExt}(\mathbf{Z}) = \text{ACB}(\mathbf{Z}, \text{advGen}(\mathbf{Z})).$$

The fact that the above satisfies (1) is not direct, but relies on further properties of the function advGen. In particular, we require that with high probability over the fixings of the random variables advGen($\mathbf{Z}$) and advGen($\mathbf{Z}'$), $\mathbf{X}$ and $\mathbf{Y}$ remain independent high min-entropy sources.

An explicit advice generator A natural first idea to construct an advice generator can be as follows: Take a slice (prefix) of $\mathbf{Z}$, say $\mathbf{Z}_1$, and use this to sample some coordinates from an encoding (using a good error correcting code) of $\mathbf{Z}$. A similar high level strategy has for example been used in [18], and other follow-up works. The intuition behind such a strategy is that since we assume $\mathbf{Z} \neq \mathbf{Z}'$, encoding it will ensure that they differ on a lot of coordinates. Thus, sampling a random set of coordinates will include one such coordinate with high probability. However, in the present case, it is not clear why this should work since it could be that $\mathbf{Z}_1$ contains all bits from say $\mathbf{X}$, and the set of coordinates where the encoding of $\mathbf{Z}$ and $\mathbf{Z}'$ differ may be a function of $\mathbf{X}$, which leads to unwanted correlations.

The next natural idea could be the following: First use the slice $\mathbf{Z}_1$ to sample a few coordinates from $\mathbf{Z}$. Let $\mathbf{Z}_2$ indicate $\mathbf{Z}$ projected onto the sampled coordinates. Now, it is not hard to prove that $\mathbf{Z}_2$ contains roughly equal number of bits from both the sources $\mathbf{X}$ and $\mathbf{Y}$. A strategy could be to now use $\mathbf{Z}_2$ to sample coordinates from an encoding of $\mathbf{Z}$. However, in this case, we run into similar problems as before: there may be unwanted correlations between the randomness used for sampling, and the random variable corresponding to the set of coordinates where the encoding of $\mathbf{Z}$ and $\mathbf{Z}'$ differ.

It turns out that the following subtler construction works:

Let $n_0 = n^{\delta'}$ for some small constant $\delta' > 0$. We take two slices from $\mathbf{Z}$, say $\mathbf{Z}_1$ and $\mathbf{Z}_2$ of lengths $n_1 = n_0^{c_0}$ and $n_2 = 10n_0$, for some constant $c_0 > 1$. Next, we use a good linear error correcting code (let the encoder of this code be E) to encode $\mathbf{Z}$ and sample n^γ coordinates (let $\mathbf{S}$ denote this set) from this encoding using $\mathbf{Z}_1$ (the sampler is based on seeded extractors [61]). Let $\mathbf{W}_1 = E(\mathbf{Z})_{\mathbf{S}}$. Next, using $\mathbf{Z}_2$, we sample a random set of indices $\mathbf{T} \subset [2n]$, and let $\mathbf{Z}_3 = \mathbf{Z}_{\mathbf{T}}$. We now use an extractor for interleaved sources, i.e., an extractor that takes as input an unknown interleaving of two independent sources and outputs uniform bits (see Section 1.4). Let iℓExt be this extractor (say from Theorem 12), and we apply it to $\mathbf{Z}_3$ to get $\mathbf{R} = \text{i}\ell\text{Ext}(\mathbf{Z}_3)$. Finally, let $\mathbf{W}_2$ be the output of a linear seeded extractor[3] LExt on $\mathbf{Z}$ with $\mathbf{R}$ as the seed. The output of the advice generator is $\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3, \mathbf{W}_1, \mathbf{W}_2$.

Notation: Define $\overline{x} = (x, 0^n)_\pi$ and $\overline{y} = (0^n, y)_\pi$. Similarly, define $\overline{f(x)} = \overline{(f(x), 0^n)_\pi}$ and $\overline{g(y)} = (0^n, g(y))_\pi$. Thus, $(x, y)_\pi = \overline{x} + \overline{y}$ and $(f(x), g(y))_\pi = \overline{f(x)} + \overline{g(y)}$. Let $\mathbf{X}_i$ be the bits of $\mathbf{X}$ in $\mathbf{Z}_i$ for $i = 1, 2, 3$ and $\mathbf{X}_4$ be the remaining bits of $\mathbf{X}$. Similarly define $\mathbf{Y}_i$'s, $i = 1, 2, 3, 4$.

[3] A linear seeded extractor is a seeded extractor where for any fixing of the seed, the output is a linear function of the source.

We now proceed to argue the correctness of the above construction. Note that the correctness of advGen is direct if $\mathbf{Z}_i \neq \mathbf{Z}'_i$ for some $i \in \{1,2,3\}$. Thus, assume $\mathbf{Z}_i = \mathbf{Z}'_i$ for $i = 1,2,3$. It follows that $\mathbf{S} = \mathbf{S}', \mathbf{T} = \mathbf{T}'$ and $\mathbf{R} = \mathbf{R}'$. Recall that $(\mathbf{X},\mathbf{Y})_\pi = \overline{\mathbf{X}} + \overline{\mathbf{Y}}$ and $(f(\mathbf{X}), g(\mathbf{Y})_\pi) = \overline{f(\mathbf{X})} + \overline{g(\mathbf{Y})}$. Since E is a linear code and LExt is a linear seeded extractor, the following hold:

$$\mathbf{W}_1 - \mathbf{W}'_1 = (E(\overline{\mathbf{X}} + \overline{\mathbf{Y}} - \overline{f(\mathbf{X})} - \overline{g(\mathbf{Y})}))_{\mathbf{S}},$$
$$\mathbf{W}_2 - \mathbf{W}'_2 = \mathrm{LExt}(\overline{\mathbf{X}} + \overline{\mathbf{Y}} - \overline{f(\mathbf{X})} - \overline{g(\mathbf{Y})}, \mathbf{R}).$$

Suppose that $\mathbf{Z}_1$ contains more bits from $\mathbf{X}$ than $\mathbf{Y}$, i.e., $|\mathbf{X}_1| \geq |\mathbf{Y}_1|$ (where $|\alpha|$ denotes the length of the string α).

Now the idea is the following: Either (i) we can fix $\overline{\mathbf{X}} - \overline{f(\mathbf{X})}$ and claim that $\mathbf{X}_1$ still has enough min-entropy, or (ii) we can claim that $\overline{\mathbf{X}} - \overline{f(\mathbf{X})}$ has enough min-entropy conditioned on the fixing of $(\mathbf{X}_2, \mathbf{X}_3)$. Let us first discuss why this is enough. Suppose we are in the first case. Then, we can fix $\overline{\mathbf{X}} - \overline{f(\mathbf{X})}$ and $\mathbf{Y}$ and argue that $\mathbf{Z}_1$ is a deterministic function of $\mathbf{X}$ and contains enough entropy. Note that $\overline{\mathbf{X}} + \overline{\mathbf{Y}} - \overline{f(\mathbf{X})} - \overline{g(\mathbf{Y})}$ is now fixed, and in fact it is fixed to a non-zero string (using the assumption that at least one of f or g has no fixed points). Thus, $E(\overline{\mathbf{X}} + \overline{\mathbf{Y}} - \overline{f(\mathbf{X})} - \overline{g(\mathbf{Y})})$ is a string with a constant fraction of the coordinates set to 1 (since E is an encoder of a linear error correcting code with constant relative distance), and it follows that with high probability $(E(\overline{\mathbf{X}} + \overline{\mathbf{Y}} - \overline{f(\mathbf{X})} - \overline{g(\mathbf{Y})}))_{\mathbf{S}}$ contains a non-zero entry (using the fact that $\mathbf{S}$ is sampled using $\mathbf{Z}_1$, which has enough entropy). This finishes the proof in this case since it implies $\mathbf{W}_1 \neq \mathbf{W}'_1$ with high probability.

Now suppose we are in case (ii). We use the fact that $\mathbf{Z}_2$ contains entropy to conclude that the sampled bits $\mathbf{Z}_3$ contain almost equal number of bits from $\mathbf{X}$ and $\mathbf{Y}$ (with high probability over $\mathbf{Z}_2$). Now we can fix $\mathbf{Z}_2$ without loosing too much entropy from $\mathbf{Z}_3$ (by making the size of $\mathbf{Z}_3$ to be significantly larger than $\mathbf{Z}_2$). Next, we observe that $\mathbf{Z}_3$ is an interleaved source, and hence $\mathbf{R}$ is close to uniform. We now fix $\mathbf{X}_3$, and argue that $\mathbf{R}$ continues to be uniform. This follows roughly from the fact that any extractor for an interleaving of 2-sources is strong. Thus, $\mathbf{R}$ now becomes a deterministic function of $\mathbf{Y}$ while at the same time, $\overline{\mathbf{X}} - \overline{f(\mathbf{X})}$ still has enough min-entropy. Hence, $\mathrm{LExt}(\overline{\mathbf{X}} - \overline{f(\mathbf{X})}, \mathbf{R})$ is close to uniform even conditioned on $\mathbf{R}$. We can now fix $\mathbf{R}$ and $\mathrm{LExt}(\overline{\mathbf{Y}} - \overline{g(\mathbf{Y})}, \mathbf{R})$ without affecting the distribution $\mathrm{LExt}(\overline{\mathbf{X}} - \overline{f(\mathbf{X})}, \mathbf{R})$, since $\mathrm{LExt}(\overline{\mathbf{Y}} - \overline{g(\mathbf{Y})}, \mathbf{R})$ is a deterministic function of $\mathbf{Y}$ while $\mathrm{LExt}(\overline{\mathbf{X}} - \overline{f(\mathbf{X})}, \mathbf{R})$ is a deterministic function of $\mathbf{X}$ conditioned on the previous fixing of $\mathbf{R}$. It follows that after these fixings, $\mathbf{W}_2 - \mathbf{W}'_2$ is close to a uniform string and hence $\mathbf{W}_2 - \mathbf{W}'_2 \neq 0$ with probability $1 - 2^{-n^{\Omega(1)}}$, which completes the proof.

The fact that it is enough to consider case (i) and case (ii) relies on a careful convex combination analysis based on the pre-image size of the function $f(x) - x$. In addition, for the above argument to work we need to carefully adjust the sizes of $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Z}_3$. We skip the details here, and refer the interested reader to later parts of the paper for more details.

An explicit advice correlation breaker We now discuss the other crucial component in the construction, the advice correlation breaker $\mathrm{ACB} : \{0,1\}^{2n} \times \{0,1\}^a \to \{0,1\}^m$. Informally, the advice correlation breaker we construct takes 2 inputs, the interleaved source $\mathbf{Z}$ (that contains some min-entropy) and an advice string $s \in \{0,1\}^a$, and outputs a distribution on $\{0,1\}^m$ with the following guarantee. If $s' \in \{0,1\}^a$ is another advice such that $s \neq s'$, then

$$\mathrm{ACB}(\mathbf{Z}, s), \mathrm{ACB}(\mathbf{Z}', s') \approx \mathbf{U}_m, \mathrm{ACB}(\mathbf{Z}', s') \tag{2}$$

Our construction crucially relies on an explict advice correlation breaker constructed in [21] that satisfies the following property: Let $\mathbf{A}$ be an (n,k)-source, and $\mathbf{A}' = f(\mathbf{A})$ be a tampered version of $\mathbf{A}$. Further let $\mathbf{B}$ be a uniform random variable, and $\mathbf{B}' = g(\mathbf{B})$. Finally, let $\mathbf{C}, \mathbf{C}'$ be arbitrary random variables such that $\{\mathbf{A}, \mathbf{A}'\}$ is independent of $\{\mathbf{B}, \mathbf{B}', \mathbf{C}, \mathbf{C}'\}$. Then [21] constructed an advice correlation breaker ACB_1 such that for advice strings $s \neq s'$,

$$\mathrm{ACB}_1(\mathbf{B}, \mathbf{A} + \mathbf{C}, s), \mathrm{ACB}_1(\mathbf{B}', \mathbf{A}' + \mathbf{C}', s') \approx U_m, \mathrm{ACB}_1(\mathbf{B}', \mathbf{A}' + \mathbf{C}', s'). \tag{3}$$

The construction of ACB_1 is based on the powerful technique of alternating extraction introduced by Dziembowski and Pietrzak [35], and later used in almost all recent works on non-malleable extractors. In particular, the construction in [21] relies on linear seeded extractors and an elegant primitive known as the *flip-flop* alternating extraction, which was introduced by Cohen [29].

Recall that since $\mathbf{Z} = \overline{\mathbf{X}} + \overline{\mathbf{Y}}$ and $\mathbf{Z}' = \overline{f(\mathbf{X})} + \overline{g(\mathbf{Y})}$, (2) can be stated as

$$\mathrm{ACB}(\overline{\mathbf{X}} + \overline{\mathbf{Y}}, s), \mathrm{ACB}(\overline{f(\mathbf{X})} + \overline{g(\mathbf{Y})}, s') \approx_\epsilon \mathbf{U}_m, \mathrm{ACB}(\overline{f(\mathbf{X})} + \overline{g(\mathbf{Y})}, s')$$

Our main idea of reducing (2) to (3) is as follows: we again take a short slice from $\mathbf{Z}$, say $\mathbf{Z}_4$ (larger than the size of $\{\mathbf{Z}_1, \mathbf{Z}_2, \mathbf{Z}_3\}$), and use a linear seeded extractor LExt to convert $\mathbf{Z}_4$ into a somewhere random source (i.e, a matrix, where some rows are uniform). This can be done by defining row i of the matrix to be $\mathbf{W}_i = \mathrm{LExt}(\mathbf{Z}_4, i)$. The idea now is to simply apply ACB_1 on each row $\mathbf{W}_i$, using the source $\mathbf{Z}$, and the concatenation of s and the index of the row as the new advice string, i.e., compute $\mathrm{ACB}_1(\mathbf{W}_i, \mathbf{Z}, s, i)$. By appealing to a slightly more general version of (3), where we allow multiple tampering, it follows that the output of ACB_1 corresponding to some uniform row is now independent of the output of ACB_1 on all other rows (including tampered rows). Thus, we can simply output $\oplus_i(\mathrm{ACB}_1(\mathbf{W}_i, \mathbf{Z}, s, i))$.

This almost works, modulo a technical caveat–the somewhere random source constructed out of $\mathbf{Z}_4$ is a tall matrix, with more rows than columns, but the parameters of ACB_1 require us to work with a fat matrix, with more columns that rows. This is roughly because, we want the uniform row to have more entropy than the total size of all tampered random variables. To fix this, we use another linear seeded extractor on the source $\mathbf{Z}$ with each row $\mathbf{W}_i$ as the seed to obtain another somewhere random source of the right shape.

2.2 From non-malleable extractors to non-malleable codes

To obtain our non-malleable codes, the decoding function corresponds to computing the extractor, which is already efficient. On the other hand, the encoding function corresponds to sampling from the pre-image of any given output of the non-malleable extractor. Thus we need to find an efficient way to do this, which is quite non-trivial. We suitably modify our extractor to support efficient sampling. Here we briefly sketch some high level ideas involved and refer the reader to the full version of our paper for more details.

Recall $\mathbf{Z} = (\mathbf{X}, \mathbf{Y})_\pi$. The first modification is that in all applications of seeded extractors in our construction, we specifically use linear seeded extractors. This allows us to argue that the pre-image we are trying to sample from is in fact a convex combination of distributions supported on subspaces. The next crucial observation is the fact that we can use smaller disjoint slices of $\mathbf{Z}$ to carry out various steps outlined in the construction. This is to ensure that the dimensions of the subspaces that we need to sample from, do not depend on the values of the random variables that we fix. For the steps where we use the entire source $\mathbf{Z}$ (in the construction of the advice correlation breaker), we replace $\mathbf{Z}$ by a large enough slice of $\mathbf{Z}$. However this is problematic if we choose the slice deterministically, since in an arbitrary interleaving of two sources, a slice of length less than n might have bits only from one source. We get around this by pseudorandomly sampling enough coordinates from $\mathbf{Z}$ (by first taking a small slice of $\mathbf{Z}$ and using a sampler that works for weak sources [61]).

We now use an elegant trick introduced by Li [46] where the output of the non-malleable extractor described above (with the modifications that we have specified) is now used as a seed in a linear seeded extractor applied to an even larger pseudorandom slice of $\mathbf{Z}$. The linear seeded extractor that we use has the property that for any fixing of the seed, the rank of the linear map corresponding to the extractor is the same, and furthermore one can efficiently sample from the pre-image of any output of the extractor. The final modification needed is a careful choice of the error correcting code used in the advice generator. For this we use a dual BCH code, which allows us to argue that we can discard some output bits of the advice generator without affecting its correctness (based on the dual distance of the code). This is crucial in order to argue that the rank of the linear restriction imposed on the free variables of $\mathbf{Z}$ does not depend on the values of the bits fixed so far. We refer the reader to the full version of our paper where we provide more intuition and complete details of the modified non-malleable extractor and sampling procedure.

2.3 Extractors for interleaved sources

Here we give a sketch of our improved extractor for interleaved sources $\mathbf{Z} = (\mathbf{X}, \mathbf{Y})_\pi$. We refer the reader to the full version of our paper for more details. We present our construction and also explain the proof along the way, as this gives more intuition to the different steps of the construction. The high level idea is the following: transform $\mathbf{Z}$ into a matrix of random variables (called

a somewhere random source) such that at least one of the random variables is uniform, and the matrix is of the right shape, i.e, a fat matrix with more columns than rows. Once we have such a matrix, the idea is to use the advice correlation breaker from [21] mentioned above to break the correlation among the rows of the matrix. The final output will just be a bit-wise XOR of the output of the advice correlation breaker on each row of the matrix. We now give some more details on how to make this approach work.

Let $\mathbf{Z} = (\mathbf{X}, \mathbf{Y})_\pi$. We start by taking a large enough slice $\mathbf{Z}_1$ from $\mathbf{Z}$ (say, of length $(2/3 + \delta/2)n$). Let $\mathbf{X}$ have more bits in this slice than $\mathbf{Y}$. Let $\mathbf{X}_1$ be the bits of $\mathbf{X}$ in $\mathbf{Z}_1$ and $\mathbf{X}_2$ be the remaining bits of $\mathbf{X}$. Similarly define $\mathbf{Y}_1$ and $\mathbf{Y}_2$. Notice that $\mathbf{X}_1$ has linear entropy and also that $\mathbf{X}_2$ has linear entropy conditioned on $\mathbf{X}_1$. We fix $\mathbf{Y}_1$ and use a condenser (from work of Raz [55]) to condense $\mathbf{Z}_1$ into a matrix with a constant number of rows such that at least one row is close to a distribution with entropy rate at least 0.9. Notice that this matrix is a deterministic function of $\mathbf{X}$. The next step is to use $\mathbf{Z}$ and each row of the matrix as a seed to a linear seeded extractor to get longer rows. This requires some care for the choice of the linear seeded extractor since the seed has some deficiency in entropy. After this step, we use the advice correlation breaker from [21] on $\mathbf{Z}$ and each row of the somewhere random source, with the row index as the advice (similar to what is done in the construction of non-malleable extractors sketched above). Finally we compute the bit-wise XOR of the different outputs that we obtain. Let $\mathbf{V}$ denote this random variable. To output $\Omega(n)$ bits, we use a linear seeded extractor on $\mathbf{Z}$ with $\mathbf{V}$ as the seed. The correctness of various steps in the proof exploits the fact that $\mathbf{Z}$ can be written as the bit-wise sum of two independent sources, and the fact that we use linear seeded extractors.

2.4 Organization

We use Section 3 to introduce some background and notation. We present our seedless non-malleable extractors with respect to interleaved split-state tampering in Section 4. We conclude with some open problems in Section 5.

3 Background and notation

We use $\mathbf{U}_m$ to denote the uniform distribution on $\{0,1\}^m$.
For any integer $t > 0$, $[t]$ denotes the set $\{1, \ldots, t\}$.
For a string y of length n, and any subset $S \subseteq [n]$, we use y_S to denote the projection of y to the coordinates indexed by S.
We use bold capital letters for random variables and samples as the corresponding small letter, e.g., $\mathbf{X}$ is a random variable, with x being a sample of $\mathbf{X}$.
For strings $x, y \in \{0,1\}^n$, we use $x + y$ (or equivalently $x - y$) to denote the bit-wise xor of the two strings.

3.1 Probability lemmas

The following result on min-entropy was proved by Maurer and Wolf [50].

Lemma 1. *Let* $\mathbf{X}, \mathbf{Y}$ *be random variables such that the random variable* $\mathbf{Y}$ *takes at most* ℓ *values. Then*

$$\mathbf{Pr}_{y\sim\mathbf{Y}}[H_\infty(\mathbf{X}|\mathbf{Y}=y) \geq H_\infty(\mathbf{X}) - \log\ell - \log(1/\epsilon)] > 1-\epsilon.$$

The following lemma is useful in bounding statistical distance of distributions after conditionings.

Lemma 2. *Let* D_1 *and* D_2 *be distributions on some universe* Ω *such that* $|X - Y| \leq \epsilon$. *Let* $\mathcal{E}$ *be some event some that* $\Pr[D_1 \in \mathcal{E}] \geq \delta$. *Then,* $|(D_1|\mathcal{E}) - (D_2|\mathcal{E})| \leq \epsilon/\delta$.

3.2 Conditional min-entropy

Definition 13 *The average conditional min-entropy of a source* $\mathbf{X}$ *given a random variable* $\mathbf{W}$ *is defined as*

$$\widetilde{H}_\infty(\mathbf{X}|\mathbf{W}) = -\log\left(\mathbf{E}_{w\sim W}\left[\max_x \Pr[\mathbf{X}=x|\mathbf{W}=w]\right]\right) = -\log\left(\mathbf{E}\left[2^{-H_\infty(\mathbf{X}|\mathbf{W}=w)}\right]\right).$$

We recall some results on conditional min-entropy from the work of Dodis et al. [32].

Lemma 3 ([32]). *For any* $\epsilon > 0$,

$$\mathbf{Pr}_{w\sim\mathbf{W}}\left[H_\infty(\mathbf{X}|\mathbf{W}=w) \geq \widetilde{H}_\infty(\mathbf{X}|\mathbf{W}) - \log(1/\epsilon)\right] \geq 1-\epsilon.$$

Lemma 4 ([32]). *If a random variable* $\mathbf{Y}$ *has support of size* 2^ℓ, *then* $\widetilde{H}_\infty(\mathbf{X}|\mathbf{Y}) \geq H_\infty(\mathbf{X}) - \ell$.

3.3 Seeded Extractors

Definition 14 *A function* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k,ϵ)*-seeded extractor if for any source* $\mathbf{X}$ *of min-entropy* k, $|\mathrm{Ext}(\mathbf{X}, \mathbf{U}_d) - \mathbf{U}_m| \leq \epsilon$. Ext *is called a strong seeded extractor if* $|(\mathrm{Ext}(\mathbf{X}, \mathbf{U}_d), \mathbf{U}_d) - (\mathbf{U}_m, \mathbf{U}_d)| \leq \epsilon$, *where* $\mathbf{U}_m$ *and* $\mathbf{U}_d$ *are independent.*

Further, if for each $s \in \mathbf{U}_d$, $\mathrm{Ext}(\cdot, s) : \{0,1\}^n \to \{0,1\}^m$ *is a linear function, then* Ext *is called a linear seeded extractor.*

We require extractors that can extract uniform bits when the source only has sufficient conditional min-entropy.

Definition 15 *A* (k,ϵ)*-seeded average case seeded extractor* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *for min-entropy* k *and error* ϵ *satisfies the following property: For any source* $\mathbf{X}$ *and any arbitrary random variable* $\mathbf{Z}$ *with* $\tilde{H}_\infty(\mathbf{X}|\mathbf{Z}) \geq k$,

$$\mathrm{Ext}(\mathbf{X}, \mathbf{U}_d), \mathbf{Z} \approx_\epsilon \mathbf{U}_m, \mathbf{Z}.$$

It was shown in [32] that any seeded extractor is also an average case extractor.

Lemma 5 ([32]). *For any $\delta > 0$, if* Ext *is a (k, ϵ)-seeded extractor, then it is also a $(k + \log(1/\delta), \epsilon + \delta)$-seeded average case extractor.*

We record a folklore lemma, and include a proof for completeness.

Lemma 6. *Let* $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a (k, ϵ) strong seeded. Then, for any source (n, k)-source $\mathbf{X}$ and any independent $(d, d - \lambda)$-source $\mathbf{Y}$,*

$$|\text{Ext}(\mathbf{X}, \mathbf{Y}), \mathbf{Y} - \mathbf{U}_m, \mathbf{Y}| \leq 2^{\lambda}\epsilon.$$

Proof. Suppose $\mathbf{Y}$ is uniform over a set $A \subset \{0,1\}^d$ of size $2^{d-\lambda}$. We have,

$$\begin{aligned}
|\text{Ext}(\mathbf{X}, \mathbf{Y}), \mathbf{Y} - \mathbf{U}_m, \mathbf{Y}| &= \frac{1}{2^{d-\lambda}} \cdot \sum_{y \in A} |\text{Ext}(\mathbf{X}, y) - \mathbf{U}_m| \\
&\leq \frac{1}{2^{d-\lambda}} \cdot \sum_{y \in \{0,1\}^d} |\text{Ext}(\mathbf{X}, y) - \mathbf{U}_m| \\
&= \frac{1}{2^{d-\lambda}} \cdot 2^d \cdot |\text{Ext}(\mathbf{X}, \mathbf{U}_d), \mathbf{U}_d - \mathbf{U}_m, \mathbf{U}_d| \\
&= 2^{\lambda} \cdot \epsilon,
\end{aligned}$$

where the last inequality follows from the fact that Ext is a (k, ϵ) strong seeded extractor.

3.4 Samplers and extractors

Zuckerman [61] showed that seeded extractors can be used as samplers given access to weak sources. This connection is best presented by a graph theoretic representation of seeded extractors. A seeded extractor $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ can be viewed as an unbalanced bipartite graph G_{Ext} with 2^n left vertices (each of degree 2^d) and 2^m right vertices. Let $\mathcal{N}(x)$ denote the set of neighbors of x in G_{Ext}.

Theorem 16 ([61]) *Let* $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a seeded extractor for min-entropy k and error ϵ. Let $D = 2^d$. Then for any set $R \subseteq \{0,1\}^m$,*

$$|\{x \in \{0,1\}^n : ||\mathcal{N}(x) \cap R| - \mu_R D| > \epsilon D\}| < 2^k,$$

where $\mu_R = |R|/2^m$.

Theorem 17 ([61]) *Let* $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a seeded extractor for min-entropy k and error ϵ. Let $\{0,1\}^d = \{r_1, \ldots, r_D\}$, $D = 2^d$. Define* $\text{Samp}(x) = \{\text{Ext}(x, r_1), \ldots, \text{Ext}(x, r_D)\}$. *Let $\mathbf{X}$ be an $(n, 2k)$-source. Then for any set $R \subseteq \{0,1\}^m$,*

$$\mathbf{Pr}_{\mathbf{x} \sim \mathbf{X}}[||\text{Samp}(\mathbf{x}) \cap R| - \mu_R D| > \epsilon D] < 2^{-k},$$

where $\mu_R = |R|/2^m$.

3.5 Explicit extractors from prior work

We recall an optimal construction of strong-seeded extractors.

Theorem 18 ([42]) *For any constant $\alpha > 0$, and all integers $n, k > 0$ there exists a polynomial time computable strong-seeded extractor* $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *with $d = O(\log n + \log(1/\epsilon))$ and $m = (1-\alpha)k$.*

The following are explicit constructions of linear seeded extractors.

Theorem 19 ([56, 59]) *For every $n, k, m \in \mathbb{N}$ and $\epsilon > 0$, with $m \leq k \leq n$, there exists an explicit strong linear seeded extractor* $\text{LExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *for min-entropy k and error ϵ, where $d = O\left(\log^2(n/\epsilon)/\log(k/m)\right)$.*

A drawback of the above construction is that the seeded length is $\omega(\log n)$ for sub-linear min-entropy. A construction of Li [45] achieves $O(\log n)$ seed length for even polylogarithmic min-entropy.

Theorem 20 ([45]) *There exists a constant $c > 1$ such that for every $n, k \in \mathbb{N}$ with $c \log^8 n \leq k \leq n$ and any $\epsilon \geq 1/n^2$, there exists a polynomial time computable linear seeded extractor* $\text{LExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *for min-entropy k and error ϵ, where $d = O(\log n)$ and $m \leq \sqrt{k}$.*

A different construction achieves seed length $O(\log(n/\epsilon))$ for high entropy sources.

Theorem 21 ([18, 46]) *For all $\delta > 0$ there exist $\alpha, \gamma > 0$ such that for all integers $n > 0$, $\epsilon \geq 2^{-\gamma n}$, there exists an efficiently computable linear strong seeded extractor* $\text{LExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{\alpha d}$, *$d = O(\log(n/\epsilon))$ for min-entropy δn. Further, for any $y \in \{0,1\}^d$, the linear map* $\text{LExt}(\cdot, y)$ *has rank αd.*

The above theorem is stated in [46] for $\delta = 0.9$, but it is straightforward to see that the proof extends for any constant $\delta > 0$.

We use a property of linear seeded extractors proved by Rao [53].

Lemma 7 ([53]). *Let* $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a linear seeded extractor for min-entropy k with error $\epsilon < \frac{1}{2}$. Let X be an affine (n,k)-source. Then*

$$\Pr_{u \sim U_d}[|\text{Ext}(X,u) - U_m| > 0] \leq 2\epsilon.$$

We recall a two-source extractor construction for high entropy sources based on the inner product function.

Theorem 22 ([28]) *For all $m, r > 0$, with $q = 2^m, n = rm$, let $\mathbf{X}, \mathbf{Y}$ be independent sources on $\mathbb{F}_q^r$ with min-entropy k_1, k_2 respectively. Let* IP *be the inner product function over the field $\mathbb{F}_q$. Then, we have:*

$$|\text{IP}(\mathbf{X},\mathbf{Y}), \mathbf{X} - \mathbf{U}_m, \mathbf{X}| \leq \epsilon, \qquad |\text{IP}(\mathbf{X},\mathbf{Y}), \mathbf{Y} - \mathbf{U}_m, \mathbf{Y}| \leq \epsilon$$

where $\epsilon = 2^{-(k_1+k_2-n-m)/2}$.

Rao [52] (based on an argument by Boaz Barak) proved that every two-source extractor is strong. It is easy to observe that the proof generalizes to the case of interleaved two-source extractors. We record this below in a slightly more general setting of unequal length sources.

Theorem 23 ([52]) *Suppose* $\text{i}\ell\text{Ext} : \{0,1\}^{n_1+n_2} \to \{0,1\}^m$ *be an interleaved source extractor that satisfies the following: if* $\mathbf{X}$ *is a* (n_1, k_1)*-source,* $\mathbf{Y}$ *is an independent* (n_2, k_2)*-source, and* $\pi : [n_1 + n_2] \to [n_1 + n_2]$ *is an arbitrary permutation, then*

$$|\text{i}\ell\text{Ext}((\mathbf{X}, \mathbf{Y})_\pi) - \mathbf{U}_m| \leq \epsilon.$$

Then, in fact $\text{i}\ell\text{Ext}$ *satisfies the following stronger properties:*

- *Let* $\mathbf{X}$ *be a* (n_1, k)*-source,* $\mathbf{Y}$ *be an independent* (n_2, k_2)*-source, and* $\pi : [n_1 + n_2] \to [n_1 + n_2]$ *be an arbitrary permutation. Then,*

$$|\text{i}\ell\text{Ext}((\mathbf{X}, \mathbf{Y})_\pi), \mathbf{X} - \mathbf{U}_m, \mathbf{X}| \leq 2^m \cdot (2^{k-k_1} + \epsilon).$$

- *Let* $\mathbf{X}$ *be a* (n_1, k_1)*-source,* $\mathbf{Y}$ *be an independent* (n_2, k)*-source, and* $\pi : [n_1 + n_2] \to [n_1 + n_2]$ *be an arbitrary permutation. Then,*

$$|2\text{i}\ell\text{Ext}(\mathbf{X}, \mathbf{Y}), \mathbf{Y} - \mathbf{U}_m, \mathbf{Y}| \leq 2^m \cdot (2^{k-k_2} + \epsilon).$$

3.6 Advice correlation breakers

We use a primitive called 'correlation breaker' in our construction. Consider a situation where we have arbitrarily correlated random variables $\mathbf{Y}^1, \ldots, \mathbf{Y}^r$, where each $\mathbf{Y}^i$ is on ℓ bits. Further suppose $\mathbf{Y}^1$ is a 'good' random variable (typically, we assume $\mathbf{Y}^1$ is uniform or has almost full min-entropy). A correlation breaker CB is an explicit function that takes some additional resource $\mathbf{X}$, where $\mathbf{X}$ is typically additional randomness (an (n, k)-source) that is independent of $\{\mathbf{Y}^1, \ldots, \mathbf{Y}^r\}$. Thus using $\mathbf{X}$, the task is to break the correlation between $\mathbf{Y}^1$ and the random variables $\mathbf{Y}^2, \ldots, \mathbf{Y}^r$, i.e., $\text{CB}(\mathbf{Y}^1, \mathbf{X})$ is independent of $\{\text{CB}(\mathbf{Y}^2, \mathbf{X}), \ldots, \text{CB}(\mathbf{Y}^r, \mathbf{X})\}$. A weaker notion is that of an advice correlation breaker that takes in some advice for each of the $\mathbf{Y}^i$'s as an additional resource in breaking the correlations. This primitive was implicitly constructed in [18] and used in explicit constructions of non-malleable extractors, and has subsequently found many applications in explicit constructions of extractors for independent sources and non-malleable extractors.

We recall an explicit advice correlation breaker constructed in [20]. This correlation breaker works even with the weaker guarantee that the 'helper source' $\mathbf{X}$ is now allowed to be correlated to the sources random variables $\mathbf{Y}^1, \ldots, \mathbf{Y}^r$ in a structured way. Concretely, we assume the source to be of the form $\mathbf{X}+\mathbf{Z}$, where $\mathbf{X}$ is assumed to be an (n, k)-source that is uncorrelated with $\mathbf{Y}^1, \ldots, \mathbf{Y}^r, \mathbf{Z}$. We now state the result more precisely.

Theorem 24 ([20]) *For all integers* $n, n_1, n_2, k, k_1, k_2, t, d, h, \lambda$ *and any* $\epsilon > 0$*, such that* $d = O(\log^2(n/\epsilon))$*,* $k_1 \geq 2d + 8tdh + \log(1/\epsilon)$*,* $n_1 \geq 2d + 10tdh + (4ht + 1)n_2^2 + \log(1/\epsilon)$*, and* $n_2 \geq 2d + 3td + \log(1/\epsilon)$*, let*

- $\mathbf{X}$ *be an* (n, k_1)*-source,* $\mathbf{X}'$ *a r.v on* n *bits,* $\mathbf{Y}^1$ *be an* $(n_1, n_1 - \lambda)$*-source,* $\mathbf{Z}, \mathbf{Z}'$ *are r.v's on* n *bits, and* $\mathbf{Y}^2, \ldots, \mathbf{Y}^t$ *be r.v's on* n_1 *bits each, such that* $\{\mathbf{X}, \mathbf{X}'\}$ *is independent of* $\{\mathbf{Z}, \mathbf{Z}', \mathbf{Y}^1, \ldots, \mathbf{Y}^t\}$*,*
- $id^1, \ldots, id^t$ *be bit-strings of length* h *such that for each* $i \in \{2, t\}$*,* $id^1 \neq id^i$*.*

Then there exists an efficient algorithm $\text{ACB} : \{0,1\}^{n_1} \times \{0,1\}^n \times \{0,1\}^h \to \{0,1\}^{n_2}$ *which satisfies the following: let*

- $\mathbf{Y}_h^1 = \text{ACB}(\mathbf{Y}^1, \mathbf{X} + \mathbf{Z}, id^1)$*,*
- $\mathbf{Y}_h^i = \text{ACB}(\mathbf{Y}^i, \mathbf{X}' + \mathbf{Z}', id^i)$*,* $i \in [2, t]$

Then,

$$\mathbf{Y}_h^1, \mathbf{Y}_h^2, \ldots, \mathbf{Y}_h^t, \mathbf{X}, \mathbf{X}' \approx_{O((h+2^\lambda)\epsilon)} \mathbf{U}_{n_2}, \mathbf{Y}_h^2, \ldots, \mathbf{Y}_h^t, \mathbf{X}, \mathbf{X}'.$$

3.7 A connection between non-malleable codes and extractors

The following theorem proved by Cheraghchi and Guruswami [27] that connects non-malleable extractors and codes.

Theorem 25 ([27]) *Let* $\text{nmExt} : \{0,1\}^n \to \{0,1\}^m$ *be an efficient seedless* (n, m, ϵ)*-non-malleable extractor with respect to a class of tampering functions* $\mathcal{F}$ *acting on* $\{0,1\}^n$*. Further suppose* nmExt *is* ϵ'*-invertible. Then there exists an efficient construction of a non-malleable code with respect to the tampering family* $\mathcal{F}$ *with block length* $= n$*, relative rate* $\frac{m}{n}$ *and error* $2^m\epsilon + \epsilon'$*.*

4 NM extractors for interleaved split-state adversaries

The main result of this section is an explicit non-malleable extractor for interleaved 2-split-state tampering families with equal length partitions, which we denote by $2\text{ISS} \subset \mathcal{F}_{2n}$.

Theorem 26 *For all integers* $n > 0$ *there exists an explicit function* $\text{nmExt} : \{0,1\}^{2n} \to \{0,1\}^m$*,* $m = n^{\Omega(1)}$*, such that the following holds: for arbitrary tampering functions* $f, g \in \mathcal{F}_n$*, any permutation* $\pi : [2n] \to [2n]$ *and independent uniform sources* $\mathbf{X}$ *and* $\mathbf{Y}$ *each on* n *bits, there exists a distribution* $\mathcal{D}_{f,g,\pi}$ *on* $\{0,1\}^m \cup \{same^\star\}$*, such that*

$$|\text{nmExt}((\mathbf{X}, \mathbf{Y})_\pi), \text{nmExt}((f(\mathbf{X}), g(\mathbf{Y}))_\pi)) - \mathbf{U}_m, \text{copy}(\mathcal{D}_{f,g,\pi}, \mathbf{U}_m)| \leq 2^{-n^{\Omega(1)}}.$$

In such settings, it was shown in [27] that it is enough to construct non-malleable extractors assuming that at least one of f and g does not have any fixed points, assuming that the sources $\mathbf{X}$ and $\mathbf{Y}$ have entropy at least $n - n^\delta$. We thus prove the following theorem, from which Theorem 26 is direct.

Theorem 27 *There exists a $\delta > 0$ such that for all integers $n, k > 0$ with $n \geq k \geq n - n^{\delta}$, there exists an explicit function* $\text{nmExt} : \{0,1\}^{2n} \to \{0,1\}^m$, *$m = n^{\Omega(1)}$, such that the following holds: for arbitrary tampering functions $f, g \in \mathcal{F}_n$, any permutation $\pi : [2n] \to [2n]$ and independent (n,k)-sources $\mathbf{X}$ and $\mathbf{Y}$, the following holds:*

$$|\text{nmExt}((\mathbf{X},\mathbf{Y})_\pi), \text{nmExt}((f(\mathbf{X}), g(\mathbf{Y}))_\pi)) - \mathbf{U}_m, \text{nmExt}((f(\mathbf{X}), g(\mathbf{Y}))_\pi)| \leq 2^{-n^{\Omega(1)}}.$$

We will prove a slightly more general result which is a direct by-product of our proof technique for proving the above theorem, and lets us re-use this non-malleable extractor for the class of linear adversaries composed with split-state adversaries. We prove the following theorem.

Theorem 28 *There exists a $\delta > 0$ such that for all integers $n, k > 0$ with $n \geq k \geq n - n^{\delta}$, there exists an explicit function* $\text{nmExt} : \{0,1\}^{2n} \to \{0,1\}^m$, *$m = n^{\Omega(1)}$, such that the following holds: Let $\mathbf{X}$ and $\mathbf{Y}$ be independent $(n, n-n^{\delta})$-sources, $\pi : [2n] \to [2n]$ any arbitrary permutation and arbitrary tampering functions $f_1, f_2, g_1, g_2 \in \mathcal{F}_n$ that satisfy the following condition:*

- $\forall x \in \textit{support}(\mathbf{X})$ *and* $y \in \textit{support}(\mathbf{Y})$, $f_1(x) + g_1(y) \neq x$ *or*
- $\forall x \in \textit{support}(\mathbf{X})$ *and* $y \in \textit{support}(\mathbf{Y})$, $f_2(x) + g_2(y) \neq y$.

Then,

$$|\text{nmExt}((\mathbf{X},\mathbf{Y})_\pi), \text{nmExt}(((f_1(\mathbf{X}) + g_1(\mathbf{Y})), (f_2(\mathbf{X}) + g_2(\mathbf{Y})))_\pi) - \mathbf{U}_m, \text{nmExt}(((f_1(\mathbf{X}) + g_1(\mathbf{Y})), (f_2(\mathbf{X}) + g_2(\mathbf{Y})))_\pi)| \leq 2^{-n^{\Omega(1)}}.$$

Clearly, Theorem 27 follows directly from the above theorem by setting $g_1(y) = 0$ for all y and $f_2(x) = 0$ for all x. We use the rest of the section to prove Theorem 28.

Our high level ideas in constructing the non-malleable extractor is via the framework set up in [18] of using advice generators and correlation breakers. We give intuition behind our construction in Section 2. We use Section 4.1 to construct an advice generator and Section 4.2 to construct an advice correlation breaker. Finally, we present the non-malleable extractor construction in Section 4.3.

Notation:

- If $\mathbf{W} = h((\mathbf{X},\mathbf{Y})_\pi)$ (for some function h), then we use $\mathbf{W}'$ or $(\mathbf{W})'$ to denote the random variable $h(((f_1(\mathbf{X}) + g_1(\mathbf{Y})), (f_2(\mathbf{X}) + g_2(\mathbf{Y})))_\pi)$.
- Define $\overline{\mathbf{X}} = (\mathbf{X}, 0^n)_\pi$, $\overline{\mathbf{Y}} = (0^n, \mathbf{Y})_\pi$, $\overline{f_1(\mathbf{X})} = (f_1(\mathbf{X}), 0^n)_\pi$, $\overline{f_2(\mathbf{X})} = (0^n, f_2(\mathbf{X}))_\pi$, $\overline{g_1(\mathbf{Y})} = (g_1(\mathbf{Y}), 0^n)_\pi$ and $\overline{g_2(\mathbf{Y})} = (0^n, g_2(\mathbf{Y}))_\pi$.
- Finally, define $\mathbf{Z} = \overline{\mathbf{X}} + \overline{\mathbf{Y}}$ and $\mathbf{Z}' = \overline{f_1(\mathbf{X})} + \overline{g_1(\mathbf{Y})} + \overline{f_2(\mathbf{X})} + \overline{g_2(\mathbf{Y})}$.

4.1 An advice generator

Lemma 8. *There exists an efficiently computable function* $\text{advGen} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{n_4}$, $n_4 = n^\delta$, *such that with probability at least* $1 - 2^{-n^{\Omega(1)}}$ *over the fixing of the random variables* $\text{advGen}((\mathbf{X},\mathbf{Y})_\pi), \text{advGen}(((f_1(\mathbf{X}) + g_1(\mathbf{Y})),(f_2(\mathbf{X}) + g_2(\mathbf{Y})))_\pi)$, *the following hold:*

- $\{\text{advGen}((\mathbf{X},\mathbf{Y})_\pi) \neq \text{advGen}(((f_1(\mathbf{X}) + g_1(\mathbf{Y})),(f_2(\mathbf{X}) + g_2(\mathbf{Y})))_\pi)\}$,
- $\mathbf{X}$ *and* $\mathbf{Y}$ *are independent,*
- $H_\infty(\mathbf{X}) \geq k - 2n^\delta$, $H_\infty(\mathbf{Y}) \geq k - 2n^\delta$.

We present the construction of our advice generator and refer the reader to the full version of our paper for the proof. We claim that the function advGen computed by Algorithm 1 satisfies the above lemma. We first set up some parameters and ingredients.

- Let C be a large enough constant and $\delta' = \delta/C$.
- Let $n_0 = n^{\delta'}, n_1 = n_0^{c_0}, n_2 = 10n_0$, for some constant c_0 that we set below.
- Let $E : \{0,1\}^{2n} \to \{0,1\}^{n_3}$ be the encoding function of a linear error correcting code $\mathcal{C}$ with constant rate α and constant distance β.
- Let $\text{Ext}_1 : \{0,1\}^{n_1} \times \{0,1\}^{d_1} \to \{0,1\}^{\log(n_3)}$ be a $(n_1/20, \beta/10)$-seeded extractor instantiated using Theorem 18. Thus $d_1 = c_1 \log n_1$, for some constant c_1. Let $D_1 = 2^{d_1} = n_1^{c_1}$.
- Let $\text{Samp}_1 : \{0,1\}^{n_1} \to [n_3]^{D_1}$ be the sampler obtained from Theorem 17 using Ext_1.
- Let $\text{Ext}_2 : \{0,1\}^{n_2} \times \{0,1\}^{d_2} \to \{0,1\}^{\log(2n)}$ be a $(n_2/20, 1/n_0)$-seeded extractor instantiated using Theorem 18. Thus $d_2 = c_2 \log n_2$, for some constant c_2. Let $D_2 = 2^{d_2}$. Thus $D_2 = 2^{d_2} = n_2^{c_2}$.
- Let $\text{Samp}_2 : \{0,1\}^{n_2} \to [2n]^{D_2}$ be the sampler obtained from Theorem 17 using Ext_2.
- Set $c_0 = 2c_2$.
- Let $\text{i}\ell\text{Ext} : \{0,1\}^{D_2} \to \{0,1\}^{n_0}$ be the extractor from Theorem 12.
- Let $\text{LExt} : \{0,1\}^{2n} \times \{0,1\}^{n_0} \to \{0,1\}^{n_0}$ be a linear seeded extractor instantiated from Theorem 22 set to extract from min-entropy $n_1/100$ and error $2^{-\Omega(\sqrt{n_0})}$.

4.2 An Advice Correlation Breaker

We recall the setup of Theorem 28. $\mathbf{X}$ and $\mathbf{Y}$ are independent (n,k)-sources, $k \geq n - n^\delta$, $\pi : [2n] \to [2n]$ is an arbitrary permutation and $f_1, f_2, g_1, g_2 \in \mathcal{F}_n$ satisfy the following conditions:

- $\forall x \in \textit{support}(\mathbf{X})$ and $y \in \textit{support}(\mathbf{Y})$, $f_1(x) + g_1(y) \neq x$ or
- $\forall x \in \textit{support}(\mathbf{X})$ and $y \in \textit{support}(\mathbf{Y})$, $f_2(x) + g_2(y) \neq y$.

Algorithm 1: advGen(z)

Input: Bit-string $z = (x, y)_\pi$ of length $2n$, where x and y are each n bit-strings and $\pi : [2n] \to [2n]$ is a permutation.
Output: Bit string v of length n_4.

1 Let $z_1 = \text{Slice}(z, n_1), z_2 = \text{Slice}(z, n_2)$.
2 Let $S = \text{Samp}_1(z_1)$.
3 Let $T = \text{Samp}_2(z_2)$ and $z_3 = z_T$.
4 Let $r = \text{i}\ell\text{Ext}(z_3)$.
5 Let $w_1 = (E(z))_S$.
6 Let $w_2 = \text{LExt}(z, r)$.
7 Output $v = z_1, z_2, z_3, w_1, w_2$.

Further, we defined the following: $\overline{\mathbf{X}} = (\mathbf{X}, 0^n)_\pi$, $\overline{\mathbf{Y}} = (0^n \circ \mathbf{Y})_\pi$, $\overline{f_1(\mathbf{X})} = (f_1(\mathbf{X}), 0^n)_\pi$, $\overline{f_2(\mathbf{X})} = (0^n, f_2(\mathbf{X}))_\pi$, $\overline{g_1(\mathbf{Y})} = (g_1(\mathbf{Y}), 0^n)_\pi$ and $\overline{g_2(\mathbf{Y})} = (0^n, g_2(\mathbf{Y}))_\pi$. It follows that $\mathbf{Z} = \overline{\mathbf{X}} + \overline{\mathbf{Y}}$ and $\mathbf{Z}' = \overline{f_1(\mathbf{X})} + \overline{g_1(\mathbf{Y})} + \overline{f_2(\mathbf{X})} + \overline{g_2(\mathbf{Y})}$. Thus, for some functions $f, g \in \mathcal{F}_{2n}$, $\mathbf{Z}' = f(\overline{\mathbf{X}}) + g(\overline{\mathbf{Y}})$. Let $\overline{\mathbf{X}'} = f(\overline{\mathbf{X}})$ and $\overline{\mathbf{Y}'} = g(\overline{\mathbf{Y}})$.

The following is the main result of this section. Assume that we have some random variables such that $\mathbf{X}$ and $\mathbf{Y}$ continue to be independent, and $H_\infty(\mathbf{X}), H_\infty(\mathbf{Y}) \geq k - 2n^\delta$.

Lemma 9. *There exists an efficiently computable function* $\text{ACB} : \{0,1\}^{2n} \times \{0,1\}^{n_1} \to \{0,1\}^m$, $n_1 = n^\delta$ *and* $m = n^{\Omega(1)}$, *such that*

$$\text{ACB}(\overline{\mathbf{X}} + \overline{\mathbf{Y}}, w), \text{ACB}(\overline{f(\mathbf{X})} + \overline{g(\mathbf{Y})}, w') \approx_\epsilon \mathbf{U}_m, \text{ACB}(\overline{f(\mathbf{X})} + \overline{g(\mathbf{Y})}, w'),$$

for any fixed strings $w, w' \in \{0,1\}^{n_1}$ *with* $w \neq w'$.

We present the construction of our advice correlation breaker, and refer the reader to the full version of our paper for the proof. We prove that the function ACB computed by Algorithm 2 satisfies the conclusion of Lemma 9.

We start by setting up some ingredients and parameters.

- Let $\delta > 0$ be a small enough constant.
- Let $n_2 = n^{\delta_1}$, where $\delta_1 = 2\delta$.
- Let $\text{LExt}_1 : \{0,1\}^{n_2} \times \{0,1\}^d \to \{0,1\}^{d_1}$, $d_1 = \sqrt{n_2}$, be a linear-seeded extractor instantiated from Theorem 19 set to extract from entropy $k_1 = n_2/10$ with error $\epsilon_1 = 1/10$. Thus $d = C_1 \log n_2$, for some constant C_1. Let $D = 2^d = n^{\delta_2}$, $\delta_2 = 2C_1\delta$.
- Set $\delta' = 20C_1\delta$.
- Let $\text{LExt}_2 : \{0,1\}^{2n} \times \{0,1\}^{d_1} \to \{0,1\}^{n_4}$, $n_4 = n^{8\delta_3}$ be a linear-seeded extractor instantiated from Theorem 19 set to extract from entropy $k_2 = 0.9k$ with error $\epsilon_2 = 2^{-\Omega(\sqrt{d_1})} = 2^{-n^{\Omega(1)}}$, such that the seed length of the extractor LExt_2 (by Theorem 19) is d_1.

– Let $\text{ACB}' : \{0,1\}^{n_{1,acb'}} \times \{0,1\}^{n_{acb'}} \times \{0,1\}^{h_{acb'}} \to \{0,1\}^{n_{2,acb'}}$, be the advice correlation breaker from Theorem 24 set with the following parameters: $n_{acb'} = 2n, n_{1,acb'} = n_4, n_{2,acb'} = m = O(n^{2\delta_2}), t_{acb'} = 2D, h_{acb'} = n_1 + d, \epsilon_{acb'} = 2^{-n^{\delta}}$, $d_{acb'} = O(\log^2(n/\epsilon_{acb'}))$, $\lambda_{acb'} = 0$. It can be checked that by our choice of parameters, the conditions required for Theorem 24 indeed hold for $k_{1,acb'} \geq n^{2\delta_2}$.

Algorithm 2: $\text{ACB}(z, w)$

Input: Bit-strings $z = (x, y)_\pi$ of length $2n$ and bit string w of length n_1, where x and y are each n bit-strings and $\pi : [2n] \to [2n]$ is a permutation.
Output: Bit string of length m.

1 Let $z_1 = \text{Slice}(z, n_2)$.
2 Let v be a $D \times n_3$ matrix, with its i'th row $v_i = \text{LExt}_1(z_1, i)$.
3 Let r be a $D \times n_4$ matrix, with its i'th row $r_i = \text{LExt}_2(z, v_i)$.
4 Let s be a $D \times m$ matrix, with its i'th row $s_i = \text{ACB}'(r_i, z, w, i)$.
5 Output $\oplus_{i=1}^{D} s_i$.

4.3 The non-malleable extractor

We are now ready to present the construction of iℓNM that satisfies the requirements of Theorem 28.

– Let $\delta > 0$ be a small enough constant, $n_1 = n^{\delta}$ and $m = n^{\Omega(1)}$.
– Let $\text{advGen} : \{0,1\}^{2n} \to \{0,1\}^{n_1}$, $n_1 = n^{\delta}$, be the advice generator from Lemma 8.
– Let $\text{ACB} : \{0,1\}^{2n} \times \{0,1\}^{n_1} \to \{0,1\}^{m}$ be the advice correlation breaker from Lemma 9.

Algorithm 3: iℓNM(z)

Input: Bit-string $z = (x, y)_\pi$ of length $2n$, where x and y are each n bit-strings, and $\pi : [2n] \to [2n]$ is a permutation.
Output: Bit string of length m.

1 Let $w = \text{advGen}(z)$.
2 Output $\text{ACB}(z, w)$

The function iℓNM computed by Algorithm 3 satisfies the conclusion of Theorem 28 as follows: Fix the random variables $\mathbf{W}, \mathbf{W}'$. By Lemma 8, it follows that $\mathbf{X}$ remains independent of $\mathbf{Y}$, and with probability at least $1 - 2^{-n^{\Omega(1)}}$, $H_\infty(\mathbf{X}) \geq k - 2n_1$ and $H_\infty(\mathbf{Y}) \geq k - 2n_1$ (recall $k \geq n - n^{\delta}$). Theorem 28 is now direct using Lemma 9.

5 Open questions

Non-malleable codes for composition of functions. Here we give efficient constructions of non-malleable codes for the tampering class Lin ∘ 2ISS. Many natural questions remain to be answered. For instance, one open problem is to efficiently construct non-malleable codes for the tampering class 2SS ∘ Lin or 2ISS ∘ Lin, which as explained before is closely related to the question of constructing explicit (r, t)-ramp non-malleable secret-sharing schemes with binary shares, where $t < r$. It looks like one needs substantially new ideas to give such constructions. More generally, for what other interesting classes of functions $\mathcal{F}$ and $\mathcal{G}$ can we construct non-malleable codes for the composed class $\mathcal{F} \circ \mathcal{G}$? Is it possible to efficiently construct non-malleable codes for any tampering class $\mathcal{F} \circ \mathcal{G}$ as long as we have efficient non-malleable codes for the classes $\mathcal{F}$ and $\mathcal{G}$?

Other applications of seedless non-malleable extractors. The explicit seedless non-malleable extractors that we construct satisfy strong pseudorandom properties. A natural question is to find more applications of these non-malleable extractors in explicit constructions of other interesting objects.

Improved seedless extractors. We construct an extractor for 2-interleaved sources that works for min-entropy rate 2/3. It is easy to verify that there exists extractors for sources with min-entropy as low as $C \log n$, and a natural question here is to come up with such explicit constructions. Given the success in constructing 2-source extractors for low min-entropy [23,47], we are optimistic that more progress can be made on this problem.

Acknowledgements

We are grateful for useful comments from anonymous referees.

References

1. Aggarwal, D.: Affine-evasive sets modulo a prime. Information Processing Letters **115**(2), 382–385 (2015)
2. Aggarwal, D., Briët, J.: Revisiting the sanders-bogolyubov-ruzsa theorem in f p n and its application to non-malleable codes. In: 2016 IEEE International Symposium on Information Theory (ISIT). pp. 1322–1326. Ieee (2016)
3. Aggarwal, D., Damgard, I., Nielsen, J.B., Obremski, M., Purwanto, E., Ribeiro, J., Simkin, M.: Stronger leakage-resilient and non-malleable secret-sharing schemes for general access structures. IACR Cryptology ePrint Archive **2018**, 1147 (2018)
4. Aggarwal, D., Dodis, Y., Kazana, T., Obremski, M.: Non-malleable reductions and applications. In: Proceedings of the forty-seventh annual ACM symposium on Theory of computing. pp. 459–468. ACM (2015)
5. Aggarwal, D., Dodis, Y., Lovett, S.: Non-malleable codes from additive combinatorics. SIAM Journal on Computing **47**(2), 524–546 (2018)

6. Aggarwal, D., Dziembowski, S., Kazana, T., Obremski, M.: Leakage resilient non-malleable codes. In: Theory of Cryptography Conference, TCC 2015. pp. 398–426 (2015)
7. Aggarwal, D., Obremski, M.: A constant-rate non-malleable code in the split-state model. IACR Cryptology ePrint Archive **2019**, 1299 (2019)
8. Agrawal, S., Gupta, D., Maji, H.K., Pandey, O., Prabhakaran, M.: A rate-optimizing compiler for non-malleable codes against bit-wise tampering and permutations. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I. pp. 375–397 (2015)
9. Badrinarayanan, S., Srinivasan, A.: Revisiting non-malleable secret sharing. IACR Cryptology ePrint Archive **2018**, 1144 (2018)
10. Ball, M., Chattopadhyay, E., Liao, J.J., Malkin, T., Tan, L.Y.: Non-malleability against polynomial tampering. In: Crypto (2020), to appear
11. Ball, M., Dachman-Soled, D., Guo, S., Malkin, T., Tan, L.Y.: Non-malleable codes for small-depth circuits. In: 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS). pp. 826–837. IEEE (2018)
12. Ball, M., Dachman-Soled, D., Kulkarni, M., Malkin, T.: Non-malleable codes for bounded depth, bounded fan-in circuits. In: TCC (2016)
13. Ball, M., Guo, S., Wichs, D.: Non-malleable codes for decision trees. IACR Cryptology ePrint Archive **2019**, 379 (2019)
14. Blakley, G.R.: Safeguarding cryptographic keys. In: Proceedings of the 1979 AFIPS National Computer Conference. pp. 313–317 (1979)
15. Bogdanov, A., Ishai, Y., Viola, E., Williamson, C.: Bounded indistinguishability and the complexity of recovering secrets. In: Annual Cryptology Conference. pp. 593–618. Springer (2016)
16. Bogdanov, A., Williamson, C.: Approximate bounded indistinguishability. In: International Colloquium on Automata, Languages, and Programming (2017)
17. Carpentieri, M., Santis, A.D., Vaccaro, U.: Size of shares and probability of cheating in threshold schemes. In: EUROCRYPT 1993, 12th Annual International Conference on the Theory and Applications of Cryptographic Techniques (1993)
18. Chattopadhyay, E., Goyal, V., Li, X.: Non-malleable extractors and codes, with their many tampered extensions. In: STOC (2016)
19. Chattopadhyay, E., Kanukurthi, B., Obbattu, S.L.B., Sekar, S.: Privacy amplification from non-malleable codes. IACR Cryptology ePrint Archive **2018**, 293 (2018)
20. Chattopadhyay, E., Li, X.: Extractors for sumset sources. In: STOC (2016)
21. Chattopadhyay, E., Li, X.: 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. pp. 1171–1184. ACM (2017)
22. Chattopadhyay, E., Zuckerman, D.: Non-malleable codes against constant split-state tampering. In: Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science. pp. 306–315 (2014)
23. Chattopadhyay, E., Zuckerman, D.: Explicit two-source extractors and resilient functions. In: STOC (2016)
24. Chattopadhyay, E., Zuckerman, D.: New extractors for interleaved sources. In: CCC (2016)
25. Cheng, K., Ishai, Y., Li, X.: Near-optimal secret sharing and error correcting codes in ac0. In: TCC. pp. 424–458 (2017)

26. Cheraghchi, M., Guruswami, V.: Capacity of non-malleable codes. IEEE Trans. Information Theory **62**(3), 1097–1118 (2016). https://doi.org/10.1109/TIT.2015.2511784, https://doi.org/10.1109/TIT.2015.2511784
27. Cheraghchi, M., Guruswami, V.: Non-malleable coding against bit-wise and split-state tampering. J. Cryptology **30**(1), 191–241 (2017)
28. Chor, B., Goldreich, O.: Unbiased bits from sources of weak randomness and probabilistic communication complexity. SIAM J. Comput. **17**(2), 230–261 (1988)
29. Cohen, G.: Local correlation breakers and applications to three-source extractors and mergers. SIAM Journal on Computing **45**(4), 1297–1338 (2016)
30. Coretti, S., Dodis, Y., Tackmann, B., Venturi, D.: Non-malleable encryption: simpler, shorter, stronger. In: Theory of Cryptography Conference. pp. 306–335. Springer (2016)
31. Coretti, S., Maurer, U., Tackmann, B., Venturi, D.: From single-bit to multi-bit public-key encryption via non-malleable codes. In: Theory of Cryptography Conference. pp. 532–560. Springer (2015)
32. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**, 97–139 (2008)
33. Dodis, Y., Wichs, D.: Non-malleable extractors and symmetric key cryptography from weak secrets. In: STOC. pp. 601–610 (2009)
34. Dziembowski, S., Kazana, T., Obremski, M.: Non-malleable codes from two-source extractors. In: CRYPTO (2). pp. 239–257 (2013)
35. Dziembowski, S., Pietrzak, K.: Intrusion-resilient secret sharing. In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science. pp. 227–237. FOCS '07, IEEE Computer Society, Washington, DC, USA (2007). https://doi.org/10.1109/FOCS.2007.35, http://dx.doi.org/10.1109/FOCS.2007.35
36. Dziembowski, S., Pietrzak, K., Wichs, D.: Non-malleable codes. J. ACM **65**(4), 20:1–20:32 (Apr 2018). https://doi.org/10.1145/3178432, http://doi.acm.org/10.1145/3178432
37. Goyal, V., Ishai, Y., Maji, H.K., Sahai, A., Sherstov, A.A.: Bounded-communication leakage resilience via parity-resilient circuits. In: Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (2016)
38. Goyal, V., Kumar, A.: Non-malleable secret sharing. In: Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing. pp. 685–698. ACM (2018)
39. Goyal, V., Kumar, A.: Non-malleable secret sharing for general access structures. In: Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I. pp. 501–530 (2018)
40. Goyal, V., Pandey, O., Richelson, S.: Textbook non-malleable commitments. In: Proceedings of the forty-eighth annual ACM symposium on Theory of Computing. pp. 1128–1141. ACM (2016)
41. Gupta, D., Maji, H.K., Wang, M.: Constant-rate non-malleable codes in the split-state model. Tech. rep., Technical Report Report 2017/1048, Cryptology ePrint Archive (2018)
42. Guruswami, V., Umans, C., Vadhan, S.P.: Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. J. ACM **56**(4) (2009)

43. Kanukurthi, B., Obbattu, S.L.B., Sekar, S.: Four-state non-malleable codes with explicit constant rate. In: Theory of Cryptography Conference. pp. 344–375. Springer (2017)
44. Kanukurthi, B., Obbattu, S.L.B., Sekar, S.: Non-malleable randomness encoders and their applications. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 589–617. Springer (2018)
45. Li, X.: Improved two-source extractors, and affine extractors for polylogarithmic entropy. In: Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on. pp. 168–177. IEEE (2016)
46. Li, X.: Improved non-malleable extractors, non-malleable codes and independent source extractors. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing. pp. 1144–1156. STOC 2017 (2017)
47. Li, X.: Non-malleable extractors and non-malleable codes: Partially optimal constructions. Electronic Colloquium on Computational Complexity (ECCC) (2018)
48. Lin, F., Cheraghchi, M., Guruswami, V., Safavi-Naini, R., Wang, H.: Secret sharing with binary shares. CoRR **arXiv:cs/1808.02974** (2018)
49. Lin, F., Cheraghchi, M., Guruswami, V., Safavi-Naini, R., Wang, H.: Non-malleable secret sharing against affine tampering. CoRR **arXiv:cs/1902.06195** (2019)
50. Maurer, U., Wolf, S.: Privacy amplification secure against active adversaries. In: Advances in Cryptology — CRYPTO '97. vol. 1294, pp. 307–321 (Aug 1997)
51. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing. pp. 73–85 (1989)
52. Rao, A.: An exposition of Bourgain's 2-source extractor. Electronic Colloquium on Computational Complexity (ECCC) **14**(034) (2007)
53. Rao, A.: Extractors for low-weight affine sources. In: Proceedings of the 24th Annual IEEE Conference on Computational Complexity (2009)
54. Rasmussen, P.M.R., Sahai, A.: Expander graphs are non-malleable codes. CoRR (2018), https://arxiv.org/abs/1810.00106
55. Raz, R.: Extractors with weak random seeds. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing. pp. 11–20 (2005)
56. Raz, R., Reingold, O., Vadhan, S.: Extracting all the randomness and reducing the error in Trevisan's extractors. JCSS **65**(1), 97–128 (2002)
57. Raz, R., Yehudayoff, A.: Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors. Journal of Computer and System Sciences **77**, 167–190 (2011). https://doi.org/10.1016/j.jcss.2010.06.013
58. Shamir, A.: How to share a secret. Communications of the ACM **22**(11), 612–613 (1979)
59. Trevisan, L.: Extractors and pseudorandom generators. J. ACM pp. 860–879 (2001)
60. Trevisan, L., Vadhan, S.P.: Extracting Randomness from Samplable Distributions. In: IEEE Symposium on Foundations of Computer Science. pp. 32–42 (2000). https://doi.org/10.1109/SFCS.2000.892063
61. Zuckerman, D.: Randomness-optimal oblivious sampling. Random Structures and Algorithms **11**, 345–367 (1997)