Non-Malleable Coding Against Bit-wise and Split-State Tampering*

Mahdi Cheraghchi[1]** and Venkatesan Guruswami[2]***

[1] CSAIL, Massachusetts Institute of Technology
mahdi@csail.mit.edu
[2] Computer Science Department, Carnegie Mellon University
guruswami@cmu.edu

Abstract. Non-malleable coding, introduced by Dziembowski, Pietrzak and Wichs (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 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.

---

* A draft of the full version of this paper appears in [6].

** Research supported in part by the Swiss National Science Foundation research grant PA00P2-141980, V. Guruswami's Packard Fellowship, and MSR-CMU Center for Computational Thinking.

*** 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.

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: coding theory; cryptography; error detection; information theory; randomness extractors; tamper-resilient storage

1 Introduction

Non-malleable codes were introduced by Dziembowski, Pietrzak, and Wichs [12] as a relaxation of the classical notions of error-detection and error-correction. Informally, a code is non-malleable if the 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" [12]. 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. [8] 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 [12] 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 [5], 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.

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 [5].

Dziembowski et al. [12] 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. *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 Section 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 [13,16]. 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*[3]. 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. Though the above scheme uses non-malleable codes in two places to construct the final non-malleable code, there is no circularity as 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 [15]. The exact details though are quite different; in particular, the crux in the analysis of [15] was ensuring that the decoder can recover the seed correctly, and towards 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.

Split-state adversaries Bit-tampering functions act on different bits independently. A much more general class of tampering functions considered in the liter-

[3] Throughout the paper, by pseudorandom permutation we mean t-wise independent permutation (as in Definition 8) for an appropriate choice of t. This should not be confused with cryptographic pseudorandom permutations, which are not used in this work.

ature [12,11,1] 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 [5]. A natural question therefore is to construct *efficient* non-malleable codes of rate approaching 1/2 in the split-state model (the results in [12] and [5] are existential, and the codes do not admit polynomial size representation or 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 [11] 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, Dodis, and Lovett [1] solved the central open problem left in [11] — 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 paper (Section 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 [1] 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 [10]. 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 [10] 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[4] (see, e.g., [2,3,7,17,20,21]). The problem becomes increasingly challenging, however, (and remains open to date) when the entropy rate of the two sources may be noticeably 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.

Furthermore, we note that for seeded non-malleable extractors (which is a relatively recent notion) there are already a few exciting explicit constructions [9,14,19][5]. The closest construction to our application is [9] which is in fact a two-source non-malleable extractor when the adversary may tamper with either of the two sources (but not simultaneously both). Moreover, the coding scheme defined by this extractor (which is the character-sum extractor of Chor and Goldreich [7]) naturally allows for an efficient encoder and decoder. Nevertheless, it appears challenging to extend known constructions of seeded non-malleable extractors to the case when both inputs can be tampered. We leave explicit constructions of non-malleable two-source extractors, even with sub-optimal parameters, as an interesting open problem for future work.

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

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

[5] [19] also establishes a connection between seeded non-malleable extractors and ordinary two-source extractors.

from. 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 $\mathcal{X}$. Two distributions $\mathcal{X}$ and $\mathcal{Y}$ being ϵ-close in statistical distance is denoted by $\mathcal{X} \approx_\epsilon \mathcal{Y}$. We will use $(\mathcal{X}, \mathcal{Y})$ for the product distribution with the two coordinates independently sampled from $\mathcal{X}$ and $\mathcal{Y}$. All unsubscripted logarithms are taken to the base 2. Support of a discrete random variable X is denoted by $\mathsf{supp}(X)$. A distribution is said to be *flat* if it is uniform on its support. 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 [12]. First, we recall the notion of *coding schemes*.

Definition 2 (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 3. For a finite set Γ, the function $\mathsf{copy}\colon (\Gamma \cup \{\underline{\mathsf{same}}\}) \times \Gamma \to \Gamma$ is defined as follows:

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

The notion of non-malleable coding schemes from [12] can now be rephrased as follows.

Definition 4 (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, $\mathscr{D}(S) \approx_\epsilon \mathscr{D}(\mathsf{copy}(S', s))$. □

Dziembowski et al. [12] also consider the following stronger variation of non-malleable codes, and show that strong non-malleable codes imply regular non-malleable codes as in Definition 4.

Definition 5 (Strong non-malleability). A pair of functions as in Definition 4 is said to be a *strong* non-malleable coding scheme with error ϵ with respect to a family $\mathcal{F}$ of tampering functions acting on $\{0,1\}^n$ if the following holds. For any message $s \in \{0,1\}^k$, let $E_s := \mathsf{Enc}(s)$, consider the random variable

$$D_s := \begin{cases} \underline{\mathsf{same}} & \text{if } f(E_s) = E_s, \\ \mathsf{Dec}(f(E_s)) & \text{otherwise,} \end{cases}$$

and let $\mathcal{D}_{f,s} := \mathscr{D}(D_s)$. It must be the case that for every pair of distinct messages $s_1, s_2 \in \{0,1\}^k$, $\mathcal{D}_{f,s_1} \approx_\epsilon \mathcal{D}_{f,s_2}$. □

Remark 1 (Efficiency of sampling $\mathcal{D}_f$). The original definition of non-malleable codes in [12] 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): "Let $S \sim \mathcal{U}_k$, and $X := f(\mathsf{Enc}(S))$. If $\mathsf{Dec}(X) = S$, output $\underline{\mathsf{same}}$. Otherwise, output $\mathsf{Dec}(X)$."

Definition 6 (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 7 (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 8 (Limited independence of permutations). The distribution of a random permutation $\Pi\colon [n] \to [n]$ 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 $(\Pi(t)\colon t \in T)$ is δ-close to that of $(\bar{\Pi}(t)\colon t \in T)$, where $\bar{\Pi}\colon [n] \to [n]$ 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. [12] for their construction of non-malleable coding schemes against bit-tampering adversaries.

Definition 9 (LECSS). [12] 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 7).
3. For every $w, w' \in \{0,1\}^n$ such that $\mathsf{Dec}(w) \neq \perp$ and $\mathsf{Dec}(w') \neq \perp$, 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

In this section, we recall the probabilistic construction of non-malleable codes introduced in [5]. This construction, depicted as Construction 1, is defined with respect to an integer parameter $t > 0$ and a *distance parameter* $\delta \in [0,1)$.

– *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 [5].

Non-malleability of the construction (for an appropriate choice of the parameters) against any bounded-size family of adversaries, and in particular bit-tampering adversaries, follows from [5]. We derive additional properties of the construction that are needed for the explicit construction of Section 4. In particular, we state the following result which is proved in the final version of the paper.

Lemma 10. *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) 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, $\Pr_{U_S}[\mathsf{Dec}(U_S = \perp)] \geqslant 1/2$.
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[6]) Let* $f\colon \{0,1\}^n \to \{0,1\}^n$ *be any bit-tampering adversary that is neither the identity function nor a constant function. Then, for every* $s \in \{0,1\}^k$, $\Pr[\mathsf{Dec}(f(\mathsf{Enc}(s))) = \perp] \geqslant 1/3$, *where the probability is taken over the randomness of the encoder.*

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

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 10; 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 8) 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 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 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.

[6] This property is a corollary of non-malleability, cube property and bounded independence.

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 this 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.

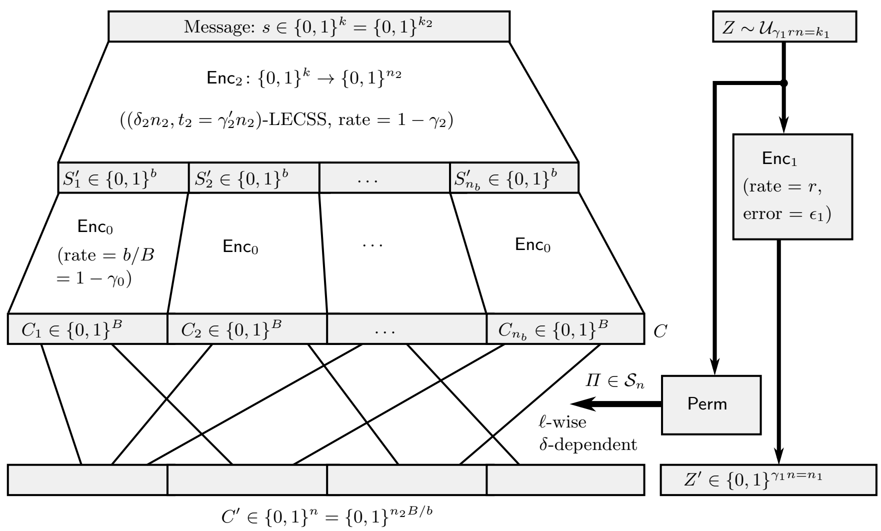

Fig. 1. Schematic description of the encoder Enc from our explicit construction.

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 10.
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 $k_1 := \gamma_1 rn$, 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). The non-malleable code $\mathcal{C}_1$ need not be strong.

3. A linear error-correcting secret sharing (LECSS) scheme $\mathcal{C}_2 = (\mathsf{Enc}_2, \mathsf{Dec}_2)$ (as in Definition 9) 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 8, for parameters ℓ and δ. In fact, it is possible to achieve $\delta \leqslant \exp(-\ell)$ and $\ell = \lceil \gamma_1 rn/\log n \rceil$ for some constant $\gamma > 0$. Namely, we may use the following result due to Kaplan, Naor and Reingold [18]:

   Theorem 11. *[18] For every integers $n, k_1 > 0$, there is an explicit 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 $\delta \leqslant \exp(-\ell)$.*

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 Figure 1.

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 $\bar{\Pi}\colon [n] \to [n]$ 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 $\bar{C}$).
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 2. As in the classical variation of concatenated codes of Forney [13] due to Justesen [16], 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 10) is small enough (e.g., $1/n^{\Omega(1)}$), our analysis still applies.

In the following theorem, we prove that the above construction is indeed a coding scheme that is non-malleable against bit-tampering adversaries with rate arbitrarily close to 1. Proof of the theorem appears in Section 4.3.

Theorem 12. *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*[7]*. The pair $(\mathsf{Enc}, \mathsf{Dec})$ defined in Section 4.1 can be set up to be a strong non-malleable coding scheme against bit-tampering adversaries, achieving block length N, rate at least $1 - \gamma_0$ and error $\epsilon \leqslant \epsilon_1 + 3\exp\big(-\Omega\big(\frac{\gamma_0' r N}{\log^3 N}\big)\big)$, where r and ϵ_1 are respectively the rate and the error of the assumed non-malleable coding scheme $\mathcal{C}_1$.* □

4.2 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. [12], is a Monte Carlo result that is summarized below.

Theorem 13. *[12, 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. [1] 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 14. *[1, implied by Theorem 5] For every integer $k > 0$ and $\epsilon > 0$, there is an efficient and explicit*[8] *coding scheme $\mathcal{C}_1$ of message length k that is non-malleable against bit-tampering adversaries achieving error at most ϵ.*

---

[7] We can extend the construction to arbitrary block lengths N by standard padding techniques and observing that the set of block lengths for which construction of Figure 1 is defined is dense enough to allow padding without affecting the rate.

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

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 12 with the Monte Carlo construction of Theorem 13, we arrive at the following corollary.

Corollary 15. *For every integer* $n > 0$ *and every positive parameter* $\gamma_0 = \Omega(1/(\log n)^{O(1)})$, *there is an efficient coding scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *of block length* n *and rate at least* $1 - \gamma_0$ *such that the coding scheme is strongly non-malleable against bit-tampering adversaries, achieving error at most* $\exp(-\tilde{\Omega}(n))$, *Moreover, there is an efficient randomized algorithm that, given* n, *outputs a description of such a code with probability at least* $1 - \exp(-\Omega(n))$.

If, instead, we instantiate Theorem 12 with the construction of Theorem 14, we obtain the following strong non-malleable extractor (even though the construction of [1] is not strong).

Corollary 16. *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* $(\mathsf{Enc}, \mathsf{Dec})$ *of block length* n *and rate at least* $1-\gamma_0$ *such that the coding scheme is strongly non-malleable against bit-tampering adversaries and achieves error upper bounded by* $\exp(-\tilde{\Omega}(n^{1/7}))$. □

4.3 Proof of Theorem 12

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 $f \colon \{0,1\}^N \to \{0,1\}^N$. 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.
- $\mathsf{Fl} \subseteq [N] \setminus \mathsf{Fr}$; the set of positions flipped by f.
- $\mathsf{Id} = [N] \setminus (\mathsf{Fr} \cup \mathsf{Fl})$; 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 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 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, 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 5 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 [12] 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. However, in order to guarantee strong non-malleability, we need the following further claim.

Claim. Suppose $t_2 \leqslant n_2/2$. Then, regardless of the choice of the message s, $\Pr[f(Z', C') = (Z', C')] = \exp(-\Omega(\gamma_0 n)) =: \epsilon'_1$.

Proof. We upper bound the probability that the adversary leaves C' unchanged. Consider the action of f on $C = (C_1, \ldots, C_{n_b})$ (which is a permutation of how f acts on each bit according to the realization of Π). Recall that all but at most t_2/b of the bits of C (and hence, all but at most t_2/b of the n_b blocks of C) are frozen to 0 or 1 by f. Let $I \subseteq [n_b]$ denote the set of blocks of C that are completely frozen by f. We can see that $|I| \geqslant n_b/2$ by the assumption that $t_2 \leqslant n_2/2 = n_b b/2$.

In the sequel, we fix the realization of S' to any fixed string. Regardless of this conditioning, the blocks of C picked by I are independent, and each block is $\Omega(\gamma_0 B)$-wise, $\exp(-\Omega(\gamma_0 B))$-dependent by property 4 of Lemma 10. It follows that for each block $i \in I$, the probability that C_i coincides with the frozen value of the ith block as defined by f is bounded by $\exp(-\Omega(\gamma_0 B))$. Since the blocks of C picked by I are independent, we can amplify this probability and conclude that the probability that f leaves $(C_i)_{i\in I}$ (and consequently, (Z', C')) unchanged is at most

$$\exp(-\Omega(\gamma_0 B|I|)) = \exp(-\Omega(\gamma_0 B n_b/2)) = \exp(-\Omega(\gamma_0 n)) \ .$$

Consider the distribution $\mathcal{D}_{f,s}$ in Definition 5. From Claim 4.3, it follows that the probability mass assigned to $\underline{\mathsf{same}}$ for this distribution is at most $\epsilon_1' = \exp(-\Omega(\gamma_0 n))$ for every s, which implies

$$\mathcal{D}_{f,s} \approx_{\epsilon_1'} \mathscr{D}(\mathsf{Dec}(f(\mathsf{Enc}(s)))),$$

since the right hand side distribution is simply obtained from $\mathcal{D}_{f,s}$ by moving the probability mass assigned to $\underline{\mathsf{same}}$ to s. Since we have shown that the distribution of $\mathsf{Dec}(f(\mathsf{Enc}(s)))$ is the same for every message s, it follows that for every $s, s' \in \{0,1\}^k$,

$$\mathcal{D}_{f,s} \approx_{2\epsilon_1'} \mathcal{D}_{f,s'},$$

which proves strong non-malleability in this case.

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. Our goal is to upper bound the probability that Dec does not output $\perp$ under the above assumptions. 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. We know that $|\mathsf{Id}' \cup \mathsf{Fl}| > t_2/b$. Moreover, 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.

If $|\mathsf{Id}'| > n - \delta_2 n_b$, 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 the remaining errors in S' are detected by the decoder, and thus, in this case the decoder always outputs $\perp$; a value that is independent of the original message s. Thus in the sequel we can assume that $|\mathsf{Fr} \cup \mathsf{Fl}| \geqslant \delta_2 n_2/b$. Moreover, we fix randomness of the LECSS $\mathcal{C}_2$ 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.

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, Π is exactly a 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

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

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

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

and

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

which together imply

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

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 lower bound can only be affected by δ. Thus, under the assumption that

$$\delta \leqslant \gamma_2''/2, \tag{3}$$

we may still ensure that

$$\Pr[\mathcal{E}_1 \wedge \mathcal{E}_2] \geqslant \gamma_2''/2. \tag{4}$$

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 (4). However, by property 5 of Lemma 10 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

$$\Pr[X_i = 1] \geqslant \gamma_2''/6. \tag{5}$$

Observe that by the argument above, (5) 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 $X_0 := 0$,

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

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

$$\begin{aligned} \Pr[\mathsf{Dec}(\bar{Z}', \bar{C}') \neq \perp] &\leqslant \Pr[X_1 = X_2 = \cdots = X_{\lfloor \ell/B \rfloor} = 0] & (7)\\ &\leqslant \left(1 - \gamma_2''/6\right)^{\lfloor \ell/B \rfloor}, & (8) \end{aligned}$$

where (7) holds since the left hand side event is a subset of the right hand side event, and (8) follows from (6) and the chain rule.

Case 3: The decoder estimates an independent permutation.

In this case, we consider the event where $\bar{\Pi}$ 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

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

Since f is fixed to an arbitrary choice only with restrictions on the number of frozen bits, without loss of generality we can assume that $\bar{\pi}$ is the identity permutation (if not, permute the action of f accordingly), and therefore, $\bar{C}' = \bar{C}$ (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 (1)), 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

$$\ell \leqslant n/2, \tag{9}$$

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 10), each individual bit of C is $\exp(-\Omega(\gamma_0 B))$-close to uniform. Therefore, with probability at least $1 - 2B^2/n - \delta$ (in particular, at least 7/8 when

$$n \geqslant 32B^2 \tag{10}$$

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 10) implies that

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

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

$$\Pr[\mathsf{Dec}(\bar{Z}', \bar{C}') \neq \perp] \leqslant (7/8)^{|T'|} \leqslant \exp(-\Omega(\ell/B^2)), \tag{11}$$

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

The general case and setting up the parameters.

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 ensure 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. Formal details of this argument, as well as the appropriate setting of the parameters leading to Theorem 12, appear in the full version of the paper.

5 Construction of non-malleable codes using non-malleable extractors

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 [10]) 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[9].

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. [11] 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.

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 [10]) 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 3 below) is somewhat cleaner and easier to work with than then definition of non-malleable codes (Definition 4) that involves subtleties such as the extra care for the "$\underline{\mathsf{same}}$" symbol.

As discussed in Section 5.2, our reduction can be modified to obtain non-malleable codes for different classes of adversaries (by appropriately defining the family of extractors based on the tampering family being considered).

5.1 Seedless non-malleable extractors

First, we introduce the following 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 17. 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 may naturally define seedless non-malleable extractors. Roughly speaking, a seedless non-malleable extractor is a seedless extractor (in the traditional sense) that is also a non-malleable function with respect to a certain class of tampering functions. The general definition is deferred to the final version of the paper. However, for our applications we are

[9] For a background on standard seeded and seedless extractors, see [4, Chapter 2].

particularly interested in the special case of two-source non-malleable extractors which is defined below.

Definition 18. 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 $(X, Y) \mapsto (f_1(X), f_2(Y))$.

The theorem below, proved in the full version, shows that non-malleable two-source extractors exist and in fact a random function is w.h.p. such an extractor.

Theorem 19. *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* $2m \leqslant k_1 + k_2 - 3\log(1/\epsilon) - \log\log(1/\gamma)$*, and* $\min\{k_1, k_2\} \geqslant \log n + \log\log(1/\gamma) + O(1)$*.* □

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 "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 20. 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 the following three tampering functions: $(X, Y) \mapsto (f_1(X), Y)$; $(X, Y) \mapsto (X, f_2(Y))$; and $(X, Y) \mapsto (f_1(X), f_2(Y))$.

Remark 3. In order to satisfy the requirements of Definition 20, it suffices (but not necessary) to ensure

$$(\mathsf{NMExt}(\mathcal{X}, \mathcal{Y}), \mathsf{NMExt}(f_1(\mathcal{X}), \mathcal{Y})) \approx_\epsilon (U_m, \mathsf{NMExt}(f_1(\mathcal{X}), \mathcal{Y})),$$
$$(\mathsf{NMExt}(\mathcal{X}, \mathcal{Y}), \mathsf{NMExt}(\mathcal{X}, f_2(\mathcal{Y}))) \approx_\epsilon (U_m, \mathsf{NMExt}(\mathcal{X}, f_2(\mathcal{Y}))),$$
$$(\mathsf{NMExt}(\mathcal{X}, \mathcal{Y}), \mathsf{NMExt}(f_1(\mathcal{X}), f_2(\mathcal{Y}))) \approx_\epsilon (U_m, \mathsf{NMExt}(f_1(\mathcal{X}), f_2(\mathcal{Y}))).$$

The proof of Theorem 19 shows that these stronger requirements (which are quite similar to the definition of seeded non-malleable extractor in [10]) 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 18) is a relaxed non-malleable two-source extractor (according to Definition 20) and with the same parameters. Interestingly, below we show that the two notions are equivalent up to a slight loss in the parameters (see the full version for a proof).

Lemma 21. *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.* □

5.2 From non-malleable extractors to non-malleable codes

In this section, we present our reduction from non-malleable extractors to non-malleable codes. For concreteness, we focus on tampering functions in the split-state model. 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. In this case, the needed extractor is a non-malleable variation of the *mixed-sources extractors* studied by Raz and Yehudayoff [22].

We note that Theorem 22 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 21) strengthens 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 22. *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. Deferred to the full version of the paper.

We can now derive the following corollary, which is the main result of this section, using Lemma 21 and Theorem 22 (see the full version for a proof).

Corollary 23. *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.* □

Finally, using the above tools and the existence result of Theorem 19, we conclude that there are non-malleable two-source extractors defining coding schemes in the split-state model and achieving constant rates; in particular, rates arbitrarily close to 1/5.

Corollary 24. *For every* $\alpha > 0$*, there is a choice of* NMExt *in Theorem 22 that makes* $(\mathsf{Enc}, \mathsf{Dec})$ *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 19 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 22 to attain the desired result.

References

1. D. Aggarwal, Y. Dodis, and S. Lovett. Non-malleable codes from additive combinatorics. *ECCC* Technical Report TR13-081, 2013.
2. B. Barak, A. Rao, R. Shaltiel, and A. Wigderson. 2-source dispersers for subpolynomial entropy and Ramsey graphs beating the Frankl-Wilson construction. *Annals of Mathematics*, 176(3):1483–1544, 2012.
3. J. Bourgain. More on the Sum-Product phenomenon in prime fields and its applications. *International Journal of Number Theory*, 1(1):1–32, 2005.
4. M. Cheraghchi. *Applications of Derandomization Theory in Coding.* PhD thesis, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 2010. (available online at `http://eccc.hpi-web.de/static/books/Applications_of_Derandomization_Theory_in_Coding/`).
5. M. Cheraghchi and V. Guruswami. Capacity of non-malleable codes. *ECCC* Technical Report TR13-118, 2013.
6. M. Cheraghchi and V. Guruswami. Explicit optimal rate non-malleable codes for bit-tampering. *IACR* Technical Report 2013/565 (`http://eprint.iacr.org/2013/565`), 2013.
7. B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 2(17):230–261, 1988.

8. R. Cramer, Y. Dodis, S. Fehr, C. Padró, and D. Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In *Proceedings of EUROCRYPT 2008*, pages 471–488, 2008.
9. Y. Dodis, X. Li, T.D. Wooley, and D. Zuckerman. Privacy amplification and non-malleable extractors via character sums. In *Proceedings of FOCS 2011*, pages 668–677, 2011.
10. Y. Dodis and D. Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *Proceedings of the 41st annual ACM Symposium on Theory of Computing*, pages 601–610, 2009.
11. S. Dziembowski, T. Kazana, and M. Obremski. Non-malleable codes from two-source extractors. In *Proceedings of CRYPTO*, pages 239–257, 2013.
12. S. Dziembowski, K. Pietrzak, and D. Wichs. Non-malleable codes. In *Proceedings of Innovations in Computer Science (ICS 2010)*, 2010.
13. G.D. Forney. *Concatenated Codes.* MIT Press, 1966.
14. G. Gohen, R. Raz, and G. Segev. Non-malleable extractors with short seeds and applications to privacy amplification. In *Proceedings of CCC 2012*, pages 298–308, 2012.
15. V. Guruswami and A. Smith. Codes for computationally simple channels: Explicit constructions with optimal rate. In *Proceedings of FOCS 2010*, pages 723–732, 2010.
16. J. Justesen. A class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18:652–656, 1972.
17. Y. Kalai, X. Li, and A. Rao. 2-source extractors under computational assumptions and cryptography with defective randomness. In *Proceedings of the* 50*th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 617–626, 2009.
18. E. Kaplan, M. Naor, and O. Reingold. Derandomized constructions of k-wise (almost) independent permutations. In *Proceedings of RANDOM 2005*, pages 113–133, 2005.
19. X. Li. Non-malleable extractors, two-source extractors and privacy amplification. In *Proceedings of the* 53*rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 688–697, 2012.
20. A. Rao. A 2-source almost-extractor for linear entropy. In *Proceedings of RANDOM 2008*, pages 549–556, 2008.
21. R. Raz. Extractors with weak random seeds. In *Proceedings of the* 37*th Annual ACM Symposium on Theory of Computing (STOC)*, pages 11–20, 2005.
22. R. Raz and A. Yehudayoff. Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors. *Journal of Computer and System Sciences*, 77(1):167–190, 2011.