Non-malleable Reductions and Applications

Divesh Aggarwal* Yevgeniy Dodis† Tomasz Kazana‡ Maciej Obremski§

April 10, 2015

Abstract

Non-malleable codes, introduced by Dziembowski, Pietrzak and Wichs [DPW10], provide a useful message integrity guarantee in situations where traditional error-correction (and even error-detection) is impossible; for example, when the attacker can completely overwrite the encoded message. Informally, a code is non-malleable if the message contained in a modified codeword is either the original message, or a completely "unrelated value". Although such codes do not exist if the family of "tampering functions" $\mathcal{F}$ allowed to modify the original codeword is completely unrestricted, they are known to exist for many broad tampering families $\mathcal{F}$. The family which received the most attention [DPW10, LL12, DKO13, ADL14, CG14a, CG14b] is the family of tampering functions in the so called (2-part) *split-state* model: here the message x is encoded into two shares L and R, and the attacker is allowed to *arbitrarily* tamper with each L and R *individually*. Despite this attention, the following problem remained open:

Build efficient, information-theoretically secure non-malleable codes in the split-state model with constant encoding rate: $|L| = |R| = O(|x|)$.

In this work, we resolve this open problem. Our technique for getting our main result is of independent interest. We

(a) develop a generalization of non-malleable codes, called *non-malleable reductions*;

(b) show simple composition theorem for non-malleable reductions;

(c) build a variety of such reductions connecting various (independently interesting) tampering families $\mathcal{F}$ to each other;

(d) construct several new non-malleable codes in the split-state model by applying the composition theorem to a series of easy to understand reductions.

Most importantly, we show several "independence amplification" reductions, showing how to reduce split-state tampering of very few parts to an easier question of split-state tampering with a much larger number of parts. In particular, our final, constant-rate, non-malleable code composes one of these reductions with the very recent, "9-split-state" code of Chattopadhyay and Zuckerman [CZ14].

*School of Computer and Communication Sciences. EPFL. Email: Divesh.Aggarwal@epfl.ch.

†Computer Science Dept. NYU. Email: dodis@cs.nyu.edu. Partially supported by gifts from VMware Labs and Google, and NSF grants 1319051, 1314568, 1065288, 1017471.

‡Computer Science Dept. NYU. Email: tkazana@mimuw.edu.pl.

§Institute of Informatics, University of Warsaw. Email: obremski@mimuw.edu.pl.

1 Introduction

Non-malleable codes, introduced by Dziembowski, Pietrzak and Wichs [DPW10], provide a useful message integrity guarantee in situations where traditional error-correction (and even error-detection) is impossible; for example, when the attacker can completely overwrite the encoded message. Informally, given a tampering family $\mathcal{F}$, a $\mathcal{F}$-non-malleable code (E, D) encodes a given message x into a codeword $y \leftarrow E(x)$ in a way that, if y is modified into $y' = f(y)$ by some $f \in \mathcal{F}$, then the message $x' = D(y')$ contained in the modified codeword y' is either the original message x, or a completely "unrelated value". In other words, non-malleable codes aim to handle a much larger class of tampering functions $\mathcal{F}$ than traditional error-correcting or error-detecting codes, at the expense of potentially allowing the attacker to replace a given message x by an unrelated message x' (and also necessarily allowing for a small "simulation error" ε). As shown by [DPW10], this relaxation still makes non-malleable codes quite useful in a variety of situations where (a) the tampering capabilities of the attacker might be too strong for error-detection, and, yet (b) changing x to unrelated x' is not useful for the attack. For example, imagine x being a secret key for a signature scheme. In this case, tampering which keeps x the same corresponds to the traditional chosen message attack (covered by the traditional definition of secure signatures), while tampering which changes x to an unrelated value x' will clearly not help in forging signatures under the original (un-tampered) verification key, as the attacker can produce such signatures under x' by himself.

Split-State Model. Although such codes do not exist if the family of "tampering functions" $\mathcal{F}$ is completely unrestricted,[1] they are known to exist for many broad tampering families $\mathcal{F}$. One such natural family is the family of tampering functions in the so called *t-split-state* model $\mathcal{S}_n^t$. Here the k-bit message x is encoded into t shares $y_1, \ldots, y_t$ of length n each, and the attacker is allowed to *arbitrarily* tamper with each y_i *individually*. The rate of such an encoding is naturally defined as $\tau = tn/k$.

The appeal of this family comes from the fact that it seems naturally enforceable in applications, especially when t is low and the shares $y_1, \ldots, y_t$ are stored in different parts of memory, or by different parties. Alternatively, non-malleable codes in this model could be interpreted as "non-malleable secret-sharing schemes": even if *all* the t message shares are independently tampered with, the recovered message is either x or is unrelated to x. Not surprisingly, the setting of $t = 2$ appears the most useful (but also the most challenging from the technical point of view), so it received the most attention so far [DPW10, LL12, DKO13, ADL14, CG14a, CG14b].

The known results can be summarized as follows. First, [DPW10] showed the existence of such non-malleable codes, and this existential result was further improved by [CG14a], who (amazingly!) showed that the optimal rate of such codes is just 2. Second, the work of [DPW10] also gave an efficient construction in the random oracle model. Third, the work of Liu and Lysyanskaya [LL12] built an efficient *computationally-secure* non-malleable code in the split model (necessarily restricting the tampering functions f_1 and f_2 to be efficient as well). The construction assumes so called common reference string (CRS) which cannot be tampered, and also uses quite heavy tools from public-key cryptography, such as robust non-interactive zero-knowledge proofs [DSDCO$^+$01] and leakage-resilient encryption [NS09]. Thus, given the clean information-theoretic definition of non-malleable codes, we believe it is important to construct such codes unconditionally.

This was first achieved by Dziembowski, Kazana and Obremski [DKO13], who constructed an elegant non-malleable code for 1-bit messages in the split-state model. Following that, Aggar-

[1] In particular, $\mathcal{F}$ should not include "re-encoding functions" $f(y) = E(f'(D(y)))$ for any non-trivial function f', as $x' = D(f(E(y))) = f'(x)$ is obviously related to x.

wal, Dodis and Lovett [ADL14] gave the first information-theoretic construction supporting k-bit messages, but where the length of each share $n = O(k^7 \log^7 k)$ [ADL14]. The security proof of this scheme also used pretty advanced results from additive combinatorics, including the so called *Quasi-polynomial Freiman-Ruzsa Theorem*, which was recently established by Sanders [San12] as a step towards resolving the Polynomial Freiman-Ruzsa conjecture [Gre05]. This construction was improved by Aggarwal [Agg14] to obtain non-malleable codes in the split state model with $n = O(k^7)$. Very recently Chattopadhyay and Zuckerman [CZ14] construct a constant-rate non-malleable code in the 9-split-state model. However, it was unclear how to reduce the number of independent parts to the optimal 2.

Hence, prior to our work, the following question remained open: *construct efficient, information-theoretically secure non-malleable codes in the* 2*-split-state model whose rate is* $o(k^6)$ *(and, ideally, constant).*

Our Results. In this work, we resolve this open problem.

Theorem 1 (Main Result). *(Informal) There exists efficient, information-theoretically secure non-malleable codes in the* 2*-split-state model with constant encoding rate:* $|L| = |R| = O(k)$*, where* k *is the length of the message.*

Our technique for getting this result is of independent interest. We

(a) develop a generalization of non-malleable codes, called *non-malleable reductions*;

(b) show simple composition theorem for non-malleable reductions;

(c) construct a variety of such reductions connecting various (independently interesting) tampering families $\mathcal{F}$ to each other; and

(d) construct our final, constant-rate, non-malleable code in the 2-split-state model by applying the composition theorem to a series of easy to understand reductions.

We briefly expand on these results below, but notice that our final result uses the above mentioned recent result of Chattopadhyay and Zuckerman [CZ14] *as a black-box*. Without using this work, we could directly achieve a very simple linear-rate $\tau = O(k)$ non-malleable code in the 2-split-state model, which is already considerably better than the prior state-of-the-art $\tau = O(k^6)$ [ADL14, Agg14].

Non-malleable Reductions. Recall, non-malleable codes encode the message x in a way that decoding a tampered codeword either returns x itself, or yields an "independent" message x'. Abstractly, this could be viewed as "reducing" a possibly complicated family of tampering functions $\mathcal{F}$ to a much simpler family NM of what we call trivial tampering functions: identity function $f(x) = x$ and constant functions $f_{x'}(x) = x'$. More generally, given two families $\mathcal{F}$ and $\mathcal{G}$, we can define a *non-malleable reduction* from $\mathcal{F}$ to $\mathcal{G}$ — denoted $(\mathcal{F} \Rightarrow \mathcal{G})$ — to be a pair (E, D) of encoding/decoding functions with the property that, for any tampering function $f \in \mathcal{F}$, the function $D(f(E(\cdot)))$ is "close" to a convex combination of functions $g(\cdot)$ for $g \in \mathcal{G}$. With this perspective, non-malleable code w.r.t. to $\mathcal{F}$ is simply a non-malleable reduction $(\mathcal{F} \Rightarrow \mathsf{NM})$. More interestingly, and ignoring error terms, it is very easy to see that the notion of non-malleable reductions is transitive: $(\mathcal{F} \Rightarrow \mathcal{G})$ and $(\mathcal{G} \Rightarrow \mathcal{H})$ imply $(\mathcal{F} \Rightarrow \mathcal{H})$. Thus, to construct a non-malleable code w.r.t. to some possibly complicated family $\mathcal{F}$, we can define some useful intermediate families $\mathcal{F}_0 = \mathcal{F}, \mathcal{F}_1, \ldots, \mathcal{F}_i = \mathsf{NM}$ (for small constant i), and show that $(\mathcal{F}_0 \Rightarrow \mathcal{F}_1), \ldots, (\mathcal{F}_{i-1} \Rightarrow \mathcal{F}_i)$.

Aside from improved modularity, our approach has the benefit that some of our intermediate families and reductions are rather natural and could find other applications. Additionally, if a

better intermediate non-malleable reduction is found in subsequent/independent work, we could immediately get an improved result for our final non-malleable code. This is precisely what happened when we discovered the recent work of Chattopadhyay and Zuckerman [CZ14], which, in our terminology, gave a better non-malleable reduction from $O(1)$-split-state family to the trivial family NM. Coupled with our already established constant-rate reduction from 2-split-state to $O(1)$-split-state family, the work of [CZ14] improved the rate of our final code from $O(k)$ to $O(1)$, giving us the desired code stated in Theorem 1.

Our Reductions. As we mentioned, we introduce several useful intermediate families and derive a variety of non-malleable reductions relating them. From a conceptual point of view, however, we present two incomparable non-malleable reductions (each of which is composed of several sub-reductions). Both of these reductions could be interpreted as *independence amplification* techniques: they reduce split-state tampering of very few parts to an easier question of split-state tampering with a much larger number of parts.

Our first main result (see Theorem 18) shows a non-malleable reduction from 5-split-state tampering to t-split-state tampering, loosing only a factor $O(t)$ in the rate of the code. In addition to the 5-split-state tampering, it can also tolerate so called "forgetful" family $\mathcal{FOR}^5$, which is allowed to (dependently) tamper all 5 memory parts as a function of any $(5-1)=4$ memory parts. (More generally, $\mathcal{FOR}^t$ can use any $(t-1)$ parts.) In turn, this reduction is composed of several sub-reductions, some of which are of independent interest (e.g., one reduction uses the alternating extraction technique of [DP07a] to reduce 2-split-state tampering to the so called family of "lookahead functions", which is a natural model for "one-pass" tampering). We defer more detailed treatment to Section 5, here only mentioning that each of our reductions is *rather elementary to state* (but not prove), using only general randomness extractors or the inner product function. In particular, the resulting non-malleable codes that we get using this reduction could be "efficient" not only in theory, but even in practice.

Our second main result (see Theorem 19) is a non-malleable reduction from 2-split-state tampering to the family containing t-split-state tampering and the t-part forgetful family $\mathcal{FOR}^t$ mentioned above. This reduction loses a factor $O(t^3)$ in the rate, but this is still a constant when $t=O(1)$. Also, although the proof of this reduction is, by far, the most technically involved part of this work, the reduction itself is *very simple and efficient*, using only the inner product function. We defer more detailed treatment of this result to Section 6.

Applications to Non-malleable Codes. We can now compose our main new reductions with the already known constructions of non-malleable codes for various families, to get the following new results. First, composing our reduction from 5 parts to t parts with known non-malleable codes in the so called independent-bit tampering model (where each of the t shares is only 1 bit) [DPW10, CG14b, FMVW14], we get a very simple linear rate non-malleable code in the 5-split-state model. See Theorem 21.

Second, we can now compose this code with our second reduction (from 2 parts to $t=5$ parts or the forgetful family $\mathcal{FOR}^5$) to get still quite simple linear rate non-malleable code in the 2-split-state model. As we mentioned, this already considerably improves the prior state-of-the-art $O(k^6)$ rate code by [ADL14, Agg14]. See Theorem 22.

Finally, instead of our own non-malleable code in the $t=5$ split-state model above, we can use the beautiful recent work of [CZ14], which uses a variety of advanced techniques to construct a constant-rate non-malleable code in the 9-split-state model (i.e., number of parts $t=9$). Composing this constant-rate code with our second reduction from 2 to $t=9$ part, which only loses a constant

factor in the rate, we get our final code claimed in Theorem 1 (and, formally, in Theorem 24).

Other Related Work. Other results that look at an (enhanced) split-state model are Faust et al. [FMNV14] which consider the model where the adversary can tamper continuously, and [ADKO14], that considers the model where the adversary, in addition to performing split-state tampering, is also allowed some limited interaction between the two states.

In fact, the result of [ADKO14] combined with our result gives a constant rate non-malleable code that also allows leakage of a 1/12-th fraction of the bits from one share of the codeword to the other. In addition to the already-mentioned results, several recent works [CCFP11, CCP12, CKM11, FMVW14, **?**, **?**] either used or built non-malleable codes for various families $\mathcal{F}$, but did not concentrate on the split-state model, which is our focus here.

The notion of non-malleability was introduced by Dolev, Dwork and Naor [DDN00], and has found many applications in cryptography. Traditionally, non-malleability is defined in the computational setting, but recently non-malleability has been successfully defined and applied in the information-theoretic setting (generally resulting in somewhat simpler and cleaner definitions than their computational counter-parts). For example, in addition to non-malleable codes studied in this work, the work of Dodis and Wichs [DW09] defined the notion of non-malleable extractors as a tool for building round-efficient privacy amplification protocols.

Finally, the study of non-malleable codes falls into a much larger cryptographic framework of providing counter-measures against various classes of tampering attacks. This work was pioneered by the early works of [ISW03, GLM+03, IPSW06], and has since led to many subsequent models. We do not list all such tampering models, but we refer to [KKS11, LL12] for an excellent discussion of various such models.

2 Preliminaries

For a set T, let U_T denote a uniform distribution over T, and, for an integer ℓ, let U_ℓ denote uniform distribution over ℓ bit strings. The *statistical distance* between two random variables A, B is defined by $\Delta(A\ ;\ B) = \frac{1}{2}\sum_v |\Pr[A = v] - \Pr[B = v]|$. We use $A \approx_\varepsilon B$ as shorthand for $\Delta(A, B) \leq \varepsilon$.

Lemma 2. *For any function α, if $\Delta(A\ ;\ B) \leq \varepsilon$, then $\Delta(\alpha(A)\ ;\ \alpha(B)) \leq \varepsilon$.*

The *min-entropy* of a random variable W is $\mathbf{H}_\infty(W) \stackrel{\text{def}}{=} -\log(\max_w \Pr[W = w])$, and the *conditional* min-entropy of W given Z is $\mathbf{H}_\infty(W|Z) \stackrel{\text{def}}{=} -\log\left(\mathbb{E}_{z\leftarrow Z} \max_w \Pr[W = w|Z = z]\right)$.

Definition 3. We say that an efficient function $\mathsf{Ext} : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^n$ is an $(N, m, d, n, \varepsilon)$-*extractor* if for all sources (W, Z) of conditional min-entropy $\mathbf{H}_\infty(W|Z) \geq m$, we have $(S, Z, \mathsf{Ext}(W; S)) \approx_\varepsilon (S, Z, U_n)$, where S is uniform on $\{0,1\}^d$.

In Section 5, we will be concerned with the case when $d = n$ (seed length equals output length), and will use the existence of the following extractors:

Lemma 4 ([GUV07]). *There exist constants c_1 and c_2, such that for any N and n satisfying $n \in [c_1 \cdot \log N, N/2]$, there exists an explicit, efficient $(N, m = 2n, d = n, n, \varepsilon = 2^{-c_2 \cdot n})$-extractor* Ext.

We also also use bit-extractors which extract only one bit. One such extractor is the bit inner product function $\langle W, S\rangle$ (which trivially follows from the Leftover Hash Lemma [HILL99]). We

state this below, for future convenience renaming the source length to n (and no longer using n for output size, as the latter is 1):

Lemma 5. *The inner product function is an $(n, m, n, \ell, 2^{-(m-\ell-1)/2})$-extractor.*

Definition 6. *We say that a function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ is an (n, k, m, ε)-*2-source extractor *if for all independent sources $X, Y \in \{0,1\}^n$ such that min-entropy $\mathbf{H}_\infty(X) + \mathbf{H}_\infty(Y) \geq k$, we have $(Y, \mathsf{Ext}(X, Y)) \approx_\varepsilon (Y, U_m)$, and $(X, \mathsf{Ext}(X, Y)) \approx_\varepsilon (X, U_m)$.*

For n being an integer multiple of m, and interpreting elements of $\{0,1\}^m$ as elements from $\mathbb{F}_{2^m}$ and those in $\{0,1\}^n$ to be from $(\mathbb{F}_{2^m})^{n/m}$, we have that the inner product function is a good 2-source extractor.

Lemma 7. *For all positive integers m, n such that n is a multiple of m, and for all $\varepsilon > 0$, there exists an efficient $(n, n + m + 2\log\left(\frac{1}{\varepsilon}\right), m, \varepsilon)$ 2-source extractor.*

We will need the following results. We include proofs in Appendix A for completeness.
The following is a simple result from [ADL14].

Lemma 8. *Let $X_1, Y_1 \in \mathcal{A}_1$, and $Y_1, Y_2 \in \mathcal{A}_2$ be random variables such that $\Delta((X_1, X_2)\ ;\ (Y_1, Y_2)) \leq \varepsilon$. Then, for any non-empty set $\mathcal{A}' \subseteq \mathcal{A}_1$, we have*

$$\Delta(X_2 \mid X_1 \in \mathcal{A}'\ ;\ Y_2 \mid Y_1 \in \mathcal{A}') \leq \frac{2\varepsilon}{\Pr(X_1 \in \mathcal{A}')}\ .$$

The following result is from [DP07a].

Lemma 9. *Let $A \in \mathcal{A}$ and $B \in \mathcal{B}$ be two independent random variables. Let $V_1, V_2, \ldots$ be random variables defined as functions of A, B satisfying the following property. For all $i \in \mathbb{N}$, if i is even then $V_i = \phi_i(V_1, \ldots, V_{i-1}, A)$ and if i is odd, then $V_i = \phi_i(V_1, \ldots, V_{i-1}, B)$ for some function ϕ_i. Then for all i, A is independent of B given $V_1, \ldots, V_i$.*

The following is (a generalization of) the Vazirani's XOR Lemma.

Lemma 10. *Let $X = (X_1, \ldots, X_t) \in \mathbb{F}^t$ be a random variable, where $\mathbb{F}$ is a finite field of order q. Assume that for all $a_1, \ldots, a_t \in \mathbb{F}^t$ not all zero, $\Delta(\sum_{i=1}^t a_i X_i\ ;\ U) \leq \varepsilon$, where U is uniform in $\mathbb{F}$. Then $\Delta(X_1, \ldots, X_t\ ;\ U_1, \ldots, U_t) \leq \varepsilon q^{(t+2)/2}$, where $U_1, \ldots, U_t$ are independent and each is uniform in $\mathbb{F}$.*

3 Non-malleable Reductions and Useful Tampering Families

Definitions. In the following we generalize the notion of non-malleable codes w.r.t. to a tampering family $\mathcal{F}$ [DPW10] to a more versatile notion of *non-malleable reductions* from $\mathcal{F}$ to $\mathcal{G}$.

Definition 11 (non-malleable reduction). Let $\mathcal{F} \subset A^A$ and $\mathcal{G} \subset B^B$ be some classes of functions (which we call *manipulation* functions). We will write:

$$(\mathcal{F} \Rightarrow \mathcal{G}, \varepsilon)$$

and say $\mathcal{F}$ *reduces to* $\mathcal{G}$, if there exist an efficient randomized *encoding* function $E : B \to A$, and an efficient deterministic *decoding* function $D : A \to B$, such that (a) for all $x \in B$, we have $D(E(x)) = x$, and (b) for all $f \in \mathcal{F}$, there exists G such that for all $x \in B$,

$$\Delta\Big(D(f(E(x)))\ ;\ G(x)\Big) \leq \varepsilon, \tag{1}$$

where G is a *distribution* over $\mathcal{G}$, and $G(x)$ denotes the distribution $g(x)$, where $g \leftarrow G$.

The pair (E, D) is called $(\mathcal{F}, \mathcal{G}, \varepsilon)$-*non-malleable reduction.*

Intuitively, $(\mathcal{F}, \mathcal{G}, \varepsilon)$-non-malleable reduction allows one to encode a value x by $y \leftarrow E(x)$, so that tampering with y by $y' = f(y)$ for $f \in \mathcal{F}$ gets "reduced" (by the decoding function $D(y') = x'$) to tampering *with x itself* via some (distribution over) $g \in \mathcal{G}$.

In particular, the notion of *non-malleable code* w.r.t. $\mathcal{F}$, is simply a reduction from $\mathcal{F}$ to the family of "trivial manipulation functions" NM_k defined below.

Definition 12. Let NM_k denote the set of *trivial manipulation functions* on k-bit strings, which consists of the identity function $I(x) = x$ and all constant functions $f_c(x) = c$, where $c \in \{0,1\}^k$.

We say that a pair (E, D) defines an $(\mathcal{F}, k, \varepsilon)$-*non-malleable code*, if it defines a $(\mathcal{F}, \mathsf{NM}_k, \varepsilon)$-non-malleable reduction.

Remark 1. The above definition might seem a little different than the definition of [DPW10] (who required a simulator outputting a distribution over constants $c \in \{0,1\}^k$, a special symbol "same", serving as a disguise for the identity function, and another special symbol $\perp$). The symbol $\perp$ is meant to indicate that the tampered codeword is invalid, and facilitates one to view non-malleable codes as a relaxation of error-detecting codes, where one wants to detect tampering. However, one can equivalently consider the non-malleable code defintion without $\perp$, simply by replacing the "bottom output" $\perp$ by a fixed message whenever the simulator or decoder outputs $\perp$. We formally discuss this issue in Appendix B, where we also show the equivalence between the definition of non-malleable code presented here and the one in [DPW10].

Remark 2. Notice, the "complexity" of the initial tampering family $\mathcal{F}$ intuitively corresponds to the complexity of the attacker on our system. Hence, when $\mathcal{F}$ consists of efficient functions (and so does the target family $\mathcal{G}$; e.g. $\mathcal{G} = \mathsf{NM}_k$), it could be useful to require that the distribution G over $\mathcal{G}$ is efficiently samplable given oracle access to $f \in \mathcal{F}$. However, we do not insist on this for two reasons: (1) our final tampering family $\mathcal{F}$ (the split-state family) will consist of arbitrary and possibly inefficient functions f, making the efficiency requirement on G less motivated; and, more importantly, (2) for the reduction from any family $\mathcal{F}$ to the trivial family NM_k (which is our final goal), the requirement that G is efficiently samplable (given oracle access to $f \in \mathcal{F}$) can be anyway ensured with mild loss of parameters, as already observed by [CG14b]. We formally state this in Lemma 13 (see Appendix B for proof). Hence, we will keep our simpler definition, but stress that our final distribution G (when $\mathcal{G} = \mathsf{NM}_k$) could be made efficiently samplable, by Lemma 13.

Lemma 13. *Let (E, D) be an $(\mathcal{F}, k, \varepsilon)$-non-malleable code for some tampering family $\mathcal{F}$. Then for all $f \in \mathcal{F}$, there exists a random function G distributed over NM_k such that for all $x \in \{0,1\}^k$,*

$$\Delta\Big(D(f(E(x)))\ ;\ G(x)\Big) \leq 2\varepsilon + \frac{1}{2^k}\ ,$$

and G is efficiently samplable given oracle access to f.

We also give a related useful notion of non-malleable transformations, where the $\mathcal{F}$-tampering is applied to *uniformly random* strings in A, and gets transformed (by some "transformation function" T) to $\mathcal{G}$-tampering over *uniformly random* strings in B.

Definition 14 (non-malleable transformation). Let $\mathcal{F} \subset A^A$ and $\mathcal{G} \subset B^B$ be some classes of manipulation functions. We will write:

$$(\mathcal{F} \to \mathcal{G}, \varepsilon)$$

and say $\mathcal{F}$ *transforms to* $\mathcal{G}$, if there exists an efficient *transformation* function $T : A \to B$ such that for all $f \in \mathcal{F}$ there exists G such that:

$$\Delta\Big(T(f(U_A)), T(U_A)\ ;\ G(U_B), U_B\Big) \leq \varepsilon, \tag{2}$$

where G is a distribution over $\mathcal{G}$, and $G(x)$ denotes the distribution $g(x)$, where $g \leftarrow G$.

The function T is called $(\mathcal{F}, \mathcal{G}, \varepsilon)$-*non-malleable transformation.*

Remark 3. Equation (2) implies the following analog of "correctness": $\Delta\Big(T(U_A); U_B\Big) \leq \varepsilon$.

The utility of non-malleable reductions and transformations comes from the following natural composition theorem, which allows to gradually make our tampering families simpler and simpler, until we eventually end up with a non-malleable code (corresponding to the trivial family NM_k).

Theorem 15 (Composition). *(a) If* $(\mathcal{F} \to \mathcal{G}, \varepsilon_1)$ *and* $(\mathcal{G} \to \mathcal{H}, \varepsilon_2)$*, then* $(\mathcal{F} \to \mathcal{H}, \varepsilon_1 + \varepsilon_2)$*. (b) Similarly, if* $(\mathcal{F} \Rightarrow \mathcal{G}, \varepsilon_1)$ *and* $(\mathcal{G} \Rightarrow \mathcal{H}, \varepsilon_2)$*, then* $(\mathcal{F} \Rightarrow \mathcal{H}, \varepsilon_1 + \varepsilon_2)$*.*

Proof. We give the proof for (slightly more involved) part (b), as the proof for part (a) is analogous. Since $(\mathcal{F} \Rightarrow \mathcal{G}, \varepsilon_1)$, there exists functions (E_1, D_1) satisfying Equation (1), and same for (E_2, D_2) for $(\mathcal{G} \Rightarrow \mathcal{H}, \varepsilon_2)$. We claim that $(E_1 \circ E_2, D_2 \circ D_1)$ is a correct reduction for $(\mathcal{F} \Rightarrow \mathcal{H}, \varepsilon_1 + \varepsilon_2)$. The correctness property is obvious, and the security follows from these equations:

$$((D_2 \circ D_1)(f((E_1 \circ E_2)(x)))) = (D_2(D_1(f(E_1(E_2(x)))))) \approx_{\varepsilon_1} D_2(G(E_2(x))) \approx_{\varepsilon_2} H(x)$$

which means

$$((D_2 \circ D_1)(f((E_1 \circ E_2)(x)))) \approx_{\varepsilon_1+\varepsilon_2} H(x),$$

as needed. □

We will also need the following trivial observation.

Observation 1 (Union). *Let* (E, D) *be an* $(\mathcal{F}, \mathcal{H}, \varepsilon)$ *and a* $(\mathcal{G}, \mathcal{H}, \varepsilon')$ *non-malleable reduction (resp. transformation). Then* (E, D) *is an* $(\mathcal{F} \cup \mathcal{G}, \mathcal{H}, \max(\varepsilon, \varepsilon'))$ *non-malleable reduction (resp. transformation).*

It is an easy observation that the decoding function in a non-malleable reduction is also a non-malleable transformation, provided it maps uniform strings to uniform strings (which is not always the case). We now show for any efficiently invertible non-malleable transformation T, the pair (T^{-1}, T) is a non-malleable reduction.

Theorem 16. *Let* $\mathcal{F} \subset A^A, \mathcal{G} \subset B^B$*. Assume* $(\mathcal{F} \to \mathcal{G}, \varepsilon)$ *with transformation* T*. For any* $x \in B$*, let* $T^{-1}(x)$ *denote a uniformly random element* U *in* A *such that* $T(U) = x$*. If for all* $x \in B$*,* $T^{-1}(x)$ *is efficiently samplable, then* $(\mathcal{F} \Rightarrow \mathcal{G}, 2\varepsilon|B|)$*.*

Proof. We define $D : A \mapsto B$ to be T and E is defined as $E(x) := T^{-1}(x)$ for all $x \in B$. Then the correctness is obvious. Consider some $x \in B$. Using Equation 2 and Lemma 8, we have that

$$\Delta(D(f(U_A)) \mid D(U_A) = x\ ;\ G(U_B) \mid U_B = x) \leq \frac{2\varepsilon}{\Pr(U_B = x)}\ ,$$

which implies

$$\Delta(D(f(E(x)))\ ;\ G(x)) \leq 2\varepsilon|B|\ .$$

□

USEFUL TAMPERING FAMILIES. We define several natural tampering families we will use in this work. For this, we first introduce the following "direct product" operator on tampering families:

Definition 17. Given tampering families $\mathcal{F} \subset A^A$ and $\mathcal{G} \subset B^B$, let $\mathcal{F} \times \mathcal{G}$ denote the class of functions h from $(A \times B)^{A \times B}$ such that

$$h(x) = h_1(x_1)||h_2(x_2)$$

for some $h_1 \in \mathcal{F}$ and $h_2 \in \mathcal{G}$ and $x = x_1||x_2$, where $x_1 \in A, x_2 \in B$.

We also let $\mathcal{F}^1 := \mathcal{F}$, and, for $t \geq 1$, $\mathcal{F}^{t+1} := \mathcal{F}^t \times \mathcal{F}$.

We can now define the following tampering families:

- $\mathcal{S}_n = (\{0,1\}^n)^{\{0,1\}^n}$ denote the class of *all* manipulation functions on n-bit strings.
- Given $t > 1$, $\mathcal{S}_n^t$ denotes the tampering family in the *t-split-state model*, where the attacker can apply t arbitrarily correlated functions $h_1, \ldots, h_t$ to t separate, n-bit parts of memory (but, of course, each h_i can only be applied to the i-th part individually).
- $\mathcal{FOR}_n^t$ denotes *forgetful* family. It is applied to t parts of memory of length n but the output value can depend only on $(t-1)$ parts. More precisely: Let $x \in \{0,1\}^{tn}$ be a bit vector and $x_i \in \{0,1\}^n$ denote i-th block of n bits. For any $h \in \mathcal{FOR}_n^t$ there exist a subset $S \subset \{1, 2, \ldots, t\}$ of size $(t-1)$ such that $h(x)$ can be evaluated from x_S. Besides that, it is not restricted in any way.
- Finally, $\mathcal{LA}_n^{\leftarrow t} \subset (\{0,1\}^{tn})^{\{0,1\}^{tn}}$ denotes the class of *lookahead manipulation functions* l that can be rewritten as $l = (l_1, \ldots, l_t)$, for $l_i : \{0,1\}^{in} \to \{0,1\}^n$, where

  $$l(x) = l_1(x_1)||l_2(x_1, x_2)|| \ldots ||l_i(x_1, \ldots, x_i)|| \ldots ||l_t(x_1, \ldots, x_t)$$

  for $x = x_1||x_2|| \ldots ||x_t$, and $x_i \in \{0,1\}^n$. In other words, if $l(x_1, \ldots, x_t) = y_1, \ldots, y_t$, then each y_i can only depend on the "prior" $x_1, \ldots, x_i$.

Notice, $\mathsf{NM}_{tn} \subset \mathcal{S}_n^t$ and $\mathcal{S}_n^t \subset \mathcal{LA}_n^{\leftarrow t}$.

4 Our Reductions and Application to Non-malleable Codes

Our Reductions. In this Section, we state our main reductions. Both our reductions could be interpreted as *independence amplification* techniques: they reduce split-state tampering of very few parts to an easier question of split-state tampering with a much larger number of parts. Our first result shows a non-malleable reduction from 5-split-state tampering to t-split-state tampering.

Theorem 18 (Independence amplification from 5 parts). $(\mathcal{S}^5_{6t^2n} \cup \mathcal{FOR}^5_{6t^2n} \Rightarrow \mathcal{S}^t_n, 2^{-\Omega(tn)})$.

In our second result, we show a non-malleable reduction from 2-split-state tampering to the family containing t-split-state tampering and the t-part forgetful family.

Theorem 19 (Independence amplification from 2 parts). $(\mathcal{S}^2_{O(t^4n)} \Rightarrow \mathcal{S}^t_n \cup \mathcal{FOR}^t_n, 2^{-\Omega(n)})$.

Theorem 18 will be proved in Section 5 and Theorem 19 will be proved in Section 6.

Application to Non-malleable Codes. We can compose the reduction in Theorem 18 with the already known constructions of non-malleable codes in the independent-bit tampering model (i.e. for tampering families $\mathcal{S}_1^k$), summarized below [DPW10, CG14b, FMVW14, **?**]:

Theorem 20 (NM code for bit tampering [CG14b]). $(\mathcal{S}_1^{1.1k} \Rightarrow \mathsf{NM}_k, 2^{-\Omega(k)})$.

Using Theorem 15, and Theorem 18 with $t = k$, and $n = 1$, we get the following result:

Theorem 21 (5-split NM code with rate $O(k)$). *There exists $n = O(k^2)$, such that $(\mathcal{S}_n^5 \cup \mathcal{FOR}_n^5 \Rightarrow \mathsf{NM}_k, 2^{-\Omega(k)})$. In particular, there exists an efficient $(\mathcal{S}_{O(k^2)}^5, k, 2^{-\Omega(k)})$-non-malleable code.*

We can compose this with Theorem 19, to reduce the number of parts from 5 to 2 by increasing the length of the codewords by a constant factor.

Theorem 22 (2-split NM code with rate $O(k)$). $(\mathcal{S}_{O(k^2)}^2 \Rightarrow \mathsf{NM}_k, 2^{-\Omega(k)})$. *Namely, there exists an efficient $(\mathcal{S}_{O(k^2)}^2, k, 2^{-\Omega(k)})$-non-malleable code.*

This result already dramatically improves upon the previous best-known $(\mathcal{S}_{O(k^7)}^2, k, 2^{-\Omega(k^{1/7})})$-non-malleable code of [ADL14, Agg14]. However, we can further improve our non-malleable code using a recent work of Chattopadhyay and Zuckerman [CZ14]. [CZ14] obtained a construction of non-malleable codes with constant rate in the 9-split-state model. Their construction was achieved using a connection of t-source non-malleable extractors to non-malleable codes in the t-split-state model shown in [CG14b]. We observe that if the extractor is also a strong extractor (which is the case for [CZ14]), then the corresponding code is also non-malleable against the forgetful family. The details can be found in Appendix C, but they imply the following result:

Theorem 23 (9-split NM code with rate $O(1)$). *There exist $n = O(k)$, such that $(\mathcal{S}_n^9 \cup \mathcal{FOR}_n^9 \Rightarrow \mathsf{NM}_k, 2^{-\Omega(k)})$.*

Combining this with our reduction given in Theorem 19, we get our main result.

Theorem 24 (Main result: 2-split NM code with rate $O(1)$). $(\mathcal{S}_{O(k)}^2 \Rightarrow \mathsf{NM}_k, 2^{-\Omega(k)})$. *Namely, there exists and efficient $(\mathcal{S}_{O(k)}^2, k, 2^{-\Omega(k)})$-non-malleable code.*

Remark 4. Clearly, from the asymptotic sense, Theorem 24 is superior to Theorem 22 (which is in turn superior to Theorem 21). However, the constant factors hidden inside the result of [CZ14] (i.e., Theorem 23) used to prove Theorem 24 are really large, as the rely on some existential results in additive combinatorics. Thus, in many concrete situations the code constructed in Theorem 22 (which was done independently from the work of [CZ14]) would be superior to the asymptotically better code obtained in Theorem 24. To a lesser extent, when increasing the number of independent parts from 2 to 5 is feasible, the simple 5-part code in Theorem 21 will likely be more efficient than the code in Theorem 22.

5 Non-Malleable Reduction from 5 parts to t parts

5.1 Our Construction

In this Section, we prove Theorem 18. We prove it by a sequence of simpler intermediate reductions/transformations (some of which could be of independent interest), and then applying the

composition theorem (Theorem 15). We now specify the intermediate steps, leaving the proofs of corresponding theorems to subsequent subsections.

Our first result is a transformation from 2-split-state model to the "t-lootahead model". Namely, we gain in introducing many parts, at the expense of dealing with more challenging tampering functions on each part (as compared to the split-state model).

Theorem 25 (2-split to lookahead). $(\mathcal{S}^2_{3tn} \to \mathcal{LA}^{\leftarrow t}_n, t^2 \cdot 2^{-\Omega(n)})$.

The proof is given in Section 5.2, but here we briefly sketch the definition of the required transformation T_1 (see Section 5.2 for more details). It is based on the alternating extraction protocol [DP07b] depicted in Figure 1. The first memory part stores random strings Q, R_0, the second memory part stores random W (where $|Q| = |W| \approx 2tn$ and $|R_0| = n$), and we let

$$T_1((Q, R_0), W) = (R_1, \ldots, R_t), \tag{3}$$

where each R_i is iteratively defined by using R_{i-1} to extract S_{i-1} from W, and then S_{i-1} to extract R_i from Q.

Next, we show how to transform two independent t-lookahead tampering families to the t-split-state tampering family (and, for future use, the forgetful family).

Theorem 26 (2-lookahead to t-split). *If $n \geq 2tm$, then*

$$(((\mathcal{LA}^{\leftarrow t}_n \times \mathcal{LA}^{\leftarrow t}_n) \cup \mathcal{FOR}^{2t}_n) \to (\mathcal{S}^t_m \cup \mathcal{FOR}^t_m), O(2^{-\Omega(n)})) \ .$$

The proof is given in Section 5.3, but here we only mention the definition of the transformation T_2 we construct. Let $\mathsf{Ext_2}$ be any $(n, n/2, n, m, 2^{-(n-2m-2)/4})$-extractor (e.g., the inner product), and define

$$T_2(L, R) := (\mathsf{Ext_2}(L_t, R_1), \mathsf{Ext_2}(L_{t-1}, R_2), \ldots, \mathsf{Ext_2}(L_1, R_t)), \tag{4}$$

where $L = (L_1 || \ldots || L_t), R = (R_1 || \ldots || R_t)$, $L_i, R_i \in \{0,1\}^n$.

As a direct corollary of Theorems 25 and 26, we get a transformation from 4-split-state model to t-split state model:

Theorem 27. $(\mathcal{S}^4_{6t^2n} \to \mathcal{S}^t_n, 2^{-\Omega(tn)})$.

Now, it is tempting to use Theorem 16 to get a non-malleable reduction from $\mathcal{S}^4_{6t^2n}$ to $\mathcal{S}^t_n$. Unfortunately, we do not know how to turn the non-malleable *transformation* in Theorem 27 into a *reduction* (i.e., how to efficiently invert T in Theorem 27, and then apply Theorem 16). Instead, we observe the following very general result allowing us to translate a non-malleable transformation from *any* $\mathcal{F}$ to t-split tampering $\mathcal{S}^t_n$, into a non-malleable reduction from $\mathcal{F} \times \mathcal{S}_n$ to $\mathcal{S}^t_n$. Namely, in the t-split model, we go from transformation to reduction at the expense of another "split-state part" $\mathcal{S}_n$.

Theorem 28. *If $(\mathcal{F} \to \mathcal{S}^t_n, \varepsilon)$, then $(\mathcal{F} \times \mathcal{S}_n \Rightarrow \mathcal{S}^t_n, 2\varepsilon)$.*
In particular, using the transformation in Theorem 27, we get

$$(\mathcal{S}^5_{6t^2n} \Rightarrow \mathcal{S}^t_n, 2^{-\Omega(tn)}) \ .$$

The proof is given in Section 5.4, but we briefly mention the reduction (E, D) as a function of the transformation T. To encode a value $x \in \{0,1\}^{tn}$, we pick a random y in the domain of $\mathcal{F}$, and let $x^* = T(y)$, $d = x \oplus x^*$, and output (y, d) (where d is stored in the extra tn-bit part). To decode (y, d), we compute $D(y, d) = T(y) \oplus d$.

Of course, by using "dummy" bits to extend the 5-th part from tn bits to $O(6t^2n)$ bits, we get a reduction from 5-split-state model to t-split state model. Also, to prove Theorem 18, we additionally need to argue that our final encoding scheme is can also handle the forgetful family $\mathcal{FOR}^5_{6t^2n}$. We sketch the proofs in Section 5.5.

5.2 From two split-state parts to lookahead (proof of Theorem 25)

We first recall the alternating extraction, which was introduced by Dziembowski and Pietrzak in [DP07b], and present a particular variant of the *alternating-extraction theorem* from Dodis and Wichs [DW09], which will be especially convenient for our purposes. We then show how to use this result to get our non-malleable transformation.

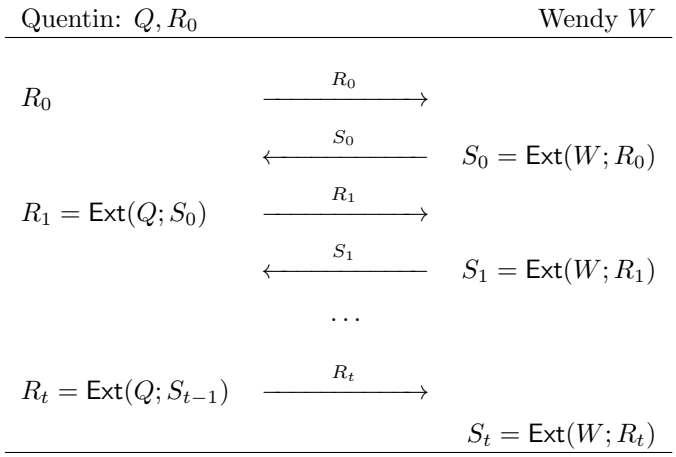

Figure 1: Alternating Extraction

Alternating Extraction. Assume that two parties, Quentin and Wendy, have uniformly random N-bit values Q and W, respectively, such that W is kept secret from Quentin and Q is kept secret from Wendy. Let Ext be the efficient $(N, 2n, n, n, 2^{-\Omega(n)})$-extractor given in Lemma 4 (where $n = \Omega(\log N)$), and assume that Quentin also has a random seed $R_0 \in \{0,1\}^n$ for the extractor Ext. The *alternating extraction protocol* (see Figure 1) is an interactive process between Quentin and Wendy, which runs in t iterations for some parameter t. In the first iteration, Quentin sends his seed R_0 to Wendy, Wendy computes $S_0 = \mathsf{Ext}(W; R_0)$, sends S_0 to Quentin, and Quentin computes $R_1 = \mathsf{Ext}(Q; S_0)$. In each subsequent iteration i, Quentin sends R_i to Wendy, who replies with $S_i = \mathsf{Ext}(W; R_i)$, and Quentin computes $R_{i+1} = \mathsf{Ext}(Q; S_i)$. Thus, Quentin and Wendy together produce the sequence:

$$R_0, S_0 = \mathsf{Ext}(W; R_0), R_1 = \mathsf{Ext}(Q; S_0), \ldots, R_t = \mathsf{Ext}(Q; S_{t-1}), S_t = \mathsf{Ext}(W; R_t) \tag{5}$$

The *alternating-extraction theorem* says that there is no better strategy that Quentin and Wendy can use to compute the above sequence. More precisely, for our purposes we will use the following version (slightly weaker than the most general version presented by [DW09]). Let us assume that, in each iteration, Quentin is limited to sending at most s bits to Wendy, who can then reply by sending at most s bits to Quentin, where s is much smaller than the entropy (i.e., length) N of Q and W (preventing Quentin from sending his entire value Q, and vice versa). Then, for any possible strategy cooperatively employed by Quentin and Wendy in the first i iterations, the value S_{i+1} (and also $S_{i+2}, \ldots, S_t$, but we won't use it) look uniformly random to Quentin (and, symmetrically,

R_{i+1}, and even $R_{i+2}, \ldots, R_t$ look random to Wendy). In other words, Quentin and Wendy acting together cannot speed up the process in some clever way, so that Quentin would learn S_{i+1} (or even distinguish it from random) in fewer than $i+1$ iterations.

More formally, the following variant of alternating extraction Theorem is a special case of Lemma 41 from [DW09].

Theorem 29 (Alternating Extraction; [DP07b, DW09]). *For any integers N, n, s, t, where $N \geq st + 2n$ and $n = \Omega(\log N)$, let W, Q be two random and independent N-bit strings, and* Ext *be an efficient $(N, 2n, n, n, \varepsilon = 2^{-\Omega(n)})$-extractor (which exists by Lemma 4). Let R_0 be uniformly random on $\{0,1\}^n$ and define $S_0, R_1, S_1, \ldots, R_t, S_t$ as in equation (5). Let $\mathcal{A}_q(Q, R_0), \mathcal{A}_w(W)$ be interactive machines such that, in each iteration, $\mathcal{A}_q$ sends at most s bits to $\mathcal{A}_w$ which replies with at most s bits to $\mathcal{A}_q$. Let V_w^i, V_q^i denote the* views *of the machines $\mathcal{A}_w, \mathcal{A}_q$ respectively, including their inputs and transcripts of communication, after the first i iterations. Then, for all $0 \leq i \leq t$,*

$$\left(V_w^i, R_i\right) \approx_{2t\varepsilon} \left(V_w^i, U_n\right) \qquad \text{and} \qquad \left(V_q^i, S_i\right) \approx_{2t\varepsilon} \left(V_q^i, U_n\right) \tag{6}$$

Our Non-Malleable Transformation. In the proof, we will use the alternating extraction theorem with per round communication $s = 2n$, so that we can set $N = 2nt + 2n$. Our first part of memory will simply random $Q \in \{0,1\}^N$ and $R_0 \in \{0,1\}^n$, and the second part will store a random $W \in \{0,1\}^N$, so that the size of each memory piece is at most $N + n \leq 3tn$ (which is what is claimed in Theorem 25), and our non-malleable transformation $T_1((Q, R_0), W)$ will simply output t strings $(R_1, \ldots, R_t)$, as defined in the alternating extraction protocol.

Now, let us fix arbitrary tampering functions $f_q(Q, R_0) = (Q', R_0')$ and $f_w(W) = W'$ on the two memory parts, and let $(R_1', \ldots, R_t')$ denote the output of an (honest) execution of the alternating extraction protocol on inputs (Q', R_0') and W'. To complete the proof, it suffices to show the validity of Equation (2). Namely, for given f_q and f_w, we need to exhibit a distribution G over "lookahead functions" $g(P_1, \ldots, P_t) = g_1(P_1) || g_2(P_1, P_2) || \ldots || g_t(P_1, \ldots, P_t)$ such that

$$\Delta\Big((R_1, R_1', \ldots, R_t, R_t') \; ; \; (P_1, P_1', \ldots, P_t, P_t')\Big) \leq t^2 \cdot 2^{-\Omega(n)}, \tag{7}$$

where each $P_i \equiv U_n$ (uniform n-bit string) and $(P_1', \ldots, P_t') = G(P_1, \ldots, P_t)$.

We describe the distribution G as a stateful probabilistic algorithm which, given P_1, produces P_1', then additionally given P_2, produces P_2', etc., which is equivalent to the lookahead restriction above. Formally, for any $1 \leq i \leq t$, given particular values $P_1 = s_1, P_1' = s_1', \ldots, P_{i-1} = s_{i-1}, P_{i-1}' = s_{i-1}, P_i = s_i$, it samples (using fresh independent coins) the value s_i' from the "real" conditional distribution

$$R_i' \mid (R_1 = s_1, R_1' = s_1', \ldots, R_{i-1} = s_{i-1}, R_{i-1}' = s_{i-1}', R_i = s_i),$$

and outputs $P_i' = s_i'$. Namely, G views its inputs s_j, which are actually sampled uniformly at random from U_n, as if coming from the "correct distribution" of running the alternating extraction protocol on random Q, R_0, W. And then G samples the tampered value s_i' under this (incorrect) assumption.

To argue Equation (7), we use the hybrid argument and define $t+1$ intermediate distributions $D_0, \ldots, D_t$, where $D_0 = (R_1, R_1', \ldots, R_t, R_t')$, $D_t = (P_1, P_1', \ldots, P_t, P_t')$, while the intermediate distribution D_i is defined as follows. For the first i steps, it honestly samples values (R_i, R_i') from the left distribution, while for the last $(t-i)$ steps it takes the partial history $(s_1, s_1', \ldots, s_{j-1}, s_{j-1}')$ so far, picks *uniformly random* $s_j \leftarrow U_n$, and samples s_j' from the conditional distribution described above (using fresh coins every time). We can indeed see that $D_0 = (R_1, R_1', \ldots, R_t, R_t')$, $D_t =$

$(P_1, P'_1, \ldots, P_t, P'_t)$, so it suffices to show that $\Delta(D_i; D_{i+1}) \leq 2t\varepsilon$, where $\varepsilon = 2^{-\Omega(n)}$ is the same as in Theorem 29.

Fortunately, this immediately follows from the alternating extraction theorem above, using the following machines $\mathcal{A}_q(Q, R_0), \mathcal{A}_w(W)$. $\mathcal{A}_q(Q, R_0)$ computes $(Q', R'_0) = f_q(Q, R_0)$, and runs two honest executions of the alternating extraction protocol with real input (Q, R_0) and tampered input (Q', R'_0), and $\mathcal{A}_w(W)$ does the same thing on its side. This indeed gives communication bound $s = 2n$ per round, and also results in a view V_w^i which includes precisely $(R_1, R'_1, \ldots, R_i, R'_i)$. Hence, using Equation (6), we get

$$\left(R_1, R'_1, \ldots, R_i, R'_i, R_{i+1}\right) \approx_{2t\varepsilon} \left(R_1, R'_1, \ldots, R_i, R'_i, U_n\right),$$

which is precisely the $(i+1)$-prefixes of our distributions D_i and D_{i+1}. But since the $(t-i-1)$-suffixes are sampled in the same way for both distributions, applying Lemma 2 yields $\Delta(D_i; D_{i+1}) \leq 2t\varepsilon$, completing the proof of Theorem 25.

5.3 From two look-ahead parts to t-split (proof of Theorem 26)

In this section, we show that if $n \geq 2tm$, then $(\mathcal{LA}_n^{\leftarrow t} \times \mathcal{LA}_n^{\leftarrow t} \cup \mathcal{FOR}_n^{2t} \to \mathcal{S}_m^t \cup \mathcal{FOR}_m^t, t2^{-\frac{n-2m-2}{4}})$.

Notation for the proof. Let $L_i, R_i \in \{0,1\}^n$ be random vectors, and let $\mathsf{Ext_2}(L_i, R_i) = b_i \in \{0,1\}^m$. Define:

$$L := (L_t, L_{t-1}, \ldots L_1) \qquad\qquad R := (R_1, R_2, \ldots R_t)$$

Consider the output after applying manipulation function from $\mathcal{LA}_n^{\leftarrow t} \times \mathcal{LA}_n^{\leftarrow t}$ to L and R. (Notice, we will generally use primed letters for manipulated values.) It can be described as:

$$L'_i := f_i(L_i, L_{i+1}, \ldots, L_t) \qquad\qquad R'_i := g_i(R_1, R_2, \ldots, R_i)$$

for some set of functions (called *manipulation family*) $f_1, g_1, f_2, g_2, \ldots f_t, g_t$. Notice, while R'_i depends on $R_1, \ldots, R_i$, the value L'_i depends on $L_i, \ldots, L_t$, since we "reversed" the vector L above, so that the manipulation function from $\mathcal{LA}_n^{\leftarrow t}$ reads the values of L "backwards".

Similar notation will be used for decoding of manipulated input bitstrings:

$$b'_i := \mathsf{Ext_2}(L'_i, R'_i).$$

We need to prove that

$$(b_1, b'_1, \ldots, b_t, b'_t) \approx_{t2^{-n/4}} (U^{(1)}, h_1(U^{(1)}, Z_1) \ldots, U^{(t)}, h_t(U^{(t)}, Z_t)), \tag{8}$$

where $U^{(1)}, \ldots, U^{(t)}$ denote independent random elements in $\{0,1\}^m$, and $Z = (Z_1, \ldots, Z_t)$ is some random variable independent of $U^{(1)}, \ldots, U^{(t)}$. Note that $Z_1, \ldots, Z_t$ might have dependence amongst themselves. In order to prove this, we need to define $Z_1, \ldots, Z_t$, which we do below.

Definition of Z. We define random variables $Z_1, \ldots, Z_t$ iteratively.

Define $\mathsf{Var_1}$ to be the joint random variable $(L_2, \ldots, L_t)$. Let Y_1 be a fresh random variable that samples the distribution L_1 given the values of $\mathsf{Var_1}$, R_1 and $\mathsf{Ext_2}(L_1, R_1)$. Thus, conditioned on $R_1, \mathsf{Ext_2}(L_1, R_1), \mathsf{Var_1}$, we have

$$L_1 \equiv \phi_1(\mathsf{Var_1}, R_1, Y_1, \mathsf{Ext_2}(L_1, R_1)),$$

for some function ϕ_1. Define $Z_1 = (Z_{1,b} : b \in \{0,1\}^m)$ indexed by b to be a $m \cdot 2^m$-bit random variable as a function of R_1, Var_1, Y_1 as follows:

$$Z_{1,b} := \mathsf{Ext}_2\left(f_1(\phi_1(\mathsf{Var}_1, R_1, Y_1, b), L_2, \ldots, L_t), g_1(R_1)\right) .$$

Note that $b'_1 = Z_{1,b_1}$ is a deterministic function of Z_1 and b_1. Let $b'_1 = h_1(b_1, Z_1)$.

Now, given $Z_1, \ldots, Z_{i-1}$, we define Y_i, Z_i. Define Var_i to be the joint random variable

$$\mathsf{Var}_i = Z_1, \ldots, Z_{i-1}, R_1, \ldots, R_{i-1}, L_{i+1}, \ldots, L_t .$$

Let Y_i be a fresh random variable that samples the distribution L_i given the values of Var_i, R_i, and $\mathsf{Ext}_2(L_i, R_i)$. Thus, conditioned on $R_i, \mathsf{Ext}_2(L_i, R_i), \mathsf{Var}_i$, we have

$$L_i \equiv \phi_i(\mathsf{Var}_i, R_i, Y_i, \mathsf{Ext}_2(L_i, R_i)) ,$$

for some function ϕ_i. Define $Z_i = (Z_{i,b} : b \in \{0,1\}^m)$ indexed by b to be a $m \cdot 2^m$-bit random variable as a function of R_i, Var_i, Y_i as follows:

$$Z_{i,b} := \mathsf{Ext}_2\left(f_i(\phi_i(\mathsf{Var}_i, R_i, Y_i, b), L_{i+1}, \ldots, L_t), g_i(R_1, \ldots, R_i)\right) .$$

Note that $b'_i = Z_{i,b_i}$ is a deterministic function of Z_i and b_i. Let $b'_i = h_i(b_i, Z_i)$.

Proof of Theorem. We prove Equation (8) using a hybrid argument. In particular, we show that for all $i = 1, \ldots, t$,

$$\begin{aligned}
&(U^{(1)}, h_1(U^{(1)}, Z_1), \ldots, U^{(i-1)}, h_{i-1}(U^{(i-1)}, Z_{i-1}), \quad b_i, \quad b'_i, \quad b_{i+1}, b'_{i+1}, \ldots, b_t, b'_t) \\
\approx_{2^{-(n-2m-2)/4}} &(U^{(1)}, h_1(U^{(1)}, Z_1), \ldots, U^{(i-1)}, \ h_{i-1}(U^{(i-1)}, Z_{i-1}), \ U^{(i)}, \ h_i(U^{(i)}, Z_i), \ b_{i+1}, b'_{i+1}, \ldots, b_t, b'_t) .
\end{aligned} \tag{9}$$

Equation (8) then follows from (9) by triangle inequality.

To prove Equation (9), consider the following. Since the total length of $(h_1(U^{(1)}, Z_1), \ \ldots, \ h_{i-1}(U^{(i-1)}, Z_{i-1}))$ is $m(i-1) \leq tm \leq \frac{n}{2}$, we have that $\mathbf{H}_\infty(L_i|\mathsf{Var}_i) = \mathbf{H}_\infty(L_i|Z_1, \ldots, Z_{i-1}) \geq \frac{n}{2}$. Also notice that R_i *is independent of* Var_i. Indeed, tracing the definition of $Z_1, \ldots, Z_{i-1}$ and using the "lookahead" property of R, we see that functions $g_1, \ldots, g_{i-1}$ were only applied to $R_1, \ldots, R_{i-1}$ when defining values $Z_1, \ldots, Z_{i-1}$. Thus, using the independence between the seed R_i and the source $L_i|\mathsf{Var}_i$, and also throwing completely fresh and independent values $U^{(1)}, \ldots, U^{(i-1)}, Y_i, R_{i+1}, \ldots, R_t$ into the mix, we can apply Lemma 5 and get

$$\begin{aligned}
&(Z_1, \ldots, Z_{i-1}, U^{(1)}, \ldots, U^{(i-1)}, \quad \mathsf{Ext}_2(L_i, R_i), R_1, \ldots, R_i, Y_i, R_{i+1}, \ldots, R_t, L_{i+1}, \ldots, L_t) \\
\approx_{2^{-n/4}} &(Z_1, \ldots, Z_{i-1}, U^{(1)}, \ldots, U^{(i-1)}, \qquad U^{(i)}, \quad R_1, \ldots, R_i, Y_i, R_{i+1}, \ldots, R_t, L_{i+1}, \ldots, L_t) \quad .
\end{aligned} \tag{10}$$

We now claim that Equation (9) directly follows from Equation (10) by applying Lemma 2. To see this, we notice that, by the "lookahead"property of L, the values $b_{i+1}, b'_{i+1}, \ldots, b_t, b'_t$ are deterministic functions of $R_1, \ldots, R_t, L_{i+1}, \ldots, L_t$ (in particular, they do not depend on L_i). Also, the value Z_i is deterministic function of $Y_i, Z_1, \ldots, Z_{i-1}, R_1, \ldots, R_i, L_{i+1}, \ldots, L_t$ (in particular, it also does not depend on the L_i). Finally, $b_i = \mathsf{Ext}_2(L_i, R_i)$ and $b'_i = h_i(b_i, Z_i)$, which means that (b_i, b'_i) (resp. $(U^{(i)}, h_i(U^{(i)}, Z_i))$) is also a deterministic function of $\mathsf{Ext}_2(L_i, R_i), Z_i$ (resp. $U^{(i)}, Z_i$). This concludes the proof of the fact that $(\mathcal{LA}_n^{\leftarrow t} \times \mathcal{LA}_n^{\leftarrow t} \to \mathcal{S}_m^t, t \cdot 2^{-(n-2m-2)/4})$, as all variables in

the left (resp. right) hand side of Equation (9) are deterministic functions of the same corresponding variables from left (resp. right) hand side of Equation (10).

Note that it is trivial to see that the reduction of this Section also gives

$$(\mathcal{FOR}_n^{2t} \to \mathcal{FOR}_m^t, 2^{-(n-m-1)/2}) .$$

This is because by the strong extractor property of Ext_2, for any i, $\mathsf{Ext}_2(L_i, R_i)$ is $2^{-\Omega(n)}$ close to uniform given

$$L_1, \ldots, L_{i-1}, L_{i+1}, \ldots, L_t, R_1, \ldots, R_{i-1}, R_{i+1}, \ldots, R_t ,$$

and one of L_i, R_i.

5.4 From transformation to reduction for t-split state tampering (proof of Theorem 28)

Let $\mathcal{F} \subset A^A$ and T be a function given by definition of non-malleable transformation $(\mathcal{F} \to \mathcal{S}_n^t, \varepsilon)$. We start with definitions of encoding and decoding functions that we claim satisfy the definition of a non-malleable reduction. Let $E : \{0,1\}^{tn} \to A \times \{0,1\}^{tn}$: be a random experiment defined as follows:

$$E(x) := \left\{ y \xleftarrow{\$} A \ ; \ x^* := T(y) \ ; \ d := x \oplus x^* \ ; \ \text{Output } (y, d) \right\}$$

and corresponding decoding:

$$D(y, d) := T(y) \oplus d.$$

It is obvious that both encoding and decoding are efficient and that $D(E(x)) = x$ for all $x \in \{0,1\}^{tn}$.

Proof of reduction property. We need also to prove Equation (1). Let us fix x, and tampering functions $f \in \mathcal{F}$ and $g \in \mathcal{S}_n$. We denote the tampered values with primed letters. Since $T(y) = x^*$ and $\mathcal{F}$ transforms to $\mathcal{S}_n^t$, Equation (2) implies that with probability at least $(1 - \varepsilon)$ all the values $(x_i^*)' = f_i(x_i^*)$ (for some functions $(f_1, \ldots, f_t)$ distributed over $\mathcal{S}_n^t$). Also let $d' := g(d)$ be the tampered value of the last part. Then, ignoring the ε-failure event above (for which we will pay ε in the statistical distance), we have:

$$x_i' = d_i' \oplus (x_i^*)' = g(d)_i \oplus f_i(x_i^*) = g(d)_i \oplus f_i(x_i \oplus d_i) = h_i(x_i, d)$$

for some function h_i. To complete the argument, it remains to argue that d is "ε-independent" from x (i.e., $(x, d) \approx_\varepsilon (x, U_{tn})$), meaning that h_i is ε-close to a valid independent tampering function. However, this follows again from Equation (2), since $d = x \oplus x^*$, and x^* is ε-close to U_{tn} (even given x, since $y \leftarrow A$ is random and independent from x).

Thus, x' is indeed 2ε-close to a convex combination of functions from $\mathcal{S}_n^t$ applied to x.

5.5 Proof Sketch for Forgetful Property

Note that to prove Theorem 18, we additionally need to argue that our scheme (E, D) from this section gives

$$(\mathcal{FOR}_{6t^2n}^5 \Rightarrow \mathcal{S}_n^t, 2^{-\Omega(tn)}) .$$

Let the five parts encoding x be $(Q, R_0), W, (P, L_0), V, d$ such that $T_1((Q, R_0), W) = (R_1, \ldots, R_t)$, $T_1((P, L_0), V) = (L_1, \ldots, L_t)$, and $x = d \oplus T_2(L, R)$. We observe that a stronger 5-out-of-5 secret sharing property holds, i.e., that any 4 of the 5-parts, it is impossible to guess x except with probability $2^{-\Omega(tn)}$.

This is obvious if we are given the first four parts but "forget" the fifth part. We sketch here why this is sufficient even if we forget one of the other four parts. Without loss of generality, assume that we are given $(P, L_0), V$ and one of (Q, R_0) or W. Thus, $L_1, \ldots, L_t$ is completely known, but we show that conditioned on this information, $R_1, \ldots, R_t$ is like a block-source, i.e. for all $i \in [t]$, if we are additionally given $R_1, \ldots, R_{i-1}$, then the string R_i still has sufficient min-entropy. This implies that $\mathsf{Ext}_2(L_i, R_i)$ is close to uniform given $\mathsf{Ext}_2(L_1, R_1), \ldots, \mathsf{Ext}_2(L_{i-1}, R_{i-1})$, and hence the required result follows by a hybrid argument. To argue that $R_1, \ldots, R_t$ is indeed a block-source we proceed as follows.

CASE 1: We are given W. In this case, a stronger condition holds, i.e. that $R_1, \ldots, R_t$ is close to uniform. This follows from Theorem 29.

CASE 2: We are given (Q, R_0) Assume that $R_1, \ldots, R_t$ is not a block-source. In this case, R_i has small min-entropy given $R_1, \ldots, R_{i-1}$. This implies that at the end of $i-1$ rounds of the alternating extraction protocol, Quentin can guess R_i with high probability. Now assume that $\mathcal{A}_w$ is honest, and $\mathcal{A}_q$ honestly follows the protocol for the first $i-1$ rounds, and sends $R_0, \ldots, R_{i-2}$, but then in the i-th round, guesses and sends R_i. This contradicts the fact that $(V_w^i, R_i) \approx_{2t\varepsilon} (V_w^i, U_n)$.

6 Non-Malleable Reduction from 2 parts to t parts

In this Section, we prove Theorem 19. We observe that Theorem 19 immediately follows by applying the composition Theorem 15 to Theorem 26 and the following result:

Theorem 30. $(\mathcal{S}^2_{10t(t^2+t+1)(n+3ts)} \Rightarrow (\mathcal{LA}_n^{\leftarrow t/2} \times \mathcal{LA}_n^{\leftarrow t/2}) \cup \mathcal{FOR}_n^t, O(t^4 \cdot 2^{-s}))$.

This result (whose proof will take the remainder of this section, and which is by far the most complicated individual reduction that we construct) can be seen as strengthening of Theorem 25. Namely, while Theorem 25 reduced 2-split-state tampering to a single lookahead tampering, in our new reduction we manage to reduce it to two *independent* lookahead tamperings.[2]

To prove Theorem 30, we need to define encoding and decoding functions (see Definition 11) and prove that they satisfy the required conditions for non-malleable reductions. Correspondingly, in Section 6.1 we will first define our efficient reduction, and then prove its security in Sections 6.2-6.4 (namely, first state the high-level proof structure, then define the intermediate "partition objects" we need, and finally prove the low-level technical lemmas about these "partition objects").

6.1 Construction

Now, we will define an encoding from $\{0,1\}^{nt}$ to $\{0,1\}^{t\alpha(n+3ts)} \times \{0,1\}^{t\alpha(n+3ts)}$ for $\alpha = 10t^2 + 10t + 10$.[3] For brevity we will consider $\mathcal{L} = \mathcal{R} = \{0,1\}^{t\alpha(n+3ts)}$ and write $E : (\{0,1\}^n)^t \mapsto \mathcal{L} \times \mathcal{R}$ for encoding.

For any integer i, let $\mathsf{Ext} : \{0,1\}^{i(n+3ts)} \times \{0,1\}^{i(n+3ts)} \to \{0,1\}^{(n+3ts)}$ be the inner-product extractor, which is an $\left(i(n+3ts), (i+1)(n+3ts) + 2\log\left(\frac{1}{\varepsilon}\right), n+3ts, \varepsilon\right)$-two-source extractor. We slightly abuse notation here, and any element in $\{0,1\}^{(n+3ts)i}$ for any integer i should be considered

[2] It is this strengthening which will eventually allow us to construct 2-part non-malleable codes as opposed to 5-part non-malleable codes.

[3] The bound is somewhat loose, since our goal is to only achieve constant rate, we do not try to optimize various parameters including α. See Section 7 for more details.

as the corresponding element (w.r.t. any bijective mapping) to an element in $\mathbb{F}^i_{2^{n+3ts}}$, whenever we take inner products.

For $i \in \{1, 2, ..., t\}$ let $h_i : \{0,1\}^{(n+3ts)} \mapsto \{0,1\}^n \cup \{\perp\}$ be defined as $h_i(x) = (x)_n$ if x is the binary expansion of an integer less than $2^{n+3(i-1)s}$, and $h_i(x) = \perp$, otherwise. (Where $(.)_\lambda$ denotes truncation to λ least significant bits.) Using this, our encoding and decoding functions are defined as follows.

Definition 31. *For any $\ell \in \mathcal{L}$, let $\ell = \ell_1 \| \ldots \| \ell_t$, where $\ell_i \in \{0,1\}^{\alpha(n+3ts)}$ for $1 \leq i \leq t$. Similarly, define $r = r_1 \| \ldots \| r_t$. Then the decoding function $D : \mathcal{L} \times \mathcal{R} \mapsto \{0,1\}^n \cup \{\perp\}$ is defined as*

$$D(\ell, r) := \begin{cases} \perp & \text{if } \exists i \in [t],\ h_i(\mathsf{Ext}(\ell_i, r_i)) = \perp \\ h_1(\mathsf{Ext}(\ell_1, r_1)) \| \ldots \| h_t(\mathsf{Ext}(\ell_t, r_t)) & \text{otherwise} \quad . \end{cases}$$

Definition 32. *The encoding function $E : (\{0,1\}^n)^t \mapsto \mathcal{L} \times \mathcal{R}$, on input $x \in (\{0,1\}^n)^t$, is naturally defined as the output of the following sampling procedure.*

1. *Choose uniformly random (L, R) such that $D(L, R) = x$.*
2. *Return (L, R).*

Our construction uses some ideas from a recent result [ADKO14], which showed a reduction from 2-split non-malleable codes with leakage to 2-split non-malleable codes (with the possibility that the two tampered parts are swapped). Also, our proof uses a similar framework. In particular, the partitioning procedure is similar. The crucial difference is that the proof of [ADKO14] was tailor-made to work for a reduction from 2 parts to 2 parts, and does not generalize easily. We crucially needed to introduce the intermediate two-lookahead family and then compose it with our reduction in Theorem 26 in order to generalize it to a reduction from 2 parts to t parts and hence conclude our result.

We would like to mention here that although we manage to prove a weaker result (which is sufficient for our purpose), we believe that our reduction from Theorem 30 is actually a reduction from $\mathcal{S}^2_{10t(t^2+t+1)(n+3ts)}$ to $\mathcal{S}^t_n \cup \mathcal{FOR}^t_n$. If one manages to prove this, then this immediately implies Theorem 19 without having to compose it with the reduction in Theorem 26. In addition to simplifying the proof, this will result in saving a constant factor in the overall code rate.

6.2 Proof Structure of Theorem 30

From the definition, it is obvious that for all $x \in (\{0,1\}^n)^t$, we have that $\Pr(D(E(x)) = x) = 1$. So, to prove theorem 30, we need to prove that for all $f \in \mathcal{L}^{\mathcal{L}}$, $g \in \mathcal{R}^{\mathcal{R}}$, and for all $x \in (\{0,1\}^n)^t$, there exists a random function P distributed over $(\mathcal{LA}_n^{\leftarrow t/2} \times \mathcal{LA}_n^{\leftarrow t/2}) \cup \mathcal{FOR}^t_n$ such that

$$\Delta(D(f(L), g(R))\ ;\ P(x)) = O(t^4 \cdot 2^{-s})\ , \tag{11}$$

where $E(x) = (L, R)$.

For the rest of this section, we fix the following notation. Let $x \in (\{0,1\}^n)^t$, and $f \in \mathcal{L}^{\mathcal{L}}$, $g \in \mathcal{R}^{\mathcal{R}}$. Let $(L, R) = E(x)$. For $\ell \in \mathcal{L}$, we write $f(\ell)$ as $f(\ell) = f_1(\ell) \| \ldots \| f_t(\ell)$, where $f_i(\ell) \in \{0,1\}^{\alpha(n+3ts)}$. We use similar notation for the parts of $g(r)$. Similarly, we write $L = L_1 \| \ldots \| L_t$. Also, we use similar notation for R.

The following simple lemma shows that it suffices to prove (11) for partitions of the ambient space. A similar idea was used both in [DKO13], and in [ADL14].

Lemma 33. *Let $f \in \mathcal{L}^{\mathcal{L}}$ and let $\mathcal{S} \subseteq \mathcal{L} \times \mathcal{R}$. Let $\mathcal{S}_1, \ldots, \mathcal{S}_j$ be a partition of $\mathcal{S}$. Also, let $P_1, \ldots, P_j$ be some random functions distributed over $(\mathcal{LA}_n^{\leftarrow t/2} \times \mathcal{LA}_n^{\leftarrow t/2}) \cup \mathcal{FOR}_n^t$. Assume that for all $1 \leq i \leq j$,*

$$\Delta\left(D(f(L), g(R))|_{(L,R)\in\mathcal{S}_i} \; ; \; P_i(x)\right) \leq \varepsilon_i.$$

Then

$$\Delta\left(D(f(L), g(R))|_{(L,R)\in\mathcal{S}} \; ; \; P(x)\right) \leq \sum \varepsilon_i \frac{|\mathcal{S}_i|}{|\mathcal{S}|} \; ,$$

for some P distributed over $(\mathcal{LA}_n^{\leftarrow t/2} \times \mathcal{LA}_n^{\leftarrow t/2}) \cup \mathcal{FOR}_n^t$.

Proof. The Lemma follows immediately from the definitions. Let $p_i = |\mathcal{S}_i|/|\mathcal{S}|$ denote the probability that $(L, R) \in \mathcal{S}_i$ conditioned on $(L, R) \in \mathcal{S}$. Then $D(f(L), g(R))$ is $(\sum p_i\varepsilon_i)$-close in statistical distance to $P(x)$ where P is a random function which is a convex combination of $P_1, \ldots, P_j$, that chooses P_i for all $i \in [j]$ with probability p_i. □

The main idea is to use Lemma 33 for a specific partition of $\mathcal{L} \times \mathcal{R}$. In fact, we give a partition for $\mathcal{L}$ and independently a partition for $\mathcal{R}$. Then the final partition of $\mathcal{L} \times \mathcal{R}$ is a Cartesian product of these two partitions.

More precisely, we partition the set $\mathcal{L}$ into the following $\frac{t^2+3t+2}{2}$ sets for $i, j \in [t]$

$$\mathcal{L}_{\mathsf{ffb},i}, \; \mathcal{L}_{\mathsf{mix},i}, \; \mathcal{L}_{\mathsf{id}}, \; \mathcal{L}_{\mathsf{perm},i\leftarrow j}, \; \mathcal{L}_{\mathsf{rem}} \; .^4$$

Similarly, we partition $\mathcal{R}$ (specific definitions are gathered in Section 6.3). Thus, the total number of parts of $\mathcal{L}\times\mathcal{R}$ we consider are $O(t^4)$. From Lemma 36, 37, 38, 39, 40, 41, 42, 43, 44 (see Section 6.4), we have that for each part $\mathcal{L}^* \times \mathcal{R}^*$ considered, either $\frac{|\mathcal{L}^*\times\mathcal{R}^*|}{|\mathcal{L}\times\mathcal{R}|} \leq 2^{-s}$, or $D(f(L), g(R))|_{(L,R)\in\mathcal{L}^*\times\mathcal{R}^*}$ is $O(2^{-s})$-close to $P^*(x)$ for some function P^* distributed over $(\mathcal{LA}_n^{\leftarrow t/2} \times \mathcal{LA}_n^{\leftarrow t/2}) \cup \mathcal{F}_n^t$. Thus, using Lemma 33 we have that

$$\Delta\left(D(f(L), g(R))|_{(L,R)\in\mathcal{S}} \; ; \; P(x)\right) \leq O(t^4) \cdot O(2^{-s}) \; ,$$

for some P distributed over $(\mathcal{LA}_n^{\leftarrow t/2} \times \mathcal{LA}_n^{\leftarrow t/2}) \cup \mathcal{FOR}_n^t$ which finishes the proof.

The partitioning procedure is quite extensive but then the proof for each partition will follow relatively easily from the two-source extractor property of Ext.

6.3 Partition

Now we define a partition of $\mathcal{L}$ based on f. Let $\beta_1 = 2t^2(n + 3ts) + 2ts + t$, and let $\beta_2 = (2t + 4)(n + 3ts) + 4s + 2$.

"Far from bijection" parts. First, we partition $\mathcal{L}$ into $\mathcal{L}_{\mathsf{ffb},1}, \ldots, \mathcal{L}_{\mathsf{ffb},t}$, and $\mathcal{L}_1$. We will define $\mathcal{L}_{\mathsf{ffb},1}, \ldots, \mathcal{L}_{\mathsf{ffb},t}$ inductively as follows. The set $\mathcal{L}_{\mathsf{ffb},i}$ is obtained by the following algorithm.

1. Initialize $\mathcal{L}_{\mathsf{ffb},i}$ to be empty, and $\mathcal{L}^* = \mathcal{L} \setminus \bigcup_{k=1}^{i-1} \mathcal{L}_{\mathsf{ffb},k}$.
2. Let $\mathcal{M}$ be a largest subset of $\mathcal{L}^*$ such that for any two $\ell, \ell' \in \mathcal{M}$, $\ell_i \neq \ell'_i$, and $f(\ell) = f(\ell')$.
3. If $|\mathcal{M}| \geq 2^{\beta_1/t}$, then set $\mathcal{L}^* = \mathcal{L}^* \setminus \mathcal{M}$, set $\mathcal{L}_{\mathsf{ffb},i} = \mathcal{L}_{\mathsf{ffb},i} \cup \mathcal{M}$, and go to step 2.

[4] Subscripts are abbreviations for intuitive meaning of the sets. Respectively: far from bijection, mixed, id, permuted and remaining.

4. Return $\mathcal{L}_{\mathsf{ffb},i}$.

The set $\mathcal{L}_1$ is defined to be

$$\mathcal{L}_1 = \mathcal{L} \setminus \bigcup_{i=1}^{t} \mathcal{L}_{\mathsf{ffb},i} \ .$$

and will be partitioned later.

The justification for this choice is that for $\tilde{L}$ chosen uniformly at random from $\mathcal{L}_{\mathsf{ffb},i}$, we have

$$\mathbf{H}_\infty(\tilde{L}_i | f(\tilde{L})) \geq \frac{\beta_1}{t} \ .$$

Also, we have that for any $y \in \{0,1\}^{\alpha(n+3ts)}$, the total number of $\ell \in \mathcal{L}_1$ such that $f(\ell) = y$ is at most $\left(2^{\frac{\beta_1}{t}}\right)^t = 2^{\beta_1}$.

One more definition. We further partition the set $\mathcal{L}_1$ depending on how different parts of $f(\ell)$ depend on different parts of ℓ for $\ell \in \mathcal{L}_1$. However, first we need one definition more:

Definition 34. *Define $T^{i \to j} \subset \mathcal{L}$ as the set of all $\ell \in \mathcal{L}$ such that*

$$\left| \left\{ \ell^* \in \mathcal{L} \;\middle|\; \ell_i = \ell^*_i \text{ and } f_j(\ell) = f_j(\ell^*) \right\} \right| \geq 2^{(t-1)\alpha(n+3ts)-\beta_2} \ .$$

We will define a partitioning of $\mathcal{L}_1$ using Definition 34, but before we do this, we prove the following simple result justifying the definition of $T^{i \to j}$. Intuitively, this result shows that if it is given that $\ell \in T^{i \to j}$, then $f_j(\ell)$ can be computed given ℓ_i and just a little more information.

Lemma 35. *Let $\ell \in T^{i \to j}$ for some $i, j \in [t]$. Then there exists some functions $a_{i,j} : T^{i \to j} \mapsto \{0,1\}^{\beta_2}$ and $b_{i,j} : \{0,1\}^{\alpha(n+3ts)} \times \{0,1\}^{\beta_2} \mapsto \{0,1\}^{\alpha(n+3ts)}$ such that for all $\ell \in T^{i \to j}$,*

$$f_j(\ell) = b_{i,j}(\ell_i, a_{i,j}(\ell)) \ .$$

Proof. Given $\ell \in T^{i \to j}$, let $T' = \{\ell^* \in T^{i \to j} \mid \ell^*_i = \ell_i\}$. Then, clearly $|T'| \leq 2^{(t-1)\alpha(n+3ts)}$.

Consider a partition of T' into sets $T'_1, \ldots, T'_m$ such that for any $u, v \in [m]$, and any $\ell' \in T'_u$, $\ell'' \in T'_v$, we have that $f_j(\ell') = f_j(\ell'')$ if and only if $u = v$. By definition of $T^{i \to j}$, we have that $|T'_u| \geq 2^{\beta_2}$ for all $u \in [m]$. Thus

$$m \leq \frac{|T'|}{2^{(t-1)\alpha(n+3ts)-\beta_2}} \leq 2^{\beta_2} \ .$$

We define $a_{i,j}(\ell)$ as the binary representation of k such that $\ell \in T'_k$. Now, it is easy to see that we can determine $f_j(\ell)$ given ℓ_i and $a_{i,j}(\ell)$.

□

"Mixed" parts. Now we define disjoint subsets $\mathcal{L}_{\mathsf{mix},1}, \ldots, \mathcal{L}_{\mathsf{mix},t}$ of $\mathcal{L}_1$ as follows.

$$\mathcal{L}_{\mathsf{mix},j} = \left\{ \ell \in \mathcal{L}_1 \setminus \bigcup_{k=1}^{j-1} \mathcal{L}_{\mathsf{mix},k} \;\middle|\; \ell \notin \bigcup_{i=1}^{t} T^{i \to j} \right\} \text{ for } j = 1, \ldots, t \ .$$

Informally speaking, $\ell \in \mathcal{L}_{\mathsf{mix},j}$ implies that $f_j(\ell)$ depends on more than one ℓ_i. Now, let

$$\mathcal{L}_2 = \mathcal{L}_1 \setminus \bigcup_{k=1}^{t} \mathcal{L}_{\mathsf{mix},k} \ .$$

"id", "permuted" and "remaining" parts. We denote $\mathcal{T}(\ell, i)$ to be the set of all $j \in [t]$ such that $\ell \in T^{i \to j}$. Note that if $\mathcal{A} \subset \mathcal{L}_2$ then by the definition of $\mathcal{L}_{\mathsf{mix},j}$, we have that for every $j \in [t]$ must belong to some $\mathcal{T}(\ell, i)$.

Let $\mathcal{B}_{i \leftarrow j}$ be set of permutations π of $[t]$ such that $\forall i' < i,\ \pi(i') = i'$ and $\pi(j) = i$. Also let $\prec$ denote standard lexicographic order. We further partition $\mathcal{L}_2$ into $\mathcal{L}_{\mathsf{Id}}$, and $\mathcal{L}_{\mathsf{perm},i \leftarrow j}$ for $i, j \in [t]$, and $\mathcal{L}_{\mathsf{rem}}$ as follows.

$$\mathcal{L}_{\mathsf{Id}} = \left\{ \ell \in \mathcal{L}_2 \;\middle|\; \forall i \in [t],\ \mathcal{T}(\ell, i) = \{i\} \right\}$$

and

$$\mathcal{L}_{\mathsf{perm},i \leftarrow j} = \left\{ \ell \in \mathcal{L}_2 \setminus (\mathcal{L}_{\mathsf{Id}} \cup \bigcup_{(i',j') \prec (i,j)} \mathcal{L}_{\mathsf{perm},i' \leftarrow j'}) \;\middle|\; \exists \pi \in \mathcal{B}_{i \leftarrow j},\ \forall i \in [t],\ \mathcal{T}(\ell, i) = \{\pi(i)\} \right\} \text{ for } j > i$$

and

$$\mathcal{L}_{\mathsf{rem}} = \mathcal{L}_2 \setminus (\mathcal{L}_{\mathsf{Id}} \cup \bigcup_{i,j} \mathcal{L}_{\mathsf{perm},i \leftarrow j})$$

Final partition. We similarly define the partitioning of $\mathcal{R}$ based on g. The final partition of $\mathcal{L} \times \mathcal{R}$ is a Cartesian product of these two partitions.

6.4 Proofs for lemmas for different cases

6.4.1 f or g is far from bijection

Lemma 36. *For all $i \in [t]$, for all $\mathcal{R}^* \subseteq \mathcal{R}$, if $|\mathcal{L}_{\mathsf{ffb},i} \times \mathcal{R}^*| \geq 2^{2\alpha t(n+3ts)-s}$ then there exists F distributed over $\mathcal{F}_n^t$, such that*

$$\Delta(D(f(L), g(R))|_{(L,R) \in \mathcal{L}_{\mathsf{ffb},i} \times \mathcal{R}^*}\ ; F(x)) \leq 2^{-s}\ .$$

Proof. Let $|\mathcal{L}_{\mathsf{ffb},i} \times \mathcal{R}^*| \geq 2^{2\alpha t(n+3ts)-s}$ for some $i \in [t]$. Let $\tilde{L}, \tilde{R}$ be distributed uniformly over $\mathcal{L}_{\mathsf{ffb},i}$ and $\mathcal{R}^*$ respectively. Note that by the assumption we have that $|\mathcal{L}_{\mathsf{ffb},i}| \geq 2^{\alpha t(n+3ts)-s}$ and $|\mathcal{R}^*| \geq 2^{\alpha t(n+3ts)-s}$. Thus, for all $k \in [t]$,

$$\mathbf{H}_\infty(\tilde{L}_k | \tilde{L}_1, \ldots, \tilde{L}_{k-1}) \geq \alpha(n + 3ts) - s\ , \tag{12}$$

and

$$\mathbf{H}_\infty(\tilde{R}_k | \tilde{R}_1, \ldots, \tilde{R}_{k-1}) \geq \alpha(n + 3ts) - s\ . \tag{13}$$

Denote $\mathsf{Ext}(\tilde{L}_k, \tilde{R}_k)$ by X_k for $k \in [t]$. Also, let $X_{-i} = X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_t$. Similarly define $\tilde{L}_{-i}$ and $\tilde{R}_{-i}$. We have that

$$\mathbf{H}_\infty(\tilde{L}_i | X_{-i}, f(\tilde{L})) \geq \frac{\beta_1}{t} - (t-1)(n + 3ts) = (t+1)(n + 3ts) + 2s + 1\ .$$

Also, using Lemma 9 by setting $A = \tilde{L}, B = \tilde{R}$, $V_1 = f(\tilde{L})$, $V_2 = \tilde{R}_{-i}$, $V_2 = X_{-i}$, we get that $\tilde{L}$ and $\tilde{R}$ (and hence $\tilde{L}_i$ and $\tilde{R}_i$) are independent given $f(\tilde{L})$, $\tilde{R}_{-i}$, and X_{-i}. Thus, using the fact that Ext is a strong two-source extractor, we have that

$$X_i, \tilde{R}_i, X_{-i}, f(\tilde{L}), \tilde{R}_{-i} \approx_{2^{-t(n+3ts)-(s+1)}} U_i, \tilde{R}_i, X_{-i}, f(\tilde{L}), \tilde{R}_{-i}\ ,$$

where U_i is uniformly random in $\{0,1\}^{\alpha(n+s)}$. This implies that

$$X_i, X_{-i}, D(f(\tilde{L}), g(\tilde{R})) \approx_{2^{-t(n+3ts)-(s+1)}} U_i, X_{-i}, D(f(\tilde{L}), g(\tilde{R}))\ . \tag{14}$$

Now, $D(f(\tilde{L}), g(\tilde{R}))$ can be seen as a randomized function of X_{-i}. Let

$$X_{-i}, D(f(\tilde{L}), g(\tilde{R})) \equiv X_{-i}, h(X_{-i}, Z) \ ,$$

where Z is an independent random variable and h is some function. Using Equation 12 and 13, and that Ext is a two-source randomness extractor, we have that X_k is $2^{-\frac{(\alpha-1)(n+3ts)-2s}{2}}$-close to uniform given $\tilde{L}_1, \ldots, \tilde{L}_{k-1}, \tilde{R}_1, \ldots, \tilde{R}_{k-1}$, and hence using the hybrid argument, we have that

$$X_{-i} \approx_{t \cdot 2^{-\frac{(\alpha-1)(n+3ts)-2s}{2}}} U_{-i} \ ,$$

where $U_{-i} = U_1, \ldots, U_{i-1}, U_{i+1}, \ldots, U_t$ for U_k, $k \in [t]$ being independent and uniformly distributed in $\{0,1\}^{\alpha(n+3ts)}$. Using Equation 14 and then applying Lemma 2, we get that

$$\begin{aligned} X_i, X_{-i}, D(f(\tilde{L}), g(\tilde{R})) \quad &\approx_{2^{-t(n+3ts)-(s+1)}} \quad X_i, U_{-i}, D(f(\tilde{L}), g(\tilde{R})) \ \equiv \ U_i, X_{-i}, h(X_{-i}, Z) \\ &\approx_{t \cdot 2^{-\frac{(\alpha-1)(n+3ts)-2s}{2}}} \quad U_i, U_{-i}, h(U_{-i}, Z) \ . \end{aligned}$$

Thus, using Lemma 8, we get that the statistical distance between $D(f(L), g(R))$ conditioned on $(L, R) \in \mathcal{L}_{\mathsf{ffb},i} \times \mathcal{R}^*$ and $h(x_1, x_{i-1}, x_{i+1}, \ldots, x_t, Z)$ is at most

$$2^{t(n+3ts)} \left(t \cdot 2^{-\frac{(\alpha-1)(n+3ts)-2s}{2}} + 2^{-t(n+3ts)-(s+1)} \right) \leq 2^{-s} \ .$$

□

Similarly, we get that

Lemma 37. *For all $i \in [t]$, for all $\mathcal{L}^* \subseteq \mathcal{L}$, if $|\mathcal{L}^* \times \mathcal{R}_{\mathsf{ffb},i}| \geq 2^{2\alpha t(n+3ts)-s}$ then there exists F distributed over $\mathcal{F}_n^t$, such that*

$$\Delta(D(f(L), g(R))|_{(L,R) \in \mathcal{L}^* \times \mathcal{R}_{\mathsf{ffb},i}} \ ; F(x)) \leq 2^{-s} \ .$$

6.4.2 Output of f or g is mixed

Lemma 38. *For all $j \in [t]$, for all $\mathcal{R}^* \subseteq \mathcal{R}_1$, if $|\mathcal{L}_{\mathsf{mix},j} \times \mathcal{R}^*| \geq 2^{2\alpha t(n+3ts)-s}$ then*

$$\Delta(D(f(L), g(R))|_{(L,R) \in \mathcal{L}_{\mathsf{mix},j} \times \mathcal{R}^*} \ ; \perp) \leq 2^{-s+1} \ .$$

Proof. Let $|\mathcal{L}_{\mathsf{mix},j} \times \mathcal{R}^*| \geq 2^{2\alpha t(n+3ts)-s}$ for some $i \in [t]$. Note that by the assumption we have that $|\mathcal{L}_{\mathsf{ffb},i}| \geq 2^{\alpha t(n+3ts)-s}$ and $|\mathcal{R}^*| \geq 2^{\alpha t(n+3ts)-s}$. Let $\tilde{L}, \tilde{R}$ be distributed uniformly over $\mathcal{L}_{\mathsf{mix},i}$ and $\mathcal{R}^*$ respectively. Denote $\mathsf{Ext}(\tilde{L}_k, \tilde{R}_k)$ by X_k for $k \in [t]$. Using a similar argument as in Lemma 36, we have that X_k is $2^{-\frac{(\alpha-1)(n+3ts)-2s}{2}}$-close to uniform given $\tilde{L}_1, \ldots, \tilde{L}_{k-1}, \tilde{R}_1, \ldots, \tilde{R}_{k-1}$, and hence using the hybrid argument, we have that

$$X_1, \ldots, X_t \approx_{t \cdot 2^{-\frac{(\alpha-1)(n+3ts)-2s}{2}}} U_1, \ldots, U_t \ , \tag{15}$$

where U_k, $k \in [t]$ being independent and uniformly distributed in $\{0,1\}^{\alpha(n+3ts)}$. We now give a lower bound for $\mathbf{H}_\infty(\tilde{L}_i | f_j(\tilde{L}))$ for any i using the definition of $\mathcal{L}_{\mathsf{mix},j}$.

$$\begin{aligned} \mathbf{H}_\infty(\tilde{L}_i | f_j(\tilde{L})) &= -\log \left(\sum_{y \in \{0,1\}^{\alpha(n+3ts)}} \max_{\ell_i \in \{0,1\}^{\alpha(n+3ts)}} \Pr(\tilde{L}_i = \ell_i \ \wedge f_j(\tilde{L}) = y) \right) \\ &\geq -\log \left(\sum_{y \in \{0,1\}^{\alpha(n+3ts)}} \frac{2^{\alpha(t-1)(n+3ts)-\beta_2}}{|\mathcal{L}_{\mathsf{mix},j}|} \right) \\ &\geq -\log \left(\frac{2^{\alpha(n+3ts)} \cdot 2^{\alpha(t-1)(n+3ts)-\beta_2}}{|\mathcal{L}_{\mathsf{mix},j}|} \right) \geq \beta_2 - s \ . \end{aligned}$$

Thus, we have that for all i, X_i is $2^{-\frac{\beta_2-n-3s}{2}}$-close to uniform given $f_j(\tilde{L}), \tilde{R}$, and hence,

$$\Delta\left(X_i, \mathsf{Ext}(f_j(\tilde{L}), g_j(\tilde{R}))\ ;\ U_i, \mathsf{Ext}(f_j(\tilde{L}), g_j(\tilde{R}))\right) \leq 2^{-\frac{\beta_2-n-3s}{2}}\ .$$

Also, since $\mathcal{L}_{\mathsf{mix},j}$ and $\mathcal{R}^*$ are subsets of $\mathcal{L}_1$, and $\mathcal{R}_1$, respectively, we have that

$$\mathbf{H}_\infty(f_j(\tilde{L})) \geq \mathbf{H}_\infty(f(\tilde{L})) - \alpha(t-1)(n+3ts) \geq \alpha(n+3ts) - \beta_1 - s\ ,$$

and

$$\mathbf{H}_\infty(g_j(\tilde{R})) \geq \alpha(n+3ts) - \beta_1 - s\ .$$

This implies that

$$\Delta\left(X_i, \mathsf{Ext}(f_j(\tilde{L}), g_j(\tilde{R}))\ ;\ U_i, U'_j\right) \leq 2^{-\frac{\beta_2-n-3s}{2}} + 2^{-\frac{(\alpha-1)(n+3ts)-2\beta_1-2s}{2}}\ , \tag{16}$$

where U'_j is uniform in $\{0,1\}^{\alpha(n+3ts)}$. Now, we claim that

$$\Delta(X_1, \ldots, X_t, \mathsf{Ext}(f_j(\tilde{L}), g_j(\tilde{R}))\ ;\ U_1, \ldots, U_t, U'_j) \leq 2^{-t(n+3ts)-s}\ . \tag{17}$$

If not, then by the generalized XOR Lemma (10), there exist $a_1, \ldots, a_{t+1}$ such that $\sum_{i=1}^t a_i X_i + a_{t+1}\mathsf{Ext}(f_j(\tilde{L}), g_j(\tilde{R}))$ is *not* $2^{-\frac{(2t+3)(n+3ts)+2s}{2}}$ close to uniform. By Equation 15, we have that $a_{t+1} \neq 0$, and by equation 16, we have that at least two of $a_1, \ldots, a_t$ are non-zero. Now, let $i_1, \ldots, i_k$ be all elements in $[t]$ such that $a_{i_j} \neq 0$. We know that $k \geq 2$. Consider two sources in $\mathbb{F}^{k+1}$ as $(a_{i_1}\tilde{L}_{i_1}, \ldots, a_{i_k}\tilde{L}_{i_k}, a_{t+1}f_j(\tilde{L}))$ and $(\tilde{R}_{i_1}, \ldots, \tilde{R}_{i_k}, g_j(\tilde{R}))$. Applying Ext to these two sources gives $\sum_i a_i X_i + a_{t+1}\mathsf{Ext}(f_j(\tilde{L}), g_j(\tilde{R}))$. The two sources have min-entropy at least $k\alpha(n+3ts) - s$, and hence $\sum_{i=1}^t a_i X_i + a_{t+1}\mathsf{Ext}(f_j(\tilde{L}), g_j(\tilde{R}))$ is

$$2^{-\frac{2k\alpha(n+3ts)-(k+1)\alpha(n+3ts)-(n+3ts)}{2}} \leq 2^{-\frac{(\alpha-1)(n+3ts)}{2}} < 2^{-\frac{(2t+3)(n+3ts)+2s}{2}}\ ,$$

which is a contradiction.

Thus, using Lemma 8 and Equation 17, we get that the statistical distance between $\mathsf{Ext}(f_j(\tilde{L}), g_j(\tilde{R}))$ conditioned on $(L,R) \in \mathcal{L}_{\mathsf{mix},j} \times \mathcal{R}^*$ and U'_j is at most $2^{t(n+3ts)} \cdot 2^{-t(n+3ts)-s} = 2^{-s}$. Thus, from Definition 31, we get that

$$\Pr(D(f(L), g(R)) = \perp \mid (L,R) \in \mathcal{L}_{\mathsf{mix},j} \times \mathcal{R}^*) \leq 2^{-s} + 2^{-3s} \leq 2^{-s+1}\ .$$

□

Similarly, we get that

Lemma 39. *For all $j \in [t]$, for all $\mathcal{L}^* \subseteq \mathcal{L}_1$, if $|\mathcal{L}^* \times \mathcal{R}_{\mathsf{mix},j}| \geq 2^{2\alpha t(n+3ts)-s}$ then*

$$\Delta(D(f(L), g(R))|_{(L,R)\in\mathcal{L}^*\times\mathcal{R}_{\mathsf{mix},j}}\ ; \perp) \leq 2^{-s+1}\ .$$

6.4.3 Both f and g are close to Identity

Lemma 40. *If $|\mathcal{L}_{\mathsf{Id}} \times \mathcal{R}_{\mathsf{Id}}| \geq 2^{\alpha t(n+3ts)-s}$ then there exists distribution H on functions $\mathcal{LA}_n^{\leftarrow t} \times \mathcal{LA}_n^{\leftarrow t}$ such that*

$$\Delta\big(D(f(L); g(R))|_{(L,R)\in\mathcal{L}_{\mathsf{Id}}\times\mathcal{R}_{\mathsf{Id}}}; H(x)\big) \leq 2^{-s+1}$$

Proof. Let $|\mathcal{L}_{\mathsf{Id}} \times \mathcal{R}_{\mathsf{Id}}| \geq 2^{2\alpha t(n+3ts)-s}$ for some $i \in [t]$. Note that by the assumption we have that $|\mathcal{L}_{\mathsf{Id}}| \geq 2^{\alpha t(n+3ts)-s}$ and $|\mathcal{R}_{\mathsf{Id}}| \geq 2^{\alpha t(n+3ts)-s}$. Let $\tilde{L}, \tilde{R}$ be distributed uniformly over $\mathcal{L}_{\mathsf{Id}}$ and $\mathcal{R}_{\mathsf{Id}}$ respectively. Denote $\mathsf{Ext}(\tilde{L}_k, \tilde{R}_k)$ by X_k, and $\mathsf{Ext}(f_k(\tilde{L}), g_k(\tilde{R}))$ by X'_k for $k \in [t]$.

Let a_i^f and a_i^g be such that $f_i(\tilde{L})$ is a function of a_i^f and $\tilde{L}_i$, and $g_i(\tilde{R})$ is a function of a_i^g and $\tilde{R}_i$, as defined in Lemma 35 (we shorthand $a_{i,i}$ by a_i and the superscript f, g are to distinguish between the corresponding functions for the two parts).

We define a few random variables. Let Y be a random variable defined as follows:

$$Y := \tilde{L}_1, \ldots, \tilde{L}_{t/2}, \;\; \tilde{R}_{t/2+1}, \ldots, \tilde{R}_t, \;\; a_1^f, \ldots, a_t^f, \;\; a_1^g, \ldots, a_t^g \; .$$

We first show that

$$X_1, \ldots, X_t, Y \approx_{t \cdot 2^{-\frac{(\alpha-1)(n+3ts)-t\beta_2-2s}{2}}} U_1, \ldots, U_t, Y \; , \tag{18}$$

where $U_1, \ldots, U_t$ are independent and uniform in $\{0,1\}^{n+3ts}$. For $i \in [t/2]$, we have that

$$\mathbf{H}_\infty(\tilde{L}_i) \geq \alpha(n+3ts) - s \; ,$$

and

$$\mathbf{H}_\infty(\tilde{R}_i | Y, \tilde{R}_{i+1}, \ldots, \tilde{R}_{t/2}) \geq \alpha(n+3ts) - t\beta_2 - s \; .$$

Similarly, for $i \in \{t/2+1, \ldots, t\}$, we have that

$$\mathbf{H}_\infty(\tilde{L}_i | Y, \tilde{L}_{i+1}, \ldots, \tilde{L}_{t/2}) \geq \alpha(n+3ts) - t\beta_2 - s \; ,$$

and

$$\mathbf{H}_\infty(\tilde{R}_i) \geq \alpha(n+3ts) - s \; ,$$

Thus, for all $i \in [t]$, using Lemma 7, we have that

$$U_1, \ldots, U_{i-1}, X_i, \ldots, X_t, Y \approx_{2^{-\frac{(\alpha-1)(n+3ts)-t\beta_2-2s}{2}}} U_1, \ldots, U_{i-1}, X_i, \ldots, X_t, Y \; .$$

Using the hybrid argument, this implies Equation 18.

Note that conditioned on Y, $(\tilde{R}_1, \ldots, \tilde{R}_{t/2})$ is independent of $(\tilde{L}_{t/2+1}, \ldots, \tilde{L}_t)$, and $X_1, \ldots, X_{t/2}$ (resp. $X_{t/2+1}, \ldots, X_t$) is a deterministic function of $(\tilde{R}_1, \ldots, \tilde{R}_{t/2})$ (resp. $(\tilde{L}_{t/2+1}, \ldots, \tilde{L}_t)$).

Define a sequence of random variables $W_1, \ldots, W_{t/2}$ iteratively as follows. Let W_i be independent randomness required to sample $\tilde{R}_i$ conditioned on $Y, W_1, \ldots, W_{i-1}, X_1, \ldots, X_i$.

Similarly, define a sequence of random variables $Z_1, \ldots, Z_{t/2}$ iteratively as follows. Let Z_i be independent randomness required to sample $\tilde{L}_{t/2+i}$ conditioned on $Y, Z_1, \ldots, Z_{i-1}, X_{t/2+1}, \ldots, X_{t/2+i}$.

Since X'_i is a function of Y and $\tilde{R}_i$ for $i \in \{1, \ldots, t/2\}$, and that of Y and $\tilde{L}_i$ for $i \in \{t/2+1, \ldots, t\}$, we have that conditioned on $X_1, \ldots, X_t$,

$$\begin{aligned} X'_1, \ldots, X'_t \;\; \equiv \;\; & h_1(X_1, W_1, Y), \; \ldots, h_{t/2}(X_1, \ldots, X_{t/2}, W_1, \ldots, W_{t/2}, Y) \; , \\ & h_{t/2+1}(X_{t/2+1}, Z_1, Y), \ldots, h_t(X_{t/2+1}, \ldots, X_t, Z_1, \ldots, Z_{t/2}, Y) \; , \end{aligned}$$

for some functions $h_1, \ldots, h_t$.

Note that $W_1, \ldots, W_{t/2}$, $Z_1, \ldots, Z_{t/2}$ are mutually independent and also independent of $X_1, \ldots, X_t$, Y. Thus, using equation 18, and Lemma 2, we get that $X_1, \ldots, X_t, X'_1, \ldots, X'_t$ has statistical distance at most $t \cdot 2^{-\frac{(\alpha-1)(n+3ts)-t\beta_2-2s}{2}}$ from

$$\begin{aligned} & U_1, \ldots, U_t, h_1(U_1, W_1, Y), \; \ldots, \; h_{t/2}(U_1, \ldots, U_{t/2}, W_1, \ldots, W_{t/2}, Y), \; h_{t/2+1}(U_{t/2+1}, Z_1, Y), \\ & \ldots, h_t(U_{t/2+1}, \ldots, U_t, Z_1, \ldots, Z_{t/2}, Y) \; . \end{aligned}$$

Thus, using Lemma 8, we get that the statistical distance between $D(f(L), g(R))$ conditioned on $(L, R) \in \mathcal{L}_{\mathsf{Id}} \times \mathcal{R}_{\mathsf{Id}}$ and a function H (described by $h_1, \ldots, h_t$) distributed over $\mathcal{LA}_n^{\leftarrow t/2} \times \mathcal{LA}_n^{\leftarrow t/2}$ is at most

$$2^{t(n+3ts)} \left(t \cdot 2^{-\frac{(\alpha-1)(n+3ts)-t\beta_2-2s}{2}} \right) \leq 2^{-s} .$$

□

6.4.4 f or g is Close to a non-identity Permutation

Lemma 41. *For all $i, j \in [t]$, such that $j > i$, and for all $\mathcal{R}^* \subseteq \mathcal{R}_1$, if $|\mathcal{L}_{\mathsf{perm},i\leftarrow j} \times \mathcal{R}^*| \geq 2^{2\alpha t(n+3ts)-s}$ then*

$$\Delta\big[D(f(L), g(R))|_{(L,R)\in\mathcal{L}_{\mathsf{perm},i\leftarrow j}\times\mathcal{R}^*} \; ; \perp\big] \leq 6 \cdot 2^{-s},$$

where $\perp$ denotes constant function equal $\perp$.

Proof. Let $\tilde{L}, \tilde{R}$ be distributed uniformly over $\mathcal{L}_{\mathsf{perm},i\leftarrow j}$ and $\mathcal{R}_1$ respectively. For purpose of this proof let us define two random vectors:

$$C = \big[\tilde{R},\, (\tilde{L}_k)_{k\neq j},\, a_{j,i}(\tilde{L})\big],$$

$$C' = \big[(\tilde{L}_k)_{k\neq j},\, a_{j,i}(\tilde{L})\big],$$

where $a_{j,i}$ is defined in Lemma 35. Let us observe :

$$\begin{aligned}
&\Delta\big[D(f(L), g(R))|_{(L,R)\in\mathcal{L}_{\mathsf{perm},i\leftarrow j}\times\mathcal{R}_1} \; ; \perp\big] \leq \Delta\big[D(f(\tilde{L}), g(\tilde{R})) \; ; \perp \;\big|\; C\big] = \\
&= \sum_c \Pr\big[C = c\big] \cdot \Pr\big[D(f(\tilde{L}), g(\tilde{R})) \neq \perp \;\big|\; C = c\big] \leq \\
&\leq \sum_c \Pr\big[C = c\big] \cdot \Pr\big[h_i(\mathsf{Ext}(f_i(\tilde{L}), g_i(\tilde{R})) \neq \perp \big| \, C = c,\, h_j(\mathsf{Ext}(\tilde{L}_j, \tilde{R}_j) \neq \perp\big] = \\
&= \sum_c \Pr\big[C = c\big] \cdot \frac{\Pr\big[h_i(\mathsf{Ext}(f_i(\tilde{L}), g_i(\tilde{R})) \neq \perp \text{ and } h_j(\mathsf{Ext}(\tilde{L}_j, \tilde{R}_j) \neq \perp \big| \, C = c\big]}{\Pr\big[h_j(\mathsf{Ext}(\tilde{L}_j, \tilde{R}_j) \neq \perp \big| \, C = c\big]} = (*)
\end{aligned}$$

Let us notice that by Lemma 35 and size of $|\mathcal{L}_{\mathsf{perm},i\leftarrow j} \times \mathcal{R}_1|$ by similar argument as in Lemma 36 for every c we get

$$\mathbf{H}_\infty(f_i(\tilde{L})|C' = c') = \mathbf{H}_\infty(b_{j,i}(\tilde{L}_j, a_{j,i}(\tilde{L})|C' = c') \geq \alpha(n + 3ts) - s - \beta_1 - \beta_2$$

and analogously

$$\mathbf{H}_\infty(g_i(\tilde{R})|C' = c') \geq \alpha(n + 3ts) - s - \beta_1 - \beta_2.$$

Therefor by strong extractor properties we obtain

$$\Delta\big[\mathsf{Ext}(f_i(\tilde{L}), g_i(\tilde{R})) \; ; \; U\big|C' = c', \tilde{R}\big] \leq 2^{-\left[1/2(\alpha-2)(n+3ts)-s-\beta_1-\beta_2\right]} \leq 2^{-3ts}$$

and similarly

$$\Delta\big[\mathsf{Ext}(\tilde{L}_j, \tilde{R}_j) \; ; \; U\big|C' = c', \tilde{R}\big] \leq 2^{-3ts}.$$

Thus above we obtain that

$$\Delta\big[h_i(\mathsf{Ext}(f_i(\tilde{L}), g_i(\tilde{R}))) \; ; \; \perp\big| \; C' = c', \tilde{R}\big] \leq 2^{-3[t-i+1]s} + 2^{-3ts} \leq 2 \cdot 2^{-3(t-i+1)s},$$

while

$$\Delta\big[h_j(\mathsf{Ext}(\tilde{L}_j, \tilde{R}_j)) \; ; \; \perp\big| \; C' = c', \tilde{R}\big] \geq 2^{-3[t-j+1]s} - 2^{-3ts} \geq 1/2 \cdot 2^{-3(t-j+1)s}.$$

Since

$$2 \cdot 2^{-(t-i+1)s} \;\geq\; \Delta\big[h_i(\mathsf{Ext}(f_i(\tilde{L}), g_i(\tilde{R}))) \;;\; \bot \big|\; C' = c', \tilde{R}\big] =$$
$$= \sum_r \Pr(\tilde{R} = r) \cdot \Pr\big[h_i(\mathsf{Ext}(f_i(\tilde{L}), g_i(\tilde{R}))) \;;\; \bot \big|\; C' = c', \tilde{R} = r\big].$$

Let us define set $\mathcal{A}$ of all r such that

$$\Pr\big[h_i(\mathsf{Ext}(f_i(\tilde{L}), g_i(\tilde{R}))) = \;\bot \;\big|\; C' = c', \tilde{R} = r\big] > 2 \cdot 2^{-3(t-i+1)s+s}$$

by Markov argument we obtain that $\Pr(\tilde{R} \in \mathcal{A}) \leq 2^{-s}$. Similarly for set $\mathcal{B}$ of all r such that

$$\Pr\big[h_j(\mathsf{Ext}(\tilde{L}_j, \tilde{R}_j)) = \;\bot \;\big|\; C' = c', \tilde{R} = r\big] < 1/2 \cdot 2^{-3(t-j+1)s-s}$$

we obtain that $\Pr(\tilde{R} \in \mathcal{B}) \leq 2^{-s}$. To finish the proof we notice that

$$\begin{aligned}
(*) = &\sum_{c', r \in \mathcal{A} \cup \mathcal{B}} \Pr\big[C' = c' \text{ and } \tilde{R} = r\big] \cdot \Delta\big[D(f(\tilde{L}), g(\tilde{R})) \;; \bot \;\big|\; C' = c', \tilde{R} = r,\, h_j(\mathsf{Ext}(\tilde{L}_j, \tilde{R}_j) \neq \bot\big] + \\
&+ \sum_{c', r \notin \mathcal{A} \cup \mathcal{B}} \Pr\big[C' = c' \text{ and } \tilde{R} = r\big] \cdot \frac{\Pr\big[h_i(\mathsf{Ext}(f_i(\tilde{L}), g_i(\tilde{R})) \neq \bot \text{ and } h_j(\mathsf{Ext}(\tilde{L}_j, \tilde{R}_j) \neq \bot \,\big|\, C = c\big]}{\Pr\big[h_j(\mathsf{Ext}(\tilde{L}_j, \tilde{R}_j) \neq \bot \,\big|\, C = c\big]} \\
&\leq \sum_{c', r \in \mathcal{A} \cup \mathcal{B}} \Pr\big[C' = c' \text{ and } \tilde{R} = r\big] \cdot \Delta\big[D(f(\tilde{L}), g(\tilde{R})) \;; \bot \;\big|\; C' = c', \tilde{R} = r,\, h_j(\mathsf{Ext}(\tilde{L}_j, \tilde{R}_j) \neq \bot\big] + \\
&+ \sum_{c', r \notin \mathcal{A} \cup \mathcal{B}} \Pr\big[C' = c' \text{ and } \tilde{R} = r\big] \cdot \frac{\Pr\big[h_i(\mathsf{Ext}(f_i(\tilde{L}), g_i(\tilde{R})) \neq \bot \big|\, C' = c', \tilde{R} = r\big]}{\Pr\big[h_j(\mathsf{Ext}(\tilde{L}_j, \tilde{R}_j) \neq \bot \,\big|\, C' = c', \tilde{R} = r\big]} \leq \\
&\leq 2 \cdot 2^{-s} + \frac{2 \cdot 2^{-3(t-i+1)s+s}}{1/2 \cdot 2^{-3(t-j+1)s-s}} = \\
&\text{now notice that } j \geq i+1 \\
&= 2 \cdot 2^{-s} + 4 \cdot 2^{-s} = 6 \cdot 2^{-s} \,.
\end{aligned}$$

□

Similarly, we have that

Lemma 42. *For all $i, j \in [t]$, such that $j > i$, and for all $\mathcal{L}^* \subseteq \mathcal{L}_1$, if $|\mathcal{L}^* \times \mathcal{R}_{\mathsf{perm},i\leftarrow j}| \geq 2^{2\alpha t(n+3ts)-s}$ then*

$$\Delta\big[D(f(L), g(R))|_{(L,R) \in \mathcal{L}^* \times \mathcal{R}_{\mathsf{perm},i\leftarrow j}} \;; \bot\big] \leq 6 \cdot 2^{-s},$$

where $\bot$ denotes constant function equal $\bot$.

6.4.5 Remaining case happens with small probability

Lemma 43. $|\mathcal{L}_{\mathsf{rem}}| \leq 2^{2\alpha t(n+3ts)-s}$.

Proof. Consider any $\ell \in \mathcal{L}_{\mathsf{rem}}$. For any $j \in [t]$, we know that $\ell \notin \mathcal{L}_{\mathsf{mix},j}$, and hence $j \in \mathcal{T}(\ell, i)$ for some i. Thus, $\cup_{i=1}^{t} \mathcal{T}(\ell, i) = [t]$. Also, since $\ell \notin \mathcal{L}_{\mathsf{perm},\pi}$ for any π, there exists some $k \in [t]$, such that $|\mathcal{T}(\ell, k)| \geq 2$. Let $j_1, j_2 \in \mathcal{T}(\ell, k)$. For any $j \in [t] \setminus \{j_1, j_2\}$, let $e(j)$ be some $i \in [t]$ such that $j \in \mathcal{L}(T, e(j))$. Thus, using Lemma 35, we have that $f(\ell)$ can be determined given ℓ_k, a_{k,j_1}, a_{k,j_2} and $\ell_{e(j)}, a_{e(j),j}$ for all $j \in [t] \setminus \{j_1, j_2\}$. This implies that $f(\ell)$ can be determined given at most $(t-1)\alpha(n+3ts) + t\beta_2 \leq t\alpha(n+3ts) - s$ bits. This implies $|\mathcal{L}_{\mathsf{rem}}| \leq 2^{t\alpha(n+3ts)-s}$.

□

Similarly,

Lemma 44. $|\mathcal{R}_{\mathsf{rem}}| \leq 2^{2\alpha t(n+3ts)-s}$.

7 Conclusions and Open Problems

We have built the first efficient, information-theoretically secure non-malleable codes in the split-state model with constant encoding rate. Although asymptotically optimal, the constant we achieve appears astronomical, as it relies on some results in additive combinatorics from [CZ14]. In contrast, we know existentially that the optimal rate is equal to 2. Closing this gap is an interesting open problem.

We have also introduced the notion of non-malleable reductions, and showed that they allow to build non-malleable codes in a modular manner. We hope that our modularity will find further applications in the design of other non-malleable codes.

Acknowledgments. We would like to thank Eshan Chattopadhyay and David Zuckerman for sharing with us an early version of their work.

References

[ADKO14] Divesh Aggarwal, Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Non-malleable codes in the interactive split-state model. Cryptology ePrint Archive, Report 2014/807, 2014. `http://eprint.iacr.org/`.

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

[Agg14] Divesh Aggarwal. Affine-evasive sets modulo a prime. Cryptology ePrint Archive, Report 2014/328, 2014. `http://eprint.iacr.org/`.

[CCFP11] Hervé Chabanne, Gérard Cohen, J Flori, and Alain Patey. Non-malleable codes from the wire-tap channel. In *Information Theory Workshop (ITW), 2011 IEEE*, pages 55–59. IEEE, 2011.

[CCP12] Herve Chabanne, Gerard Cohen, and Alain Patey. Secure network coding and non-malleable codes: Protection against linear tampering. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 2546–2550. IEEE, 2012.

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

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

[CKM11] Seung Geol Choi, Aggelos Kiayias, and Tal Malkin. Bitr: built-in tamper resilience. In *Advances in Cryptology–ASIACRYPT 2011*, pages 740–758. Springer, 2011.

[CZ14] Eshan Chattopadhyay and David Zuckerman. Non-malleable codes in the constant split-state model. *To appear in FOCS*, 2014.

[DDN00] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM*, 30:391–437, 2000.

[DKO13] Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Non-malleable codes from two-source extractors. In *Advances in Cryptology-CRYPTO 2013*. Springer, 2013.

[DP07a] Yevgeniy Dodis and Prashant Puniya. Feistel networks made public, and applications. In Moni Naor, editor, *Advances in Cryptology - EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 534–554. Springer-Verlag, 2007.

[DP07b] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *FOCS*, pages 227–237. IEEE Computer Society, 2007.

[DPW10] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In Andrew Chi-Chih Yao, editor, *ICS*, pages 434–452. Tsinghua University Press, 2010.

[DSDCO+01] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In *Advances in Cryptology-Crypto 2001*, pages 566–598. Springer, 2001.

[DW09] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 601–610, Bethesda, MD, USA, 2009. ACM.

[FMNV14] S. Faust, P. Mukherjee, J. Nielsen, and D. Venturi. Continuous non-malleable codes. In *Theory of Cryptography Conference - TCC*. Springer, 2014.

[FMVW14] S. Faust, P. Mukherjee, D. Venturi, and D. Wichs. Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In *Eurocrypt*. Springer, 2014. To appear.

[GLM+03] Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic Tamper-Proof (ATP) security: Theoretical foundations for security against hardware tampering. In Moni Naor, editor, *First Theory of Cryptography Conference — TCC 2004*, volume 2951 of *LNCS*, pages 258–277. Springer-Verlag, February 19–21 2003.

[Gre05] B Green. Finite field models in additive number theory. *Surveys in Combinatorics*, pages 1–29, 2005.

[GUV07] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. Unbalanced expanders and randomness extractors from parvaresh-vardy codes. In *IEEE Conference on Computational Complexity*, pages 96–108. IEEE Computer Society, 2007.

[HILL99] J. Håstad, R. Impagliazzo, L.A. Levin, and M. Luby. Construction of pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[IPSW06] Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private circuits II: Keeping secrets in tamperable circuits. In Serge Vaudenay, editor, *Advances in Cryptology—EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 308–327. Springer-Verlag, 2006.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology—CRYPTO 2003*, volume 2729 of *LNCS*. Springer-Verlag, 2003.

[KKS11] Yael Tauman Kalai, Bhavana Kanukurthi, and Amit Sahai. Cryptography with tamperable and leaky memory. In *Advances in Cryptology–CRYPTO 2011*, pages 373–390. Springer, 2011.

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

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *LNCS*, pages 18–35. Springer-Verlag, 2009.

[San12] T Sanders. On the bogolyubov-ruzsa lemma, anal. *PDE*, 5:627–655, 2012.

A Proofs of Lemmata from Section 2

Lemma 8 *Let $X_1, Y_1 \in \mathcal{A}_1$, and $Y_1, Y_2 \in \mathcal{A}_2$ be random variables such that $\Delta((X_1, X_2)\ ;\ (Y_1, Y_2)) \leq \varepsilon$. Then, for any non-empty set $\mathcal{A}' \subseteq \mathcal{A}_1$, we have*

$$\Delta(X_2 \mid X_1 \in \mathcal{A}'\ ;\ Y_2 \mid Y_1 \in \mathcal{A}') \leq \frac{2\varepsilon}{\Pr(X_1 \in \mathcal{A}')}\ .$$

Proof.

$$\begin{aligned}
\Delta(X_2 \mid X_1 \in \mathcal{A}'\ ;\ Y_2 \mid Y_1 \in \mathcal{A}') &= \frac{1}{2}\sum_{x\in\mathcal{A}_2}\Big|\Pr(X_2 = x \mid X_1 \in \mathcal{A}') - \Pr(Y_2 = x \mid Y_1 \in \mathcal{A}')\Big| \\
&\leq \frac{1}{2}\sum_{x\in\mathcal{A}_2}\Big(\Big|\frac{\Pr(X_2 = x\ \wedge\ X_1 \in \mathcal{A}')}{\Pr(X_1 \in \mathcal{A}')} - \frac{\Pr(Y_2 = x\ \wedge\ Y_1 \in \mathcal{A}')}{\Pr(X_1 \in \mathcal{A}')}\Big| \\
&\qquad + \Pr(Y_2 = x\ \wedge\ Y_1 \in \mathcal{A}')\Big|\frac{1}{\Pr(Y_1 \in \mathcal{A}')} - \frac{1}{\Pr(X_1 \in \mathcal{A}')}\Big|\Big) \\
&\leq \frac{\varepsilon}{\Pr(X_1 \in \mathcal{A}')} + \frac{\varepsilon \cdot \sum_{x\in\mathcal{A}_2}\Pr(Y_1 \in \mathcal{A}'\ \wedge\ Y_2 = x)}{\Pr(Y_1 \in \mathcal{A}') \cdot \Pr(X_1 \in \mathcal{A}')} \\
&= \frac{2\varepsilon}{\Pr(X_1 \in \mathcal{A}')}\ .
\end{aligned}$$

□

Lemma 9 *Let $A \in \mathcal{A}$ and $B \in \mathcal{B}$ be two independent random variables. Let $V_1, V_2, \ldots$ be random variables defined as functions of A, B satisfying the following property. For all $i \in \mathbb{N}$, if i is even then $V_i = \phi_i(V_1, \ldots, V_{i-1}, A)$ and if i is odd, then $V_i = \phi_i(V_1, \ldots, V_{i-1}, B)$ for some function ϕ_i. Then for all i, A is independent of B given $V_1, \ldots, V_i$.*

Proof. For any even j, and any fixed values $v_1, \ldots, v_j$ in the support of $V_1, \ldots, V_j$, respectively, define the set $\phi_j^{-1}(v_j | v_1, \ldots, v_{j-1})$ as follows:

$$\phi_j^{-1}(v_j | v_1, \ldots, v_{j-1}) = \{a \in \mathcal{A} \mid \phi_j(v_1, \ldots, v_{j-1}, a) = v_j\} .$$

Similarly define $\phi_j^{-1}(v_j | v_1, \ldots, v_{j-1})$ for odd j with A replaced by B. Now, we show the result when i is even. The result follows similarly for odd i. The event (call it $E_{v_1,\ldots,v_i}$) that $V_1 = v_1, \ldots, V_i = v_i$ is the same as

$$B \in \phi_1^{-1}(v_1) \ \wedge \ A \in \phi_2^{-1}(v_2 | v_1) \ \cdots \ \wedge A \in \phi_i^{-1}(v_i | v_1, \ldots, v_{i-1}) ,$$

which is the same as

$$A \in \bigcap_{j=1}^{i/2} \phi_{2j}^{-1}(v_{2j} | v_1, \ldots, v_{2j-1}) \ \wedge \ B \in \bigcap_{j=1}^{i/2} \phi_{2j-1}^{-1}(v_{2j-1} | v_1, \ldots, v_{2j-2}) .$$

Thus, A and B are independent given the event $E_{v_1,\ldots,v_i}$ for all $v_1, \ldots, v_i$, which implies the result. □

Lemma 10 *Let $X = (X_1, \ldots, X_t) \in \mathbb{F}^t$ be a random variable, where $\mathbb{F}$ is a finite field of order q. Assume that for all $a_1, \ldots, a_t \in \mathbb{F}^t$ not all zero, $\Delta(\sum_{i=1}^t a_i X_i \ ; \ U) \leq \varepsilon$, where U is uniform in $\mathbb{F}$. Then $\Delta(X_1, \ldots, X_t \ ; \ U_1, \ldots, U_t) \leq \varepsilon q^{(t+2)/2}$, where $U_1, \ldots, U_t$ are independent and uniform in $\mathbb{F}^t$.*

Proof. The proof uses basic Fourier analysis. Assume $\mathbb{F}$ has characteristic p. Let $\omega = e^{2\pi i/p}$ be a primitive p-th root of unity. Let $\mathrm{Tr} : \mathbb{F} \to \mathbb{F}_p$ denote the trace operator from $\mathbb{F}$ to $\mathbb{F}_p$. The additive characters of $\mathbb{F}$ are given by $\{\chi_a(x) : \mathbb{F} \to \mathcal{C} : a \in \mathbb{F}\}$ defined as

$$\chi_a(x) = \omega^{\mathrm{Tr}(ax)}.$$

The additive characters of $\mathbb{F}^t$ are given by $\chi_{a_1,\ldots,a_t}(x_1, \ldots, x_t) = \Pi_{i=1}^t \chi_{a_i}(x_i)$ for $a_1, \ldots, a_t \in \mathbb{F}$.

First, we bound the Fourier coefficients of the distribution of $X = (X_1, \ldots, X_t)$. The $(a_1, \ldots, a_t)$ Fourier coefficient, for all non-zero $(a_1, \ldots, a_t)$, is given by

$$\begin{aligned} \mathbb{E}[\chi_{a_1,\ldots,a_t}(X_1, \ldots, X_t)] &= \mathbb{E}[\omega^{\mathrm{Tr}(\sum_{i=1}^t a_i X_i)}] = \sum_{b \in \mathbb{F}} \omega^{\mathrm{Tr}(b)} \Pr[\sum_{i=1}^t a_i X_i = b] \\ &= \sum_{b \in \mathbb{F}} \omega^{\mathrm{Tr}(b)} \left(\Pr_X[\sum_{i=1}^t a_i X_i = b] - \frac{1}{|\mathbb{F}|} \right), \end{aligned}$$

where we used the fact that $\sum_{b \in \mathbb{F}} \omega^{\mathrm{Tr}(b)} = 0$. Hence for all non-zero $(a_1, \ldots, a_t)$,

$$\left| \mathbb{E}[\chi_{a_1,\ldots,a_t}(X_1, \ldots, X_t)] \right| \leq \sum_{b \in \mathbb{F}} \left| \Pr[\sum_{i=1}^t a_i X_i = b] - \frac{1}{|\mathbb{F}|} \right| \leq 2\varepsilon \cdot |\mathbb{F}| .$$

Let $p_{a_1,\ldots,a_t} = \Pr[(X_1,\ldots,X_t) = (a_1,\ldots,a_t)]$. By Parseval's identity,

$$\sum_{a_1,\ldots,a_t \in \mathbb{F}} \left(p_{a_1,\ldots,a_t} - \frac{1}{|\mathbb{F}|} \right)^2 = \sum_{(a_1,\ldots,a_t) \neq 0} \mathbb{E}[\chi_{a_1,\ldots,a_t}(X_1,\ldots,X_t)]^2 \leq 4\varepsilon^2 |\mathbb{F}|^{t+2} ,$$

□

B Equivalence of our definition with [DPW10]

We first recall the definition of non-malleable codes from [DPW10].

Definition 45. A *coding scheme* consists of two functions: a randomized encoding function $E : \{0,1\}^k \mapsto \{0,1\}^n$, and a deterministic decoding function $D : \{0,1\}^n \mapsto \{0,1\}^k \cup \{\perp\}$ such that, for each $x \in \{0,1\}^k$, $\Pr(D(E(x)) = x) = 1$ (over the randomness of the encoding algorithm).

Definition 46. Let $\mathcal{F}$ be some family of tampering functions. For each $f \in \mathcal{F}$, and $x \in \{0,1\}^k$, define the tampering-experiment

$$\mathsf{Tamper}_x^f := \left\{ \begin{array}{c} c \leftarrow E(x),\ \tilde{c} \leftarrow f(c),\ \tilde{x} = D(\tilde{c}) \\ \text{Output: } \tilde{x}. \end{array} \right\}$$

which is a random variable over the randomness of the encoding function E. We say that a coding scheme (E, D) *is ε-non-malleable w.r.t.* $\mathcal{F}$ if for each $f \in \mathcal{F}$, there exists a distribution (corresponding to the simulator) D_f over $\{0,1\}^k \cup \{\perp, \mathsf{same}\}$, such that, for all $x \in \{0,1\}^k$, we have that the statistical distance between Tamper_x^f and

$$\mathsf{Sim}_x^f := \left\{ \begin{array}{c} \tilde{x} \leftarrow D_f \\ \text{Output: } x \text{ if } \tilde{x} = \mathsf{same}, \text{ and } \tilde{x}, \text{ otherwise.} \end{array} \right\}$$

is at most ε.

We now show that Definition 12 and Definition 46 are equivalent.

Theorem 47. *There is a coding scheme $E : \{0,1\}^k \mapsto \{0,1\}^n$, $D : \{0,1\}^n \mapsto \{0,1\}^k \cup \{\perp\}$ that is ε-non-malleable w.r.t. $\mathcal{F}$ according to Definition 46 if and only if there is a coding scheme $E' : \{0,1\}^k \mapsto \{0,1\}^n$, $D' : \{0,1\}^n \mapsto \{0,1\}^k$ that is ε-non-malleable w.r.t. $\mathcal{F}$ according to Definition 12.*

Proof. Consider the coding scheme $E : \{0,1\}^k \mapsto \{0,1\}^n$ and $D : \{0,1\}^n \mapsto \{0,1\}^k$ that is non-malleable according to Definition 46. Define E' to be identical to E and D' to be such that for all $c \in \{0,1\}^n$, $D'(c) = D(c)$, if $D(c) \neq \perp$, and $D'(c) = 0^k$, otherwise. Consider an arbitrary $f \in \mathcal{F}$, $x \in \{0,1\}^k$.

Let D_f be as in Definition 46. Then, we need to define G such that it is a non-malleable reduction from $\mathcal{F}$ to NM_k. We define G as follows. The function

$$G = \left\{ \begin{array}{ll} f_{x'} & \text{if } D_f = x' \in \{0,1\}^k \\ f_{0^k} & \text{if } D_f = \perp \\ I & \text{if } D_f = \mathsf{same} . \end{array} \right.$$

Then, it is easy to see that $G(x) = \mathsf{Sim}_x^f$, if $\mathsf{Sim}_x^f \neq \perp$, and $G(x) = 0^k$, otherwise. Similarly, $D'(f(E'(x)) = \mathsf{Tamper}_x^f$, if $\mathsf{Tamper}_x^f \neq \perp$, and $D'(f(E'(x))) = 0^k$, otherwise. Thus, using Definition 46, and Lemma 2, we have that

$$\Delta(D'(f(E'(x))), G(x)) \leq \varepsilon .$$

Now assume $E' : \{0,1\}^k \mapsto \{0,1\}^n$, $D' : \{0,1\}^n \mapsto \{0,1\}^k$ is any coding scheme that is ε-non-malleable w.r.t. $\mathcal{F}$ according to definition 12. We now show that E', D' is also non-malleable according to Definition 46. The proof is even simpler for this case. Consider an arbitrary $f \in \mathcal{F}$, $x \in \{0,1\}^k$. We define distribution D_f with support $\{0,1\}^k \cup \{\mathsf{same}\}$ (i.e., it never outputs $\perp$) as follows.

$$D_f = \begin{cases} x' & \text{if } G = f_{x'}, \text{ for some } x' \in \{0,1\}^k \\ \mathsf{same} & \text{if } G = I \ . \end{cases}$$

Then, clearly, $\mathsf{Sim}^f_x = G(x)$, and $\mathsf{Tamper}^f_x = D'(f(E'(x))$, and hence the result follows from Definition 12. □

Note that the definition in [DPW10] also had an additional requirement that D_f is efficiently samplable given oracle access to f. We did not include this in Definition 46 since it is implied by the fact that E, D are efficient, as shown below.

Lemma 13. *Let (E, D) be an $(\mathcal{F}, k, \varepsilon)$-non-malleable code for some tampering family $\mathcal{F}$. Then for all $f \in \mathcal{F}$, there exists a random function G distributed over NM_k such that for all $x \in \{0,1\}^k$,*

$$\Delta\Big(D(f(E(x))) \ ; \ G(x)\Big) \leq 2\varepsilon + \frac{1}{2^k} \ ,$$

and G is efficiently samplable given oracle access to f.

Proof. By Theorem 47, E, D is also ε-non-malleable w.r.t. $\mathcal{F}$ acording to Definition 46. We will use this definition for this proof. Fix $f \in \mathcal{F}$, and let D_f distributed over $\{0,1\}^k \cup \{\mathsf{same}\}$ be as in the proof of Theorem 47. Then for all $x \in \{0,1\}^k$, we have

$$\begin{aligned} 2\varepsilon \ \geq \ & \left|\Pr\left(D(f(E(x))) = x\right) - \Pr(D_f = \mathsf{same}) - \Pr(D_f = x)\right| \\ & + \sum_{x' \in \{0,1\}^k \setminus \{x\}} \left|\Pr\left(D(f(E(x))) = x'\right) - \Pr(D_f = x')\right| \ . \end{aligned}$$

Summing over all $x \in \{0,1\}^k$, this implies that

$$\begin{aligned} 2^{k+1}\varepsilon \ \geq \ & \sum_{x \in \{0,1\}^k} \left|\Pr\left(D(f(E(x))) = x\right) - \Pr(D_f = \mathsf{same})\right| - \sum_{x \in \{0,1\}^k} \left|\Pr(D_f = x)\right| \\ & + \sum_{\substack{x, x' \in \{0,1\}^k \\ x' \neq x}} \left|\Pr\left(D(f(E(x))) = x'\right) - \Pr(D_f = x')\right| \ . \end{aligned}$$

This implies

$$\begin{aligned} 2^{k+1}\varepsilon + 1 \ \geq \ & \sum_{x \in \{0,1\}^k} \left|\Pr\left(D(f(E(x))) = x\right) - \Pr(D_f = \mathsf{same})\right| \\ & + \sum_{\substack{x, x' \in \{0,1\}^k \\ x' \neq x}} \left|\Pr\left(D(f(E(x))) = x'\right) - \Pr(D_f = x')\right| \ . \end{aligned} \tag{19}$$

Now consider the following distribution $\tilde{D}_f$ defined by the following sampling procedure.

1. Sample uniformly random element $U_k \leftarrow \{0,1\}^k$.
2. Compute $D(f(E(U_k)))$.

3. If $D(f(E(U_k))) = U_k$, then output same, else output $D(f(E(U_k)))$.

Clearly, $\tilde{D}_f$ is efficiently samplable given oracle access to f. We now bound $\Delta(\tilde{D}_f, D_f)$.

$$\begin{aligned}
2 \cdot \Delta(\tilde{D}_f, D_f) &\leq |\Pr(D(f(E(U_k))) = U_k) - \Pr(D_f = \mathsf{same})| \\
&\quad + \sum_{x \in \{0,1\}^k} |\Pr(D(f(E(U_k))) = x, U_k \neq x) - \Pr(D_f = x)| \\
&= \left| \frac{1}{2^k} \sum_{x \in \{0,1\}^k} \Pr(D(f(E(x))) = x) - \Pr(D_f = \mathsf{same}) \right| \\
&\quad + \sum_{x \in \{0,1\}^k} \left| \frac{1}{2^k} \sum_{x' \neq x} \Pr(D(f(E(x'))) = x) - \Pr(D_f = x) \right| \\
&\leq \frac{1}{2^k} \sum_{x \in \{0,1\}^k} |\Pr\left(D(f(E(x))) = x\right) - \Pr(D_f = \mathsf{same})| \\
&\quad + \frac{1}{2^k} \sum_{\substack{x, x' \in \{0,1\}^k \\ x' \neq x}} \left|\Pr\left(D(f(E(x))) = x'\right) - \Pr(D_f = x')\right| + \frac{\sum_{x \in \{0,1\}^k} \Pr(D_f = x)}{2^k} \\
&\leq 2\varepsilon + \frac{1}{2^k} + \frac{1}{2^k} = 2\varepsilon + \frac{2}{2^k} \ ,
\end{aligned}$$

where the last inequality uses Equation 19. Thus, $\Delta(\tilde{D}_f, D_f) \leq \varepsilon + \frac{1}{2^k}$, which implies the result. □

C Proof of Theorem 23 using [CZ14]

Cheraghchi and Guruswami [CG14b] showed that it is sufficient to construct t-source non-malleable extractors with sources of length n in order to construct non-malleable codes against the tampering family $\mathcal{S}_n^t$. We observe here that if the non-malleable extractor is also a strong extractor, then the corresponding coding scheme is also non-malleable against the t-part forgetful tampering family. The following is a definition of a t-source strong non-malleable extractor.

Definition 48. *A function* $\mathsf{nmExt} : (\{0,1\}^n)^t \mapsto \{0,1\}^k$ *is a* t-source ε-strong non-malleable extractor *if, for* $X_1, \ldots, X_t$ *uniformly distributed in* $\{0,1\}^n$*, and* Y *distributed uniformly in* $\{0,1\}^m$ *it satisfies the following properties.*

- *For any* $f_1, \ldots, f_t : \{0,1\}^n \mapsto \{0,1\}^n$*, there exist some* $G \in \mathsf{NM}_k$ *such that*

$$\Delta\left((\mathsf{nmExt}(X_1, \ldots, X_t), \mathsf{nmExt}(f_1(X_1), \ldots, f_t(X_t))) \ ; \ (Y, G(Y))\right) \leq \varepsilon \ .$$

- *For any* $i \in [t]$*,*

$$\Delta\left(\mathsf{nmExt}(X_1, \ldots, X_t) \ ; \ Y \mid X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_t\right) \leq \varepsilon \ .$$

The following result follows almost immediately from the definitions.

Lemma 49. *If there exists an efficiently computable* t*-source* ε*-strong non-malleable extractor* $\mathsf{nmExt} : (\{0,1\}^n)^t \mapsto \{0,1\}^k$*, then*

$$(\mathcal{S}_n^t \cup \mathcal{FOR}_n^t \to \mathsf{NM}_k, \varepsilon) \ .$$

If nmExt *is efficiently invertible, then using Theorem 16, this implies*

$$(\mathcal{S}_n^t \cup \mathcal{FOR}_n^t \Rightarrow \mathsf{NM}_k, \varepsilon \cdot 2^{k+1}) \ .$$

Proof. The transformation T is simply nmExt. From the first condition in Definition 48, it is clear that for this transformation, $(\mathcal{S}_n^t \to \mathsf{NM}_k, \varepsilon)$.

Now, consider the forgetful tampering family. Let $(X_1, \ldots, X_t)$ be distributed uniformly in $(\{0,1\}^n)^t$. Fix $i \in [t]$, and let $X'_1, \ldots, X'_t \in \{0,1\}^n$ be random variables that depend arbitrarily on

$$X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_t \ .$$

Then using the second condition of Definition 48, and Lemma 2, we have that

$$\Delta\left(T(X_1, \ldots, X_t) \ ; \ Y \mid T(X'_1, \ldots, X'_t)\right) \leq \varepsilon \ ,$$

where Y is uniformly random in $\{0,1\}^k$, and independent of $X_1, \ldots, X_t, X'_1, \ldots, X'_t$. This implies that for the transformation T, we have $(\mathcal{FOR}_n^t \to \mathsf{NM}_k, \varepsilon)$ (In fact, for this we don't even need the identity function in the family NM_k).

The result then follows from Observation 1. □

Now, we mention the result obtained by an independent work by Chattopadhyay and Zuckerman [CZ14] that gives an efficient (and efficiently invertible) construction of a 9-source strong non-malleable extractor. The property that the extractor is strong is proved in Appendix B of [CZ14].

Theorem 50 ([CZ14]). *There exists an efficient and efficiently invertible* 9 *source* $2^{-k-\Omega(k)}$*-strong non-malleable extractor* $\mathsf{nmExt} : (\{0,1\}^n)^t \mapsto \{0,1\}^k$ *with* $n = O(k)$*.*

Using Lemma 49, this implies Theorem 23.