J Cryptol (2017) 30:191–241
DOI: 10.1007/s00145-015-9219-z

Journal of
CRYPTOLOGY

Non-malleable Coding Against Bit-Wise and Split-State Tampering*

Mahdi Cheraghchi
Department of Computing, Imperial College London, London, UK
m.cheraghchi@imperial.ac.uk

Venkatesan Guruswami
Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA
guruswami@cmu.edu

Communicated by Rafail Ostrovsky.

Received 5 December 2013
Online publication 6 October 2015

Abstract. Non-malleable coding, introduced by Dziembowski et al. (ICS 2010), aims for protecting the integrity of information against tampering attacks in situations where error detection is impossible. Intuitively, information encoded by a non-malleable code either decodes to the original message or, in presence of any tampering, to an unrelated message. Non-malleable coding is possible against any class of adversaries of bounded size. In particular, Dziembowski et al. show that such codes exist and may achieve positive rates for any class of tampering functions of size at most $2^{2^{\alpha n}}$, for any constant $\alpha \in [0, 1)$. However, this result is existential and has thus attracted a great deal of subsequent research on explicit constructions of non-malleable codes against natural classes of adversaries. In this work, we consider constructions of coding schemes against two well-studied classes of tampering functions; namely, bit-wise tampering functions (where the adversary tampers each bit of the encoding independently) and the much more general class of split-state adversaries (where two independent adversaries arbitrarily tamper each half of the encoded sequence). We obtain the following results for these models. (1) For bit-tampering adversaries, we obtain explicit and efficiently encodable and decodable non-malleable codes of length n achieving rate $1 - o(1)$ and error (also known as "exact security") $\exp(-\tilde{\Omega}(n^{1/7}))$. Alternatively, it is possible to improve the error to $\exp(-\tilde{\Omega}(n))$ at the cost of making the construction Monte Carlo with success probability $1 - \exp(-\Omega(n))$ (while still allowing a compact description

*A preliminary version of this article appears under the same title in proceedings of Theory of Cryptography Conference (TCC 2014) [9]; Mahdi Cheraghchi: Research supported in part by V. Guruswami's Packard Fellowship, MSR-CMU Center for Computational Thinking, and the Swiss National Science Foundation research grant PA00P2-141980. Work done for the most part while the author was with the Computer Science Department of Carnegie Mellon University and MIT Computer Science and Artificial Intelligence Laboratory; Venkatesan Guruswami: Research supported in part by the National Science Foundation under Grant No. CCF-0963975. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

of the code). Previously, the best known construction of bit-tampering coding schemes was due to Dziembowski et al. (ICS 2010), which is a Monte Carlo construction achieving rate close to .1887. (2) We initiate the study of *seedless non-malleable extractors* as a natural variation of the notion of non-malleable extractors introduced by Dodis and Wichs (STOC 2009). We show that construction of non-malleable codes for the split-state model reduces to construction of non-malleable two-source extractors. We prove a general result on existence of seedless non-malleable extractors, which implies that codes obtained from our reduction can achieve rates arbitrarily close to $1/5$ and exponentially small error. In a separate recent work, the authors show that the optimal rate in this model is $1/2$. Currently, the best known explicit construction of split-state coding schemes is due to Aggarwal, Dodis and Lovett (ECCC TR13-081) which only achieves vanishing (polynomially small) rate.

Keywords. Information theory, Tamper-resilient cryptography, Coding theory, Error detection, Randomness extractors.

1. Introduction

Non-malleable codes were introduced by Dziembowski et al. [15] as a relaxation of the classical notions of error detection and error correction. Informally, a code is non-malleable if decoding a corrupted codeword either recovers the original message, or a completely unrelated message. Non-malleable coding is a natural concept that addresses the basic question of storing messages securely on devices that may be subject to tampering, and they provide an elegant solution to the problem of protecting the integrity of data and the functionalities implemented on them against "tampering attacks" [15]. This is part of a general recent trend in theoretical cryptography to design cryptographic schemes that guarantee security even if implemented on devices that may be subject to physical tampering. The notion of non-malleable coding is inspired by the influential theme of non-malleable encryption in cryptography which guarantees the intractability of tampering the ciphertext of a message into the ciphertext encoding a related message.

The definition of non-malleable codes captures the requirement that if some adversary (with full knowledge of the code) tampers the codeword $\mathsf{Enc}(s)$ encoding a message s, corrupting it to $f(\mathsf{Enc}(s))$, he cannot control the relationship between s and the message the corrupted codeword $f(\mathsf{Enc}(s))$ encodes. For this definition to be feasible, we have to restrict the allowed tampering functions f (otherwise, the tampering function can decode the codeword to compute the original message s, flip the last bit of s to obtain a related message $\tilde{s}$, and then re-encode $\tilde{s}$), and in most interesting cases also allow the encoding to be randomized. Formally, a (binary) non-malleable code against a family of tampering functions $\mathcal{F}$ each mapping $\{0,1\}^n$ to $\{0,1\}^n$, consists of a randomized encoding function $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^n$ and a deterministic decoding function $\mathsf{Dec} : \{0,1\}^n \to \{0,1\}^k \cup \{\perp\}$ (where $\perp$ denotes error detection) which satisfy $\mathsf{Dec}(\mathsf{Enc}(s)) = s$ always, and the following non-malleability property with error ϵ: For every message $s \in \{0,1\}^k$ and every function $f \in \mathcal{F}$, the distribution of $\mathsf{Dec}(f(\mathsf{Enc}(s))$ is ϵ-close to a distribution $\mathcal{D}_f$ that depends only on f and is independent of s (ignoring the issue that f may have too many fixed points).

If some code enables error detection against some family $\mathcal{F}$, for example if $\mathcal{F}$ is the family of functions that flips between 1 and t bits and the code has minimum distance more than t, then the code is also non-malleable (by taking $\mathcal{D}_f$ to be supported entirely

on $\perp$ for all f). Error detection is also possible against the family of "additive errors," namely $\mathcal{F}_{\mathsf{add}} = \{f_\Delta \mid \Delta \in \{0,1\}^n\}$ where $f_\Delta(x) := x + \Delta$ (the addition being bit-wise XOR). Cramer et al. [12] constructed "Algebraic Manipulation Detection" (AMD) codes of rate approaching 1 such that offset by an arbitrary $\Delta \neq 0$ will be detected with high probability, thus giving a construction of non-malleable codes against $\mathcal{F}_{\mathsf{add}}$.

The notion of non-malleable coding becomes more interesting for families against which error detection is not possible. A simple example of such a class consists of all constant functions $f_c(x) := c$ for $c \in \{0,1\}^n$. Since the adversary can map all inputs to a valid codeword c^*, one cannot in general detect tampering in this situation. However, non-malleability is trivial to achieve in this case as the output distribution of a constant function is trivially independent of the message (so the rate 1 code with identity encoding function is itself non-malleable).

The original work [15] showed that non-malleable codes of positive rate exist against *every* not-too-large family $\mathcal{F}$ of tampering functions, specifically with $|\mathcal{F}| \leqslant 2^{2^{\alpha n}}$ for some constant $\alpha < 1$. In a companion paper [8], we proved that in fact one can achieve a rate approaching $1 - \alpha$ against such families, and this is best possible in that there are families of size $\approx 2^{2^{\alpha n}}$ for which non-malleable coding is not possible with rate exceeding $1 - \alpha$. (The latter is true both for random families as well as natural families such as functions that only tamper the first αn bits of the codeword.)

1.1. *Our Results*

This work is focused on two natural families of tampering functions that have been studied in the literature.

1.1.1. *Bit-Tampering Functions*

The first class consists of *bit-tampering functions* f in which the different bits of the codewords are tampered independently (i.e., each bit is either flipped, set to 0/1, or left unchanged, independent of other bits); formally $f(x) = (f_1(x_1), f_2(x_2), \ldots, f_n(x_n))$, where $f_1, \ldots, f_n \colon \{0,1\} \to \{0,1\}$. As this family is "small" (of size 4^n), by the above general results, it admits non-malleable codes with positive rate, in fact rate approaching 1 by our recent result [8].

Dziembowski et al. [15] gave a Monte Carlo construction of a non-malleable code against this family; i.e., they gave an efficient randomized algorithm to produce the code along with efficient encoding and decoding functions such that w.h.p the encoder/decoder pair ensures non-malleability against all bit-tampering functions. The rate of their construction is, however, close to .1887 and thus falls short of the "capacity" (best possible rate) for this family of tampering functions, which we now know equals 1.

Our main result in this work is the following:

Theorem 1.1. *For all integers $n \geqslant 1$, there is an explicit (deterministic) construction, with efficient encoding/decoding procedures, of a non-malleable code against bit-tampering functions that achieves rate $1 - o(1)$ and error at most $\exp(-n^{\Omega(1)})$.*

If we seek error that is $\exp(-\tilde{\Omega}(n))$, we can guarantee that with an efficient Monte Carlo construction of the code that succeeds with probability $1 - \exp(-\Omega(n))$.

The basic idea in the above construction (described in detail in Sect. 4.1) is to use a concatenation scheme with an outer code of rate close to 1 that has large relative distance and large dual relative distance, and as (constant-sized) inner codes the non-malleable codes guaranteed by the existential result (which may be deterministically found by brute-force if desired). This is inspired by the classical constructions of concatenated codes [16,18]. The outer code provides resilience against tampering functions that globally fix too many bits or alter too few. For other tampering functions, in order to prevent the tampering function from locally freezing many entire inner blocks (to possibly wrong inner codewords), the symbols of the concatenated codeword are permuted by a *pseudorandom permutation*.[1]

The seed for the permutation is itself included as the initial portion of the final codeword, after encoding by a non-malleable code (of possibly low rate). This protects the seed and ensures that any tampering of the seed portion results in the decoded permutation being essentially independent of the actual permutation, which then results in many inner blocks being error-detected (decoded to $\perp$) with noticeable probability each. The final decoder outputs $\perp$ if any inner block is decoded to $\perp$, an event which happens with essentially exponentially small probability in n with a careful choice of the parameters. The above scheme uses non-malleable codes in two places to construct the final non-malleable code, but there is no circularity because the codes for the inner blocks are of constant size, and the code protecting the seed can have very low rate (even sub-constant) as the seed can be made much smaller than the message length.

The structure of our construction bears some high level similarity to the optimal rate code construction for correcting a bounded number of additive errors in [17]. The exact details though are quite different; in particular, the crux in the analysis of [17] was ensuring that the decoder can recover the seed correctly, and toward this end the seed's encoding was distributed at random locations of the final codeword. Recovering the seed is both impossible and not needed in our context here.

1.1.2. *Split-State Adversaries*

Bit-tampering functions act on different bits independently. A much more general class of tampering functions considered in the literature [2,14,15] is the so-called *split-state model*. Here the function $f : \{0,1\}^n \to \{0,1\}^n$ must act on each half of the codeword independently (assuming n is even), but can act arbitrarily within each half. Formally, $f(x) = (f_1(x_1), f_2(x_2))$ for some functions $f_1, f_2 : \{0,1\}^{n/2} \to \{0,1\}^{n/2}$ where x_1, x_2 consist of the first $n/2$ and last $n/2$ bits of x. This represents a fairly general and useful class of adversaries which are relevant for example when the codeword is stored on two physically separate devices, and while each device may be tampered arbitrarily, the attacker of each device does not have access to contents stored on the other device.

The capacity of non-malleable coding in the split-state model equals $1/2$, as established in our recent work [8]. A natural question therefore is to construct *efficient* non-malleable codes of rate approaching $1/2$ in the split-state model (the results in [15] and [8] are existential, and the codes do not admit polynomial size representation or

[1] Throughout the paper, by pseudorandom permutation we mean t-wise independent permutation (as in Definition 2.7) for an appropriate choice of t. This should not be confused with cryptographic pseudorandom permutations, which are not used in this work.

polynomial time encoding/decoding). This remains a challenging open question, and in fact constructing a code of positive rate itself seems rather difficult. A code that encodes one-bit messages is already non-trivial, and such a code was constructed in [14] by making a connection to two-source extractors with sufficiently strong parameters and then instantiating the extractor with a construction based on the inner product function over a finite field. We stress that this connection to two-source extractor only applies to encoding one-bit messages and does not appear to generalize to longer messages.

Recently, Aggarwal et al. [2] solved the central open problem left in [14]—they construct a non-malleable code in the split-state model that works for arbitrary message length, by bringing to bear elegant techniques from additive combinatorics on the problem. The rate of their code is polynomially small: k-bit messages are encoded into codewords with $n \approx k^7$ bits.

In the second part of this article (Sect. 5), we study the problem of non-malleable coding in the split-state model. We do not offer any explicit constructions, and the polynomially small rate achieved in [2] remains the best known. Our contribution here is more conceptual. We define the notion of non-malleable two-source extractors, generalizing the influential concept of non-malleable extractors introduced by Dodis and Wichs [13]. A non-malleable extractor is a regular seeded extractor Ext whose output $\mathsf{Ext}(X, S)$ on a weak random source X and uniform random seed S remains uniform even if one knows the value $\mathsf{Ext}(X, f(S))$ for a related seed $f(S)$ where f is a tampering function with no fixed points. In a two-source non-malleable extractor, we allow both sources to be weak and independently tampered, and we further extend the definition to allow the functions to have fixed points in view of our application to non-malleable codes. We prove, however, that for construction of two-source non-malleable extractors, it suffices to only consider tampering functions that have no fixed points, at cost of a minor loss in the parameters.

We show that given a two-source non-malleable extractor NMExt with exponentially small error in the output length, one can build a non-malleable code in the split-state model by setting the extractor function NMExt to be the decoding function (the encoding of s then picks a pre-image in $\mathsf{NMExt}^{-1}(s)$).

This identifies a possibly natural avenue to construct improved non-malleable codes against split-state adversaries by constructing non-malleable two-source extractors, which seems like an interesting goal in itself. Towards confirming that this approach has the potential to lead to good non-malleable codes, we prove a fairly general existence theorem for seedless non-malleable extractors, by essentially observing that the ideas from the proof of existence of seeded non-malleable extractors in [13] can be applied in a much more general setting. Instantiating this result with split-state tampering functions, we show the existence of non-malleable two-source extractors with parameters that are strong enough to imply non-malleable codes of rate arbitrarily close to $1/5$ in the split-state model.

Explicit construction of (ordinary) two-source extractors and closely-related objects is a well-studied problem in the literature, and an abundance of explicit constructions for this problem is known[2] (see, e.g., [3,4,10,19,21,22]). The problem becomes increasingly challenging, however, (and remains open to date) when the entropy rate of the two

[2] Several of these constructions are structured enough to easily allow for efficient sampling of a uniform pre-image from $\mathsf{Ext}^{-1}(s)$.

sources may be substantially below $1/2$. Fortunately, we show that for construction of constant-rate non-malleable codes in the split-state model, it suffices to have two-source non-malleable extractors for source entropy rate .99 and with some output length $\Omega(n)$ (against tampering functions with no fixed points). Thus the infamous "$1/2$ entropy rate barrier" on two-source extractors does not concern our particular application.

The rest of this article is organized as follows. Section 2 introduces the notation and basic definition used throughout the article. In Sect. 3 we recall the existence of optimal rate non-malleable codes proved in [8], and show additional properties achieved by this construction. The construction and related properties are used as building blocks of our explicit construction. The explicit construction of optimal rate non-malleable codes against bit tampering is presented in Sect. 4, where Sect. 4.1 introduces the construction, Sect. 4.2 proves the correctness of the construction, and Sect. 4.3 sets up the parameters in order to prove the final result. Section 5 considers the more general model of split-state tampering and introduces the notion of seedless non-malleable extractors (in Sect. 5.1). Section 5.2 shows how this notion can be used to construct non-malleable coding schemes in the split-state tampering model, and Sect. 5.3 shows existence of such seedless non-malleable extractors using the probabilistic method.

Subsequent Work

After publication of the preliminary version of this work [9], numerous exciting new developments related to the work have emerged. In particular, Chattopadhyay and Zuckerman [6] use ideas from additive combinatorics to construct explicit seedless multiple-source non-malleable extractors, according to the notion of seedless non-malleable extractors defined in Sect. 4. Combining this result with the reduction discussed in Sect. 4, they obtain explicit non-malleable codes for a relaxation of the split-state model where the number of independent adversaries is lower bounded by a constant (at least 10). This model reduces to the bit tampering model when the number of independent adversaries is equal to the block length of the code, in which case the result of Chattopadhyay and Zuckerman yields explicit and rate-optimal non-malleable codes for the bit-tampering model with exponentially small error. Aggarwal et al. [1] introduce the notion of "non-malleable reductions" and in particular show that the problem of constructing explicit non-malleable codes in the standard split-state model (i.e., with two independent adversaries) can be reduced to the same problem with many independent adversaries. Combined with the explicit construction of [6], they obtain the first constant rate and explicit non-malleable codes in the split-state model with two adversaries. Finally, Chattopadhyay et al. [5] obtain, among other results, the first explicit construction of two-source non-malleable extractors which directly lead (via the reduction of Sect. 4) to non-malleable codes in the split-state model against two adversaries.

2. Preliminaries

2.1. *Notation*

We use $\mathcal{U}_n$ for the uniform distribution on $\{0, 1\}^n$ and U_n for the random variable sampled from $\mathcal{U}_n$ and independently of any existing randomness. For a random variable X, we

denote by $\mathscr{D}(X)$ the probability distribution that X is sampled from. Observe that this notation even makes sense when X only assumes a deterministic value; i.e., $X = x$ with probability 1, in which case $\mathscr{D}(x)$ would naturally be the distribution trivially supported on the singleton set $\{x\}$.

Generally, we will use calligraphic symbols (such as $\mathcal{X}$) for probability distributions and the corresponding capital letters (such as X) for related random variables. We use $X \sim \mathcal{X}$ to denote that the random variable X is drawn from the distribution $\mathcal{X}$. The statistical distance (also known as total variation distance) between two distributions $\mathcal{X}$ and $\mathcal{Y}$ over a finite probability space Ω is defined as half the ℓ_1 distance between the two distributions; i.e.,

$$\frac{1}{2} \sum_{x \in \Omega} |\mathcal{X}(x) - \mathcal{Y}(x)|,$$

where $\mathcal{X}(x)$ (resp., $\mathcal{Y}(x)$) denotes the probability assigned by $\mathcal{X}$ (resp., $\mathcal{Y}$) to the outcome x. The two distributions $\mathcal{X}$ and $\mathcal{Y}$ are called ϵ-close (resp., ϵ-far) if their statistical distance is at most (resp., at least) ϵ. It is a well-known fact that two distributions $\mathcal{X}$ and $\mathcal{Y}$ are ϵ-close if and only if for every distinguisher $h\colon \Omega \to \{0, 1\}$, and $X \sim \mathcal{X}$ and $Y \sim \mathcal{Y}$,

$$\left| \Pr_X[h(X) = 1] - \Pr_Y[h(Y) = 1] \right| \leqslant \epsilon.$$

We use the notation $\mathcal{X} \approx_\epsilon \mathcal{Y}$ to indicate that $\mathcal{X}$ and $\mathcal{Y}$ are ϵ-close. We will use $(\mathcal{X}, \mathcal{Y})$ for the product distribution with the two coordinates independently sampled from $\mathcal{X}$ and $\mathcal{Y}$. For a distribution $\mathcal{X}$ on a finite domain Ω, the *min-entropy* of $\mathcal{X}$ (in bits) is defined as

$$H_\infty(\mathcal{X}) := \min_{x \in \Omega} - \log \mathcal{X}(x),$$

All unsubscripted logarithms are taken to base 2. Support of a discrete random variable X (that is, the set of possible outcomes of X) is denoted by $\mathsf{supp}(X)$, and we naturally extend the notation to the underlying probability distribution of X. A distribution is said to be *flat* if it is uniform on its support. For a sequence $x = (x_1, \ldots, x_n)$ and set $S \subseteq [n]$, we use $x|_S$ to denote the restriction of x to the coordinate positions chosen by S. We use $\tilde{O}(\cdot)$ and $\tilde{\Omega}(\cdot)$ to denote asymptotic estimates that hide poly-logarithmic factors in the involved parameter.

2.2. *Definitions*

In this section, we review the formal definition of non-malleable codes as introduced in [15]. First, we recall the notion of *coding schemes*.

Definition 2.1. (*Coding schemes*) A pair of functions $\mathsf{Enc}\colon \{0, 1\}^k \to \{0, 1\}^n$ and $\mathsf{Dec}\colon \{0, 1\}^n \to \{0, 1\}^k \cup \{\perp\}$ where $k \leqslant n$ is said to be a coding scheme with block length n and message length k if the following conditions hold.

1. The encoder Enc is a randomized function; i.e., at each call it receives a uniformly random sequence of coin flips that the output may depend on. This random input is usually omitted from the notation and taken to be implicit. Thus for any $s \in \{0,1\}^k$, $\mathsf{Enc}(s)$ is a random variable over $\{0,1\}^n$. The decoder Dec is; however, deterministic.
2. For every $s \in \{0,1\}^k$, we have $\mathsf{Dec}(\mathsf{Enc}(s)) = s$ with probability 1.

The *rate* of the coding scheme is the ratio k/n. A coding scheme is said to have relative distance δ (or minimum distance δn), for some $\delta \in [0,1)$, if for every $s \in \{0,1\}^k$ the following holds. Let $X := \mathsf{Enc}(s)$. Then, for any $\Delta \in \{0,1\}^n$ of Hamming weight at most δn, $\mathsf{Dec}(X + \Delta) = \perp$ with probability 1. □

Before defining non-malleable coding schemes, we find it convenient to define the following notation.

Definition 2.2. For a finite set Γ, the function $\mathsf{copy}\colon (\Gamma \cup \{\underline{\mathsf{same}}\}) \times \Gamma \to \Gamma$ is defined as follows:

$$\mathsf{copy}(x, y) := \begin{cases} x & x \neq \underline{\mathsf{same}}, \\ y & x = \underline{\mathsf{same}}. \end{cases}$$

□

The notion of non-malleable coding schemes from [15] can now be rephrased as follows.

Definition 2.3. (*Non-malleability*) A coding scheme $(\mathsf{Enc}, \mathsf{Dec})$ with message length k and block length n is said to be non-malleable with error ϵ (also called *exact security*) with respect to a family $\mathcal{F}$ of tampering functions acting on $\{0,1\}^n$ (i.e., each $f \in \mathcal{F}$ maps $\{0,1\}^n$ to $\{0,1\}^n$) if for every $f \in \mathcal{F}$ there is a distribution $\mathcal{D}_f$ over $\{0,1\}^k \cup \{\perp, \underline{\mathsf{same}}\}$ such that the following holds for all $s \in \{0,1\}^k$. Define the random variable

$$S := \mathsf{Dec}(f(\mathsf{Enc}(s))),$$

and let S' be independently sampled from $\mathcal{D}_f$. Then,

$$\mathscr{D}(S) \approx_\epsilon \mathscr{D}(\mathsf{copy}(S', s)).$$

Remark 2.4. (Efficiency of sampling $\mathcal{D}_f$) The original definition of non-malleable codes in [15] also requires the distribution $\mathcal{D}_f$ to be efficiently samplable given oracle access to the tampering function f. It should be noted, however, that for any non-malleable coding scheme equipped with an efficient encoder and decoder, it can be shown that the following is a valid and efficiently samplable choice for the distribution $\mathcal{D}_f$ (possibly incurring a constant factor increase in the error parameter):

1. Let $S \sim \mathcal{U}_k$, and $X := f(\mathsf{Enc}(S))$.
2. If $\mathsf{Dec}(X) = S$, output $\underline{\mathsf{same}}$. Otherwise, output $\mathsf{Dec}(X)$.

Definition 2.5. (*Sub-cube*) A sub-cube over $\{0,1\}^n$ is a set $S \subseteq \{0,1\}^n$ such that for some $T = \{t_1, \ldots, t_\ell\} \subseteq [n]$ and $w = (w_1, \ldots, w_\ell) \in \{0,1\}^\ell$,

$$S = \{(x_1, \ldots, x_n) \in \{0,1\}^n : x_{t_1} = w_1, \ldots, x_{t_\ell} = w_\ell\};$$

the ℓ coordinates in T are said to be *frozen* and the remaining $n - \ell$ are said to be random.

Throughout the paper, we use the following notions of limited independence.

Definition 2.6. (*Limited independence of bit strings*) A distribution $\mathcal{D}$ over $\{0,1\}^n$ is said to be *ℓ-wise δ-dependent* for an integer $\ell > 0$ and parameter $\delta \in [0,1)$ if the marginal distribution of $\mathcal{D}$ restricted to any subset $T \subseteq [n]$ of the coordinate positions where $|T| \leqslant \ell$ is δ-close to $\mathcal{U}_{|T|}$. When $\delta = 0$, the distribution is ℓ-wise independent.

Definition 2.7. (*Limited independence of permutations*) The distribution of a random permutation $\Pi : [n] \to [n]$ is said to be *ℓ-wise δ-dependent* for an integer $\ell > 0$ and parameter $\delta \in [0,1)$ if for every $T \subseteq [n]$ such that $|T| \leqslant \ell$, the marginal distribution of the sequence $(\Pi(t) : t \in T)$ is δ-close to that of $(\bar{\Pi}(t) : t \in T)$, where $\bar{\Pi} : [n] \to [n]$ is a uniformly random permutation.

We will use the following notion of *Linear Error-Correcting Secret Sharing Schemes* (LECSS) as formalized by Dziembowski et al. [15] for their construction of non-malleable coding schemes against bit-tampering adversaries.

Definition 2.8. (*LECSS*) [15] A coding scheme $(\mathsf{Enc}, \mathsf{Dec})$ of block length n and message length k is a (d,t)-*Linear Error-Correcting Secret Sharing Scheme* (LECSS), for integer parameters $d, t \in [n]$ if

1. The minimum distance of the coding scheme is at least d,
2. For every message $s \in \{0,1\}^k$, the distribution of $\mathsf{Enc}(s) \in \{0,1\}^n$ is t-wise independent (as in Definition 2.6).
3. For every $w, w' \in \{0,1\}^n$ such that $\mathsf{Dec}(w) \neq \perp$ and[3] $\mathsf{Dec}(w') \neq \perp$, we have $\mathsf{Dec}(w + w') = \mathsf{Dec}(w) + \mathsf{Dec}(w')$, where we use bit-wise addition over $\mathbb{F}_2$.

3. Existence of Optimal Bit-Tampering Coding Schemes

Our main construction of explicit non-malleable codes against bit-tampering adversaries (presented in Sect. 4) uses various building blocks, the most important of which is a small inner coding scheme achieving rate close to 1 which is, in turn, non-malleable against bit-tampering adversaries. Similar to classical code concatenation techniques (e.g., [16]), as long as existence of such inner code is known, an exhaustive search can be used to find the inner coding scheme, incurring only a small cost in the overall construction time due to the assumption that the length of the inner code is sufficiently small. In fact, as it turns out, for a target overall rate of $1 - \gamma$, the length of the inner code would only depend, almost inverse linearly, on γ. In particular, if γ is an absolute positive constant, then so is the length of the inner code that is found via brute force.

[3] Although we use LECSS codes in our explicit construction, contrary to [15] we do not directly use the linearity of the code for our proof.

In this section, we recall the probabilistic construction of non-malleable codes introduced in [8] which will then be used to show existence of the inner code needed by our explicit construction. This construction, depicted as Construction 1, is defined with respect to an integer parameter $t > 0$ (which determines the number of possible codewords that correspond to each message) and a *distance parameter* $\delta \in [0, 1)$. The distance parameter determines the relative minimum distance of the code construction that will be used in the analysis of the final code.

The following, proved in [8], shows non-malleability of the probabilistic construction.

- *Given:* Integer parameters $0 < k \leqslant n$ and integer $t > 0$ such that $t2^k \leqslant 2^n$, and a distance parameter $\delta \geqslant 0$.
- *Output:* A pair of functions $\mathsf{Enc}\colon \{0,1\}^k \to \{0,1\}^n$ and $\mathsf{Dec}\colon \{0,1\}^n \to \{0,1\}^k$, where Enc may also use a uniformly random seed which is hidden from that notation, but Dec is deterministic.
- *Construction:*
  1. Let $\mathcal{N} := \{0,1\}^n$.
  2. For each $s \in \{0,1\}^k$, in an arbitrary order,
     - Let $E(s) := \emptyset$.
     - For $i \in \{1, \ldots, t\}$:
       (a) Pick a uniformly random vector $w \in \mathcal{N}$.
       (b) Add w to $E(s)$.
       (c) Let $\Gamma(w)$ be the Hamming ball of radius δn centered at w. Remove $\Gamma(w)$ from $\mathcal{N}$ (note that when $\delta = 0$, we have $\Gamma(w) = \{w\}$).
  3. Given $s \in \{0,1\}^k$, $\mathsf{Enc}(s)$ outputs an element of $E(s)$ uniformly at random.
  4. Given $w \in \{0,1\}^n$, $\mathsf{Dec}(s)$ outputs the unique s such that $w \in E(s)$, or $\perp$ if no such s exists.

Construction 1: Probabilistic construction of non-malleable codes in [8].

Theorem 3.1. ([8]) *Let $\mathcal{F}\colon \{0,1\}^n \to \{0,1\}^n$ be any family of tampering functions. For any $\epsilon, \eta > 0$, with probability at least $1 - \eta$, the coding scheme $(\mathsf{Enc}, \mathsf{Dec})$ of Construction 1 is a non-malleable code with respect to $\mathcal{F}$ and with error ϵ and relative distance δ, provided that both of the following conditions are satisfied.*

1. *$t \geqslant t_0$, for some*
$$t_0 = O\left(\frac{1}{\epsilon^6}\left(\log \frac{|\mathcal{F}|2^n}{\eta}\right)\right). \tag{1}$$
2. *$k \leqslant k_0$, for some*
$$k_0 \geqslant n(1 - h(\delta)) - \log t - 3\log(1/\epsilon) - O(1), \tag{2}$$
where $h(\cdot)$ denotes the binary entropy function.

Remark 3.2. The Proof of Theorem 3.1 explicitly defines the choice of $\mathcal{D}_f$ of Definition 2.3 to be the distribution of the following random variable:

$$D := \begin{cases} \underline{\mathsf{same}} & \text{if } f(U_n) = U_n, \\ \mathsf{Dec}(f(U_n)) & \text{if } f(U_n) \neq U_n \text{ and } f(U_n) \in H, \\ \perp & \text{otherwise,} \end{cases} \tag{3}$$

where $H \subseteq \{0,1\}^n$ is the set

$$H := \{x \in \{0,1\}^n \colon \Pr[f(U_n) = x] > 1/r\}, \tag{4}$$

for an appropriately chosen $r = \Theta(\epsilon^2 t)$.

We now instantiate the above result to the specific case of bit-tampering adversaries. Apart from non-malleability of the inner code with respect to bit-tampering adversaries, our final construction will use additional properties of the inner code that we show to be satisfied by the probabilistic construction above (Construction 1). One of these properties is what we call the *cube property*. A useful property of Construction 1 is that the decoder function maps most points of the codeword space to the error symbol $\perp$, and in that sense the code is quite sparse (i.e., the chance that a random vector turns out to be a valid codeword is small). The cube property ensures the stronger requirement that, a random string is a codeword of the inner code with probability less than $1/2$ even after an adversary fixes all but at least one of its bits to arbitrary values. In other words, the cube property ensures that the inner code remains sparse even over any non-trivial sub-cube of the codeword space. This is formalized in the lemma below.

Lemma 3.3. (Cube property) *Consider the coding scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *of Construction 1 with parameters t and δ, and assume that* $t2^{k-n(1-h(\delta))} \leqslant 1/8$, *where* $h(\cdot)$ *is the binary entropy function. Then, there is a* $\delta_0 = O(\log n/n)$ *such that if* $\delta \geqslant \delta_0$, *the following holds with probability at least* $1 - \exp(-n)$ *over the randomness of the code construction. For any sub-cube* $S \subseteq \{0,1\}^n$ *of size at least* 2, *and* $U_S \in \{0,1\}^n$ *taken uniformly at random from S,*

$$\Pr_{U_S}[\mathsf{Dec}(U_S) = \perp] \geqslant 1/2.$$

Proof. Let $S \subseteq \{0,1\}^n$ be any sub-cube, and let $\gamma := tK/2^n$, where $K := 2^k$. The assumption implies that $\gamma V \leqslant 1/8$, where $V \leqslant 2^{nh(\delta)}$ is the volume of a Hamming ball of radius δn. Let $E_1, \ldots, E_{tK}$ be the codewords chosen by the code construction in the order they are picked.

If $|S| \geqslant 2tK$, the claim obviously holds (since the total number of codewords in $\mathsf{supp}(\mathsf{Enc}(\mathcal{U}_k))$ is tK, thus we can assume otherwise.

Arbitrarily order the elements of S as $s_1, \ldots, s_{|S|}$, and for each $i \in [|S|]$, let the indicator random variable X_i be so that $X_i = 1$ iff $\mathsf{Dec}(s_i) \neq \perp$. Define $X_0 = 0$. Our goal is to upper bound

$$\mathbb{E}[X_i | X_0, \ldots, X_{i-1}]$$

for each $i \in [|S|]$. Instead of conditioning on $X_1, \ldots, X_{i-1}$, we condition on a more restricted event and show that regardless of the more restricted conditioning, the expectation of X_i can still be upper bounded as desired. Namely, we condition on the knowledge of not only $\mathsf{Dec}(s_j)$ for all $j < i$ but also the unique $j' \in [tK]$ such that $E_{j'} = s_j$, if $\mathsf{Dec}(s_j) \neq \perp$. Obviously the knowledge of this information determines the values of $X_1, \ldots, X_{i-1}$, and thus Proposition 5.15 applies. Under the more restricted conditioning, some of the codewords in $E_1, \ldots, E_{tK}$ (maybe all) will be revealed. Obviously, the revealed codewords have no chance of being assigned to s_i (since the codewords are picked without replacement). By a union bound, the chance that any of the up to tK remaining codewords is assigned to s_i by the decoder is thus at most

$$\frac{tK}{2^n - |S|V} \leqslant \frac{tK}{2^n(1 - 2\gamma V)} \leqslant (4/3)tK/2^n = (4/3)\gamma \leqslant 1/6.$$

Since the above holds for any realization of the information that we condition on, we conclude that

$$\mathbb{E}[X_i | X_0, \ldots, X_{i-1}] \leqslant 1/6.$$

Let $X := X_1 + \cdots + X_{|S|}$, which determines the number of vectors in S that are hit by the code. We can apply Proposition 5.20 to deduce that

$$\Pr[X > |S|/2] \leqslant \exp(-|S|/18).$$

Therefore, if $|S| > S_0$ for some $S_0 = O(n)$, the upper bound can be made less than $\exp(-n)3^{-n}$. In this case, a union bound on all possible sub-cubes satisfying the size lower bound ensures that the desired cube property holds for all such sub-cubes with probability at least $1 - \exp(-n)$.

The proof is now reduced to sub-cubes with at most $\delta_0 n = O(\log n)$ random bits, where we choose $\delta_0 := (\log S_0)/n$. In this case, since the relative distance of the coding scheme of Construction 1 is always at least $\delta \geqslant \delta_0$, we deduce that

$$|\{x \in S : \mathsf{Dec}(x) \neq \perp\}| \leqslant 1 \leqslant |S|/2,$$

where the first inequality is due to the minimum distance of the code and the second is due to the assumption that $|S| \geqslant 2$. Thus, whenever $2 \leqslant |S| \leqslant S_0$, we always have the property that

$$\Pr_{U_S}[\mathsf{Dec}(U_S = \perp)] \geqslant 1/2.$$

□

In addition to the cube property, our analysis of the final construction requires the inner code to satisfy a *bounded independence* property. Intuitively, bounded independence requires that the output of the encoder for any fixed message, seen as a random variable

over $\{0,1\}^n$, is nearly uniform when restricted to any small fraction of the coordinate positions. This ensures that any "local" view of the encoding of a message would not reveal any significant information about the message. The following lemma formalizes this intuition.

Lemma 3.4. (Bounded independence) *Let $\ell \in [n]$, $\epsilon > 0$ and suppose the parameters are as in Construction 1. Let $\gamma := t2^{k-n(1-h(\delta))}$, where $h(\cdot)$ denotes the binary entropy function. There is a choice of*

$$t_0 = O\Big(\frac{2^\ell + n}{\epsilon^2}\Big)$$

such that, provided that $t \geqslant t_0$, with probability $1 - \exp(-n)$ over the randomness of the code construction the coding scheme $(\mathsf{Enc}, \mathsf{Dec})$ *satisfies the following: For any $s \in \{0,1\}^k$, the random vector* $\mathsf{Enc}(s)$ *is ℓ-wise ϵ'-dependent, where*

$$\epsilon' := \max\Big\{\epsilon, \frac{2\gamma}{1-\gamma}\Big\}.$$

Proof. Consider any message $s \in \{0,1\}^k$ and suppose the t codewords in $\mathsf{supp}(\mathsf{Enc}(s))$ are denoted by $E_1, \ldots, E_t$ in the order they are picked by the construction.

Let $T \subseteq [n]$ be any set of size at most ℓ. Let $E'_1, \ldots, E'_t \in \{0,1\}^{|T|}$ be the restriction of $E_1, \ldots, E_t$ to the coordinate positions picked by T. Observe that the distribution of $\mathsf{Enc}(s)$ restricted to the coordinate positions in T is exactly the empirical distribution of the vectors $E'_1, \ldots, E'_t$, and the support size of this distribution is bounded by 2^ℓ.

Let $K := 2^k$, $N := 2^n$, and $V \leqslant 2^{nh(\delta)}$ be the volume of a Hamming ball of radius δn. By the code construction, for $i \in [t]$, conditioned on the knowledge of $E_1, \ldots, E_{i-1}$, the distribution of E_i is uniform on $\{0,1\}^n \setminus (\Gamma(E_1) \cup \ldots \cup \Gamma(E_{i-1}))$ which is a set of size at least $N(1 - tKV) \geqslant N(1-\gamma)$. By Proposition 5.16, it follows that the conditional distribution of each E_i remains $(\gamma/(1-\gamma))$-close to $\mathcal{U}_n$. Since the E'_i are simply restrictions of the E_i to some subset of the coordinates, the same holds for the E'_i; i.e., the distribution of E'_i conditioned on the knowledge of $E'_1, \ldots, E'_{i-1}$ is $(\gamma/(1-\gamma))$-close to $\mathcal{U}_{|T|}$.

Observe that $\epsilon' - \gamma/(1-\gamma) \geqslant \epsilon'/2$. By applying Lemma 5.22 to the sample outcomes $E'_1, \ldots, E'_t$, we can see that with probability at least $\exp(-3n)$ over the code construction, the empirical distribution of the E'_i is ϵ'-close to uniform provided that $t \geqslant t_0$ for some

$$t_0 = O\Big(\frac{2^\ell + n}{\epsilon'^2}\Big) = O\Big(\frac{2^\ell + n}{\epsilon^2}\Big).$$

Now, we can take a union bound on all choices of the message s and the set T and obtain the desired conclusion. □

We now put together the above results to conclude our main existence result about the codes that we will use at the "inner" level to encode blocks in our construction of non-malleable codes against bit tampering functions. Among the properties guaranteed below,

we in fact do not need the precise non-malleability property (item 2 in the statement of Lemma 3.5 below) in our eventual proof, although we use non-malleability to prove the last property (item 5) which is needed in the proof. The error-detection property ensures that any nontrivial tampering adversary can be detected by the decoder with a substantial probability (e.g., $1/3$).

Lemma 3.5. *Let $\alpha > 0$ be any parameter. Then, there is an $n_0 = O(\log^2(1/\alpha)/\alpha)$ such that for any $n \geqslant n_0$, Construction 1 can be set up so that with probability $1 - 3\exp(-n)$ over the randomness of the construction, the resulting coding scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *satisfies the following properties:*

1. *(Rate) Rate of the code is at least $1 - \alpha$.*
2. *(Non-malleability) The code is non-malleable against bit-tampering adversaries with error $\exp(-\Omega(\alpha n))$.*
3. *(Cube property) The code satisfies the* cube property *of Lemma 3.3.*
4. *(Bounded independence) For any message $s \in \{0,1\}^k$, the distribution of $\mathsf{Enc}(s)$ is $\exp(-\Omega(\alpha n))$-close to an $\Omega(\alpha n)$-wise independent distribution with uniform entries.*
5. *(Error detection) Let $f : \{0,1\}^n \to \{0,1\}^n$ be any bit-tampering adversary that is neither the identity function nor a constant function. Then, for every message $s \in \{0,1\}^k$,*

$$\Pr[\mathsf{Dec}(f(\mathsf{Enc}(s))) = \perp] \geqslant 1/3,$$

where the probability is taken over the randomness of the encoder.

Proof. Consider the family $\mathcal{F}$ of bit-tampering functions and observe that $|\mathcal{F}| = 4^n$. First, we apply Theorem 3.1 with error parameter $\epsilon := 2^{-\alpha n/27}$, distance parameter $\delta := h^{-1}(\alpha/3)$, and success parameter $\eta := \exp(-n)$. Let $N := 2^n$ and observe that $\log(N|\mathcal{F}|/\eta) = O(n)$. We choose $t = \Theta(n/\epsilon^6)$ so as to ensure that the coding scheme $(\mathsf{Enc}, \mathsf{Dec})$ is non-malleable for bit-tampering adversaries with error at most ϵ, relative distance at least δ, and message length

$$k \geqslant n(1 - h(\delta)) - 9\log(1/\epsilon) - \log n - O(1) \geqslant (1 - 2\alpha/3)n - \log n - O(1),$$

which can be made at least $n(1-\alpha)$ if $n \geqslant n_1$ for some $n_1 = O(\log(1/\alpha)/\alpha)$. This ensures that properties 1 and 2 are satisfied.

In order to ensure the cube property (property 3), we can apply Lemma 3.3. Let $K := 2^k$ and note that our choices of the parameters imply $tK/N^{1-h(\delta)} = O(\epsilon^3) \ll 1/8$. Furthermore, consider the parameter $\delta_0 = O((\log n)/n)$ of Lemma 3.3 and observe that $\alpha/3 = h(\delta) = O(\delta \log(1/\delta))$. We thus see that as long as $n \geqslant n_2$ for some $n_2 = O(\log^2(1/\alpha)/\alpha)$, we may ensure that $\delta n \geqslant \delta_0 n$. By choosing $n_0 := \max\{n_1, n_2\}$, we see that the requirements of Lemma 3.3 is satisfied, implying that with probability at least $1 - \exp(-n)$, the cube property is satisfied.

As for the bounded independence property (Property 4), consider the parameter γ of Lemma 3.4 and recall that we have shown $\gamma = O(\epsilon^3)$. Thus by Lemma 3.4, with

probability at least $1 - \exp(-n)$, every encoding $\mathsf{Enc}(s)$ is ℓ-wise $\sqrt{\epsilon}$-dependent for some

$$\ell \geqslant \log t - 2\log(1/\sqrt{\epsilon}) - \log n - O(1) \geqslant 5\log(1/\epsilon) - O(1) = \Omega(\alpha n). \quad (5)$$

Finally, we show that property 5 is implied by properties 2, 3, and 4 that we have so far shown to simultaneously hold with probability at least $1 - 3\exp(-n)$. In order to do so, we first recall that Theorem 3.1 explicitly defines the choice of $\mathcal{D}_f$ in Definition 2.3 according to (3). Let $H \subseteq \{0,1\}^n$ be the set of heavy elements as in (4) and $r = \Theta(\epsilon^2 t)$ be the corresponding threshold parameter in the same equation. Let $f : \{0,1\}^n \to \{0,1\}^n$ be any non-identity bit-tampering function and let $\ell' \in [n]$ be the number of bits that are either flipped or left unchanged by f. We consider two cases.

Case 1 $\ell' \geqslant \log r$. In this case, for every $x \in \{0,1\}^n$, we have

$$\Pr[f(\mathcal{U}_n) = x] \leqslant 2^{-\ell'} \leqslant r,$$

and thus $H = \emptyset$. Also observe that, for $U \sim \mathcal{U}_n$,

$$\Pr[f(U) = U] \leqslant 1/2,$$

the maximum being achieved when f freezes only one bit and leaves the remaining bits unchanged (in fact, if f flips any of the bits, the above probability becomes zero).
We conclude that in this case, the entire probability mass of $\mathcal{D}_f$ is supported on $\{\underline{\mathsf{same}}, \perp\}$ and the mass assigned to $\underline{\mathsf{same}}$ is at most $1/2$. Thus, by definition of non-malleability, for every message $s \in \{0,1\}^k$,

$$\Pr[\mathsf{Dec}(f(\mathsf{Enc}(s))) = \perp] \geqslant 1/2 - \epsilon \geqslant 1/3.$$

Case 2 $\ell' < \log r$. Since $r = \Theta(\epsilon^2 t)$, by plugging in the value of t we see that $r = O(n/\epsilon^4)$, and thus we know that $\ell' < \log n + 4\log(1/\epsilon) + O(1)$.
Consider any $s \in \{0,1\}^k$, and recall that, by the bounded independence property, we already know that $\mathsf{Enc}(s)$ is ℓ-wise $\sqrt{\epsilon}$-dependent. Furthermore, by (5),

$$\ell \geqslant 5\log(1/\epsilon) - O(1) \geqslant \ell',$$

where the second inequality follows by the assumed lower bound $n \geqslant n_0$ on n. We thus can use the ℓ-wise independence property of $\mathsf{Enc}(s)$ and deduce that the distribution of $f(\mathsf{Enc}(s))$ is $(\sqrt{\epsilon})$-close to the uniform distribution on a subcube $S \subseteq \{0,1\}^n$ of size at least 2. Combined with the cube property (property 3), we see that

$$\Pr[\mathsf{Dec}(f(\mathsf{Enc}(s))) = \perp] \geqslant 1/2 - \sqrt{\epsilon} \geqslant 1/3.$$

Finally, by applying a union bound on all the failure probabilities, we conclude that with probability at least $1 - 3\exp(-n)$, the code resulting from Construction 1 satisfies all the desired properties. □

4. Explicit Construction of Optimal Bit-Tampering Coding Schemes

In this section, we describe an explicit construction of codes achieving rate close to 1 that are non-malleable against bit-tampering adversaries. Throughout this section, we use N to denote the block length of the final code.

4.1. *The Construction and Underlying Intuitions*

At a high level, we combine the following tools in our construction: (1) an inner code $\mathcal{C}_0$ (with encoder Enc_0) of constant length satisfying the properties of Lemma 3.5; (2) an existing non-malleable code construction $\mathcal{C}_1$ (with encoder Enc_1) against bit-tampering achieving a possibly low (even sub-constant) rate; (3) a linear error-correcting secret sharing scheme (LECSS) $\mathcal{C}_2$ (with encoder Enc_2); (4) an explicit function Perm that, given a uniformly random seed, outputs a pseudorandom permutation (as in Definition 2.7) on a domain of size close to N. Figure 1 depicts how various components are put together to form the final code construction.

At the outer layer, LECSS is used to pre-code the message. The resulting string is then divided into blocks, where each block is subsequently encoded by the inner encoder Enc_0. For a "typical" adversary that flips or freezes a prescribed fraction of the bits, we expect many of the inner blocks to be sufficiently tampered so that many

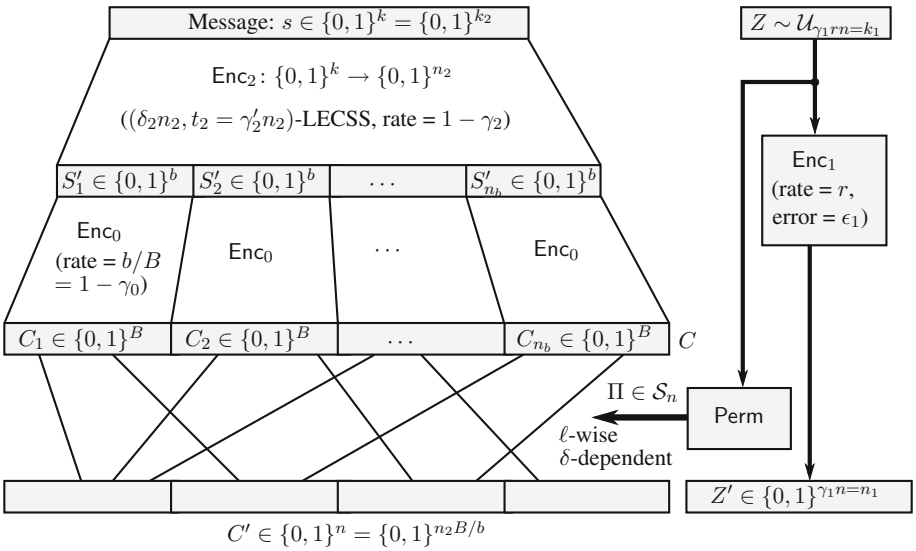

Fig. 1. Schematic description of the encoder Enc from our explicit construction.

of the inner blocks detect an error when the corresponding inner decoder is called. However, this ideal situation cannot necessarily be achieved if the fraction of global errors is too small, or if too many bits are frozen by the adversary (in particular, the adversary may freeze all but few of the blocks to valid inner codewords). In this case, we rely on the distance and bounded independence properties of LECSS to ensure that the outer decoder, given the tampered information, either detects an error or produces a distribution that is independent of the source message.

A problem with the above approach is that the adversary knows the location of various blocks and may carefully design a tampering scheme that, for example, freezes a large fraction of the blocks to valid inner codewords and leaves the rest of the blocks intact. To handle adversarial strategies of this type, we permute the final codeword using the pseudorandom permutation generated by Perm and include the seed in the final codeword. Doing so has the effect of randomizing the action of the adversary, but on the other hand creates the problem of protecting the seed against tampering. In order to solve this problem, we use the sub-optimal code $\mathcal{C}_1$ to encode the seed and prove in the analysis that non-malleability of the code $\mathcal{C}_1$ can be used to make the above intuitions work. We set up the permutation generator Perm so that the length of its seed is sufficiently small compared to the block length of the code, so that the sub-optimal rate of $\mathcal{C}_1$ would not have a significant effect on the overall rate of the final code.

The analysis given in Sect. 4.2 follows the following roadmap: Let Π be the random variable describing the pseudorandom permutation sampled by the encoder (i.e., the output of Perm given a uniformly random seed). Moreover, let $\bar{\Pi}$ be the permutation as "perceived" by the decoder; i.e., the output of the decoder of $\mathcal{C}_1$ given the (possibly tampered) portion of the codeword corresponding to the seed of Perm. We first consider three key cases in the analysis.

The first case is when the adversary freezes too many bits of the codeword. In this case, the decoder's output is a function of the frozen bits (which do not carry any information about the message), the portion of the codeword encoding the seed given to Perm (again independent of the message), and the remaining (few) bits of the encoding. We use the bounded independence property of the LECSS pre-code to show that any local view of the codeword is independent of the message. This would suffice to show that the decoding of the tampered codeword is independent of the message.

After eliminating the first case, the second case considered is when $\bar{\Pi} = \Pi$. This would be the case when the adversary does not tamper the description of the permutation. In this case, the code achieves the above-mentioned goal of permuting the action of the adversary. Therefore, assuming that the adversary does not freeze too many bits (which is taken care of by the first case) and that it does not change too few bits (also handled by the minimum distance property of the LECSS pre-code), a large number of the inner code blocks are expected to decode to the error symbol $\perp$. Thus in this case the overall code detects the tampering of the adversary with high probability.

The third case being considered is when the random variables $\bar{\Pi}$ and Π are independent; i.e., when conditioning $\bar{\Pi}$ on any fixed value does not affect the distribution of Π. This is the case, for instance, when the adversary freezes all the bits describing the seed of the permutation generator Perm, or replaces them with independent random bits. In this case, assuming that not too many bits are frozen by the adversary, we use the bounded independence and cube properties of the inner code $\mathcal{C}_0$ to show that the

decoder is able to detect tampering of the adversary at some inner code block with high probability.

Finally, we show that the general analysis reduces to the above key cases. Due to the non-malleability of the code $\mathcal{C}_1$ protecting the description of the pseudorandom permutation, the joint distribution of $(\Pi, \bar{\Pi})$ is essentially a convex combination of the second case ($\Pi = \bar{\Pi}$) and the third case (Π independent of $\bar{\Pi}$). Thus, after eliminating the case where the adversary freezes too many bits, we can combine the analysis of the second and third cases discussed above to conclude that, in general, the non-malleability requirement is satisfied for the overall code.

4.1.1. *The Building Blocks*

In the construction, we use the following building blocks, with some of the parameters to be determined later in the analysis.

1. An inner coding scheme $\mathcal{C}_0 = (\mathsf{Enc}_0, \mathsf{Dec}_0)$ with rate $1 - \gamma_0$ (for an arbitrarily small parameter $\gamma_0 > 0$), some block length B, and message length $b = (1-\gamma_0)B$. We assume that $\mathcal{C}_0$ is an instantiation of Construction 1 and satisfies the properties promised by Lemma 3.5.
2. A coding scheme $\mathcal{C}_1 = (\mathsf{Enc}_1, \mathsf{Dec}_1)$ with rate $r > 0$ (where r can in general be sub-constant), block length $n_1 := \gamma_1 n$ (where n is defined later), and message length $k_1 := \gamma_1 r n$, that is non-malleable against bit-tampering adversaries with error ϵ_1. Without loss of generality, assume that Dec_1 never outputs $\perp$ (otherwise, identify $\perp$ with an arbitrary fixed message; e.g., 0^{k_1}).
3. A linear error-correcting secret sharing (LECSS) scheme $\mathcal{C}_2 = (\mathsf{Enc}_2, \mathsf{Dec}_2)$ (as in Definition 2.8) with message length $k_2 := k$, rate $1 - \gamma_2$ (for an arbitrarily small parameter $\gamma_2 > 0$) and block length n_2. We assume that $\mathcal{C}_2$ is a $(\delta_2 n_2, t_2 := \gamma_2' n_2)$-linear error-correcting secret sharing scheme (where $\delta_2 > 0$ and $\gamma_2' > 0$ are constants defined by the choice of γ_2). Since b is a constant, without loss of generality assume that b divides n_2, and let $n_b := n_2/b$ and $n := n_2 B/b$.
4. A polynomial-time computable mapping $\mathsf{Perm}\colon \{0,1\}^{k_1} \to \mathcal{S}_n$, where $\mathcal{S}_n$ denotes the set of permutations on $[n]$. We assume that $\mathsf{Perm}(U_{k_1})$ is an ℓ-wise δ-dependent permutation (as in Definition 2.7, for parameters ℓ and δ. In fact, it is possible to achieve $\delta \leqslant \exp(-\ell)$ and $\ell = \lceil \gamma_1 r n / \log n \rceil$ for some constant $\gamma > 0$. Namely, we may use the following result due to Kaplan, Naor and Reingold [20]:

Theorem 4.1. [20] *For every integers $n, k_1 > 0$, there is a function $\mathsf{Perm}\colon \{0,1\}^{k_1} \to \mathcal{S}_n$ computable in worst-case polynomial-time (in k_1 and n) such that $\mathsf{Perm}(U_{k_1})$ is an ℓ-wise δ-dependent permutation, where $\ell = \lceil k_1 / \log n \rceil$ and $\delta \leqslant \exp(-\ell)$.* □

4.1.2. *The Encoder*

Let $s \in \{0,1\}^k$ be the message that we wish to encode. The encoder generates the encoded message $\mathsf{Enc}(s)$ according to the following procedure.

1. Let $Z \sim \mathcal{U}_{k_1}$ and sample a random permutation $\Pi\colon [n] \to [n]$ by letting $\Pi := \mathsf{Perm}(Z)$. Let $Z' := \mathsf{Enc}_1(Z) \in \{0,1\}^{\gamma_1 n}$.
2. Let $S' = \mathsf{Enc}_2(s) \in \{0,1\}^{n_2}$ be the encoding of s using the LECSS code $\mathcal{C}_2$.

3. Partition S' into blocks $S'_1, \ldots, S'_{n_b}$, each of length b, and encode each block independently using $\mathcal{C}_0$ so as to obtain a string $C = (C_1, \ldots, C_{n_b}) \in \{0,1\}^n$.
4. Let $C' := \Pi(C)$ be the string C after its n coordinates are permuted by Π.
5. Output $\mathsf{Enc}(s) := (Z', C') \in \{0,1\}^N$, where $N := (1+\gamma_1)n$, as the encoding of s.

A schematic description of the encoder summarizing the involved parameters is depicted in Fig. 1.

4.1.3. *The Decoder*

We define the decoder $\mathsf{Dec}(\bar{Z}', \bar{C}')$ as follows:

1. Compute $\bar{Z} := \mathsf{Dec}_1(\bar{Z}')$.
2. Compute the permutation $\bar{\Pi} : [n] \to [n]$ defined by $\bar{\Pi} := \mathsf{Perm}(\bar{Z})$.
3. Let $\bar{C} \in \{0,1\}^n$ be the permuted version of $\bar{C}'$ according to $\bar{\Pi}^{-1}$.
4. Partition $\bar{C}$ into n_1/b blocks $\bar{C}_1, \ldots, \bar{C}_{n_b}$ of size B each (consistent to the way that the encoder does the partitioning of C).
5. Call the inner code decoder on each block, namely, for each $i \in [n_b]$ compute $\bar{S}'_i := \mathsf{Dec}_0(\bar{C}_i)$. If $\bar{S}'_i = \perp$ for any i, output $\perp$ and return.
6. Let $\bar{S}' = (\bar{S}'_1, \ldots, \bar{S}'_{n_b}) \in \{0,1\}^{n_2}$. Compute $\bar{S} := \mathsf{Dec}_2(\bar{S}')$, where $\bar{S} = \perp$ if $\bar{S}'$ is not a codeword of $\mathcal{C}_2$. Output $\bar{S}$.

Remark 4.2. As in the classical variation of concatenated codes of Forney [16] due to Justesen [18], the encoder described above can enumerate a *family* of inner codes instead of one fixed code in order to eliminate the exhaustive search for a good inner code $\mathcal{C}_0$. In particular, one can consider all possible realizations of Construction 1 for the chosen parameters and use each obtained inner code to encode one of the n_b inner blocks. If the fraction of good inner codes (i.e., those satisfying the properties listed in Lemma 3.5) is large enough (e.g., $1 - 1/n^{\Omega(1)}$), our analysis still applies. It is possible to ensure that the size of the inner code family is not larger than n_b by appropriately choosing the parameter η in Theorem 3.1 (e.g., $\eta \geqslant 1/\sqrt{n}$).

4.2. *Analysis*

In this section, we prove that the construction of Sect. 4.1 (depicted in Fig. 1) is indeed a coding scheme that is non-malleable against bit-tampering adversaries with rate arbitrarily close to 1. More precisely, we prove the following theorem.

Theorem 4.3. *For every $\gamma_0 > 0$, there is a $\gamma'_0 = \gamma_0^{O(1)}$ and $N_0 = O(1/\gamma_0^{O(1)})$ such that for every integer $N \geqslant N_0$, the following holds.*[4] *The pair $(\mathsf{Enc}, \mathsf{Dec})$ defined in Sects. 4.1.2 and 4.1.3 can be set up to be a non-malleable coding scheme against bit-tampering adversaries, achieving block length N, rate at least $1 - \gamma_0$ and error*

[4] We can extend the construction to arbitrary block lengths N by standard padding techniques and observing that the set of block lengths for which the construction is defined is dense enough to allow padding without affecting the rate.

$$\epsilon \leqslant \epsilon_1 + 2\exp\Big(-\Omega\Big(\frac{\gamma_0' r N}{\log^3 N}\Big)\Big),$$

where r and ϵ_1 are, respectively, the rate and the error of the assumed non-malleable coding scheme $\mathcal{C}_1$.

Remark 4.4. Dziembowski et al. [15, Definition 3.3] also introduce a "strong" variation of non-malleable codes which implies the standard definition (Definition 2.3) but is more restrictive. It can be argued that the stronger definition is less natural in the sense that an error-correcting code that is able to fully correct the tampering incurred by the adversary does not satisfy the stronger definition, while it is non-malleable in the standard sense, which is what naturally expected to be the case. In this work, we focus on the standard definition and prove the results with respect to Definition 2.3. However, it can be verified (by minor adjustments of the Proof of Theorem 4.3) that the construction of this section satisfies strong non-malleability (without any loss in the parameters) as well provided that the non-malleable code $(\mathsf{Enc}_1, \mathsf{Dec}_1)$ encoding the description of the permutation Π satisfies the strong definition.

Proof of Theorem 4.3

It is clear that, given (Z', C'), the decoder can unambiguously reconstruct the message s; that is, $\mathsf{Dec}(\mathsf{Enc}(s)) = s$ with probability 1. Thus, it remains to demonstrate non-malleability of $\mathsf{Enc}(s)$ against bit-tampering adversaries.

Fix any such adversary $f : \{0,1\}^N \to \{0,1\}^N$. The adversary f defines the following partition of $[N]$:

- $\mathsf{Fr} \subseteq [N]$; the set of positions frozen to either zero or one by f.
- $\mathsf{Fl} \subseteq [N] \backslash \mathsf{Fr}$; the set of positions flipped by f.
- $\mathsf{Id} = [N] \backslash (\mathsf{Fr} \cup \mathsf{Fl})$; the set of positions left unchanged by f.

Since f is not the identity function (otherwise, there is nothing to prove), we know that $\mathsf{Fr} \cup \mathsf{Fl} \neq \emptyset$.

We use the notation used in the description of the encoder Enc and decoder Dec for various random variables involved in the encoding and decoding of the message s. In particular, let $(\bar{Z}', \bar{C}') = f(Z', C')$ denote the perturbation of $\mathsf{Enc}(s)$ by the adversary, and let $\bar{\Pi} := \mathsf{Perm}(\mathsf{Dec}_1(\bar{Z}'))$ be the induced perturbation of Π as viewed by the decoder Dec. In general Π and $\bar{\Pi}$ are correlated random variables, but independent of the remaining randomness used by the encoder.

We first distinguish three cases and subsequently use a convex combination argument to show that the analysis of these cases suffices to guarantee non-malleability in general. The first case considers the situation where the adversary freezes too many bits of the encoding. The remaining two cases can thus assume that a sizeable fraction of the bits are not frozen to fixed values.

Case 1: Too Many Bits of C' are Frozen by the Adversary

First, assume that f freezes at least $n - t_2/b$ of the n bits of C'. In this case, we show that the distribution of $\mathsf{Dec}(f(Z', C'))$ is always independent of the message s and thus

the non-malleability condition of Definition 2.3 is satisfied for the chosen f. In order to achieve this goal, we rely on bounded independence property of the LECSS code $\mathcal{C}_2$. We remark that a similar technique has been used in [15] for their construction of non-malleable codes (and for the case where the adversary freezes too many bits).

Observe that the joint distribution of $(\Pi, \bar{\Pi})$ is independent of the message s. Thus it suffices to show that conditioned on any realization $\Pi = \pi$ and $\bar{\Pi} = \bar{\pi}$, for any fixed permutations π and $\bar{\pi}$, the conditional distribution of $\mathsf{Dec}(f(Z', C'))$ is independent of the message s.

We wish to understand how, with respect to the particular permutations defined by π and $\bar{\pi}$, the adversary acts on the bits of the inner code blocks $C = (C_1, \ldots, C_{n_b})$.

Consider the set $T \subseteq [n_b]$ of the blocks of $C = (C_1, \ldots, C_{n_b})$ (as defined in the algorithm for Enc) that are not completely frozen by f (after permuting the action of f with respect to the fixed choice of π). We know that $|T| \leqslant t_2/b$.

Let S'_T be the string $S' = (S'_1, \ldots, S'_{n_b})$ (as defined in the algorithm for Enc) restricted to the blocks defined by T; that is, $S'_T := (S'_i)_{i \in T}$. Observe that the length of S'_T is at most $b|T| \leqslant t_2$. From the t_2-wise independence property of the LECSS code $\mathcal{C}_2$, and the fact that the randomness of Enc_2 is independent of $(\Pi, \bar{\Pi})$, we know that S'_T is a uniform string, and in particular, independent of the original message s. Let C_T be the restriction of C to the blocks defined by T; that is, $C_T := (C_i)_{i \in T}$. Since C_T is generated from S_T (by applying the encoder Enc_0 on each block, whose randomness is independent of $(\Pi, \bar{\Pi})$), we know that the distribution of C_T is independent of the original message s as well.

Now, observe that $\mathsf{Dec}(f(Z', C'))$ is only a function of T, C_T, the tampering function f and the fixed choices of π and $\bar{\pi}$ (since the bits of C that are not picked by T are frozen to values determined by the tampering function f), which are all independent of the message s. Thus in this case, $\mathsf{Dec}(f(Z', C'))$ is independent of s as well. This suffices to prove non-malleability of the code in this case. In particular, in Definition 2.3, we can take $\mathcal{D}_f$ to be the distribution of $\mathsf{Dec}(f(Z', C'))$ for an arbitrary message and satisfy the definition with zero error.

Case 2: The Adversary Does not Alter Π

In this case, we assume that $\Pi = \bar{\Pi}$, both distributed according to $\mathsf{Perm}(\mathcal{U}_{k_1})$ and independently of the remaining randomness used by the encoder. This situation in particular occurs if the adversary leaves the part of the encoding corresponding to Z' completely unchanged. We furthermore assume that Case 1 does not occur; i.e., more than $t_2/b = \gamma'_2 n_2/b$ bits of C' are not frozen by the adversary. To analyze this case, we rely on bounded independence of the permutation Π. The effect of the randomness of Π is to prevent the adversary from gaining any advantage of the fact that the inner code independently acts on the individual blocks.

Let $\mathsf{Id}' \subseteq \mathsf{Id}$ be the positions of C' that are left unchanged by f. Similarly, let $\mathsf{Fl}' \subseteq \mathsf{Fl}$ and $\mathsf{Fr}' \subseteq \mathsf{Fr}$, respectively, denote the positions of C' that are flipped and frozen by f. Since we have eliminated the case where too many bits of C' are frozen, we may assume that $|\mathsf{Id}' \cup \mathsf{Fl}'| > t_2/b$, or equivalently,

$$|\mathsf{Fr}'| < n - t_2/b. \tag{6}$$

Recall that the adversary freezes the bits of C corresponding to the positions in $\Pi^{-1}(\mathsf{Fr}')$ and either flips or leaves the rest of the bits of C unchanged. We consider two sub-cases.

Case 2.1: $|\mathsf{Id}'| > n - \delta_2 n_b$

In this case, all but less than $\delta_2 n_b$ of the inner code blocks are decoded to the correct values by the decoder. Thus, the decoder correctly reconstructs all but less than $b(n - |\mathsf{Id}'|) \leqslant \delta_2 n_2$ bits of S'. Now, the distance property of the LECSS code $\mathcal{C}_2$ ensures that occurrence of any errors in S' can be detected by the decoder. Roughly speaking, this means that the decoder would either output the correct message or the error symbol $\perp$, and thus the distribution $\mathcal{D}_f$ should be only supported on $\{\underline{\mathsf{same}}, \perp\}$. However, more work is needed to ensure that the probability of the decoder outputting the error symbol is not sensitive to the choice of the original message s.

Let $T_0 \subseteq [n_b]$ be the set of blocks of C that are affected by the action of f (that is, those blocks in which there is a position $i \in [n]$ where $\Pi(i) \notin \mathsf{Id}$), and $T_1 \subseteq [n_2]$ (resp., $T_2 \subseteq [n]$) be the coordinate positions of S' (resp., C) contained in the blocks defined by T_0. Observe that $|T_0| < \delta_2 n_b$, $|T_1| = b|T_0| < \delta_2 n_2$ and $|T_2| = B|T_0|$.

The bounded independence property of $\mathcal{C}_2$ ensures that the restriction of S' to the positions in T_1 is uniformly distributed, provided that

$$\gamma_2' \geqslant \delta_2 \tag{7}$$

that we will assume in the sequel. Consequently, the restriction of C to the positions in T_2 has the exact same distribution regardless of the encoded message s.

Recall that the decoder either outputs the correct message s or $\perp$, and the former happens if and only if S' is correctly decoded at the positions in T_1. This event (that is, $\bar{S}'|_{T_1} = S'|_{T_1}$) is independent of the encoded message s, since the estimate $\bar{S}'|_{T_1}$ is completely determined by $S'|_{T_1}$, Π, and f, which are all independent of s. Thus, the probability of the decoder outputting $\perp$ is the same regardless of the message s. Since the decoder either outputs the correct s or $\perp$, we can conclude non-malleability of the code in this case is achieved with zero error and a distribution $\mathcal{D}_f$ that is only supported on $\{\underline{\mathsf{same}}, \perp\}$.

Case 2.2: $|\mathsf{Id}'| \leqslant n - \delta_2 n_b$

In this case, we have $|\mathsf{Fr}' \cup \mathsf{Fl}'| \geqslant \delta_2 n_2 / b$. Moreover, we fix the randomness of the LECSS code $\mathcal{C}_2$ so that S' becomes a fixed string. Recall that $C_1, \ldots, C_{n_b}$ are independent random variables, since every call of the inner encoder Enc_0 uses fresh randomness. In this case, our goal is to show that the decoder outputs $\perp$ with high probability, thus ensuring non-malleability by choosing $\mathcal{D}_f$ to be the singleton distribution on $\{\perp\}$.

Since $\Pi = \bar{\Pi}$, the decoder is able to correctly identify positions of all the inner code blocks determined by C. In other words, we have

$$\bar{C} = f'(C),$$

where f' denotes the adversary obtained from f by permuting its action on the bits as defined by Π^{-1}; that is,

$$f'(x) := \Pi^{-1}(f(\Pi(x))).$$

Let $i \in [n_b]$. We consider the dependence between C_i and its tampering $\bar{C}_i$, conditioned on the knowledge of Π on the first $i-1$ blocks of C. Let $C(j)$ denote the jth bit of C, so that the ith block of C becomes $(C(1+(i-1)B), \ldots, C(iB))$. For the moment, assume that $\delta = 0$; that is, assume that Π is exactly an ℓ-wise independent permutation.

Suppose $iB \leqslant \ell$, meaning that the restriction of Π on the ith block (i.e., $(\Pi(1+(i-1)B), \ldots, \Pi(iB))$ conditioned on any fixing of $(\Pi(1), \ldots, \Pi((i-1)B))$ exhibits the same distribution as that of a uniformly random permutation.

We define events $\mathcal{E}_1$ and $\mathcal{E}_2$ as follows. $\mathcal{E}_1$ is the event that $\Pi(1+(i-1)B) \notin \mathsf{Id}'$, and $\mathcal{E}_2$ is the event that $\Pi(2+(i-1)B) \notin \mathsf{Fr}'$. That is, $\mathcal{E}_1$ occurs when the adversary does not leave the first bit of the ith block of C intact, and $\mathcal{E}_2$ occurs when the adversary does not freeze the second bit of the ith block. We are interested in lower bounding the probability that both $\mathcal{E}_1$ and $\mathcal{E}_2$ occur, conditioned on any particular realization of $(\Pi(1), \ldots, \Pi((i-1)B))$.

Suppose the parameters are set up so that

$$\ell \leqslant \frac{1}{2}\min\{\delta_2 n_2/b, \gamma_2' n_2/b\}. \tag{8}$$

Under this assumption, we show that even conditioned on any fixing of $(\Pi(1), \ldots, \Pi((i-1)B))$, we can ensure that

$$\Pr[\mathcal{E}_1] \geqslant \delta_2 n_2/(2bn), \tag{9}$$

and

$$\Pr[\mathcal{E}_2|\mathcal{E}_1] \geqslant \gamma_2' n_2/(2bn). \tag{10}$$

To see (9), note that among the particular outcomes of $\Pi(1), \ldots, \Pi((i-1)B)$, at most $(i-1)B < \ell$ can fall outside Id'. Since we have assumed that the distribution of $\Pi(1+(i-1)B)$ remains uniformly random conditioned on $(\Pi(1), \ldots, \Pi((i-1)B))$, it follows that

$$\begin{aligned}
\Pr[\mathcal{E}_1] &\geqslant \frac{(n - |\mathsf{Id}'|) - \ell}{n} \\
&\geqslant \frac{\delta_2 n_b}{n} - \frac{\ell}{n} \\
&\overset{(8)}{\geqslant} \frac{\delta_2 n_b}{n} - \frac{\delta_2 n_2}{2bn} \\
&= \frac{\delta_2 n_2}{2bn},
\end{aligned}$$

where for the last equality we recall that $n_b = n_2/b$.

Similarly, in order to verify (10) we note that among the particular outcomes of $\Pi(1), \ldots, \Pi((i-1)B)$, $\Pi(1+(i-1)B)$, at most $(i-1)B+1 \leqslant \ell$ can fall outside Fr'.

Again we recall that the distribution of $\Pi(2+(i-1)B)$ is uniformly random conditioned on $(\Pi(1), \ldots, \Pi((i-1)B), \Pi(1+(i-1)B))$ and write

$$\begin{aligned}\Pr[\mathcal{E}_2|\mathcal{E}_1] &\geqslant \frac{(n-|\mathsf{Fr}'|)-\ell}{n} \\ &\overset{(6)}{\geqslant} \frac{t_2}{bn} - \frac{\ell}{n} \\ &\overset{(8)}{\geqslant} \frac{t_2}{bn} - \frac{\gamma_2' n_2}{2bn} \\ &= \frac{\gamma_2' n_2}{2bn}.\end{aligned}$$

Note that (9) and (10) together imply that

$$\Pr[\mathcal{E}_1 \wedge \mathcal{E}_2] = \Pr[\mathcal{E}_1] \cdot \Pr[\mathcal{E}_2|\mathcal{E}_1] \geqslant \delta_2 \gamma_2' \left(\frac{n_2}{2bn}\right)^2 =: \gamma_2''. \tag{11}$$

We let γ_2'' to be the right-hand side of the above inequality.

In general, when the random permutation is ℓ-wise δ-dependent for $\delta \geqslant 0$, the above probability lower bound in (11) can only be affected by at most δ (by the definition of statistical distance). Thus, under the assumption that

$$\delta \leqslant \gamma_2''/2, \tag{12}$$

we may still ensure that

$$\Pr[\mathcal{E}_1 \wedge \mathcal{E}_2] \geqslant \gamma'' - \delta \geqslant \gamma_2''/2. \tag{13}$$

Let $X_i \in \{0, 1\}$ indicate the event that $\mathsf{Dec}_0(\bar{C}_i) = \perp$. We can write

$$\Pr[X_i = 1] \geqslant \Pr[X_i = 1|\mathcal{E}_1 \wedge \mathcal{E}_2] \Pr[\mathcal{E}_1 \wedge \mathcal{E}_2] \geqslant (\gamma_2''/2) \Pr[X_i = 1|\mathcal{E}_1 \wedge \mathcal{E}_2],$$

where the last inequality follows from (13). However, by property 5 of Lemma 3.5 (error detection) that is attained by the inner code $\mathcal{C}_0$, we also know that

$$\Pr[X_i = 1|\mathcal{E}_1 \wedge \mathcal{E}_2] \geqslant 1/3,$$

and therefore it follows that

$$\Pr[X_i = 1] \geqslant \gamma_2''/6. \tag{14}$$

Observe that by the argument above, (14) holds even conditioned on the realization of the permutation Π on the first $i-1$ blocks of C. By recalling that we have fixed the randomness of Enc_2, and that each inner block is independently encoded by Enc_0, we can deduce that, letting $X_0 := 0$,

$$\Pr[X_i = 1|X_0, \ldots, X_{i-1}] \geqslant \gamma_2''/6. \tag{15}$$

Using the above result for all $i \in \{1, \ldots, \lfloor \ell/B \rfloor\}$, we conclude that

$$\Pr[\mathsf{Dec}(\bar{Z}', \bar{C}') \neq \perp] \leqslant \Pr[X_1 = X_2 = \cdots = X_{\lfloor \ell/B \rfloor} = 0] \tag{16}$$

$$\leqslant \left(1 - \gamma_2''/6\right)^{\lfloor \ell/B \rfloor}, \tag{17}$$

where (16) holds since the left-hand side event is a subset of the right-hand side event, and (17) follows from (15) and the chain rule.

Thus, by appropriately setting the parameters as we will do later, we can ensure that the decoder outputs $\perp$ with high probability. This ensures non-malleability of the code in this case with the choice of $\mathcal{D}_f$ in Definition 2.3 being entirely supported on $\{\perp\}$ and error bounded by the right hand side of (17).

Case 3: The Decoder Estimates an Independent Permutation

In this case, we consider the event that $\bar{\Pi}$ attains a particular value $\bar{\pi}$. Suppose it so happens that under this conditioning, the distribution of Π remains unaffected; that is, $\bar{\Pi} = \pi$ and $\Pi \sim \mathsf{Perm}(\mathcal{U}_{k_1})$. This situation may occur if the adversary completely freezes the part of the encoding corresponding to Z' to a fixed valid codeword of $\mathcal{C}_1$. Recall that the random variable Π is determined by the random string Z and that it is independent of the remaining randomness used by the encoder Enc. Similar to the previous case, our goal is to upper bound the probability that Dec does not output $\perp$. Furthermore, we can again assume that Case 1 does not occur; i.e., more than t_2/b bits of C' are not frozen by the adversary. For the analysis of this case, we can fix the randomness of Enc_2 and thus assume that S' is fixed to a particular value.

As before, our goal is to determine how each block C_i of the inner code is related to its perturbation $\bar{C}_i$ induced by the adversary. Recall that

$$\bar{C} = \bar{\pi}^{-1}(f(\Pi(C))).$$

We observe that, without loss of generality, we can assume that $\bar{\pi}$ is the identity permutation, which would substantially clean up the notation in the analysis. To see this, first note that for any fixed permutation $\sigma : [n] \to [n]$, the non-malleability analysis for some joint distribution of permutations $(\Pi, \bar{\Pi})$ and a bit-wise tampering adversary $f(x)$ is equivalent to the analysis with respect to joint distribution of permutations $(\sigma \circ \Pi, \sigma \circ \bar{\Pi})$ and bit-wise tampering adversary $f'(x) := \sigma(f(\sigma^{-1}(x)))$. That is, if the bit-wise tampering function f is replaced by f' (that simply permutes the action of adversary with respect to σ), the non-malleability requirement would be satisfied with respect to f if and only if it is satisfied with respect to f' when the encoder uses permutation $\sigma \circ \Pi$ instead of Π and the decoder perceives the permutation $\sigma \circ \bar{\Pi}$ instead of $\bar{\Pi}$ (or in other words, the components used by the analysis, that is the adversary and permutations used by the encoder and decoder, are all permuted with respect to the same permutation σ). In the present case, we may take $\sigma := \bar{\pi}^{-1}$ so that $\sigma \circ \bar{\Pi}$ becomes the identity permutation and observe that 1) the distribution of $\sigma \circ \Pi$ remains ℓ-wise δ-dependent and 2) the bit-wise tampering adversary $f'(x)$ only permutes the action of the original tampering function f, resulting in the same number of frozen, unchanged, and flipped bits.

Fixing $\bar{\pi}$ to the identity permutation allows us to simplify $\bar{C}' = \bar{C}$ (since $\bar{C}' = \bar{\pi}(\bar{C})$), and

$$\bar{C} = f(\Pi(C)).$$

For any $\tau \in [n_b]$, let $f_\tau \colon \{0,1\}^B \to \{0,1\}^B$ denote the restriction of the adversary to the positions included in the τth block of $\bar{C}$.

Assuming that $\ell \leqslant t_2$ (which is implied by (8)), let $T \subseteq [n]$ be any set of size $\lfloor \ell/B \rfloor \leqslant \lfloor t_2/B \rfloor \leqslant t_2/b$ of the coordinate positions of C' that are either left unchanged or flipped by f. Let $T' \subseteq [n_b]$ (where $|T'| \leqslant |T|$) be the set of blocks of $\bar{C}$ that contain the positions picked by T. With slight abuse of notation, for any $\tau \in T'$, denote by $\Pi^{-1}(\tau) \subseteq [n]$ the set of indices of the positions belonging to the block τ after applying the permutation Π^{-1} to each one of them. In other words, $\bar{C}_\tau$ (the τth block of $\bar{C}$) is determined by taking the restriction of C to the bits in $\Pi^{-1}(\tau)$ (in their respective order), and applying f_τ on those bits (recall that for $\tau \in T'$ we are guaranteed that f_τ does not freeze all the bits).

In the sequel, our goal is to show that with high probability, $\mathsf{Dec}(\bar{Z}, \bar{C}') = \perp$. In order to do so, we first assume that $\delta = 0$; i.e., that Π is exactly an ℓ-wise independent permutation. Suppose $T' = \{\tau_1, \ldots, \tau_{|T'|}\}$, and consider any $i \in |T'|$.

We wish to lower bound the probability that $\mathsf{Dec}_0(\bar{C}_{\tau_i}) = \perp$, conditioned on the knowledge of Π on the first $i-1$ blocks in T'. Subject to the conditioning, the values of Π becomes known on up to $(i-1)B \leqslant (|T'|-1)B \leqslant \ell - B$ points. Since Π is ℓ-wise independent, Π on the B bits belonging to the ith block remains B-wise independent. Now, assuming

$$\ell \leqslant n/2, \tag{18}$$

we know that even subject to the knowledge of Π on any ℓ positions of C, the probability that a uniformly random element within the remaining positions falls in a particular block of C is at most $B/(n-\ell) \leqslant 2B/n$.

Now, for $j \in \{2, \ldots, B\}$, consider the jth position of the block τ_i in T'. By the above argument, the probability that Π^{-1} maps this element to a block of C chosen by any of the previous $j-1$ elements is at most $2B/n$. By a union bound on the choices of j, with probability at least

$$1 - 2B^2/n,$$

the elements of the block τ_i all land in distinct blocks of C by the permutation Π^{-1}. Now we observe that if $\delta > 0$, the above probability is only affected by at most δ. Moreover, if the above distinctness property occurs, the values of C at the positions in $\Pi^{-1}(\tau)$ become independent random bits; since Enc uses fresh randomness upon each call of Enc_0 for encoding different blocks of the inner code (recall that the randomness of the first layer using Enc_2 is fixed).

Recall that by the bounded independence property of $\mathcal{C}_0$ (i.e., property 4 of Lemma 3.5), each individual bit of C is $\exp(-\Omega(\gamma_0 B))$-close to uniform. Therefore, using Proposition 5.19, with probability at least $1 - 2B^2/n - \delta$ (in particular, at least $7/8$ when

$$n \geqslant 32B^2 \tag{19}$$

and assuming $\delta \leqslant 1/16$) we can ensure that the distribution of C restricted to positions picked by $\Pi^{-1}(\tau)$ is $O(B\exp(-\Omega(\gamma_0 B)))$-close to uniform, or in particular $(1/4)$-close to uniform when B is larger than a suitable constant. If this happens, we can conclude that distribution of the block τ_i of $\bar{C}$ is $(1/4)$-close to a sub-cube with at least one random bit (since we have assumed that $\tau \in T'$ and thus f does not fix all the bit of the τth block). Now, the cube property of $\mathcal{C}_0$ (i.e., property 3 of Lemma 3.5) implies that

$$\Pr_{\mathsf{Enc}_0}[\mathsf{Dec}_0(\bar{C}_{\tau_i}) \neq \perp \mid \Pi(\tau_1), \ldots, \Pi(\tau_{i-1})] \leqslant 1/2 + 1/4 = 3/4,$$

where the extra term $1/4$ accounts for the statistical distance of $\bar{C}_{\tau_i}$ from being a perfect sub-cube.

Finally, using the above probability bound, and running i over all the blocks in T', and recalling the assumption that $\bar{C} = \bar{C}'$, we deduce that

$$\Pr[\mathsf{Dec}(\bar{Z}', \bar{C}') \neq \perp] \leqslant (7/8)^{|T'|} \leqslant \exp(-\Omega(\ell/B^2)), \tag{20}$$

where the last inequality follows from the fact that $|T'| \geqslant \lfloor \ell/b \rfloor / B$.

In a similar way to Case 2.2 above, this concludes non-malleability of the code in this case with the choice of $\mathcal{D}_f$ in Definition 2.3 being entirely supported on $\{\perp\}$ and error bounded by the right-hand side of (20).

The General Case

Recall that Case 1 eliminates the situation in which the adversary freezes too many of the bits. For the remaining cases, Cases 2 and 3 consider the special situations where the two permutations Π and $\bar{\Pi}$ used by the encoder and the decoder either completely match or are completely independent. However, in general we may not reach any of the two cases. Fortunately, the fact that the code $\mathcal{C}_1$ encoding the permutation Π is non-malleable ensures that we always end up with a *combination* of the Case 2 and 3. In other words, in order to analyze any event depending on the joint distribution of $(\Pi, \bar{\Pi})$, it suffices to consider the two special cases where Π is always the same as $\bar{\Pi}$, or when Π and $\bar{\Pi}$ are fully independent.

The joint distribution of $(\Pi, \bar{\Pi})$ may be understood using Lemma 5.18. Namely, the lemma applied on the non-malleable code $\mathcal{C}_1$ implies that the joint distribution of $(\Pi, \bar{\Pi})$ is ϵ_1-close (recall that ϵ_1 is the error of non-malleable code $\mathcal{C}_1$) to the convex combination

$$\alpha \cdot \mathscr{D}(\Pi, \Pi) + (1-\alpha) \cdot \mathscr{D}(\Pi, \Pi'), \tag{21}$$

for some parameter $\alpha \in [0, 1]$ and an independent random variable Π' distributed over $\mathcal{S}_n$.

For a random variable $\bar{P}$ jointly distributed with Π over $\mathcal{S}_n$, and with a slight overload of notation, define the random variable $D_{s,\bar{P}}$ over $\{0,1\}^k \cup \{\perp\}$ as the output of the following experiment (recall that $s \in \{0,1\}^k$ is the message to be encoded):

1. Let $(Z', C') := \mathsf{Enc}(s)$ be the encoding $\mathsf{Enc}(s)$, as described in Sect. 4.1.2, and $(\bar{Z}', \bar{C}') = f(Z', C')$ be the corrupted codeword under the adversary f.
2. Apply the decoder's procedure, described in Sect. 4.1.3, where in the second line of the procedure the assignment $\bar{\Pi} := \mathsf{Perm}(\bar{Z})$ is replaced with $\bar{\Pi} := \bar{P}$, and output the result.

Intuitively, $D_{s,\bar{P}}$ captures decoding of the perturbed codeword when the decoder uses an arbitrary estimate $\bar{P}$ (given in the subscript) of the random permutation Π instead of reading it off the codeword (i.e., instead of $\bar{\Pi} := \mathsf{Perm}(\bar{Z})$ defined by the decoder's procedure). Using this notation, $D_{s,\bar{\Pi}}$ (that is, when the choice of $\bar{P}$ is indeed the natural estimate $\mathsf{Perm}(\bar{Z})$) is the same as $\mathsf{Dec}(f(\mathsf{Enc}(s)))$.

Define $D_s := D_{s,\bar{\Pi}}$, $D'_s := D_{s,\Pi}$ and $D''_s := D_{s,\Pi'}$. The results obtained in Cases 2 and 3 can be summarized as follows:

- (Case 2): There is a distribution $\mathcal{D}'_f$ over $\{0,1\}^k \cup \{\underline{\mathsf{same}}, \perp\}$ such that the statistical distance between the distribution of D'_s and $\mathsf{copy}(\mathcal{D}'_f, s)$ (that is, the distribution obtained by reassigning the mass of $\underline{\mathsf{same}}$ in $\mathcal{D}'_f$ to s) is at most

$$\left(1 - \gamma''_2/6\right)^{\lfloor \ell/B \rfloor}$$

  (in fact, $\mathcal{D}'_s$ is entirely supported on $\{\perp, \underline{\mathsf{same}}\}$).
- (Case 3): There is a distribution $\mathcal{D}''_f$ over $\{0,1\}^k \cup \{\underline{\mathsf{same}}, \perp\}$ such that the statistical distance between the distribution of D''_s and $\mathsf{copy}(\mathcal{D}''_f, s)$ is at most

$$\exp(-\Omega(\ell/B^2))$$

  (in fact, $\mathcal{D}''_s$ is the distribution entirely supported on $\{\perp\}$).

The convex decomposition (21) implies that the distribution of D_s may be decomposed as a convex combination as well, that is (recalling that the action of f on the part of the codeword corresponding to C' is independent of Z' and that the randomness used by Enc_1 is independent of the randomness used by Enc_2 and each invocation of Enc_0),

$$\mathscr{D}(D_s) \approx_{\epsilon_1} \alpha \mathscr{D}(D'_s) + (1-\alpha)\mathscr{D}(D''_s). \tag{22}$$

Now we set

$$\mathcal{D}_f := \alpha \mathcal{D}'_f + (1-\alpha)\mathcal{D}''_f$$

and define

$$\epsilon' := \left(1 - \gamma''_2/6\right)^{\lfloor \ell/B \rfloor} + \exp(-\Omega(\ell/B^2)) + \epsilon_1. \tag{23}$$

From the above observations, it follows that the distribution of D_s (equivalently, $\mathsf{Dec}(f(\mathsf{Enc}(s)))$) is ϵ'-close to $\mathsf{copy}(\mathcal{D}_f, s)$. This proves non-malleability of the code in the general case with error bounded by ϵ'.

Setting up the Parameters

The final encoder Enc maps k bits into

$$\Big(\frac{k}{1-\gamma_2}\cdot\frac{1}{1-\gamma_0}\Big)(1+\gamma_1)$$

bits. Thus the rate R of the final code is

$$R=\frac{(1-\gamma_0)(1-\gamma_2)}{1+\gamma_1}.$$

We set up $\gamma_1,\gamma_2\in[\gamma_0/2,\gamma_0]$ so as to ensure that

$$R\geqslant 1-O(\gamma_0).$$

Thus, the rate of the final code can be made arbitrarily close to 1 if γ_0 is chosen to be a sufficiently small constant.

Before proceeding with the choice of other parameters, we recap the constraints that we have assumed on the parameters; namely, (7), (25), (26), (18), (12) (where we recall that $\gamma_2''=\delta_2\gamma_2'\left(\frac{n_2}{2bn}\right)^2$) which are again listed below to assist the reader.

$$\gamma_2'\geqslant\delta_2 \tag{24}$$
$$n\geqslant 32B^2, \tag{25}$$
$$\ell\leqslant\frac{1}{2}\min\{\delta_2 n_2/b,\gamma_2' n_2/b\}, \tag{26}$$
$$\ell\leqslant n/2, \tag{27}$$
$$\delta\leqslant\gamma_2''/2, \tag{28}$$

For the particular choice of γ_0, there is a constant

$$B=O((\log^2\gamma_0)/\gamma_0) \tag{29}$$

for which Lemma 3.5 holds.

Note that the choice of B only depends on the constant γ_0. If desired, a brute-force search[5] can thus find an explicit choice for the inner code $\mathcal{C}_0$ in time only depending on γ_0. Moreover, (25) can be satisfied as long as $N\geqslant N_0$ for some $N_0=\mathsf{poly}(1/\gamma_0)$.

Now, for the assumed value for the constant $\gamma_2\approx\gamma_0$, one can use Corollary 5.14 and set up $\mathcal{C}_2$ to be an $(\Omega(\gamma_0 n_2/\log n_2),\Omega(\gamma_0 n_2/\log n_2))$-linear error-correcting secret sharing code. Thus, we may assume that $\delta_2=\gamma_2'=\Omega(\gamma_0/\log N)$ (since, trivially, $n_2\leqslant N$) and also satisfy (24).

[5] Alternatively, it is possible to sample a random choice for $\mathcal{C}_0$ and then verify that it satisfies properties of Lemma 3.5, thereby obtaining a Las Vegas construction which is more efficient (in terms of the dependence on the constant γ_0) than a brute-force search. The construction would be even more efficient in Monte Carlo form; i.e., if one avoids verification of the candidate $\mathcal{C}_0$.

Finally, using Theorem 4.1 we can set up Perm so that $\ell = \Omega(\gamma_1 rn/\log n) = \Omega(\gamma_0 rn/\log n)$ and $\delta \leqslant 1/n^{\ell}$. We can lower the value of ℓ if necessary (since an ℓ-wise δ-dependent permutation is also an ℓ'-wise δ-dependent permutation for any $\ell' \leqslant \ell$) so as to ensure that $\ell = \Omega(\gamma_0 rn/(B\log n))$ and the assumptions (26) and (27) are satisfied (recall that $n_2/b = n_b = n/B$ and $r \leqslant 1$). Observe that our choices of the parameters imply that the quantity γ_2'' defined in (11) satisfies $\gamma_2'' = \Omega(\gamma_0^2/(B\log N)^2)$. We see that the choice of δ is small enough to satisfy the assumption (28).

By our choice of the parameters, the upper bound on the failure probability in (17) is

$$\Big(1-\gamma_2''/6\Big)^{\lfloor \ell/B\rfloor} = \exp\Big(-\Omega\Big(\frac{\gamma_0^3 rN}{B^3\log^3 N}\Big)\Big), \tag{30}$$

which can be seen by recalling the lower bound on γ_2'' and the fact that $N = n(1+\gamma_1) \in [n, 2n]$.

On the other hand, the upper bound on the failure probability in (20) can be written as

$$\exp(-\Omega(\ell/B^2)) = \exp\Big(-\Omega\Big(\frac{\gamma_0 rN}{B^3\log N}\Big)\Big), \tag{31}$$

which is dominated by the estimate in (30).

Now we can substitute the upper bound (29) on B to conclude that (30) is at most

$$\exp\Big(-\Omega\Big(\frac{\gamma_0^6 rN}{\log^6(1/\gamma_0)\log^3 N}\Big)\Big) = \exp\Big(-\Omega\Big(\frac{\gamma_0' rN}{\log^3 N}\Big)\Big),$$

where

$$\gamma_0' := (\gamma_0/\log(1/\gamma_0))^6.$$

We conclude that the error of the final coding scheme $(\mathsf{Enc}, \mathsf{Dec})$ which is upper bounded by ϵ' as defined in (23) is at most

$$\epsilon_1 + 2\exp\Big(-\Omega\Big(\frac{\gamma_0' rN}{\log^3 N}\Big)\Big).$$

4.3. *Instantiations*

We present two possible choices for the non-malleable code $\mathcal{C}_1$ based on existing constructions. The first construction, due to Dziembowski et al. [15], is a Monte Carlo result that is summarized below.

Theorem 4.5. [15, Theorem 4.2] *For every integer $n > 0$, there is an efficient coding scheme $\mathcal{C}_1$ of block length n, rate at least .18, that is non-malleable against bit-tampering adversaries achieving error $\epsilon = \exp(-\Omega(n))$. Moreover, there is an efficient randomized algorithm that, given n, outputs a description of such a code with probability at least $1-\exp(-\Omega(n))$.*

More recently, Aggarwal et al. [2] construct an *explicit* coding scheme which is non-malleable against the much more general class of split-state adversaries. However, this construction achieves inferior guarantees than the one above in terms of the rate and error. Below we rephrase this result restricted to bit-tampering adversaries.

Theorem 4.6. [2, impliedbyTheorem 5] *For every integer $k > 0$ and $\epsilon > 0$, there is an efficient and explicit*[6] *coding scheme $\mathcal{C}_1$ of message length k that is non-malleable against bit-tampering adversaries achieving error at most ϵ. Moreover, the block length n of the coding scheme satisfies*

$$n = \tilde{O}((k + \log(1/\epsilon))^7).$$

By choosing $\epsilon := \exp(-k)$, we see that we can have $\epsilon = \exp(-\tilde{\Omega}(n^{1/7}))$, while the rate r of the code satisfies

$$r = \tilde{\Omega}(n^{-6/7}).$$

By instantiating Theorem 4.3 with the Monte Carlo construction of Theorem 4.5, we arrive at the following corollary.

Corollary 4.7. *For every integer $n > 0$ and every positive parameter $\gamma_0 = \Omega(1/(\log n)^{O(1)})$, there is an efficient coding scheme* (Enc, Dec) *of block length n and rate at least $1 - \gamma_0$ such that the following hold.*

1. *The coding scheme is non-malleable against bit-tampering adversaries, achieving error at most $\exp(-\tilde{\Omega}(n))$,*
2. *There is an efficient randomized algorithm that, given n, outputs a description of such a code with probability at least $1 - \exp(-\Omega(n))$.* □

If, instead, we instantiate Theorem 4.3 with the construction of Theorem 4.6, we obtain the following non-malleable code.

Corollary 4.8. *For every integer $n > 0$ and every positive parameter $\gamma_0 = \Omega(1/(\log n)^{O(1)})$, there is an explicit and efficient coding scheme* (Enc, Dec) *of block length n and rate at least $1 - \gamma_0$ such that the coding scheme is non-malleable against bit-tampering adversaries and achieves error at most $\exp(-\tilde{\Omega}(n^{1/7}))$.* □

5. Construction of Non-malleable Codes Using Non-malleable Extractors

For decades, randomness extractors (cf. [24, Chapter 6]) have served as a fundamental building block in building combinatorial objects with various pseudorandom properties, including error-correcting codes. Intuitively, a randomness extractor is a function that takes a weak source of randomness and a short truly random seed and outputs a

[6] To be precise, explicitness is guaranteed assuming that a large prime $p = \exp(\tilde{\Omega}(k + \log(1/\epsilon)))$ is available.

sequence of nearly independent unbiased coin flips of length close to the entropy of the weak source. Recently a non-malleable variation of classical randomness extractors has found applications in non-malleable cryptography, namely for privacy amplification in presence of tampering adversaries [13]. The output of a non-malleable extractor remains close to uniform to an adversary even given the knowledge of the randomness seed and the extractor's output on any different seed. Intuitively this means that the truth tables of the extractor function restricted to different seeds are uncorrelated even if the input is sampled from a weak random source.

In this section, we introduce the notion of seedless non-malleable extractors that extends the existing definition of seeded non-malleable extractors (as defined in [13]) to sources that exhibit structures of interest. This is similar to how classical seedless extractors are defined as an extension of seeded extractors to sources with different kinds of structure.[7]

Furthermore, we obtain a reduction from the non-malleable variation of two-source extractors to non-malleable codes for the split-state model. Dziembowski et al. [14] obtain a construction of non-malleable codes encoding one-bit messages based on a variation of strong (standard) two-source extractors. This brings up the question of whether there is a natural variation of two-source extractors that directly leads to non-malleable codes for the split-state model encoding messages of arbitrary lengths (and ideally, achieving constant rate). Our notion of non-malleable two-source extractors can be regarded as a positive answer to this question.

At an intuitive level, the reduction takes a two-source extractor with a certain *non-malleability property* as the decoder of a coding scheme, so the encoder would take a uniformly random pre-image of the decoder function for the given message. The non-malleability property, as we define later in this section, implies that the knowledge of the extractor's output, when one or both of the extractor's inputs are tampered by an adversary, reveals no information about the extractor's output on the original (untampered) inputs. That is, the extractor's output remains nearly uniform even conditioned on the adversary's knowledge. In terms of the coding scheme, this property implies that the knowledge of decoder's output on any tampered encoding reveals essentially no information about the original message, and this is exactly the requirement of a non-malleable coding scheme.

Our reduction does not imply a characterization of non-malleable codes using extractors, and non-malleable codes for the split-state model do not necessarily correspond to non-malleable extractors (since those implied by our reduction achieve slightly sub-optimal rates). However, since seeded non-malleable extractors (as studied in the line of research starting [13]) are already subject of independent interest, we believe our characterization may be seen as a natural approach (albeit not the only possible approach) for improved constructions of non-malleable codes. Furthermore, the definition of two-source non-malleable extractors (especially the criteria described in Remark 5.5 below) is somewhat cleaner and easier to work with than the definition of non-malleable codes (Definition 2.3) that involves subtleties such as the extra care for the "same" symbol.

It should also be noted that our reduction can be modified to obtain non-malleable codes for different classes of adversaries (by appropriately defining the family of extrac-

[7] For a background on standard seeded and seedless extractors, see [7, Chapter 2].

tors based on the tampering family being considered) such as the variation of split-state model where the adversary may arbitrarily choose in advance how to partition the encoding into two blocks (in which case one has to consider the non-malleable variation of mixed two-source extractors studied by Raz and Yehudayoff [23]).

5.1. *Seedless Non-malleable Extractors*

Before defining seedless non-malleable extractors, it is convenient to introduce a related notion of *non-malleable functions* that is defined with respect to a function and a distribution over its inputs. As it turns out, non-malleable "extractor" functions with respect to the uniform distribution and limited families of adversaries are of particular interest for construction of non-malleable codes.

Definition 5.1. A function $g\colon \Sigma \to \Gamma$ is said to be non-malleable with error ϵ with respect to a distribution $\mathcal{X}$ over Σ and a tampering function $f\colon \Sigma \to \Sigma$ if there is a distribution $\mathcal{D}$ over $\Gamma \cup \{\underline{\mathsf{same}}\}$ such that for an independent $Y \sim \mathcal{D}$,

$$\mathscr{D}(g(X), g(f(X))) \approx_\epsilon \mathscr{D}(g(X), \mathsf{copy}(Y, g(X))).$$

Using the above notation, we can now define seedless non-malleable extractors as follows.

Definition 5.2. A function $\mathsf{NMExt}\colon \{0,1\}^n \to \{0,1\}^m$ is a (seedless) non-malleable extractor with respect to a class $\mathfrak{X}$ of sources over $\{0,1\}^n$ and a class $\mathcal{F}$ of tampering functions acting on $\{0,1\}^n$ if, for every distribution $\mathcal{X} \in \mathfrak{X}$, and for every tampering function $f \in \mathcal{F}$, $f\colon \{0,1\}^n \to \{0,1\}^n$, the following hold for an error parameter $\epsilon > 0$.

1. NMExt is an extractor for the distribution $\mathcal{X}$; that is, $\mathsf{NMExt}(\mathcal{X}) \approx_\epsilon \mathcal{U}_m$.
2. NMExt is a non-malleable function with error ϵ for the distribution $\mathcal{X}$ and with respect to the tampering function f.

Of particular interest is the notion of *two-source* seedless extractors. This is a special case of Definition 5.2 where $\mathfrak{X}$ is the family of two sources (i.e., each $\mathcal{X}$ is a product distribution $(\mathcal{X}_1, \mathcal{X}_2)$, where $\mathcal{X}_1$ and $\mathcal{X}_2$ are arbitrary distributions defined over the first and second half of the input, each having a sufficient amount of entropy. Moreover, the family of tampering functions consists of functions that arbitrarily but independently tamper each half of the input. Formally, we distinguish this special case of Definition 5.2 as follows.

Definition 5.3. A function $\mathsf{NMExt}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ is a two-source non-malleable (k_1, k_2, ϵ)-extractor if, for every product distribution $(\mathcal{X}, \mathcal{Y})$ over $\{0,1\}^n \times \{0,1\}^n$ where $\mathcal{X}$ and $\mathcal{Y}$ have min-entropy at least k_1 and k_2, respectively, and for any arbitrary functions $f_1\colon \{0,1\}^n \to \{0,1\}^n$ and $f_2\colon \{0,1\}^n \to \{0,1\}^n$, the following hold.

1. NMExt is a two-source extractor for $(\mathcal{X}, \mathcal{Y})$; that is, $\mathsf{NMExt}(\mathcal{X}, \mathcal{Y}) \approx_\epsilon \mathcal{U}_m$.
2. NMExt is a non-malleable function with error ϵ for the distribution $(\mathcal{X}, \mathcal{Y})$ and with respect to the tampering function $(X, Y) \mapsto (f_1(X), f_2(Y))$.

In general, a tampering function may have fixed points and act as the identity function on a particular set of inputs. Definitions of non-malleable codes, functions, and extractors all handle the technicalities involved with such fixed points by introducing a special symbol $\underline{\mathsf{same}}$. Nevertheless, it is more convenient to deal with adversaries that are promised to have no fixed points. For this restricted model, the definition of two-source non-malleable extractors can be modified as follows. We call extractors satisfying the less stringent requirement *relaxed* two-source non-malleable extractors. Formally, the relaxed definition is as follows.

Definition 5.4. A function $\mathsf{NMExt}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ is a relaxed two-source non-malleable (k_1, k_2, ϵ)-extractor if, for every product distribution $(\mathcal{X}, \mathcal{Y})$ over $\{0,1\}^n \times \{0,1\}^n$ where $\mathcal{X}$ and $\mathcal{Y}$ have min-entropy at least k_1 and k_2, respectively, the following holds. Let $f_1\colon \{0,1\}^n \times \{0,1\}^n$ and $f_2\colon \{0,1\}^n \times \{0,1\}^n$ be functions such that for every $x \in \{0,1\}^n$, $f_1(x) \neq x$ and $f_2(x) \neq x$. Then, for $(X, Y) \sim (\mathcal{X}, \mathcal{Y})$,

1. NMExt is a two-source extractor for $(\mathcal{X}, \mathcal{Y})$; that is, $\mathsf{NMExt}(\mathcal{X}, \mathcal{Y}) \approx_\epsilon \mathcal{U}_m$.
2. NMExt is a non-malleable function with error ϵ for the distribution of (X, Y) and with respect to all of the tampering functions

$$(X, Y) \mapsto (f_1(X), Y), \qquad (X, Y) \mapsto (X, f_2(Y)), \qquad (X, Y) \mapsto (f_1(X), f_2(Y)).$$

Remark 5.5. In order to satisfy the requirements of Definition 5.4, it suffices (but not necessary[8]) to satisfy the following conditions which closely resemble the requirements of seeded non-malleable extractors (as defined in [13]) and may be more convenient to work with:

$$\begin{aligned}
(\mathsf{NMExt}(\mathcal{X}, \mathcal{Y}), \mathsf{NMExt}(f_1(\mathcal{X}), \mathcal{Y})) &\approx_\epsilon (\mathcal{U}_m, \mathsf{NMExt}(f_1(\mathcal{X}), \mathcal{Y})),\\
(\mathsf{NMExt}(\mathcal{X}, \mathcal{Y}), \mathsf{NMExt}(\mathcal{X}, f_2(\mathcal{Y}))) &\approx_\epsilon (\mathcal{U}_m, \mathsf{NMExt}(\mathcal{X}, f_2(\mathcal{Y}))),\\
(\mathsf{NMExt}(\mathcal{X}, \mathcal{Y}), \mathsf{NMExt}(f_1(\mathcal{X}), f_2(\mathcal{Y}))) &\approx_\epsilon (\mathcal{U}_m, \mathsf{NMExt}(f_1(\mathcal{X}), f_2(\mathcal{Y}))).
\end{aligned}$$

The Proof of Theorem 5.10 shows that these stronger requirements can be satisfied with high probability by random functions.

It immediately follows from the definitions that a two-source non-malleable extractor (according to Definition 5.3) is a relaxed non-malleable two-source extractor (according to Definition 5.4) and with the same parameters. However, non-malleable extractors are in general meaningful for arbitrary tampering functions that may potentially have fixed

[8] To see that the listed conditions do not necessarily follow from Definition 5.4 for every pair of adversaries (f_1, f_2), suppose $\mathcal{X}$ and $\mathcal{Y}$ are fully uniform and consider the function (with a single-bit output) $\mathsf{NMExt}(X, Y) = \langle X + Y, \mathbf{1}^n \rangle$, where the addition is bit-wise XOR, the inner product is over the binary field, and $\mathbf{1}^n$ is the all ones vector of length n. Trivially, $\mathsf{NMExt}(X, Y)$ is uniform in this case. Now consider tampering functions $f_1(X)$ and $f_2(Y)$ that respectively flip the first two bits of X and Y. Note that $\mathsf{NMExt}(f_1(X), Y) = \mathsf{NMExt}(X, f_2(Y)) = \mathsf{NMExt}(f_1(X), f_2(Y)) = \mathsf{NMExt}(X, Y)$. Therefore, with respect to the chosen adversaries, the function NMExt can be seen to be non-malleable according to Definition 5.1 by taking a one-point distribution $\mathcal{D}$ that is fully supported on $\{\underline{\mathsf{same}}\}$. However, in this case none of the requirements listed in Remark 5.5 is satisfied.

points. Interestingly, below we show that the two notions are equivalent up to a slight loss in the parameters.

Lemma 5.6. *Let* NMExt *be a relaxed two-source non-malleable* $(k_1 - \log(1/\epsilon), k_2 - \log(1/\epsilon), \epsilon)$*-extractor. Then,* NMExt *is a two-source non-malleable* $(k_1, k_2, 4\epsilon)$*-extractor.*

Proof. Since the two-source extraction requirement of Definition 5.4 implies the extraction requirement of Definition 5.3, it suffices to prove the non-malleability condition of Definition 5.3.

Let $f_1 \colon \{0,1\}^n \to \{0,1\}^n$ and $f_2 \colon \{0,1\}^n \to \{0,1\}^n$ be a pair of tampering functions, $(\mathcal{X}, \mathcal{Y})$ be a product (without loss of generality, component-wise flat) distribution with min-entropy at least (k_1, k_2), and $(X, Y) \sim (\mathcal{X}, \mathcal{Y})$. Define the parameters

$$\epsilon_1 := \Pr[f_1(X) = X],$$
$$\epsilon_2 := \Pr[f_2(Y) = Y].$$

Moreover, define the distributions $\mathcal{X}_0, \mathcal{X}_1$ to be the distribution of X conditioned on the events $f_1(X) = X$ and $f_1(X) \neq X$, respectively. Let $\mathcal{Y}_0, \mathcal{Y}_1$ be similar conditional distributions for the random variable Y and the events $f_2(Y) = Y$ and $f_2(Y) \neq Y$. Let X_0, X_1, Y_0, Y_1 be random variables drawn independently and in order from $\mathcal{X}_0, \mathcal{X}_1, \mathcal{Y}_0, \mathcal{Y}_1$. Observe that $(\mathcal{X}, \mathcal{Y})$ is now a convex combination of four product distributions:

$$(\mathcal{X}, \mathcal{Y}) = \alpha_{00}(\mathcal{X}_0, \mathcal{Y}_0) + \alpha_{01}(\mathcal{X}_0, \mathcal{Y}_1) + \alpha_{10}(\mathcal{X}_1, \mathcal{Y}_0) + \alpha_{11}(\mathcal{X}_1, \mathcal{Y}_1)$$

where

$$\alpha_{00} := \epsilon_1\epsilon_2,$$
$$\alpha_{01} := \epsilon_1(1 - \epsilon_2),$$
$$\alpha_{10} := (1 - \epsilon_1)\epsilon_2,$$
$$\alpha_{11} := (1 - \epsilon_1)(1 - \epsilon_2).$$

We now need to verify Definition 5.3 for the tampering function

$$(X, Y) \mapsto (f_1(X), f_2(Y)).$$

Let us consider the distribution

$$\mathcal{E}_{01} := \mathsf{NMExt}(f_1(\mathcal{X}_0), f_2(\mathcal{Y}_1)) = \mathsf{NMExt}(\mathcal{X}_0, f_2(\mathcal{Y}_1)).$$

Suppose $\alpha_{01} \geqslant \epsilon$, which implies $\epsilon_1 \geqslant \epsilon$ and $1 - \epsilon_2 \geqslant \epsilon$. Thus, $\mathcal{X}_0$ and $\mathcal{Y}_1$ have min-entropy at least $k_1 - \log(1/\epsilon)$ and $k_2 - \log(1/\epsilon)$, respectively. In particular, since $f_2(\mathcal{Y}_1)$ has no fixed points, by Definitions 5.4 and 5.1, there is an distribution $\mathcal{D}_{01}$ over $\{0,1\}^m \cup \{\underline{\mathsf{same}}\}$ (where m is the output length of NMExt) such that for an independent random variable $E_{01} \sim \mathcal{D}_{01}$,

$$\mathscr{D}(\mathsf{NMExt}(X_0, Y_1), \mathsf{NMExt}(f_1(X_0), f_2(Y_1))) \approx_\epsilon \mathscr{D}(U_m, \mathsf{copy}(E_{01}, U_m)).$$

For $\alpha_{01} < \epsilon$, the above distributions may be 1-far; however, we can still write the following for general $\alpha_{01} \in [0, 1]$:

$$\alpha_{01}\mathscr{D}(\mathsf{NMExt}(X_0, Y_1), \mathsf{NMExt}(f_1(X_0), f_2(Y_1))) \approx_\epsilon \alpha_{01}\mathscr{D}(U_m, \mathsf{copy}(E_{01}, U_m)), \tag{32}$$

where in the above notation, we interpret distributions as vectors of probabilities that can be multiplied by a scalar (i.e., α_{01}) and use half the ℓ_1 distance of vectors as the measure of proximity. Similar results hold for

$$\mathcal{E}_{10} := \mathsf{NMExt}(f_1(\mathcal{X}_1), f_2(\mathcal{Y}_0)) = \mathsf{NMExt}(f_1(\mathcal{X}_1), \mathcal{Y}_0)$$

and

$$\mathcal{E}_{11} := \mathsf{NMExt}(f_1(\mathcal{X}_1), f_2(\mathcal{Y}_1)),$$

so that for distributions $\mathcal{D}_{10}$ and $\mathcal{D}_{01}$ over $\{0, 1\}^m \cup \{\underline{\mathsf{same}}\}$ and independent random variables $E_{10} \sim \mathcal{D}_{10}$ and $E_{11} \sim \mathcal{D}_{11}$,

$$\alpha_{10}\mathscr{D}(\mathsf{NMExt}(X_1, Y_0), \mathsf{NMExt}(f_1(X_1), f_2(Y_0))) \approx_\epsilon \alpha_{10}\mathscr{D}(U_m, \mathsf{copy}(E_{10}, U_m)), \tag{33}$$

and

$$\alpha_{11}\mathscr{D}(\mathsf{NMExt}(X_1, Y_1), \mathsf{NMExt}(f_1(X_1), f_2(Y_1))) \approx_\epsilon \alpha_{11}\mathscr{D}(U_m, \mathsf{copy}(E_{11}, U_m)). \tag{34}$$

We can also write, using the fact that NMExt is an ordinary extractor,

$$\alpha_{00}\mathscr{D}(\mathsf{NMExt}(X_0, Y_0), \mathsf{NMExt}(f_1(X_0), f_2(Y_0))) \approx_\epsilon \alpha_{00}\mathscr{D}(U, U). \tag{35}$$

where $U \sim \mathcal{U}_m$.

Denote by $\mathcal{D}'_{01}$ the distribution $\mathcal{D}_{01}$ conditioned on the complement of the event $\{\underline{\mathsf{same}}\}$. Thus, $\mathcal{D}'_{01}$ is a distribution over $\{0, 1\}^m$. Similarly, define $\mathcal{D}'_{10}$ and $\mathcal{D}'_{11}$ from $\mathcal{D}_{10}$ and $\mathcal{D}_{11}$ by conditioning on the event $\{0, 1\}^m \setminus \{\underline{\mathsf{same}}\}$. Observe that

$$\mathscr{D}(U_m, \mathsf{copy}(E_{01}, U_m)) = p_{01}\mathscr{D}(U_m, U_m) + (1 - p_{01})(\mathcal{U}_m, \mathcal{D}'_{01}), \tag{36}$$

where $p_{01} = \Pr[E_{01} = \underline{\mathsf{same}}]$. Similarly, one can write

$$\mathscr{D}(U_m, \mathsf{copy}(E_{10}, U_m)) = p_{10}\mathscr{D}(U_m, U_m) + (1 - p_{10})(\mathcal{U}_m, \mathcal{D}'_{10}) \tag{37}$$

and

$$\mathscr{D}(U_m, \mathsf{copy}(E_{11}, U_m)) = p_{11}\mathscr{D}(U_m, U_m) + (1 - p_{11})(\mathcal{U}_m, \mathcal{D}'_{11}). \tag{38}$$

Now, we can add up (32), (33), (34), and (35), using the triangle inequality, and expand each right-hand side according to (36), (33), and (34) to deduce that

$$\mathscr{D}(\mathsf{NMExt}(X, Y), \mathsf{NMExt}(f_1(X), f_2(Y))) \approx_{4\epsilon} p\mathscr{D}(U_m, U_m) + (1-p)(\mathcal{U}_m, \mathcal{D}') \quad (39)$$

for some distribution $\mathcal{D}'$ which is a convex combination

$$\mathcal{D}' = \frac{1}{1-p}(\alpha_{01}(1-p_{01})\mathcal{D}'_{01} + \alpha_{10}(1-p_{10})\mathcal{D}'_{10} + \alpha_{11}(1-p_{11})\mathcal{D}'_{11})$$

and coefficient $p = \alpha_{00} + \alpha_{01}p_{01} + \alpha_{10}p_{10} + \alpha_{11}p_{11}$. Let $\mathcal{D}$ be a distribution given by

$$\mathcal{D} := (1-p)\mathcal{D}' + p\mathscr{D}(\underline{\mathsf{same}}),$$

and observe that the right-hand side of (39) is equal to $\mathscr{D}(U_m, \mathsf{copy}(E, U_m))$, where $E \sim \mathcal{D}$ is an independent random variable. Thus, we conclude that

$$\mathscr{D}(\mathsf{NMExt}(X, Y), \mathsf{NMExt}(f_1(X), f_2(Y))) \approx_{4\epsilon} \mathscr{D}(U_m, \mathsf{copy}(E, U_m)),$$

which implies the non-malleability requirement of Definition 5.3. □

5.2. *From Non-malleable Extractors to Non-malleable Codes*

In this section, we show a reduction from non-malleable extractors to non-malleable codes. For concreteness, we focus on tampering functions in the split-state model. That is, when the input is divided into two blocks of equal size, the adversary may choose arbitrary functions that independently tamper each block. It is straightforward to extend the reduction to different families of tampering functions, for example:

1. When the adversary divides the input into $b \geqslant 2$ known parts, not necessarily of the same length, and applies an independent tampering function on each block. In this case, a similar reduction from non-malleable codes to multiple-source non-malleable extractors may be obtained.
2. When the adversary behaves as in the split-state model, but the choice of the two parts is not known in advance. That is, when the code must be simultaneously non-malleable for every splitting of the input into two equal-sized parts. In this case, the needed extractor is a non-malleable variation of the *mixed-sources extractors* studied by Raz and Yehudayoff [23].

We note that Theorem 5.7 below (and similar theorems that can be obtained for the other examples above) only require non-malleable extraction from the uniform distribution. However, the reduction from arbitrary tampering functions to ones without fixed points (e.g., Lemma 5.6) reduces the entropy requirement of the source while imposing a structure on the source distribution which is related to the family of tampering functions being considered.

Theorem 5.7. *Let* $\mathsf{NMExt}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$ *be a two-source non-malleable* (n, n, ϵ)*-extractor. Define a coding scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *with message length* k *and block length* $2n$ *as follows. The decoder* Dec *is defined by* $\mathsf{Dec}(x) := \mathsf{NMExt}(x)$.

The encoder, given a message s*, outputs a uniformly random string in* $\mathsf{NMExt}^{-1}(s)$. *Then, the pair* $(\mathsf{Enc}, \mathsf{Dec})$ *is a non-malleable code with error* $\epsilon' := \epsilon(2^k + 1)$ *for the family of split-state adversaries.*

Proof. By construction, for every $s \in \{0,1\}^k$, $\mathsf{Dec}(\mathsf{Enc}(s)) = s$ with probability 1. It remains to verify non-malleability.

Take a uniformly random message $S \sim \mathcal{U}_k$, and let $Y := \mathsf{Enc}(S)$ be its encoding. First, we claim that Y is close to be uniformly distributed on $\{0,1\}^{2n}$. □

Claim 5.8. *The distribution of* $\mathsf{Enc}(S)$ *is* ϵ*-close to uniform.*

Proof. Let $Y' \sim \mathcal{U}_{2n}$, and $S' := \mathsf{Dec}(Y') = \mathsf{NMExt}(Y')$. Observe that, since NMExt is an ordinary extractor for the uniform distribution,

$$\mathscr{D}(S') \approx_\epsilon \mathscr{D}(S) = \mathcal{U}_k. \tag{40}$$

On the other hand, since $\mathsf{Enc}(s)$ samples a uniformly random element of $\mathsf{NMExt}^{-1}(s)$, it follows that $\mathscr{D}(\mathsf{Enc}(S')) = \mathscr{D}(Y') = \mathcal{U}_{2n}$. Since S and S' correspond to statistically close distributions [by (40)], this implies that

$$\mathscr{D}(\mathsf{Enc}(S)) \approx_\epsilon \mathscr{D}(\mathsf{Enc}(S')) = \mathcal{U}_{2n}.$$

□

In light of the above claim, in the sequel without loss of generality we can assume that Y is exactly uniformly distributed at the cost of an ϵ increase in the final error parameter.

Let $Y = (Y_1, Y_2)$ where $Y_1, Y_2 \in \{0,1\}^n$. The assumption that NMExt is a non-malleable extractor according to Definition 5.3 implies that it is a non-malleable function with respect to the distribution of Y and tampering function $f\colon \{0,1\}^{2n} \to \{0,1\}^{2n}$

$$f(Y) := (f_1(Y_1), f_2(Y_2)),$$

for any choice of the functions f_1 and f_2. Let $\mathcal{D}_f$ be the distribution $\mathcal{D}$ defined in Definition 5.1 that assures non-malleability of the extractor NMExt and observe that its choice only depends on the functions f_1 and f_2 and not the particular value of S. We claim that this is the right choice of $\mathcal{D}_f$ required by Definition 2.3.

Let $S'' \sim \mathcal{D}_f$ be sampled independently from $\mathcal{D}_f$. Since, by Definition 5.3, NMExt is a non-malleable function with respect to the distribution of Y, Definition 5.1 implies that

$$\mathscr{D}(\mathsf{NMExt}(Y), \mathsf{NMExt}(f(Y))) \approx_\epsilon \mathscr{D}(\mathsf{NMExt}(Y), \mathsf{copy}(S'', \mathsf{NMExt}(Y))),$$

which, after appropriate substitutions, simplifies to

$$\mathscr{D}(S, \mathsf{Dec}(f(\mathsf{Enc}(S)))) \approx_{\epsilon} \mathscr{D}(S, \mathsf{copy}(S'', S)). \tag{41}$$

Let $s \in \{0,1\}^k$ be any fixed message. We can now condition the above equation on the event $S = s$, and deduce, using Proposition 5.17, that

$$\mathscr{D}(s, \mathsf{Dec}(f(\mathsf{Enc}(s)))) \approx_{\epsilon 2^k} \mathscr{D}(s, \mathsf{copy}(S'', s)),$$

or more simply, that

$$\mathscr{D}(\mathsf{Dec}(f(\mathsf{Enc}(s)))) \approx_{\epsilon 2^k} \mathscr{D}(\mathsf{copy}(S'', s)),$$

which is the condition required to satisfy Definition 2.3. It follows that $(\mathsf{Enc}, \mathsf{Dec})$ is a non-malleable coding scheme with the required parameters. □

We can now derive the following corollary, using the tools that we have developed so far.

Corollary 5.9. *Let* $\mathsf{NMExt}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *be a relaxed two-source non-malleable* (k_1, k_2, ϵ)*-extractor, where* $m = \Omega(n)$, $n - k_1 = \Omega(n)$, $n - k_2 = \Omega(n)$, *and* $\epsilon = \exp(-\Omega(m))$. *Then, there is a* $k = \Omega(n)$ *such that the following holds. Define a coding scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *with message length* k *and block length* $2n$ *(thus rate* $\Omega(1)$*) as follows. The decoder* Dec*, given* $x \in \{0,1\}^{2n}$*, outputs the first* k *bits of* $\mathsf{NMExt}(x)$. *The encoder, given a message* x*, outputs a uniformly random string in* $\mathsf{Dec}^{-1}(x)$. *Then, the pair* $(\mathsf{Enc}, \mathsf{Dec})$ *is a non-malleable code with error* $\exp(-\Omega(n))$ *for the family of split-state adversaries.*

Proof. Take $k = \frac{1}{2}\min\{m, n - k_1, n - k_2, \log(1/\epsilon)\}$, which implies that $k = \Omega(n)$ by the assumptions on parameters. Furthermore, we let $\epsilon' := 2^{-2k} \geqslant \epsilon$.

Let $\mathsf{NMExt}'\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$ to be defined from NMExt by truncating the output to the first k bits. Observe that as in ordinary extractors, truncating the output of a non-malleable extractor does not affect any of the parameters other than the output length. In particular, NMExt' is also a relaxed two-source non-malleable (k_1, k_2, ϵ)-extractor with output length $\Omega(n)$.

In fact, our setup implies that NMExt' is a relaxed two-source non-malleable $(n - \log(1/\epsilon'), n - \log(1/\epsilon'), \epsilon')$-extractor with output length $\Omega(n)$. By Lemma 5.6, we see that NMExt' is a two-source non-malleable $(n, n, 4\epsilon')$-extractor. We can now apply Theorem 5.7 to conclude that $(\mathsf{Enc}, \mathsf{Dec})$ is a non-malleable code with error $4\epsilon'(2^k + 1) = \Omega(2^{-k}) = \exp(-\Omega(n))$ for split-state adversaries. □

5.3. *Existence Bounds on Non-malleable Extractors*

So far we have introduced different notions of seedless non-malleable extractors without focusing on their existence. In this section, we show that the same technique used by [13] applies in a much more general setting and can in fact show that non-malleable extractors exist with respect to every family of randomness sources and every family of

tampering adversaries, both of bounded size. The main technical tool needed for proving this general claim is the following theorem.

Theorem 5.10. *Let $\mathcal{X}$ be a distribution over $\{0,1\}^n$ having min-entropy at least k, and consider arbitrary functions $f\colon \{0,1\}^n \to \{0,1\}^n$ and $g\colon \{0,1\}^n \to \{0,1\}^d$. Let $\mathsf{NMExt}\colon \{0,1\}^n \to \{0,1\}^m$ be a uniformly random function. Then, for any $\epsilon > 0$, with probability at least $1 - 8\exp(2^{2m+d} - \epsilon^3 2^{k-6})$ the following hold.*

1. *The function NMExt extracts the randomness of $\mathcal{X}$ even conditioned on the knowledge of $g(X)$; i.e.,*

$$\mathscr{D}(g(X), \mathsf{NMExt}(X)) \approx_\epsilon \mathscr{D}(g(X), \mathcal{U}_m). \tag{42}$$

2. *Let $X \sim \mathcal{X}$ and $U \sim \mathcal{U}_m$. Define the following random variable over $\{0,1\}^m \cup \{\underline{\mathsf{same}}\}$:*

$$Y := \begin{cases} \underline{\mathsf{same}} & \text{if } f(X) = X \\ \mathsf{NMExt}(f(X)) & \text{if } f(X) \neq X. \end{cases} \tag{43}$$

   Then,

$$\mathscr{D}(g(X), \mathsf{NMExt}(X), \mathsf{NMExt}(f(X))) \approx_\epsilon \mathscr{D}(g(X), U, \mathsf{copy}(Y, U)). \tag{44}$$

3. NMExt *is a non-malleable function with respect to the distribution $\mathcal{X}$ and tampering function f.*

Proof. The proof borrows ideas from the existence proof of seeded non-malleable extractors in [13]. The only difference is that we observe the same argument holds in a much more general setting.

First, we observe that it suffices to prove (44), since (42) follows from (44). Also, the result on non-malleability of the function NMExt follows from (44); in particular, one can use the explicit choice (43) of the random variable Y in Definition 5.1. Thus, it suffices to prove (44).

Let $X \sim \mathcal{X}$, $S := \mathsf{supp}(X)$, and $N := 2^n$, $K := 2^k$, $M := 2^m$, $D := 2^d$. We will use the short-hands

$$\mathsf{NMExt}_{g,f}(x) := (g(x), \mathsf{NMExt}(x), \mathsf{NMExt}(f(x))).$$

and

$$\mathsf{NMExt}_{g,f}(x, y) := (g(x), y, \mathsf{NMExt}(f(x))).$$

We separate the analysis between the fixed points of f (i.e., inputs x such that $f(x) = x$) and the rest of inputs. In order to do so, let $\beta = \Pr[f(X) \neq X]$, and let us first assume that $\beta \geqslant \epsilon/2$. Let $\mathcal{X}'$ be the distribution of X conditioned on the event $f(X) \neq X$, and $X' \sim \mathcal{X}'$. The min-entropy of $\mathcal{X}'$ is

$$H_\infty(\mathcal{X}') \geqslant H_\infty(\mathcal{X}) - \log(1/\beta) \geqslant k - \log(2/\epsilon).$$

Instead of working with the tampering function f, for technical reasons it is more convenient to consider a related function f' that does not have any fixed points. Namely, let $f' \colon \{0,1\}^n \to \{0,1\}^n$ be any function such that

$$\begin{cases} f'(x) = f(x) & \text{if } f(x) \neq x, \\ f'(x) \neq f(x) & \text{if } f(x) = x. \end{cases}$$

We now consider the distribution $\mathcal{X}'$ (instead of the original $\mathcal{X}$) and the adversary f' (which behaves the same as f on $\mathcal{X}'$). By construction, $\Pr[f'(X') = X'] = 0$ (and in fact, also $\Pr[f'(X) = X] = 0$).

Consider any distinguisher $h \colon \{0,1\}^d \times \{0,1\}^{2m} \to \{0,1\}$. Let

$$P := \Pr_{X'}[h(\mathsf{NMExt}_{g,f'}(X')) = 1]$$

and

$$\bar{P} := \Pr_{X',U_m}[h(\mathsf{NMExt}_{g,f'}(X', U_m)) = 1].$$

Here, the probability is taken only over the random variable X' and with respect to the particular realization of the function NMExt. That is, P and $\bar{P}$ are random variables depending on the randomness of the random function NMExt. Our goal, in order to prove the statistical closeness in (44), is to show that for any distinguisher h as defined above, the distribution of the bit output by h is insensitive (up to a bias change of ϵ) to whether the distinguisher is given a sample from the left-hand side of (44) or the right hand side of (44). In fact, in the sequel we show that this is the case with overwhelming probability over the randomness of the choice of extractor NMExt and we will then take a union bound on all possible choices of h.

For $x \in \{0,1\}^n$, we define

$$P_x := h(\mathsf{NMExt}_{g,f'}(x)),$$

and

$$\bar{P}_x := |\{y \in \{0,1\}^m \colon h(\mathsf{NMExt}_{g,f'}(x,y)) = 1\}|/M.$$

Note that P_x and $\bar{P}_x$ are defined similarly to P and $\bar{P}$ but with respect to a fixed choice of x (thus P and $\bar{P}$ would be the expectation of P_x and $\bar{P}_x$, respectively, when x is randomly drawn from $\mathcal{X}'$). Again, P_x and $\bar{P}_x$ are random variables depending only on the randomness of the function NMExt. Since for any x, $\mathsf{NMExt}(x)$ and $\mathsf{NMExt}(f'(x))$ are uniformly distributed and independent (due to the assumption that $f'(x) \neq x$), it follows that P_x and $\bar{P}_x$ both have the same distribution as $h(g(x), \mathcal{U}_{2m})$ and thus

$$\mathbb{E}[P_x - \bar{P}_x] = 0.$$

As in [13], we represent f' as a directed graph $G = (V, E)$ with $V := \{0,1\}^n$ and $(x, y) \in E$ iff $f'(x) = y$. By construction, G has no self loops and the out-degree of each vertex is one. As shown in [13, Lemma 39 of the full version], V can be partitioned as $V = V_1 \cup V_2$ such that $|V_1| = |V_2|$ and moreover, restrictions of G to the vertices in V_1 and V_2 (respectively, denoted by G_1 and G_2) are both acyclic graphs.

For $x \in \{0,1\}^n$, define $q(x) := \Pr[X' = x]$. It is clear that

$$P = \sum_{x \in V} q(x) P_x,$$

and,

$$\bar{P} = \sum_{x \in V} q(x) \bar{P}_x,$$

and consequently,

$$P - \bar{P} = \sum_{x \in V} q(x)(P_x - \bar{P}_x) = \sum_{x \in V_1} q(x)(P_x - \bar{P}_x) + \sum_{x \in V_2} q(x)(P_x - \bar{P}_x).$$

Let $x_1, \ldots, x_{N/2}$ be the sequence of vertices of G_1 in reverse topological order. This means that for every $i \in [N/2 - 1]$,

$$f'(x_i) \notin \{x_{i+1}, \ldots, x_{N/2}\}. \tag{45}$$

In general, the random variables $(P_x - \bar{P}_x)$ are not necessarily independent for different values of x. However, (45) allows us to assert conditional independence of these variables in the following form.

$$(\forall i \in [N/2 - 1])\colon \mathbb{E}[P_{x_{i+1}} - \bar{P}_{x_{i+1}} | P_1, \ldots, P_i, \bar{P}_1, \ldots, \bar{P}_i] = 0. \tag{46}$$

Therefore, the sequence

$$\Big(\sum_{i=1}^{j} q(x_i)(P_{x_i} - \bar{P}_{x_i}) \Big)_{j \in [N/2]}$$

forms a Martingale, and by Azuma's inequality, we have the concentration bound

$$\Pr\Big[\Big| \sum_{x \in V_1} q(x)(P_x - \bar{P}_x) \Big| > \epsilon/4 \Big] \leqslant 2 \exp\Big(-\epsilon^2 / \Big(32 \sum_{x \in V_1} q^2(x) \Big) \Big).$$

The assumption on the min-entropy of X', on the other hand, implies that

$$\sum_{x \in V_1} q^2(x) \leqslant 2^{-k + \log(2/\epsilon)} \sum_{x \in V_1} q(x) \leqslant 2/(\epsilon K).$$

A similar result can be proved for V_2; and using the above bounds combined with triangle inequality we can conclude that

$$\Pr[|P - \bar{P}| > \epsilon/2] \leqslant 4\exp(-\epsilon^3 K/64) =: \eta.$$

That is, with probability at least $1 - \eta$ over the randomness of NMExt,

$$\left| \Pr_{X'}[h(\mathsf{NMExt}_{g,f'}(X')) = 1] - \Pr_{X'}[h(\mathsf{NMExt}_{g,f'}(X', U_m)) = 1] \right| \leqslant \epsilon/2.$$

Since f and f' are designed to act identically on the support of X', in the above result we can replace f' by f. Moreover, by taking a union bound on all possible choices of the distinguisher, we can ensure that with probability at least $1 - \eta 2^{M^2 D}$, the realization of NMExt is so that

$$\mathscr{D}(g(X'), \mathsf{NMExt}(X'), \mathsf{NMExt}(f(X'))) \approx_{\epsilon/2} \mathscr{D}(g(X'), U_m, \mathsf{NMExt}(f(X'))). \quad (47)$$

We conclude that, regardless of the value of β, we can write

$$\beta\mathscr{D}(g(X'), \mathsf{NMExt}(X'), \mathsf{NMExt}(f(X'))) \approx_{\epsilon/2} \beta\mathscr{D}(g(X'), U_m, \mathsf{NMExt}(f(X'))), \quad (48)$$

where in the above notation, probability distributions are seen as vectors of probabilities that can be multiplied by a scalar β, and the distance measure is half the ℓ_1 distance between vectors (note that (48) trivially holds for the case $\beta < \epsilon/2$).

Observe that (47) in particular implies that

$$\mathscr{D}(g(X'), \mathsf{NMExt}(X')) \approx_{\epsilon/2} \mathscr{D}(g(X'), U_m), \quad (49)$$

and the argument above does not use any property of the tampering functions f and f' (i.e., not having fixed points on $\mathsf{supp}(X')$) in order to prove (49) holds with high probability (note that the tampering functions f and f' do not appear in the expression (49) and that (49) simply represents the guarantee that needs to be satisfied by standard extractors). That is, in the above we have shown that (49) holds with probability at least $1 - \eta 2^{M^2 D}$ regardless of what the functions f and f' are.

Now we consider the distribution of X conditioned on the event $f(X) = X$, that we denote by $\mathcal{X}''$. Again, we first assume that $1 - \beta \geqslant \epsilon/2$, in which case we get

$$H_\infty(\mathcal{X}'') \geqslant k - \log(2/\epsilon).$$

In this case, the above argument leading to (49) (this time, with the random variable X' replaced by X'') shows that with probability at least $1 - \eta 2^{M^2 D}$ over the choice of NMExt, and for $U \sim \mathcal{U}_m$, we have

$$\mathscr{D}(g(X''), \mathsf{NMExt}(X''), \mathsf{NMExt}(f(X''))) \approx_{\epsilon/2} \mathscr{D}(g(X''), U, U).$$

For general β, we can thus write (in a similar fashion to (48))

$$(1-\beta)\mathscr{D}(g(X''), \mathsf{NMExt}(X''), \mathsf{NMExt}(f(X''))) \approx_{\epsilon/2} (1-\beta)\mathscr{D}(g(X''), U, U). \quad (50)$$

Now, we may add up (48) and (50) and use the triangle inequality to deduce that, with probability at least $1 - 2\eta 2^{M^2 D}$ over the choice of NMExt,

$$\begin{aligned}\mathscr{D}(g(X), \mathsf{NMExt}(X), \mathsf{NMExt}(f(X))) &\approx_{\epsilon} \beta\mathscr{D}(g(X'), U, \mathsf{NMExt}(f(X'))) \\ &+ (1-\beta)\mathscr{D}(g(X''), U, U). \end{aligned} \quad (51)$$

The result (44) now follows after observing that the convex combination on the right-hand side of (51) is the same as $\mathscr{D}(g(X), U, \mathsf{copy}(Y, U))$. □

As mentioned before, the above theorem is powerful enough to show existence of any desired form of non-malleable extractors, as long as the class of sources and the family of tampering functions (which are even allowed to have fixed points) are of bounded size. In particular, it is possible to use the theorem to recover the result in [13] on the existence of strong seeded non-malleable extractors by considering both the seed and input of the extractor as an n-bit string, and letting "the side information function" $g(X)$ be one that simply outputs the seed part of the input. The family of tampering functions, on the other hand, would be all functions that act on the portion of the n-bit string corresponding to the extractor's seed.

For our particular application, we apply Theorem 5.10 to show existence of two-source non-malleable extractors. In fact, it is possible to prove existence of *strong* two-source extractors in the sense that we may allow any of the two sources revealed to the distinguisher, and still guarantee extraction and non-malleability properties. However, such strong extractors are not needed for our particular application.

Theorem 5.11. *Let* $\mathsf{NMExt}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *be a uniformly random function. For any* $\gamma, \epsilon > 0$ *and parameters* $k_1, k_2 \leqslant n$*, with probability at least* $1-\gamma$ *the function* NMExt *is a two-source non-malleable* (k_1, k_2, ϵ)*-extractor provided that*

$$\begin{aligned} 2m &\leqslant k_1 + k_2 - 3\log(1/\epsilon) - \log\log(1/\gamma), \\ \min\{k_1, k_2\} &\geqslant \log n + \log\log(1/\gamma) + O(1). \end{aligned}$$

Proof. First we note that, similar to ordinary extractors, Definition 5.3 remains unaffected if one only considers random sources where each component is a flat distribution.

Let $K_1 := 2^{k_1}$, $K_2 := 2^{k_2}$, $N := 2^n$, $M := 2^m$. Without loss of generality, assume that K_1 and K_2 are integers. Let $\mathfrak{X}$ be the class of distributions $\mathcal{X} = (\mathcal{X}_1, \mathcal{X}_2)$ over $\{0,1\}^n \times \{0,1\}^n$ such that $\mathcal{X}_1$ and $\mathcal{X}_2$ are flat sources with min-entropy at least k_1 and k_2, respectively. Note that the min-entropy of $\mathcal{X}$ is at least $k_1 + k_2$. Without loss of generality, we assume that $k_1 \leqslant k_2$. The number of such sources can be bounded as

$$|\mathfrak{X}| \leqslant \binom{N}{K_1}\binom{N}{K_2} \leqslant N^{K_1+K_2} \leqslant N^{2K_2}.$$

The family $\mathcal{F}$ of tampering functions can be written as $\mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2$, where $\mathcal{F}_1$ and $\mathcal{F}_2$ contain functions that act on the first and second n bits, respectively. For the family $\mathcal{F}_1$, it suffices to only consider functions that act arbitrarily on some set of K_1 points in $\{0,1\}^n$, but are equal to the identity function on the remaining inputs. This is because a tampering function $f_1 \in \mathcal{F}_1$ will be applied to some distribution $\mathcal{X}_1$ which is only supported on a particular set of K_1 points in $\{0,1\}^n$, and thus the extractor's behavior on $\mathcal{X}_1$ is not affected by how f_1 is defined outside the support of $\mathcal{X}_1$. From this observation, we can bound the size of $\mathcal{F}$ as

$$|\mathcal{F}| \leqslant \binom{N}{K_1} N^{K_1} \cdot \binom{N}{K_2} N^{K_2} \leqslant N^{2(K_1+K_2)} \leqslant N^{4K_2}.$$

Now, we can apply Theorem 5.10 on the input domain $\{0,1\}^n \times \{0,1\}^n$. The choice of the function g is not important for our result, since we do not require two-source extractors that are strong with respect to either of the two sources. We can thus set $g(x) = 0$ for all $x \in \{0,1\}^{2n}$. By taking a union bound on all choices of $\mathcal{X} \in \mathfrak{X}$ and $(f_1, f_2) \in \mathcal{F}$, we deduce that the probability that NMExt fails to satisfy Definition 5.3 for some choice of the two sources in $\mathfrak{X}$ and tampering function in $\mathcal{F}$ is at most

$$8\exp(2M^2 - \epsilon^3 K_1 K_2/16)|\mathfrak{X}| \cdot |\mathcal{F}| \leqslant 8N^{4K_2}\exp(2M^2 - \epsilon^3 K_1 K_2/64).$$

This probability can be made less than γ provided that

$$\begin{aligned} 2m &\leqslant k_1 + k_2 - 3\log(1/\epsilon) - \log\log(1/\gamma), \\ k_1 &\geqslant \log n + \log\log(1/\gamma) + O(1), \end{aligned}$$

as desired. □

We are finally ready to prove that there are non-malleable two-source extractors defining coding schemes secure in the split-state model and achieving constant rates; in particular, arbitrarily close to 1/5.

Corollary 5.12. *For every $\alpha > 0$, there is a choice of* NMExt *in Theorem 5.7 that makes* (Enc, Dec) *a non-malleable coding scheme against split-state adversaries achieving rate $1/5 - \alpha$ and error $\exp(-\Omega(\alpha n))$.*

Proof. First, for some α', we use Theorem 5.11 to show that if $\mathsf{NMExt}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$ is randomly chosen, with probability at least .99 it is a two-source non-malleable $(n, n, 2^{-k(1+\alpha')})$-extractor, provided that

$$k \leqslant n - (3/2)\log(1/\epsilon) - O(1) = n - (3/2)k(1+\alpha') - O(1),$$

which can be satisfied for some $k \geqslant (2/5)n - \Omega(\alpha' n)$. Now, we can choose $\alpha' = \Omega(\alpha)$ so as to ensure that $k \geqslant 2n(1-\alpha)$ (thus, keeping the rate above $1-\alpha$) while having $\epsilon \leqslant 2^{-k}\exp(-\Omega(\alpha n))$. We can now apply Theorem 5.7 to attain the desired result. □

Acknowledgements

The authors would like to thank anonymous referees for their careful reading of an earlier draft of this work and their numerous helpful comments.

Appendix 1: Construction of LECSS Codes

In this section, we recall a well-known construction of LECSS codes based on linear error-correcting codes [11,15]. Construction 2 defines the reduction.

The main tool that we use is the following lemma, which appears (in a slightly different form) in [15] (which in turn is based on [11]). We include a proof for completeness.

Lemma 5.13. *The pair* $(\mathsf{Enc}, \mathsf{Dec})$ *of Construction 2 is a* $(\delta N/\log q, \tau N/\log q)$*-linear error-correcting coding scheme.*

Proof. First, observe that the linearity condition of Definition 2.8 follows from the fact that Enc is an injective linear function of $(s_1, \ldots, s_k)$ as defined in Construction 2. Furthermore, the distance property of the coding scheme follows from the fact that Enc encodes an error-correcting of distance at least $\delta n = \delta N/(\log q)$.

In order to see the bounded independence property of Definition 2.8, consider a fixed message $s \in \{0,1\}^K$, which in turn fixes the vector $(s_{k_0+1}, \ldots, s_k)$ in Construction 2. Let G_0 denote the sub-matrix of G defined by the first k_0 rows. Consider the vector $S' \in \mathbb{F}_q^n$ given by

$$S' := (s_1, \ldots, s_k) \cdot G = (s_1, \ldots, s_{k_0}) \cdot G_0 + a,$$

- *Given:* A $k \times n$ matrix G over $\mathbb{F}_q$, where q is a power of two and $n \geqslant k$ such that
  1. Rows of G span a code with relative distance at least $\delta > 0$,
  2. For some $k_0 \in [k]$, the first k_0 rows of G span a code with dual relative distance at least $\tau > 0$.
- *Output:* A coding scheme $(\mathsf{Enc}, \mathsf{Dec})$ of block length $N := n \log q$ and message length $K := (k - k_0) \log q$.
- *Construction of the encoder* $\mathsf{Enc}(s)$*, given a message* $s \in \{0,1\}^K$*:*
  1. Pick a uniformly random vector $(s_1, \ldots, s_{k_0}) \in \mathbb{F}_q^{k_0}$.
  2. Interpret s as a vector over $\mathbb{F}_q$; namely, $(s_{k_0+1}, \ldots, s_k) \in \mathbb{F}_q$.
  3. Output $(s_1, \ldots, s_k) \cdot G \in \mathbb{F}_q^n$ in binary form (i.e., as a vector in $\{0,1\}^N$).
- *Construction of the decoder* $\mathsf{Dec}(w)$*, given an input* $w \in \{0,1\}^N$*:*
  1. Interpret w as a vector $(w_1, \ldots, w_n) \in \mathbb{F}_q^n$.
  2. If there is a vector $(s_1, \ldots, s_k) \in \mathbb{F}_q^k$ such that $(s_1, \ldots, s_k) \cdot G = (w_1, \ldots, w_n)$, output $(s_{k_0+1}, \ldots, s_k) \in \mathbb{F}_q^{k-k_0}$ in binary form (i.e., as a vector in $\{0,1\}^K$). Otherwise, output $\perp$.

Construction 2: Explicit construction of LECSS codes from linear codes.

where $a \in \mathbb{F}_q^n$ is an affine shift uniquely determined by s. Recall that the assumption on the dual distance of the code spanned by the rows of G_0 implies that every τn columns of G_0 are linearly independent. Since $(s_1, \ldots, s_{k_0})$ is a uniformly random vector, this implies that the restriction of S' to any set of $\tau n = \tau N/(\log q)$ coordinates is uniformly random (as a vector in $\mathbb{F}_q^{\tau n}$). Since $\mathsf{Enc}(s)$ is the bit-representation of S', it follows that the random vector $\mathsf{Enc}(s)$ is $(\tau N/(\log q))$-wise independent. □

Instantiation Using Reed–Solomon codes

A simple way to instantiate Construction 2 is using Reed-Solomon codes. For a target rate parameter $r := 1 - \alpha$, we set up the parameters as follows. For simplicity, assume that n is a power of two.

1. The field size is $q := n$. Therefore, $N = n \log n$.
2. Set $k := \lceil n(1 - \alpha/2) \rceil$ and $k_0 := \lfloor \alpha n/2 \rfloor$. Therefore, $K := (k - k_0) \log q \geqslant n(1 - \alpha) \log n$, which ensures that the rate of the coding scheme is at least $1 - \alpha$.
3. Since G generates a Reed–Solomon code, which is an MDS code, we have $\delta = 1 - k/n \geqslant \alpha/2 - 1/n = \Omega(\alpha)$.
4. We note that the matrix G is a $k \times n$ Vandermonde matrix whose first k_0 rows also form a Vandermonde matrix spanning a Reed–Solomon code. The dual distance of the code formed by the span of the first k_0 rows of G is thus equal to $\tau = k_0/n \geqslant \alpha/2 - 1/n = \Omega(\alpha)$.

In particular, Lemma 5.13 applied to the above set up of the parameters implies that the resulting coding scheme is an $(\Omega(\alpha N/\log n), \Omega(\alpha N/\log n))$-linear error-correcting secret sharing code.

When n is not a power of two, it is still possible to pick the least $q \geqslant n$ which is a power of two and obtain similar results. In general, we have the following corollary of Lemma 5.13.

Corollary 5.14. *For every integer $n \geqslant 1$ and $\alpha \in (0, 1)$, there is an explicit construction of a binary coding scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *of block length n and message length $k \geqslant n(1 - \alpha)$ which is an $(\Omega(\alpha n/\log n), \Omega(\alpha n/\log n))$-linear error-correcting secret sharing code.* □

Appendix 2: Useful Tools

In some occasions in the paper, we deal with a chain of correlated random variables $0 = X_0, X_1, \ldots, X_n$ where we wish to understand an event depending on X_i conditioned on the knowledge of the previous variables. That is, we wish to understand

$$\mathbb{E}[f(X_i)|X_0, \ldots, X_{i-1}].$$

The following proposition shows that in order to understand the above quantity, it suffices to have an estimate with respect to a more restricted event than the knowledge of

$X_0, \ldots, X_{i-1}$. Formally, we can state the following, where X stands for X_i in the above example and Y stands for $(X_0, \ldots, X_{i-1})$.

Proposition 5.15. *Let X and Y be possibly correlated random variables and let Z be a random variable such that the knowledge of Z determines Y; that is, $Y = f(Z)$ for some function f. Suppose that for every possible outcome of the random variable Z, namely, for every $z \in \mathsf{supp}(Z)$, and for some real-valued function g, we have*

$$\mathbb{E}[g(X)|Z = z] \in I. \tag{52}$$

for a particular interval I. Then, for every $y \in \mathsf{supp}(Y)$,

$$\mathbb{E}[g(X)|Y = y] \in I.$$

Proof. Let $T = \{z \in \mathsf{supp}(Z) \colon f(z) = y\}$, and let $p(z) := \Pr[Z = z|Y = y]$. Then,

$$\mathbb{E}[g(X)|Y = y] = \sum_{z \in T} p(z)\mathbb{E}[g(X)|Z = z].$$

Since by (52), each $\mathbb{E}[g(X)|Z = z]$ lies in I and $\sum_{z \in T} p(z) = 1$, we deduce that

$$\mathbb{E}[g(X)|Y = y] \in I.$$

□

Proposition 5.16. *Let the random variable $X \in \{0, 1\}^n$ be uniform on a set of size at least $(1 - \epsilon)2^n$. Then, $\mathscr{D}(X)$ is $(\epsilon/(1 - \epsilon))$-close to $\mathcal{U}_n$.*

Proposition 5.17. *Let $\mathcal{D}$ and $\mathcal{D}'$ be distributions over the same finite space Ω, and suppose they are ϵ-close to each other. Let $E \subseteq \Omega$ be any event such that $\mathcal{D}(E) = p$. Then, the conditional distributions $\mathcal{D}|E$ and $\mathcal{D}'|E$ are (ϵ/p)-close.*

Lemma 5.18. *Let* $(\mathsf{Enc}, \mathsf{Dec})$ *be a coding scheme of message length k which is non-malleable with respect to a family $\mathcal{F}$ of adversaries with error ϵ. Let $S \in \{0, 1\}^k$ be a message drawn randomly according to any distribution and $S' := \mathsf{Dec}(f(\mathsf{Enc}(S)))$ for some $f \in \mathcal{F}$. Then, there is an independent random variable $S'' \in \{0, 1\}^k$ and parameter $\alpha \in [0, 1]$ only depending on the code and f such that*

$$\mathscr{D}(S, S') \approx_\epsilon \alpha \cdot \mathscr{D}(S, S) + (1 - \alpha) \cdot \mathscr{D}(S, S'').$$

Proof. Let $\mathcal{D}_f$ be the distribution from Definition 2.3 over $\{0, 1\}^k \cup \{\underline{\mathsf{same}}\}$ and let $\alpha = \mathcal{D}_f(\underline{\mathsf{same}})$ be the probability assigned to $\underline{\mathsf{same}}$ by $\mathcal{D}_f$. Let $S_0 \sim \mathcal{D}_f$ be an independent random variable and S'' be an independent random variable drawn from the distribution of S_0 conditioned on the event $S_0 \neq \underline{\mathsf{same}}$. By Definition 2.3, we have

$$\mathscr{D}(S, S') \approx_\epsilon \mathscr{D}(S, \mathsf{copy}(S_0, S)),$$

which can be seen by applying the definition for every fixing of S and taking a convex combination. In turn, we have

$$\mathscr{D}(S, \mathsf{copy}(S_0, S)) = \alpha \mathscr{D}(S, S) + (1 - \alpha)\mathscr{D}(S, S'').$$

which completes the proof. □

Proposition 5.19. *Let $\mathcal{D}$ be the distribution of n independent bits, where each bit is ϵ-close to uniform. Then, $\mathcal{D}$ is $O(n\epsilon)$-close to $\mathcal{U}_n$.*

Proof. Let $x \in \{0, 1\}^n$ be any fixed string. Then

$$\mathcal{D}(x) \leqslant (1/2 + \epsilon)^n = 2^{-n}(1 + 2\epsilon)^n \leqslant 2^{-n}(1 + O(\epsilon n)).$$

Similarly, one can show that $\mathcal{D}(x) \geqslant 2^{-n}(1 - O(\epsilon n))$. Now, the claim follows from the definition of statistical distance and using the above bounds for each x. □

We will use the following tail bound on summation of possibly dependent random variables, which is a direct consequence of Azuma's inequality.

Proposition 5.20. *Let $0 = X_0, X_1, \ldots, X_n$ be possibly correlated indicator random variables such that for every $i \in [n]$ and for some $\gamma \geqslant 0$,*

$$\mathbb{E}[X_i | X_0, \ldots, X_{i-1}] \leqslant \gamma.$$

Then, for every $c \geqslant 1$,

$$\Pr\left[\sum_{i=1}^{n} X_i \geqslant cn\gamma\right] \leqslant \exp(-n\gamma^2(c-1)^2/2),$$

or equivalently, for every $\delta > \gamma$,

$$\Pr\left[\sum_{i=1}^{n} X_i \geqslant n\delta\right] \leqslant \exp(-n(\delta - \gamma)^2/2).$$

Proof. See [8] for a proof. □

In a similar fashion (using Azuma's inequality for sub-martingales rather than super-martingales in the proof), we may obtain a tail bound when we have a lower bound on conditional expectations.

Proposition 5.21. *Let $0 = X_0, X_1, \ldots, X_n$ be possibly correlated random variables in $[0, 1]$ such that for every $i \in [n]$ and for some $\gamma \geqslant 0$,*

$$\mathbb{E}[X_i | X_0, \ldots, X_{i-1}] \geqslant \gamma.$$

Then, for every $\delta < \gamma$,

$$\Pr\left[\sum_{i=1}^{n} X_i \leqslant n\delta\right] \leqslant \exp(-n(\delta-\gamma)^2/2).$$

The lemma below shows that it is possible to sharply approximate a distribution $\mathcal{D}$ with finite support by sampling possibly correlated random variables $X_1, \ldots, X_n$ where the distribution of each X_i is close to $\mathcal{D}$ conditioned on the previous outcomes, and computing the empirical distribution of the drawn samples.

Lemma 5.22. [8] *Let* $\mathcal{D}$ *be a distribution over a finite set* Σ *such that* $|\mathsf{supp}(\mathcal{D})| \leqslant r$. *For any* $\eta, \epsilon, \gamma > 0$ *such that* $\gamma < \epsilon$, *there is a choice of*

$$n_0 = O((r + 2 + \log(1/\eta))/(\epsilon-\gamma)^2)$$

such that for every $n \geqslant n_0$ *the following holds. Suppose* $0 = X_0, X_1, \ldots, X_n \in \Sigma$ *are possibly correlated random variables such that for all* $i \in [n]$ *and all values* $0 = x_0, x_1 \ldots, x_n \in \mathsf{supp}(\mathcal{D})$,

$$\mathscr{D}(X_i | X_0 = x_0, \ldots, X_{i-1} = x_{i-1}) \approx_\gamma \mathcal{D}.$$

Then, with probability at least $1 - \eta$, *the empirical distribution of the outcomes* $X_1, \ldots, X_n$ *is* ϵ*-close to* $\mathcal{D}$.

References

[1] D. Aggarwal, Y. Dodis, T. Kazana, M. Obremski, Non-malleable reductions and applications, in *Cryptology* ePrint Archive, Report 2014/821 (2014). http://eprint.iacr.org/

[2] D. Aggarwal, Y. Dodis, S. Lovett, Non-malleable codes from additive combinatorics, in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing* (2014), pp.774–783

[3] B. Barak, A. Rao, R. Shaltiel, A. Wigderson, 2-Source dispersers for sub-polynomial entropy and Ramsey graphs beating the Frankl–Wilson construction. *Ann. Math.* **176**(3), 1483–1544 (2012)

[4] J. Bourgain, More on the Sum–Product phenomenon in prime fields and its applications. *Int. J. Number Theory* **1**(1), 1–32 (2005)

[5] E. Chattopadhyay, V. Goyal, X. Li, Non-malleable extractors and codes, with their many tampered extensions. Preprint arXiv:1505.00107 (2015)

[6] E. Chattopadhyay, D. Zuckerman, Non-malleable codes against constant split-state tampering, in *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (2014), pp. 306–315

[7] M. Cheraghchi, *Applications of Derandomization Theory in Coding*. Ph.D. Thesis, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland (2010). http://eccc.hpi-web.de/static/books/Applications_of_Derandomization_Theory_in_Coding/

[8] M. Cheraghchi, V. Guruswami, Capacity of non-malleable codes, in *Proceedings of Innovations in Theoretical Computer Science (ITCS 2014)* (2014)

[9] M. Cheraghchi, V. Guruswami, Non-malleable coding against bit-wise and split-state tampering, in *Proceedings of Theory of Cryptography Conference (TCC 2014)* (2014)

[10] B. Chor, O. Goldreich, Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM J. Comput.*, **2**(17), 230–261 (1988)
[11] R. Cramer, H. Chen, S. Goldwasser, R. de Haan, V. Vaikuntanathan, Secure computation from random error-correcting codes, in *Proceedings of Eurocrypt 2007* (2007), pp. 291–310
[12] R. Cramer, Y. Dodis, S. Fehr, C. Padró, D. Wichs, Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors, in *Proceedings of EUROCRYPT 2008* (2008), pp. 471–488
[13] Y. Dodis, D. Wichs, Non-malleable extractors and symmetric key cryptography from weak secrets, in *Proceedings of the 41st annual ACM Symposium on Theory of Computing* (2009), pp. 601–610. Full version published in Cryptology ePrint Archive, Report 2008/503 (eprint.iacr.org/2008/503)
[14] S. Dziembowski, T. Kazana, M. Obremski, Non-malleable codes from two-source extractors, in *Proceedings of CRYPTO* (2013), pp. 239–257
[15] S. Dziembowski, K. Pietrzak, D. Wichs, Non-malleable codes, in *Proceedings of Innovations in Computer Science (ICS 2010)* (2010)
[16] G.D. Forney, *Concatenated Codes* (MIT Press, Cambridge, 1966)
[17] V. Guruswami, A. Smith. Codes for computationally simple channels: Explicit constructions with optimal rate, in *Proceedings of FOCS 2010* (2010), pp. 723–732
[18] J. Justesen, A class of constructive asymptotically good algebraic codes. *IEEE Trans. Inf. Theory* **18**, 652–656 (1972)
[19] Y. Kalai, X. Li, A. Rao, in *2th Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (2009), pp. 617–626
[20] E. Kaplan, M. Naor, O. Reingold, Derandomized constructions of k-wise (almost) independent permutations, in *Proceedings of RANDOM 2005* (2005), pp. 113–133
[21] A. Rao, A 2-source almost-extractor for linear entropy, in *Proceedings of RANDOM 2008* (2008), pp. 549–556
[22] R. Raz, Extractors with weak random seeds, in *Proceedings of the37th Annual ACM Symposium on Theory of Computing (STOC)* (2005), pp. 11–20
[23] R. Raz, A. Yehudayoff, Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors. *J. Comput. Syst. Sci* **77**(1), 167–190 (2011)
[24] S. Vadhan, Pseudorandomness. *Found. Trends Theor. Comput. Sci.* **7**(1–3), 1–336 (2012)