Electronic Colloquium on Computational Complexity, Report No. 180 (2016)

# Non-Malleable Codes and Extractors for Small-Depth Circuits, and Affine Functions

Eshan Chattopadhyay*
Institute for Advanced Study, Princeton
eshanc@ias.edu

Xin Li†
Department of Computer Science,
John Hopkins University
lixints@cs.jhu.edu

November 15, 2016

**Abstract**

Non-malleable codes were introduced by Dziembowski, Pietrzak and Wichs [DPW10] as an elegant relaxation of error correcting codes, where the motivation is to handle more general forms of tampering while still providing meaningful guarantees. This has led to many elegant constructions and applications in cryptography. However, most works so far only studied tampering in the split-state model where different parts of the codeword are tampered independently, and thus do not apply to many other natural classes of tampering functions. The only exceptions are the work of Agrawal et al. [AGM+15], which studied non-malleable codes against bit permutation composed with bit-wise tampering, and the work of Ball et al [BDKM16], which studied non-malleable codes against local functions. However, in both cases each tampered bit only depends on a subset of input bits.

In this work, we study the problem of constructing non-malleable codes against more general tampering functions that act on the entire codeword. We give the first efficient constructions of non-malleable codes against $AC^0$ tampering functions and affine tampering functions. These are the first explicit non-malleable codes against tampering functions where each tampered bit can depend on all input bits. We also give efficient non-malleable codes against $t$-local functions for $t = o(\sqrt{n})$, where a $t$-local function has the property that any output bit depends on at most $t$ input bits. In the case of deterministic decoders, this improves upon the results of Ball et al [BDKM16], which can handle $t \leq n^{\frac{1}{4}}$.

All our results on non-malleable codes are obtained by using the connection between non-malleable codes and seedless non-malleable extractors discovered by Cheraghchi and Guruswami [CG14b]. Therefore, we also give the first efficient constructions of seedless non-malleable extractors against $AC^0$ tampering functions, $t$-local tampering functions for $t = o(\sqrt{n})$, and affine tampering functions. To derive our results on non-malleable codes, we design efficient algorithms to almost uniformly sample from the pre-image of any given output of our non-malleable extractor.

With the recent flurry of work on non-malleable extractors and various connections to more standard seedless extractors [CZ16,Li16], we believe that our results on non-malleable extractors and the techniques developed here are of independent interest.

*This research was party done when the author was a student in UT Austin. Partially supported by NSF Grants CCF-1412958, CCF-1526952, and the Simons Collaboration on Algorithms and Geometry.

†Partially supported by NSF Grant CCF-1617713.

ISSN 1433-8092

# 1 Introduction

Error-correcting codes encode a message $m$ into a longer codeword $c$ enabling recovery of $m$ even after part of $c$ is corrupted. We can view this corruption as a tampering function $f$ acting on the codeword, where $f$ is from some small allowable family $\mathcal{F}$ of tampering functions. The strict requirement of retrieving the encoded message $m$ imposes restrictions on the kind of tampering functions that can be handled. Unique decoding is limited by the minimum distance of the code, and various bounds are known in the case of list decoding. Hence, many natural classes of tampering functions cannot be handled in this framework.

One might hope to achieve a weaker goal of only detecting errors, possibly with high probability. Cramer et al. [CDF+08] constructed one such class of error-detecting codes, known as Algebraic Manipulation Detection codes (AMD codes), where the allowable tampering functions consist of all functions of the form $f_a(x) = a + x$. However error detection is impossible with respect to the family of constant functions. This follows since one cannot hope to detect errors against a function that always outputs some fixed codeword.

Dziembowski, Pietrzak and Wichs [DPW10] introduced non-malleable codes as a natural generalization of error-detecting codes. Informally, a non-malleable code with respect to a tampering function family $\mathcal{F}$ is equipped with a randomized encoder Enc and a deterministic decoder Dec such that $\mathrm{Dec}(\mathrm{Enc}(m)) = m$ and for any tampering function $f \in \mathcal{F}$ the following holds: for any message $m$, $\mathrm{Dec}(f(\mathrm{Enc}(m)))$ is either the message $m$ or is $\epsilon$-close (in statistical distance) to a distribution $D_f$ independent of $m$. The parameter $\epsilon$ is called the error. Besides being a natural generalization of error correcting codes, [DPW10] also showed that such non-malleable codes have several applications in tamper-resilient cryptography.

We now introduce some notions before formally defining non-malleable codes.

**Definition 1.1.** *For any function $f : S \to S$, $f$ has a fixed point at $s \in S$ if $f(s) = s$. We say $f$ has no fixed points in $T \subseteq S$, if $f(t) \neq t$ for all $t \in T$. $f$ has no fixed points if $f(s) \neq s$ for all $s \in S$.*

**Definition 1.2** (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.*

*Further, a function is called $t$-local if every output bit depends on at most $t$ input bits.*

**Definition 1.3.** *The statistical distance between two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ over some universal set $\Omega$ 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 $\epsilon$-close to $\mathcal{D}_2$ if $|\mathcal{D}_1 - \mathcal{D}_2| \leq \epsilon$ and denote it by $\mathcal{D}_1 \approx_\epsilon \mathcal{D}_2$.*

We now formally define non-malleable codes. We need to define the following function.

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

$$\mathrm{copy}^{(t)}((x_1, \ldots, x_t), (y_1, \ldots, y_t)) = (\mathrm{copy}(x_1, y_1), \ldots, \mathrm{copy}(x_t, y_t))$$

**Definition 1.4** (Coding schemes)**.** *Let* $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$ *and* $\mathrm{Dec} : \{0,1\}^n \to \{0,1\}^k \cup \{\perp\}$ *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* $(\mathrm{Enc}, \mathrm{Dec})$ *is a coding scheme with block length $n$ and message length $k$ if for all $s \in \{0,1\}^k$,* $\Pr[\mathrm{Dec}(\mathit{Enc}(s)) = s] = 1$ *(the probability is over the randomness in* Enc*).*

**Definition 1.5** (Non-malleable codes)**.** *A coding scheme* (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* $\epsilon$ *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 *such that for all messages* $s \in \{0,1\}^k$*, it holds that*

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

The rate of a non-malleable code $\mathcal{C}$ is given by $\frac{k}{n}$.

As an easy example, suppose the tampering function family is $\mathcal{F}_{constant}$, consisting of all constant functions, $f_c(x) = c$ for all $x$. In this case, to get a non-malleable code we can use any coding scheme and for any tampering function $f_c \in \mathcal{F}_{constant}$, we can take $D_{f_c}$ to be $\mathrm{Dec}(c)$ with probability 1.

Note that there cannot exist a code with block length $n$ which is non-malleable with respect to $\mathcal{F}_n$ (recall this is family of all functions from $n$ bits to $n$ bits). This follows since the tampering function could then use the function Dec to decode the message $m$, get a related message $m'$ by flipping all the bits in $m$, and use the encoding function to pick any codeword in $\mathrm{Enc}(m')$.

Therefore, it is natural to restrict the size of the family of tampering functions. It follows from the works in [DPW10, CG14a] that there exist non-malleable codes with respect to any tampering function family of size at most $2^{2^{\delta n}}$ with rate close to $1-\delta$ and error $2^{-\Omega(n)}$, for any constant $\delta > 0$. The bounds obtained in these works are existential, and some progress has been made since then in giving explicit constructions against useful classes of tampering functions.

A well studied model of tampering functions is the $C$-split-state model, where we assume that the codeword is partitioned into $C$ parts and each part is independently tampered by an arbitrary function. Several works [DPW10, CG14a, CG14b, DKO13, ADL14, CZ14, ADKO15, CGL16, Li16] studied this model resulting in explicit constructions of rate $\Omega(1/\log n)$ non-malleable codes in the 2-split state model (note that there cannot exist non-malleable codes in the 1-split model).

However, a severe limitation of non-malleable codes in the split-state model is that they cannot handle even simple tampering functions that depend on all bits of the codeword. In addition, very few work has constructed non-malleable codes for such tampering functions, partially because handling global functions seems challenging. Indeed, to the best of our knowledge, only the following two works give non-malleable codes against a class of global tampering functions not captured by the split-state model in the information theoretic setting: the first one is the work of Agrawal et al. [AGM+15], which studied non-malleable codes against bit permutation composed with bit-wise tampering. In this case, the authors gave optimal constructions of non-malleable codes achieving rate $1 - o(1)$ and error $2^{-\Omega(n)}$. The second one is a recent work by Ball et al. [BDKM16] , which constructed non-malleable codes against $t$-local functions with $t \leq n^{1/4}$ and rate $O(1/t^2)$.[1] We note that the class of tampering functions with bit permutation composed with bit-wise tampering is a special class of 1-local functions.

In this work, we make further progress towards constructing non-malleable codes that can handle more general global tampering functions. In particular, we give the first explicit construction of non-malleable codes when the tampering functions are restricted to be in $\mathrm{AC}^0$ (constant depth circuits with unbounded fan-in gates). We also construct efficient non-malleable codes against $t$-local functions, with $t \leq n^{\frac{1}{2}-\delta}$ for any constant $0 < \delta < 1$. This improves the tolerance of locality of [BDKM16] in the case of deterministic decoders. Finally, we give the first explicit construction of non-malleable codes against affine tampering functions. Notice that the class of tampering

[1] [BDKM16] also considers a relaxed notion of non-malleable codes where the decoder is allowed to be randomized as well, and shows how to handle locality up to $o(n/\log n)$ in this relaxed notion.

functions which consists of bit permutation composed with bit-wise tampering is a strict subset of affine functions. Thus in terms of the class of tampering functions, our work subsumes that of [AGM+15], although we do not achieve optimal rate and error as in their construction. We also note that for $\mathrm{AC}^0$ tampering functions and affine tampering functions, each tampered bit can depend on all input bits. This is in contrast to all previous works where any tampered bit only depends on a subset of the input bits. Our results thus give the first explicit constructions of non-malleable codes against such tampering functions.

Our constructions of non-malleable codes exploit a particularly useful connection between such codes and seedless non-malleable extractors found by Cheraghchi and Guruswami [CG14b].

We first recall the definition of seedless non-malleable extractors, which generalizes the more commonly studied seeded non-malleable extractors [DW09]. We ntoe that apart from the connection to non-malleable codes, seedless non-malleable extractors are interesting objects on their own as evident by recent connections found by Li [Li16] between constructing optimal two-source extractors and constructing seedless non-malleable extractors. Thus, our results on seedless non-malleable extractors may be of independent interest.

**Definition 1.6** ([CG88, Zuc90])**.** *The min-entropy of a source* $\mathbf{X}$ *is defined to be:* $H_\infty(\mathbf{X}) = \min_x(-\log(\Pr[\mathbf{X} = x]))$*. The min-entropy rate of a source* $\mathbf{X}$ *on* $\{0,1\}^n$ *is* $H_\infty(\mathbf{X})/n$*. Any source* $\mathbf{X}$ *on* $\{0,1\}^n$ *with min-entropy at least* $k$ *is called an* $(n,k)$*-source.*

We present a slightly simplified definition of seedless non-malleable extractors, and refer the reader to Section 5.1 for the general definition.

**Definition 1.7** (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 non-malleable extractor with respect to* $\mathcal{F}$ *and a class of sources* $\mathcal{X}$ *with error* $\epsilon$ *if for every distribution* $\mathbf{X} \in \mathcal{X}$ *and every tampering function* $f \in \mathcal{F}$*,*

$$|\mathrm{nmExt}(\mathbf{X}) \circ \mathrm{nmExt}(f(\mathbf{X})) - \mathbf{U}_m \circ \mathrm{nmExt}(f(\mathbf{X}))| \leq \epsilon.$$

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

The only known constructions of seedless non-malleable extractors are in the case where the family $\mathcal{F}$ is the $C$-split-state tampering class [CZ14, CGL16, Li16]. In particular, when $C = 2$, it amounts to constructing non-malleable extractors that have access to two independent sources $\mathbf{X}_1$ and $\mathbf{X}_2$, with $\mathbf{X}_1$ being independently tampered by a function $f_1$ and $\mathbf{X}_2$ being independently tampered by another function $f_2$.

In this work, we construct seedless non-malleable extractors that can handle functions which tamper the entire source globally. In particular, we give the first construction of seedless non-malleable extractors when the tampering functions are $\mathrm{AC}^0$ circuits. To the best of our knowledge, there was no known construction of seedless non-malleable extractors even when the tampering functions are from $\mathrm{NC}^0$. We also construct seedless non-malleable extractors for the class of $t$-local tampering functions and affine tampering functions. Using these constructions and the connection to non-malleable codes by Cheraghchi and Guruswami [CG14b], we obtain non-malleable codes against the corresponding families of tampering functions.

### 1.1 Our Results

We state our results on non-malleable extractors assuming the tampering functions have no fixed points, since it is easier this way. However, all our results generalize to handle fixed points and we refer the reader to later sections of the paper for the more general versions of the theorems.

We first define oblivious bit-fixing sources and affine sources to present our results.

**Definition 1.8.** *An oblivious bit-fixing source* $\mathbf{X}$ *on* $n$ *bits is a source where some subset of the bits are are chosen independently and uniformly and remaining bits are fixed (and do not depend on the choice of the random bits).*

**Definition 1.9.** *A distribution* $\mathbf{X}$ *on* $\{0,1\}^n$ *is called an affine source with min-entropy* $k$ *if it is uniform over some affine subspace in* $\mathbb{F}_2^n$ *of dimension* $k$*.*

Our first result gives explicit seedless non-malleable extractors for affine sources against affine tampering functions. We note that no such construction was known even for the case of full entropy.

**Theorem 1.** *For all* $n, k > 0$*, any* $\delta > 0$ *and* $k \geq n - n^{\delta/2}$*, there exists an efficient function* $\text{anmExt} : \{0,1\}^n \to \{0,1\}^m$*,* $m = n^{\Omega(1)}$*, such that if* $\mathbf{X}$ *is an affine source with min-entropy at least* $k$ *and* $\mathcal{A} : \{0,1\}^n \to \{0,1\}^n$ *is an affine function with no fixed point, then*

$$|\text{anmExt}(\mathbf{X}), \text{anmExt}(\mathcal{A}(\mathbf{X})) - \mathbf{U}_m, \text{anmExt}(\mathcal{A}(\mathbf{X}))| \leq 2^{-n^{\Omega(1)}}.$$

Next, we construct seedless non-malleable extractors for oblivious bit-fixing sources against $t$-local tampering functions. Again, we are not aware of any such explicit construction before (even for full entropy).

**Theorem 2.** *For any* $\delta > 0$ *and all* $n > 0$*, there exists an efficient function* $\text{localnmExt} : \{0,1\}^n \to \{0,1\}^m$*,* $m = n^{\Omega(1)}$*, such that if* $\mathbf{X}$ *is an oblivious bit-fixing source on* $n$ *bits with min-entropy* $k$ *and* $f : \{0,1\}^n \to \{0,1\}^n$ *is a* $t$*-local function,* $t \leq k/n^{\frac{1}{2}+\delta}$ *and has no fixed point, then*

$$|\text{localnmExt}(\mathbf{X}), \text{localnmExt}(f(\mathbf{X})) - \mathbf{U}_m, \text{localnmExt}(f(\mathbf{X}))| \leq 2^{-n^{\Omega(1)}}.$$

In particular, if we start with an oblivious bit-fixing source with min-entropy $k = \Omega(n)$, then we can handle $t$-local tampering functions for $t$ up to $n^{1/2-\delta}$.

Finally, we give seedless non-malleable extractors when the tampering functions are from $\text{AC}^0$.

**Theorem 3.** *For all* $n > 0$ *and any* $d = O(1)$*, there exists an efficient function* $\text{acnmExt} : \{0,1\}^n \to \{0,1\}^m$*,* $m = n^{\Omega(1)}$*, such that if* $\mathbf{X}$ *is uniform on* $n$ *bits and* $\mathcal{C} : \{0,1\}^n \to \{0,1\}^n$ *is an* $\text{AC}^0$ *circuit of size at most* $n^d$ *with no fixed point, then*

$$|\text{acnmExt}(\mathbf{X}), \text{acnmExt}(\mathcal{C}(\mathbf{X})) - \mathbf{U}_m, \text{acnmExt}(\mathcal{C}(\mathbf{X}))| \leq \frac{1}{n^{\Omega(\log n)}}.$$

Next, we derive our results on non-malleable codes based on the above extractors. Note that to use this connection (Theorem 4.1), we need additional properties from the non-malleable extractors. Specifically, we need to be able to efficiently sample almost uniformly from the pre-image of any output. We show how to do this in Section 8, which is an essential ingredient for construting efficient non-malleable codes.

The following theorem gives efficient non-malleable codes against affine tampering. Prior to this, there were no known explicit constructions in this model.

**Theorem 4.** *There exists a constant $\gamma > 0$ such that for all $n > 0$ there exists an efficient construction of non-malleable codes against affine tampering functions with block-length $n$, relative rate $n^{\gamma}/n$ and error $2^{-n^{\Omega(1)}}$.*

The following theorem gives efficient non-malleable codes against $t$-local tampering functions, allowing locality up to $t = n^{\frac{1}{2}-\delta}$. In the case of deterministic decoders, this improves upon the work of [BDKM16] where they only handle locality up to $n^{1/4}$.

**Theorem 5.** *There exist constants $\gamma, \delta > 0$ such that for all $n > 0$ there exists an efficient construction of non-malleable codes against $t$-local tampering functions, $t \leq n^{\frac{1}{2}-\delta}$ with block-length $n$, relative rate $n^{\gamma}/n$ and error $2^{-n^{\Omega(1)}}$.*

Next, we give the first explicit constructions of non-malleable codes against the class of $\mathrm{AC}^0$ tampering functions.

**Theorem 6.** *There exists a constant $\gamma > 0$ such that for all $n > 0$ there exists an efficient construction of non-malleable codes for $\mathrm{AC}^0$ tampering functions with block-length $n$, relative rate $(\log^2 n)/n$ and error $n^{-\Omega(\log n)}$.*

**Remark 1.10.** *Note that in our non-malleable codes for $\mathrm{AC}^0$ tampering functions, the block length is actually super-polynomial in the message length. However, the construction is still explicit and efficient in the sense that the encoding and decoding can be done in polynomial time of the block length. This notion of "efficient" is also generally used in previous constructions of non-malleable codes, such as that in [AGM+15].*

### Organization

In Section 2, we give an overview of our explicit constructions of non-malleable codes and extractors. We use Section 3 for preliminaries. In Section 4, we recall the connection between non-malleable codes and non-malleable extractors. We use Section 5 to construct non-malleable extractors for low-weight affine sources (see Definition 5) against affine adversaries. This forms the heart of our other extractor constructions. In Section 6, we construct non-malleable extractors against $t$-local functions and $\mathrm{AC}^0$ tampering. In Section 7, we present non-malleable extractors for affine sources against affine adversaries. Finally, in Section 8, we suitably modify our non-malleable extractor constructions and present efficient sampling algorithms for almost uniformly sampling from the pre-image of any output of the extractor. The non-malleable extractors along with the sampling algorithms directly imply our results on non-malleable codes.

## 2 Overview of the constructions and techniques

Here we provide an overview of our constructions and techniques. As mentioned earlier, our starting point is the general connection between non-malleable codes and seedless non-malleable extractors in [CG14b], where Cheraghchi and Guruswami showed that non-malleable extractors with sufficiently good parameters for a class of tampering functions implies non-malleable codes for such tampering functions. Therefore, we will first construct non-malleable extractors for $\mathrm{AC}^0$ tampering functions, local tampering functions and affine tampering functions, and then transform them into non-malleable codes by providing efficient algorithms to sample almost uniformly from the pre-image of any output.

We deal with $AC^0$ functions and local functions by an argument similar to that of Viola [Vio14]. Specifically, a standard application of the switching lemma shows that if one applies a random restriction to an $AC^0$ function, then with high probability the function collapses into a small depth decision tree. If the depth $d$ of the tree is small (e.g., $d < \log n$), then it is also a local function that depends on at most $2^d$ input bits. Thus we have reduced $AC^0$ functions to local functions. Now, given any output bit $y_i$, by picking a particular input bit $x_j$ that $y_i$ depends on, and fixing all the neighbors of $\{y_\ell\}$ except $x_j$, where the set of $\{y_\ell\}$ consists of all neighbors of $x_j$ (here we view the input-output dependence as a bipartite graph), we can ensure that $y_i$ only depends on $x_j$. We repeat this process until each $y_i$ is either fixed or depends on a single $x_j$. In this way, the function now becomes an affine function on the input bits. We note that an important difference between our case and the case of Viola [Vio14] is that, there one needs to preserve the min-entropy of the output bits (since the goal is to design an extractor for the output source), while here we don't need that. All we need is that the function now becomes an affine function, while some of the input bits are left unfixed. In other words, the initial input bits now become an oblivious bit-fixing source. Thus, we have reduced the problem of constructing non-malleable extractors with respect to $AC^0$ and local tampering functions to the problem of constructing non-malleable extractors for bit-fixing sources with respect to affine tampering functions (in fact, a very special class of affine tampering functions where each output bit depends on at most one input bit).

### 2.1 The non-malleable extractor

For simplicity, let us assume that the tampering function has no fixed point (we show in Section 5.1 that such non-malleable extractors also imply general non-malleable extractors that can deal with fixed points). Now our high level idea of constructing the non-malleable extractor will follow the non-malleable two-source extractor of [CGL16], in which one first obtains a small advice and then use a correlation breaker with advice (implicitly introduced in [CGL16] and formally defined in [Coh16b]) to get the final output. One can show that with high probability the advice obtained from the initial (untampered) sources is different from the advice obtained from the tampered sources. Thus the correlation breaker with advice guarantees that the output of the extractor is uniform given the tampered output.

Here we would like to do something similar. However, there are several tricky issues that we need to solve. First, how do we generate the advice? Second, once we have the advice, how do we construct the correlation breaker? Our starting point is the correlation breaker for affine sources developed in [Li15a, CL16b], which is based on alternating extraction between part of the affine source and the source itself, using strong linear seeded extractors. Assuming that we have already obtained the advice (which is different from its tampered version), and conditioned on the fixing of the advice and its tampered version, the original source is still an affine source, then we can use the above described correlation breaker to obtain an output that is uniform given its tampered version. Yet, this leads to another problem: to use the affine correlation breaker, we need the original affine source to have high entropy, while here (after the random restriction and fixing additional bits) the source can have quite small entropy.

Luckily, what we have here is a special kind of affine source — an oblivious bit-fixing source. For such sources we can first use the linear condenser developed in [Rao09, Vio14] to get another small affine source which has some entropy. The point is that this new affine source has a small length, which is smaller than the entropy of the original bit-fixing source. Thus we can apply a strong linear seeded extractor to convert the new source into a somewhere random matrix[2] (by

[2] A somewhere random matrix is a random matrix such that at least one row is uniform.

trying all possible seeds) and then use each row of the matrix to extract from the original source (by using another strong linear seeded extractor). The property of strong linear seeded extractors guarantees that we end up with another somewhere random matrix which has a small number of rows where each row is quite long. In addition, conditioned on the fixing of the short affine source, each row of the new matrix is a linear function of the original source. The new matrix is much easier to handle, and will be the starting point of our following constructions.

The somewhere random matrix also makes it easier for us to obtain the advice. Specifically, the advice in [CGL16] is obtained by taking a small slice of each source (recall that there we have two independent high min-entropy source), and use a function of the two small slices (actually, a two-source extractor such as inner product) to sample some bits from the encoding of each source by an error correcting code. The final advice is the concatenation of the two slices and the sampled bits. The analysis is that if the slices are different from their tampered version in the first place, then we are already done; otherwise with high probability the sampled bits are different from their tampered version. Here, since we have a somewhere random source, if we know which row is uniform, then we can try to take a small slice of that row and do something similar. In order to keep the source to be an affine source given the fixing of the advice, we will need to use a linear error correcting code. However, again we have two problems. First, we don't know which row is the uniform row. Thus, we will use each row to produce an advice, and we append the index of each row to the corresponding advice to guarantee that the advice from the "good" row is different from all the other advices (including the tampered advices and the advices from other rows). Since the number of rows is small compared to the length of each row, we can fix all the advices and each row still has high entropy. Now we can apply the correlation breaker to each row and the original source $\mathbf{X}$, and finally take the XOR of the outputs, which is still enough for our purpose. Second, and more seriously, unlike the setting of [CGL16] where we have two independent sources, now we only have one affine source $\mathbf{X}$. This means that each row in the somewhere random matrix is actually correlated with the original source $\mathbf{X}$. Therefore it is not clear if the sampling from an encoding of $\mathbf{X}$ can give us anything, since the random bits used to sample from the encoding are actually correlated with $\mathbf{X}$. It is indeed quite non-trivial to make this work, but given that the source is an affine source and the tampering function is also affine, we managed to show (in Section 5.4) that an appropriate modification of the advice generator in [CGL16] still works in this case. This gives the construction of our non-malleable extractor.

We note that the above construction works not only for bit-fixing sources, but also for any low-weight affine sources[3]. In addition, it also works for general affine sources with high entropy, since in this case we can just take a small slice to serve the purpose of what we get from the linear condenser.

## 2.2 Efficient sampling

We now turn to our algorithm to efficiently sample almost uniformly form the pre-image of any given output. Since our construction uses the affine correlation breaker which consists of multiple steps of alternating extraction, to directly inverting this process such as that done in [CGL16] seems pretty troublesome. Thus, we would like to use the much simpler sampling method recently developed by Li [Li16], which treats most of the construction details as a black box. A direct translation of that method to our case results in the following modified extractor construction and sampling algorithm: first, we modify the last step of the extractor construction and apply the correlation breaker to

[3] A low-weight affine source is the uniform distribution over some affine subspace which can be expressed as the linear combination of vectors with low Hamming weight.

two new larger slices of each row (or a slice of each row and the concatenation of another slice from every row). When we have obtained the outputs from the correlation breaker, we use each of them to extract from a new, longer part of the corresponding row using a linear invertible seeded extractor developed in [CGL16, Li16]. Note here we are using longer and longer slices from each row. The correlation breaker and the linear invertible seeded extractor both require their input source to have high entropy (conditioned on the fixing of previous random variables), which can be guaranteed since the length of each row is much larger than the number of rows in the somewhere random matrix. Finally, we take the XOR of all the outputs.

Now to sample from the pre-image of a given output, we first uniformly generate the small affine source $\mathbf{V}$ which we obtain from the linear condenser. This gives us a system of linear equations on the bits of the original source $\mathbf{X}$ (obtained from $\mathbf{V}$), and another system of linear functions of the bits of $\mathbf{X}$ corresponding to each row in the somewhere random matrix. Then, we uniformly generate the slice of each row and the advices, which give us a system of linear equations. We now uniformly generate a larger slice of each row and apply the correlation breaker to obtain the outputs. Finally, given the output of the non-malleable extractor we uniformly generate each part that appears in the XOR of the outputs of the linear invertible seeded extractor, and use the invertible extractor to obtain the pre-images which correspond to the parts of the rows in the somewhere random matrix that are used as the inputs to the linear invertible extractor. Thus we have obtained the somewhere random matrix, now we put together all the linear equations we have obtained before and we uniformly generate $\mathbf{X}$ according to these equations.

A serious problem with the above approach is that each time we obtain a system of linear equations, the rank of the coefficient matrix may not be the same. Thus if we just naively solve the equations and uniformly generate the pre-image, then overall the distribution may not be the uniform distribution over the pre-image of a given output. To solve this problem, we need to find a way to ensure that almost surely the coefficient matrix has the same rank. Again, this is quite non-trivial to achieve, and indeed most of our effort in the sampling part goes into ensuring this property.

We illustrate our ideas by staring from the first steps of the extractor construction. When we use the linear condenser to obtain the short affine source $V$, we create a system of linear equations between the bits of $\mathbf{X}$ and the bits of $\mathbf{V}$. Note that the coefficient matrix of this system is fixed. Next we use $\mathbf{V}$ and a strong linear seeded extractor to convert $X$ into a somewhere random matrix $\mathbf{R}$, where conditioned on the fixing of $\mathbf{V}$ each row is a linear function of $\mathbf{X}$. Here however, for different fixing of $\mathbf{V} = v$ the coefficient matrix is different and may have different rank. In addition, it may have different dependence on the previous system of equations (which is obtained from $\mathbf{V}$). Our solution to this problem is that we are going to manually pick a subset of the rows from the matrix $\mathbf{R}$, and for each row in the subset we manually pick a subset of its bits. We will ensure that by doing this, the new matrix is still a somewhere random matrix, but with the additional property that the linear functions given by the bits in this matrix are linearly independent, and further they are linearly independent of the previous system of equations (obtained from $\mathbf{V}$).

The idea is as follows. First, we can show by a standard argument that with high probability $(1 - 2^{-n^{\Omega(1)}})$ over the fixing of $\mathbf{V}$, most of the rows (i.e., $1 - 2^{-n^{\Omega(1)}}$ fraction) in the somewhere random matrix $\mathbf{R}$ are uniform. Assuming this happens, now our crucial observation is that, (1) if the output bits are uniform, then the linear functions corresponding to these bits must be linearly independent, and (2) linear independence of vectors can be tested efficiently. Indeed, if the linear functions corresponding to some bits are linearly dependent, then these bits already have a linear correlation and cannot be uniform. Therefore, assuming that we have $D$ rows in the matrix $\mathbf{R}$, then we know that there are at least $(1 - 2^{-n^{\Omega(1)}})D$ rows such that if we look at each row, then

the linear functions corresponding to the bits in the row are linearly independent. We can thus pick say $0.9D$ such rows. Note that this also guarantees that at least one row is still uniform, since we throw away only $0.1D$ rows. However, the bits in different rows might be correlated, and may also have correlation with the system of equations obtained from $\mathbf{V}$. Thus, our next step is to go through the $0.9D$ rows one by one, and for each row select some $\ell$ bits. We select these bits in such a way that they are linearly independent of the equations obtained from $\mathbf{V}$, and the bits selected later are linearly independent of the bits selected before. We can do this because the bits in each row are linearly independent, and thus they span a subspace of dimension $m$ if $m$ is the length of each row in the matrix $\mathbf{R}$. Suppose the length of $\mathbf{V}$ is $s$. Then as long as $m > s + D\ell$, we know that when we look at any particular row, there will always be bits that are linearly independent of the previous bits and the equations obtained from $\mathbf{V}$ (otherwise the subspace spanned by the bits in this row will have dimension at most $s + 0.9D\ell$). Finally, note that if a row is uniform, then any subset of its bits are also uniform. Therefore at the end we obtain a new somewhere random matrix $\mathbf{R}'$, whose coefficient matrix of the bits has a fixed rank and is linearly independent of the equations obtained from $\mathbf{V}$.

The above is the basic idea behind our approach to ensure the coefficient matrix of the final system of linear equations has the same rank. We then carry it out in the following steps of the extractor construction. In the next step where we generate the advices, two additional problems arise. First, we need to use a slice of each row of $\mathbf{R}'$ to sample from the encoding of the original source $\mathbf{X}$ by some linear error correcting code. We would like to make sure that the sampled bits have the same rank and are linearly independent of the bits in $\mathbf{V}$ and the used slices. This however is hard to guarantee if we sample some bits for each row. Thus, we modify the construction again to use the concatenation of the slices from each row (which is now a weak random source) to sample a single string from the encoding of $\mathbf{X}$. It is well known how to use weak random sources to do sampling (e.g., by using a seeded extractor). To ensure linearly independence, we use a dual BCH code, dBCH, to encode $\mathbf{X}$ with the parameter $t_{\mathrm{BCH}}$ set such that dBCH has constant distance and polynomial rate. Thus the corresponding BCH code has minimum distance at least $2t$. By a well known fact from coding theory, it thus guarantees that any $2t_{\mathrm{BCH}}$ bits in the codeword are linearly independent. Next, we throw away those bits that may be dependent of the bits in $\mathbf{V}$ and the used slices. We choose the parameters such that the number of bits from $\mathbf{V}$ and the used slices is at most $n^{\alpha}$ for some small constant $0 < \alpha < 1$, thus we can sample $n^{\beta}$ coordinates from the dBCH-encoding of $\mathbf{X}$ with $\beta > \alpha$ and $n^{\beta} << t$. Since the dBCH-code has a constant relative distance, we can show with high probability that the number of different bits (from the tampered version) in the sampled bits is a constant fraction. Thus even if we throw away $o(1)$ fraction of the sampled bits, we are still left with many different bits. When doing this, we also make sure the number of remaining sampled bits is the same, thus they have the same rank.

Second, the advice generating process is actually quite subtle due to the correlated sampling problem we described above; and in fact we need to use another random variable which is uniform conditioned on the fixing of the slices to extract from $\mathbf{X}$, to get another part of the advice. We do this by using a new, larger slice and use a linear seeded extractor with all possible seeds to convert it to a somewhere random matrix first, and then use each row to extract from $\mathbf{X}$ and concatenate all the outputs. When doing this, again we need to use the above described process to select from the output bits those that are linearly independent and independent of previously obtained bits from $\mathbf{V}$, the used slices and the previous advice.

After we are done with the above step, we actually introduced some new linear constraints which may be dependent of the remaining bits in each row of $\mathbf{R}'$ (note that we cannot ensure the bits we obtained from the advices are linearly independent of all the bits in the somewhere random

matrix $\mathbf{R}'$, since we need the size of the advices to be smaller than the length of each row in $\mathbf{R}'$, so that even conditioned on the fixing of the advices the good row in $\mathbf{R}'$ still has high min-entropy). Thus, we repeat the picking process above to pick from the remaining bits of $\mathbf{R}'$ those that are linearly independent and independent of previously obtained bits. As long as each row of $\mathbf{R}'$ is long enough, we can always succeed in this step. Finally we can compute the output of the correlation breakers, and use the linear invertible extractor to generate the remaining bits of $\mathbf{R}'$. Eventually, when we put them together we can argue that almost surely (with probability $1 - 2^{-n^{\Omega(1)}}$) the final system of linear equations has the same rank. Thus we can almost uniformly sample from the pre-image of any given output.

**Discussion and open problems.** In this paper we give the first explicit constructions of non-malleable codes against $\mathrm{AC}^0$ tampering functions and affine tampering functions, and give improved constructions of non-malleable codes against $t$-local tampering functions. We do this by giving the first explicit constructions of seedless non-malleable extractors against $\mathrm{AC}^0$ tampering functions, $t$-local tampering functions, and affine tampering functions. An obvious open problem here is to improve the rate and error of our codes. Especially, currently our non-malleable code for $\mathrm{AC}^0$ tampering functions are not so good, only achieving rate $\Omega(\log^2 n/n)$. We note that the main obstacle here is the error of our non-malleable extractor for $\mathrm{AC}^0$ tampering functions, which is now only $2^{-\Omega(\log^2 n)}$. This error comes from the fact that when we use the switching lemma, we need to make sure that the resulted decision tree has length $< \log n$, and thus the failure probability of a random restriction becomes $2^{-\Omega(\log^2 n)}$. Any way to get around this should be enough to result in a better non-malleable extractor and a better non-malleable code. Of course, ideally we would like to improve the rate of all our codes to some constant.

Another interesting open question is to construct non-malleable codes or extractors for larger classes of tampering functions. However, as noted in [BDKM16], since the separation of $\mathrm{NC}^1$ and P is not known, it is unlikely to come up with explicit non-malleable codes for $\mathrm{NC}^1$ tampering functions without computational assumptions.

# 3 Preliminaries

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 $\mathbf{X}$.

## 3.1 Explicit Extractors from Prior Work

**Definition 3.1.** *A function* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* $(k, \epsilon)$*-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 recall an optimal construction of strong-seeded extractors.

**Theorem 3.2** ([GUV09]). *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 is an explicit construction of linear seeded extractors.

**Theorem 3.3** ([Tre01, RRV02]). *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 $\epsilon$, where* $d = O\left(\frac{\log^2(n/\epsilon)}{\log(k/m)}\right)$.

**Theorem 3.4.** *There exists a constant $\alpha$ such that for all $n > 0$, and $\epsilon > 2^{-\alpha n}$, there exists a strong linear seeded extractor* iExt *satisfying the following: If $X$ is a $(n, 0.9n)$ source and $S$ is an independent uniform seed on $\{0,1\}^d$, $d = O(\log(n/\epsilon))$, then the following holds:*

$$|\text{iExt}(X,S), S - U_m, S| < 2^{-n^{\Omega(1)}},$$

*where $m = \Omega(d)$. Further for any $r \in \{0,1\}^m$ and any $s \in \{0,1\}^d$, $|\text{iExt}(\cdot, s)^{-1}(r)| = 2^{n-m}$.*

We also use a seeded extractor construction by Zuckerman [Zuc07] that achieves seed length $\log(n) + O(\log(\frac{1}{\epsilon}))$ to extract from any source with constant min-entropy rate.

**Theorem 3.5** ([Zuc07]). *For all $n > 0$ and constants $\alpha, \delta, \epsilon > 0$ there exists an efficient construction of a $(k = \delta n, \epsilon)$-strong seeded extractor* $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *with $m \geq (1-\alpha)k$ and $D = 2^d = O(n)$.*

We use a property of linear seeded extractors proved by Rao [Rao09].

**Lemma 3.6** ([Rao09]). *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 an explicit affine extractor constructed by Bourgain [Bou07].

**Theorem 3.7.** *For all $n, k > 0$ and any constant $\delta > 0$ there exists an explicit affine extractor* $\text{aExt} : \{0,1\}^n \to \{0,1\}^m$, $m = \Omega(k)$, *for min-entropy $k$ with error $2^{-\Omega(k)}$.*

### 3.2 Conditional Min-Entropy

**Definition 3.8.** *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. [DORS08].

**Lemma 3.9** ([DORS08]). *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 3.10** ([DORS08]). *If a random variable* $\mathbf{Y}$ *has support of size* $2^\ell$*, then* $\widetilde{H}_\infty(\mathbf{X}|\mathbf{Y}) \geq H_\infty(\mathbf{X}) - \ell$*.*

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

**Definition 3.11.** *A* $(k, \epsilon)$*-seeded average case seeded extractor* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *for min-entropy* $k$ *and error* $\epsilon$ *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 [DORS08] that any seeded extractor is also an average case extractor.

**Lemma 3.12** ([DORS08]). *For any* $\delta > 0$*, if* Ext *is a* $(k, \epsilon)$*-seeded extractor, then it is also a* $(k + \log(1/\delta), \epsilon + \delta)$*-seeded average case extractor.*

### 3.3 Some Probability Lemmas

The following result on min-entropy was proved by Maurer and Wolf [MW97].

**Lemma 3.13.** *Let* $\mathbf{X}, \mathbf{Y}$ *be random variables such that the random variable* $\mathbf{Y}$ *takes at* $\ell$ *values. Then*

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

**Lemma 3.14** ([BIW06]). *Let* $\mathbf{X}_1, \ldots, \mathbf{X}_\ell$ *be independent random variables on* $\{0,1\}^m$ *such that* $|\mathbf{X}_i - \mathbf{U}_m| \leq \epsilon$*. Then,* $|\sum_{i=1}^{\ell} \mathbf{X}_i - \mathbf{U}_m| \leq \epsilon^\ell$*.*

### 3.4 Sampling Using Weak Sources

Seeded extractors can be used as samplers with access to weak sources. Recall a graph-theoretic view of seeded extractors. A seeded extractor $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ can be viewed as an unbalanced bipartite graph $G_{\mathrm{Ext}}$ 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 $G_{\mathrm{Ext}}$.

**Theorem 3.15** ([Zuc97]). *Let* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a seeded extractor for min-entropy* $k$ *and error* $\epsilon$*. 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 3.16** ([Zuc97]). *Let* $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a seeded extractor for min-entropy* $k$ *and error* $\epsilon$*. Let* $\{0,1\}^d = \{r_1, \ldots, r_D\}$*,* $D = 2^d$*. Define* $\mathrm{Samp}(x) = \{\mathrm{Ext}(x, r_1), \ldots, \mathrm{Ext}(x, r_D)\}$*. Let* $\mathbf{X}$ *be an* $(n, 2k)$*-source. Then for any set* $R \subseteq \{0,1\}^m$*,*

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

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

# 4 Non-malleable codes via Seedless Non-Malleable Extractors

We recall a connection discovered between non-malleable codes and seedless non-malleable extractors by Sheraghchi and Guruswami [CG14b]. The version we state here is more general than stated in [CG14b]. It is easy to see that their proof generalizes to this more general version.

**Theorem 4.1.** *Let* $\mathrm{nmExt} : \{0,1\}^n \to \{0,1\}^m$ *be a polynomial time computable seedless non-malleable extractor that works for min-entropy* $n$ *with error* $\epsilon$ *with respect to a class of tampering functions* $\mathcal{F}$ *acting on* $\{0,1\}^n$*. Further suppose there is a sampling algorithm* Samp *that on any input* $z \in \{0,1\}^m$ *runs in time* $\mathrm{poly}(n)$ *and samples from a distribution that is* $\epsilon'$*-close to uniform on the set* $\mathrm{nmExt}^{-1}(s)$*.*

*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* $2^m\epsilon + \epsilon'$*.*

The non-malleable code is define in the following way: For any message $s \in \{0,1\}^m$, the encoder of the non-malleable code outputs $\mathrm{Samp}(s)$. For any codeword $c \in \{0,1\}^n$, the decoder outputs $\mathrm{nmExt}(c)$.

# 5 Seedless Non-Malleable Extractors for Low-Weight Affine Sources against Affine Adversaries

We begin by defining a sub-class of affine sources, called low-weight affine sources, which was first studied by Rao [Rao09].

**Definition 5.1** (Low-Weight Affine Source)**.** *Any affine source* $\mathbf{X}$ *with min-entropy* $k$ *which has a set of basis vectors* $\{v_1, \ldots, v_k\}$ *such that the hamming weight of each* $v_i$ *is bounded by* $w$ *is called a* $w$*-affine source.*

Rao [Rao09] constructed extractors for $k^\epsilon$-affine sources for min-entropy $k \geq \log^C n$, for some constant $c$. This was subsequently improved by Viola [Vio14] to achieve explicit extractors for $k^{1-\delta}$-affine sources.

We construct explicit seedless non-malleable extractors for $k^{1-\delta}$-affine sources against arbitrary affine tampering functions. However the extractor construction we present here is not invertible, which is a crucial property required from the extractor when we use the connection of Cheraghchi and Gurswami [CG14b] to construct corresponding non-malleable codes. In Section 8, we suitable modify this construction that enables us to make the extractor invertible. The modified construction in Section 8 builds on the construction in this section and is much more involved.

To construct the non-malleable extractor, we need to recall some components developed in prior work and develop many new components as well. In Section 5.1, we recall the definition of seedless non-malleable extractors, and then prove a couple of results which shows that for the cases relevant to this paper, it suffices to consider tampering functions without any fixed points. We recall some ingredients from prior work in Section 5.2 and 5.3. Our new components and the extractor construction are then presented in Section 5.4, Section 5.5 and Section 7.

## 5.1 Tampering functions with fixed points

Here we show that in the cases that are relevant to our applications, our non-malleable affine extractors for tampering functions with no fixed points imply general non-malleable affine extractors

for arbitrary tampering functions. We first recall the definition of a general seedless non-malleable extractor w.r.t. a class of tampering functions.

**Definition 5.2** (Seedless Non-Malleable Extractor)**.** *A function* nmExt : $\{0,1\}^n \to \{0,1\}^m$ *is a* $(k,\varepsilon)$*-seedless non-malleable extractor with respect to a class* $\mathcal{X}$ *of sources over* $\{0,1\}^n$ *and a class* $\mathcal{F}$ *of tampering functions acting on* $\{0,1\}^n$*, if for every* $\mathbf{X} \in \mathcal{X}$ *with min-entropy* $k$ *and every* $f \in \mathcal{F}$*, there is a distribution* $\mathcal{D}$ *over* $\{0,1\}^m \cup \{same^\star\}$ *such that for an independent* $\mathbf{Y}$ *sampled from* $D$*, we have*

$$(\text{nmExt}(\mathbf{X}), \text{nmExt}(f(\mathbf{X}))) \approx_\varepsilon (U_m, \text{copy}(\mathbf{Y}, U_m)),$$

*where the second* $U_m$ *is the same random variable as the first one.*

We define the following two classes of tampering functions. Let $\mathcal{F}_{affine}$ be the set of tampering functions from $\{0,1\}^n$ to $\{0,1\}^n$ where each output bit is an affine function of the input bits. Let $\mathcal{F}_{baffine}$ be a subset of $\mathcal{F}_{affine}$, where for each $f \in \mathcal{F}_{baffine}$ and every $i \in [n]$, $f(x)_i$ is either $x_j$ or $x_j \oplus 1$ for some $j \in [n]$, or a fixed bit. We show the following two lemmas.

**Lemma 5.3.** *Let* nmExt *be a* $(k - 2k/w, \varepsilon)$*-non-malleable extractor for weight* $w$ *affine sources, w.r.t affine tampering functions with no fixed points. Then* nmExt *is a* $(k, \varepsilon + (n+1)2^{-k/w})$*-non malleable extractor for oblivious bit-fixing sources, w.r.t.* $\mathcal{F}_{baffine}$*.*

*Proof.* For any $f \in \mathcal{F}_{baffine}$ and oblivious bit-fixing source $\mathbf{X}$ with entropy $k$, we define the following events. For every $i \in [n]$, let $E_i$ be the event s.t. $f(\mathbf{X})_j = \mathbf{X}_j, \forall j < i$ and $f(\mathbf{X})_i \neq \mathbf{X}_i$. Let $E_0$ be the event s.t. $\forall j \in [n], f(\mathbf{X})_j = \mathbf{X}_j$. Note that these are disjoint events that sum up to 1. Furthermore, each event also defines an affine subspace.

Now consider any event $E_s, s \in [n]$. Note that each constraint in the event is one the following forms: $x_i = x_j$, $x_i = x_j + 1$, or $x_i = c$, where $i, j \in [n]$ and $c$ is some fixed bit. In some cases, these constraints may not be consistent, which lead to an empty affine subspace that has no effect on our analysis. Thus from now on we only consider constraints that are consistent. We can view constraints of the form $x_i = x_j$, $x_i = x_j + 1$ as edges in the graph of $n$ vertices corresponding to $x_1, \cdots, x_n$. In this sense, there may exist paths of any length in the graph, where a path means that all the vertices on the path are determined by the starting vertex. In this sense, we can decompose all the constraints into disjoint paths, where each path is either fixed or has dimension one. We can thus now write this affine subspace as

$$\sum_{i \in T} \mathbf{X}_i v_i + b,$$

where $T \subseteq [n]$ is some subset where the corresponding bits $\{\mathbf{X}_i\}$ are not fixed, and $v_i$ is a vector that has 1's only in the coordinates of the path which $\mathbf{X}_i$ belongs to, and 0's everywhere else. Thus the weight of this vector is at most the length of the path. We now consider the dimension of this subspace, where we have two cases.

Case 1: the dimension is small, i.e., $\leq k - k/w$. In this case, the probability mass of this subspace is at most $2^{k-k/w} \cdot 2^{-k} = 2^{-k/w}$.

Case 2: the dimension is large, i.e., $> k - k/w$. In this case, we will further decompose the affine subspace by fixing those $\mathbf{X}_i$'s where the corresponding $v_i$ has weight larger than $w$. Note that a path can only occur in the bits of $\mathbf{X}$ which are originally unfixed (recall that $\mathbf{X}$ is a bit-fixing source). Thus the total number of such $\mathbf{X}_i$'s is at most $k/w$. Therefore for any particular fixing of these $\mathbf{X}_i$'s, the resulted new affine subspace has dimension at least $k - k/w - k/w = k - 2k/w$.

Note that now it has weight at most $w$, and $f$ has no fixed point in this affine subspace. Thus for this particular affine source $\mathbf{X}^\ell$ (the uniform distribution over this affine subspace), we have that

$$(\text{nmExt}(\mathbf{X}^\ell), \text{nmExt}(f(\mathbf{X}^\ell))) \approx_\varepsilon (U_m, \text{nmExt}(f(\mathbf{X}^\ell))).$$

Thus, we conclude that for any event $E_s, s \in [n]$, either its probability mass is at most $2^{-k/w}$, or conditioned on the event $E_s$, we have that

$$(\text{nmExt}(\mathbf{X}^s), \text{nmExt}(f(\mathbf{X}^s))) \approx_\varepsilon (U_m, \text{nmExt}(f(\mathbf{X}^s))),$$

where $\mathbf{X}^s$ is the affine source conditioned on $E_s$. Now consider the event $E_0$. We can use the same argument, where we either have that $\Pr[E_0] \leq 2^{-k/w}$, or that the affine source conditioned on $E_0$ is a convex combination of weight $w$ affine sources with entropy $k - 2k/w$. In the second case, since nmExt itself is an affine extractor for weight $w$ sources with entropy $k - 2k/w$, we have that

$$(\text{nmExt}(\mathbf{X}^0), \text{nmExt}(f(\mathbf{X}^0))) \approx_\varepsilon (U_m, U_m),$$

where $\mathbf{X}^0$ is the affine source conditioned on $E_0$, and the two $U_m$'s are the same random variables. Thus, now let $\mathcal{D}$ be the distribution that is the convex combination of $\{\text{nmExt}(f(\mathbf{X}^1)), \cdots, \text{nmExt}(f(\mathbf{X}^n)), same^\star\}$ with coefficients $\{\Pr[E_1], \cdots, \Pr[E_n], \Pr[E_0]\}$, we have that

$$(\text{nmExt}(\mathbf{X}), \text{nmExt}(f(\mathbf{X}))) \approx_{\varepsilon+(n+1)2^{-k/w}} (U_m, \text{copy}(\mathbf{Y}, U_m)).$$

□

Our second lemma is in a similar flavor, but simper to prove.

**Lemma 5.4.** *Let* nmExt *be a $(k - \ell, \varepsilon)$-non-malleable extractor for affine sources, w.r.t affine tampering functions with no fixed points. Then* nmExt *is a $(k, \varepsilon + (n+1)2^{-\ell})$-non malleable extractor for affine sources w.r.t. $\mathcal{F}_{affine}$.*

*Proof.* The proof is similar to the previous lemma. Specifically, for any $f \in \mathcal{F}_{affine}$ and affine source $\mathbf{X}$ with entropy $k$, we define the following events. For every $i \in [n]$, let $E_i$ be the event s.t. $f(\mathbf{X})_j = \mathbf{X}_j, \forall j < i$ and $f(\mathbf{X})_i \neq \mathbf{X}_i$. Let $E_0$ be the event s.t. $\forall j \in [n], f(\mathbf{X})_j = \mathbf{X}_j$. Note that these are disjoint events that sum up to 1. Furthermore, each event also defines an affine subspace.

Consider any event $E_s, s \in [n]$, we have two cases.

Case 1: the dimension is small, i.e., $\leq k - \ell$. In this case, the probability mass of this subspace is at most $2^{k-\ell} \cdot 2^{-k} = 2^{-\ell}$.

Case 2: the dimension is large, i.e., $> k - \ell$. In this case, note that conditioned on $E_s$, the affine source $X^s$ has entropy at least $k - \ell$, and $f$ has no fixed point in this affine subspace. Thus we have that

$$(\text{nmExt}(\mathbf{X}^s), \text{nmExt}(f(\mathbf{X}^s))) \approx_\varepsilon (U_m, \text{nmExt}(f(\mathbf{X}^s))).$$

Similarly, for the event $E_0$ we also have two cases. Either we have $\Pr[E_0] \leq 2^{-\ell}$ or $X^0$ has entropy at least $k - \ell$. In the latter case since nmExt is itself an affine extractor, we have

$$(\text{nmExt}(\mathbf{X}^0), \text{nmExt}(f(\mathbf{X}^0))) \approx_\varepsilon (U_m, U_m),$$

where the two $U_m$'s are the same random variables. Thus, now let $\mathcal{D}$ be the distribution that is the convex combination of $\{\mathrm{nmExt}(f(\mathbf{X}^1)), \cdots, \mathrm{nmExt}(f(\mathbf{X}^n)), same^\star\}$ with coefficients $\{\Pr[E_1], \cdots, \Pr[E_n], \Pr[E_0]\}$, we have that

$$(\mathrm{nmExt}(\mathbf{X}), \mathrm{nmExt}(f(\mathbf{X}))) \approx_{\varepsilon+(n+1)2^{-\ell}} (U_m, \mathrm{copy}(\mathbf{Y}, U_m)).$$

□

### 5.2 A Linear Condenser

We use a linear condenser for low-weight affine sources [Rao09, Vio14]. The condenser is essentially the parity check matrix of a binary code with high relative rate that achieves good relative distance as well (see e.g., [ABN+92]).

**Lemma 5.5** ([Rao09, Vio14])**.** *For any constant $0 < \alpha < 1$, and for all $n, k, w \in \mathbb{N}$, there exists a linear function $\mathrm{LCon} : \{0,1\}^n \to \{0,1\}^{n_1}$, $n_1 = O(wk^\alpha \log n)$ such that if $\mathbf{X}$ is a $w$-affine source with min-entropy $k$, then $\mathrm{LCon}(X)$ has min-entropy at least $k^\alpha$.*

### 5.3 Some Primitives from Prior Work

Let $\mathbf{Y}^1, \ldots, \mathbf{Y}^t$ be correlated r.v's. We recall an explicit construction from [CL16b], that breaks the correlations between these r.v's using an additional correlated source of the form $\mathbf{X}+\mathbf{Z}$, assuming $\mathbf{X}$ is independent of $\mathbf{Z}, \mathbf{Y}^1, \ldots, \mathbf{Y}^t$ (and $\mathbf{Z}$ is allowed to have arbitrary correlations with $\mathbf{Y}^1, \ldots, \mathbf{Y}^t$). The technique is based on the powerful method of alternating extraction, which was introduced by Dziembowski and Pietrzak [DP07] and has found many applications in many explicit constructions of pseudorandom objects (e.g., [DW09, Li13, Li15b, Coh15, CGL16, Li15a, CL16b]).

**Alternating Extraction** We briefly recall the method of alternating extraction. Assume that there are two parties, Quentin with a source $\mathbf{Q}$ and a uniform seed $\mathbf{S}_0$, and Wendy with a source $\mathbf{W}$. The alternating extraction protocol is an interactive process between Quentin and Wendy, and starts off with Quentin sending the seed $\mathbf{S}_0$ to Wendy. Wendy uses $\mathbf{S}_0$ and a strong seeded extractor $\mathrm{Ext}_w$ to extract a seed $\mathbf{R}_0 = \mathrm{Ext}_w(\mathbf{W}, \mathbf{S}_0)$ using $\mathbf{W}$, and sends $\mathbf{R}_0$ back to Quentin. This constitutes a round of the alternating extraction protocol. In the next round, Quentin uses a strong extractor $\mathrm{Ext}_q$ to extract a seed $\mathbf{S}_1 = \mathrm{Ext}_q(\mathbf{Q}, \mathbf{R}_o)$ from $\mathbf{Q}$ using $\mathbf{R}_0$, and sends it to Wendy and so on. The protocol is run for $h$ steps, where $h$ is an input parameter. Thus, the following sequence of random variables is generated:

$$\mathbf{S}_0, \mathbf{R}_0 = \mathrm{Ext}_w(\mathbf{W}, \mathbf{S}_0), \mathbf{S}_1 = \mathrm{Ext}_q(\mathbf{Q}, \mathbf{R}_0), \ldots, \mathbf{S}_h = \mathrm{Ext}_q(\mathbf{Q}, \mathbf{R}_{h-1}), \mathbf{R}_h = \mathrm{Ext}_w(\mathbf{W}, \mathbf{S}_h).$$

**Look-Ahead Extractor:** We define the following look-ahead extractor:

$$\mathrm{laExt}(\mathbf{W}, (\mathbf{Q}, \mathbf{S}_0)) = \mathbf{R}_1, \ldots, \mathbf{R}_h.$$

Algorithm 1 uses alternating extraction in a flip-flop way. This was introduced by Cohen [Coh15], and has been extensively used in explicit constructions of pseudorandom objects [Coh15, CGL16, CL16b, Coh16b, CL16a, Coh16a].

Algorithm 2 chains together several flip-flop steps along with an 'advice' string. This object is called a correlation breaker with advice, and was implicitly introduced by Chattopadhyay et al. [CGL16]. This has since been used in other constructions [CL16b, Coh16b].

**Algorithm 1:** flip-flop$(y^i, y^i_j, w, b)$

**Input:** Bit strings $y^i, y^i_j, w = x + z$ of length $n_1, n_2, n_1$ respectively, and a bit $b$.

**Output:** Bit string $y^i_{j+1}$ of length $n_2$.

**Subroutines:** Let $\mathrm{LExt}_1 : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^d$, $\mathrm{LExt}_2 : \{0,1\}^{n_2} \times \{0,1\}^d \to \{0,1\}^d$ be $(k, \epsilon)$-strong linear seeded extractors. Let $\mathrm{LExt}_3 : \{0,1\}^{n_1} \times \{0,1\}^d \to \{0,1\}^{n_2}$ be a $(k_2, \epsilon)$-strong linear seeded extractor.

Let $\mathrm{laExt} : \{0,1\}^n \times \{0,1\}^{n_2+d} \to \{0,1\}^{2d}$ be a look-ahead extractor for an alternating extraction protocol run for 2 rounds using $\mathrm{LExt}_1, \mathrm{LExt}_2$ as the seeded extractors.

1 Let $s^i_{0,j} = \mathrm{Slice}(y^i_j, d)$, $\mathrm{laExt}(w, (y^i_j, s^i_{0,j})) = r^i_{0,j}, r^i_{1,j}$

2 Let $\overline{y^i_{1,j}} = \mathrm{LExt}_3(y^i, r^i_{b,j})$

3 Let $\overline{s^i_{0,j}} = \mathrm{Slice}(\overline{y^i_{1,j}}, d)$, $\mathrm{laExt}(w, (\overline{y^i_{1,j}}, \overline{s^i_{0,j}})) = \overline{r^i_{0,j}}, \overline{r^i_{1,j}}$

4 Output $y^i_{j+1} = \mathrm{LExt}_3(y^i, \overline{r^i_{1-b,j}})$

**Algorithm 2:** ACB$(y^i, w, id)$

**Input:** Bit strings $y^i, w = x + z, id$ of length $n_1, n_1, h$ respectively.

**Output:** Bit string $y_{h+1}$ of length $n_2$.

1 Let $y^i_1 = \mathrm{Slice}(y, n_2)$

2 **for** $j = 1$ *to* $h$ **do**

3 $\quad y^i_{j+1} = \text{flip-flop}(y^i, y^i_j, w, id[j])$

4 **end**

5 Output $y^i_{h+1}$.

The following result was proved by Chattopadhyay and Li [CL16b]. We state here a version that slightly more general. Since this is easy to obtain from the original proof in [CL16b], we do not repeat the proof but briefly sketch the ideas to get this more general theorem. We allow the seed $\mathbf{Y}^1$ to be a $(n_1, n_1 - \lambda)$-source (instead of being fully uniform). Using the fact that any strong seeded extractor with error $\epsilon$ also works for a weak seed with deficiency $\lambda$ with error $2^\lambda \epsilon$ (e.g., see Lemma 6.4 in [CGL16]), the error parameter in Theorem 5.6 changes appropriately. Further, we allow tamperings on the sources $\mathbf{X}$ and $\mathbf{Z}$ as well, and it is easy to see that the proof in [CL16b] (which is based on alternating extraction) generalizes to handle this as well.

**Theorem 5.6.** *For all integers $n, n_1, n_2, k, k_1, k_2, t, d, h, \lambda$ and any $\epsilon > 0$, such that $k_1 \geq k + 8tdh + \log(1/\epsilon)$, $n_1 \geq k + 10tdh + (4ht + 1)n_2^2 + \log(1/\epsilon)$, $n_2 \geq k + 3td + \log(1/\epsilon)$ and $k_2 \geq n_2^{1.1}$, 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 $i \in \{2, t\}$, $id^1 \neq id^i$.*
- *$\mathbf{Y}^1_{h+1} = \mathrm{ACB}(\mathbf{Y}^1, \mathbf{X} + \mathbf{Z}, id^i)$ where ACB is the function computed by Algorithm 2.*
- *$\mathbf{Y}^i_{h+1} = \mathrm{ACB}(\mathbf{Y}^i, \mathbf{X}' + \mathbf{Z}', id^i)$, $i \in [2, t]$, where ACB is the function computed by Algorithm 2.*

*Then,*

$$\mathbf{Y}^1_{h+1}, \mathbf{Y}^2_{h+1}, \ldots, \mathbf{Y}^t_{h+1} \approx_{O((h+2^\lambda)\epsilon)} \mathbf{U}_{n_2}, \mathbf{Y}^2_{h+1}, \ldots, \mathbf{Y}^t_{h+1}.$$

### 5.4 Seedless Advice Generators for Affine Sources against Affine Adversaries

Chattopadhyay, Goyal and Li [CGL16] implicitly introduced objects called 'advice generators' in the context of constructing seeded non-malleable extractors. Cohen [Coh16b] formally defined these object, and follow-up works on non-malleable extractors also used explicit constructions of advice generators [CL16a, Coh16a]. Informally, an advice generator takes as input a weak source $\mathbf{X}$ and an independent seed $\mathbf{Y}$ to produce a short string such that for any random variable $\mathbf{Y}'$, such that $\mathbf{Y}' \neq \mathbf{Y}$, we have $\text{advGen}(\mathbf{X}, \mathbf{Y}) \neq \text{advGen}(\mathbf{X}, \mathbf{Y}')$ (with high probability).

Here we construct an advice generator for an affine source $\mathbf{X}$ assuming access to a short uniform seed $\mathbf{Y} = L(\mathbf{X})$, where $L$ is a linear function. Further we assume that $\mathbf{X}$ is tampered by an affine adversary $\mathcal{A}$ such that $\mathcal{A}$ has no fixed points. Note that unlike in previous work, the seed $\mathbf{Y}$ here is not independent of $\mathbf{X}$ and is in fact a deterministic function of $\mathbf{X}$.

**Theorem 5.7.** *There exists a constant $C$ such that for all $n > 0$, any constant $\delta > 0$ and $d \geq Cn^\delta$, there exists an efficiently computable function* $\text{advGen} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\ell$, $\ell = O(n^\delta)$ *such that if $\mathbf{X}$ is an affine source on $\{0,1\}^n$ with entropy at least $n^{10\delta}$ and $\mathbf{Y} = \mathcal{A}(\mathbf{X})$ is uniform on $d$ bits, where $A$ is some affine function, and $\mathbf{X}' = L(\mathbf{X}), \mathbf{Y}' = \mathcal{A}(\mathbf{X}')$ for some affine function $L : \{0,1\}^n \to \{0,1\}^n$ with no fixed points, then* $\Pr[\text{advGen}(\mathbf{X}, \mathbf{Y}) \neq \text{advGen}(\mathbf{X}', \mathbf{Y}')] \geq 1 - 2^{-n^{\Omega(1)}}$.

*Proof.* For easier presentation, we use the following notation: For any r.v $\mathbf{W} = f(\mathbf{X})$, we use $\mathbf{W}'$ to denote $f(\mathbf{X}')$.

We setup some ingredients for our construction.

- Let $n_1 = n^\delta, n_2 = n - n^\delta$.
- Let $\text{Enc} : \{0,1\}^n \to \{0,1\}^{n_3}$ be the encoding function of an asymptotically good linear binary code with constant relative rate $1/\lambda$ and constant relative distance $\beta$ (e.g, see [ABN+92]). Thus $n_3 = \lambda n$.
- Let $n_3 = 2^{n_4}$. Let $\text{Ext}_1 : \{0,1\}^{n_1} \times \{0,1\}^{d_1} \to \{0,1\}^{n_4}$ be the seeded extractor from Theorem 3.5 set to extract from min-entropy $n_1/2$ with error $\epsilon = \beta/2$. Thus $d_1 = \log(n_1) + O(1)$.
- Let $\text{Samp} : \{0,1\}^{n_1} \to (\{0,1\}^{n_4})^D$, be the sampler from Theorem 3.16 using $\text{Ext}_1$. Thus $D = 2^{d_1} = C_1 n_1$ for some constant $C_1$.
- Let $\ell_1 = C_1 n_1$ for a large enough constant $C_1$ such that $\text{aExt}_1 : \{0,1\}^{\ell_1} \to \{0,1\}^{n_1}$ is an affine extractor from Theorem 3.7 set to extract from min-entropy $\ell_1/2$ with error $2^{-\Omega(\ell_1)}$.
- Let $\text{LExt} : \{0,1\}^n \times \{0,1\}^{d_2} \to \{0,1\}^{n_5}$, $n_5 = n^{\delta/2}$, be a linear seeded extractor from Theorem 3.3 set to extract from min-entropy $\ell_1/4$ with error $2^{-n^{\delta/10}}$. It follows that $d_2 < n^{\delta/4}$.
- Let $\ell_2 = C_2 d_2$ for a large enough constant $C_2$ such that $\text{aExt}_2 : \{0,1\}^{\ell_2} \to \{0,1\}^{d_2}$ is an affine extractor from Theorem 3.7 set to extract from min-entropy $\ell_2/2$ with error $2^{-\Omega(\ell_2)}$.

We prove that $\Pr[\text{advGen}(\mathbf{X}, \mathbf{Y}) \neq \text{advGen}(\mathbf{X}', \mathbf{Y}')] > 1 - 2^{-n^{\Omega(1)}}$, where advGen is the function computed by Algorithm 3.

**Algorithm 3:** advGen$(x, y)$

**Input:** Bit strings $x, y$ of length $n, d$ respectively.
**Output:** Bit string $z$ of length $(\ell_1 + \ell_2 + D + n_5) = O(n^\delta)$.

Let $y = y_1 \circ y_2 \circ y_3$, where $|y_1| = \ell_1$, $|y_2| = \ell_2$.
Let $s_1 = \text{aExt}_1(y_1)$, $s_2 = \text{aExt}_2(y_2)$.
Let $w_1 = \text{Enc}(x)_{|\text{ Samp}(s_1)}$, $w_2 = \text{LExt}(x, s_2)$.
Output $z = y_1 \circ y_2 \circ w_1 \circ w_2$.

Assume that $\mathbf{Y}_1 = \mathbf{Y}'_1$ and $\mathbf{Y}_2 = \mathbf{Y}'_2$, since otherwise we directly have $\mathbf{Z} \neq \mathbf{Z}'$. Without loss of generality assume $\mathbf{X}$ is uniform on some subspace of dimension $k$. Let $\ell = \ell_1 + \ell_2$. Further let $\overline{\mathbf{Y}_1} = \mathbf{Y}_1 \circ 0^{n-\ell_1}$ and $\overline{\mathbf{Y}_2} = 0^{\ell_1} \circ \mathbf{Y}_2 \circ 0^{n-\ell}$. Using simple linear algebra, it follows that there exist disjoint subspaces $\mathbf{A}$ and $\mathbf{B}$ such that $\mathbf{X} = \mathbf{A}+\mathbf{B}$, $\dim(\mathbf{A}) = \ell$, $\dim(\mathbf{B}) = k-\ell$ and $T(\overline{\mathbf{Y}_1}+\overline{\mathbf{Y}_2}) = \mathbf{A}$ for some linear function $T : \{0,1\}^n \to \{0,1\}^n$.

We have,

$$\begin{aligned}\mathbf{W}_1 - \mathbf{W}'_1 &= \text{Enc}(\mathbf{X})_{|\text{ Samp}(\mathbf{S}_1)} - \text{Enc}(\mathbf{X}')_{|\text{ Samp}(\mathbf{S}_1)} \\ &= \text{Enc}(\mathbf{X} - \mathbf{X}')_{|\text{ Samp}(\mathbf{S}_1)},\end{aligned}$$

and

$$\begin{aligned}\mathbf{W}_2 - \mathbf{W}'_2 &= \text{LExt}(\mathbf{X}, \mathbf{S}_2) - \text{LExt}(\mathbf{X}', \mathbf{S}_2) \\ &= \text{LExt}(\mathbf{X} - \mathbf{X}', \mathbf{S}_2)\end{aligned}$$

Further,

$$\begin{aligned}\mathbf{X} - \mathbf{X}' &= \mathbf{A} + \mathbf{B} - L(\mathbf{A}) - L(\mathbf{B}) \\ &= T(\overline{\mathbf{Y}_1}) - L(T(\overline{\mathbf{Y}_1})) + T(\overline{\mathbf{Y}_2}) - L(T(\overline{\mathbf{Y}_2})) + \mathbf{B} - L(\mathbf{B})\end{aligned}$$

Now consider the following cases.

- $H_\infty(T(\overline{\mathbf{Y}_1}) - L(T(\overline{\mathbf{Y}_1}))) \leq \ell_1/2$.

  In this case, we can fix $T(\overline{\mathbf{Y}_1}) - L(T(\overline{\mathbf{Y}_1}))$, and it follows that $\mathbf{Y}_1$ has min-entropy at least $\ell_1/2$ after this fixing. Further note that $\mathbf{Y}_1$ is still an affine source, and hence it follows that $\mathbf{S}_1$ is $2^{-\Omega(\ell_1)}$-close to uniform. Further fix $\mathbf{B}, \mathbf{Y}_2$, noting that it is independent of $\mathbf{Y}_1$. This in fact fixes $\mathbf{X} - \mathbf{X}'$. Since $\mathbf{X} \neq \mathbf{X}'$, it follows that $\text{Enc}(\mathbf{X})$ differs from $\text{Enc}(\mathbf{X}')$ in at least $\beta$ fraction of the coordinates. Using Theorem 3.16, it follows that with probability at least $1-2^{-\Omega(n_1)}$, $\text{Samp}(\mathbf{S}_1)$ intersects one of the coordinates on which $\text{Enc}(\mathbf{X})$ differs from $\text{Enc}(\mathbf{X}')$ (and thus $\mathbf{W}_1 \neq \mathbf{W}'_1$).

- $H_\infty(T(\overline{\mathbf{Y}_1}) - L(T(\overline{\mathbf{Y}_1}))) > \ell_1/2$.

  We have,

  $$\begin{aligned}\mathbf{W}_2 - \mathbf{W}'_2 &= \text{LExt}(\mathbf{X} - \mathbf{X}', \mathbf{S}_2) \\ &= \text{LExt}(T(\overline{\mathbf{Y}_1}) - L(T(\overline{\mathbf{Y}_1})), \mathbf{S}_2) + \text{LExt}(T(\overline{\mathbf{Y}_2}) - L(T(\overline{\mathbf{Y}_2})), \mathbf{S}_2) + \text{LExt}(\mathbf{B} - L(\mathbf{B}), \mathbf{S}_2)\end{aligned}$$

  It follows that $\text{LExt}(T(\overline{\mathbf{Y}_1}) - L(T(\overline{\mathbf{Y}_1})), \mathbf{S}_2)$ is $2^{-n^{\Omega(1)}}$-close to uniform. We fix $\mathbf{S}_2$ since LExt is a strong seeded extractor, and thus $\text{LExt}(T(\overline{\mathbf{Y}_1}) - L(T(\overline{\mathbf{Y}_1})), s_2)$ is now a deterministic

function of $\mathbf{Y}_1$. We now fix $\mathbf{Y}_2, \mathbf{B}$ using the fact that they are independent of $\mathbf{Y}_1$. Thus, after these fixings $\mathbf{W}_2 - \mathbf{W}_2' = \mathrm{LExt}(T(\overline{\mathbf{Y}_1}) - L(T(\overline{\mathbf{Y}_1})), s_2) + \alpha$ (for some constant $\alpha \in \{0,1\}^{n_5}$), and hence is $2^{-n^{\Omega(1)}}$-close to uniform on average. Thus, $\Pr[\mathbf{W}_2 = \mathbf{W}_2'] \leq 2^{-n^{\Omega(1)}} + 2^{-n_5}$.

□

### 5.5 The Extractor Construction

**Theorem 5.8.** *There exists a constant $C_1$ such that for all $n, k, w > 0$ and any $\delta > 0$ with $w < n^{\beta}$, $\beta = \delta/(3C_1)$ and $k \geq n^{C_1\delta}$, there exists an efficient function* $\mathrm{anmExt} : \{0,1\}^n \to \{0,1\}^m$, *$m = k^{\Omega(1)}$, such that if $\mathbf{X}$ is a $w$-affine source with min-entropy at least $k$ and $\mathcal{A} : \{0,1\}^n \to \{0,1\}^n$ is an affine function with no fixed points, then*

$$|\mathrm{anmExt}(\mathbf{X}), \mathrm{anmExt}(\mathcal{A}(\mathbf{X})) - \mathbf{U}_m, \mathrm{anmExt}(\mathcal{A}(\mathbf{X}))| \leq 2^{-n^{\Omega(1)}}$$

We use the rest of the section to prove the above theorem. We reuse the following notation from previous section: For any r.v $\mathbf{W} = f(\mathbf{X})$, we use $\mathbf{W}'$ to denote $f(\mathbf{X}')$. We first set up the required ingredients with appropriate parameters.

- Let $\mathrm{LCon} : \{0,1\}^n \to \{0,1\}^{n_1}$ be a linear condenser from Theorem 5.5 set to work for min-entropy $k_1 = n^{\delta}$ and parameter $\alpha$ (which we fix below). Thus $n_1 = wk_1^{\alpha} \log n$.
- Let $\mathrm{LExt}_2 : \{0,1\}^n \times \{0,1\}^{d_2} \to \{0,1\}^{n_2}$, $n_2 = k^{1-2\delta}$, be a linear seeded extractor from Theorem 3.3 set to extract from min-entropy $k_3 = k/2$ with error $\epsilon_2 = 2^{-n^{\delta_1}}$, $\delta_1 = \alpha\delta/10$. Thus $d_2 = n^{2\delta_1} \log^2 n < k_1^{\alpha/2}$.
- Let $\mathrm{LExt}_1 : \{0,1\}^{n_1} \times \{0,1\}^{d_1} \to \{0,1\}^{d_2}$ be a linear seeded extractor from Theorem 3.3 set to extract from min-entropy $k_2 = k_1^{\alpha}$ with error $\epsilon_1 = 1/20$. It follows that $d_1 = O(\log(n_1))$. Let $D_1 = 2^{d_1}$. Thus $D_1 = n_1^{C_1}$, for some constant $C_1$.
- Fix $\alpha = 1/(2C_1), \beta = \delta/(2C_1)$.
- Let $\mathrm{advGen} : \{0,1\}^n \times \{0,1\}^{d_3} \to \{0,1\}^a$ be the advice generator from Theorem 5.7 set to work with parameter $\delta_{5.7} = \delta$. Thus we can fix $d_3 = C_{5.7} n^{\delta_{5.7}}$.
- Let $\mathrm{ACB} : \{0,1\}^{n_2} \times \{0,1\}^n \times \{0,1\}^h \to \{0,1\}^m$, $h = a + d_1$, be the correlation breaker from Theorem 5.6 setup using the following components:
  - $\mathrm{LExt}_{1,\mathrm{ACB}} : \{0,1\}^n \times \{0,1\}^{d_4} \to \{0,1\}^{d_4}$ and $\mathrm{LExt}_{2,\mathrm{ACB}} : \{0,1\}^m \times \{0,1\}^{d_4} \to \{0,1\}^{d_4}$ be $(k_{\mathrm{ACB}} = C_2 a^2 D_1^2 \log^2 n, \epsilon_{\mathrm{ACB}} = 2^{-4aD_1})$-strong linear seeded extractors, $m = k_{\mathrm{ACB}} + C_2 D_1 d_4 + 4aD_1$ for some large enough constant $C_2$, instantiated from Theorem 3.3. Thus, $d_4 < k_{\mathrm{ACB}}/2$, for an appropriately chosen $C_2$.
  - $\mathrm{LExt}_{3,\mathrm{ACB}} : \{0,1\}^{n_1} \times \{0,1\}^{d_4} \to \{0,1\}^m$ be a $(k_{2,\mathrm{ACB}} = m^{1.1}, \epsilon_{\mathrm{ACB}})$-strong linear seeded extractor.

The following bounds follow directly from our choice of parameters.

1. $D_1 \leq n^{\delta}$, $a \leq n^{\delta}$, $D_1 n_2 \leq k^{1-\delta}$,
2. $k_{\mathrm{ACB}} < m < n^{5\delta}$.

**Algorithm 4:** anmExt($x$)

**Input:** Bit string $x$ of length $n$.
**Output:** Bit string $z$ of length $m$.

1 Let $v = \text{LCon}(x)$.
2 Let $r$ be a $D_1 \times n_2$ matrix, whose $i$'th row $r_i$, is $\text{LExt}_2(x, \text{LExt}_1(v, i))$.
3 Let $s_i$ be a $D_1 \times m$ matrix whose $i$'th row $s_i$, is $\text{ACB}(r_i, x, \text{advGen}(x, \text{Slice}(r_i, d_3)) \circ i)$.
4 Output $z = \oplus_{j=1}^{D_1} s_i$.

**Claim 5.9.** *Conditioned on $\mathbf{V}, \mathbf{V}'$, the r.v $\mathbf{R}$ is $2^{-n^{\Omega(1)}}$-close to an affine somewhere random source.*

*Proof.* Using Theorem 5.5 it follows that $\mathbf{V}$ is a $(n_1, k_1^{\alpha})$-source. Now by Lemma 3.6, there exists a set $I \subseteq [D_1]$, $|I| \geq 0.9D_1$ such that for any $i \in [I]$, $\text{LExt}_1(\mathbf{V}, i) = \mathbf{U}_{d_2}$. We note that $\mathbf{V}, \mathbf{V}'$ are obtained by applying linear functions on $\mathbf{X}$. Thus there exist disjoint subspaces $\mathbf{A}$ and $\mathbf{B}$ such that $\mathbf{X} = \mathbf{A} + \mathbf{B}$, $H_\infty(\mathbf{B}) \geq k - 2n_1$ and $\mathbf{B}$ is independent of $\mathbf{V}, \mathbf{V}'$. Now for any $i \in I$, we have

$$\mathbf{R}_i = \text{LExt}_2(\mathbf{X}, \text{LExt}_1(\mathbf{V}, i)) = \text{LExt}_2(\mathbf{A}, \text{LExt}_1(\mathbf{V}, i)) + \text{LExt}_2(\mathbf{B}, \text{LExt}_1(\mathbf{V}, i)).$$

Since $\mathbf{B}$ is independent of $\mathbf{V}$, it follows that $\text{LExt}_2(\mathbf{B}, \text{LExt}_1(\mathbf{V}, i))$ is $\epsilon_2$-close to $\mathbf{U}_{n_2}$. Further, since $\text{LExt}_2$ is strong seeded, we fix $\text{LExt}_1(\mathbf{V}, i)$. Thus, $\text{LExt}_2(\mathbf{B}, \text{LExt}_1(\mathbf{V}, i))$ is now a deterministic function of $\mathbf{B}$. We also fix $\text{LExt}_2(\mathbf{A}, \text{LExt}_1(\mathbf{V}, i))$ since it is independent of $\mathbf{B}$. $\mathbf{R}_i$ remains $\epsilon_2$-close to $\mathbf{U}_{n_2}$ on average after these fixings. Further fix $\mathbf{V}, \mathbf{V}'$ noting that it does not affect the distribution of $\mathbf{R}_j$. Each $\mathbf{R}_i$ is now a linear function of $\mathbf{X}$. Hence it follows that $\mathbf{R}$ is $\epsilon_2$-close to an affine somewhere random source. □

**Claim 5.10.** $|(\mathbf{Z}, \mathbf{Z}') - (\mathbf{U}_m, \mathbf{Z}')| = 2^{-n^{\Omega(1)}}$.

*Proof.* We fix $\mathbf{V}, \mathbf{V}'$ and note by the previous claim that $\mathbf{R}$ is now an affine somewhere random source. Without loss of generality, suppose $\mathbf{R}_1$ is the random row. We claim that

$$|\mathbf{S}_1, \{\mathbf{S}_i : i \in [2, D_1]\}, \{\mathbf{S}'_i : i \in [D_1]\} - \mathbf{U}_m, \{\mathbf{S}_i : i \in [2, D_1]\}, \{\mathbf{S}'_i : i \in [D_1]\}| \leq 2^{-n^{\Omega(1)}}.$$

Since Claim 5.10 is direct from this, we now focus on proving the above bound. Theorem 5.7 guarantees that $\text{advGen}(\mathbf{X}, \text{Slice}(\mathbf{R}_1, d_3)) \neq \text{advGen}(\mathbf{X}', \text{Slice}(\mathbf{R}'_j, d_3))$ (with probability at least $1 - 2^{-n^{\Omega(1)}}$) for any $j \in [D_1]$. Fix the r.v's $\{\text{advGen}(\mathbf{X}, \text{Slice}(\mathbf{R}_i, d_3)) : i \in [D_1]\}, \{\text{advGen}(\mathbf{X}', \text{Slice}(\mathbf{R}'_i, d_3)) : i \in [D_1]\}$, and the following hold with probability at least $1 - 2^{-n^{\Omega(1)}}$:

- $\mathbf{R}_1$ has min-entropy at least $n_2 - 2aD_1$
- $\mathbf{X}$ remains an affine source and has min-entropy at least $k - 2aD_1 - n_1$
- $\mathbf{R}$ and $\mathbf{R}'$ are obtained by applying affine functions on $\mathbf{X}$
- $\text{advGen}(\mathbf{X}, \text{Slice}(\mathbf{R}_1, d_3)) \neq \text{advGen}(\mathbf{X}, \text{Slice}(\mathbf{R}'_j, d_3))$

Thus we can write $\mathbf{X} = \overline{\mathbf{A}} + \overline{\mathbf{B}}$ such that $\overline{\mathbf{B}}$ is independent of $\{\mathbf{R}, \mathbf{R}'\}$ and $H_\infty(\overline{\mathbf{B}}) = k_b \geq k - 2aD_1 - n_1 - 2D_1n_2 > k/2$. Let $\lambda = 2aD_1$. The claim is now direct from Theorem 5.6 noting that the following hold:

- $k_b \geq C(k_{\text{ACB}} + d_4D_1h + \log(1/\epsilon_{\text{ACB}}))$, for any constant $C$.

- $n_2 \geq C(k_{\text{ACB}} + d_4 D_1 h + h D_1 m^2 + \log(1/\epsilon_{\text{ACB}}))$, for any constant $C$.
- $(2^\lambda + h)\epsilon = 2^{-n^{\Omega(1)}}$.

□

# 6 Seedless Non-Malleable Extractors for Local and $\text{AC}^0$ Adversaries

The main results in this section is are seedless non-malleable extractor against t-local and $\text{AC}^0$ functions. Our main idea is to use techniques developed by Viola [Vio11], where he designed extractors for sources sampled by $\text{AC}^0$ circuits. We first reduce the problem of constructing non-malleable extractors against $\text{AC}^0$ adversaries to the problem of constructing non-malleable extractors against local adversaries. Next, we show reduce the problem of constructing non-malleable extractors for local adversaries to the problem of constructing extractors for low-weight affine sources against affine adversaries. Note that in Theorem 5.8 we exactly construct such non-malleable extractors, and hence this gives non-malleable extractors against local and $\text{AC}^0$ functions.

The following are the main results of this section. We derive these results assuming the reductions in Section 6.1 and Section 6.2 (see Lemma 6.4 and Lemma 6.5).

**Theorem 6.1.** *For any $\delta > 0$ and for all $n > 0$, there exists an efficient function* $\text{localnmExt} : \{0,1\}^n \to \{0,1\}^m$, *$m = n^{\Omega(1)}$, such that if $\mathbf{X}$ is an oblivious bit-fixing source on $n$ bits with min-entropy $k$ and $f : \{0,1\}^n \to \{0,1\}^n$ is a $t$-local function, $t \leq k/n^{\frac{1}{2}+\delta}$, then there exists a distribution $\mathbf{Y}$ on $\{0,1\}^m \cup \{same^\star\}$ that is independent of $\mathbf{X}$, and*

$$|\text{localnmExt}(\mathbf{X}), \text{localnmExt}(f(\mathbf{X})) - \mathbf{U}_m, \text{copy}(\mathbf{Y}, \mathbf{U}_m))| \leq 2^{-n^{\Omega(1)}}.$$

*Proof.* Let $\text{anmExt} : \{0,1\}^n \to \{0,1\}^m$, $m = n^\gamma$ for some small enough constant $\gamma$, be the non-malleable extractor from Theorem 5.8 set to extract from min-entropy $n^\delta - n^{\delta/2}$ with the parameter $w$ set to $n^{\delta/2}$ and error $2^{-n^{\Omega(1)}}$. Define $\text{localnmExt} := \text{anmExt}$. Using Lemma 6.5, it follows that $(\mathbf{X}, f(\mathbf{X}))$ is a convex combination of sources of the form $(\mathbf{Z}_i, A_i(\mathbf{Z}_i))$, such that each $\mathbf{Z}_i$ is an oblivious bit-fixing source with min-entropy at least $n^{2\delta}/2$ and $A_i$ is a 1-local function. Now by Lemma 5.3, it follows that anmExt is a non-malleable extractor for $\mathbf{Z}_i$ against 1-local tampering with error $2^{-n^{\Omega(1)}} + n2^{-n^{\delta/2}}$. The theorem now follows by a convex combination argument. □

**Theorem 6.2.** *For all $n > 0$ and any $d = O(1)$, there exists an efficient function* $\text{acnmExt} : \{0,1\}^n \to \{0,1\}^m$,*$m = n^{\Omega(1)}$, such that if $\mathbf{X}$ is uniform on $n$ bits and $\mathcal{C} : \{0,1\}^n \to \{0,1\}^n$ is an $\text{AC}^0$ circuit, there exists a distribution $\mathbf{Y}$ on $\{0,1\}^m \cup \{same^\star\}$ that is independent of $\mathbf{X}$, and*

$$|\text{acnmExt}(\mathbf{X}), \text{acnmExt}(\mathcal{C}(\mathbf{X})) - \mathbf{U}_m, \text{copy}(\mathbf{Y}, \mathbf{U}_m))| \leq \frac{1}{n^{\Omega(\log n)}}.$$

*Proof.* Let localnmExt be the extractor from Theorem 6.1 set to extract from min-entropy $k = n^{1-\gamma}$, for a small enough constant $\gamma > 0$ and locality parameter $t$ set to $n^\gamma$. Define $\text{acnmExt} = \text{localnmExt}$. Now by Lemma 6.4, it follows that $(\mathbf{X}, \mathcal{C}(\mathbf{X}))$ is $n^{-\Omega(\log n)}$-close to a convex combination of distributions of the form $(\mathbf{Z}_i, f_i(\mathbf{Z}_i))$, where each $\mathbf{Z}_i$ is an oblivious bit-fixing source with min-entropy at least $n^{1-\gamma}$ and $f_i$ is a $n^\gamma$-local function. The theorem now follows directly. □

### 6.1 A Reduction from $\mathrm{AC}^0$ to Local Adversaries

The reduction from $\mathrm{AC}^0$ to local adversaries is based on the well known switching lemma [Has87]. We first recall the definition of a random restriction. A $p$-restriction $\rho$ acting on a string $x \in \{0,1\}^n$ independently fixes each bit to 0 with probability $(1-p)/2$, to 1 with probability $(1-p)/2$ and leaves it unfixed with probability $p$. Let $P_p$ denote the set of all $p$-restrictions acting on $n$ bits. For a function $f : zo^n \to \{0,1\}^m$, and a $p$-restriction $\rho$, let $f_\rho$ denote the function after applying the restriction $\rho$.

**Lemma 6.3** ([Has87])**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a function computed by a size $s$ $\mathrm{AC}^0$ circuit of depth $d$. Then, for a uniformly drawn $p$-restriction $\rho$ from $P_p$, we have*

$$\Pr_\rho[f(x_{|\rho}) \text{ is not a } 2^t\text{-local function}] \leq s(9p^{1/d}t)^t.$$

**Lemma 6.4.** *For any $d = O(1)$ and any constants $\delta, \gamma > 0$ the following holds: Let $\mathcal{C} : \{0,1\}^n \to \{0,1\}^n$ be any $\mathrm{AC}^0$ circuit of size $n^d$ and $\mathbf{X}$ be the uniform distribution on $n$ bits. Then there exists oblivious bit-fixing sources $\mathbf{X}_1, \ldots, \mathbf{X}_\ell$ on $\{0,1\}^n$, and $f_1 : \{0,1\}^n \to \{0,1\}^n, \ldots, f_\ell : \{0,1\}^n \to \{0,1\}^n$, such that*

- *$H_\infty(\mathbf{X}_i) \geq n^{1-\gamma}$ for each $i \in [\ell]$,*
- *Each $f_i$ is a $n^\delta$-local function.*
- *$(\mathbf{X}, \mathcal{C}(\mathbf{X}))$ is $1/n^{\Omega(\log n)}$-close to a convex combination of the distributions $(\mathbf{X}_i, f_i(\mathbf{X}_i))$.*

*Proof.* Let $p = n^{\gamma/2}$ and $t = \delta \log n$. $\mathbf{X}$ is a convex connotation of the sources $\mathbf{X}_{|\rho}$, $\rho \in P_p$. Uniformly sample a $\rho$ from $P_p$. By a Chernoff bound, it follows that the probability that at least $n^{1-\gamma}$ of the bits in $\mathbf{X}$ are unfixed is at least $1 - 2^{-n^{\Omega(1)}}$. Further, by Lemma 6.3, the probability that $\mathcal{C}_\rho$ is not a $2^t$-local function is bounded by $n^{O(1)}(9/(tn^{\gamma/(2d)}))^t = n^{-\Omega(\log n)}$. The lemma now follows by a union bond. □

### 6.2 A Reduction from Local to Affine Adversaries

Our reduction from local to affine adversaries uses some ideas from [Vio11] but the analysis is much simpler.

**Lemma 6.5.** *Let $\mathbf{X}$ be an oblivious bit-fixing source on $n$ bits with min-entropy $k$, and let $f : \{0,1\}^n \to \{0,1\}^n$ be any $t$-local function. Then there exists sources $\mathbf{Z}^1, \ldots, \mathbf{Z}^\ell$, each on $n$ bits, and affine functions $A_1, \ldots, A_\ell$, such that*

- *$(\mathbf{X}, f(\mathbf{X}))$ is a convex combination of $(\mathbf{Z}^i, A_i(\mathbf{Z}^i))$, $i \in [\ell]$.*
- *Each $\mathbf{Z}_i$ is an oblivious bit-fixing source with min-entropy at least $k^2/(8t^2n)$.*

*Proof.* Without loss of generality, suppose $x_1, \ldots, x_k$ are the random bits in the oblivious source $X$. Let $\mathbf{Y} = f(\mathbf{X})$. Consider a bipartite graph $G$ with left nodes $\mathcal{L} = \{x_1, \ldots, x_k\}$ and right nodes $\mathbb{R} = \{y_1, \ldots, y_n\}$, with an edge present between $x_i$ and $y_j$ if the bit $y_j$ depends on $x_i$. Note that since $f$ is $t$-local, it follows that the degree of any right vertex in $G$ is bounded by $t$. Let $d_\ell = 2tn/k$. For any vertex $v$, let $N(v)$ denote the set neighbors of $v$. Further, for a set $S$ of vertices, let $N(S)$ denote the union of the neighbors of $v$.

By a markov argument, it follows that at most $k/2$ of the left vertices have degree more than $d_\ell$. Let $\mathcal{L}'$ set of vertices on the left with degree more $d_\ell$. Let $k' = k^2/(8t^2n)$. For any set $S \subset [n]$, we use $x_S$ to denote the set $\{x_i : i \in S\}$. Consider the following iterative process:

1. Uniformly sample, and fix the variables in $\mathcal{L}'$.
2. Let $\mathcal{L}_0 = \mathcal{L} \setminus \mathcal{L}'$, $V_0 = \emptyset$.
3. For $i = 1$ to $k'$, do:
4. Pick some $x_j$ in $\mathcal{L}_{i-1}$, and uniformly sample and fix the variables in $N(N(x))$.
5. Set $\mathcal{L}_i$ to $\mathcal{L}_{i-1} \setminus N(N(x_j))$, and set $V_i = V_{i-1} \cup \{j\}$.
6. Set the bits $k+1, \ldots, x_n$ consistent with $\mathbf{X}$.
7. Sample uniformly and fix the unfixed variables in $\{x_j : j \in [n] \setminus V_{k'}\}$.
8. Let $V'_{k'} = [n] \setminus V_{k'}$.
9. Let $A : \{0,1\}^n \to \{0,1\}^n$ be the function defined in the following way: For any $w \in \{0,1\}^n$, $A(w_1, \ldots, w_n) = f(x_{V'_{k'}} \circ w_{V_{k'}})$.
10. Sample $\{x_j : j \in V_{k'}\}$ uniformly.
11. Output $(x, A(x))$.

Note that at each stage, at most $d_\ell t$ of the variables in $\mathcal{L}$ are fixed, and hence each $\mathcal{L}_i$ is non-empty. It is easy to see that the above sampling process indeed produces the distribution $(\mathbf{X}, f(\mathbf{X}))$. Each source $\mathbf{Z}^i$ corresponds to some fixing of the variables $\{x_j : j \in V'_{k'}\}$. Further, note that $f(\mathbf{Z}^i) = A(\mathbf{Z}^i)$. The source $\mathbf{Z}^i$ is indeed an oblivious bit-fixing source with min-entropy $k'$, with each bit indexed by $V_{k'}$ being uniform and independent. Further, note that each output bit of $A(z)$ depends on exactly one variable in $\{z_j : j \in V_{k'}\}$. Since the variables in the coordinates $[n] \setminus V_{k'}$ are fixed in $\mathbf{Z}^i$, it follows that $A$ is an affine function on the source $\mathbf{Z}^i$. This completes the proof of the lemma. □

# 7 Seedless Non-Malleable Extractors for Affine Sources against Affine Adversaries

In this section we construct seedless non-malleable extractors for arbitrary affine sources against affine adversaries. We construct extractors for affine sources on $n$ bits with min-entropy at least $n - n^\delta$ and error $2^{-n^{\Omega(1)}}$ assuming that the affine adversary has no fixed points. By Lemma 5.4, it directly implies affine extractors for affine sources with min-entropy at least $n - n^\delta/2$ and error $2^{-n^{\Omega(1)}} + n2^{-n^\delta/2}$ against arbitrary affine functions. The following is the main theorem in this section.

**Theorem 7.1.** *For all $n, k > 0$, any $\delta > 0$ and $k \geq n - n^\delta$, there exists an efficient function* $\mathrm{anmExt} : \{0,1\}^n \to \{0,1\}^m$ *such that if $\mathbf{X}$ is an affine source with min-entropy at least $k$ and $\mathcal{A} : \{0,1\}^n \to \{0,1\}^n$ is an affine function with no fixed points, then*

$$|\mathrm{anmExt}(\mathbf{X}), \mathrm{anmExt}(\mathcal{A}(\mathbf{X})) - \mathbf{U}_m, \mathrm{anmExt}(\mathcal{A}(\mathbf{X}))| \leq 2^{-n^{\Omega(1)}}.$$

The construction is almost the same as the extractor construction in Section 5.5, and we reuse components and parameters that were setup in Section 5.5. We omit the proof of Theorem 7.1 since it follows along the lines of the proof of Theorem 5.8.

**Algorithm 5:** anmExt$(x)$

**Input:** Bit string $x$ of length $n$.
**Output:** Bit string $z$ of length $m$.

1 Let $v = \text{Slice}(x, n_1)$.
2 Let $r$ be a $D_1 \times n_2$ matrix, whose $i$'th row $r_i$, is $\text{LExt}_2(x, \text{LExt}_1(v, i))$.
3 Let $s_i$ be a $D_1 \times m$ matrix whose $i$'th row $s_i$, is $\text{ACB}(r_i, x, \text{advGen}(x, \text{Slice}(r_i, d_3)) \circ i)$.
4 Output $z = \oplus_{j=1}^{D_1} s_i$.

# 8 Efficient Sampling Algorithms

In this section we suitably modify the non-malleable extractor anmExt : $\{0,1\}^n \to \{0,1\}^m$ from Algorithm 5 and the advice generator advGen from Algorithm 3, and present an efficient algorithm that on any input $z \in \{0,1\}^m$ samples from a distribution that is uniform on the set $\text{anmExt}^{-1}(z)$. Since our other non-malleable extractors constructions are either reductions to anmExt or are very similar to the construction of anmExt, we do not explicitly present sampling algorithms for these constructions.

Most of the work in this section is in carefully executing the steps of the extractor such that each of these steps impose linear constraints on $x$ (fixing appropriate variables along the way), and further ensuring that we can argue about the rank of the composition of these linear maps. This makes the extractor construction much more tedious, but on the other hand, the sampling algorithm is easy to state and the correctness almost follows directly.

## 8.1 The modified extractor

We now describe the construction of the extractor ianmExt : $\{0,1\}^n \to \{0,1\}^m$ which takes as input $x \in \{0,1\}^n$ and outputs $z \in \{0,1\}^m$.

We use the following notation: for any linear map $L : \{0,1\}^a \to \{0,1\}^b$, given by $L(\alpha) = M\alpha$ for some $b \times a$ matrix $M$, let $con_L$ be a maximal set of linearly independent rows of $M$.

1. Let LCon : $\{0,1\}^n \to \{0,1\}^{n_1}$ be a linear condenser from Theorem 5.5 set to work for min-entropy $k_1 = n^\delta$ and parameter $\alpha$ (which we fix below). Thus $n_1 = wk_1^\alpha \log n$. Let
$$v = \text{LCon}(x).$$

2. Let $d_2$ be a parameter which we set below. Let $\text{LExt}_1 : \{0,1\}^{n_1} \times \{0,1\}^{d_1} \to \{0,1\}^{d_2}$ be a linear seeded extractor from Theorem 3.3 set to extract from min-entropy $k_2 = k_1^\alpha/2$ with error $\epsilon_1 = 1/20$. Thus $d_1 = O(\log n_1)$. Let $D_1 = 2^{d_1}$. Thus, $D_1 = n_1^{C_1}$ for some constant $C_1$. Let $r'$ be the $D_1 \times n_1$ matrix whose $i$'th row $r'_i$ is defined as
$$r'_i = \text{LExt}_1(v, i).$$

3. Let $\mathrm{LExt}_2 : \{0,1\}^n \times \{0,1\}^{d_2} \to \{0,1\}^{n_2}$, $n_2 = k^{1-2\delta}$, be a linear seeded extractor from Theorem 3.3 set to extract from min-entropy $k_3 = k/2$, and error $\epsilon_2 = 2^{-n^{\delta_1}}$, $\delta_1 = \alpha\delta/10$. Thus $d_2 = n^{2\delta_1}\log^2 n < \sqrt{k_2}$.

4. Set $\alpha = 1/(10C_1)$.

5. Let $n_3 = n_2/(2D_1)$ and $D' = 0.9D_1$. We now define a sequence of extractors $\{\mathrm{LExt}^v_{2,i}\}_{i=1}^{D'}$ in the following iterative way:

   (a) Let $i \leftarrow 1$, $\mathcal{L} \leftarrow con_{\mathrm{LCon}}$, $counter \leftarrow 0$, $rowlist \leftarrow \emptyset$.

   (b) while $i \le D_1$ and $counter < D'$:

      i. Define the function $f_i(x) = \mathrm{LExt}_2(x, \mathrm{LExt}_1(v,i))$. Note that $f_i : \{0,1\}^n \to \{0,1\}^{n_2}$ is a linear function. If $f_i(\{0,1\}^n) \neq \{0,1\}^{n_2}$, go to (v).

      ii. Update $counter \leftarrow counter + 1, rowlist \leftarrow rowlist \cup \{i\}$.

      iii. For any $S \subseteq [n_2]$, let $f_{i,S} : \{0,1\}^n \to \{0,1\}^{|S|}$ be the projection of $f_i$ to the coordinates in $S$. If $S = \{j\}$ is a singleton, we use the notation $f_{i,j}$ instead of $f_{i,\{j\}}$. Thus, $f_{i,j}(x) = \langle q_{i,j}, x\rangle$ for some $q_{i,j} \in \{0,1\}^n$. Pick a subset $S^v_i$ of $[n_2]$, $|S^v_i| = n_3$ such that $\mathcal{L} \cup \{q_{i,j} : j \in S^v_i\}$ is a set of linearly independent vectors. Define

$$\mathrm{LExt}^v_{2,i}(x,y) = \mathrm{LExt}_2(x,y)_{|S^v_i}.$$

      iv. Update $\mathcal{L} \leftarrow \mathcal{L} \cup \{q_{i,j} : j \in S^v_i\}$.

      v. $i \leftarrow 1 + 1$

6. If $|rowlist| < D'$, output

$$z = 0^m.$$

7. Let $r$ be the $D' \times n_3$ matrix whose $i$'th row $r_i$ is defined as

$$r_i = \mathrm{LExt}^v_{2,i}(x, \mathrm{LExt}_1(v,i)).$$

8. Let $d_3 = n^\delta$. For each $i \in [D']$, let $y_{1,i} = \mathrm{Slice}(r_i, 1, d_3)$.

9. Let $d_4 = d_3 + d_3(D')^2$. Let $y_2 = \mathrm{Slice}(r_1, d_3+1, d_4) \circ \ldots \circ \mathrm{Slice}(r_{D'}, d_3+1, d_4)$. Let $n_4 = (d_4 - d_3)D$.

10. Let $\mathcal{C}_{\mathrm{BCH}}$ be a BCH code with parameters: $[n_{\mathrm{BCH}}, n_{\mathrm{BCH}} - t_{\mathrm{BCH}}(\log n_{\mathrm{BCH}}), 2t_{\mathrm{BCH}}]_2$, $t_{\mathrm{BCH}} = \beta_{\mathrm{BCH}}\sqrt{n_{\mathrm{BCH}}}/2$, for some small enough constant $\beta_{bch}$, for some parameter $n_{\mathrm{BCH}}$. Let dBCH be the dual code. As is well known, thus dBCH is a $[n_{\mathrm{BCH}}, t_{\mathrm{BCH}}\log n_{\mathrm{BCH}}, \frac{n}{2} - t_{\mathrm{BCH}}\sqrt{n_{\mathrm{BCH}}}]_2$-code. Set $n_{\mathrm{BCH}}$ such that $t_{\mathrm{BCH}}(\log n_{\mathrm{BCH}}) = n$. Let enc be the encoder of dBCH.

11. Let $\mathrm{Samp} : \{0,1\}^{n_4} \to (\{0,1\}^{n_6})^{D_s}$, $n_6 = (\log n_{\mathrm{BCH}})$, be the sampler from Theorem 3.16 using a seeded extractor $\mathrm{Ext}_s$ from Theorem 3.2 for min-entropy $n_4/8$ and error $\epsilon_{\mathrm{Samp}} = 1/20$. Thus $D_s = n_4^{C_s}$ for some constant $C_s$.

12. $L_{1,2} : \{0,1\}^n \to \{0,1\}^{n_1 + d_4 D'}$ be a linear map such that $L_{1,2}(x) = v \circ y_{1,1} \circ y_{1,2} \ldots \circ y_{1,D'} \circ y_2$.

13. Let $adv_y(x) = \mathrm{enc}(x)_{|\mathrm{Samp}(y_2)}$. Note that $adv_y : \{0,1\}^n \to \{0,1\}^{D_s}$ is a linear function. For any subset of coordinates $S \subseteq D_s$, let $adv_S$ denote the linear map that is a projection of $adv_y$ to the coordinates in $S$.
    Pick a subset $S_y$ of size $D_s - (n_1 + d_4 D')$ such that the vectors in $con_{adv_{S_y}}$ and the vectors in $con_{L_{1,2}}$ are linearly independent.

14. We now construct a sequence of linear extractors in an iterative way similar to Step (5). Let $\text{LExt}_3 : \{0,1\}^n \times \{0,1\}^{d_3} \to \{0,1\}^{n_{adv}}$, $n_{adv} = n^{10\delta}$ be a linear seeded extractor from Theorem 3.3 set to extract from min-entropy $k/2$ with error $2^{-\Omega(\sqrt{d_3})}$. Let $D = 0.9D'$.

    (a) Let $i \leftarrow 1$, $\mathcal{L} \leftarrow con_{L_{1,2}} \cup con_{adv_{S_y}}$, $counter \leftarrow 0$, $rowlist \leftarrow \emptyset$.

    (b) while $i \leq D'$ and $counter < D$:

        i. Define the function $f_i(x) = \text{LExt}_3(x, y_{1,i})$. Note that $f_i : \{0,1\}^n \to \{0,1\}^{n_{adv}}$ is a linear function. If $f_i(\{0,1\}^n) \neq \{0,1\}^{n_{adv}}$, go to (v).

        ii. Update $counter \leftarrow counter + 1, rowlist \leftarrow rowlist \cup \{i\}$.

        iii. For any $S \subseteq [n_{adv}]$, let $f_{i,S} : \{0,1\}^n \to \{0,1\}^{|S|}$ be the projection of $f_i$ to the coordinates in $S$. Pick a subset $S_i$ of $[n_{adv}]$, $|S_i| = n_{adv}/2D$ such that $con_{f_{i,S_i}} \cup \mathcal{L}$ is a set of linearly independent vectors. Define

$$\text{LExt}_{3,i}(x, y) = \text{LExt}_3(x, y)_{|S_i}.$$

        iv. Update $\mathcal{L} \leftarrow \mathcal{L} \cup con_{f_{i,S_i}}$.

        v. $i \leftarrow 1 + 1$

15. If $counter < D$, output $0^m$.

16. For each $i \in [D]$, define $adv_i = i \circ y_{1,i} \circ y_2 \circ adv_{S_y}(x) \circ \text{LExt}_{3,i}(x, y_{1,i})$. Let $|adv_i| = h$.

17. $L_{adv} : \{0,1\}^n \to \{0,1\}^{n_1 + d_4 D + hD}$ be a linear map such that $L_{1,2}(x) = v \circ y_{1,1} \circ y_{1,2} \ldots \circ y_{1,D} \circ y_2 \circ adv_1 \circ \ldots \circ adv_D$.

18. For each $i \in [D]$, let $y_{-2,i} = slice(r_i, d_4 + 1, n_3)$. We iteratively pick subsets of coordinates and project these variables to these subsets in the following way:

    (a) Let $i \leftarrow 1$, $\mathcal{L} \leftarrow con_{L_{adv}}$:

    (b) while $i \leq D$:

        i. Define the function $f_i(x) = y_{-2,i}$. Note that $f_i : \{0,1\}^n \to \{0,1\}^{n_3 - d_4}$ is a linear function.

        ii. Update $counter \leftarrow counter + 1, rowlist \leftarrow rowlist \cup \{i\}$.

        iii. For any $S \subseteq [n_3 - d_4]$, let $f_{i,S} : \{0,1\}^n \to \{0,1\}^{|S|}$ be the projection of $f_i$ to the coordinates in $S$. Pick a subset $S_i$ of $[n_3 - d_4]$, $|S_i| = (n_3 - d_4)/2$ such that $con_{f_{i,S_i}} \cup \mathcal{L}$ is a set of linearly independent vectors.

        iv. Update $r_i \leftarrow y_1 \circ y_2 \circ (y_{-2,i})_{|S_i}$

        v. Update $\mathcal{L} \leftarrow \mathcal{L} \cup con_{f_{i,S_i}}$.

        vi. $i \leftarrow 1 + 1$

19. Let $n_3' = (n_3 + d_4)/2$.

20. Let $d_5 = d_4 + D^{20} d_4$. For each $i \in [D]$, let $y_{3,i} = \text{Slice}(r_i, d_4 + 1, d_5)$. Let $n_5 = d_5 - d_4$.

21. Let $d_6 = d_5 + D^{40} d_5$. Let $y_4 = \text{Slice}(r_1, d_5 + 1, d_6) \circ \ldots \circ \text{Slice}(r_D, d_5 + 1, d_6)$. Let $n_6 = D(d_6 - d_5)$.

22. Let $\text{ACB} : \{0,1\}^{n_5} \times \{0,1\}^{n_6} \times \{0,1\}^h \to \{0,1\}^{m_1}$ be the correlation breaker from Theorem 5.6 setup using the following components:

- $\mathrm{LExt}_{1,\mathrm{ACB}} : \{0,1\}^{n_6} \times \{0,1\}^{d_7} \to \{0,1\}^{d_7}$ and $\mathrm{LExt}_{2,\mathrm{ACB}} : \{0,1\}^{m_1} \times \{0,1\}^{d_7} \to \{0,1\}^{d_7}$ be $(k_{\mathrm{ACB}} = C_2 h D^3 \log^2(n/\epsilon_{ACB}), \epsilon_{\mathrm{ACB}} = 2^{-4d_4 D})$-strong linear seeded extractors, $m_1 = k_{\mathrm{ACB}} + C_2 D d_7 + 4d_4 D$ instantiated from Theorem 3.3, where $C_2$ is a large enough constant.
- $\mathrm{LExt}_{3,\mathrm{ACB}} : \{0,1\}^{n_5} \times \{0,1\}^{d_7} \to \{0,1\}^{m_1}$ be a $(k_{2,\mathrm{ACB}} = (n_2')^{1.1}, \epsilon_{\mathrm{ACB}})$-strong linear seeded extractor.

23. For each $i \in [D]$, let
$$w_i = \mathrm{ACB}(y_{3,i}, y_4, adv_i).$$
24. For each $i \in [D]$, let $y_{5,i} = \mathrm{Slice}(r_i, d_6 + 1, n_3')$. Let $n_7 = (n_3' - d_6)$.
25. Let $\mathrm{iExt} : \{0,1\}^{n_7} \times \{0,1\}^{m_1} \to \{0,1\}^m$, $m = \Omega(m_1)$, be the linear seeded extractor from Theorem 3.4 set to extract from min-entropy $0.9n_7$ with error $2^{-\Omega(m_1)}$.
26. Let
$$y_{6,i} = \mathrm{iExt}(y_{5,i}, w_i).$$
27. Output
$$z = \bigoplus_{i=1}^{D} y_{6,i}.$$

The following estimates hold by our choice of parameters:

- $D \le n^\delta$, $d_3 \le n^\delta < d_4 \le 2n^{3\delta} < d_5 < n^{25\delta} < d_6 \le n^{70\delta}$.
- $D_s \le n^{2C_s\delta}$, $h \le n^{10\delta}$.

We first prove that this modified construction is indeed a non-malleable extractor. In Section 8.2 we present an efficient sampling procedure that samples almost uniformly from the pre-image of any output of ianmExt.

**Theorem 8.1.** *There exists a constant $C$ such that for all $n, k, w > 0$ and any $\delta > 0$ with $w < n^\beta$, $\beta = \delta/(3C)$ and $k \ge n^{C\delta}$, there exists an efficient function* $\mathrm{anmExt} : \{0,1\}^n \to \{0,1\}^m$ *such that if $\mathbf{X}$ is a $w$-affine source with min-entropy at least $k$ and $\mathcal{A} : \{0,1\}^n \to \{0,1\}^n$ is an affine function with no fixed points, then*

$$|\mathrm{ianmExt}(\mathbf{X}), \mathrm{anmExt}(\mathcal{A}(\mathbf{X})) - \mathbf{U}_m, \mathrm{ianmExt}(\mathcal{A}(\mathbf{X}))| \le 2^{-n^{\Omega(1)}}$$

We begin with following claim.

**Claim 8.2.** $\Pr[|rowlist| < D'] < 2^{-n^{\Omega(1)}}$.

*Proof.* By Lemma 3.6 we have that $\Pr_{y \sim \mathbf{U}_{d_2}}[|\mathrm{LExt}_2(\mathbf{U}_n, y) - \mathbf{U}_{n_2}| > 0] < 2\epsilon_2$. Thus, there exists a set $Good \subset \{0,1\}^{d_2}$ such that $|Good| > (1 - 2\epsilon_2)D_2$ and for any $y \in Good$, $\mathrm{LExt}_2(\{0,1\}^n, y) = \{0,1\}^{n_2}$.

Let $T_v = \{\mathrm{LExt}_1(\mathbf{V}, i) : i \in \{0,1\}^{d_1}\} \subseteq \{0,1\}^{d_2}$. Using Theorem 3.16, it follows that

$$\Pr_{v \sim \mathbf{V}}[|T_v \cap Good| < (1 - \epsilon_1 - \epsilon_2)T_v] \le 2^{-k_2/2}.$$

The claim now follows noting that $\epsilon_1 + \epsilon_2 < 0.1$. □

From now on, we assume that $|rowlist| = D'$, and we add an error of $2^{-n^{\Omega(1)}}$ to the final error analysis of the extractor.

**Claim 8.3.** *Conditioned on $\mathbf{V}, \mathbf{V}'$, the r.v $\mathbf{R}$ is $2^{-n^{\Omega(1)}}$-close to an affine somewhere random source.*

*Proof.* Using Theorem 5.5 it follows that $\mathbf{V}$ is a $(n_1, k_1^{\alpha})$-source. Now by Lemma 3.6, there exists a set $I \subseteq [D_1]$, $|I| \geq 0.9D_1$ such that for any $i \in [I]$, $\text{LExt}_1(\mathbf{V}, i) = \mathbf{U}_{d_2}$. We note that $\mathbf{V}, \mathbf{V}'$ are obtained by applying linear functions on $\mathbf{X}$. Thus there exist disjoint subspaces $\mathbf{A}$ and $\mathbf{B}$ such that $\mathbf{X} = \mathbf{A} + \mathbf{B}$, $H_{\infty}(\mathbf{B}) \geq k - 2n_1$ and $\mathbf{B}$ is independent of $\mathbf{V}, \mathbf{V}'$. Now for any $i \in I$, we have

$$\text{LExt}_2(\mathbf{X}, \text{LExt}_1(\mathbf{V}, i)) = \text{LExt}_2(\mathbf{A}, \text{LExt}_1(\mathbf{V}, i)) + \text{LExt}_2(\mathbf{B}, \text{LExt}_1(\mathbf{V}, i)).$$

Since $\mathbf{B}$ is independent of $\mathbf{V}$, it follows that $\text{LExt}_2(\mathbf{B}, \text{LExt}_1(\mathbf{V}, i))$ is $\epsilon_2$-close to $\mathbf{U}_{n_2}$. Further, since $\text{LExt}_2$ is strong seeded, we fix $\text{LExt}_1(\mathbf{V}, i)$. Thus, $\text{LExt}_2(\mathbf{B}, \text{LExt}_1(\mathbf{V}, i))$ is now a deterministic function of $\mathbf{B}$. We also fix $\text{LExt}_2(\mathbf{A}, \text{LExt}_1(\mathbf{V}, i))$ since it is independent of $\mathbf{B}$. $\text{LExt}_2(\mathbf{B}, \text{LExt}_1(\mathbf{V}, i))$ remains $\epsilon_2$-close to $\mathbf{U}_{n_2}$ on average after these fixings. Further fix $\mathbf{V}, \mathbf{V}'$ noting that it does not affect the distribution of $\text{LExt}_2(\mathbf{X}, \text{LExt}_1(\mathbf{V}, i))$. Since $\mathbf{R}_i = \text{LExt}_2(\mathbf{B}, \text{LExt}_1(\mathbf{V}, i))_{|S_i^v}$, it follows that $\mathbf{R}_i$ is $\epsilon_2$-close to $\mathbf{U}_{n_3}$.

Finally, observe that after these fixings, the r.v $\text{LExt}_2(\mathbf{X}, \text{LExt}_1(\mathbf{V}, j))$, for any $j \in [D_1]$, is now a linear function of $\mathbf{X}$. Hence it follows that $\mathbf{R}$ is $\epsilon_2$-close to an affine somewhere random source. □

**Claim 8.4.** *In Step (13) of the algorithm for* ianmExt*, there indeed exists a set $S_y$ satisfying the required conditions.*

*Proof.* Let $G$ be the generator matrix for the linear code dBCH. Since BCH has distance at least $2t_{\text{BCH}}$, it follows that any set of $2t_{\text{BCH}} - 1$ rows of $G$ are linearly independent. Note that the size of $con_{L_{1,2}}$ is bounded by $n_1 + D'd_4$. By our choice of parameters, (i) $D_s << 2t_{\text{BCH}}$ and $n_1 + D'd_4 \leq D_s/n^{\delta}$. Thus it follows that there will always exist a choice for picking a set $S_y$ of size $D_s - n_1 - D'd_4$ satisfying the required linear independence condition. □

**Claim 8.5.** *With probability at least $1 - 2^{-n^{\Omega(1)}}$, the iterative process in Step (14) of the algorithm for* ianmExt *indeed succeeds in producing a sequence of $D$ extractors $\{\text{LExt}_{3,i}\}_{i=1}^{D}$.*

*Proof.* The proof is very similar to the proof of Claim 8.2 using the fact that $\text{LExt}_3$ is a strong linear seeded extractor (and Lemma 3.6). We omit the details. □

For the following claims, we continue to condition on $\mathbf{V}, \mathbf{V}'$. Note that Claim 8.3 in fact shows that $0.8D'$ rows of $\mathbf{R}$ is close to uniform. Thus, even after Step (14) of the algorithm computing ianmExt, where we eliminate $0.1D'$ rows, it follows that at least $0.7D$ of the remaining row of $\mathbf{R}$ are still close uniform. Thus, we assume that $\mathbf{R}$ is an affine somewhere random source (and we add an error of $2^{-n^{\Omega(1)}}$ to the final error of the extractor). Let $k_4 = k - 2n^{\delta}$. Note that $H_{\infty}(\mathbf{X}) \geq k_4$.

**Claim 8.6.** *Let $i \in [D]$ be an index such that $\mathbf{R}_i$ is uniform. Then, conditioned on $\{\mathbf{Y}_{1,j} : j \in [D]\}$, $\{\mathbf{Y}'_{1,j} : j \in [D]\}, \mathbf{Y}_2, \mathbf{Y}'_2\}$, we have*

$$\Pr[adv_i \neq adv'_i] \geq 1 - 2^{-n^{\Omega(1)}}.$$

*Proof.* The proof of this claim is similar to the proof of Theorem 5.7 but requires more work. Assume that $\mathbf{Y}_2 = \mathbf{Y}'_2$ and $\mathbf{Y}_{1,i} = \mathbf{Y}'_{1,i}$, since otherwise we directly have $adv_i \neq adv'_i$. Note that conditioned on $\{\mathbf{Y}_{1,j} : j \in [D]\}, \{\mathbf{Y}'_{1,j} : j \in [D]\}, \mathbf{Y}_2, \mathbf{Y}'_2\}$, the r.v. $\mathbf{Y}_2$ has min-entropy at least $n_4 - 2Dd_3 > n_4/2$. Thus, we can write $\mathbf{Y}_2 = \mathbf{Y}_{2,1} + \mathbf{Y}_{2,2}$ where $\mathbf{Y}_{2,1}$ and $\mathbf{Y}_{2,2}$ are independent affine sources, $\mathbf{Y}_{2,2}$ is independent of $\{\mathbf{Y}_{1,j} : j \in [D]\}, \{\mathbf{Y}'_{1,j} : j \in [D]\}, \mathbf{Y}_2, \mathbf{Y}'_2\}$ and has min-entropy at least $n_4/2$.

Let $\overline{\mathbf{Y}_{1,i}} = \mathbf{Y}_{1,i} \circ 0^{n-d_3}$, $\overline{\mathbf{Y}_{2,1}} = 0^{d_3} \circ \mathbf{Y}_{2,1} \circ 0^{n-n_4-d_3}$ and $\overline{\mathbf{Y}_{2,2}} = 0^{d_3} \circ \mathbf{Y}_{2,2} \circ 0^{n-n_4-d_3}$. It follows that there exist disjoint subspaces $\mathbf{A}_y$ and $\mathbf{B}_y$ such that $\mathbf{X} = \mathbf{A}_y + \mathbf{B}_y$, $\dim(\mathbf{A}) \leq n_4 + d_3$ (thus, $\dim(\mathbf{B}) = k_4 - \dim(\mathbf{A})$) and $T(\overline{\mathbf{Y}_{1,i}} + \overline{\mathbf{Y}_{2,1}} + \overline{\mathbf{Y}_{2,2}}) = \mathbf{A}_y$ for some linear function $T : \{0,1\}^n \to \{0,1\}^n$. Let $\overline{\mathbf{Y}_2} = \overline{\mathbf{Y}_{2,1}} + \overline{\mathbf{Y}_{2,2}}$.

We have,

$$\mathrm{enc}(\mathbf{X})_{|\mathrm{Samp}(\mathbf{Y}_2)} - \mathrm{enc}(\mathbf{X}')_{|\mathrm{Samp}(\mathbf{Y}_2)} = \mathrm{enc}(\mathbf{X} - \mathbf{X}')_{|\,\mathrm{Samp}(\mathbf{Y}_2)},$$

and

$$\mathrm{LExt}(\mathbf{X}, \mathbf{Y}_{1,i}) - \mathrm{LExt}(\mathbf{X}', \mathbf{Y}_{1,i}) = \mathrm{LExt}(\mathbf{X} - \mathbf{X}', \mathbf{Y}_{1,i}).$$

Further,

$$\begin{aligned}
\mathbf{X} - \mathbf{X}' &= \mathbf{A}_y + \mathbf{B}_y - \mathcal{A}(\mathbf{A}_y) - \mathcal{A}(\mathbf{B}_y) \\
&= T(\overline{\mathbf{Y}_2}) - \mathcal{A}(T(\overline{\mathbf{Y}_2})) + T(\overline{\mathbf{Y}_{1,i}}) - \mathcal{A}(T(\overline{\mathbf{Y}_{1,i}})) + \mathbf{B}_y - \mathcal{A}(\mathbf{B}_y) \\
&= T(\overline{\mathbf{Y}_{2,1}}) - \mathcal{A}(T(\overline{\mathbf{Y}_{2,1}})) + T(\overline{\mathbf{Y}_{2,2}}) - \mathcal{A}(T(\overline{\mathbf{Y}_{2,2}})) + T(\overline{\mathbf{Y}_{1,i}}) - \mathcal{A}(T(\overline{\mathbf{Y}_{1,i}})) + \mathbf{B}_y - \mathcal{A}(\mathbf{B}_y)
\end{aligned}$$

Now consider the following cases.

- $H_\infty(T(\overline{\mathbf{Y}_{2,2}}) - \mathcal{A}(T(\overline{\mathbf{Y}_{2,2}}))) \leq n_4/4$.

  In this case, we can fix $T(\overline{\mathbf{Y}_{2,2}}) - \mathcal{A}(T(\overline{\mathbf{Y}_{2,2}}))$, and it follows that $\mathbf{Y}_{2,2}$ has min-entropy at least $n_4/4$ after this fixing. Further fix $\mathbf{B}_y, \mathbf{Y}_{2,1}, \{\mathbf{Y}_{1,j} : j \in [D]\}, \{\mathbf{Y}'_{1,j} : j \in [D]\}, \mathbf{Y}_2, \mathbf{Y}'_2\}$ noting that it is independent of $\mathbf{Y}_{2,2}$. Observe that in fact $\mathbf{X} - \mathbf{X}'$ is now fixed. Since $\mathbf{X} \neq \mathbf{X}'$, it follows that $\mathrm{enc}(\mathbf{X})$ differs from $\mathrm{enc}(\mathbf{X}')$ in at least $1 - \frac{n_5}{n}$ fraction of the coordinates. Using Theorem 3.16, it follows that with probability at least $1 - 2^{-\Omega(n_1)}$, $\mathrm{Samp}(\mathbf{Y}_1)$ contains at least 2/5th fraction of coordinates where $\mathrm{enc}(\mathbf{X})$ differs from $\mathrm{enc}(\mathbf{X}')$. Now, by projecting onto $S_y$ (in Step (13) of the algorithm), by our choice of parameters, we discard $o(1)$ fraction of the bits (since we discard $(n_1 + D'd_4)$ bits, and $D_s > (n_1 + D'd_4)n^\delta$). Thus, at least of the surviving coordinates has a 1 bit, and hence $adv_i \neq adv'_i$.

- $H_\infty(T(\overline{\mathbf{Y}_{2,2}}) - \mathcal{A}(T(\overline{\mathbf{Y}_{2,2}}))) > n_4/4$.

  We have,

$$\begin{aligned}
\mathrm{LExt}_{3,i}(\mathbf{X} - \mathbf{X}', \mathbf{Y}_{1,i}) &= \mathrm{LExt}_{3,i}(T(\overline{\mathbf{Y}_2}) - \mathcal{A}(T(\overline{\mathbf{Y}_2})), \mathbf{Y}_{1,i}) \\
&\quad + \mathrm{LExt}_{3,i}(T(\overline{\mathbf{Y}_{1,i}}) - \mathcal{A}(T(\overline{\mathbf{Y}_{1,i}})), \mathbf{Y}_{1,i}) + \mathrm{LExt}_{3,i}(\mathbf{B} - \mathcal{A}(\mathbf{B}), \mathbf{Y}_{1,i}) \\
&= \mathrm{LExt}_{3,i}(T(\overline{\mathbf{Y}_{2,1}}) - \mathcal{A}(T(\overline{\mathbf{Y}_{2,1}})), \mathbf{Y}_{1,i}) + \mathrm{LExt}_{3,i}(T(\overline{\mathbf{Y}_{2,2}}) - \mathcal{A}(T(\overline{\mathbf{Y}_{2,2}})), \mathbf{Y}_{1,i}) \\
&\quad + \mathrm{LExt}_{3,i}(T(\overline{\mathbf{Y}_{1,i}}) - \mathcal{A}(T(\overline{\mathbf{Y}_{1,i}})), \mathbf{Y}_{1,i}) + \mathrm{LExt}_{3,i}(\mathbf{B} - \mathcal{A}(\mathbf{B}), \mathbf{Y}_{1,i})
\end{aligned}$$

  It follows that $\mathrm{LExt}_{3,i}(T(\overline{\mathbf{Y}_{2,2}}) - \mathcal{A}(T(\overline{\mathbf{Y}_{2,2}})), \mathbf{Y}_{1,i})$ is $2^{-n^{\Omega(1)}}$-close to uniform. We fix $\mathbf{Y}_{1,i}$ since $\mathrm{LExt}_{3,i}$ is a strong seeded extractor, and thus $\mathrm{LExt}_{3,i}(T(\overline{\mathbf{Y}_{2,2}}) - \mathcal{A}(T(\overline{\mathbf{Y}_{2,2}})), y_{1,i})$ is

now a deterministic function of $\mathbf{Y}_{2,2}$. Further fix $\mathbf{B}_y, \mathbf{Y}_{2,1}, \{\mathbf{Y}_{1,j} : j \in [D]\}, \{\mathbf{Y}'_{1,j} : j \in [D]\}$, $\mathbf{Y}_2, \mathbf{Y}'_2\}$ noting that it is independent of $\mathbf{Y}_{2,2}$. Thus, after these fixings $\mathrm{LExt}_{3,i}(\mathbf{X} - \mathbf{X}', \mathbf{Y}_{1,i}) = \mathrm{LExt}_{3,i}(T(\overline{\mathbf{Y}_{2,2}}) - \mathcal{A}(T(\overline{\mathbf{Y}_{2,2}})), y_{1,i}) + \alpha$ (for some constant $\alpha \in \{0,1\}^{n_5}$), and hence is $2^{-n^{\Omega(1)}}$-close to uniform on average. Thus, $\Pr[\mathrm{LExt}_{3,i}(\mathbf{X} - \mathbf{X}', y_{1,i}) = 0] \leq 2^{-n^{\Omega(1)}} + 2^{-n_5}$.

□

Next we observe that the iterative pruning process of the rows of $\mathbf{R}$ in Step (17) is indeed valid since we have ensured that each row is much longer than the size of the set $con_{L_{adv}}$. We conclude with following claim.

**Claim 8.7.** $|(\mathbf{Z}, \mathbf{Z}') - (\mathbf{U}_m, \mathbf{Z}')| = 2^{-n^{\Omega(1)}}$.

*Proof.* We condition on the random variables $\mathbf{V}, \mathbf{V}', \{\mathbf{Y}_{1,j} : j \in [D]\}, \{\mathbf{Y}'_{1,j} : j \in [D]\}, \mathbf{Y}_2, \mathbf{Y}'_2$, $\{adv_i : i \in [D]\}, \{adv'_i : i \in [D]\}$. Further, we assume that $adv_i \neq adv'_i$ for some index $i \in [D]$. Note that the corresponding row $\mathbf{R}_i$ has min-entropy at least $n_3 - 2Dd_3 - 2n_4 - 2Dh > n_3 - n^{5\delta}$.

The claim now follows almost directly from our choice of parameters and Theorem 5.6 (and using a conditioning argument similar to ones used many times previously in this paper). We omit the details. □

## 8.2 The Sampling Algorithm

We use the variable names in algorithm for ianmExt in Section 8.1 to describe our sampling algorithm. We use the following simple algorithm that takes as input $z \in \{0,1\}^m$:

1. Sample $v$ from $\mathbf{U}_{n_1}$. For each $i \in [D]$, sample $y_{1,i}$ from $\mathbf{U}_{d_3}$. Sample $y_2$ from $U_{n_4}$. For each $i \in [D]$, generate the remaining bits of $adv_i$ uniformly.
2. For each $i \in [D]$, sample $y_{3,i}$ from $\mathbf{U}_{n_5}$. Sample $y_4$ from $U_{n_6}$.
3. Using the variables samples, and the algorithm for ianmExt in Section 8.1, for each $i \in [D]$ compute the variable $w_i = \mathrm{ACB}(y_{3,1}, y_4, adv_i)$.
4. For each $i \in [D-1]$, generate independent uniform strings $y_{6,i} \in \{0,1\}^m$. Set $y_{6,D} = (\bigoplus_{i=1}^{D-1} y_{6,i}) \oplus z$.
5. For each $i \in [D]$, uniformly sample the variable $y_{5,i}$ from the space of all solutions of the equation $\mathrm{iExt}(., w_i) = y_{6,i}$.
6. Note that all the random variables sampled so far are linear functions in $x$, and hence places linear constraints on $x$. Sample $x$ uniformly from the largest subspace in $\{0,1\}^n$ that satisfies all these linear constraints.

We observe that all the steps of the above algorithm can be executed in time $\mathrm{poly}(n)$ with access to random bits. This follows from the fact that in each step, we are either sampling uniform bits, or sampling uniformly from a subspace with access to a basis.

We now argue that the above algorithm indeed samples from a distribution that is $2^{-n^{\Omega(1)}}$-close to uniform on $\mathrm{ianmExt}^{-1}(z)$. The correctness of our sampling algorithm is indeed direct from the following claim.

**Claim 8.8.** *With probability* $1 - 2^{-n^{\Omega(1)}}$ *over the fixing of the variables* $v, \{y_{1,i} : i \in [D]\}, y_2, \{y_{3,i} : i \in [D]\}, \{adv_i : i \in D\}, y_4, \{y_{6,i} : i \in [D]\}$, *and* $\{y_{5,i} : i \in D\}$ *by the above sampling algorithm, the dimension of the subspace to which* $x$ *is restricted is the same.*

*Proof.* The probability that for a random sample of $v$ that $|rowlist| < D'$ (in Step (6) of the algorithm for ianmExt) is bounded by $2^{-n^{\Omega(1)}}$ (by Claim 8.2). Thus suppose that the sampled v is such that $|rowlist| = D'$ (and we incur an error of $2^{-n^{\Omega(1)}}$). Similarly, The probability that for a random sample of $\{y_{1,i} : i \in [D]\}$, the probability that $counter < D$ is bounded by $2^{-n^{\Omega(1)}}$ (by Claim 8.5) The main observation is that in the algorithm for ianmExt we ensure that fixing such 'good' fixing of $v, \{y_{1,i} : i \in [D]\}$ and the other variables, the constraints imposed by each such variable on $x$ are linearly independent. Indeed this can be directly verified from the algorithm. Thus for any consistent fixing of the random variables, the number of linearly independent constraints imposed on $x$ are the same, and hence the claim follows. □

## References


[ABN+92] Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ron M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38:509–516, 1992.

[ADKO15] D. Aggarwal, Y. Dodis, T. Kazana, and M. Obremski. Non-malleable reductions and applications. To appear in STOC, 2015.

[ADL14] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. In *STOC*, 2014.

[AGM+15] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. A rate-optimizing compiler for non-malleable codes against bit-wise tampering and permutations. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 375–397, 2015.

[BDKM16] Marshall Ball, Dana Dachman-Soled, Mukul Kulkarni, and Tal Malkin. Non-malleable codes for bounded depth, bounded fan-in circuits. In *TCC*, 2016.

[BIW06] Boaz Barak, Russell Impagliazzo, and Avi Wigderson. Extracting randomness using few independent sources. *SIAM J. Comput.*, 36(4):1095–1118, December 2006.

[Bou07] Jean Bourgain. On the construction of affine extractors. *GAFA Geometric And Functional Analysis*, 17(1):33–57, 2007.

[CDF+08] Ronald Cramer, Yevgeniy Dodis, Serge Fehr, Carles Padró, and Daniel Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In *EUROCRYPT*, pages 471–488, 2008.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[CG14a] Mahdi Cheraghchi and Venkatesan Guruswami. Capacity of non-malleable codes. In *ITCS*, pages 155–168, 2014.

[CG14b] Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. In *TCC*, pages 440–464, 2014.

[CGL16] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In *STOC*, 2016.

[CL16a] Eshan Chattopadhyay and Xin Li. Explicit non-malleable extractors, multi-source extractors and almost optimal privacy amplification protocols. *Electronic Colloquium on Computational Complexity (ECCC)*, 2016.

[CL16b] Eshan Chattopadhyay and Xin Li. Extractors for sumset sources. In *STOC*, 2016.

[Coh15] Gil Cohen. Local correlation breakers and applications to three-source extractors and mergers. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, 2015.

[Coh16a] Gil Cohen. Making the most of advice: New correlation breakers and their applications. *Electronic Colloquium on Computational Complexity (ECCC)*, 2016.

[Coh16b] Gil Cohen. Non-malleable extractors - new tools and improved constructions. In *CCC*, 2016.

[CZ14] Eshan Chattopadhyay and David Zuckerman. Non-malleable codes against constant split-state tampering. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science*, pages 306–315, 2014.

[CZ16] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. In *STOC*, 2016.

[DKO13] Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Non-malleable codes from two-source extractors. In *CRYPTO (2)*, pages 239–257, 2013.

[DORS08] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38:97–139, 2008.

[DP07] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 227–237, Washington, DC, USA, 2007. IEEE Computer Society.

[DPW10] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In *ICS*, pages 434–452, 2010.

[DW09] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *STOC*, pages 601–610, 2009.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. *J. ACM*, 56(4), 2009.

[Has87] Johan Hastad. *Computational Limitations of Small-depth Circuits*. MIT Press, Cambridge, MA, USA, 1987.

[Li13] Xin Li. Extractors for a constant number of independent sources with polylogarithmic min-entropy. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 100–109, 2013.

[Li15a] Xin Li. Improved two-source extractors, and affine extractors for polylogarithmic entropy. Technical Report TR15-125, ECCC, 2015.

[Li15b] Xin Li. Three-source extractors for polylogarithmic min-entropy. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, 2015.

[Li16] Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. *arXiv preprint arXiv:1608.00127*, 2016.

[MW97] Ueli Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In *Advances in Cryptology — CRYPTO '97*, volume 1294, pages 307–321, August 1997.

[Rao09] Anup Rao. Extractors for low-weight affine sources. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity*, 2009.

[RRV02] Ran Raz, Omer Reingold, and Salil Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. *JCSS*, 65(1):97–128, 2002.

[Tre01] Luca Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, pages 860–879, 2001.

[Vio11] Emanuele Viola. Extractors for circuit sources. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 220–229, 2011.

[Vio14] Emanuele Viola. Extractors for circuit sources. *SIAM J. Comput.*, 43(2):655–672, 2014.

[Zuc90] D. Zuckerman. General weak random sources. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 0:534–543 vol.2, 1990.

[Zuc97] David Zuckerman. Randomness-optimal oblivious sampling. *Random Structures and Algorithms*, 11:345–367, 1997.

[Zuc07] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, pages 103–128, 2007.



ECCC ISSN 1433-8092

http://eccc.hpi-web.de