Non-Malleable Extractors and Codes, with their Many Tampered Extensions

Eshan Chattopadhyay* Vipul Goyal † Xin Li‡

March 21, 2016

Abstract

Randomness extractors and error correcting codes are fundamental objects in computer science. Recently, there have been several natural generalizations of these objects, in the context and study of tamper resilient cryptography. These are *seeded non-malleable extractors*, introduced by Dodis and Wichs [DW09]; *seedless non-malleable extractors*, introduced by Cheraghchi and Guruswami [CG14b]; and *non-malleable codes*, introduced by Dziembowski, Pietrzak and Wichs [DPW10]. Besides being interesting on their own, they also have important applications in cryptography, e.g, privacy amplification with an active adversary, explicit non-malleable codes etc, and often have unexpected connections to their non-tampered analogues [Li12b] [CZ15].

However, the known constructions are far behind their non-tampered counterparts. Indeed, the best known seeded non-malleable extractor requires min-entropy rate at least 0.49 [Li12b]; while explicit constructions of non-malleable two-source extractors were not known even if both sources have full min-entropy, and was left as an open problem in [CG14b].

In this paper we make progress towards solving the above problems and other related generalizations. Our contributions are as follows.

- We construct an explicit seeded non-malleable extractor for min-entropy $k \geq \log^2 n$. This dramatically improves all previous results and gives a simpler 2-round privacy amplification protocol with optimal entropy loss, matching the best known result in [Li15a]. In fact, we construct more general seeded non-malleable extractors (that can handle multiple adversaries) which were used in the recent construction of explicit two-source extractors for polylogarithmic min-entropy [CZ15].
- We construct the first explicit non-malleable two-source extractor for min-entropy $k \geq n - n^{\Omega(1)}$, with output size $n^{\Omega(1)}$ and error $2^{-n^{\Omega(1)}}$, thus resolving the open question in [CG14b].
- We motivate and initiate the study of two natural generalizations of seedless non-malleable extractors and non-malleable codes, where the sources or the codeword may be tampered many times. For this, we construct the first explicit non-malleable two-source extractor with tampering degree t up to $n^{\Omega(1)}$. By using the connection in [CG14b] and providing efficient sampling algorithms, we obtain the first explicit non-malleable codes with tampering degree t up to $n^{\Omega(1)}$, relative rate $n^{\Omega(1)}/n$, and error $2^{-n^{\Omega(1)}}$. We call these stronger notions *one-many* and *many-many* non-malleable codes. This provides a stronger information theoretic analogue of a primitive known as continuous non-malleable codes.

Our basic technique used in all of our constructions can be seen as inspired, in part, by the techniques previously used to construct *cryptographic non-malleable commitments*.

*Department of Computer Science, University of Texas at Austin. Research supported in part by NSF Grant CCF-1218723. Research done in part while visiting MSR India and John Hopkins University. eshanc@cs.utexas.edu.

†Microsoft Research, India. vipul@microsoft.com

‡Department of Computer Science, John Hopkins University. lixints@cs.jhu.edu

1 Introduction

Randomness extractors are fundamental objects in the study of randomness in computation. They are efficient algorithms that transform imperfect randomness into almost uniform random bits. Here we use the standard model of *weak random source* to model imperfect randomness. The *min-entropy* of a random variable X is defined as $H_\infty(X) = \min_{x \in \mathsf{Supp}(X)} \log_2(1/\Pr[X = x])$. For a source X supported on $\{0,1\}^n$, we call X an $(n, H_\infty(X))$-source, and we say X has *entropy rate* $H_\infty(X)/n$.

As one can show that no deterministic extractor works for all weak random sources even with min-entropy $k = n - 1$, randomness extractors are studied in two different settings. In one setting the extractor is given a short independent uniform random seed, and these extractors are called *seeded extractors*. Informally, a seeded extractor $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ for min-entropy k and error ϵ takes as input any (n,k) source X and a uniform seed S, and has the property that $|\text{Ext}(X,S) - U_m| < \epsilon$, where the distance used is the standard statistical distance. If the output of the extractor is guaranteed to be close to uniform even after seeing the value of the seed S, then it is called a strong seeded extractor. In the other setting there is no such random seed, but the source is assumed to have some special structure. These extractors are called *seedless extractors* (see Section 3 for formal definitions). A special kind of seedless extractors that received a lot of attention is extractors for independent weak random sources. Here one can use the probabilistic method to show that such extractors exist for only two independent sources (such extractors are called *two-source extractors*).

Both kinds of extractors have been studied extensively, and shown to have many connections and applications in computer science. For example, seeded extractors can be used to simulate randomized algorithms with access to only weak random sources, and are closely related to pseudorandom generators, error-correcting code and expanders. Independent source extractors can be used to generate high quality random bits for distributed computing and cryptography [KLRZ08], [KLR09], and are closely related to Ramsey graphs and other seedless extractors.

In cryptographic applications, however, one faces a new situation where the inputs of an extractor may be tampered by an adversary. For example, an adversary may tamper with the seed of a seeded extractor, or both sources of a two-source extractor. In this case, one natural question is how the output of the tampered inputs will depend on the output of the initial inputs. In order to be resilient to adversarial tampering, one natural way is to require that original output of the extractor be (almost) independent of the tampered output. This leads to the notion of *non-malleable extractors*, in both the seeded case and seedless case. These extractors not only are interesting in their own rights, but also have important applications in cryptography.

Definition 1.1 (Tampering Funtion)**.** *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 $s \in S$.*

Seeded non-malleable extractors were introduced by Dodis and Wichs in [DW09], as a generalization of strong seeded extractors.

Definition 1.2 (Non-malleable extractor)**.** *A function* $\text{snmExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a seeded non-malleable extractor for min-entropy k and error ϵ if the following holds : If X is a source on $\{0,1\}^n$ with min-entropy k and $\mathcal{A} : \{0,1\}^n \to \{0,1\}^n$ is an arbitrary tampering function with no fixed points, then*

$$|\text{snmExt}(X, U_d) \circ \text{snmExt}(X, \mathcal{A}(U_d)) \circ U_d - U_m \circ \text{snmExt}(X, \mathcal{A}(U_d)) \circ U_d| < \epsilon$$

where U_m is independent of U_d and X.

The original motivation for seeded non-malleable extractors is to study the problem of privacy amplification with an active adversary. This is a basic problem in information theoretic cryptography, where two

parties want to communicate with each other to convert their shared secret weak random source X into shared secret nearly uniform random bits. However, the communication channel is watched by an adversary Eve, where we assume Eve has unlimited computational power and the two parties have local (non-shared) uniform random bits.

In the case where Eve is passive (i.e., can only see the messages but cannot change them), this problem can be solved by just applying a strong seeded extractor. However, in the case where Eve is active (i.e., can arbitrarily change, delete and reorder messages), the problem becomes much more complicated. The major goal here is to design a protocol that uses as few number of interactions as possible, and output a uniform random string R that has length as close to $H_\infty(X)$ as possible (the difference is called *entropy loss*). There has been extensive research on this problem (we give more details in Section 1.4). Along the line, a major progress was made by Dodis and Wichs [DW09], who showed that seeded non-malleable extractors can be used to construct privacy amplification protocols with optimal round complexity and entropy loss.

This connection makes constructing non-malleable extractors a very promising approach to privacy amplification. However, all known constructions of such extractors ([DLWZ14], [CRS14], [Li12a], [DY13], [Li12b]) require the entropy of the weak source to be at least $0.49n$. Moreover, all known constructions are essentially based on known two-source extractors, and the entropy requirement is exactly the same as the best known two-source extractor [Bou05].

In this work, we revive the original approach of constructing non-malleable extractors using alternating extraction [DW09]. We dramatically improve all previous results and give explicit seeded non-malleable extractors that work for any min-entropy $k \geq \log^2 n$. We in fact consider a generalization, called as t-non-malleable extractors (introduced by Cohen et al. [CRS14]), where there are t tampering functions acting on the seed. We give explicit constructions of t-non-malleable extractors for min-entropy $k = \mathrm{poly}(t, \log n)$.

We now discuss the seedless variant of non-malleable extractors. Cheraghchi and Guruswami [CG14b] introduced seedless non-malleable extractors as a natural generalization of seeded non-malleable extractors. Furthermore, they found an elegant connection between seedless non-malleable extractors and non-malleable codes, which are a generalization of error-correcting codes to handle a much larger class of tampering functions (rather than just bit erasure or modification). Informally, non-malleable codes are w.r.t a family of tampering functions $\mathcal{F}$, and require that the decoding of any codeword that is tampered by a function $f \in \mathcal{F}$, is either the original message itself or something totally independent of the message (see Section 1.1). Non-malleable codes have also been extensively studied recently (we provide more details in Section 1.1), and Cheraghchi and Guruswami [CG14b] showed a universal way of constructing explicit non-malleable codes by first constructing non-malleable seedless extractors.

In this paper we focus on one of the most interesting and well studied family of tampering functions, where the function tampers the original message independently in two separate parts. This is called the 2-split-state model (see Section 1.1 for a formal discussion). The corresponding seedless non-malleable extractor is then a generalization of two-source extractors, where both sources can be tampered. For ease of presentation,we present a simplified definition here and we refer the reader to Section 4 for the formal definition.

Definition 1.3 (Seedless 2-Non-Malleable Extractor)**.** *A function* $\mathrm{nmExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *is a seedless 2-non-malleable extractor at min-entropy* k *and error* ϵ *if it satisfies the following property: If* X *and* Y *are independent* (n,k)*-sources and* $\mathcal{A} = (f,g)$ *is an arbitrary* 2*-split-state tampering function, such that at least one of* f *and* g *has no fixed points, then*

$$|\mathrm{nmExt}(X,Y) \circ \mathrm{nmExt}(\mathcal{A}(X,Y)) - U_m \circ \mathrm{nmExt}(f(X), g(Y))| < \epsilon$$

where both U_m*'s refer to the same uniform* m*-bit string.*

Again, the connection in [CG14b] makes constructing seedless 2-non-malleable extractors a very interesting and promising approach to non-malleable codes in the 2-split-state model. However, no explicit constructions of 2-non-malleable extractors were known even when both sources are perfectly uniform. Indeed, finding an explicit construction of such extractors was left as an open problem in [CG14b], and none of the known constructions of seeded non-malleable extractors seem to satisfy this stronger notion. In this paper we solve this open problem and give the first explicit construction of 2-non-malleable extractors. Furthermore we show that given any output of the extractor, we can efficiently sample uniformly from its pre-image. By the connection in [CG14b] this also gives explicit non-malleable codes in the above mentioned well studied 2-split-state model.

We note that our results about non-malleable codes in the 2-split-state model do not improve the already nearly optimal construction in the recent work of Aggarwal et al. [ADKO15]. However, our construction of seedless 2-non-malleable extractors is of independent interest, and provides a more direct way to construct non-malleable codes.[1]

Finally, as in the case of seeded non-malleable extractors [CRS14], we consider the situation where the sources can be tampered many times. For this, we introduce a natural generalization of seedless 2-non-malleable extractors which we call seedless $(2,t)$-non-malleable extractors (i.e., the sources are tampered t times). Correspondingly, in the case of non-malleable codes we also consider the situation where a codeword can be tampered many times. For this, we also introduce a natural generalization of non-malleable codes which we call *one-many non-malleable codes* (see Section 1.1). We initiate the study of these two objects in this paper and show that one-many non-malleable codes have several natural and interesting applications in cryptography.

We present a simplified definition of seedless $(2,t)$-non-malleable extractors here, and refer the reader to Section 4 for the formal definition.

Definition 1.4 (Seedless (2,t)-Non-Malleable Extractor)**.** *A function* $\text{nmExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *is a seedless* $(2,t)$*-non-malleable extractor at min-entropy* k *and error* ϵ *if it satisfies the following property: If* X *and* Y *are independent* (n,k)*-sources and* $\mathcal{A}_1 = (f_1, g_1), \ldots, \mathcal{A}_t = (f_t, g_t)$ *are* t *arbitrary* 2*-split-state tampering functions, such that for each* $i \in \{1, \ldots, t\}$ *at least one of* f_i *and* g_i *has no fixed points, then*

$$|\text{nmExt}(X,Y), \text{nmExt}(\mathcal{A}_1(X,Y)), \ldots, \text{nmExt}(\mathcal{A}_t(X,Y)) - U_m, \text{nmExt}(\mathcal{A}_1(X,Y)), \ldots, \text{nmExt}(\mathcal{A}_t(X,Y))| < \epsilon,$$

where both U_m*'s refer to the same uniform* m*-bit string.*

We provide explicit constructions of seedless $(2,t)$-non-malleable extractors for t up to n^δ for a small enough constant δ. Just as the connection between 2-non-malleable extractors and regular non-malleable codes, we show that these extractors lead to explicit constructions of one-many non-malleable codes in the 2-split-state model. We note that as in the case of regular non-malleable codes, the construction based on $(2, t)$-non-malleable extractors may not be the only way to construct one-many non-malleable codes. However, it appears non-trivial to extend other existing constructions of non-malleable codes to satisfy this stronger notion. We discuss this in more details in Section 1.3.

Subsequent Work In a recent work, Chattopadhyay and Zuckerman [CZ15] used our construction of seeded t-non-malleable extractors as a key component for explicitly constructing two-source extractors for polylogarithmic min-entropy. It was crucial to their construction that the min-entropy requirement and the seed-length of our t-non-malleable extractor are both $\text{poly}(t, \log n)$.

[1] In [ADKO15], non-malleable codes in the 2-split-state model are constructed by giving efficient reductions from the 2-split-state model to t-split-state model, and then using a known constructions of NM codes in the t-split-state model with almost optimal parameters [CZ14].

1.1 Non-malleable codes

We introduce the notion of what we call one-many and many-many non-malleable codes, generalizing the notion of non-malleable codes introduced by Dziembowski, Pietrzak and Wichs [DPW10]. Since the introduction of non-malleable codes, there has been a flurry of recent work on finding explicit constructions, resulting in applications to tamper-resilient cryptography [DPW10], robust versions of secret sharing schemes [ADL14], and connections to the seemingly unrelated area of derandomization [CG14b]. We discuss prior work in detail in Section 1.5.

We briefly motivate the notion of non-malleable codes. Traditional 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. One might hope to achieve a weaker goal of only detecting errors, possibly with high probability. However the notion of error detection fails to work with respect to the family of constant functions since one cannot hope to detect errors against a function that always outputs some fixed codeword.

The notion of non-malleable codes is an elegant 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 $\text{Dec}(\text{Enc}(m)) = m$ and for any tampering function $f \in \mathcal{F}$ the following holds: for any message m, $\text{Dec}(f(\text{Enc}(m)))$ is either the message m or is ϵ-close (in statistical distance) to a distribution D_f independent of m. The parameter ϵ is called the error. Thus, in some sense, either the message arrives correctly, or, the message is entirely lost and becomes gibberish. A formal definition of non-malleable codes is given below.

First we define the replace function $\text{replace} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$. If the second input to replace is a single value s, replace all occurrences of $same^\star$ in the first input with s and output the result. If the second input to replace is a set $(s_1, \ldots, s_n)$, replace all occurrences of $same^\star{}_i$ in the first input with s_i for all i and output the result.

Definition 1.5 (Coding schemes)**.** *Let* $\text{Enc} : \{0,1\}^k \to \{0,1\}^n$ *and* $\text{Dec} : \{0,1\}^n \to \{0,1\}^k \cup \{\perp\}$ *be functions such that* Enc *is a randomized function (i.e. it has access to a private randomness) and* Dec *is a deterministic function. We say that* (Enc, Dec) *is a coding scheme with block length* n *and message length* k *if for all* $s \in \{0,1\}^k$, $\Pr[\text{Dec}(Enc(s)) = s] = 1$ *(the probability is over the randomness in* Enc*).*

Definition 1.6 (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* ϵ *if for every* $f \in \mathcal{F}$ *there exists a random variable* D_f *on* $\{0,1\}^k \cup \{same^\star\}$ *which is independent of the randomness in* Enc *such that for all messages* $s \in \{0,1\}^k$*, it holds that*

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

The rate of a non-malleable code $\mathcal{C}$ is given by $\frac{k}{n}$. Observe that to construct non-malleable codes, it is still necessary to restrict the class of tampering functions. This follows since the tampering function could then use the function Dec to decode the message m, get a message m' by flipping all the bits in m, and use the encoding function to pick any codeword in $\text{Enc}(m')$. However presumably, the class of tampering functions can now be much richer than what was possible for error correction and error detection.

Tampering Multiple Codewords. Observe that the above definition envisions the adversary receiving a single codeword $\text{Enc}(s)$ and outputting a single tampered codeword $f(\text{Enc}(s))$. We refer to this as the

"one-one" setting. While indeed this is very basic, we argue that this does not capture scenarios where the adversary may be getting multiple codewords as input or be allowed to output multiple codewords. As an example, consider the following.

Say there is an auction where each party can submit its bid, and, the item goes to the highest bidder. An honest party, wishing to bid for value s, encodes its bid using NM codes and sends $\text{Enc}(s)$. This indeed would prevent an adversary (which belongs to an appropriate class of tampering functions) from constructing his own bid by tampering $\text{Enc}(s)$ and coming up with $\text{Enc}(s+1)$, which would completely compromise the sanity of the auction process. However what if the adversary can submit *two* bids out of which exactly one is guaranteed to be a winning bid? For example, the adversary can submit bids to r and $2s - r$ (for some r not known to the adversary). This is not ruled out by NM codes!

Towards that end, we introduce a stronger notion which we call *one-many NM codes*. Intuitively, this guarantees the following. Consider the set of codewords output by the adversary. We require that even the joint distribution of the encoded value be independent of the value encoded in the input. A formal definition is given below:

Definition 1.7 (One-Many 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* $\mathcal{F} \subset (\mathcal{F}_n)^t$ *and error* ϵ *if for every* $(f_1, \dots f_t) \in \mathcal{F}$*, there exists a random variable* $D_{\vec{f}}$ *on* $(\{0,1\}^k \cup \{same^\star\})^t$ *which is independent of the randomness in* Enc *such that for all messages* $s \in \{0,1\}^k$*, it holds that*

$$|(\text{Dec}(f_1(X)), \dots, \text{Dec}(f_t(X))) - \text{replace}(D_{\vec{f}}, s)| \leq \epsilon$$

Where $X = \text{Enc}(s)$*. We refer to* t *as the tampering degree of the non-malleable code.*

Thus one-many non-malleable codes is a natural more robust version of the well studied notion of non-malleable codes, and can be used in all applications of non-malleable codes in tamper-resilient cryptography with this stronger form of security.

An expert in cryptography by now would have noticed this is analogous to the well studied notion of one-many non-malleable *commitments* [PR08a]. Even though both notions deal with related concerns, we note non-malleable codes and non-malleable commitment are fundamentally different objects with the latter necessarily based on complexity assumptions. To start with, we prove a simple impossibility result for one-many non-malleable codes (whereas for one-many non-malleable commitments, a corresponding *positive* result is known [PR08a]).

Lemma 1.8. *One-many non-malleable codes which work for any arbitrary tampering degree and* $\epsilon < 1/4$ *cannot exist for a large class of tampering functions.*

Proof. The class of tampering functions which we consider are the ones where each function is allowed to read any one bit X_i of its choice from the input code X, and output a fresh encoding of X_i. Most natural tampering functions (including split state ones [DPW10] [CG14a]) considered in the literature fall into this class. Assume that the encoded value s has at least 4 possibilities (length 2 bits or higher). The case of a single bit s is discussed later.

Recall that n is the length of the code. We set $t = n$. Let $X = \text{Enc}(s)$ be the input codeword where s is chosen at random. We consider n tampering functions where F_i simply reads X_i and outputs a fresh encoding $W_i = \text{Enc}(X_i)$. Now consider $(\text{Dec}(f_1(X)), \dots, \text{Dec}(f_n(X)))$. Observe that this is exactly the bits of the string X. If the distinguisher applies the decode procedure on X, it will recover s. Now consider any possible output $(d_1, \dots, d_n)$ of $D_{\vec{f}}$. Now note that there cannot exist d_i which is $same^\star$. This is because otherwise it will be replaced by s (see Definition 1.7) which is at least 2 bits while $\text{Dec}(W_i)$ is just a single

bit. This in turn implies that the value replace$(D_{\vec{f}}, s)$ (from Definition 1.7) is independent of s and X. Thus a distinguisher (given access to s) can easily have an advantage exceeding ϵ.

For a single bit s, we modify our tampering functions to encode two bits: $W_i = \text{Enc}(X_i||0)$. Then again we can argue that neither of d_i will be $same^\star$ since then it will be replaced by s which is only one bit. This in turn again implies that replace$(D_{\vec{f}}, s)$ is independent of s and X. This concludes the proof.

□

We also introduce a natural generalization which we call *many-many non-malleable codes*. This refers to the situation where the adversary is given multiple codewords as input.

Definition 1.9 (Many-Many 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* $\mathcal{F} \subset (\mathcal{F}_n)^t$ *and error* ϵ *if for every* $(f_1, \ldots f_t) \in \mathcal{F}$*, there exists a random variable* $D_{\vec{f}}$ *on* $(\{0,1\}^k \cup \{same^\star{}_i\}_{i\in[u]})^t$ *which is independent of the randomness in* Enc *such that for all vector of messages* $(s_1, \ldots, s_u)$*,* $s_i \in \{0,1\}^k$*, it holds that*

$$|(\text{Dec}(f_1(\vec{X})), \ldots, \text{Dec}(f_t(\vec{X}))) - \text{replace}(D_{\vec{f}}, (s_1, \ldots, s_u))| \leq \epsilon$$

Where $X_i = \text{Enc}(s_i)$ *and* $\vec{X} = (X_1, \ldots, X_u)$

The following lemma relates one-many non-malleable codes to many-many non-malleable codes. This lemma is analogous to a similar lemma for non-malleable commitments [PR08a].

Lemma 1.10. *One-many non-malleable codes with tampering degree* t *and error* ϵ *are also many-many non-malleable codes for tampering degree* t *and error* $u\epsilon$ *(where* u *is as in Definition 1.9).*

Proof. This proof relies on a simple hybrid argument and the fact that all sources $X_1, \ldots, X_u$ are independent. We only provide a proof sketch here. Assume towards contradiction that there exists a one-many code with error ϵ, which, under the many-many tampering adversary has error higher than $u.\epsilon$. That is, the adversary $\vec{(f)}$ is given as input $(X_1, \ldots, X_u)$ which are encodings of $(s_1, \ldots, s_u)$ respectively. This is referred to as the hybrid 0. Now consider the following hybrid experiment. In the i-th hybrid experiment, the code X_i is changed to be an encoding of 0 (as opposed to be an encoding of s_i). We claim that in this experiment, the error changes by at most ϵ. This is because otherwise we can construct a one-many tampering adversary with error higher than ϵ. To construct such an adversary $\vec{(f^i)}$, each f^i_j has $X_{k k\neq i}$ hardcoded in it and takes X_i as input. This would show an adversary against which one-many non-malleable codes have an error higher than ϵ.

By the time we reach $(u-1)$-th hybrid experiment, the error could only have reduced by at most $(u-1)\epsilon$. However in the $(u-1)$-th hybrid experiment, the error can at most be ϵ since it corresponds to the one-many setting. Hence, the error in the hybrid 0 could have been at most $u.\epsilon$. This concludes the proof.

□

Relation to Continuous Non-Malleable Codes A primitive related to one-many non-malleable codes that we introduce, known as continuous non-malleable codes, was introduced by Faust et al. [FMNV14]. Informally, in a continuous non-malleable code, the codewords are allowed to be tampered multiple times (without allowing fresh encoding of the message), with the additional guarantee that the tampering experiment stops (called "self destruct") whenever an error message is detected. This model is weaker than the notion we consider since we do not allow for such a self-destruct option. However the work of [FMNV14] allows for unbounded number of tamperings. On the other hand, their constructions are based on computational assumptions while ours are purely information-theoretic.

The work of Jafargholi and Wichs [JW15] studied variants of continuous non-malleable codes, depending on whether the tampering is persistent (i.e., the new tampering is on the current tampered version of the codeword) or non-persistent (i.e., the tampering is always on the original codeword). Further [JW15] considered variants depending on whether the self-destruct option is available.

It was shown in [FMNV14] that continuous non-malleable codes against unbounded tampering in the non-persistent model cannot exist in the information theoretic setting. Subsequently, the work of [JW15] proved the existence of continuous non-malleable codes against unbounded tampering in the persistent model (with self-destruct) in the information theoretic setting. Following this, in a recent work Aggarwal, Kazana and Ombreski [AKO15] provided explicit constructions of such codes.

Thus, our result on one-many non-malleable codes can be interpreted as an explicit construction of continuous non-malleable codes in the non-persistent model (without self-destruct) against a bounded tampering in the information-theoretic model. We note that as implied by the result of [FMNV14], one cannot hope to handle unbounded tampering in this model in the information theoretic setting.

Non-malleable Codes in the Split-State Model An important and well studied family of tampering functions (which is also relevant to the current work) is the family of tampering functions in the C-split-state model, for $C \geq 2$. In this model, each tampering function f is of the form $(f_1, \ldots, f_C)$ where $f_i \in \mathcal{F}_{n/C}$, and for any codeword $x = (x_1, \ldots, x_C) \in (\{0,1\}^{n/C})^C$ we define $(f_1, \ldots, f_C)(x_1, \ldots, x_C) = (f_1(x_1), \ldots, f_C(x_C))$. Thus each f_i independently tampers a fixed partition of the codeword. Non-malleable codes in this model can also be viewed as *non-malleable secret sharing*. This is because the strings $(x_1, \ldots, x_C)$ can be seen as the shares of s and tampering each share individually does not allow one to "maul" the shared secret s.

There has been a lot of recent work on constructing explicit and efficient non malleable codes in the C-split-state model. Since $C = 1$ includes all of $\mathcal{F}_n$, the best one can hope for is $C = 2$. A Monte-Carlo construction of non-malleable codes in this model was given in the original paper on non-malleable codes [DPW10] for $C = 2$ and then improved in [CG14a]. However, both of these constructions are inefficient. For $C = 2$, these Monte-Carlo constructions imply existence of codes of rate close to $\frac{1}{2}$ and corresponds to the hardest case. On the other extreme, when $C = n$, it corresponds to the case of bit tampering where each function f_i acts independently on a particular bit of the codeword. By a recent line of work [DKO13] [ADL14] [CG14b] [CZ14] [ADKO15], we now have almost optimal constructions of non-malleable codes in the C-state-state model, for any $C \geq 2$.

Many-many non-malleable secret sharing. Consider the example of non-malleable secret sharing [ADL14]. What if there are shares of multiple secrets which the adversary can tamper with? What if the adversary is allowed to output shares of multiple secrets? For example, say there are two secret and two devices. Each device stores one share of each of the secrets. Say that an adversary is able to tamper with the data stored on each device individually (or infect each of them with a virus). Then, the current notion of one-one NM codes does not rule out a non-trivial relationship between two resulting secrets and the two original secrets we start with. It is conceivable that what we need here is a *two-two* non-malleable secret sharing. Our many-many non-malleable codes directly lead to such a many-many non-malleable secret sharing scheme.

1.2 Summary of results

Our first main result is an explicit construction of a $(2, t)$-seedless non-malleable extractor. We note that prior to this work, such a construction was not known for even $t = 1$ and full min-entropy.

Theorem 1. *There exists a constant $\gamma > 0$ such that for all $n > 0$ and $t \leq n^{\gamma}$, there exists an efficient seedless $(2, t)$-NM extractor at min-entropy $n - n^{\gamma}$ with error $2^{-n^{\Omega(1)}}$ and output length $m = n^{\Omega(1)}$.*

Next, we show that it is possible to efficiently sample almost uniformly from the pre-image of any output of this extractor. We prove this in Section 8. Combining this with Theorem 5.1 and a hybrid argument, we immediately have the following result.

Theorem 2. *There exists a constant $\gamma > 0$ such that for all $n > 0$ and $t \leq n^{\gamma}$, there exists an efficient construction of one-many non-malleable codes in the 2-split state model with tampering degree t, relative rate $n^{\Omega(1)}/n$, and error $2^{-n^{\Omega(1)}}$.*

We next improve the min-entropy rate requirements of seeded non-malleable extractors. As mentioned above, prior to this work, the best known such construction worked for min-entropy rate 0.499 [Li12b]. We have the following result.

Theorem 3. *There exists a constant c such that for all $n > 0$ and $\epsilon > 0$, and $k \geq c \log^2\left(\frac{n}{\epsilon}\right)$, there exists an explicit construction of a seeded non-malleable extractor* $\mathrm{snmExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$*, with $m = \Omega(k)$ and $d = O\left(\log^2\left(\frac{n}{\epsilon}\right)\right)$.*

We in fact have a more general result, and can handle t-adversarial functions in the seeded non-malleable case as well, which improves the result of [CRS14]. As discussed above, such t-non-malleable extractors was used in the recent constructions of two-source extractors for polylogarithmic min-entropy by Chattopadhyay and Zuckerman [CZ15]. We refer the reader to Section 7 for more details.

Combined with the protocol developed in [DW09], this immediately gives the following result about privacy amplification, which matches the best known result in [Li15a] but has a simpler protocol.

Theorem 4. *There exists a constant C such that for any $\varepsilon > 0$ with $k \geq C(\log n + \log(1/\varepsilon))^2$, there exists an explicit 2-round privacy amplification protocol with an active adversary for (n, k) sources, with security parameter $\log(1/\varepsilon)$ and entropy loss $O(\log n + \log(1/\varepsilon))$.*

1.3 Other possible approaches to construct one-many non-malleable codes

Since a major part of this paper is devoted to constructing explicit seedless $(2, t)$-non-malleable extractors (and providing efficient sampling algorithms for almost uniformly sampling from the pre-image of any output), and one of the major motivation for such explicit extractors is to construct one-many non-malleable codes in the 2-split-state model, a natural question is whether existing constructions of non-malleable codes in the 2-split state model can be modified to satisfy the stronger notion of one-many non-malleable codes.

Our first observation is that not every construction of a one-one non-malleable code satisfies the stronger notion of being a one-many non-malleable code. Intuitively this is because, say Enc and Dec are the encoding and decoding function of some non-malleable code against some class of tampering functions $\mathcal{F}$. Thus, for $f_1, f_2 \in \mathcal{F}$, and any message m, $\mathrm{Dec}(f_1(\mathrm{Enc}(m)))$ is close to D_{f_i}, $i = 1, 2$. But it is possible this does not rule out the possibility that, for instance $\mathrm{Dec}(f_2(\mathrm{Enc}(m))) = \mathrm{Dec}(f_1(\mathrm{Enc}(m))) + m + 1$. Clearly, this code is not non-malleable against two adversaries from $\mathcal{F}$, and hence is not one-many.

We now take a specific example. Suppose $\mathcal{C}$ is a one-one NM code against 2-split-state adversaries. We construct another code $\mathcal{C}'$ where the message m is broken into m_1 and m_2 using additive secret sharing. Then one encodes both m_1 and m_2 separately using the encoder of $\mathcal{C}$ (and includes both as part of the code, each encoding being equally divided in two halves). It is easy to show that $\mathcal{C}'$ is still one-one NM.

On the other hand, if the two adversaries act in the following way: one adversary can take the encoding of m_1 and put an encoding of 1 on his own. That will be the first output code (to message $m_1 + 1$). Next,

the other adversary can take the encoding of m_2 and put an encoding of 0 on its own. That will be second output code (to message m_2). Now it can be seen that the two output code sum to $m + 1$. Thus, $\mathcal{C}'$ is not one-many in the 2-split-state model.

It turns out that existing constructions of non-malleable codes in the split-state model either fail to satisfy stronger notion of one-many, or at least it appears non-trivial to generalize the proofs of non-malleability against multiple adversaries. We briefly discuss these approaches, and why it appears non-trivial to extend them to handle multiple adversaries. A first approach could be to generalise the reductions in the recent work of Aggarwal et al.[ADKO15], and possibly show that one-many non-malleable codes in the 2-split-state model can be reduced to the problem of constructing one-many non-malleable codes in the bit-tampering model. However, each known construction of a NM code in the bit-tampering model [CG14b] [AGM$^+$14] follows the general approach of starting out with an initial non-malleable code in the 2-split-state model (which is also an NM code against bit-wise tampering) of possibly low rate, and then amplifies the rate to almost optimal. Thus, it is not clear how to use this approach to construct one-many NM codes in the 2-split-state model.

Another approach could be to show that the non-malleable codes constructed by Aggarwal et al. [ADL14] generalize to handle many adversaries. However, from a careful examination of their proof it turns out that it is crucially used that the inner product function is an extractor for weak sources at min-entropy rate slightly greater than $\frac{1}{2}$. It turns out that this fact is tailor made for exactly one adversary, and for handling $t > 1$ adversaries one needs that the inner product function is an extractor for min-entropy rate approximately $\frac{1}{t+1}$, which is not true. Thus, it is not clear how to extend their approach as well.

Finally, a third approach could be to extend the construction of the seedless non-malleable extractor for 10 sources in the work of Chattopadhyay and Zuckerman [CZ14]. However, from a careful examination of the proof it follows that the crucial step of first constructing a seedless non-malleable condenser based on a sum-product theorem fails to generalize when there are more than one adversary.

Thus, it appears that our new explicit constructions of seedless $(2, t)$-non-malleable extractors are a necessity for constructing one-many non-malleable codes in the split-state model.

1.4 Related work on privacy amplification

As mentioned above, seeded non-malleable extractors were introduced by Dodis and Wichs in [DW09], to study the problem of privacy amplification with an active adversary.

The goal is roughly as follows. We pick a security parameter s, and if the adversary Eve remains passive during the protocol then the two parties should achieve shared secret random bits that are 2^{-s}-close to uniform. On the other hand, if Eve is active, then the probability that Eve can successfully make the two parties output two different strings without being detected is at most 2^{-s}. We refer the readers to [DLWZ14] for a formal definition.

Here, while one can still design protocols for an active adversary, the major goal is to design a protocol that uses as few number of interactions as possible, and output a uniform random string R that has length as close to $H_\infty(X)$ as possible (the difference is called *entropy loss*). When the entropy rate of X is large, i.e., bigger than $1/2$, there exist protocols that take only one round (e.g., [MW97], [DKRS06]). However these protocols all have very large entropy loss. On the other hand, [DW09] showed that when the entropy rate of X is smaller than $1/2$, then no one round protocol exists; furthermore the length of R has to be at least $O(s)$ smaller than $H_\infty(X)$. Thus, the natural goal is to design a two-round protocol with such optimal entropy loss. There has been a lot of effort along this line [MW97], [DKRS06], [DW09], [RW03], [KR09], [CKOR10], [DLWZ14], [CRS14], [Li12a], [Li12b]. However, all protocols before the work of [DLWZ14] either need to use $O(s)$ rounds, or need to incur an entropy loss of $O(s^2)$.

In [DW09], Dodis and Wichs showed that the previously defined seeded non-malleable extractor can be used to construct 2-round privacy amplification protocols with optimal entropy loss. They further showed that seeded non-malleable extractors exist when $k > 2m + 3\log(1/\varepsilon) + \log d + 9$ and $d > \log(n - k + 1) + 2\log(1/\varepsilon) + 7$. However, they were not able to construct such extractors. The first explicit construction of seeded non-malleable extractors appeared in [DLWZ14], with subsequent improvements in [CRS14], [Li12a], [DY13]. However, all these constructions require the entropy rate of the weak source to be bigger than $1/2$. In another paper, Li [Li12b] gave the first explicit non-malleable extractor that breaks this barrier, which works for entropy rate $1/2-\delta)$ for some constant $\delta > 0$. This is the best known seeded non-malleable extractor to date. Further, [Li12b] also showed a connection between seeded non-malleable extractors and two-source extractors, which suggests that constructing explicit seeded non-malleable extractors with small seed length for smaller entropy may be hard.

In a different line of work, Li [Li12a] introduced the notion of *non-malleable condenser*, which is a weaker object than seeded non-malleable extractor. He then constructed explicit non-malleable condensers for entropy as small as $k = \text{polylog}(n)$ in [Li15a] and used them to give the first two-round privacy amplification protocol with optimal entropy loss, subject to the constraint that $k \geq s^2$.

1.5 Related work on non-malleable codes

We give a summary of known constructions of non-malleable codes. As remarked above, all known explicit constructions of non-malleable codes in the information theoretic setting are in framework of what we call as one-one non-malleable codes. That is, the adversary is only given as input a single code and outputs a single code.

Since the introduction of non-malleable codes by Dziembowski, Pietrzak and Wichs [DPW10], the most well studied model is the C-split-state model introduced above. By a recent line of work [DKO13] [ADL14] [CG14b] [CZ14] [ADKO15], we now have almost optimal constructions of non-malleable codes in the C-state-state model, for any $C \geq 2$.

In the model of global tampering, Agrawal et al. [AGM$^+$14] constructed efficient non-malleable codes with rate $1 - o(1)$ against a class of tampering functions slightly more general than the family of permutations.

There were also some other conditional results. Liu and Lysyanskaya [LL12] constructed efficient constant rate non-malleable codes in the split-state model against computationally bounded adversaries under strong cryptographic assumptions. The work of Faust et al. [FMVW13] constructed almost optimal non-malleable codes against the class of polynomial sized circuits in the CRS framework. [CCP12], [CCFP11], [CKM11], and [FMNV14] considered non-malleable codes in other models.

The recent work of Chandran et al. [CGM$^+$15] found interesting connections between non-malleable codes in a model slightly more general than the split-wise model and non-malleable commitment schemes.

Organization

We give an overview of all our explicit constructions in Section 2. We introduce some preliminaries in Section 3, and formally define seeded and seedless non-malleable extractors in Section 4. We use Section 5 to present the connection between seedless $(2, t)$-non-malleable extractors and one-many non-malleable codes in the 2-split-state model. In Section 6, we present an explicit construction of a seedless $(2, t)$-non-malleable extractor. An explicit construction of a seeded non-malleable extractor construction at polylogarithmic min-entropy is presented in Section 7. Finally, we use Section 8 to give efficient encoding and decoding algorithms for the resulting one-many non-malleable codes.

2 Overview of Our Constructions

In this section, we give an overview of the main ideas involved in our constructions. The main ingredient in all our constructions is an explicit seedless $(2, t)$-non-malleable extractor. Further, we give efficient algorithms for almost uniformly sampling from the pre-image of any output of this extractor. The explicit construction of many-many non-malleable codes in the 2-split state model with tampering degree t then follow in a straightforward way using the connection via Theorem 5.1.

It turns out that by a simple modification of the construction of our seedless non-malleable extractor, we also have an explicit construction of a seeded non-malleable extractor which works for any min-entropy $k \geq \log^2 n$. We will give an overview of how to achieve this as well.

2.1 A Seedless $(2, t)$-Non-Malleable Extractor

Let γ be a small enough constant and C a large enough one. Let $t = n^{\gamma/C}$.

We construct an explicit function nmExt : $\{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$, $m = n^{\Omega(1)}$ which satisfies the following property: If X and Y be independent $(n, n - n^\gamma)$-sources on $\{0,1\}^n$, and $\mathcal{A}_1 = (f_1, g_1), \ldots,$ $\mathcal{A}_t = (f_t, g_t)$ are arbitrary 2-split sate tampering functions such that for any $i \in [t]$, at least one of f_i or g_i has no fixed points, the following holds:

$$|\text{nmExt}(X, Y) \circ \text{nmExt}(\mathcal{A}_1(X, Y)) \circ \ldots \text{nmExt}(\mathcal{A}_t(X, Y)) - U_m \circ \text{nmExt}(\mathcal{A}_1(X, Y)) \circ \ldots \text{nmExt}(\mathcal{A}_t(X, Y))| \leq \epsilon,$$

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

By a convex combination argument (Lemma 6.14), we show that if nmExt satisfies the property above, then it is indeed a seedless $(2, t)$-non-malleable extractor (Definition 4.4).

We introduce some notation.

Notation: For any function H, and $V = H(X, Y)$, we use $V^{(i)}$ to denote the random variable $H(\mathcal{A}_i(X, Y))$. If $Z_a, Z_{a+1}, \ldots, Z_b$ are random variables, we use $Z_{[a,b]}$ to denote the random variable $(Z_a, \ldots, Z_b)$. For any bit string z, let $z_{\{h\}}$ denote the symbol in the h'th co-ordinate of x. For a string x of length m, and $T \subseteq [m]$, let $x_{\{T\}}$ be the projection of x onto the co-ordinates indexed by T. For a string x of length m, define the string $\text{Slice}(x, w)$ to be the prefix of x with length w.

The high level idea of the non-malleable extractor is as follows. Initially we have two independent sources (X, Y) and t tampered version $\{\mathcal{A}_i(X, Y)\}$, which can depend arbitrarily on (X, Y). We would like to gradually break the dependence of $\{\mathcal{A}_i(X, Y)\}$ on (X, Y), until at the end we get an output $\text{nmExt}(X, Y)$ which is independent of all $\{\text{nmExt}(\mathcal{A}_i(X, Y))\}$.

Towards this end, we would first like to create a short tag (string) from (X, Y) that can distinguish from $\{\mathcal{A}_i(X, Y)\}$. More specifically, we will obtain a tag Z of length $n^{\Omega(1)}$ from (X, Y), such that with high probability Z is different from all $\{Z^{(i)}\}$ obtained from $\{\mathcal{A}_i(X, Y)\}$. Next, we will run some iterative steps of extraction from (X, Y), with each step based on one bit of Z. The crucial property we will have here is that whenever we reach a bit of Z which is different from the corresponding bits of $\{Z^{(i)}, i \in S\}$ for some subset $S \subseteq [t]$, in that particular step the output of our extraction from (X, Y) will be (close to) uniform and independent of all the corresponding outputs obtained from $\{A_i(X, Y), i \in S\}$. Furthermore this will remain true in all subsequent steps of extraction. Therefore, since Z is different from all $\{Z^{(i)}, i \in [t]\}$, we know that at the end our output $\text{nmExt}(X, Y)$ will be independent of all $\{\text{nmExt}(\mathcal{A}_i(X, Y)), i \in [t]\}$. We now elaborate about the two steps in more details below.

Step 1: Here we use the sources X and Y to obtain a tag Z, such that for each $i \in [t]$, $Z \neq Z^{(i)}$ with

probability at least $1 - 2^{-n^{\Omega(1)}}$. Thus by a union bound with probability $1 - 2^{-n^{\Omega(1)}}$ we have that Z is different form all $\{Z^{(i)}, i \in [t]\}$. To obtain Z, we first take two small slices X_1 and Y_1 from the sources X and Y respectively, with size at least $3n^{\gamma}$; and use the strong inner product 2-source extractor IP to generate an almost uniform random variable $V = \text{IP}(X_1, Y_1)$. Now we take an explicit asymptotically good binary linear error correcting code, and obtain encodings $(E(X), E(Y))$ of (X, Y) respectively. We now use V to pseudorandomly sample $n^{\Omega(1)}$ bits from $E(X)$ to obtain X_2, and we do the same thing to obtain Y_2 from $E(Y)$. We use known constructions of an averaging sampler Samp [Zuc97] [Vad04] (see Definition 8.4) to do this (in fact, we can even sample completely randomly since V is close to uniform).

Now define

$$Z = X_1 \circ Y_1 \circ X_2 \circ Y_2.$$

The length of Z is $\ell = n^{\beta}$ bits for some small constant β. Fix some $i \in [t]$. We claim that $Z \neq Z^{(i)}$ with probability at least $1 - 2^{-n^{\Omega(1)}}$. To see this, assume without loss of generality that f_i has no fixed points. If $X_1 \neq X_1^{(i)}$ or $Y_1 \neq Y_1^{(i)}$, then we have $Z \neq Z^{(i)}$. Now suppose $X_1 = X_1^{(i)}$ and $Y_1 = Y_1^{(i)}$. Thus, $V = V^{(i)}$. We fix X_1, since IP is a strong extractor (Theorem 3.17), V is still close to uniform and now it is a function of Y, and thus independent of X. Since $X \neq X^{(i)}$, by the property of the code, we know that $E(X)$ and $E(X^{(i)})$ must differ in at least a constant fraction of co-ordinates. Thus, if we uniformly (or pseudorandomly) sample $n^{\Omega(1)}$ bits from these coordinates, then with probability $1 - 2^{-n^{\Omega(1)}}$ the sampled strings will be different.

We can now fix $Z, \{Z^{(i)} : i \in [t]\}$, such that $Z \neq Z^{(i)}$ for any i. Since the size of each $Z^{(i)}$ is small, we have that conditioned on this fixing, the sources X and Y are still independent and have min-entropy at least $n - O(t\ell)$ each (with high probability).

Step 2: Here our goal is to gradually break the dependence of $\{\mathcal{A}_i(X, Y)\}$ on (X, Y), until at the end we get an output $\text{nmExt}(X, Y)$ which is independent of all $\{\text{nmExt}(\mathcal{A}_i(X, Y))\}$. To achieve this, our crucial observation is that while many other techniques in constructing non-malleable seeded extractors (such as those in [DLWZ14], [CRS14], [Li12b] fail in the case where both sources are tampered, the powerful technique of alternating extraction, introduced by Dziembowski and Pietrzak [DP07], still works. Thus, we will be relying on this technique, which has been used a lot in recent studies of extractors and privacy amplification [DW09], [Li12a], [Li12b], [Li13b], [Li13a], [Coh15], [Li15b]. We now briefly recall the details. The alternating extraction protocol is an interactive protocol between two parties, Quentin and Wendy, using two strong seeded extractors Ext_q, Ext_w. Assume initially Wendy has a weak source X and Quentin has another source Q and a short uniform random string S_1.[2] Suppose that X is independent of (Q, S_1). In the first round, Quentin sends S_1 to Wendy, Wendy computes $R_1 = \text{Ext}_w(X, S_1)$ and sends it back to Quentin, and Quentin then computes $S_2 = \text{Ext}_q(Q, R_1)$. Continuing in this way, in round i, Quentin sends S_i, Wendy computes the random variables $R_i = \text{Ext}_w(X, S_i)$ and sends it to Quentin, and Quentin then computes the random variable $S_{i+1} = \text{Ext}_q(Q, R_i)$. This is done for some u steps, and each of the random variables R_i, S_i is of length m. Thus, the following sequence of random variables is generated:

$$S_1, R_1 = \text{Ext}_w(X, S_1), S_2 = \text{Ext}_q(Q, R_1), \ldots, S_u = \text{Ext}_q(Q, R_{u-1}), R_u = \text{Ext}_w(X, S_u).$$

Also define the following look-ahead extractor:

$$\text{laExt}(X, (Q, S_1)) = R_1, \ldots, R_u$$

Now suppose we have t tampered versions of X: $X^{(1)}, \ldots, X^{(t)}$, which can depend on X arbitrarily; and t tampered versions of (Q, S_1): $(Q^{(1)}, S_1^{(1)}), \ldots, (Q^{(t)}, S_1^{(t)})$, which can depend on (Q, S_1) arbitrarily. Let $\text{laExt}(X, (Q, S_1)) = R_1, \ldots, R_u$, and for $h \in [t]$, let $\text{laExt}(X^{(h)}, (Q^{(h)}, S_1^{(h)})) = R_1^{(h)}, \ldots, R_t^{(h)}$. As

[2] In fact, S_1 can be a slightly weak random source as well.

long as $(X, X^{(1)}, \ldots, X^{(t)})$ is independent of $((Q, S_1), (Q^{(1)}, S_1^{(1)}), \ldots, (Q^{(t)}, S_1^{(t)}))$ and t, u, m are small compared to the entropy of X and Q, one can use induction together with standard properties of strong seeded extractors to show that the following holds: for any $j \in [u]$,

$$\begin{aligned} &R_j, \{R_i^{(h)} : i \in [j-1], h \in [t]\}, \{(Q^{(h)}, S_1^{(h)}) : h \in [t]\} \\ \approx\ &U_m, \{R_i^{(h)} : i \in [j-1], h \in [t]\}, \{(Q^{(h)}, S_1^{(h)}) : h \in [t]\} \end{aligned}$$

Based on this property, we describe two different approaches to achieve our goal in Step 2. The first approach was our initial construction, while the second approach is inspired by new techniques in a recent work of Cohen [Coh15]. It turns out the second approach is simpler and more suitable for our application to many-many non-malleable codes, thus we only provide the formal proof for the second approach in this paper (see Section 6). Recall that the high level idea in both approaches is that we will proceed bit by bit based on the previously obtained string Z, which is different from all $\{Z^{(i)}, i \in [t]\}$. Whenever we reach a bit of Z which is different from the corresponding bits of $\{Z^{(i)}, i \in S\}$ for some subset $S \subseteq [t]$, in that particular step the output of our extraction from (X, Y) will be (close to) uniform and independent of all the corresponding outputs obtained from $\{A_i(X, Y), i \in S\}$. Furthermore this will remain true in all subsequent steps of extraction. We will achieve this by running some alternating extraction protocol for ℓ times, where ℓ is the length of Z. Each time the alternating extraction will be between X and a new $(Q_h, S_{1,h})$ obtained from Y, where we take $S_{1,h}$ to be a small slice of Q_h.

Construction 1:[3] Our first approach is based on a generalization of the techniques in [Li13a]. Here we fist achieve an intermediate goal: whenever we reach a bit of Z which is different from the corresponding bits of $\{Z^{(i)}, i \in S\}$ for some subset $S \subseteq [t]$, the output of our extraction from (X, Y) will *have some entropy* conditioned on all the corresponding outputs obtained from $\{A_i(X, Y), i \in S\}$. Suppose at step h $(1 \leq h \leq \ell)$ we have obtained Q_h from Y (in the first step we can take a small slice of Y to be Q_1) and use it to run an alternating extraction protocol with X. We run the alternating extraction for $t + 2$ rounds and obtain outputs $R_{h,1}, \ldots, R_{h,t+2}$. The crucial idea is to use the h'th bit of Z, to set a random variable W_h as either $(R_{h,1}, \ldots, R_{h,t+1})$ or $R_{h,t+2}$ (appended with an appropriate number of 0's to make them the same length).

Now consider the subset $S \subseteq [t]$ where the h'th bit of Z is different from the h'th bit of $\{Z^{(i)}, i \in S\}$. If $W_h = (R_{h,1}, \ldots, R_{h,t+1})$ then for all $i \in S$, we have $W_h^{(i)} = R_{h,t+2}^{(i)}$. Since S has at most t elements, the size of $\{W_h^{(i)}\}$ is at most tm. Note that W_h has size $(t + 1)m$ and is close to uniform. Thus W_h has entropy roughly m conditioned on $\{W_h^{(i)}, i \in S\}$ (here we can ignore the appended 0's in $W_h^{(i)}$ since they won't affect the entropy in W_h). On the other hand, if then $W_h = R_{h,t+2}$ then for all $i \in S$, we have $W_h^{(i)} = R_{h,1}^{(i)}, \ldots, R_{h,t+1}^{(i)}$. By the property of alternating extraction we have that W_h is close to uniform conditioned on $\{W_h^{(i)}, i \in S\}$.

We can now go from having conditional entropy to being conditional uniform, as follows. We first convert W_h into a somewhere random source by applying an optimal seeded extractor and trying all possible choices of the seed. One can show that conditioned on previous random variables generated in our algorithm, W_h is now a deterministic function of X and thus independent of Y and Q_h. We now take another optimal seeded extractor and use each row in this somewhere random source as a seed to extract a longer output from Q_h. In this way we obtain a new somewhere random source. If we choose parameters appropriately we can ensure that the size of W_h is much smaller than the entropy of Q_h, and thus the number of rows in this new somewhere random source is much smaller than its row length. Therefore, by using an extractor from [BRSW06] we can use this somewhere random source to extract a close to uniform output V_h from

[3] formal proofs of the claims in the sketch of Construction 1 are not provided in this paper.

X. Since W_h has entropy at least m conditioned on $\{W_h^{(i)}, i \in S\}$, as long as the size of V_h is small, using standard arguments one can show that V_h will be close to uniform conditioned on all $\{V_h^{(i)}, i \in S\}$.

We now go into the next step of alternating extraction, where we will take a strong seeded extractor and use V_h to extract a uniform string Q_{h+1} from Y. We will then use X and Q_{h+1} to do the alternating extraction for next step. The point here is that whenever we have V_h is close to uniform conditioned on all $\{V_h^{(i)}, i \in S\}$ for some $S \subseteq [t]$, we can show that Q_{h+1} is close to uniform conditioned on all $\{Q_{h+1}^{(i)}, i \in S\}$. Thus in the next step of alternating extraction, we can first fix all $\{Q_{h+1}^{(i)}, i \in S\}$, and then fix all the $\{R_{h+1,j}^{(i)}, i \in [S], j \in [t+2]\}$, and all the $\{V_{h+1}^{(i)}, i \in S\}$ (these will now be deterministic functions of X). Conditioned on this fixing Q_{h+1} is still close to uniform, and X still has a lot of entropy left (as long as the size of each $R_{h+1,j}^{(i)}$ and $V_{h+1}^{(i)}$ is small). Therefore, in this step V_{h+1} will be close to uniform even conditioned on all $\{V_{h+1}^{(i)}, i \in S\}$, i.e., once we have independence it will continue to hold in subsequent steps. Thus our goal is achieved.

Construction 2: Here we replace our approach in Construction 1 with a more direct approach, by using the idea of "flip-flop" alternating extraction introduced in a recent paper by Cohen [Coh15], which is again based on the techniques developed in [Li13a]. Again, assume we are now looking at the h'th bit of the tag Z, and we have obtained Q_h from Y.

Now each step of alternating extraction will consist of two sub steps of alternating extraction, with each sub step taking two rounds. In the first sub step, we use X and Q_h to perform an alternating extraction for two rounds and output $R_{h,1}, R_{h,2}$. If the h'th bit of Z is 0, we take $V_h = R_{h,1}$; otherwise we take $V_h = R_{h,2}$. Now we will take a strong seeded extractor Ext and use V_h to extract $\overline{Q}_h = \text{Ext}(Y, V_h)$ from Y. We then use $\overline{Q}_h$ and X to perform the second sub step of alternating extraction, which again runs for two rounds and outputs $\overline{R}_{h,1}, \overline{R}_{h,2}$. Now if the h'th bit of Z is 0, we take $\overline{V}_h = \overline{R}_{h,2}$; otherwise we take $\overline{V}_h = \overline{R}_{h,1}$. One can see that this is indeed in a "flip-flop" manner.

The idea is as follows. Consider the h'th bit of Z, and let $S \subseteq [t]$ be such that for all $i \in S$, we have $Z_{\{h\}} \neq Z_{\{h\}}^{(i)}$. Now consider the h'th step of alternating extraction. If $Z_{\{h\}} = 0$, then in the first sub step of alternating extraction, $V_h = R_{h,1}$; while for all $i \in S$, we have $V_h^{(i)} = R_{h,2}^{(i)}$. Now it's possible that V_h depends on $\{V_h^{(i)}, i \in S\}$, and thus $\overline{Q}_h$ also depends on $\{\overline{Q}_h^{(i)}, i \in S\}$. However, when we go into the second sub step of alternating extraction, we will choose $\overline{V}_h = \overline{R}_{h,2}$; while for all $i \in S$, we have $\overline{V}_h^{(i)} = \overline{R}_{h,1}^{(i)}$. Thus by the property of alternating extraction, we have that $\overline{V}_h$ is close to uniform conditioned on all $\{\overline{V}_h^{(i)}, i \in S\}$.

On the other hand, if $Z_{\{h\}} = 1$, then in the first sub step of alternating extraction, $V_h = R_{h,2}$; while for all $i \in S$, we have $V_h^{(i)} = R_{h,1}^{(i)}$. Thus in this sub step, by the property of alternating extraction, we have that V_h is close to uniform conditioned on all $\{V_h^{(i)}, i \in S\}$. Therefore we also have that $\overline{Q}_h$ is close to uniform conditioned on all $\{\overline{Q}_h^{(i)}, i \in S\}$, and they are deterministic functions of Y given V_h and $\{V_h^{(i)}, i \in S\}$. Thus, when we go into the second sub step of alternating extraction, we can first fix all $\{\overline{Q}_h^{(i)}, i \in S\}$ and $\overline{Q}_h$ is still close to uniform. Now all $\{\overline{V}_h^{(i)}, i \in S\}$ will be deterministic functions of X, and thus we can further fix them. As long as the size of each $\overline{R}_{h,j}$ is small, conditioned on this fixing X still has a lot of entropy left. Therefore $\overline{Q}_h$ can still be used to perform an alternating extraction with X, and this gives us that $\overline{V}_h = \overline{R}_{h,1}$ is close to uniform. That is, again we get that $\overline{V}_h$ is close to uniform conditioned on all $\{\overline{V}_h^{(i)}, i \in S\}$.

Once we have this property, we can go into the next step of alternating extraction. We will now take a strong seeded extractor and use $\overline{V}_h$ to extract Q_{h+1} from Y, and then use X and Q_{h+1} to perform the next

step of alternating extraction. Since $\overline{V}_h$ is close to uniform conditioned on all $\{\overline{V}_h^{(i)}, i \in S\}$, we also have that Q_{h+1} is close to uniform conditioned on all $\{Q_{h+1}^{(i)}, i \in S\}$. Thus by the same argument above, we can first fix all $\{Q_{h+1}^{(i)}, i \in S\}$ and all $\{V_{h+1}^{(i)}, i \in S\}$, and conditioned on this fixing Q_{h+1} is still close to uniform. Therefore V_{h+1} will be close to uniform conditioned on all $\{V_{h+1}^{(i)}, i \in S\}$. Thus, going into the second sub step, we will also have that $\overline{Q}_{h+1}$ is close to uniform conditioned on all $\{\overline{Q}_{h+1}^{(i)}, i \in S\}$. Thus again we can first fix all $\{\overline{Q}_{h+1}^{(i)}, i \in S\}$ and all $\{\overline{V}_{h+1}^{(i)}, i \in S\}$, and conditioned on this fixing $\overline{Q}_{h+1}$ is still close to uniform. Therefore we get that $\overline{V}_{h+1}$ is close to uniform conditioned on all $\{\overline{V}_{h+1}^{(i)}, i \in S\}$, i.e., once we have independence it will continue to hold in subsequent steps. Thus our goal is achieved.

2.2 An Explicit Seeded Non-Malleable Extractor for Polylogarithmic Min-Entropy

Let $\gamma > 0$ be a small constant. For any $\epsilon > 0$, let $k \geq O\left(\log^{2+\gamma}\left(\frac{n}{\epsilon}\right)\right)$, $t \leq k^{\gamma/2}$ and $d = O\left(t^2 \log^2(\frac{n}{\epsilon})\right)$. We construct a function snmExt : $\{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$, $m = O\left(\log\left(\frac{n}{\epsilon}\right)\right)$, such that the following holds: If X is a (n,k)-source, Y is an independent uniform seed of length d, and $\mathcal{A}_1, \ldots, \mathcal{A}_t$ are arbitrary functions with no fixed points, then the following holds:

$$\begin{aligned}&\text{snmExt}(X,Y), \text{snmExt}(X, \mathcal{A}_1(Y)), \ldots, \text{snmExt}(X, \mathcal{A}_t(y))\\ \approx_\epsilon\ &U_m, \text{snmExt}(X, \mathcal{A}_1(Y)), \ldots, \text{snmExt}(X, \mathcal{A}_t(y))\end{aligned}$$

We now describe our construction, which is essentially a simple modification of our seedless non-malleable extractor construction.

Step 1: Let Y_1 be a small slice of Y. Compute $V = \text{Ext}(X, Y_1)$, where Ext is a strong seeded extractor. Now we use V to randomly sample bits from $E(Y)$, where E is the encoder of an asymptotically good error correcting code. Let the sampled bits be Y_2. We define

$$Z = Y_1 \circ Y_2$$

We show that with high probability $Z \neq Z^{(i)}$ for all $i \in [t]$. We provide a brief sketch of the argument. Fix any $i \in [t]$. If $Y_1 \neq Y_1^{(i)}$, then clearly $Z \neq Z^{(i)}$. Now suppose $Y_1 = Y_1^{(i)}$. We fix Y_1, and since Ext is a strong seeded extractor, it follows that V is still close to uniform, and is a deterministic function of X, thus independent of $Y, \{Y^i, i \in [t]\}$. Therefore V can be used to sample bits from Y. Since $\mathcal{A}_i$ has no fixed points, it follows that $Y \neq Y^{(i)}$. Thus $E(Y)$ and $E(Y^{(i)})$ must differ in at least a constant fraction of coordinates. Therefore with high probability $Y_2 \neq Y_2^{(i)}$. By a union bound, with high probability $Z \neq Z^{(i)}$ for all $i \in [t]$.

Step 2: As long as the size of (Y_1, V, Y_2) is small, we can show that conditioned on the fixing of these variables, X and Y are still independent. Moreover both X and Y only lose a small amount of entropy. Now we can use any of Construction 1 and Construction 2 above to finish the extraction. The same argument will show that at the end $\text{snmExt}(X,Y)$ will be close to uniform conditioned on all $\{\text{snmExt}(A, \mathcal{A}_i(Y)), i \in [t]\}$.

We refer the reader to Section 7 for more details.

Comparison to the LCB in [Coh15] Our second approach in constructing non-malleable two-source extractors is inspired by the work of [Coh15]. Especially, we use the idea of "flip-flop" alternating extraction introduced there. However, there are also some differences between our construction and the "Local Correlation Breaker" constructed in [Coh15], which are worth pointing out.

First, in our construction, both sources X and Y are tampered. This results in t random variables $X^{(1)}$, $\ldots, X^{(t)}$ that are arbitrarily correlated with X, and t random variables $Y^{(1)}, \ldots, Y^{(t)}$ that are arbitrarily

correlated with Y. In contrast, in the case of Local Correlation Breaker constructed in [Coh15], there are only correlated random variables with one source, while the other source is not tampered. In this sense, our construction can actually be viewed as given a stronger version of the LCB.

Second, the way to obtain a string that distinguishes the correlated parts is quite different. In the case of the LCB, one can simply use the index of each row in the somewhere random source. On the other hand, in our case we do not have such an index, since the only access we have are the two sources X and Y. Thus, we have to take extra efforts to create such a string from these two sources, by using error correcting codes and random sampling.

Connection to cryptographic non-malleable commitments In our constructions, we first generate a short "unique" tag from X and Y. Then, very roughly, our construction proceeds in multiple stage with each stage dependent on a particular bit of the tag. Independence (or non-malleability) is achieved in a stage where the corresponding bit of the tag is different from the tag for the tampered execution. This idea can be seen as inspired partly from the literature on non-malleable commitments. In particular, the seminal paper introducing non-malleability by Dolev, Dwork and Naor [DDN91] used such an idea. Further, observe that in our construction we gain independence from a particular adversary in a single (unknown) stage (when the bit in the tampered tag is 0 and the non-tampered bit is 1). Such ideas were also used in constructions of non-malleable commitment protocols in works of Pass and Rosen [PR08b], and Goyal [Goy11].

2.3 Efficient Algorithms for Many-Many Non-Malleable Codes

The above construction gives a $(2, t)$ non-malleable extractor. However, for our application to constructing explicit many-many non-malleable codes, given any output of the extractor we need to efficiently sample (almost) uniformly from its pre-image. To do this using the construction described above is highly non-trivial. Therefore, in order to make it easy to efficiently sample from the pre-image of an output (i.e., "inverting" the extractor), we use additional ideas to modify the non-malleable extractor. We now briefly describe the main ideas that we use. Recall that t is the number of tampered versions of the sources, and ℓ is the length of the string Z we obtained.

Idea 1: Since our construction of the $(2, t)$ non-malleable extractor involves multiple steps of alternating extraction, we need to first invert the extractors used in these steps. For this purpose, we will use linear seeded strong extractors in all alternating extraction steps. A linear seeded strong extractor is an extractor such that for any fixed seed, the output is a linear function of the input. Thus for any fixed seed, in order to sample uniformly from an output's pre-image, we can just sample uniformly according to a system of linear equations, which can be done efficiently.

Idea 2: Next, we will divide the sources X and Y into blocks. In each step of alternating extraction, we will also divide Q_h and $\overline{Q}_h$ into blocks. Then, whenever we use an extractor to extract from X, Q_h or $\overline{Q}_h$, we will use a completely new block of X, Q_h or $\overline{Q}_h$. When we apply an extractor to Y to generate Q_h or $\overline{Q}_h$, we will also use completely new blocks of Y to do this. This ensures that we do not have to deal with multiple compositions of extractors on the same string. That is, different applications of extractors are used on different parts of the inputs; so to invert them we can invert each part separately. Note that each alternating extraction takes at most 2 rounds, so it suffices to divide Q_h and $\overline{Q}_h$ into two blocks.

Here, we need to choose the parameters appropriately. Let the size of each $S_{h,j}$ and $R_{h,j}$ produced in alternating extraction be roughly d, and the size of each block of Q_h and $\overline{Q}_h$ be n_q. Since in the analysis of each alternating extraction we need to fix $O(t)$ tampered versions of $(S_{h,j}, \overline{S}_{h,j})$ and $(R_{h,j}, \overline{R}_{h,j})$, we need to have $n_q \geq \Theta(td)$. Now in the analysis of the entire non-malleable extractor, we need to fix $O(t)$ tampered versions of Q_h and $\overline{Q}_h$, and $O(t\ell)$ tampered versions of $(S_{h,j}, \overline{S}_{h,j})$. The total size of this is $O(t\ell d)$. Thus

we can take all t, ℓ, d to be some small enough $n^{\Omega(1)}$ such that the total entropy loss of X and Y is some small $n^{\Omega(1)}$. Note that X and Y initially have almost full entropy. Therefore, we can divide X and Y into $O(\ell)$ blocks (or even $O(t\ell)$ blocks, for a reason we will explain below), such that even conditioned on the fixing of all $\{S_{h,j}, R_{h,j}, \overline{S}_{h,j}, \overline{R}_{h,j}, Q_h, \overline{Q}_h\}$ and all previous blocks, each block still has entropy rate say at least 0.9 (this can be achieved as long as $n/(t\ell) \gg t\ell d$). This ensures that each time we apply an extractor, we can use new blocks of X and Y.

Idea 3: However, there is one issue with inverting a linear seeded extractor. The problem is that the pre-image size for different seeds may not be the same. For example, if we have a linear seeded extractor that outputs m bits from an n-bit input, then one can show that for most seeds the pre-image size is 2^{n-m}, while for some seed the pre-image size can be 2^n. If we first generate the seed uniformly and then sample uniformly from the pre-image given each seed, then the overall distribution is not uniform over the entire pre-image, due to the above mentioned size difference. To rectify this, we construct a new linear seeded extractor $\mathrm{iExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $m = d/2$ that works for entropy rate 0.9 sources. Moreover iExt has the property that given any output, for any fixed seed the pre-image size is the same. The idea is as follows. We first take $0.1d$ bits from the seed and use an average sampler to sample $0.9d$ distinct bits from the source. Since we are using a sampler and the source has entropy rate 0.9, an argument in [Vad04] shows that with high probability conditioned on the $0.1d$ bits of the seed, the sampled $0.9d$ bits from the source also has entropy rate roughly 0.9. Now we take the rest 0.9 bits of the seed and the sampled $0.9d$ bits from the source and apply the inner product two-source extractor (or just use leftover hash lemma), which can output $d/2$ uniform random bits. Now the point is that given any output and any fixed seed, the pre-image of the inner product part has the same size,[4] and now the pre-image of any sampled bits also have the same size (since the pre-image is just the sampled bits adding any possible choice of the other $n - 0.9d$ bits).

Note that each time we apply iExt, the output length becomes half of the seed length. Thus in the alternating extraction if we start with seed length d, then after one sub step of alternating extraction, the output length will become $\Omega(d)$ since the sub step takes at most 2 rounds. We will truncate the output if necessary to keep it to be the same length, no matter we choose $R_{h,1}$ or $R_{h,2}$ (since they have different sizes). Now we need to use this output to extract $\overline{Q}_h$ or Q_{h+1} from Y. Since the size of $\overline{Q}_h$ or Q_{h+1} is $\Theta(td)$, we will take $\Theta(t)$ new blocks from Y and apply iExt to them using the *same* seed, and then concatenate the outputs. Since the blocks of Y form a block source, and iExt is a strong seeded extractor, one can show that the concatenated outputs is close to uniform. We then do the same thing for the next sub step of alternating extraction. Since we need to repeat alternating extraction for $O(\ell)$ steps, we need to divide Y into $O(t\ell)$ blocks; while we can divide X into only $O(\ell)$ blocks.

Idea 4: Now given any output, our sampling strategy is as follows. We first uniformly generate X_1 and Y_1, from whom we can compute $V = \mathrm{IP}(X_1, Y_1)$. Then we know which bits of the codeword we are sampling. We then uniformly generate these sampled bits X_2, Y_2 and thus we obtain Z. Once we have Z, we will now uniformly generate all $\{S_{h,j}, R_{h,j}, \overline{S}_{h,j}, \overline{R}_{h,j}\}$ produced in alternating extractions. Based on Z and these variables, we can now generate all the blocks of X used and all the $\{Q_h, \overline{Q}_h\}$ by inverting iExt. Finally, based on $\{Q_h, \overline{Q}_h\}$ we can generate all the blocks of Y used by again inverting iExt.

This almost works except for the following problem. The blocks of X and Y generated must also satisfy the linear equations imposed by X_2, Y_2, which are the bits sampled from the codewords of encodings of X and Y by using a linear error correcting code. However, it is unclear what is the dependence between the linear equations imposed by X_2, Y_2 and the other linear equations that we obtain earlier. Of course, if they are linearly independent then we are in good shape.

To solve this problem, our crucial observation is that if ℓ is small and the number of blocks is large

[4] Except when the seed is 0, but we can deal with this by adding a 1 to both the source and the seed.

enough (say we divide the rest of X into $O(\ell)$ blocks and the rest of Y into $O(t\ell)$ blocks for a large enough constant in $O(\cdot)$), then the entire alternating extraction steps only consume say half of the bits of X and Y. Thus, whatever linear equations we obtain from these steps are only imposing constraints to say the first half bits of X and Y. Therefore, we can hope that the encodings of X and Y use all the bits of X and Y, and thus the linear equations imposed by these encodings will be linearly independent of the equations we obtain from alternating extraction (i.e., the second half bits act as "free variables").

We indeed succeed with this idea. More specifically, we are going to divide the rest of X and Y (the parts excluding X_1 and Y_1, which has length $n - n^{\Omega(1)}$) into chunks of length $b = \lceil \log n \rceil$. We will now view each chunk as an element in the field $\mathbb{F}_{2^b}$. We then take say $0.9n$ bits and view it as a string in $\mathbb{F}^{0.9n/b}$. We can now use Reed-Solomn code (RS-code for short) in $\mathbb{F}_{2^b}$ to encode this string into a codeword in $\mathbb{F}^{2^b}$. Note that $2^b > n > 0.9n/b$, so this encoding is feasible, and it has distance rate $(2^b - 0.9n/b)/(2^b) > 0.9$. Now, instead of using $V = \text{IP}(X_1, Y_1)$ to sample $n^{\Omega(1)}$ bits, we will sample $n^{\Omega(1)}$ field elements from the encoding of X and Y, and then view them as bit strings. Since the RS-code has distance rate 0.9, again we have that if two strings are different, then with probability $1-2^{-n^{\Omega(1)}}$, the sampled strings of their encodings will also be different. Moreover, the sampled bit string now has length roughly $n^{\Omega(1)} \log n$, which is still small enough.

Now we can continue with our sampling strategy. As before we first generate all the blocks of X and Y used in all alternating extraction steps. This only consists of the first half bits of X and Y. Now, any fixing of these bits can be viewed equivalently as fixing the first $0.5n/b$ field elements in a message. Thus we are still left with $0.4n/b$ free field elements, and we have $n^{\Omega(1)}$ linear equations in $\mathbb{F}_{2^b}$ according to the RS-code. As long as the number of free variables is larger than the number of equations (i.e., $0.4n/b > n^{\Omega(1)}$), the property of the RS-encoding ensures that this set of linear equations are linearly independent. Thus, for any fixed first half bits of X and Y, the pre-image according to the linear equations imposed by the sampled bits X_2, Y_2 has the same size.

Summary. Now we are basically done. Again, given any output, our sampling strategy is as follows. We first uniformly generate X_1 and Y_1, from whom we can compute $V = \text{IP}(X_1, Y_1)$. Then we know which co-ordinates of the codeword we are sampling. We then uniformly generate these sampled bits X_2, Y_2 and thus we obtain Z. Once we have Z, we will now uniformly generate all $\{S_{h,j}, R_{h,j}, \overline{S}_{h,j}, \overline{R}_{h,j}\}$ produced in alternating extractions. Based on Z and these variables, we can now generate all the blocks of X used and all the $\{Q_h, \overline{Q}_h\}$ by inverting iExt. Based on $\{Q_h, \overline{Q}_h\}$ we can generate all the blocks of Y used by again inverting iExt. Finally, we use the linear equations imposed by X_2, Y_2 to generate the rest of the bits in X and Y.

To show that we are indeed sampling uniformly from the output's pre-image, we will establish the following two facts.

Fact 1: For any fixed $Z = z$, any choice of $\{s_{h,j}, r_{h,j}, \overline{s}_{h,j}, \overline{r}_{h,j}\}$ gives the same pre-image size of (x, y). This follows directly from the fact that our linear seeded extractor has the same pre-image size for any seed, and the argument about the linear equations imposed by the RS-code above.

Fact 2: For different $Z = z$, and different choice of $\{s_{h,j}, r_{h,j}, \overline{s}_{h,j}, \overline{r}_{h,j}\}$, the pre-image size of (x, y) is also the same. This follows because the "flip-flop" alternating extraction has a symmetric manner. More specifically, no matter each bit of z is 0 or 1, we will use two sub steps of alternating extraction, with each step taking two rounds of alternating extraction. Thus by symmetry no matter each bit of z is 0 or 1, the pre-image size of the blocks of X and $\{q_h, \overline{q}_h\}$ is the same. Moreover, although depending on the h'th bit z, we may choose either $r_{h,1}$ or $r_{h,2}$ (or either $\overline{r}_{h,1}$ or $\overline{r}_{h,2}$), we truncate them if necessary to the same size. So when we generate the blocks of Y using them and $\{q_h, \overline{q}_h\}$, the pre-image of the blocks of Y will also have the same size. Thus, the pre-image size of the blocks of X and Y used for this bit is the same. Therefore, for different $Z = z$ and different $\{s_{h,j}, r_{h,j}, \overline{s}_{h,j}, \overline{r}_{h,j}\}$, the pre-image size is also the same.

Now the conclusion that we are sampling uniformly from the output's pre-image follows from the above two facts, and the observation that any (x, y) in the output's pre-image produces exactly one sequence of $z, \{s_{h,j}, r_{h,j}, \overline{s}_{h,j}, \overline{r}_{h,j}\}$.

3 Preliminaries

3.1 Notations

We use capital letters to denote distributions and their support, and corresponding small letters to denote a sample from the source. Let $[m]$ denote the set $\{1, 2, \ldots, m\}$, and U_r denote the uniform distribution over $\{0,1\}^r$. For a string x of length m, define the string $\text{Slice}(x, w)$ to be the prefix of length w of x. For any $i \in [m]$, let $x_{\{i\}}$ denote the symbol in the i'th co-ordinate of x, and for any $T \subseteq [m]$, let $x_{\{T\}}$ denote the projection of x to the co-ordinates indexed by T.

3.2 Min Entropy, Flat Distributions

The min-entropy of a source X is defined to be $H_\infty(X) = \min_{s \in \text{support}(X)} \{1/\log(\Pr[X = s])\}$. A distribution (source) D is flat if it is uniform over a set S. A (n, k)-source is a distribution on $\{0,1\}^n$ with min-entropy k. It is a well known fact that any (n, k)-source is a convex combination of flat sources supported on sets of size 2^k.

3.3 Statistical Distance, Convex Combination of Distributions and Probability Lemmas

Definition 3.1 (Statistical distance)**.** *Let D_1 and D_2 be two distributions on a set S. The statistical distance between D_1 and D_2 is defined to be:*

$$|D_1 - D_2| = \max_{T \subseteq S} |D_1(T) - D_2(T)| = \frac{1}{2} \sum_{s \in S} |\Pr[D_1 = u] - \Pr[D_2 = u]|$$

D_1 is ϵ-close to D_2 if $|D_1 - D_2| \leq \epsilon$.

Definition 3.2 (Convex combination)**.** *A distribution D on a set S is a convex combination of distributions $D_1, \ldots, D_l$ on S if there exists non-negative constants (called weights) $w_1, \ldots, w_\ell$ with $\sum_{i=1}^{l} w_i = 1$ such that $\Pr[D = s] = \sum_{i=1}^{l} w_i \cdot \Pr[D_i = s]$ for all $s \in S$. We use the notation $D = \sum_{i=1}^{l} w_i \cdot D_i$ to denote the fact that D is a convex combination of the distributions $D_1, \ldots, D_\ell$ with weights $w_1, \ldots, w_\ell$.*

Definition 3.3. *For random variables X and Y, we use $X|Y$ to denote a random variable with distribution: $\Pr[(X|Y) = x] = \sum_{y \in \text{support}(Y)} \Pr[Y = y] \cdot \Pr[X = x|Y = y]$.*

We record the following lemma which follows from the above definitions.

Lemma 3.4. *Let X and Y be distributions on a set S such that $X = \sum_{i=1}^{l} w_i \cdot X_i$ and $Y = \sum_{i=1}^{l} w_i \cdot Y_i$. Then $|X - Y| \leq \sum_i w_i \cdot |X_i - Y_i|$.*

3.4 Seeded and Seedless Extractors

Definition 3.5 (Strong seeded extractor)**.** *A function $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is called a strong seeded extractor for min-entropy k and error ϵ if for any (n, k)-source X and an independent uniformly random string U_d, we have*

$$|\text{Ext}(X, U_d) \circ U_d - U_m \circ U_d| < \epsilon,$$

where U_m is independent of U_d. Further if the function $\text{Ext}(\cdot, u)$ *is a linear function over $\mathbb{F}_2$ for every $u \in \{0,1\}^d$, then* Ext *is called a linear seeded extractor.*

Definition 3.6 (Independent Source Extractor)**.** *A function* $\text{IExt} : (\{0,1\}^n)^t \to \{0,1\}^m$ *is an extractor for independent (n,k) sources that uses t sources and outputs m bits with error ϵ, if for any t independent (n,k) sources $X_1, X_2, \cdots, X_t$, we have*

$$|\text{IExt}(X_1, X_2, \cdots, X_t) - U_m| \leq \epsilon.$$

In the special case where $t = 2$, we say IExt *is a* two-source extractor.

3.5 Conditional Min-Entropy

Definition 3.7. *The average conditional min-entropy is defined as*

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

We recall some results on conditional min-entropy from [DORS08].

Lemma 3.8 ([DORS08])**.** *For any $s > 0$,* $\Pr_{w \leftarrow W}\left[H_\infty(X|W = w) \geq \widetilde{H}_\infty(X|W) - s\right] \geq 1 - 2^{-s}$.

Lemma 3.9 ([DORS08])**.** *If a random variable B can take at most ℓ values, then $\widetilde{H}_\infty(A|B) \geq H_\infty(A) - \ell$.*

It is sometimes convenient to work with average case seeded extractors, where if a source X has average case conditional min-entropy $\tilde{H}_\infty(X|Z) \geq k$ then the output of the extractor is uniform even when Z is given.

Lemma 3.10 ([DORS08])**.** *For any $\delta > 0$, if* Ext *is a (k, ϵ)-extractor then it is also a $\left(k + \log\left(\frac{1}{\delta}\right), \epsilon + \delta\right)$ average case extractor.*

The following result on conditional min-entropy was proved in [MW97].

Lemma 3.11. *Let X, Y be random variables such that the random variable Y takes at ℓ values. Then*

$$\Pr_{y \sim Y}\left[H_\infty(X|Y = y) \geq H_\infty(X) - \log \ell - \log\left(\frac{1}{\epsilon}\right)\right] > 1 - \epsilon.$$

We also need the following lemma from [Li12b].

Lemma 3.12. *Let X, Y be random variables with supports $S, T \subseteq V$ such that (X, Y) is ϵ-close to a distribution with min-entropy k. Further suppose that the random variable Y can take at most ℓ values. Then*

$$\Pr_{y \sim Y}\left[(X|Y = y) \text{ is } 2\epsilon^{1/2}\text{-close to a source with min-entropy } k - \log \ell - \log\left(\frac{1}{\epsilon}\right)\right] \geq 1 - 2\epsilon^{1/2}.$$

3.6 Somewhere Random Sources

Definition 3.13. *A source X is a $t \times k$ somewhere random source if it comprises of t rows on $\{0,1\}^k$ such that at least one of the rows is uniformly distributed. The rows may have arbitrary correlations among themselves.*

3.7 Some Known Extractor Constructions

We use explicit constructions of strong linear seeded extractors [Tre01] [RRV02].

Theorem 3.14 ([Tre01][RRV02])**.** *For every $n, k, m \in \mathbb{N}$ and $\epsilon > 0$ such that $m \leq k \leq n$, there exists an explicit linear strong seeded extractor* $\text{LSExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *for min-entropy k, error ϵ, and* $d = O\left(\frac{\log^2(n/\epsilon)}{\log(k/m)}\right)$.

The following is an explicit construction of a strong seeded extractor with optimal parameters [GUV09].

Theorem 3.15. *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 $d = O(\log n + \log(\frac{1}{\epsilon}))$ and $m = (1-\alpha)k$.*

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

Theorem 3.16 ([Zuc07])**.** *For all constant $\alpha, \delta, \epsilon > 0$ and for all $n > 0$ there exists an efficient construction of a strong seeded extractor* $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *with $m \geq (1-\alpha)n$ and $D = 2^d = O(n)$.*

We recall a folklore construction of a two-source extractors based on the inner product function [CG88]. We include a proof for completeness.

Theorem 3.17 ([CG88])**.** *For all $m, r > 0$, with $q = 2^m, n = rm$, let X, Y be independent sources on $\mathbb{F}_q^r$ with min-entropy k_1, k_2 respectively. Let* IP *be the inner product function over the field $\mathbb{F}_q$. Then, we have:*

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

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

Proof. Let X, Y be uniform on sets $A, B \subseteq \mathbb{F}_q^r$ respectively, with $|A| = 2^{k_1}$ and $|B| = 2^{k_2}$. Let ψ be any non-trivial additive character of the finite field $\mathbb{F}_q$. For short, we use $\cdot$ to denote the standard inner product over $\mathbb{F}_q$. We have

$$\begin{aligned}\sum_{y \in B} |\sum_{x \in A} \psi(x \cdot y)| &\leq (|B|)^{\frac{1}{2}} \left(\sum_{y \in \mathbb{F}_q^r} \sum_{x,x' \in A} \psi((x-x') \cdot y) \right)^{\frac{1}{2}} \\ &\leq |B|^{\frac{1}{2}} \left(\sum_{x,x' \in A} \left(\sum_{y \in \mathbb{F}_q^r} \psi((x-x') \cdot y) \right) \right)^{\frac{1}{2}}\end{aligned}$$

where the first inquality follows by an application of the Cauchy-Schwartz inequality. Further, whenever $x \neq x'$, we have

$$\sum_{y \in \mathbb{F}_q^r} \psi((x-x') \cdot y) = 0.$$

Thus, continuing with our estimate, we have

$$\sum_{y \in B} |\sum_{x \in A} \psi(x \cdot y)| \leq |B|^{\frac{1}{2}} \, (|A| q^r)^{\frac{1}{2}} = 2^{\frac{n+k_1+k_2}{2}}$$

Thus,

$$E_Y|\mathbb{E}_X\psi(\text{IP}(X,Y))| \leq 2^{\frac{n-k_1-k_2}{2}}$$

Using Vazirani's XOR Lemma (see [Rao07] for a proof), it now follows that

$$|\text{IP}(X,Y), Y - U_m, Y| \leq 2^{\frac{n+m-k_1-k_2}{2}}$$

It can be similarly shown that $|\text{IP}(X,Y), X - U_m, X| \leq 2^{\frac{n+m-k_1-k_2}{2}}$. □

4 Seeded and Seedless Non-Malleable Extractors

We give formally introduce seedless $(2,t)$-non-malleable extractors in this section. We first recall the definition of seeded t-non-malleable extractors from [CRS14], which generalizes the definition introduced in [DW09].

Definition 4.1 (t-Non-malleable Extractor)**.** *A function* snmExt : $\{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a seeded t-non-malleable extractor for min-entropy k and error ϵ if the following holds : If X is a source on $\{0,1\}^n$ with min-entropy k and $\mathcal{A}_1 : \{0,1\}^n \to \{0,1\}^n, \ldots, \mathcal{A}_t : \{0,1\}^n \to \{0,1\}^n$ are arbitrary tampering function with no fixed points, then*

$$\begin{aligned}|\text{snmExt}(X,U_d) \circ \text{snmExt}(X,\mathcal{A}_1(U_d)) \circ \ldots \circ \text{snmExt}(X,\mathcal{A}_t(U_d)) \circ U_d \\ -U_m \circ \text{snmExt}(X,\mathcal{A}_1(U_d)) \circ \ldots \circ \text{snmExt}(X,\mathcal{A}_t(U_d)) \circ U_d| < \epsilon\end{aligned}$$

where U_m is independent of U_d and X.

We now proceed to define seedless non-malleable extractors, which were introduced by Cheraghchi and Guruswami in [CG14b].

We need the following functions.

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

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

Definition 4.2 (Seedless Non-Malleable Extractor)**.** *A function* nmExt : $\{0,1\}^n \to \{0,1\}^m$ *is a seedless non-malleable extractor with respect to a class of sources $\mathcal{X}$ and a family of tampering functions $\mathcal{F}$ with error ϵ if for every distribution $X \in \mathcal{X}$ and every tampering function $f \in \mathcal{F}$, there exists a random variable $D_{X,f}$ on $\{0,1\}^m \cup \{same^\star\}$ which is independent of the source X such that*

$$|\text{nmExt}(X) \circ \text{nmExt}(f(X)) - U_m \circ \text{copy}(D_{X,f}, U_m)| \leq \epsilon$$

where both U_m's refer to the same uniform m-bit string.

When the class of tampering functions are 2-split-state, the definition of seedless non-malleable extractors specializes as follows.

Definition 4.3 (Seedless 2-Non-Malleable Extractor)**.** *A function* nmExt : $\{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *is a seedless 2-non-malleable extractor at min-entropy k and error ϵ if it satisfies the following property: If X and Y are independent (n,k)-sources and $\mathcal{A} = (f,g)$ is an arbitrary 2-split-state tampering function, then*

there exists a random variable $D_{f,g}$ on $\{0,1\}^m \cup \{same^\star\}$ which is independent of the sources X and Y, such that

$$|\text{nmExt}(X,Y) \circ \text{nmExt}(\mathcal{A}(X,Y)) - U_m \circ \text{copy}(D_{f,g}, U_m)| < \epsilon$$

where both U_m's refer to the same uniform m-bit string.

In this work, we introduce the following natural generalization where the sources X, Y are tampered by t tampering functions, each of which is from the 2-split-state family.

Definition 4.4 (Seedless (2,t)-Non-Malleable Extractor)**.** *A function* nmExt : $\{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *is a seedless $(2,t)$-non-malleable extractor at min-entropy k and error ϵ if it satisfies the following property: If X and Y are independent (n,k)-sources and $\mathcal{A}_1 = (f_1, g_1), \ldots, \mathcal{A}_t = (f_t, g_t)$ are t arbitrary 2-split-state tampering functions, then there exists a random variable $D_{\vec{f},\vec{g}}$ on $(\{0,1\}^m \cup \{same^\star\})^t$ which is independent of the sources X and Y, such that*

$$|\text{nmExt}(X,Y), \text{nmExt}(\mathcal{A}_1(X,Y)), \ldots, \text{nmExt}(\mathcal{A}_t(X,Y)) - U_m, \text{copy}^{(t)}(D_{\vec{f},\vec{g}}, U_m)| < \epsilon$$

where both U_m's refer to the same uniform m-bit string.

5 Non-malleable codes via Seedless non-malleable extractors

The following theorem is a straightforward generalization of the connection found between non-malleable codes and seedless non-malleable extractors [CG14b].

Theorem 5.1. *Let* nmExt : $(\{0,1\}^n)^2 \to \{0,1\}^m$ *be a polynomial time computable seedless $(2,t)$-non-malleable extractor for min-entropy n with error ϵ. Then there exists a one-many non-malleable code with an efficient decoder in the 2-split-state model with tampering degree t, block length $= 2n$, relative rate $\frac{m}{2n}$, and error $= \epsilon 2^{mt+1}$.*

The one-many non-malleable codes in the 2-split-state model is define in the following way: For any message $s \in \{0,1\}^m$, the encoder $\text{Enc}(s)$ outputs a uniformly random string from the set $\text{nmExt}^{-1}(s) \subset \{0,1\}^{2n}$. For any codeword $c \in \{0,1\}^{2n}$, the decoder Dec outputs $\text{nmExt}(c)$. Thus for the encoder to be efficient, one need to sample almost uniform from $\text{nmExt}^{-1}(s)$.

6 An Explicit Seedless $(2,t)$-Non-Malleable Extractor

We first set up some tools that we use in our extractor construction.

6.1 Averaging Samplers

In our contsruction, we need to pseudorandomly sample a subset T in $[n]$ such that it intersects any large enough subset with high probability. It turns out that a stronger sampling problem has been extensively studied with the following stronger requirement: For any function $f : [n] \to [0,1]$, the average of f on the sampled subset T is close to its actual mean with high probability. Such sampling procedures are known as averaging samplers. We use the definition from [Vad04].

Definition 6.1 (Averaging sampler [Vad04]). *A function* $\text{Samp} : \{0,1\}^r \to [n]^t$ *is a* (μ, θ, γ) *averaging sampler if for every function* $f : [n] \to [0,1]$ *with average value* $\frac{1}{n}\sum_i f(i) \geq \mu$*, it holds that*

$$\Pr_{i_1,\ldots,i_t \leftarrow \text{Samp}(U_R)} \left[\frac{1}{t} \sum_i f(i) \leq \mu - \theta \right] \leq \gamma.$$

Samp *has distinct samples if for every* $x \in \{0,1\}^r$*, the samples produced by* $\text{Samp}(x)$ *are all distinct.*

The following theorem proved by Zuckerman [Zuc97] essentially shows that seeded extractors are equivalent to averaging samplers.

Theorem 6.2 ([Zuc97]). *Let* $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a strong seeded extractor for min-entropy* k *and error* ϵ*. Let* $\{0,1\}^d = \{s_1, \ldots, s_{2^d}\}$*. Then* $\text{Samp}(x) = (\text{Ext}(x, s_1) \circ s_1, \ldots, \text{Ext}(x, s_{2^d}) \circ s_{2^d})$ *is a* (μ, θ, γ) *averaging sampler with distinct samples for any* $\mu > 0$ *,* $\theta = \epsilon$ *and* $\gamma = 2^{k-n}$*.*

Using known constructions of strong seeded extractors, we have the following corollary.

Corollary 6.3. *For any constants* $\delta_{\text{Samp}}, \nu_{\text{Samp}} > 0$*, there exist constants* $\alpha, \beta < \nu_{\text{Samp}}$ *such that for all* $n > 0$ *and any* $r \geq n^\alpha$ *there exists a polynomial time computable function* $\text{Samp} : \{0,1\}^r \to [n]^{t_{\text{Samp}}}$ $t_{\text{Samp}} = O(n^\beta)$ *satisfying the following property: for any set* $S \subset [n]$ *of size* $\delta_{\text{Samp}} n$*,*

$$\Pr[|\text{Samp}(U_r) \cap S| \geq 1] \geq 1 - 2^{-\Omega(n^\alpha)}.$$

Further Samp *has distinct samples.*

Proof. We set the parameter α as follows. Let $\text{Ext} : \{0,1\}^{n^\alpha} \times \{0,1\}^d \to \{0,1\}^m$ be the strong linear seeded extractor for min-entropy $k = \frac{n^\alpha}{2}$ and error $\epsilon = \frac{\delta}{2}$ from Theorem 3.14. Thus $t = 2^d = O(n^{\alpha c})$ for some constant c. We choose $\alpha < \nu_{\text{Samp}}$ small enough such that $c\alpha < \nu_{samp}$ (and set $\beta = c\alpha$). The result now follows by using Theorem 6.2. □

6.2 Alternating Extraction

We recall the method of alternating extraction, which we use as a crucial component in our construction.

The alternating extraction protocol takes in two integer parameters $u, m > 0$. Assume that there are two parties, Quentin with a source Q and a uniform seed S_1 (which may be correlated with Q), and Wendy with source W. Further suppose that (Q, S_1) is kept as a secret from Wendy and W is kept a secret from Quentin.The protocol is an interactive process between Quentin and Wendy, and runs for u steps.

Let Ext_q, Ext_w be strong seeded extractors. In the first step, Quentin sends S_1 to Wendy, Wendy computes $R_1 = \text{Ext}_w(X, S_1)$ and sends it back to Quentin, and Quentin then computes $S_2 = \text{Ext}_q(Q, R_1)$. Continuing in this way, in step i, Quentin sends S_i, Wendy computes the random variables $R_i = \text{Ext}_w(X, S_i)$ and sends it to Quentin, and Quentin then computes the random variable $S_{i+1} = \text{Ext}_q(Q, R_i)$. This is done for u steps. Each of the random variables R_i, S_i is of length m. Thus, the following sequence of random variables is generated:

$$S_1, R_1 = \text{Ext}_w(X, S_1), S_2 = \text{Ext}_q(Q, R_1), \ldots, S_u = \text{Ext}_q(Q, R_{u-1}), R_u = \text{Ext}_w(X, S_u).$$

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

$$\text{laExt}(X, (Q, S_1)) = R_1, \ldots, R_u$$

In our application of the alternating extraction protocol, the initial seed S_1 is not guaranteed to be uniform but only has high min-entropy[5]. We first prove a lemma which shows that strong seeded extractors work even when the seed is not uniform but has high enough min-entropy.

Lemma 6.4. *Let* $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a strong seeded extractor for min-entropy* k*, and error* ϵ*. Let* X *be a* (n,k)*-source and let* Y *be a source on* $\{0,1\}^d$ *with min-entropy* $d-\lambda$*. Then,*

$$|\text{Ext}(X,Y) \circ Y - U_m \circ Y| \leq 2^{\lambda}\epsilon.$$

Proof. Since Y is a source with min-entropy $d-\lambda$, we can assume it is uniform on a set A of size $2^{d-\lambda}$. Thus

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

where the last inequality uses the fact that Ext is a strong seeded extractor. □

Notation: If $Z_a, Z_{a+1}, \ldots, Z_b$ are random variables, we use $Z_{[a,b]}$ to denote the random variable Z_a, $\ldots, Z_b$.

We now prove a general lemma which establishes a strong property satisfied by the alternating extraction protocol. The proof uses ideas from a result proved by Li on alternating extraction [Li13a], and in fact generalizes this result.

Lemma 6.5. *Let* X *be a* (n_w, k_w)*-source and let* $X^{(1)}, \ldots, X^{(t)}$ *be random variables on* $\{0,1\}^{n_w}$ *that are arbitrarily correlated with* X*. Let* $Y = (Q, S_1), Y^{(1)} = (Q^{(1)}, S_1^{(1)}), \ldots, Y^{(t)} = (Q^{(t)}, S_1^{(t)})$ *be arbitrarily correlated random variables that are independent of* $(X, X^{(1)}, X^{(2)}, \ldots, X^{(t)})$*. Suppose that* Q *is a* (n_q, k_q)*-source,* S_1 *is a* $(m, m-\lambda)$*-source,* $Q^{(1)}, \ldots, Q^{(t)}$ *are each on* n_q *bits, and* $S^{(1)}, \ldots, S^{(t)}$ *are each on* m *bits. Let* $\text{Ext}_q, \text{Ext}_w$ *be strong seeded extractors that extract* m *bits at min-entropy* k *with error* ϵ *and seed length* m*. Let* laExt *be the look-ahead extractor for an alternating extraction protocol with parameters* u, m*, with* $\text{Ext}_q, \text{Ext}_w$ *being the strong seeded extractors used by Quentin and Wendy respectively. Let* $\text{laExt}(X, Y) = R_1, \ldots, R_u$ *and for* $j \in [t]$, $\text{laExt}(X^{(j)}, Y^{(j)}) = R_1^{(j)}, \ldots, R_u^{(j)}$*. If* $k_w, k_q \geq k + u(t+1)m + 2\log(\frac{1}{\epsilon})$*, then the following holds for each* $i \in [u]$*:*

$$R_i, R_{[1,i-1]}, R^{(1)}_{[1.i-1]}, \ldots, R^{(t)}_{[1,i-1]}, Q, Q^{(1)}, \ldots, Q^{(t)} \approx_{\epsilon_i} U_m, R_{[1,i-1]}, R^{(1)}_{[1.i-1]}, \ldots, R^{(t)}_{[1,i-1]}, Q, Q^{(1)}, \ldots, Q^{(t)}$$

where $\epsilon_i = O(u\epsilon + 2^{\lambda}\epsilon)$.

Proof. We in fact prove the following stronger claim.

Claim 6.6. *For each* $i \in [u]$ *the following hold:*

$$\begin{aligned}
&R_i, R_{[1,i-1]}, R^{(1)}_{[1,i-1]}, \ldots, R^{(t)}_{[1,i-1]}, S_{[1,i]}, S^{(1)}_{[1,i]}, \ldots, S^{(t)}_{[1,i]}, Q, Q^{(1)}, \ldots, Q^{(t)} \\
\approx_{\epsilon_i} &U_m, R_{[1,i-1]}, R^{(1)}_{[1,i-1]}, \ldots, R^{(t)}_{[1,i-1]}, S_{[1,i]}, S^{(1)}_{[1,i]}, \ldots, S^{(t)}_{[1,i]}, Q, Q^{(1)}, \ldots, Q^{(t)}
\end{aligned}$$

[5] another way to handle this is to use the extractor from [Raz05], but we avoid this to ensure invertibility of the final extractor.

and

$$S_{i+1}, S_{[1,i]}, S^{(1)}_{[1,i]}, \ldots, S^{(t)}_{[1,i]}, R_{[1,i]}, R^{(1)}_{[1,i]}, \ldots, R^{(t)}_{[1,i]}, X, X^{(1)}, \ldots, X^{(t)}$$
$$\approx_{\epsilon_i + 2\epsilon} U_m, S_{[1,i]}, S^{(1)}_{[1,i]}, \ldots, S^{(t)}_{[1,i]}, R_{[1,i]}, R^{(1)}_{[1,i]}, \ldots, R^{(t)}_{[1,i]}, X, X^{(1)}, \ldots, X^{(t)}$$

where $\epsilon_i = 4(i-1)\epsilon + 2^{\lambda}\epsilon$. *Further, conditioned on* $R_{[1,i-1]}, R^{(1)}_{[1,i-1]}, \ldots, R^{(t)}_{[1,i-1]}, S_{[1,i]}, S^{(1)}_{[1,i]}, \ldots, S^{(t)}_{[1,i]}$, *(a)* $(X, X^{(1)}, \ldots, X^{(t)})$ *is independent from* $(Y, Y^{(1)}, \ldots, Y^{(t)})$, *(b)* X, Q *each have average conditional min-entropy at least* $(u-i)(t+1)m + k + 2\log\left(\frac{1}{\epsilon}\right)$ *and (c)* $R_i, R^{(1)}_i, \ldots, R^{(t)}_i$ *are deterministic functions of* $(X, X^{(1)}, \ldots, X^{(t)})$.

Proof. We prove this claim by induction on i.

Let $i = 1$. Since $R_1 = \text{Ext}_w(X, S_1)$, and Ext_w is a strong-seeded extractor, it follows by Lemma 6.4 that $\text{Ext}_w(X, S_1), S_1 \approx_{\epsilon_1} U_m, S_1$, where $\epsilon_1 = 2^{\lambda}\epsilon$. Thus we can fix S_1, and R_1 is still ϵ_1-close to uniform on average. We note that R_1 is a deterministic function of X. Since the random variables $S^{(1)}_1, \ldots, S^{(t)}_1, Q$, $Q^{(1)}, \ldots, Q^{(t)}$ are deterministic functions of $Y, Y^{(1)}, \ldots, Y^{(t)}$ and thus uncorrelated with X, we have

$$R_1, S_1, S^{(1)}_1, \ldots, S^{(t)}_1, Q, Q^{(1)}, \ldots, Q^{(t)} \approx_{\epsilon_1} U_m, S_1, S^{(1)}_1, \ldots, S^{(t)}_1, Q, Q^{(1)}, \ldots, Q^{(t)}.$$

We fix the random variables $S_1, S^{(1)}_1, \ldots, S^{(t)}_1$. By Lemma 3.9, the source Q has average conditional min-entropy at least $k_q - m(t+1) = k + (u-1)m(t+1) + 2\log\left(\frac{1}{\epsilon}\right)$ after this fixing. Using Lemma 3.10 it follows that Ext_q is a $\left(k + \log\left(\frac{1}{\epsilon}\right), 2\epsilon\right)$ strong average case extractor. We also note that $R_1, R^{(1)}_1, \ldots, R^{(t)}_1$ are now deterministic functions of $X, X^{(1)}, \ldots, X^{(t)}$. Thus recalling that $S_2 = \text{Ext}_q(Q, R_1)$, we have $S_2, R_1 \approx_{(2\epsilon+\epsilon_1)} U_m, R_1$, since R_1 is ϵ_1-close to uniform and using the fact that by Lemma 3.10 Ext_w is a $\left(k + \log\left(\frac{1}{\epsilon}\right), 2\epsilon\right)$ strong average case extractor. Thus on fixing R_1, S_2 is $(2\epsilon + \epsilon_1)$-close to U_m on average and is a deterministic function of Y. Since the random variables $R^{(1)}_1, \ldots, R^{(t)}_1$ are deterministic functions of $X, X^{(1)}, \ldots, X^{(t)}$, we thus have

$$S_2, S_1, S^{(1)}_1, S^{(t)}_1, R_1, R^{(1)}_1, \ldots, R^{(t)}_1, X, X^{(1)}, \ldots, X^{(t)}$$
$$\approx_{\epsilon_1 + 2\epsilon} U_m, S_1, S^{(1)}_1, S^{(t)}_1, R_1, R^{(1)}_1, \ldots, R^{(t)}_1, X, X^{(1)}, \ldots, X^{(t)}$$

Further, it still holds that $(X, X^{(1)}, \ldots, X^{(t)})$ is independent from $(Y, Y^{(1)}, \ldots, Y^{(t)})$. This proves the base case of our induction.

Now suppose that the claim is true for i and we will prove it for $i+1$. Fix the random variables $R_{[1,i-1]}, R^{(1)}_{[1,i-1]}, \ldots, R^{(t)}_{[1,i-1]}, S_{[1,i]}, S^{(1)}_{[1,i]}, \ldots, S^{(t)}_{[1,i]}$. By induction hypothesis, it follows that X, Q each have average conditional min-entropy at least $(u-i)m(t+1) + k + 2\log\left(\frac{1}{\epsilon}\right)$ after this fixing. We now fix the random variables $R_i, R^{(1)}_i, \ldots, R^{(t)}_i$ (these random variables are deterministic functions of $X, X^{(1)}$, $\ldots, X^{(t)}$ by induction hypothesis). Thus by Lemma 3.9, the source X has conditional min-entropy at least $(u-i)(t+1)m + k + 2\log\left(\frac{1}{\epsilon}\right) - (t+1)m = (u-i-1)(t+1)m + k + 2\log\left(\frac{1}{\epsilon}\right)$ after this fixing.

Since $S_{i+1} = \text{Ext}_q(Q, R_i)$ is now independent of X and $(\epsilon_i + 2\epsilon)$-close to U_m on average (by induction hypothesis), it follows that $\text{Ext}_w(X, S_{i+1}), S_{i+1} \approx_{\epsilon_i + 4\epsilon} U_m, S_{i+1}$. Thus on fixing S_{i+1}, $R_{i+1} = \text{Ext}_w(X, S_{i+1})$ is $(\epsilon_i + 4\epsilon)$-close to U_m on average, and is a deterministic function of X. We also fix the random variables $S^{(1)}_{i+1}, \ldots, S^{(t)}_{i+1}$. Since we have fixed the random variables $R^{(1)}_i, \ldots, R^{(t)}_i$, thus $S^{(1)}_{i+1}, \ldots, S^{(t)}_{i+1}$ are deterministic functions of $Y, Y^{(1)}, \ldots, Y^{(t)}$. Hence R_{i+1} is still ϵ_{i+1}-close to uniform on average and a

deterministic function of X after this fixing. Thus,

$$\begin{aligned}&R_{i+1}, R_{[1,i]}, R^{(1)}_{[1,i]}, \ldots, R^{(t)}_{[1,i]}, S_{[1,i+1]}, S^{(1)}_{[1,i+1]}, \ldots, S^{(t)}_{[1,i+1]}, Q, Q^{(1)}, \ldots, Q^{(t)}\\ \approx_{\epsilon_{i+1}} &U_m, R_{[1,i]}, R^{(1)}_{[1,i]}, \ldots, R^{(t)}_{[1,i]}, S_{[1,i+1]}, S^{(1)}_{[1,i+1]}, \ldots, S^{(t)}_{[1,i+1]}, Q, Q^{(1)}, \ldots, Q^{(t)}.\end{aligned}$$

The source Q has conditional min-entropy at least $(u-i)(t+1)m + k + 2\log\left(\frac{1}{\epsilon}\right) - (t+1)m = (u-i-1)(t+1)m + k + 2\log\left(\frac{1}{\epsilon}\right)$.

Recall that $S_{i+2} = \text{Ext}_q(Q, R_{i+1})$. Since Ext_q is a $(k+\log\left(\frac{1}{\epsilon}\right), 2\epsilon)$ strong average case extractor, it follows that $\text{Ext}_q(Q, R_{i+1}), R_{i+1} \approx_{\epsilon_{i+2}+2\epsilon} U_m$. Since the random variables $R^{(1)}_{i+1}, \ldots, R^{(t)}_{i+1}$ are deterministic functions of $X, X^{(1)}, \ldots, X^{(t)}$ (recall that we have fixed $S^{(1)}_{i+1}, \ldots, S^{(t)}_{i+1}$), it follows that

$$\begin{aligned}&S_{i+2}, S_{[1,i+1]}, S^{(1)}_{[1,i+1]}, \ldots, S^{(t)}_{[1,i+1]}, R_{[1,i+1]}, R^{(1)}_{[1,i+1]}, \ldots, R^{(t)}_{[1,i+1]}, X, X^{(1)}, \ldots, X^{(t)}\\ \approx_{\epsilon_{i+1}+2\epsilon} &U_m, S_{[1,i+1]}, S^{(1)}_{[1,i+1]}, \ldots, S^{(t)}_{[1,i+1]}, R_{[1,i+1]}, R^{(1)}_{[1,i+1]}, \ldots, R^{(t)}_{[1,i+1]}, X, X^{(1)}, \ldots, X^{(t)}.\end{aligned}$$

Also, we maintain at each step that $(X, X^{(1)}, \ldots, X^{(t)})$ is independent from $(Y, Y^{(1)}, \ldots, Y^{(t)})$. This completes the proof. □

□

Remark 6.7. *We note that if instead of using a strong seeded extractor to generate R_1 (recall $R_1 = \text{Ext}_w(X, S_1)$), we used the extractor constructed by Raz [Raz05], then the error achieved is $O(u\epsilon)$.*

6.3 Construction of Some Key Components

In this section, we construct functions which are key ingredients in all our explicit extractor constructions. It is based on a new way of using the technique of alternating extraction, and is inspired by a recent elegant work of Cohen [Coh15] on constructing local correlation breakers.

We define the following function which is inspired by the "flip-flop" method introduced by Cohen [Coh15].

We now prove the following lemma.

Lemma 6.8. *Let $b, \{b^{(h)} : h \in [j]\}$ be $j+1$ bits such that for all $h \in [j]$, $b \neq b^{(h)}$. Let X be a (n_w, k_w)-source and let $\{X^{(h)} : h \in [j]\}$ be random variables on $\{0,1\}^{n_w}$ that are arbitrarily correlated with X. Let $Y, \{Y^{(h)} : h \in [j]\}$ be arbitrarily correlated random variables that are independent of $(X, \{X^{(h)} : h \in [j]\})$. Suppose that Y is a (n_y, k_y)-source, $k_y = n_y - \lambda$, each random variable in $\{Y^{(h)} : h \in [j]\}$ is on n_y bits. Let Q_i be some function of Y on n_q bits with min-entropy at least $n_q - \lambda$, and for each $h \in [j]$, let $Q^{(h)}$ be an an arbitrary function of $Y, \{Y^{(a)} : a \in [j]\}$ on n_q bits.*

Let 2laExt *be the function computed by Algorithm* 1. *Let* $\text{2laExt}(X, Y, Q_i, b) = Q_{i+1}$, *and for* $h \in [j]$, *let* $\text{2laExt}(X^{(h)}, Y^{(h)}, Q^{(h)}_i, b^{(h)}) = Q^{(h)}_{i+1}$. *Suppose* $k_y \geq \max\{k, k_1\} + 10\left(jn_q + jm + \log\left(\frac{1}{\epsilon}\right)\right)$, $k_w \geq k + 10\left(jm + \log\left(\frac{1}{\epsilon}\right)\right)$, *and* $n_q \geq k + 10jm + 2\log(\frac{1}{\epsilon}) + \lambda$.

Then with probability at least $1 - \epsilon'$, where $\epsilon' = O(2^{\lambda}\epsilon)$, over the fixing of the random variables Q_i, $\{Q^{(h)}_i : h \in [j]\}, R_{i,1}, R_{i,2}, \{R^{(h)}_{i,1}, R^{(h)}_{i,2} : h \in [j]\}, \overline{Q}_i, \{\overline{Q}^{(h)}_i : h \in [j]\}, \overline{R}_{i,1}, \overline{R}_{i,2}, \{\overline{R}^{(h)}_{i,1}, \overline{R}^{(h)}_{i,2} : h \in [j]\}$, $\{Q^{(h)}_{i+1} : h \in [j]\}$: (a) Q_{i+1} is ϵ'-close to U_{n_q} and is a deterministic function of Y (b) The random variables $(X, \{X^{(h)} : h \in [j]\})$ and $(Y, \{Y^{(h)} : h \in [j]\})$ are independent (c) X has min-entropy at least $k_w - 10\left(jm + \log\left(\frac{1}{\epsilon}\right)\right)$ and Y has min-entropy at least $k_y - 10\left(jn_q + jm + \log\left(\frac{1}{\epsilon}\right)\right)$.

Algorithm 1: 2laExt(x, y, q_i, b)

Input: Bit strings x, y, q_i of length n_w, n_y, n_q respectively, and a bit b.
Output: A bit string of length n_q.
Subroutine: Let $\text{Ext}_q : \{0,1\}^{n_q} \times \{0,1\}^m \to \{0,1\}^m$ be a strong seeded extractor set to extract from min-entropy k with error ϵ and seed length m. Let $\text{Ext}_w : \{0,1\}^{n_w} \times \{0,1\}^m \to \{0,1\}^m$ be a strong seeded extractor set to extract from min-entropy k with error ϵ and seed length d.
Let laExt : $\{0,1\}^{n_w} \times \{0,1\}^{n_q+m} \to \{0,1\}^{2m}$ be the look ahead extractors defined in Section 6.2 for an alternating extraction protocol with parameters $m, u = 2$ (recall u is the number of steps in the protocol, m is the length of each random variable that is communicated between the players), and using $\text{Ext}_q, \text{Ext}_w$ as the strong seeded extractors.
Let Ext : $\{0,1\}^{n_y} \times \{0,1\}^m \to \{0,1\}^{n_q}$ be a strong seeded extractor set to extract from min-entropy k_1 with error ϵ.

Let $s_{i,1} = \text{Slice}(q_i, m)$
Let $\text{laExt}(x, (q_i, s_{i,1})) = r_{i,1}, r_{i,2}$
if $b = 0$, let $\overline{q_i} = \text{Ext}(y, r_{i,1})$
    else let $\overline{q_i} = \text{Ext}(y, r_{i,2})$
endif
Let $\overline{s_{i,1}} = \text{Slice}(\overline{q_i}, m)$.
Let $\text{laExt}(x, (\overline{q_i}, \overline{s_{i,1}})) = \overline{r_{i,1}}, \overline{r_{i,2}}$.
if $b = 0$, let $q_{i+1} = \text{Ext}(y, \overline{r_{i,2}})$
    else let $q_{i+1} = \text{Ext}(y, \overline{r_{i,1}})$
endif
Ouput q_{i+1}.

Proof. **Notation:** For any function H, if $V = H(X, Y)$, let $V^{(a)}$ denote the random variable $H(X^{(a)}, Y^{(a)})$.

We split the proof into two cases, depending on b.

Case 1: Suppose $b = 1$. By Lemma 6.5, it follows that

$$R_{i,2}, \{R_{i,1}^{(h)} : h \in [j]\}, Q_i, \{Q_i^{(h)} : h \in [j]\}$$
$$\approx_{\epsilon_1} U_m, \{R_{i,1}^{(h)} : h \in [j]\}, Q_i, \{Q_i^{(h)} : h \in [j]\}$$

, where $\epsilon_1 = c2^{\lambda}\epsilon$, for some constant c. Thus, we can fix $\{R_{i,1}^{(h)} : h \in [j]\}, Q_i, \{Q_i^{(h)} : h \in [j]\}$, and with probability at least $1 - O(\epsilon_1)$, $R_{i,2}$ is $O(\epsilon_1)$-close to U_m. Note that $R_{i,2}$ is now a deterministic function of X. Further, by Lemma 3.11, Y loses min-entropy at most $(j+1)n_q + \log\left(\frac{1}{\epsilon}\right)$ with probability at least $1 - \epsilon$ due to this fixing. Since on fixing $Q_i, \{Q_i^{(h)} : h \in [j]\}$, the random variables $\{R_{i,1}^{(h)} : h \in [j]\}$ are deterministic function of $X, \{X^{(h)} : h \in [j]\}$, the source X loses min-entropy at most $jm + \log\left(\frac{1}{\epsilon}\right)$ with probability at least $1 - \epsilon$ due to this fixing. We now note that the random variables $\{\overline{Q}_i^{(h)} : h \in [j]\}$ are deterministic functions of $Y, \{Y^{(h)} : h \in [j]\}$. Thus, we fix $\{\overline{Q}_i^{(h)} : h \in [j]\}$, and by Lemma 3.11, Y loses min-entropy at most $jn_q + \log\left(\frac{1}{\epsilon}\right)$ with probability at least $1 - \epsilon$ due to this fixing. Since Ext extracts from min-entropy k_1, and k_y was chosen large enough, it follows that the random variable $\overline{Q}_i$ is $(\epsilon + \epsilon_1)$-close to U_{n_q} with probability at least $1 - O(\epsilon_1)$ even after the fixing. Further, we fix $R_{i,2}$ since Ext is a strong seeded extractor, and by Lemma 3.11, X loses min-entropy at most $m + \log\left(\frac{1}{\epsilon}\right)$ with probability at least $1 - \epsilon$ due to this fixing. Thus $\overline{Q}_i$ is now a deterministic function of Y. We now fix the random variables

$\{R^{(h)}_{i,2} : h \in [j]\}$, noting that they are deterministic functions of X and hence does not affect the distribution of $\overline{Q}_i$. X loses min-entropy at most $jm + \log\left(\frac{1}{\epsilon}\right)$ with probability at least $1-\epsilon$ due to this fixing.

We now note that the random variables $\{\overline{R}^{(h)}_{i,1}, \overline{R}^{(h)}_{i,2} : h \in [j]\}$ are deterministic function of $X, \{X^{((j))} : j \in [h]\}$ since we have fixed $\{\overline{Q}^{(h)}_i : h \in [j]\}$. Thus, we can fix $\{\overline{R}^{(h)}_{i,1}, \overline{R}^{(h)}_{i,2} : h \in [j]\}$ and X loses min-entropy at most $2jm + \log\left(\frac{1}{\epsilon}\right)$ with probability at least $1-\epsilon$. Thus it follows by Lemma 6.5 that $|\overline{R}_{i,1}, \overline{Q}_i - U_m, \overline{Q}_i| < \epsilon + O(\epsilon_1)$. We fix $\overline{Q}_i$ and Y loses min-entropy at most $n_q + \log\left(\frac{1}{\epsilon}\right)$ using Lemma 3.11. Finally, we note that $\{Q^{(h)}_{i+1} : h \in [j]\}$ is now a deterministic function of $Y, \{Y^{(h)} : h \in [j]\}$. Thus, we can fix $\{Q^{(h)}_{i+1} : h \in [j]\}$ variables and Y loses min-entropy at most $jn_q + \log\left(\frac{1}{\epsilon}\right)$ with probability at least $1-\epsilon$ due to this fixing. Further, $\overline{R}_{i,1}$ is now a deterministic function of X. It follows that Q_{i+1} is $O(\epsilon_1+\epsilon)$-close to U_{n_q} since k_y is chosen large enough. We further fix $\overline{R}_{i,1}$ noting that Ext is a strong extractor and X loses min-entropy at most $m + \log\left(\frac{1}{\epsilon}\right)$ with probability at least $1-\epsilon$ due to this fixing.

Case 2: Now suppose $b = 0$. We fix the random variables $Q_i, \{Q^{(h)}_i : h \in [j]\}$. Conditioned on this fixing, it follows by Lemma 6.5 that $|R_{i,1} - U_m| < \epsilon_1$, $\epsilon_1 = O(2^{\lambda}\epsilon)$, with probability at least $1-\epsilon$. Since Ext is a strong seeded extractor (and k_y is large enough) and $R_{i,1}$ is a deterministic function of X, it follows that $|\overline{Q}_i, R_{1,i} - U_{n_q}, R_{i,1}| < \epsilon + \epsilon_1$ with probability at least ϵ. We fix $R_{i,1}$, and observe that $\overline{Q}_i$ is now a deterministic function of Y. We can now fix $\{R^{(h)}_{i,1}, R^{(h)}_{i,2} : h \in [j]\}$ since $\{R^{(h)}_{i,2} : h \in [j]\}$ is a deterministic function of $X, \{X^{(h)} : h \in [j]\}$, and hence does not affect the distribution of $\overline{Q}_i$. As a result of these fixings, it is clear that $(X, \{X^{(h)} : h \in [j]\})$ is independent of $(Y_i, \{Y^{(h)} : h \in [j]\})$. Further X loses min-entropy of at most $2(j+1)m + \log\left(\frac{1}{\epsilon}\right)$ with probability at least $1-\epsilon$, and Y loses min-entropy of at most $2(j+1)n_q + (j+1)m + 3\log\left(\frac{1}{\epsilon}\right)$ with probability at least $1-3\epsilon$. Note that now $\overline{Q}_i, \{Q^{(h)}_i : h \in [j]\}$ are deterministic functions of $Y, \{Y^{(h)} : h \in [j]\}$, and $\overline{Q}_i$ is $O(\epsilon_1)$-close to U_{n_q}. By Lemma 6.5, it follows that

$$\overline{R}_{i,2}, \{\overline{R}^{(h)}_{i,1} : h \in [j]\}, \overline{Q}_i, \{\overline{Q}^{(h)}_i : h \in [j]\} \approx_{\epsilon_2} U_m, \{\overline{R}^{(h)}_{i,1} : h \in [j]\}, \overline{Q}_i, \{\overline{Q}^{(h)}_i : h \in [j]\}$$

where $\epsilon_2 = c(\epsilon_1 + \epsilon + \epsilon)$, for some constant c. Thus, we can fix $\{\overline{R}^{(h)}_{i,1} : h \in [j]\}, \overline{Q}_i, \{\overline{Q}^{(h)}_i : h \in [j]\}$ and with probability at least $1 - O(\epsilon_2)$, $\overline{R}_{i,2}$ is $O(\epsilon_2)$-close to U_m. Note that $\overline{R}_{i,2}$ is now a deterministic function of X. Further, by Lemma 3.11, Y loses min-entropy at most $(j+1)n_q + \log\left(\frac{1}{\epsilon}\right)$ with probability at least $1-\epsilon$ due to this fixing. Since on fixing $\overline{Q}_i, \{\overline{Q}^{(h)}_i : h \in [j]\}$, the random variables $\{\overline{R}^{(h)}_{i,1} : h \in [j]\}$ are deterministic functions of $X, \{X^{(h)} : h \in [j]\}$, the source X loses min-entropy at most $jm + \log\left(\frac{1}{\epsilon}\right)$ with probability at least $1-\epsilon$ due to this fixing. We now note that the random variables $\{Q^{(h)}_{i+1} : h \in [j]\}$ are deterministic functions of $Y, \{Y^{(h)} : h \in [j]\}$. Thus, we fix $\{Q^{(h)}_{i+1} : h \in [j]\}$ and by Lemma 3.11, Y loses min-entropy at most $(j+1)n_q + \log\left(\frac{1}{\epsilon}\right)$ with probability at least $1-\epsilon$ due to this fixing. Since Ext extracts from min-entropy k_1, (and k_y is large enough) it follows that random variable Q_{i+1} is $O(\epsilon_2)$-close to U_{n_q} even after the fixing. Further, we fix $\overline{R}_{i,2}$ since Ext is a strong seeded extractor, and by Lemma 3.11, X loses min-entropy $m + \log\left(\frac{1}{\epsilon}\right)$ with probability at least $1-\epsilon$ due to this fixing. Further Q_{i+1} is now a deterministic function of Y. Thus we can fix the random variables $\{\overline{R}^{(h)}_{i,2} : h \in [j]\}$ since they are deterministic function sod X and does not affect the distribution of Q_{i+1}. X loses min-entropy at most $m + \log\left(\frac{1}{\epsilon}\right)$ with probability at least $1-\epsilon$ due to this fixing. This completes the proof. □

We now construct a function that is a crucial ingredient in our non-malleable extractor constructions. (Recall that for any string z, we use $z_{\{h\}}$ to denote the symbol in the h'th co-ordinate of z.)

Lemma 6.9. *Let $z, z^{(1)}, \ldots, z^{(t)}$ each be ℓ bit strings such that for all $i \in [t]$, $z \neq z^{(i)}$. Let X be a (n_w, k_w)-source and let $X^{(1)}, \ldots, X^{(t)}$ be random variables on $\{0,1\}^{n_w}$ that are arbitrarily correlated with X.*

Algorithm 2: $\text{nmExt}_1(x, y, z)$

Input: Bit strings x, y, z of length n_w, n_y, ℓ respectively.
Output: A bit string of length n_q.

Let $q_1 = \text{Slice}(y, n_q)$
for $h = 1$ *to* ℓ **do**
  $q_{h+1} = \text{2laExt}(x, y, q_h, z_{\{h\}})$
end
Ouput $q_{\ell+1}$.

Let $Y, Y^{(1)}, , \ldots, Y^{(t)}$ be random variables on n_y bits that are independent of $(X, X^{(1)}, X^{(2)}, \ldots, X^{(t)})$. Suppose that Y is a (n_y, k_y)-source, $k_y = n_y - \lambda$.

Let nmExt_1 *be the function computed by Algorithm* 2. *Let* $\text{nmExt}_1(X, Y, z) = Q_{\ell+1}$, *and for* $h \in [t]$, *let* $\text{nmExt}_1(X^{(h)}, Y^{(h)}, z^{(h)}) = Q^{(h)}_{\ell+1}$. *Suppose* $k_y \geq \max\{k, k_1\} + 20\ell\left(tn_q + tm + \log\left(\frac{1}{\epsilon}\right)\right)$, $k_w \geq k + 20\ell\left(tm + \log\left(\frac{1}{\epsilon}\right)\right)$ *and* $n_q \geq k + 10tm + 2\log(\frac{1}{\epsilon}) + \lambda$. *Then, we have*

$$Q_{\ell+1}, Q^{(1)}_{\ell+1}, \ldots, Q^{(t)}_{\ell+1} \approx_{\epsilon'} U_{n_q}, Q^{(1)}_{\ell+1}, \ldots, Q^{(t)}_{\ell+1}$$

where $\epsilon'_\ell = O((2^\lambda + \ell)\epsilon)$.

Proof. **Notation:** For any function H, if $V = H(X, Y)$, let $V^{(a)}$ denote the random variable $H(X^{(a)}, Y^{(a)})$.

For $h \in [\ell]$, define the sets

$$\text{Ind}_h = \{i \in [t] : z_{\{h\}} \neq z^{(i)}_{\{h\}}\}, \qquad \overline{\text{Ind}}_h = [t] \setminus \text{Ind}_h,$$

$$\text{Ind}_{[h]} = \cup_{i=1}^{h} \text{Ind}_h, \qquad \overline{\text{Ind}}_{[h]} = [t] \setminus \text{Ind}_{[h]}.$$

We record a simple claim.

Claim 6.10. *For each $i \in [t]$, there exists $h \in [\ell]$ such that $i \in \text{Ind}_h$.*

Proof. Recall that we have fixed $Z, Z^{(1)}, \ldots, Z^{(t)}$ such that $Z \neq Z^{(i)}$ for any $i \in [t]$. Thus it follows that for each $i \in [t]$, there exists some $h \in [\ell]$ such that $Z_{\{h\}} \neq Z^{(i)}_{\{h\}}$, and hence $i \in \text{Ind}_h$. □

We now prove our main claim, which combined with Lemma 6.8 and a simple inductive argument proves Lemma 6.9.

Claim 6.11. *For any $h \in \{0, 1, \ldots, \ell\}$, suppose the following holds:*

With probability at least $1 - \epsilon_h$ over the fixing of the random variables $\{Q_i : i \in [h]\}, \{Q^{(j)}_i : i \in [h], j \in [t]\}, \{R_{i,1}, R_{i,2} : i \in [h]\}, \{R^{(j)}_{i,1}, R^{(j)}_{i,2} : i \in [h], j \in [t]\}, \{\overline{Q}_i : i \in [h]\}, \{\overline{Q}^{(j)}_i : i \in [h], j \in [t]\}, \{\overline{R}_{i,1}, \overline{R}_{i,2} : i \in [h]\}, \{\overline{R}^{(j)}_{i,1}, \overline{R}^{(j)}_{i,2} : i \in [h], j \in [t]\}, \{Q^{(j)}_{i+1} : j \in \text{Ind}_{[h]}\}$: (a) Q_{h+1} is ϵ_h-close to a source with min-entropy at least $n_q - \lambda$ and is a deterministic function of Y (b) $\{Q^{(j)}_{h+1} : j \in \overline{\text{Ind}}_{[h]}\}$ is a deterministic function of $Y, \{Y^{(j)} : j \in [t]\}$ (c) The random variables $(X, \{X^{(j)} : j \in [t]\})$ and $(Y, \{Y^{(j)} : j \in [t]\})$ are independent (d) X has min-entropy at least $k_w - 10h\left(tm + \log\left(\frac{1}{\epsilon}\right)\right) > k + 10\left(tm + \log\left(\frac{1}{\epsilon}\right)\right)$ and Y has min-entropy at least $k_y - 10h\left(tn_q + tm + \log\left(\frac{1}{\epsilon}\right)\right) > \max\{k, k_1\} + 10\left(tn_q + tm + \log\left(\frac{1}{\epsilon}\right)\right)$.

Then, the following holds:

Let $\epsilon_{h+1} = \epsilon_h + c2^\lambda\epsilon$ for some constant c. With probability at least $1 - \epsilon_{h+1}$ over the fixing of the random variables $\{Q_i : i \in [h+1]\}, \{Q_i^{(j)} : i \in [h+1], j \in [t]\}, \{R_{i,1}, R_{i,2} : i \in [h+1]\}, \{R_{i,1}^{(j)}, R_{i,2}^{(j)} : i \in [h+1], j \in [t]\}, \{\overline{Q}_i : i \in [h+1]\}, \{\overline{Q}_i^{(j)} : i \in [h+1], j \in [t]\}, \{\overline{R}_{i,1}, \overline{R}_{i,2} : i \in [h]\}, \{\overline{R}_{i,1}^{(j)}, \overline{R}_{i,2}^{(j)} : i \in [h+1], j \in [t]\}, \{Q_{i+1}^{(j)} : j \in \text{Ind}_{[h+1]}\}$: (a) Q_{h+2} is ϵ_{h+1}-close to U_{n_q} and is a deterministic function of Y (b) $\{Q_{h+2}^{(j)} : j \in \overline{\text{Ind}}_{[h+1]}\}$ is a deterministic function of $Y, \{Y^{(j)} : j \in [t]\}$ (c) The random variables $(X, \{X^{(j)} : j \in [t]\})$ and $(Y, \{Y^{(j)} : j \in [t]\})$ are independent (d) X has min-entropy at least $k_w - 10(h+1)\left(tm + \log\left(\frac{1}{\epsilon}\right)\right)$ and Y has min-entropy at least $k_y - 10(h+1)\left(tm + \log\left(\frac{1}{\epsilon}\right)\right)$.

Proof. We fix the random variables $\{Q_i : i \in [h]\}, \{Q_i^{(j)} : i \in [h], j \in [t]\}, \{R_{i,1}, R_{i,2} : i \in [h]\}, \{R_{i,1}^{(j)}, R_{i,2}^{(j)} : i \in [h], j \in [t]\}, \{\overline{Q}_i : i \in [h]\}, \{\overline{Q}_i^{(j)} : i \in [h], j \in [t]\}, \{\overline{R}_{i,1}, \overline{R}_{i,2} : i \in [h]\}, \{\overline{R}_{i,1}^{(j)}, \overline{R}_{i,2}^{(j)} : i \in [h], j \in [t]\}, \{Q_{i+1}^{(j)} : j \in \text{Ind}_{[h]}\}$ such that (a), (b), (c), (d) holds (this happens with probability at least $1 - \epsilon_h$. We also fix the random variables $\{R^{(j)}_{h+1,\psi_1(z_{h+1}^{(j)})} : j \in \text{Ind}_{[h]}\}$, noting that they are deterministic functions of X. Thus X has min-entropy at least $k_w - 10h\left(jm + \log\left(\frac{1}{\epsilon}\right)\right) - tm - \log\left(\frac{1}{\epsilon}\right)$ with probabilitiy at least $1 - \epsilon$. Further, Q has min-entropy at least $k_y - 10h\left(tn_q + tm + \log\left(\frac{1}{\epsilon}\right)\right)$. The claim now follows directly from Lemma 6.8. □

To complete the proof of Lemma 6.9, we now note that the hypothesis of Claim 6.11 is indeed satisfied when $h = 0$. Thus, by ℓ applications of Claim 6.11, it follows that the $Q_{\ell+1}$ is ϵ'_ℓ-close to U_{n_q}, where $\epsilon'_\ell = O(2^\lambda\epsilon + \ell\epsilon)$. This follows since for all applications of Claim 6.11 except the first time, Q_h is ϵ_h-close to uniform, and hence the parameter $\lambda = 0$. This concludes the proof of Lemma 6.9. □

6.4 An Explict Seedless $(2, t)$-Non-Malleable Extractor Construction

We are now ready to present our construction. We first set up the various ingredients developed so far with appropriate parameters.

Subroutines and Parameters

1. Let γ be a small enough constant and C a large one. Let $t = n^{\gamma/C}$.
2. Let $n_1 = n^{\beta_1}$, $\beta_1 = 10\gamma$. Let IP : $\{0,1\}^{n_1} \times \{0,1\}^{n_1} \to \{0,1\}^{n_2}$, $n_2 = \frac{n_1}{10}$, be the strong two-source extractor from Theorem 3.17.
3. Let $\mathcal{C}$ be an explicit $[\frac{n}{\alpha}, n, \frac{1}{10}]$-binary linear error correcting code with encoder $E : \{0,1\}^n \to \{0,1\}^{\frac{n}{\alpha}}$. Such explicit codes are known, for example from the work of Alon et al. [ABN+92].
4. Let Samp : $\{0,1\}^{n_2} \to \left[\frac{n}{\alpha}\right]$ be the sampler from Corollary 6.3 with parameters $\delta_{\text{Samp}} = \frac{1}{10}$ and $\nu_{\text{Samp}} = \beta_1$. Let the number of samples $t_{\text{Samp}} = n^{\beta_2}$. Thus, $\beta_2 \le \beta_1$.
5. Let $\ell = 2(n^{\beta_1} + n^{\beta_2})$. Thus $\ell \le n^{11\gamma}$.
6. We set up the parameters for the components used by 2laExt (computed by Algorithm 1) as follows.
   (a) Let $n_3 = n^{\beta_3}, n_4 = n^{\beta_4}$, with $\beta_3 = 100\gamma$ and $\beta_4 = 50\gamma$.
   Let $\text{Ext}_q : \{0,1\}^{n_3} \times \{0,1\}^{n_4} \to \{0,1\}^{n_4}$ be the strong seeded linear extractor from Theorem 3.14 set to extract from min-entropy $k_q = \frac{n_3}{4}$ with error $\epsilon = 2^{-\Omega(n^{\gamma_q})}$, $\gamma_q = \frac{\beta_4}{2}$. Thus, by Theorem 3.14, we have that the seed length $d_q = O\left(\frac{\log^2(n_3/\epsilon)}{\log(k_q/n_4)}\right) = O(n^{2\gamma_q}) = n_4$.

Let $\text{Ext}_w : \{0,1\}^n \times \{0,1\}^{n_4} \to \{0,1\}^{n_4}$ be the strong linear seeded extractor from Theorem 3.14 set to extract from min-entropy $k_w = \frac{n}{2}$ with error $\epsilon = 2^{-\Omega(n^{\gamma_q})}$.

(b) Let $\text{laExt} : \{0,1\}^n \times \{0,1\}^{n_3} \to \{0,1\}^{2n_4}$ be the look ahead extractor used by 2laExt (recall that the parameters in the alternating extraction protocol are set as $m = n_4, u = 2$ where u is the number of steps in the protocol, m is the length of each random variable that is communicated between the players, and $\text{Ext}_q, \text{Ext}_w$ are the strong seeded extractors used in the protocol.).

(c) Let $\text{Ext} : \{0,1\}^n \times \{0,1\}^{n_4} \to \{0,1\}^{n_3}$ be the linear strong seeded extractor from Theorem 3.14 set to extract from min-entropy $\frac{n}{2}$ with seed length n_4 and error $2^{-\Omega(n^{\beta_4/2})}$.

7. Let nmExt_1 be the function computed by Algorithm 2, which uses the function 2laExt set up as above.

Algorithm 3: nmExt(x,y)

Input: Bit strings x, y, each of length n.
Output: A bit string of length n_4.

1 Let $x_1 = \text{Slice}(x, n_1)$, $y_1 = \text{Slice}(y, n_1)$. Compute $v = \text{IP}(x, y)$.
2 Compute $T = \text{Samp}(v) \subset [\frac{n}{\alpha}]$.
3 Let $z = x_1 \circ x_2 \circ y_1 \circ y_2$ where $x_2 = (E(x))_{\{T\}}, y_2 = (E(y))_{\{T\}}$.
4 Output $\text{nmExt}_1(x, y, z)$.

We now state our main theorem.

Theorem 6.12. *Let* nmExt *be the function computed by Algorithm* 3*. Then* nmExt *is a seedless* $(2, t)$*-non-malleable extractor with error* $2^{-n^{\Omega(1)}}$.

We establish the following two lemmas, from which the above theorem is direct.

Lemma 6.13. nmExt $: \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{n_4}$ *satisfies the following property* $\mathcal{P}_n$*: If* X, Y *are independent* $(n, n - n^\gamma)$*-sources and* $\mathcal{A}_1 = (f_1, g_1), \ldots, \mathcal{A}_t = (f_t, g_t)$ *are arbitrary* 2*-split-state tampering functions, such that for each* $i \in [t]$*, at least one of* f_i, g_i *has no fixed points, then the following holds:*

$$|\text{nmExt}(X, Y), \text{nmExt}(\mathcal{A}_1(X, Y)), \ldots, \text{nmExt}(\mathcal{A}_t(X, Y)) - U_{n_4}, \text{nmExt}(\mathcal{A}_1(X, Y)), \ldots, \text{nmExt}(\mathcal{A}_t(X, Y))| \leq \epsilon,$$

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

Lemma 6.14. *Suppose* nmExt $: \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{n_4}$*, satisfies property* $\mathcal{P}_n$ *(from Lemma* 6.13*). Then,* nmExt *is a seedless* $(2, t)$*-non-malleable extractor with error* $(2^{-n^\gamma} + \epsilon)2^{2t}$.

Notation: For any function H, if $V = H(X, Y)$, let $V^{(i)}$ denote the random variable $H(\mathcal{A}_i(X, Y))$.

Proof of Lemma 6.13. We begin by proving the following claim.

Claim 6.15. *With probability at least* $1 - 2^{-n^{\Omega(1)}}$*,* $Z \neq Z^{(i)}$ *for each* $i \in [t]$.

Proof. Pick an arbitrary $i \in [t]$. Without loss of generality, suppose f_i has no fixed points. If $X_1 \neq X_1^{(i)}$ or $Y_1 \neq Y_1^{(i)}$, then $Z \neq Z^{(i)}$. Now suppose $X_1 = X_1^{(i)}$ and $Y_1 = Y_1^{(i)}$. We fix X_1, and note that since IP is a strong extractor (Theorem 3.17), V is $2^{-\Omega(n_1)}$-close to U_{n_2} after this fixing (with probability at least $1 - 2^{-\Omega(n_1)}$). Also note that $V = V^{(i)}$.

Since f_i has no fixed points, it follows that since E is an encoder of a code with relative distance distance $\frac{1}{10}$, $\Delta(E(X), E(X^{(i)})) \geq \frac{n}{10\alpha}$. Let $D = \{j \in \left[\frac{n}{\alpha}\right] : E(X)_{\{j\}} \neq E(X^{(i)})_{\{j\}}\}$. Thus $|D| \geq \frac{n}{10\alpha}$. Using Corollary 6.3, it follows that with probability at least $1 - 2^{-\Omega(n_1)}$, $|D \cap \text{Samp}(V)| \geq 1$, and thus $X_2 \neq X_2^{(i)}$ (since $\text{Samp}(V) = \text{Samp}(V^{(i)})$). This proves the claim. □

We fix $Z, Z^{(1)}, \ldots, Z^{(t)}$ such that $Z \neq Z^{(i)}$ for any $i \in [t]$ (from the lemma above, this occurs with probability $1 - 2^{-n^{\Omega(1)}}$). We note that by the Lemma 6.15 and Lemma 3.11, each of the sources X and Y still has min-entropy at least $n - n^{\gamma} - (t+1)\ell - n^{\gamma/10} > n - n^{12\gamma}$ with probability at least $1 - 2^{-n^{\gamma/10}}$.

Lemma 6.13 now follows directly from Lemma 6.9 by noting that the following hold by our choice of parameters:

- $n - n^{12\gamma} > \frac{n}{2} + 20(n^{\beta_1} + n^{\beta_2})(n^{\gamma/C}(n^{\beta_3} + n^{\beta_4}) + n^{\beta_4})$
- $n^{\beta_3} > \frac{4}{3}(20tn^{\beta_4} + n^{12\gamma})$
- $2^{n^{12\gamma}} 2^{-\Omega(n^{\beta_4/2})} < 2^{-\Omega(n^{\beta_4/4})}$.

This concludes the proof. □

Proof of Lemma 6.14. Let $\mathcal{A}_1 = (f_1, g_1), \ldots, \mathcal{A}_t = (f_t, g_t)$ be arbitrary 2-split-state adversaries. We partition $\{0,1\}^n$ in two different ways based on the fixed points of the tampering functions.

For any $R \subseteq [t]$, define

$$W^{(R)} = \{x \in \{0,1\}^n : f_i(x) = x \text{ if } i \in R, \text{ and } f_i(x) \neq x \text{ if } i \in [t] \setminus R\}.$$

Similary, for any $S \subseteq [t]$, define

$$V^{(S)} = \{y \in \{0,1\}^n : g_i(y) = y \text{ if } i \in S, \text{ and } g_i(y) \neq y \text{ if } i \in [t] \setminus S\}.$$

Thus the sets $W^{(R)}, R \subseteq [t]$ defines a partition of $\{0,1\}^n$. Similarly $V^{(S)}, S \subseteq [t]$ defines a partition of $\{0,1\}^n$. For $R, S \subseteq [t]$, let $X^{(R)}$ be a random variable uniform on $W^{(R)}$, and $Y^{(S)}$ be a random variable uniform on $V^{(S)}$.

Let U_{n_4} be uniform on $\{0,1\}^{n_4}$ and independent of X^R, Y^S, for all $R, S \subseteq [t]$.

Define

$$D^{(R,S)}_{\vec{f},\vec{g}} = (U_{n_4}, Z_1^{(R,S)}, \ldots, Z_t^{(R,S)})$$

where we define the random variable

$$Z_i^{(R,S)} = \begin{cases} \text{nmExt}(f_i(X^{(R)}), g_i(Y^{(S)})) & \text{if } i \in [t] \setminus (R \cap S) \\ same^{\star} & \text{if } i \in R \cap S \end{cases}$$

Define the distribution:

$$D_{\vec{f},\vec{g}} = \sum_{R,S} \alpha_{R,S} D^{(R,S)}_{\vec{f},\vec{g}}$$

, where $\alpha_{R,S} = \frac{|W^{(R,S)}||V^{(R,S)}|}{2^{2n}}$.

We first prove the following claim.

Claim 6.16. *Let*

$$\Delta_{R,S} = \alpha_{R,S}|\text{nmExt}(X^{(R)}, Y^{(S)}), \text{nmExt}(f_1(X^{(R)}), g_1(Y^{(S)})), \ldots, \\ \text{nmExt}(f_t(X^{(R)}), g_t(Y^{(S)})) - D^{(R,S)}_{\vec{f},\vec{g}}|.$$

Then, for every $R, S \subseteq [t]$, $\Delta_{R,S} \leq 2^{-n^\gamma} + \epsilon$.

Proof. If $|W^{(R)}| \leq 2^{n-n^\gamma}$, it follows that $\alpha_{R,S} \leq 2^{-n^\gamma}$, and hence the claim follows. Thus, assume that $H_\infty(X^{(R)}) \geq n - n^\gamma$. Using a similar argument, we can assume that $H_\infty(Y^{(S)}) \geq n - n^\gamma$.

Let $\overline{R \cap S} = [t] \setminus (R \cap S) = \{i_1, \ldots, i_j\}$. It follows that for any $c \in \overline{R \cap S}$, at least one the following is true: (1) f_c has no fixed points on $W^{(R)}$ (2) g_c has no fixed points on $V^{(S)}$. Thus, invoking Lemma 6.13, we have

$$|\text{nmExt}(X^{(R)}, Y^{(S)}), \text{nmExt}(f_{i_1}(X^{(R)}), g_{i_1}(Y^{(S)})), \ldots, \text{nmExt}(f_{i_j}(X^{(R)}), g_{i_j}(Y^{(S)})) \\ -U_{n_4}, \text{nmExt}(f_{i_1}(X^{(R)}), g_{i_1}(Y^{(S)})), \ldots, \text{nmExt}(f_{i_j}(X^{(R)}), g_{i_j}(Y^{(S)}))| \leq \epsilon$$

The claim now follows by observing that for each $c \in R \cap S$, f_c and g_c are the identity functions on the sets $W^{(R)}$ and $V^{(S)}$ respectively. □

Let X, Y be independent and uniformly random on $\{0,1\}^n$. Thus, we have

$$|\text{nmExt}(X, Y), \text{nmExt}(\mathcal{A}_1(X, Y)), \ldots, \text{nmExt}(\mathcal{A}_t(X, Y)) \\ -U_{n_4}, \text{copy}^{(t)}(D_{\vec{f},\vec{g}}, U_{n_4})| = \sum_{R,S \subseteq [t]} \Delta_{R,S} \leq 2^{2t}(\epsilon + 2^{-n^\gamma}).$$

Thus nmExt is a $(2, t)$-non-malleable extractor with error $(\epsilon + 2^{-n^\gamma})2^{2t}$. □

7 An Explict Seeded Non-Malleable Extractor at Polylogarithmic Min-entropy

Subroutines and Parameters

1. Let γ be a small enough constant and C a large one. Let t, k, d be parameters such that $t \leq k^{\gamma/2}$.
2. Let $n_1 = \log\left(\frac{tn}{\epsilon}\right)$. Let $\text{Ext}_s : \{0,1\}^n \times \{0,1\}^{n_1} \to \{0,1\}^{n_1}$ be the strong seeded extractor from Theorem 3.15 set to extract from min-entropy $2n_1$ and error $2^{-\Omega(n_1)}$.
3. Let $\mathcal{C}$ be an explicit $[\frac{d}{\alpha}, d, \frac{1}{10}]$-binary linear error correcting code with encoder $E : \{0,1\}^d \to \{0,1\}^{\frac{d}{\alpha}}$. Such explicit codes are known, for example from the work of Alon et al. [ABN+92].
4. Let $\text{Ext}_{\text{Samp}} : \{0,1\}^{n_1} \times \{0,1\}^{d_1} \to \{0,1\}^{n_2}$ be the strong seeded extractor from Theorem 3.16 set to extract from min-entropy $\frac{n_1}{2}$ with error $\frac{1}{20}$ and output length n_2, such that $N_2 D_1 = \frac{d}{\alpha}$, where $N_2 = 2^{n_2}$ and $D_1 = 2^{d_1}$. Let $\{0,1\}^{d_1} = \{s_1, \ldots, s_{D_1}\}$. Define $\text{Samp} : \{0,1\}^{n_1} \to [\frac{d}{\alpha}]^{D_1}$ as: $\text{Samp}(x) = (\text{Ext}(x, s_1) \circ s_1, \ldots, \text{Ext}(x, s_{D_1}) \circ s_{D_1})$. By Theorem 3.16, we have $D_1 = c_1 n_1$, for some constant c_1.
5. Let $\ell = n_1 + D_1 = (c_1 + 1)n_1$.
6. We set up the parameters for the components used by 2laExt (computed by Algorithm 1) as follows.

(a) Let $n_3 = c_3 t\ell, n_4 = 10\ell$, for some large enough constant c_3.
Let $\text{Ext}_q : \{0,1\}^{n_3} \times \{0,1\}^{n_4} \to \{0,1\}^{n_4}$ be the strong seeded extractor from Theorem 3.15 set to extract from min-entropy $k_q = \frac{n_3}{4}$ with error $\epsilon = 2^{-\Omega(n_4)}$.
Let $\text{Ext}_w : \{0,1\}^{n} \times \{0,1\}^{n_4} \to \{0,1\}^{n_4}$ be the strong seeded extractor from Theorem 3.15 set to extract from min-entropy $\frac{k}{2}$ with error $\epsilon = 2^{-\Omega(n_4)}$.

(b) Let $\text{laExt} : \{0,1\}^{n} \times \{0,1\}^{n_3+n_4} \to \{0,1\}^{2n_4}$ be the look ahead extractor used by 2laExt. Recall that the parameters in the alternating extraction protocol are set as $m = n_4, u = 2$ where u is the number of steps in the protocol, m is the length of each random variable that is communicated between the players, and $\text{Ext}_q, \text{Ext}_w$ are the strong seeded extractors used in the protocol.

(c) Let $\text{Ext} : \{0,1\}^{d} \times \{0,1\}^{n_4} \to \{0,1\}^{n_3}$ be the strong seeded extractor from Theorem 3.15 set to extract from min-entropy $\frac{d}{2}$ with seed length n_4 and error $2^{-\Omega(n_4)}$.

7. Let nmExt_1 be the function computed by Algorithm 2, which uses the function 2laExt set up as above.

8. Let $n_5 = \frac{k}{100t}$. Let $\text{Ext}_1 : \{0,1\}^{n} \times \{0,1\}^{n_4} \to \{0,1\}^{n_5}$ be the strong seeded extractor from Theorem 3.15 set to extract from min-entropy $\frac{k}{4}$ with seed length n_4, error $2^{-\Omega(n_4)}$.

Algorithm 4: snmExt(x,y)

Input: Bit strings x, y, of length n, d respectively.
Output: A bit string of length n_4.

1 $y_1 = \text{Slice}(y, n_1)$. Compute $v = \text{Ext}_s(x, y_1)$.
2 Compute $T = \text{Samp}(v) \subset [\frac{n}{\alpha}]$.
3 Let $z = y_1 \circ y_2$ where $y_2 = (E(y))_{\{T\}}$.
4 Output $\text{Ext}_1(x, \text{nmExt}_1(x, y, z))$.

We now state our main theorem.

Theorem 7.1. *Let* $\text{snmExt} : \{0,1\}^{n} \times \{0,1\}^{d} \to \{0,1\}^{n_5}$ *be the function computed by Algorithm* 4*. Then* snmExt *satisfies the following property: For any* $\epsilon > 0$*,* $k \geq C \log^{2+\gamma}\left(\frac{n}{\epsilon}\right)$*,* $t \leq k^{\gamma/2}$ *and* $d \geq Ct^2 \log^2\left(\frac{n}{\epsilon}\right)$*, if* X *is a* (n,k)*-source, and* Y *is an independent and uniform distribution on* $\{0,1\}^d$*, and* $\mathcal{A}_1 \ldots, \mathcal{A}_t$ *are arbitrary tampering functions, such that for each* $i \in [t]$*,* $\mathcal{A}_i$ *has no fixed points, then the following holds:*

$$|\text{snmExt}(X,Y), \text{snmExt}(X, \mathcal{A}_1(Y)), \ldots, \text{snmExt}(X, \mathcal{A}_t(Y)), Y - U_{n_5}, \text{snmExt}(X, \mathcal{A}_1(Y)), \ldots, \text{snmExt}(X, \mathcal{A}_t(Y)), Y| \leq O(\epsilon),$$

Notation: For any function H, if $V = H(X,Y)$, let $V^{(i)}$ denote the random variable $H(X, \mathcal{A}_i(Y))$.

Proof. We first prove the following claim.

Claim 7.2. *With probability at least* $1 - \epsilon$*,* $Z \neq Z^{(i)}$ *for each* $i \in [t]$*.*

Proof. Pick an arbitrary $i \in [t]$. If $Y_1 \neq Y_1^{(i)}$, then we have $Z \neq Z^{(i)}$. Now suppose $Y_1 = Y_1^{(i)}$. We fix Y_1, and note that since Ext_s is a strong extractor (Theorem 3.17), B is $2^{-\Omega(n_1)}$-close to U_{n_1}.

Since $\mathcal{A}_i$ has no fixed points, it follows that since E is an encoder of a code with relative distance distance $\frac{1}{10}$, $\Delta(E(Y), E(Y^{(i)})) \geq \frac{d}{10\alpha}$. Let $\mathcal{D} = \{j \in \left[\frac{d}{\alpha}\right] : E(Y)_{\{j\}} \neq E(Y^{(i)})_{\{j\}}\}$. Thus $|\mathcal{D}| \geq \frac{d}{10\alpha}$. Using Theorem 6.2, it follows that with probability at least $1 - \epsilon$, $|\mathcal{D} \cap \text{Samp}(V)| \geq 1$, and thus $Y_2 \neq Y_2^{(i)}$ (since $\text{Samp}(V) = \text{Samp}(V^{(i)})$). The claim now follows by a simple union bound. □

We fix $Z, Z^{(1)}, \ldots, Z^{(t)}$ such that $Z \neq Z^{(i)}$ for any $i \in [t]$ (from the lemma above, this occurs with probability $1 - \epsilon$). We note that by the Lemma 7.2 and Lemma 3.11, the source X has min-entropy at least $k - 2n_1$ and the source Y has min-entropy at least $d - 2\ell$ with probability at least $1 - \epsilon$.

Lemma 6.13 now follows directly from Lemma 6.9 by noting that the following hold by our choice of parameters:

- $\frac{d}{2} > 20\ell(t(n_3 + n_4) + \log(\frac{1}{\epsilon}))$
- $k - 2n_1 \geq \frac{n_3}{4} + 20\ell(tn_4 + \log(\frac{1}{\epsilon}))$
- $n_3 - 2n_1 \geq \frac{4}{3}(10tn_4 + 2\log(\frac{1}{\epsilon}))$

This concludes the proof. □

8 Efficient Encoding and Decoding Algorithms for One-Many Non-Malleable Codes

In this section, we construct efficient algorithms for almost uniformly sampling from the pre-image of any output of a modified version of the $(2, t)$-non-malleable extractor constructed in Section 6. Combining this with Theorem 5.1 and Theorem 6.12 gives us efficient constructions of one-many non-malleable codes in the 2-split state model, with tampering degree $t = n^{\Omega(1)}$, relative rate $n^{\Omega(1)}/n$ and error $2^{-n^{\Omega(1)}}$.

A major part of this section is on modifying the components used in the construction of nmExt (Algorithm 3) so that the overall extractor is much simpler to analyze as a function, and this enables us to develop efficient sampling algorithms from the pre-image. We present the modified extractor construction in Section 8.2. However, we first need to solve a simpler problem.

8.1 A New Linear Seeded Extractor

A crucial sub-problem that we have to solve is almost uniformly sampling from the pre-image of a linear seeded extractor in polynomial time. Towards this, we recall a well known property of linear seeded extractors.

Lemma 8.1 ([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*

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

□

Definition 8.2. *For any seeded extractor* $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$*, any* $s \in \{0,1\}^d$ *and* $r \in \{0,1\}^m$*, we define:*

- $\text{Ext}(\cdot, s) : \{0,1\}^n \to \{0,1\}^m$ *to be the map* $\text{Ext}(\cdot, s)(x) = \text{Ext}(x, s)$.
- $\text{Ext}^{-1}(r)$ *to be the set* $\{(x, y) \in \{0,1\}^n \times \{0,1\}^d : \text{Ext}(x, y) = r\}$.
- $\text{Ext}^{-1}(\cdot, s)$ *to be the set* $\{x : \text{Ext}(x, s) = r\}$.

We now present a natural way of sampling from pre-images of linear seeded extractors.

Claim 8.3. *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* $\epsilon < 2^{-1.5m}$*. For any* $r \in \{0,1\}^m$*, consider the following efficient sampling procedure* $\mathcal{S}$ *which on input* r *does the following: (a) Sample* $s \sim U_d$*, (b) sample* x *uniformly from the subspace* $\text{Ext}(\cdot, s)^{-1}(r)$*. (c) Output* (x, s)*. Let* $\mathcal{D}_r$ *be the distribution uniform on* $\text{Ext}^{-1}(r)$*, and let* $\mathcal{S}(r)$ *denote the distribution produced by* $\mathcal{S}$ *on input* r*.*

Then,

$$|\mathcal{S}(r) - \mathcal{D}_r| \leq 2^{-\Omega(m)}$$

Proof. Define the sets:

$$Good = \{s \in \{0,1\}^d : \text{rank}(\text{Ext}(\cdot, s)) = m\}, \quad Bad = \{0,1\}^d \setminus Good.$$

It follows by Lemma 8.1 that $|Good| \geq (1-\epsilon)2^d$. Thus, for any $s \in Good$, $|\text{Ext}(\cdot, s)^{-1}(r)| = 2^{n-m}$. Thus, we have

$$\sum_{s \in Good} |\text{Ext}^{-1}(\cdot, s)(r)| \geq 2^{d+n-m-1}.$$

Further, for any $s' \in Bad$, $|\text{Ext}^{-1}(\cdot, s')(r)| \leq 2^n$, and hence

$$\sum_{s' \in Bad} |\text{Ext}^{-1}(\cdot, s')(r)| \leq \epsilon 2^{d+n} < 2^{d+n-1.5m}.$$

Thus $|\cup_{s' \in bad} \text{Ext}^{-1}(\cdot, s')(r)| < 2^{-0.5m}|\text{Ext}^{-1}(r)|$. It now follows that

$$|\mathcal{S}(r) - \mathcal{D}_r| \leq 2^{-0.4m}$$

□

We note that ϵ must be $o(2^{-m})$ for the above sampling procedure to work with low enough error. However, this would require a seed length of $d = O(m^2)$ (by Theorem 3.14). For each step of the alternating extraction protocol the seed length then goes down by a quadratic factor, which is insufficient for our application.

To get past this difficulty, we construct a new strong linear seeded extractor for high min-entropy sources with the seed length close to the output length with the property that the size of the pre-image of any output is the same for any fixing of the seed. Algorithm 5 provides this construction.

Parameters and Subroutines:

1. Let $\delta > 0$ be any constant. Let $d = n^\delta$. Let $d = d_1 + d_2$, where $d_1 = n^{\delta_1}$, $\delta > 10\delta_1$. Let $m = d/2$.
2. Let $\text{Samp} : \{0,1\}^{d_1} \to [n]^t$, $t = d_2$, be an (μ, θ, γ) averaging sampler with distinct samples, such that $\mu = \frac{(\delta - 2\tau)}{\log(1/\tau)}$, $\theta = \frac{\tau}{\log(1/\tau)}$ and $\gamma = 2^{-\Omega(d_1)}$, $\tau = 0.05$.
3. Let $\text{IP} : \{0,1\}^{d_2} \times \{0,1\}^{d_2} \to \{0,1\}^{\frac{d}{2}}$ be the strong 2-source extractor from Theorem 3.17.

Informally the construction of iExt is as follows. Given a uniform seed S, we use a slice S_1 of S to sample co-ordinates from the weak source X, and then apply a strong 2-source extractor (based on the inner product function) to the source X_1 (which is the projection of X to the sampled co-ordinates) and the remaining bits S_2 of S to extract $\frac{d}{2}$ uniform bits.

The correctness of this procedure relies on the fact that by pseudorandomly sampling co-ordinates of X and projecting X to these co-ordinates, the min-entropy rate is roughly the preserved for most choices

Algorithm 5: iExt(x,s)

Input: Bit strings x, s of length n, d respectively.
Output: A bit string of length m.

1 Let $s_1 = \text{Slice}(s, d_1)$. Let s_2 be the remaining d_2 bits of s.
2 Let $T = \text{Samp}(s_1) \subset [n]$. Let $x_1 = x_{\{T\}}$.
3 Output $\text{IP}(x_1, s_2)$.

of the uniform seed [Zuc97] [Vad04] [Li12a]. Thus, we can fix S_1, and the strong two-source extractor IP now receives two independent inputs S_2 and X_2 with almost full min-entropy. Thus, the output is close to uniform. Further we show that the number of linear constraints on the source X is the same for any fixing of the seed. This allows us to show that size of the pre-image of any particular output is the same for any fixing of the seed. We now formally prove these ideas.

We need the following theorem proved by Vadhan [Vad04].

Theorem 8.4 ([Vad04])**.** *Let $1 \geq \delta \geq 3\tau > 0$. Let* Samp $: \{0,1\}^r \to [n]^t$ *be an (μ, θ, γ) averaging sampler with distinct samples, such that $\mu = \frac{(\delta - 2\tau)}{\log(1/\tau)}$ and $\theta = \frac{\tau}{\log(1/\tau)}$. If X is a $(n, \delta n)$ source, then the random variable $(U_r, X_{\{Samp(U_r)\}})$ is $(\gamma + 2^{-\Omega(\tau n)})$-close to (U_r, W) where for every $a \in \{0,1\}^r$, the random variable $W|U_r = a$ is a $(t, (\delta - 3\tau)t)$-source.*

Lemma 8.5. *Let* iExt *be the function computed by Algorithm 5. If X is a $(n, 0.9n)$ source and S is an independent uniform seed on $\{0,1\}^d$, then the following holds:*

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

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

Proof. Using Theorem 8.4, it follows that X_1 is $2^{-n^{\Omega(1)}}$-close to a source with min-entropy at least $0.8n$ for any fixing of S_1. Further, we note that after fixing S_1, S_2 and X_1 are independent sources. We now think of X_1, S_2 as sources in $\{0,1\}^{d_2+1}$ by appending a 1 to both the sources, so that $S_2 \neq \vec{0}$, and then apply the inner product map. This results in an entropy loss of only 1. It now follows by Theorem 3.17 that

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

It is easy to see that for any fixing of the seed $S = s$, $\text{iExt}(\cdot, s)$ is a linear map. Let X be uniform on n bits. We note that for any fixing of $S_2 = s_2$, X_1 lies in a subspace of dimension $d_2 - m$ over F_2. Further, the bits outside T have no restrictions placed on them. Thus the size of $\text{iExt}(\cdot, s)^{-1}(r)$ is exactly $2^{d_2 - m + n - d_2} = 2^{n-m}$. This completes the proof of the lemma. □

Based on the above lemma, we construct an efficient procedure for sampling uniformly from the pre-image of the function iExt.

Claim 8.6. *Let* $\text{iExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be the function computed by Algorithm 5. Then there exists a polynomial time algorithm* Samp_1 *that takes as input $r \in \{0,1\}^m$, and samples from a distribution that is uniform on* $\text{iExt}^{-1}(r)$.

Proof. It follows by Lemma 8.5 that for any fixing of the seed s, the size of the set $\text{iExt}(\cdot, s)^{-1}(r)$ is exactly 2^{n-m}. Thus we can use the following strategy: (a) Sample $s \sim U_d$ (b) Sample x uniformly random from the subspace $\text{iExt}(\cdot, s)^{-1}(r)$ (c) Output (x, s). It follows that each element in $\text{iExt}^{-1}(r)$ is picked with probability exactly $\frac{1}{2^d} \cdot \frac{1}{2^{n-m}}$. Thus the output of our sampling procedure is indeed uniform on $\text{iExt}^{-1}(r)$. □

8.2 A Modified Construction of the Seedless $(2, t)$-Non-Malleable Extractor

We first describe the high level ideas involved in modifying the construction of nmExt (Algorithm 3), before presenting the formal construction.

- We use the linear seeded extractor iExt (Algorithm 5) for any seeded extractor used in the construction of nmExt.
- Next we divide the sources X and Y into blocks of size $n^{1-\delta}$ respectively for a small constant δ. Since each of X and Y have almost full min-entropy, we now have two block sources, where each block has almost full min-entropy conditioned on the previous blocks. The idea is to use new blocks of X and Y for each round of alternating extraction in nmExt.

  To implement this however, we need some care. Recall that the alternating extraction protocol is run for two rounds between either X and Q_h, or X and $\overline{Q}_h$ in the function 2laExt. The idea now is to run these two of alternating extraction by dividing Q_h into two blocks, and using two new partitions of X (each round being run by using a block from either X or Q_h). Now to generate these Q_h's, we use a $O(t)$ blocks of Y, and for each block apply the strong seeded extractor iExt, using as seed the output of the alternating extraction from the previous step, and finally concatenate the outputs. This works because these $O(t)$ blocks form a block source, and using the same seed to extract from all the blocks is a well known technique of extracting from block sources.
- By appropriate setting of the lengths of the seeds in the alternating extraction, we ensure that each block of X and Y still has min-entropy rate $1 - o(1)$ even after fixing all the intermediate seeds, the random variables $Q_h, \overline{Q}_h$ and their tampered versions. This can be ensured since each of these variables are of length at most n^{δ_1} for some small constant δ_1, and the number of adversaries is also $n^{\Omega(1)}$).
- The above modification is almost sufficient for us to successfully sample from the pre-image of any output. One final modification is to use a specific error correcting code (the Reed-Solomon code over a field of size $n + 1$ with characteristic 2) in the initial step of the construction, when we encode the sources and sample bits from it. We give some intuition as to why this is necessary. Since we are using linear seeded extractors in the alternating extraction, by fixing the seeds we impose linear restrictions on the blocks of X and Y. Now, if we fix the output of the initial sampling step (the random variable Z in Algorithm 3), we are imposing more linear constraints on the blocks (assuming we are using a linear code). Now, it is not clear if the constraints imposed by the linear seeded extractor is independent from the constraints imposed by Z, and thus for different fixings of the Z and the seeds the size of the pre-image of any output of the non-malleable extractor may be different.

  To get past this difficulty, our idea is to first partition X and Y into slightly smaller blocks (which does not affect the correctness of the extractor) such that at least half of the blocks are unused by the alternating extraction steps. Now, we show that by using the Reed-Solomon code over $\mathbb{F} = \mathbb{F}_{2^{\log(n+1)}}$ to encode the sources, fixing Z imposes linear constraints involving the variables from these unused blocks, and we show that this is sufficient to argue that it is linearly independent of the restrictions imposed by the alternating extraction part. We provide complete details of the sampling algorithms in Section 8.3.

We now proceed to present the extractor construction. Recall that if $Z_a, Z_{a+1}, \ldots, Z_b$ are random variables, we use $Z_{[a,b]}$ to denote the random variable $Z_a, \ldots, Z_b$.

Subroutines and Parameters (used by Algorithm 6, Algorithm 7, Algorithm 8)

1. Let γ be a small enough constant and C a large one. Let $t = n^{\gamma/C}$.
2. Let $n_1 = n^{\beta_1}$, $\beta_1 = 10\gamma$. Let $n_2 = n - n_1$. Let $\text{IP}_1 : \{0,1\}^{n_1} \times \{0,1\}^{n_1} \to \{0,1\}^{n_3}$, $n_3 = \frac{n_1}{10}$ be the strong two-source extractor from Theorem 3.17.
3. Let $\mathbb{F}$ be the finite field $\mathbb{F}_{2^{\log(n+1)}}$. Let $n_4 = \frac{n_2}{\log(n+1)}$. Let $\text{RS} : \mathbb{F}^{n_4} \to \mathbb{F}^n$ be the Reed-Solomon code encoding n_4 symbols of $\mathbb{F}$ to n symbols in $\mathbb{F}$ (we overload the use of RS, using it to denote both the code and the encoder). Thus RS is a $[n, n_4, n - n_4 + 1]_n$ error correcting code.
4. Let $\text{Samp} : \{0,1\}^{n_3} \to [n]^{n_5}$ be a $(\mu, \frac{1}{10}, 2^{-n^{\Omega(1)}})$ averaging sampler with distinct samples. By using the strong seeded extractor from Theorem 3.15, we can set $n_5 = n^{\beta_2}$, $\beta_2 < \beta_1/2$.
5. Let $\ell = 2(n_1 + n_5 \log n) < 4n^{\beta_1}$. Thus $\ell \leq n^{11\gamma}$.
6. Let $n_6 = 50Ct\ell$. Let $\text{IP}_2 : \{0,1\}^{n_6} \times \{0,1\}^{n_6} \to \{0,1\}^{2n_q}$, $n_q = 10Ct\ell$, be the strong two-source extractor from Theorem 3.17.
7. Let $n_7 = n - n_1 - n_6$. Let $n_x = \frac{n_7}{8\ell}$. Let $n_y = \frac{n_7}{16Ct\ell}$. Thus $n_x, n_y \geq n^{1-15\gamma}$.
8. Let $d_1 = 80\ell$.
9. Let $\text{iExt}_1 : \{0,1\}^{n_x} \times \{0,1\}^{d_1} \to \{0,1\}^{d_2}$, $d_2 = 40\ell$, be the extractor computed by Algorithm 5.
10. Let $\text{iExt}_2 : \{0,1\}^{n_q} \times \{0,1\}^{d_2} \to \{0,1\}^{d_3}$, $d_3 = 20\ell$, be the extractor computed by Algorithm 5.
11. Let $\text{iExt}_3 : \{0,1\}^{n_x} \times \{0,1\}^{d_3} \to \{0,1\}^{d_4}$, $d_4 = 10\ell$ be the extractor computed by Algorithm 5.
12. Let $\text{iExt}_4 : \{0,1\}^{n_y} \times \{0,1\}^{d_4} \to \{0,1\}^{d_5}$, $d_5 = 5\ell$, be the extractor computed by Algorithm 5.
13. Let $\text{Ext} : \{0,1\}^{4Ctn_y} \times \{0,1\}^{d_4} \to \{0,1\}^{2n_q}$ be defined in the following way. Let $v_1, \ldots, v_{4t}$ be strings, each of length n_y. Define $\text{Ext}(v_1 \circ \ldots \circ v_{4Ct}, s) = \text{iExt}_4(v_1, s) \circ \ldots \circ \text{iExt}_4(v_{4Ct}, s)$.

Algorithm 6: inmExt(x,y)

Input: Bit strings x, y, each of length n.
Output: A bit string of length m.

1 Let $x_1 = \text{Slice}(x, n_1)$, $y_1 = \text{Slice}(y, n_1)$. Compute $\nu = \text{IP}_1(x, y)$.
2 Let x_2, y_2 be n_2 length strings formed by cutting x_1, y_1 from x, y respectively.
3 Let $T = \text{Samp}(\nu) \subset [n]$.
4 Interpret x_2, y_2 as elements in $\mathbb{F}^{n_4}$.
5 Let $\overline{x}_2 = \text{RS}(x_2), \overline{y}_2 = \text{RS}(y_2)$.
6 Let $\overline{x}_1 = (\overline{x}_2)_{\{T\}}, \overline{y}_1 = (\overline{y}_2)_{\{T\}}$, interpreting $\overline{x}_2, \overline{y}_2 \in \mathbb{F}^n$.
7 Let $z = x_1 \circ \overline{x}_1 \circ y_1 \circ \overline{y}_1$, where z is interpreted as a binary string.
8 Interpret x_2, y_2 as binary strings.
9 Output $\text{inmExt}_1(x_2, y_2, z)$.

Theorem 8.7. *Let* inmExt *be the function computed by Algorithm* 7. *Then* inmExt *is a seedless* $(2, t)$*-non-malleable extractor with error* $2^{-n^{\Omega(1)}}$.

Algorithm 7: $\text{inmExt}_1(x_2, y_2, z)$

Let $x_3 = \text{Slice}(x_2, n_6)$, $y_3 = \text{Slice}(y_2, n_6)$. Let w, v be the remaining parts of x_2, y_2 respectively.
Let $\text{IP}_2(x_3, y_3) = (q_{1,1}, q_{1,2})$, where each of $q_{1,1}, q_{1,2}$ is of length n_q.
Let $w_1, \ldots, w_{8\ell}$ be an equal sized partition of the string w into 8ℓ stings.
Let $v_1, \ldots, v_{16t\ell}$ be an equal sized partition of the string v into $16Ct\ell$ stings.
for $h = 1$ *to* ℓ **do**
  $(q_{h+1,1}, q_{h+1,2}) = \text{2ilaExt}(v_{[8C(h-1)t+1,8Cht]}, w_{[4h-3,4h]}, q_{h,1}, q_{h,2}, h, z_{\{h\}})$
end
Ouput $(q_{\ell+1,1}, q_{\ell+1,2})$.

Algorithm 8: $\text{2ilaExt}(v_{[8C(h-1)t+1,8Cht]}, w_{[4h-3,4h]}, q_{h,1}, q_{h,2}, h, b)$

Let $s_{h,1} = \text{Slice}(q_{h,1}, d_1)$, $r_{h,1} = \text{Ext}_1(w_{4h-3}, s_{h,1})$, $s_{h,2} = \text{Ext}_2(q_{h,2}, r_{h,1})$, $r_{h,2} = \text{Ext}_3(w_{4h-2}, s_{h,2})$.
if $b = 0$ **then**
  Let $r_h = \text{Slice}(r_{h,1}, d_4)$.
else
  Let $r_h = r_{h,2}$
end
Let $\text{Ext}(v_{[8C(h-1)t+1,8(h-1)t+4Ct]}, r_h) = (\overline{q}_{h,1}, \overline{q}_{h,2})$, where both $\overline{q}_{h,1}, \overline{q}_{h,2}$ are of length n_q.
Let $\overline{s}_{h,1} = \text{Slice}(\overline{q}_{h,1}, d_1)$, $\overline{r}_{h,1} = \text{Ext}_1(w_{4h-1}, \overline{s}_{h,1})$, $\overline{s}_{h,2} = \text{Ext}_2(\overline{q}_{h,2}, \overline{r}_{h,1})$, $\overline{r}_{h,2} = \text{Ext}_3(w_{4h}, \overline{s}_{h,2})$.
if $b = 0$ **then**
  Let $\overline{r}_h = \overline{r}_{h,2}$.
else
  Let $\overline{r}_h = \text{Slice}(\overline{r}_{h,1}, d_4)$.
end
Let $\text{Ext}(v_{[8C(h-1)t+4Ct+1,8Cht]}, r_h) = (q_{h+1,1}, q_{h+1,2})$, where both $q_{h+1,1}, q_{h+1,2}$ are of length n_q.
Ouput $(q_{h+1,1}, q_{h+1,2})$.

The proof of the above theorem is essentially the same as the proof provided in Section 6, and we do not repeat it. The correctness of inmExt follows directly from the proof of Theorem 6.12, and the correctness of the extractor iExt (Lemma 8.5), the fact that by our choice of parameters each block of X and Y still has min-entropy rate at least 0.9 after appropriate conditioning of the intermediate random variables and their tampered versions, and the fact that using the RS in place of a binary error correcting code does not affect correctness of the procedure.

8.3 Efficiently Sampling from the Pre-Image of inmExt

Since the construction of the non-malleable extractor inmExt (Algorithm 6, Algorithm 7, Algorithm 8) is composed of various sub-parts and sub-functions, we first argue about the invertibility of these parts and then show a way to compose these sampling procedure to sample almost uniformly from the pre-image of inmExt. We refer to all the variables, sub-routines and notations introduced in these algorithms while developing the sampling procedures. Unless we state otherwise, by a subspace we mean a subspace over $\mathbb{F}_2$.

We first show how to sample uniformly from the pre-image of 2ilaExt (Algorithm 8), since it is a crucial sub-part of inmExt. We have the following claim.

Claim 8.8. *For any fixing of the variables* $\{s_{1,i}, r_{1,i}, \overline{s}_{1,i}, \overline{r}_{1,i} : i \in \{1,2\}\}$*, and any* $b \in \{0,1\}$ *define the set:*

$$\text{2ilaExt}^{-1}(q_{2,1}, q_{2,2}) = \{(x_3, y_3, v_{[1,4Ct]}, w_{[1,4]}) \in \{0,1\}^{2n_6+4Ctn_y+4n_x} : \\ \text{2ilaExt}(v_{[1,4Ct]}, w_{[1,4]}, q_{1,1}, q_{1,2}, b) = (q_{2,1}, q_{2,2})\}$$

There exists an efficient algorithm Samp_2 *that takes as input* $q_{2,1}, q_{2,2}, b, \{s_{1,i}, r_{1,i}, \overline{s}_{1,i}, \overline{r}_{1,i} : i \in \{1,2\}\}$*, and samples uniformly from* $\text{2ilaExt}^{-1}(q_{2,1}, q_{2,2})$.

Further, the set $\text{2ilaExt}^{-1}(q_{2,1}, q_{2,2})$ *is a subspace over* $\mathbb{F}_2$ *of dimension* d_1*, and its size does not depend on the inputs to* Samp_2.

Proof. The general idea is that by fixing the seeds in the alternating extraction, each block of w takes values independent of the fixing of the other blocks of w and the $q_{i,j}$'s, and similarly the $q_{i,j}$'s takes values independent of each other and the blocks of w. We now formally prove this intuition.

Since, $s_{1,1}$ is a slice of $q_{1,1}$ it follows that $q_{1,1}$ is restricted to the subspace of size $2^{n_q-d_1}$. Since $r_{1,1} = \text{iExt}_1(w_1, s_{1,1})$, it follows that w_1 is restricted to the set $\text{iExt}_1(\cdot, s_{1,1})^{-1}(r_{1,1})$. Further, it follows by Lemma 8.5 that this is a subspace of size $2^{n_x-d_2}$. Similar arguments show that $q_{1,2}$ is restricted to the subspace of dimension $2^{n_q-d_3}$, and w_2 is restricted to a subspace of dimension $2^{n_x-d_4}$. Further, we note that each of these variables have no correlation.

By repeating this argument for the next two rounds of alternating extraction, it follows that $\overline{q}_{1,1}$ is restricted to a subspace of size $2^{n_q-d_1}$, w_3 is restricted to a subspace of size $2^{n_x-d_2}$, $\overline{q}_{1,2}$ is restricted to a subspace of size $2^{n_q-d_3}$, and w_4 is restricted to a subspace of size $2^{n_x-d_4}$.

Further since $(q_{2,1}, q_{2,2}) = \text{Ext}(v_{[4Ct+1,8t]}, r_1) = \text{iExt}_4(v_{4Ct+1}, r_1) \circ \ldots \circ \text{iExt}_4(v_{8Ct}, r_1)$, it follows by an application of Lemma 8.5 that for any fixed $q_{2,1}$, $v_{[4Ct+1,6t]}$ is restricted to a subspace of size $2^{2Ct(n_y-d_5)}$. A similar argument shows that for any fixed $q_{2,2}$, $v_{[6Ct+1,8Ct]}$ is restricted to a subspace of size $2^{2Ct(n_y-d_5)}$.

Finally, since $\text{IP}_1(x_3, y_3) = (q_{1,1}, q_{1,2})$, it follows that for any fixed $x_3, q_{1,1}, q_{1,2}$, the variable y_3 lies in a subspace of size $2^{n_6-\log(2n_q)}$ since by fixing the variables $x_3, q_{1,1}, q_{1,2}$, we are restricting y_3 to a subspace of dimension $\left(\frac{n_6}{\log(2n_q)} - 1\right)$ over the field $\mathbb{F}_{2^{\log(2n_q)}}$.

It is clear from the arguments that we did not use any specific values of the inputs given to the algorithm Samp_1 (including the value of the bit b) to argue about the size of $\text{2ilaExt}^{-1}(q_{2,1}, q_{2,2})$. Also note that each of $x_3, y_3, v_{[1,4Ct]}, w_{[1,4]}$ is restricted to some subspace. Since $\text{2ilaExt}^{-1}(q_{2,1}, q_{2,2})$ is the cartesian product of these subspaces, it follows that it is a subspace over $\mathbb{F}_2$. Thus the lemma now follows since we can efficiently sample from a given subspace. □

Using arguments very similar to the above claim, we obtain the following result.

Claim 8.9. *For any* $h \in \{2, \ldots, \ell\}$*, any fixing of the variables* $\{s_{h,i}, r_{h,i}, \overline{s}_{h,i}, \overline{r}_{h,i} : i \in \{1,2\}\}$*, and any* $b \in \{0,1\}$ *define the set:*

$$\text{2ilaExt}^{-1}(q_{h+1,1}, q_{h+1,2}) = \{(v_{[8C(h-1)t-4Ct+1,8C(h-1)t+4Ct]}, w_{[4h-3,4h]}) \in \{0,1\}^{8Ctn_y+4n_x} : \\ \text{2ilaExt}(v_{[8C(h-1)t+1,8Cht]}, w_{[4h-3,4h]}, q_{1,1}, q_{1,2}, b) = (q_{h+1,1}, q_{h+1,2})\}.$$

There exists an efficient algorithm Samp_{h+1} *that takes as input* $q_{h+1,1}, q_{h+1,2}, b, \{s_{h,i}, r_{h,i}, \overline{s}_{h,i}, \overline{r}_{h,i} : i \in \{1,2\}\}$*, and samples uniformly from* $\text{2ilaExt}^{-1}(q_{h+1,1}, q_{h+1,2})$.

Further, $\text{2ilaExt}^{-1}(q_{h+1,1}, q_{h+1,2})$ *is a subspace over* $\mathbb{F}_2$*, and its size does not depend on the inputs to* Samp_{h+1}.

□

We now show a way of efficiently sampling from the pre-image of the function inmExt_1 (Algorithm 7).

Claim 8.10. *For any string $\alpha \in \{0,1\}^\ell$, and any fixing of the variables $\{s_{h,i}, r_{h,i}, \overline{s}_{h,i}, \overline{r}_{h,i} : h \in [\ell], i \in \{1, 2\}\}$ define the set:*

$$\text{inmExt}_1^{-1}(q_{\ell+1,1}, q_{\ell+1,2}) = \{(x_2, y_2) \in \{0,1\}^{2n_2} : \text{inmExt}_1(x_2, y_2, \alpha) = (q_{\ell+1,1}, q_{\ell+1,2})\}.$$

There exists an efficient algorithm Samp_{nm_1} that takes as input $\{s_{h,i}, r_{h,i}, \overline{s}_{h,i}, \overline{r}_{h,i} : h \in [\ell], i \in \{1,2\}\}, \alpha$, $q_{\ell+1,1}, q_{\ell+1,2}$, and samples uniformly from $\text{inmExt}_1^{-1}(q_{\ell+1,1}, q_{\ell+1,2})$.

Further, $\text{inmExt}_1^{-1}(q_{\ell+1,1}, q_{\ell+1,2})$ is a subspace over $\mathbb{F}_2$, and its size does not depend on the inputs to Samp_{nm_1}.

Proof. We observe that once we fix all the seeds $\{s_{h,i}, r_{h,i}, \overline{s}_{h,i}, \overline{r}_{h,i} : h \in [\ell], i \in \{1,2\}\}$, for different $h \in [\ell]$, the blocks $(v_{[8C(h-1)t-4Ct+1,8C(h-1)t+4Ct]}, w_{[4h-3,4h]})$ can be sampled independently. Thus, by using the algorithms $\{\text{Samp}_{h+1} : h \in \ell\}$ from Claim 8.8 and Claim 8.9, we sample the variable x_3, y_3, $w_{[1,4]}, v_{[1,4Ct]}, \{v_{[8C(h-1)t-4Ct+1,8C(h-1)t+4Ct]}, w_{[4h-3,4h]} : h \in [\ell]\}$.

Finally, since $\text{Ext}(v_{[8C(\ell-1)t+4Ct+1,8C\ell t]}, \overline{r}_\ell) = (q_{\ell+1,1}, q_{\ell+1,2})$, it follows by the arguments in Lemma 8.8, that the block $v_{[8C(\ell-1)t+4Ct+1,8C\ell t)]}$ is restricted to a subspace of size $2^{4Ct(n_y-d_5)}$. Thus, we can efficiently sample this block as well.

Further the variable $w_{[4\ell+1,8\ell]}$ is unused by the algorithm inmExt_1, and hence takes all values in $\{0,1\}^{4\ell n_x}$. Similarly the variable $v_{[8C\ell t+1,16C\ell t]}$ is unused by the algorithm inmExt_1 and hence takes all values in $\{0,1\}^{8Ct\ell}$. Thus, we sample these variables as uniform strings of the appropriate length.

Since x_2, y_2 are concatenations of the various blocks sampled above, we can indeed sample efficiently from a distribution uniform on $\{(x_2, y_2) \in \{0,1\}^{2n_2} : \text{inmExt}(x, y, \alpha) = (q_{\ell+1,1}, q_{\ell+1,2})\}$. Further since by Claim 8.8 and Claim 8.9, the size of the pre-images of each of the blocks generated do not depend on the inputs (and is also a subspace), it follows that $2\text{inmExt}_1^{-1}(q_{\ell+1,1}, q_{\ell+1,2})$ is a subspace, and its size does not depend on the inputs to Samp_{nm_1}. □

We now proceed to construct an algorithm to uniformly sample from the pre-image of any output of the function inmExt (Algorithm 6), which will yield the required efficient encoder for the resulting one-many non-malleable codes.

Claim 8.11. *For any fixing of the variable $z = x_1 \circ \overline{x}_1 \circ y_1 \circ \overline{y}_1$ and the variables $\{s_{h,i}, r_{h,i}, \overline{s}_{h,i}, \overline{r}_{h,i} : h \in [\ell]$, $i \in \{1,2\}\}$, define the set:*

$$\text{inmExt}^{-1}(q_{\ell+1,1}, q_{\ell+1,2}) = \{(x, y) \in \{0,1\}^{2n} : \text{inmExt}(x, y) = (q_{\ell+1,1}, q_{\ell+1,2})\}.$$

There exists an efficient algorithm Samp_{nm} that takes as input $\{s_{h,i}, r_{h,i}, \overline{s}_{h,i}, \overline{r}_{h,i} : h \in [\ell], i \in \{1,2\}\}, z$, $q_{\ell+1,1}, q_{\ell+1,2}$, and samples uniformly from $\text{inmExt}^{-1}(q_{\ell+1,1}, q_{\ell+1,2})$.

Further, $\text{inmExt}^{-1}(q_{\ell+1,1}, q_{\ell+1,2})$ is a subspace over $\mathbb{F}_2$, and its size does not depend on the inputs to Samp_{nm}.

Proof. We fix the variables x_1 and y_1. Let $T = \text{Samp}(\nu) = \{t_1, \ldots, t_{n_5}\}$. We now think of x_2 as an element in $\mathbb{F}^{n_4}$, $\mathbb{F} = \mathbb{F}_{2^{\log(n+1)}}$. Let $x_2 = (x_{2,1}, \ldots, x_{2,n_4})$, where each $x_{2,i}$ is in $\mathbb{F}$. Recall that the $n_4 \times n$ generator matrix G of the code RS is the following:

$$G = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{n_4-1} & \alpha_2^{n_4-1} & \cdots & \alpha_n^{n_4-1} \end{pmatrix}$$

where $\alpha_1, \ldots, \alpha_n$ are distinct non-zero field elements of $\mathbb{F}$.

Let

$$G_T = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_{t_1} & \alpha_{t_2} & \cdots & \alpha_{t_{n_5}} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{t_1}^{n_4-1} & \alpha_{t_2}^{n_4-1} & \cdots & \alpha_{t_{n_5}}^{n_4-1} \end{pmatrix}$$

Since $\overline{x}_1 = \text{RS}(x_2)_{\{T\}}$, we have the following identity:

$$\begin{pmatrix} x_{2,1} & \cdots & x_{2,n_4} \end{pmatrix} G_T = \overline{x}_1 \tag{1}$$

Thus, for any fixing of $\overline{x}_1$, the variable x_2 is restricted to a subspace of dimension $(n_4 - n_5)$ over the field $\mathbb{F}$.

Now, let $j \in [n_4]$ be such that $(x_{2,1}, \ldots, x_{2,j})$ is the string $(x_3, w_{[1,4\ell]})$, and $(x_{2,j+1}, \ldots, x_{2,n_4})$ is the string $w_{[4\ell+1,8\ell]}$. Clearly, $(n_4 - j)\log n = 4\ell n_x$, and thus by our choice of parameters it follows that $j = n_4 - \frac{4\ell n_x}{\log n} = \frac{n_4}{2} + \frac{n_6}{log(n+1)} < \frac{2n_4}{3} < n_4 - n_5$.

We further note since any $n_5 \times n_5$ sub-matrix of G_T has full rank (since it is the Vandermonde's matrix), it follows by the rank-nullity thorem that any $j \times n_5$ sub-matrix of G_T has null space of dimension exactly $j - n_5$. Thus for any $\lambda \in \mathbb{F}^{n_5}$, the equation:

$$\begin{pmatrix} x_{2,j+1} & \cdots & x_{2,n_4} \end{pmatrix} \begin{pmatrix} \alpha_{t_1}^{j} & \alpha_{t_2}^{j} & \cdots & \alpha_{t_{n_5}}^{j} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{t_1}^{n_4-1} & \alpha_{t_2}^{n_4-1} & \cdots & \alpha_{t_{n_5}}^{n_4-1} \end{pmatrix} = \overline{x}_1 + \lambda \tag{2}$$

has exactly $|\mathbb{F}|^{(j-n_5)}$ solution.

Thus, for any fixing of the variables, $x_{2,1}, \ldots, x_{2,j}$, equation (1) has exactly $|\mathbb{F}|^{j-n_5}$ solutions. In other words, for any fixing of $x_3, w_{[1,4\ell]}, \overline{x}_1$, the variable $w_{[4\ell+1,8\ell]}$ is restricted to a subspace, and the size of the subspace does not depend on the fixing of $x_3, w_{[1,4\ell]}, \overline{x}_1$. Using, a similar argument, we can show that for any fixing of $y_3, v_{[1,8Ct\ell]}, \overline{y}_1$, the variable $v_{[8Ct\ell+1,16Ct\ell]}$ is restricted to a subspace, and the size of the subspace does not depend on the fixing of $y_3, v_{[1,8Ct\ell]}, \overline{y}_1$.

Now consider any fixing of the variables $\{s_{h,i}, r_{h,i}, \overline{s}_{h,i}, \overline{r}_{h,i} : h \in [\ell], i \in \{1,2\}\}, z$. As proved in the Claim 8.10, we can efficiently sample the variables $x_3, w_{[1,4\ell]}, y_3, v_{[1,8Ct\ell]}$. By the above argument, the variables $v_{[4\ell+1,8\ell]}$ and $w_{[8Ct\ell+1,16Ct\ell]}$ now lie in a subspace, and hence we can efficiently sample these variables as well. Thus we have an efficient procedure Samp_{nm} for uniformly sampling (x, y) from the set $\text{inmExt}^{-1}(q_{\ell+1,1}, q_{\ell+1,2})$.

It also follows by Claim 8.10, that the total size of the pre-image of the variables $x_3, w_{[1,4\ell]}, y_3, v_{[1,8Ct\ell]}$ does not depend on z or the variables $\{s_{h,i}, r_{h,i}, \overline{s}_{h,i}, \overline{r}_{h,i} : h \in [\ell], i \in \{1,2\}\}$. Further, for any fixing of $x_3, w_{[1,4\ell]}, y_3, v_{[1,8Ct\ell]}, z$, as argued above, the variables $v_{[4\ell+1,8\ell]}$ and $w_{[8Ct\ell+1,16Ct\ell]}$ now lie in a subspace, whose size does not depend on the fixed variables. Thus, overall the size of the total pre-image of x, y does not depend on the inputs to Samp_{nm}. □

We now state the main result of this section.

Theorem 8.12. *There exists an efficient procedure that given an input $(q_{\ell+1,1}, q_{\ell+1,2}) \in \{0,1\}^{n_q} \times \{0,1\}^{n_q}$, samples uniformly from the set $\{(x, y) : \text{inmExt}(x, y) = (q_{\ell+1,1}, q_{\ell+1,2})\}$.*

Proof. We use the following simple strategy.

1. Uniformly sample the variables $z, \{s_{h,i}, r_{h,i}, \overline{s}_{h,i}, \overline{r}_{h,i} : h \in [\ell], i \in \{1,2\}\}$,

2. Use the variables sampled in Step (1) as input to the algorithm Samp_{nm} to sample (x, y).

The correctness of this procedure follows directly from Claim 8.11, since it was proved that for any fixing of the variables of Step 1, the size of pre-image of inmExt is the same. □

Acknowledgments

The first author would like to thank his advisor, David Zuckerman, for his constant guidance and encouragement.

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+14] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Explicit non-malleable codes resistant to permutations. Cryptology ePrint Archive, Report 2014/316, 2014. `http://eprint.iacr.org/`.

[AKO15] Divesh Aggarwal, Tomasz Kazana, and Maciej Obremski. Inception makes non-malleable codes stronger. Cryptology ePrint Archive, Report 2015/1013, 2015. `http://eprint.iacr.org/`.

[Bou05] J. Bourgain. More on the sum-product phenomenon in prime fields and its applications. *International Journal of Number Theory*, 01(01):1–32, 2005.

[BRF13] Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors. *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. ACM, 2013.

[BRSW06] Boaz Barak, Anup Rao, Ronen Shaltiel, and Avi Wigderson. 2 source dispersers for $n^{o(1)}$ entropy and Ramsey graphs beating the Frankl-Wilson construction. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, 2006.

[CCFP11] Hervé Chabanne, Gérard D. Cohen, Jean-Pierre Flori, and Alain Patey. Non-malleable codes from the wire-tap channel. *CoRR*, abs/1105.3879, 2011.

[CCP12] Hervé Chabanne, Gérard D. Cohen, and Alain Patey. Secure network coding and non-malleable codes: Protection against linear tampering. In *ISIT*, pages 2546–2550, 2012.

[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.

[CGM+15] Nishanth Chandran, Vipul Goyal, Pratyay Mukherjee, Omkant Pandey, and Jalaj Upadhyay. Block-wise non-malleable codes. *IACR Cryptology ePrint Archive*, 2015:129, 2015.

[CKM11] SeungGeol Choi, Aggelos Kiayias, and Tal Malkin. Bitr: Built-in tamper resilience. In DongHoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 740–758. 2011.

[CKOR10] N. Chandran, B. Kanukurthi, R. Ostrovsky, and L. Reyzin. Privacy amplification with asymptotically optimal entropy loss. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing*, pages 785–794, 2010.

[Coh15] Gil Cohen. Local correlation breakers and applications to three-source extractors and mergers. 2015.

[CRS14] Gil Cohen, Ran Raz, and Gil Segev. Non-malleable extractors with short seeds and applications to privacy amplification. *SIAM Journal on Computing*, 43(2):450–476, 2014.

[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.

[CZ15] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. Technical Report TR15-119, ECCC, 2015.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. In *Proceedings of the Twenty-third Annual ACM Symposium on Theory of Computing*, STOC '91, pages 542–552, New York, NY, USA, 1991. ACM.

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

[DKRS06] Y. Dodis, J. Katz, L. Reyzin, and A. Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In *Advances in Cryptology — CRYPTO '06, 26th Annual International Cryptology Conference, Proceedings*, pages 232–250, 2006.

[DLWZ14] Yevgeniy Dodis, Xin Li, Trevor D. Wooley, and David Zuckerman. Privacy amplification and non-malleable extractors via character sums. *SIAM Journal on Computing*, 43(2):800–830, 2014.

[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.

[DY13] Yevgeniy Dodis and Yu Yu. Overcoming weak expectations. In *10th Theory of Cryptography Conference*, 2013.

[FMNV14] Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. Continuous non-malleable codes. In *TCC*, pages 465–488, 2014.

[FMVW13] Sebastian Faust, Pratyay Mukherjee, Daniele Venturi, and Daniel Wichs. Efficient non-malleable codes and key-derivation for poly-size tampering circuits. *IACR Cryptology ePrint Archive*, 2013:702, 2013.

[Goy11] Vipul Goyal. Constant round non-malleable protocols using one way functions. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 695–704, New York, NY, USA, 2011. ACM.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. *Journal of the ACM*, 56(4), 2009.

[JW15] Zahra Jafargholi and Daniel Wichs. Tamper detection and continuous non-malleable codes. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 451–480, 2015.

[KLR09] Yael Kalai, Xin Li, and Anup Rao. 2-source extractors under computational assumptions and cryptography with defective randomness. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 617–628, 2009.

[KLRZ08] Yael Tauman Kalai, Xin Li, Anup Rao, and David Zuckerman. Network extractor protocols. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 654–663, 2008.

[KR09] B. Kanukurthi and L. Reyzin. Key agreement from close secrets over unsecured channels. In *EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2009.

[Li12a] Xin Li. Design extractors, non-malleable condensers and privacy amplification. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, pages 837–854, 2012.

[Li12b] Xin Li. Non-malleable extractors, two-source extractors and privacy amplification. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science*, pages 688–697, 2012.

[Li13a] 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.

[Li13b] Xin Li. New independent source extractors with exponential improvement. In Boneh et al. [BRF13], pages 783–792.

[Li15a] Xin Li. Non-malleable condensers for arbitrary min-entropy, and almost optimal protocols for privacy amplification. In *12th Theory of Cryptography Conference*, 2015.

[Li15b] Xin Li. Three-source extractors for polylogarithmic min-entropy. 2015.

[LL12] Feng-Hao Liu and Anna Lysyanskaya. Tamper and leakage resilience in the split-state model. In *CRYPTO*, pages 517–532, 2012.

[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.

[PR08a] Rafael Pass and Alon Rosen. Concurrent nonmalleable commitments. *SIAM J. Comput.*, 37(6):1891–1925, 2008.

[PR08b] Rafael Pass and Alon Rosen. New and improved constructions of nonmalleable cryptographic protocols. *SIAM Journal on Computing*, 38(2):702–752, 2008.

[Rao07] Anup Rao. An exposition of Bourgain's 2-source extractor. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(034), 2007.

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

[Raz05] Ran Raz. Extractors with weak random seeds. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 11–20, 2005.

[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.

[RW03] Renato Renner and Stefan Wolf. Unconditional authenticity and privacy from an arbitrarily weak secret. In *Advances in Cryptology — CRYPTO '03, 23rd Annual International Cryptology Conference, Proceedings*, pages 78–95, 2003.

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

[Vad04] Salil P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *J. Cryptology*, 17(1):43–77, 2004.

[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.