Rate One-Third Non-malleable Codes[*]

Divesh Aggarwal
Department of Computer Science and Center for Quantum Technologies, National University of Singapore
Singapore
dcsdiva@nus.edu.sg

Bhavana Kanukurthi
Department of Computer Science and Automation, Indian Institute of Science, Bangalore
India
bhavana@iisc.ac.in

Sai Lakshmi Bhavana Obbattu
Microsoft Research
India
oslbhavana@gmail.com

Maciej Obremski
Center for Quantum Technologies, National University of Singapore
Singapore
obremski.math@gmail.com

Sruthi Sekar[†]
Department of Computer Science, University of California, Berkeley
USA
sruthi.sekar1@gmail.com

ABSTRACT

At ITCS 2010, Dziembowski, Pietrzak, and Wichs introduced Non-malleable Codes (NMCs) which protect against tampering of a codeword of a given message into the codeword of a related message. A well-studied model of tampering is the 2-split-state model where the codeword consists of two independently tamperable states. As with standard error-correcting codes, it is of great importance to build codes with high rates.

Following a long line of work, Aggarwal and Obremski (FOCS 2020) showed the first constant rate non-malleable code in the 2−split state model; however, this constant was a minuscule 10^{-6}! In this work, we build a Non-malleable Code with rate 1/3. This nearly matches the rate 1/2 lower bound for this model due to Cheraghchi and Guruswami (ITCS 2014). Our construction is simple, requiring just an inner-product extractor, a seeded extractor, and an affine-evasive function.

CCS CONCEPTS

• **Theory of computation → Pseudorandomness and derandomization**.

KEYWORDS

Non-malleable Codes, Randomness Extractors.

ACM Reference Format:
Divesh Aggarwal, Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, Maciej Obremski, and Sruthi Sekar. 2022. Rate One-Third Non-malleable Codes. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC '22), June 20–24, 2022, Rome, Italy*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3519935.3519972

[*]All authors have contributed equally.
[†]Work done while at Indian Institute of Science, Bangalore.

1 INTRODUCTION

The security of cryptographic primitives, oftentimes, hold only under idealized assumptions. If we take the case of encryption or digital signatures (or most other primitives), the definition of security follows a standard template: the adversary gets to observe input-output behaviour and then needs to attack the primitive. In most cases, the implicit assumption is that the adversary gets to learn nothing other than this legitimate input-output behaviour. There are many instances where such an idealized setting is far from reality. For instance, there is a powerful class of attacks where the adversary could tamper with the secret keys of the system and observe input-output behaviour on these tampered keys. Such attacks can completely break the security of the system [10, 14, 45].

Motivated by the need to protect data against such tampering attacks, Dziembowski, Pietrzak and Wichs, in their pioneering work, introduced "Non-malleable Codes" (NMCs). Non-malleability is a widely studied notion in cryptography and strives to defend against *related* tampering. Non-malleable codes, in specific, are coding schemes where the adversary cannot tamper the codeword into the codeword of a related message. Informally, NMCs provide the following guarantee: given a codeword C of a message m and a function f chosen from a tampering family $\mathcal{F}$, decoding $f(C)$ gives something that is either equal to m or completely independent of m. This becomes extremely relevant when NMCs are used to store the secret key. NMCs assure us that the encoding of the secret key may be tampered with, but the tampered codeword decodes to a message that is independent of the original secret key. A consequence of this for the above-mentioned scenario is that observing input-output behaviour on this tampered key is now rendered useless and the cryptosystem continues to remain secure.

As observed in [32], it is impossible to build NMCs secure against unrestricted tampering i.e., when $\mathcal{F}$ corresponds to the family of all functions. (To see this, simply consider the function $f(c) = \mathsf{Enc}(\mathsf{Dec}(c) + 1)$.) NMCs are therefore built with respect to specific classes of functions. A well-studied class of tampering functions is the 2-split-state model where the codeword consists of two states L and R, and the adversary tampers with each of these states independently. This is a very natural model and indeed such NMCs have found many applications in securing against physical (leakage and tampering) attacks [32, 46], domain extension of encryption schemes [23, 24], non-malleable commitments [36], non-malleable secret sharing [4, 12, 35, 47] and privacy amplification [18].

As with standard error correcting codes, the most important parameter for a non-malleable code is its rate = $\frac{\kappa}{n}$ where κ denotes the message length and n the codeword length. Cheraghchi and Guruswami [21] showed that the optimal achievable rate for the 2-split-state family is $\frac{1}{2}$. The quest for this holy grail has inspired a long line of fascinating research, introduced new pseudo-random objects and intricate proof techniques. Most recently, Aggarwal and Obremski [9] constructed a constant rate 2-split-state NMC (with negligible error). While asymptotically significant, they achieve a constant rate of only around 1/1, 500, 000. Thus, there is an embarrassingly large gap between the best achievable rate (1/2) and what we currently know (1/1, 500, 000)! *In this work, we build 2-split state non-malleable codes with a near-optimal rate $\frac{1}{3}$.*

1.1 Our Results

We obtain our construction of a rate-$\frac{1}{3}$ NMC via rate boosters which compile low rate NMCs into high rate ones. Using similar, but simpler, techniques we also obtain rate boosters for related primitives called "non-malleable 2-source extractors" (nm2Ext). We start by explaining our results on nm2Ext before proceeding to describe our results on NMCs.

Towards the goal of constructing non-malleable codes, Cheraghchi and Guruswami [22] introduced Non-malleable Extractors as a stronger primitive that immediately yields efficient non-malleable codes as long as the preimage of the extractor is efficiently samplable. Informally, a non-malleable two-source extractor nm2Ext guarantees that for any independent random sources X, Y, and any functions f, g with at least one of them having no fixed points, $\mathsf{nm2Ext}(X, Y)$ is indistinguishable from uniform even given $\mathsf{nm2Ext}(f(X), g(Y))$.[1] It is easy to see that a non-malleable two-source extractor gives non-malleability for a uniformly random message (average-case security) while a non-malleable code achieves non-malleability for every message (worst-case security). A non-malleable two-source extractor can be transformed into a non-malleable code (Enc, Dec) by setting[2] $\mathsf{Enc}(m) := \mathsf{nm2Ext}^{-1}(m)$, and $\mathsf{Dec}(x, y) := \mathsf{nm2Ext}(x, y)$. Non-malleable 2−source extractors have been the focus of intense study, with several recent results focusing on improving their rate[3].

Rate Boosters for Non-malleable Extractors. In this work, we give a rate booster for unbalanced non-malleable two-source extractors, with one source being much smaller than the other:

Informal Theorem 1. If $\mathsf{nm2Ext} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^d$ *is a strong non-malleable two-source extractor, with* $n_2 = o(n_1)$*, and* $\mathsf{Ext} : \{0,1\}^{n_1} \times \{0,1\}^d \to \{0,1\}^{\frac{n_1}{2}}$ *is a strong seeded extractor, then* $\mathsf{nm2Ext}'(x, y) := \mathsf{Ext}(x; \mathsf{nm2Ext}(x, y))$ *is a non-malleable extractor with rate* 1/2.

[1] We say that the extractor is a strong non-malleable two-source extractor if for any independent random sources X, Y, and any functions f, g with at least one of them having no fixed points, $\mathsf{nm2Ext}(X, Y)$ is indistinguishable from uniform even given $\mathsf{nm2Ext}(f(X), g(Y))$ and Y.

[2] Here, the inverse function $\mathsf{nm2Ext}^{-1}(.)$, on input m, picks and outputs x, y from the source distributions X and Y, conditioned on $\mathsf{nm2Ext}(X, Y) = m$, respectively.

[3] The rate of a non-malleable 2-source extractor is defined to be the ratio of the output length to the sum of the sizes of its two sources.

NME to NMC: Limitations. We note that the transformation from nm2Ext to NMCs requires arguing worst-case security from average-case security, which incurs a factor $2^{|\text{message size}|}$ penalty in the security parameter. Most results on building 2-split-state NMCs have focused on improving the rate of non-malleable two-source extractors and relied on this *lossy* transformation to build NMCs. We argue that this may not help.

Note that the output of a seeded extractor, with seed length d, is at best $2^{-d/2}$-close to uniform. So we cannot use the transformation from non-malleable two-source extractors to non-malleable codes mentioned above, since our construction is unable to handle the penalty $2^{|\text{message size}|} = 2^{n_1/2}$ in the error. (For the error term to even be just less than 1, $d > n_1/2$ i.e., a setting of parameters where the transformation above is useless.) We observe that this is not just a shortcoming of our construction – any construction of a non-malleable two-source extractor with rate greater than 1/5 can likely not be transformed to a non-malleable code via the above average to worst-case transformation. To see this, we note that the existential construction via the probabilistic method (which is believed to have optimal parameters) of an ε-non-malleable two-source extractor given in [22] (Theorem 5.10) requires $\varepsilon^3 \cdot 2^{2n} > 2^{2\kappa}$, where κ is the message length. This implies that $\varepsilon > 2^{(2\kappa-2n)/3}$. To obtain a non-trivial construction of a non-malleable code, we require $\varepsilon < 2^{-\kappa}$ which gives $2n > 5\kappa$, i.e., the rate of the non-malleable two-source extractor is less than 1/5.

Our Non-malleable Code. With this intuition, we deviate entirely from the approach of building NMCs via Non-malleable 2-source Extractors. We significantly modify the transformation (in Informal Theorem 1) and introduce techniques that allow us to circumvent the average to worst-case penalty to obtain a rate booster directly for non-malleable codes.

Informal Theorem 2 (Main Result). There exists an efficient, information - theoretically secure ε-right-augmented[4] non-malleable code in the 2-split-state model with rate 1/3.
We show two instantiations of the scheme: the first gives us a strikingly simple construction (as we describe in Section 1.2) and achieves an error of $2^{-\Omega(\kappa^{1/5}/\mathrm{polylog}(\kappa))}$; the second instantiation loses out on the simplicity but achieves an error of $\varepsilon = 2^{-\Omega(\frac{\kappa}{\log^3 \kappa})}$, where κ is the size of the message.

In the full version of our paper, we also show an application of our rate-1/3 non-malleable codes to get the first non-malleable commitment scheme with computational hiding and statistical hiding, achieving a communication cost of approximately 41 times the length of the message being committed to.

1.2 Technical Overview

In order to highlight the challenges of building high-rate, 2-split-state NMCs, we start by recalling a related primitive called Non-malleable Randomness Encoders introduced by Kanukurthi, Obbattu and Sekar [42]. Similar to a 2-split-state NMC, a 2-split-state NMRE consists of two independently tamperable states L and R.

[4] Right-augmented property guarantees that the right state of the NMC is simulatable independent of the message, along with the tampered message.

Contrary to an NMC, where the encoder encodes arbitrary messages, an NMRE's encoder outputs L and R such that they decode to a random string, and herein lies all the difference: while the problem of building high-rate NMCs has eluded researchers for over a decade, we know how to build NMREs with rate $\frac{1}{2}$. At the same time, we emphasize that obtaining a high-rate NMC (instead of an NMRE) is critical for many applications (such as non-malleable commitments.)

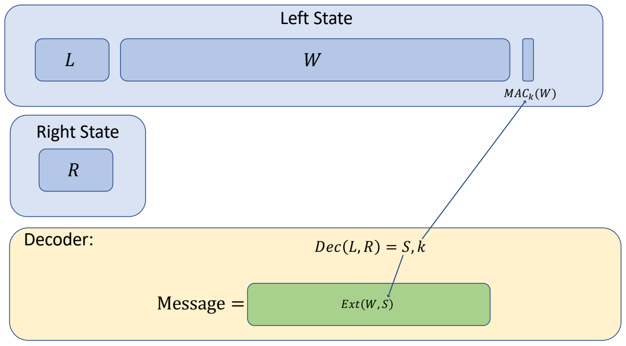

Figure 1: Construction of the non-malleable randomness encoder by [42].

Informally, the NMRE of [42] (see Figure 1.) picks the source w and seed s to a strong seeded extractor (Ext) as well as a key k_w to a message authentication code (MAC). The code consists of two states, left: $\ell\|w\|\sigma_w$, and right: r, where ℓ, r are a low-rate non-malleable encoding of s, k_w and σ_w is a tag evaluated on w with k_w as the key. The codeword, if valid, decodes to $\mathsf{Ext}(w;s)$. The high-level idea behind security is that if the codeword was tampered and leads to a different $\underline{s}$, then it is independent of S and extractor security applies. This suffices to argue independence of the tampered message $\underline{m}$ from m. (In the case where $\underline{s}, \underline{k_w} = s, k_w$, MAC security comes into play.)

In order to extend this construction to encode an arbitrary message m, one option would be to reverse sample w and s such that $\mathsf{Ext}(w;s) = m$. Unfortunately, this won't work because, on the one hand, we require the seed s to be short (as it is encoded using a poor-rate NMC) and, on the other hand, given a source w, there will be at most $2^{|s|}$ possible messages that could have been encoded. In other words, to obtain any meaningful security, s needs to be as long as the message. However, if s is long, the above approach will not yield rate gains. These orthogonal constraints are the main stumbling block that we overcome in this work. To do so, we need to somehow use $\mathsf{Ext}(w;s)$ to "mask" the message m. Clearly, since $\mathsf{Ext}(w;s)$ needs to be stored in a state different from the one storing w, we will consider a target code which has the form $(\ell\|w\|\sigma_w)$, $(r\|c)$, where c would depend on the message as well as the extractor output $\mathsf{Ext}(w;s)$ i.e., w, s, c "shares" m. While there are many candidates for c, the solution to our puzzle lies in figuring out a c with which we can prove non-malleability. Here's the tricky part: when the adversary learns the tampered message, she learns information that depends on $\mathsf{Ext}(w;s)$. Furthermore, her tampering of r, which influences the tampered seed $\underline{s}$ and therefore $\mathsf{Ext}(\underline{w};\underline{s})$, also depends on $\mathsf{Ext}(w;s)$. In a nutshell, regardless of what c is, it is clear that leveraging the security of the strong randomness extractor to argue non-malleability is not straightforward.

Our first idea to overcome this challenge is to allow c to be such that $2\mathsf{Ext}(\mathsf{Ext}(w;s), c) = m$ where 2Ext is a 2−source extractor. As it turns out, this construction is secure for the following reason: even if the tampered output leaks some information about $\mathsf{Ext}(w;s)$, the two source extractor[5] will ensure that the "masking" remains secure. This leads to an NMC with rate $\frac{1}{9}$ in the split-state model.

The main idea behind our final construction is to tackle the information leakage on $\mathsf{Ext}(w;s)$ head-on. In particular, though the tampered message may leak information about $\mathsf{Ext}(w;s)$, we can ensure that this information is useless as far as breaking non-malleability is concerned. In fact, we do this while further simplifying our construction: We set the second half of the right state to be $(c = \mathsf{Ext}(w;s) \oplus m)\|\sigma_c$ where σ_c is a MAC tag of c evaluated on key k_c. In retrospect, our encoding scheme is nearly identical to that due to Kanukurthi, Obbattu and Sekar [41]. While [41] gave a four-state construction, we merge states to obtain a two-state construction.

We now offer a more detailed overview of the construction as well as the proof. Our construction uses the following building blocks:

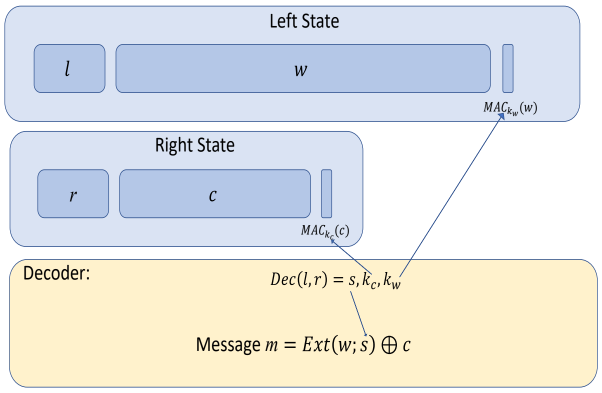

Figure 2: Construction Overview. ℓ, r come from augmented non-malleable code. The encoder proceeds in steps: first, we randomly sample s, k_w, k_c, w (all independently of the message we are encoding), then we encode s, k_w, k_c using NMC into ℓ, r. We then set $c = \mathsf{Ext}(w;s) \oplus m$, and evaluate σ_c as a MAC tag of c on key k_c, and σ_w as MAC tag of w on key k_w.

a message authentication code, a strong seeded extractor, and a low-rate non-malleable code which we shall use to encode the keys of the message authentication code and the seed for the seeded extractor. The overview of the construction can be found in Figure 2. We will use the notation from the above-mentioned figure. Also, for a variable X, $\underline{X}$ will denote its tampering. We proceed with a slightly simplified sketch of the proof.

Proof Overview. In order to prove non-malleability, we need to demonstrate the existence of a simulator whose outputs is indistinguishable from the output of the tampering experiment. The simulator doesn't use the message; however, it outputs a special symbol to indicate that the tampered message is unchanged. The

[5]A two-source extractor takes two entropic sources and outputs perfect randomness.

simulator's output is run through a special wrapper function (typically called "Copy" function) that, in this case, outputs the original message.

Our proof proceeds by partitioning our codeword space. Our simulator will pick a codeword at random, check which partition it lies in to determine its output. We describe the partitions below:

- $\mathcal{P}_1 = \{(\ell, w, \sigma_w, r, c, \sigma_c) : (w, \sigma_w, c, \sigma_c) = (\underline{w}, \underline{\sigma_w}, \underline{c}, \underline{\sigma_c}) \wedge \mathsf{NMDec}(\ell, r) = \mathsf{NMDec}(\underline{\ell}, \underline{r})\}$. This partition captures the scenario when, even after tampering, the inner codeword (ℓ, r) decodes to the same message, and W, σ_w, C, σ_c remain unchanged. In this case, the final codeword must decode to the same message.
- $\mathcal{P}_2 = \{(\ell, w, \sigma_w, r, c, \sigma_c) : (w, \sigma_w, c, \sigma_c) \neq (\underline{w}, \underline{\sigma_w}, \underline{c}, \underline{\sigma_c}) \wedge \mathsf{NMDec}(\ell, r) = \mathsf{NMDec}(\underline{\ell}, \underline{r})\}$. $\mathcal{P}_2$ captures the scenario when, the decoding of the inner code remains unchanged after tampering, while one of the pairs (W, σ_w) or (C, σ_c) are changed. We will show that if this event occurs then, using the security of MACs, the tampering is detected with overwhelming probability.
- $\mathcal{P}_3 = \{(\ell, w, \sigma_w, r, c, \sigma_c) : \mathsf{NMDec}(\ell, r) \neq \mathsf{NMDec}(\underline{\ell}, \underline{r})\}$. $\mathcal{P}_3$ captures the scenario that the inner code is non-trivially tampered and does not decode to $\mathsf{NMDec}(L, R)$. In this case, we will show that the tampered codeword is independent of the original message m.

The simulator generates the codeword $((L, W, \sigma_w), (R, \widetilde{C}, \widetilde{\sigma}_c))$ of a random message. If this simulated codeword is in $\mathcal{P}_1$, it outputs same*. Recall that the wrapper function will then output the original message. If the simulated codeword is in $\mathcal{P}_2$, the simulator outputs $\perp$, else the simulator outputs Dec $\left((L, W, \sigma_w), (R, \widetilde{C}, \widetilde{\sigma}_c)\right)$. (Note that our code is right-augmented i.e., it satisfies a stronger notion of security where the right state of the codeword can be revealed without breaking non-malleability.)

To prove non-malleability, we need to show that this behaviour of the simulator is indistinguishable from that of the tampering experiment. To do this, we first show that the probability of a codeword being in any given partition is independent of the message[6]. Next, we show that the output of the tampering experiment is, in each case, indistinguishable from the simulator's output.

For the case where the codeword is in partition $\mathcal{P}_1$, it is clear that the simulator output is identical to that of the tampering experiment. We, therefore, focus on the other two cases.

1.2.1 *Case 1: Codeword is in $\mathcal{P}_2$ i.e.,* $\mathsf{NMDec}(\underline{L}, \underline{R}) = \mathsf{NMDec}(L, R)$.

Intuitively, we would like to argue that the tag keys K_w, K_c will remain securely hidden from the adversary, and if he decides to tamper with W or C he will not be able to fake tags σ_w, σ_c. Thus either the whole codeword remains untampered (in which case, we are in $\mathcal{P}_1$) or the new codeword will not be valid.
The standard approach would be to argue that if $\Pr(\mathsf{NMDec}(\underline{L}, \underline{R}) = \mathsf{NMDec}(L, R))$ is not too small then

$\Pr(\text{tampered codeword is valid} \wedge (\underline{W}, \underline{C}) \neq (W, C) \mid \mathsf{NMDec}(\underline{L}, \underline{R}) = \mathsf{NMDec}(L, R))$ is negligible.

However we have to be delicate here. For example if adversary wants to tamper with W he has access to L and knows that $\mathsf{NMDec}(\underline{L}, \underline{R}) = \mathsf{NMDec}(L, R)$. This reveals some information about R and thus adversary potentially might get hold of some partial information about the encoded data (and K_w, K_c in particular). This is why it is actually easier to directly argue that

$$\Pr(\text{tampered codeword is valid } \wedge (\underline{W}, \underline{C}) \neq (W, C) \wedge \mathsf{NMDec}(\underline{L}, \underline{R}) = \mathsf{NMDec}(L, R)) \quad (1)$$

is negligible. Notice that the codeword will not be valid in only one of three cases: if $\mathsf{NMDec}(\underline{L}, \underline{R}) = \perp$ or if one of the MACs on W or C does not verify correctly. Since $\mathsf{NMDec}(\underline{L}, \underline{R}) = \mathsf{NMDec}(L, R)$ we know that the only options left are the failures to verify MACs. Moreover we know that $(\underline{K_w}, \underline{K_c}) = (K_w, K_c)$, thus Inequality 1 can be rewritten:

$$\Pr(\mathsf{Vrfy}_{K_w}(\underline{W}, \underline{\sigma_w}) = \mathsf{Vrfy}_{K_c}(\underline{C}, \underline{\sigma_c}) = 1 \wedge (\underline{W}, \underline{C}) \neq (W, C) \wedge \mathsf{NMDec}(\underline{L}, \underline{R}) = \mathsf{NMDec}(L, R)) \quad (2)$$

is negligible. Now we can upper-bound the term in the Inequality 2 by the following

$$\Pr(\mathsf{Vrfy}_{K_w}(\underline{W}, \underline{\sigma_w}) = \mathsf{Vrfy}_{K_c}(\underline{C}, \underline{\sigma_c}) = 1 \wedge (\underline{W}, \underline{C}) \neq (W, C)).$$

Which by the union bound can be upper-bounded with

$$\Pr(\mathsf{Vrfy}_{K_w}(\underline{W}, \underline{\sigma_w}) = 1 \wedge \underline{W} \neq W) + \Pr(\mathsf{Vrfy}_{K_c}(\underline{C}, \underline{\sigma_c}) = 1 \wedge \underline{C} \neq C).$$

Finally, we can argue that each of the elements of the sum is negligible. Notice that when tampering with W adversary has access to L but that can not reveal any information about K_w since every non-malleable code is a secret sharing scheme. The rest follows from the security of MACs.

1.2.2 *Case 2: Codeword is in $\mathcal{P}_3$ i.e.,* $\mathsf{NMDec}(\underline{L}, \underline{R}) \neq \mathsf{NMDec}(L, R)$.

In this case, we will follow the adventures of the seed S; the MACs and keys do not play any role here. In fact, for the purposes of this proof sketch, we will ignore the MAC keys and tags. We will also assume that this case (i.e., codeword $\in \mathcal{P}_3$) occurs with substantial probability (else we don't have to worry about it). In such a case, we will argue that the final message is independent of the original message.

To provide a proof sketch for this case, we consider the following modified construction: Let $(\ell, r) \leftarrow \mathsf{NMEnc}(s)$, where $(\mathsf{NMEnc}, \mathsf{NMDec})$ is the low-rate non-malleable code. Pick a random W, the source for a strong extractor. Then the left state of the final non-malleable code is $\ell \| w$ (where w is sampled from W). The right state is $r \| (c = \mathsf{Ext}(w; s))$. In other words, we've simply removed the MAC keys and tags. While this is obviously not a secure non-malleable code, we use this construction for the sake of ease of exposition.

By the properties of the augmented NMC[7] we know that there is a simulator NMSim that outputs $S^{\mathsf{sim}}, L^{\mathsf{sim}}$. Given that we are in the case where $\mathsf{NMDec}(\underline{L}, \underline{R}) \neq \mathsf{NMDec}(L, R)$, S^{sim} is independent of S. As discussed earlier, this knowledge alone doesn't suffice for us. The tampering of R (which influences $\underline{S}$) depends on C which in turn uses the extractor output. In order to output the tampered message, we need to be able to compute $\mathsf{Ext}(\underline{W}; \underline{S})$. We handle this by using a Markov style argument, as we shall see shortly.

[6]This proof relies on the secret sharing property of the non-malleable code as well as the security of the strong randomness extractor.

[7]Augmented NMC is a natural extension of NMC (and equivalent of strong extraction property). We require that $\mathsf{NMDec}(\underline{L}, \underline{R})$ is equal or independent of $\mathsf{NMDec}(L, R)$ and revealing L or R doesn't ruin this independence.

Non-malleability of the underlying NMC:. To use the non-malleability of the inner code, $(\mathsf{NMEnc}, \mathsf{NMDec})$, consider the tampering functions f^*, g^* on (L, R) that sample $W, \widetilde{C}$ uniformly and then compute $f^*(L) = f_1(L, W)$ and $g^*(R) = g_1(R, \widetilde{C})$. Let $\underline{S} = \mathsf{NMDec}(f^*(L), g^*(R))$. By the augmented non-malleability of $(\mathsf{NMEnc}, \mathsf{NMDec})$, there exists a simulator NMSim such that

$$\underline{S}, L, S, W, \widetilde{C} \approx \mathsf{Copy}(\mu, \mathsf{NMSim}), S, W, \widetilde{C}\ , \tag{3}$$

where the output of NMSim depends on $W, \widetilde{C}$ and the functions f_1, g_1. By augmented non-malleability, recall that NMSim is parsed as the joint distribution $(\mathsf{NMSim}_1, L^{\mathsf{Sim}})$. We further denote NMSim_1 by S^{sim}, when NMSim_1 is conditioned to not output same^* or $\perp$. We denote by E_1 the event that $(L, W, R, \widetilde{C}) \in \mathcal{P}_3$, E_2 the event that $S \neq \underline{S}$ (Note that E_1 is the same as E_2), and E_3 the event that $\mathsf{NMSim}_1 \notin \{(\mathsf{same}^*, S)\}$. We first show that $\Pr[E_3]$ is high.

From here, using properties of statistical distance, and expanding out the definition of copy, we show that:

$$(\underline{S}, L, S, W, \widetilde{C})|_{E_2} \approx (S^{\mathsf{sim}}, L^{\mathsf{Sim}}, S, W, \widetilde{C})|_{E_3} \tag{4}$$

We next show that the (average) conditional entropy of W given $S^{\mathsf{sim}}, \underline{L}^{\mathsf{Sim}}, \mathsf{Ext}(\underline{W}; \underline{S})$ is high; here $\underline{L}^{\mathsf{Sim}}$ and $\underline{W}$ are tamperings of L^{Sim}, W and can be computed as their deterministic function. In particular, we have that

$$\widetilde{\mathbf{H}}_\infty\left(W | S^{\mathsf{sim}}, \underline{L}^{\mathsf{Sim}}, \mathsf{Ext}(\underline{W}; S^{\mathsf{sim}})\right)$$

is high (as $(S^{\mathsf{sim}}, \underline{L}^{\mathsf{Sim}}, \mathsf{Ext}(\underline{W}; S^{\mathsf{sim}}))$ are "small enough").

Further, since S is independent of S^{sim}, we use the strong extractor property of Ext, to show the following:

$$\mathsf{Ext}(W; S) \approx U | S, L^{\mathsf{Sim}}, \widetilde{C}, \underline{L}^{\mathsf{Sim}}, \mathsf{Ext}(\underline{W}; S^{\mathsf{sim}}), S^{\mathsf{sim}} \tag{5}$$

In order to proceed with the proof, we will prove a statement similar to the one above; except that instead of the tampered seed being simulated, we will let it be the tampered seed (in the underlying non-malleable code). (Eventually we would need to replace $\widetilde{C}$ by $C := \mathsf{Ext}(W; S) \oplus m$.)
For the remainder of the proof, let $L, W, R, \widetilde{C}$ be distributed as $L, W, R, \widetilde{C}|_{E_1}$. Also, let $\underline{S}, \underline{L}, \underline{R}, \underline{W}$ be the corresponding tampered random variables, after tampering conditioned on the event that $(L, W, R, \widetilde{C}) \in \mathcal{P}_3$. For ease of notation, in the remainder of the exposition below, we skip writing the conditioning on $E_1/E_2/E_3$ (but this exists throughout). By inequality 4 and using simple property of statistical distance, we can get the following inequalities corresponding to the LHS and RHS of inequality 5 respectively:

$$\mathsf{Ext}(W; S), S, L^{\mathsf{Sim}}, \widetilde{C}, \underline{L}^{\mathsf{Sim}}, \mathsf{Ext}(\underline{W}; S^{\mathsf{sim}}), S^{\mathsf{sim}} \approx \mathsf{Ext}(W; S), S, L, \widetilde{C}, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S} \tag{6}$$

$$\mathcal{U}, S, L^{\mathsf{Sim}}, \widetilde{C}, \underline{L}^{\mathsf{Sim}}, \mathsf{Ext}(\underline{W}; S^{\mathsf{sim}}), S^{\mathsf{sim}} \approx \mathcal{U}, S, L, \widetilde{C}, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S} \tag{7}$$

Applying triangle inequality on Inequalities (5), (6), (7), we get that

$$\mathsf{Ext}(W; S) \approx U | S, L, \widetilde{C}, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S} \tag{8}$$

Note that in the equation above, there is no dependence on m on either side as $\widetilde{C}$ is independent of m. Ultimately, we would like to say that the output of the tampering experiment is indistinguishable from the simulated output. We accomplish this in three steps:

1. Adding R. In equation 8, the only information correlated to W and R is $\underline{S}$. Since $\mathsf{Ext}(W; S) \approx \mathcal{U}$ even given $\underline{S}$, we can safely add R to Equation 8.

$$\mathsf{Ext}(W; S) \approx U | S, L, \widetilde{C}, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, R, g_1(R, \widetilde{C})\ .$$

From here, we would ideally like to drop $\widetilde{C}$ and somehow bring back the dependence on m via C. For now, we drop $\widetilde{C}$

$$\mathsf{Ext}(W; S) \approx U | S, L, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, R, \underline{R}\ . \tag{9}$$

The way we'll bring C is to condition $\widetilde{C}$ on being a "cipher of m". For that, we first need to prove that $\widetilde{C}$ is independent of W given appropriate auxiliary information.

2. Capturing $\widetilde{C}$'s correlation with W. In this step, we will prove that $\widetilde{C}$ is independent of W given $S, L, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, R, \underline{R}$. We first observe that $\widetilde{C}$ is independent of W given (L, R, S). Now we would like to add the other random variables in the auxiliary information. We use a Lemma due to Dziembowski and Pietrzak [31, Lemma 4] which states that independence in the presence of additional auxiliary information is indeed possible, provided it satisfies a few properties:

- The auxiliary information may be computed in multiple steps.
- Computation in all of the steps can use (L, R, S) and the part of the auxiliary information generated in previous steps.
- Computation in a given step can either depend on $\widetilde{C}$ or W but not both.

By computing auxiliary information in the order $\underline{L}$ followed by $\underline{R}$ followed by $\mathsf{Ext}(\underline{W}; \underline{S})$, we can easily prove that $\widetilde{C}$ is independent of W given $S, L, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, R, \underline{R}$.

3. Conditioning $\widetilde{C}$ appropriately. Since W is independent of $\widetilde{C}$ given appropriate auxilliary information, in Equation 9, we can condition $\widetilde{C}$ to either be $m \oplus \mathsf{Ext}(W, S)$ or $m \oplus U$. (Note that the former is identical to C.) By doing so, Equation 9 will lead to the following $C, S, L, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, R, \underline{R} \approx \mathcal{U}, S, L, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, R, \underline{R}$, where $\underline{R}, \underline{S}$ are appropriately computed.
The desired result follows by observing that the tampered codeword is a function of

$$\underline{L}, \underline{R}, \mathsf{Ext}(\underline{W}; \underline{S}), C, R\ .$$

Putting it Together. So far, we've described the simulator and sketched the proof for showing that the simulated output is indistinguishable from the tampered output in each of the cases. To complete the proof, we need to combine all three cases and, in particular, the probability that the codewords (tampered vs simulated) lie in each of the partitions needs to be analysed.

Candidate Instatiation. While we can turn any augmented non-malleable code (or randomness encoder) into a good rate non-malleable code, a very simple result can be obtained using [6]. To encode a message m all we will need is *an affine evasive function h*. It is a function $h : \mathbb{F}_p \to \mathcal{M} \cup \perp$ such that $\Pr(h(aU+b) \neq \perp \mid h(U) = m)$ is negligible for all a, b, m, and $U | h(U) = m$ should be efficiently samplable, the construction of the said function can be found in [1, 6]. The encoding procedure is described in Figure 3.

Short and Simple: The Encoding Procedure:

(1) Sample s, k_w, k_c, w uniformly at random.

(2) Sample x uniformly random, such that $h(x) = s, k_w, k_c$.

(3) Sample $\ell, r \in \mathbb{F}_p^n$ uniformly random, such that $\langle \ell, r\rangle = x$.

(4) Evaluate $c = \mathsf{Ext}(w; s) \oplus m$.

(5) Calculate MACs $\sigma_c = \mathsf{Tag}_{k_c}(c)$ and $\sigma_w = \mathsf{Tag}'_{k_w}(w)$.

The final output is: on the left: ℓ, w, σ_w, and on the right: r, c, σ_c.

Figure 3: Simple non-malleable code with a great rate. Here h is an affine evasive function. The decoding procedure is analogous: the decoder inverts Steps 3 and 2, obtains keys k_w, k_c, verifies MACs from the Step 5 and proceeds to obtain the message via the Step 4. If in Step 2 the function h outputs $\perp$, then the decoder aborts and outputs $\perp$.

1.3 Related Work

We now sketch the landscape of this area and particularly summarize the results on 2-split-state NMCs in Table 1. In [32], in addition to introducing non-malleable codes, Dziembowski *et al.* also introduced a model of tampering called the t-split-state model, where the codeword consists of t independently tamperable states. They give the first NMC constructions in the n-split-state model (where n is the codeword length), with rate ≈ 0.19, and the 2-split-state model (using random oracles). Dziembowski, Kazana and Obremski [30] provided the first construction of 2-split-state NMCs without any assumptions. Their construction enabled encoding of 1-bit messages and used two-source extractors. The first NMC in the 2-split-state model for k-bit messages was given by Aggarwal, Dodis and Lovett [6], which used inner product extractors with tools from additive combinatorics. In [21], Cheraghchi and Guruswami brought focus to the rate $= \frac{\text{message length}}{\text{codeword length}}$ of non-malleable codes. In particular, they showed that the optimal achievable rate for the t-split-state family is $1 - 1/t$. Note that in the split-state tampering model, having as few states is most desirable, with 2 states being the best achievable. By the above result, the best possible rate for the 2-split-state model is therefore $\frac{1}{2}$. A long series of works[8] [5, 9, 17, 20, 22, 37, 41–44] has made partial progress towards achieving these parameters. We now discuss some of these results. The work of Cheraghchi and Guruswami [22] gave the first optimal rate non-malleable code in the n-split-state model (where n is the codeword length). More importantly, this work introduced non-malleable two-source extractors and demonstrated that these special extractors can be used to generically build 2-split-state NMCs. This connection has led to several fascinating works [17, 20, 43, 44] striving to improve the rate and number of states of non-malleable codes. Most notably, Chattopadhyay and Zuckerman [20] built a 10-state NMC with a constant rate, making this the first constant rate construction with a sublinear number of states. They achieve their result by first building a non-malleable extractor with 10 sources and then using the connection due to [22]. Aggarwal, Dodis, Kazana and Obremski [5] introduced the concept of non-malleable reductions – which would later be used to build constant rate NMCs [9]. The work of Kanukurthi, Obbattu and Sekar [41] used seeded extractors to build a compiler that transforms a low rate non-malleable code into one with high rate and, in particular, obtained a rate 1/3, 4−state non-malleable code. This was subsequently improved to three states in the works of Kanukurthi, Obbattu and Sekar [42] as well as Gupta, Maji and Wang [37]. Li [44] obtained 2-split-state NMC with rate $O(\frac{\log\log\log(1/\epsilon)}{\log\log(1/\epsilon)})$ (where ϵ is the error). Particularly, this gave a rate of $O(\log\log(n)/\log(n))$, for negligible error $\epsilon = 2^{-\Omega(n)}$, and a constant rate for constant error, making this the first constant rate scheme in the 2−split-state model. The most recent work of Aggarwal and Obremski [9] relied on the concept of non-malleable reductions and built the first constant rate 2-split-state NMC with negligible error.

Table 1: Prior Work on 2-state NMCs (n is codeword length)

Work	Rate
[32]	1/6 (Existential, Random Oracle Model)
[21]	1/2 (Existential, Lower bound)
[30]	$\Omega(1/n)$ (Only for 1-bit messages)
[1, 3, 6]	$\Omega(1/n^{4/5})$
[5]	$n^{-\Omega(1)}$
[17]	$n^{-\Omega(1)}$
[43]	$\Omega(1/\log(n))$
[44]	$\Omega(\log\log(n)/\log(n))$
[44]	$\Omega(1)$ (with constant error)
[9]	$\approx 1/1,500,000$
Our Result	1/3

1.4 Organization of the Paper

We describe the preliminary building blocks in Section 2. We then describe our rate boosters for the non-malleable codes in Section 3. Finally, we describe our rate boosters for the non-malleable randomness extractors in Section 4.

2 PRELIMINARIES

2.1 Notation

We begin by describing some notations which we use. We use $s \in_R S$ to denote that s is sampled uniformly from the set S, and $x \leftarrow X$ to denote that x is sampled from the probability distribution X. The concatenation of two binary strings x and y is denoted

[8]Other works have considered non-malleable codes in models other than the 2-split-state model or under computational assumptions [2, 8, 11, 34, 39], [5, 13, 15, 16, 19, 25, 26, 29, 33, 37], of which, rate optimizing compilers have been built for some (weaker) tampering models like the bitwise and permutation tampering [11], and the local tampering [37], or in the computational setting [2].

by $x\|y$. We denote the length of a binary string x by $|x|$, and the cardinality of any set S by $|S|$. For any set S, U_S denotes the uniform distribution on S, and we use the shorthand U_ℓ to represent $U_{\{0,1\}^\ell}$, for any integer $\ell > 0$. All logarithms are over base 2. The set $X \backslash Y \overset{\text{def}}{=} \{x : x \in X, x \notin Y\}$, is the set of elements in X that are not in Y.

For any event E, and any random variable X_1, $X_1|_E$ denotes the distribution of the random variable X_1 conditioned on the event E. For any random variables X_1, X_2 and an event E, $(X_2, X_1|_{E,X_2})$ denotes the distribution where we sample X_2 according to its marginal distribution, and then sample X_1 conditioned on the choice of X_2 and the event E.

2.2 Statistical Distance and Entropy

Let X_1, X_2 be two probability distributions over some set S. Their statistical distance is

$$\Delta(X_1; X_2) \overset{\text{def}}{=} \max_{T \subseteq S}\{\Pr[X_1 \in T] - \Pr[X_2 \in T]\} = \frac{1}{2}\sum_{s \in S}\left|\Pr_{X_1}[s] - \Pr_{X_2}[s]\right|$$

(they are said to be ε-close if $\Delta(X_1; X_2) \le \varepsilon$ and denoted by $X_1 \approx_\varepsilon X_2$). We shorthand $\Delta(X_1, Y; X_2, Y)$ by $\Delta(X_1; X_2|Y)$ for any random variables X_1, X_2, Y. The *min-entropy* of a random variable W is $\mathbf{H}_\infty(W) = -\log(\max_w \Pr[W = w])$. For a joint distribution (W, E), the (average) conditional min-entropy of W given E is defined in [28] as

$$\widetilde{\mathbf{H}}_\infty(W \mid E) = -\log(\mathop{\mathbf{E}}_{e \leftarrow E}(\max_w \Pr[W = w | E = e]))$$

(here the expectation is taken over e for which $\Pr[E = e]$ is nonzero). For a random variable W over $\{0,1\}^n$, $W|E$ is said to be an $(n,t)-$source if $\widetilde{\mathbf{H}}_\infty(W|E) \ge t$. We require the following lemma about conditional min-entropy.

Lemma 2.1. *[28] If B has at most 2^λ possible values, then $\widetilde{\mathbf{H}}_\infty(A \mid B) \ge \mathbf{H}_\infty(A,B) - \lambda \ge \mathbf{H}_\infty(A) - \lambda$, and, more generally, $\widetilde{\mathbf{H}}_\infty(A \mid B, C) \ge \widetilde{\mathbf{H}}_\infty(A, B \mid C) - \lambda \ge \widetilde{\mathbf{H}}_\infty(A \mid C) - \lambda$.*

Lemma 2.2. *For any random variables A, B, if $A \approx_\epsilon B$, then for any (possibly randomized) function f, $f(A) \approx_\epsilon f(B)$.*

We require the following lemma from [6], which states that if the pairs of random variables (X_1, X_2) and (Y_1, Y_2) are statistically close, then for any set $\mathcal{A}$, the conditional random variables X_2 conditioned on the event $X_1 \in \mathcal{A}$, i.e., $X_2|_{X_1 \in \mathcal{A}}$, and Y_2 conditioned on the event $Y_1 \in \mathcal{A}$, i.e., $Y_2|_{Y_1 \in \mathcal{A}}$ are also close.

Lemma 2.3. *[6, Claim 4] Let X_1, X_2, Y_1, Y_2 be random variables such that $(X_1, X_2) \approx_\epsilon (Y_1, Y_2)$. Then, for any non-empty set $\mathcal{A}$, we have:*

$$\Delta(X_2|_{X_1 \in \mathcal{A}}; Y_2|_{Y_1 \in \mathcal{A}}) \le \frac{2\epsilon}{\Pr[X_1 \in \mathcal{A}]}\ .$$

We require the following lemmas.

Lemma 2.4. *[9] Let S be some random variable distributed over a set $\mathcal{S}$, and let $\mathcal{S}_1, \dots, \mathcal{S}_j$ be a partition of $\mathcal{S}$. Let $\phi : \mathcal{S} \to \mathcal{T}$ be some function, and let $D_1, \dots, D_j$ be some random variables over the set $\mathcal{T}$. Assume that for all $1 \le i \le j$,*

$$\Delta\left(\phi(S)|_{S \in \mathcal{S}_i}\ ;\ D_i\right) \le \varepsilon_i.$$

Then

$$\Delta\left(\phi(S)\ ;\ D\right) \le \sum \varepsilon_i \Pr[S \in S_i]\ ,$$

for some random variable $D \in \mathcal{T}$ such that for all d $\Pr[D = d] = \sum_i \Pr[S \in \mathcal{S}_i] \cdot \Pr[D_i = d]$.

The following result is from [31]. For a proof, see [7].

Lemma 2.5. *Let $A \in \mathcal{A}$ and $B \in \mathcal{B}$, and V_0 be random variables such that A is independent of B given any fixing of the random variable V_0. Let $V_1, V_2, \dots$ be random variables defined as functions of A, B satisfying the following property. For all $i \in \mathbb{N}$, if i is even then $V_i = \phi_i(V_0, V_1, \dots, V_{i-1}, A)$ and if i is odd, then $V_i = \phi_i(V_0, V_1, \dots, V_{i-1}, B)$ for some function ϕ_i. Then for all i, A is independent of B given any fixing of $V_0, V_1, \dots, V_i$.*

Further, we require the following simple lemma about the statistical distance between conditional distributions, given the distance between a pair of joint distributions.

Lemma 2.6. *For any random variables X, Y and Z and uniform random variable U, with X, Y, U defined over some binary space $\mathcal{X}$, such that Y is independent of X conditioned on Z and also independent of U conditioned on Z, if $(X, Z) \approx_\varepsilon (U, Z)$, then it implies that $(Y, Z)|(X \oplus Y = x) \approx_\varepsilon (U, Z)|(U \oplus Y = x)$, for any $x \in \mathcal{X}$.*

Proof. Assume to the contrary that there exists an unbounded adversary, $\mathcal{D}$, who can distinguish between $(Y, Z)|(X \oplus Y = x)$ and $(U, Z)|(U \oplus Y = x)$ for some $x \in \mathcal{X}$, with advantage $> \varepsilon$. Then, we claim that we can build an adversary distinguishing (X, Z) and (U, Z) with the same advantage. The reduction is simple: given (a, c) from (X, Z) or (U, Z), we sample $b \leftarrow Y|a \oplus Y = x$. Given the independence of Y from both $X|Z$ and $U|Z$, (b, c) is correctly simulated to be either $(Y, Z)|(X \oplus Y = x)$ (if (a, c) was from (X, Z)) or $(U, Z)|(U \oplus Y = x)$ (if (a, c) was from (U, Z)), for $\mathcal{D}$. Hence, the lemma is proved. □

2.3 Randomness Extractors

Definition 2.7 (Seeded Extractors). We say that a polynomial time probabilistic function $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\ell$, using d bits of randomness, is an $(n, k, d, \ell, \varepsilon)$-strong seeded extractor if for all random variables $W \in \{0,1\}^n$, E correlated with W, such that $\mathbf{H}_\infty(W|E) \ge k$, and X independent of W, E and uniform in $\{0,1\}^d$, we have $(\text{Ext}(W; X), X, E) \approx_\varepsilon (U_\ell, X, E)$, where U is uniform in $\{0,1\}^\ell$ and independent of X, E.

We require the following seeded extractor construction from [38].

Lemma 2.8. *[38] For every constant $\nu > 0$, all integers $n \ge k$ and all $\varepsilon \ge 0$, there is an explicit (efficient) $(n, k, d, \ell, \varepsilon)-$strong seeded extractor with $\ell = k - O\left(\log n + \log\frac{1}{\varepsilon}\right)$ and $d = O\left(\log n + \log\frac{1}{\varepsilon}\right)$.*

2.4 One-Time Message Authentication Codes

Our construction also requires the use of information theoretic one-time message authentication codes, which give a guarantee that, given a message-tag pair, (m, t), where t is generated using a tag generation algorithm, using a random authentication key, the probability with which an (unbounded) adversary can come up with a valid message-tag pair (m', t'), for $m' \ne m$, is negligible.

Definition 2.9. A family of functions $\{\mathsf{Tag}_{k_a} : \{0,1\}^\gamma \to \{0,1\}^\delta, \mathsf{Vrfy}_{k_a} : \{0,1\}^\gamma \times \{0,1\}^\delta \to \{0,1\}\}_{k_a \in \{0,1\}^\tau}$ is said to be a $\mu-$secure one time message authentication code if

(1) For $k_a \in_R \{0,1\}^\tau$, $\forall\ m \in \{0,1\}^\gamma$, $\Pr[\mathsf{Vrfy}_{k_a}(m, \mathsf{Tag}_{k_a}(m)) = 1] = 1$,
where for any (m,t), $\mathsf{Vrfy}_{k_a}(m,t) \overset{\text{def}}{=} \begin{cases} 1 \text{ if } \mathsf{Tag}_{k_a}(m) = t \\ 0 \text{ otherwise} \end{cases}$
(2) For any $m \neq m', t, t'$, $\Pr_{k_a}[\mathsf{Tag}_{k_a}(m) = t | \mathsf{Tag}_{k_a}(m') = t'] \leq \mu$, where $k_a \in_R \{0,1\}^\tau$.

For simplicity, we consider Tag to be a deterministic function. The following lemma guarantees that there exists efficient one-time message authentication codes where the key and tag can be much shorter than the message to be authenticated.

Lemma 2.10. *[40] For any $\gamma, \varepsilon > 0$ there is an efficient $\varepsilon-$secure one time* MAC *with $\delta \leq (\log(\gamma) + \log(\frac{1}{\varepsilon}))$, $\tau \leq 2\delta$, where τ, γ, δ are key, message, tag length respectively.*

We refer the reader to [27] for a construction satisfying these parameters.

2.5 Non-malleable Codes

Non-malleable codes (NMCs) were introduced in [32]. We now state the equivalent definition of non-malleable codes given in [22].

Definition 2.11. A (possibly randomised) function pair ($\mathsf{Enc} : \mathcal{M} \to C, \mathsf{Dec} : C \to \mathcal{M} \cup \{\perp\}$) is said to be a **coding scheme** from $\mathcal{M}$ to C if $\forall m \in \mathcal{M}, \Pr[\mathsf{Dec}(\mathsf{Enc}(m)) = m] = 1$(the probability is over the randomness of Enc, Dec). The rate of the coding scheme is given by $\frac{\log|\mathcal{M}|}{\log|C|}$.

Definition 2.12. A coding scheme (Enc, Dec) from $\mathcal{M}$ to C, is said to be ϵ- **non-malleable** with respect to a tampering function family $\mathcal{F} \subseteq \{f : C \to C\}$ if $\forall\ f \in \mathcal{F}$, there exists a distribution NMSim (specific to f) over $\mathcal{M} \cup \{\text{same}^*, \perp\}$ such that $\forall\ m \in \mathcal{M}$

$$\mathsf{Tamper}_f^m \approx_\epsilon \mathsf{Copy}(m, \mathsf{NMSim})$$

where Tamper_f^m denotes the distribution $\mathsf{Dec}(f(\mathsf{Enc}(m)))$ and $\mathsf{Copy}(m, \mathsf{NMSim})$ is defined as

$$\tilde{m} \leftarrow Sim_f$$

$$\mathsf{Copy}(m, \mathsf{NMSim}) = \begin{cases} m \text{ if } \tilde{m} = \text{same}^* \\ \tilde{m} \text{ otherwise} \end{cases}$$

NMSim should be efficiently samplable given oracle access to $f(.)$.

Split-state Tampering Family. Specifically, we consider NMCs with respect to the two split-state tampering family defined below, for message space $\{0,1\}^\alpha$ and codeword space $\{0,1\}^\beta \times \{0,1\}^\beta$, comprising two states:

$$\mathcal{F}_{\text{split}} \overset{\text{def}}{=} \{(f,g) : f, g : \{0,1\}^\beta \to \{0,1\}^\beta\}$$

Our construction requires NMCs with respect to the split-state tampering family satisfying an additional property called "augmented security", which was introduced in [2], and guarantees that in addition to the tampered message, one of the states, say L, of the codeword is also simulatable independent of the message.

We formally define NMCs against the split-state model, with augmented security below.

Definition 2.13 (Augmented Non-malleable Codes). A coding scheme (Enc, Dec) from $\{0,1\}^\alpha$ to $\{0,1\}^\beta \times \{0,1\}^\beta$ is called an ϵ-augmented non-malleable code with respect to $\mathcal{F}_{\text{split}}$, if the following holds. For any (possibly randomized) $(f,g) \in \mathcal{F}_{\text{split}}$, let $\mathsf{Tamper}_{f,g}^m$ denote the distribution $\mathsf{Dec}(f(L), g(R))$, for $(L,R) \leftarrow \mathsf{Enc}(m)$. There exists a simulator that, given oracle access to $(f,g)(.)$, outputs $\mathsf{NMSim} = (\mathsf{NMSim}_1, \mathsf{NMSim}_2)$ over $(\{0,1\}^\alpha \cup \{same^*, \perp\}) \times \{0,1\}^\beta$ such that, for all $m \in \{0,1\}^\alpha$

$$\mathsf{Tamper}_{f,g}^m, L \approx_\epsilon \mathsf{Copy}(m, \mathsf{NMSim});,$$

where $(\mathsf{NMSim}_1, \mathsf{NMSim}_2) = (\tilde{m}, L^{sim}) \leftarrow \mathsf{NMSim}$, and $\mathsf{Copy}(m, \mathsf{NMSim})$ is defined as

$$\mathsf{Copy}(m, \mathsf{NMSim}) \overset{\text{def}}{=} \begin{cases} (m, L^{sim}) \text{ if } \tilde{m} = \text{same}^* \\ (\tilde{m}, L^{sim}) \text{ otherwise} \end{cases}$$

We call the above code as left-augmented (NMSim outputs the left state), and say that the code is right-augmented when the simulator outputs the right state instead.

We also require the following secret sharing property of non-malleable codes in the 2-split-state model $\mathcal{F}_{\text{split}}$. It states that any 2-split-state non-malleable code is a 2-out-of-2 secret sharing scheme.

Lemma 2.14. *[5] Let $\mathsf{Enc} : \{0,1\}^\alpha \to \{0,1\}^\beta \times \{0,1\}^\beta$ and $\mathsf{Dec} : \{0,1\}^\beta \times \{0,1\}^\beta \to \{0,1\}^\alpha \cup \{\perp\}$ be an ϵ-non-malleable code in the 2-split-state model for some $\epsilon < 1/2$. For any pair of messages $m_0, m_1 \in \{0,1\}^\alpha$, $R^{m_0} \approx_{2\epsilon} R^{m_1}$, where $(L^{m_0}, R^{m_0}) \leftarrow \mathsf{Enc}(m_0)$ and $(L^{m_1}, R^{m_1}) \leftarrow \mathsf{Enc}(m_1)$.*

We instantiate our NMC with two constructions, one due to [1, 3, 6] and another due to [44], which are both stated below in Theorems 2.15 and 2.16, respectively. While the former instantiation gives us a very simple non-malleable code, as explained in the introduction, the latter instantiation gives us a better error for the same rate.

Theorem 2.15. *[1, 3, 6] There exists and efficient construction of an ε-non-malleable code in the split state model with message space $\{0,1\}^\kappa$ and codeword space $\mathbb{F}_p^q \times \mathbb{F}_p^q$, for a prime $p \leq 2^{O(\kappa)}$, $q = O(\kappa^4)$ and $\varepsilon = 2^{-\Omega(\kappa)}$.*

Theorem 2.16. *[44, Theorem 1.15] There are constants $0 < c_1, c_2 < 1$ such that for any $\beta \in \mathbb{N}$ and $2^{-\frac{c_2\beta}{\log\beta}} \leq \epsilon \leq c_1$, there exists an explicit non-malleable code in the 2-split-state model with block length 2β, rate $\Omega(\frac{\log\log\log(1/\epsilon)}{\log\log(1/\epsilon)})$ and error ϵ.*

The NMC in Theorem 2.15 is shown to be augmented secure in [2]. Further, the NMC in Theorem 2.16 relies on building a two source non-malleable extractor and then using the [22] compiler to get a 2-split-state NMC. Hence, we can show that the NMC in Theorem 2.16 is in fact an augmented NMC using the following two observations: 1) the two source non-malleable extractor constructed in [44] is in fact, a strong two source non-malleable extractor (which allows leaking one complete source); 2) as proved in [42, Proposition 2], the [22] compiler gives an augmented NMC, when a strong two-source non-malleable extractor is used.

3 A RATE BOOSTER FOR NON-MALLEABLE CODES

In this section, we prove the following result.

Theorem 3.1. *Let* $(\mathsf{NMEnc} : \{0,1\}^\kappa \to \{0,1\}^n \times \{0,1\}^n,\ \mathsf{NMDec} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^\kappa)$ *be an* ε*-augmented non-malleable code in the split-state model. Then, for any* $\kappa^* \in \omega(n)$*, there exists an efficient* $O(\sqrt{\varepsilon} + 2^{-\Omega(\kappa/\log\kappa)})$*-augmented non-malleable code (specifically, right-augmented, i.e., the simulator outputs the right state along with the tampered message) from* κ^**-bit messages to at most* $(3 + o(1))\kappa^*$*-bit codewords.*

In Section 3.1, we will give our construction, and its security is proved in Section 3.2. Using (NMEnc, NMDec) from Theorem 2.15, we get the following instantiation, which is conceptually very simple, as shown in Figure 3 in the introduction.

Corollary 3.2. *For any integer* $\kappa^* > 0, \varepsilon^* \in 2^{-\Omega((\kappa^*)^{1/5}/\mathrm{polylog}(\kappa^*))}$*, there is an efficient* ε^**-right-augmented non-malleable code from* κ^**-bit messages to* $(3 + o(1))\kappa^*$*-bit codewords in the split-state model.*

Further, using (NMEnc, NMDec) from Theorem 2.16, and setting $\kappa^* = n \log n$ we can get the following instantiation, with a better error at the expense of being more complicated.

Corollary 3.3. *For any integer* $\kappa^* > 0$*,* $\varepsilon^* \in 2^{-\Omega(\kappa^*/\log^3 \kappa^*)}$*, there is an efficient* ε^**-right-augmented non-malleable code from* κ^**-bit messages to* $(3+o(1))\kappa^*$*-bit codewords in the split-state model.*

3.1 Our Construction

Building Blocks. Let $N = 2\kappa^* + 8n$. Our construction, shown in Figure 4 requires the following building blocks.

- Let Ext be the $(N,k,d,\kappa^*,\varepsilon_2)$-strong seeded extractor from Lemma 2.8 with $k = \kappa^* + 4n$, and $\varepsilon_2 = 2^{-\Omega(\kappa/\log\kappa)}$.
- Let (Tag, Vrfy) be the one time ε_3-message authentication codes from Lemma 2.10 with key length $\tau = \Theta(\kappa)$, messages of size at most N, tag length $t = \Theta(\kappa)$, and $\varepsilon_3 = 2^{-\Omega(\kappa)}$.

We now describe our construction below. The encoder chooses a source w and a seed s corresponding to a seeded extractor Ext and then computes $c := m \oplus \mathsf{Ext}(w; s)$, where $\oplus$ is the bitwise XOR. Further, it authenticates the source w and the ciphertext c to get the tags σ_w and σ_c respectively, generates the non-malleable encoding (using the underlying NMC) of the seed and the authentication keys to get (ℓ, r) and finally outputs (ℓ, w, σ_w) as one state and (r, c, σ_c) as the other.

```
Enc(m):
  • $s \in_R \{0,1\}^d$, $k_w, k_c \in_R \{0,1\}^\tau$, $w \in_R \{0,1\}^N$.
  • $c = m \oplus \mathsf{Ext}(w; s)$.
  • $\sigma_w = \mathsf{Tag}_{k_w}(w)$ and $\sigma_c = \mathsf{Tag}_{k_c}(c)$.
  • $(\ell, r) \leftarrow \mathsf{NMEnc}(\mu)$ where $\mu = (s, k_w, k_c)$.
  • Output $(\ell, w, \sigma_w), (r, c, \sigma_c)$.

Dec($(\ell, w, \sigma_w), (r, c, \sigma_c)$):
  • $(s, k_w, k_c) = \mathsf{NMDec}(\ell, r)$.
  • If $\mathsf{Vrfy}_{k_w}(w, \sigma_w) = 1$ and $\mathsf{Vrfy}_{k_c}(c, \sigma_c) = 1$,
    Output $m = \mathsf{Ext}(w; s) \oplus c$.
    Else, Output $\perp$.
```

Figure 4: Our NMC Construction

3.2 Security Proof

The correctness of the construction is immediate by definition. We now prove the desired non-malleability.

Fix any $m \in \{0,1\}^{\kappa^*}$, tampering functions $(f, g) \in \mathcal{F}_{\mathrm{split}}$. Throughout this proof, we use the notation $\underline{X}$ to denote any part X of the message/codeword after tampering. Let $f = (f_1, f_2, f_3)$ and $g = (g_1, g_2, g_3)$, where

$$f_1 : \{0,1\}^{n+N+t} \to \{0,1\}^n,\ f_2 : \{0,1\}^{n+N+t} \to \{0,1\}^N,$$
$$f_3 : \{0,1\}^{n+N+t} \to \{0,1\}^t\ ,$$

and

$$g_1 : \{0,1\}^{n+\kappa^*+t} \to \{0,1\}^n,\ g_2 : \{0,1\}^{n+\kappa^*+t} \to \{0,1\}^{\kappa^*},$$
$$g_3 : \{0,1\}^{n+\kappa^*+t} \to \{0,1\}^t\ .$$

Let $W, S, K_w, K_c, C, \sigma_w, \sigma_c, L, R$ be the distributions of $w, s, k_w, k_c, c, \sigma_w, \sigma_c, \ell, r$ [9] sampled/computed in $\mathsf{Enc}(m)$. Now, apply the tampering to get $(\underline{L}, \underline{W}, \underline{\sigma_w}) = f(L, W, \sigma_w)$, $(\underline{R}, \underline{C}, \underline{\sigma_c}) = g(R, C, \sigma_c)$, decode to get $\underline{\mu} = \mathsf{NMDec}(\underline{L}, \underline{R})$ and $\underline{M} = \mathsf{Dec}((\underline{L}, \underline{W}, \underline{\sigma_w}), (\underline{R}, \underline{C}, \underline{\sigma_c}))$. Further, note that the distribution $\underline{M}$ is the distribution $\mathsf{Tamper}^m_{f,g}$ itself.

Our proof proceeds by partitioning the sample space of the codeword, $\{0,1\}^n \times \{0,1\}^N \times \{0,1\}^t \times \{0,1\}^n \times \{0,1\}^{\kappa^*} \times \{0,1\}^t$ and proving non-malleability in each of the partitions. Consider the following three partitions.

$$\begin{aligned}\mathcal{P}_1 = \{&(\ell, w, \sigma_w, r, c, \sigma_c)\ :\\ &(f_2(\ell, w, \sigma_w), f_3(\ell, w, \sigma_w), g_2(r, c, \sigma_c), g_3(\ell, c, \sigma_c))\\ &= (w, \sigma_w, c, \sigma_c), \mathsf{NMDec}(f_1(\ell, w, \sigma_w), g_1(r, c, \sigma_c)) = \mathsf{NMDec}(\ell, r)\}\end{aligned}$$

$$\begin{aligned}\mathcal{P}_2 = \{&(\ell, w, \sigma_w, r, c, \sigma_c)\ :\\ &(f_2(\ell, w, \sigma_w), f_3(\ell, w, \sigma_w), g_2(r, c, \sigma_c), g_3(\ell, c, \sigma_c))\\ &\neq (w, \sigma_w, c, \sigma_c), \mathsf{NMDec}(f_1(\ell, w, \sigma_w), g_1(r, c, \sigma_c)) = \mathsf{NMDec}(\ell, r)\}\end{aligned}$$

and

$$\begin{aligned}\mathcal{P}_3 := \{&(\ell, w, \sigma_w, r, c, \sigma_c)\ :\ \mathsf{NMDec}(f_1(\ell, w, \sigma_w), g_1(r, c, \sigma_c))\\ &\neq \mathsf{NMDec}(\ell, r)\}\ ,\end{aligned}$$

The event that the codeword is in partition $\mathcal{P}_1$ corresponds to the event when, even after tampering, the decoding of the inner codeword (L, R) remains unchanged, and also W, σ_w, C, σ_c remain unchanged, and hence the final tampered codeword must decode to the original message. The event that the codeword is in partition $\mathcal{P}_2$ corresponds to the event that the decoding of the inner code remains unchanged after tampering, while one of the pairs (W, σ_w) or (C, σ_c) are changed. We will show that if this event occurs then, using the security of MACs, the tampering is detected with overwhelming probability. The event that the codeword is in partition $\mathcal{P}_3$ corresponds to the event that the inner code is non-trivially tampered and does not decode to $\mathsf{Dec}(L, R)$. In this case, we will show that the tampered codeword (and hence its decoding) is independent of the original message m.

We begin by showing that the event that the codeword $(L, W, \sigma_w, R, C, \sigma_c) \in \mathcal{P}_i$ for some $i \in \{1, 2, 3\}$ is independent of the message m even given R, C, σ_c.

[9] We slightly abuse notation and let σ_w and σ_c denote both the distribution of the authentication tags σ_w and σ_c as well as samples from those distributions.

Let $\widetilde{C}$ be uniform in $\{0,1\}^{\kappa^*}$ and independent of S, K_w, K_c, W, σ_w, and hence also of L, R. Let $\widetilde{\sigma}_c = \mathsf{Tag}_{K_c}(\widetilde{C})$. Let $b_1, b_2, \widetilde{b}_2$ be boolean random variables defined as follows. The random variable b_1 indicates whether W, σ_w changes after tampering, i.e., $b_1 = 1$ if and only if $(f_2(L, W, \sigma_w), f_3(L, W, \sigma_w)) = (W, \sigma_w)$. Similarly, the random variable b_2 (respectively, $\widetilde{b}_2$) indicates whether C, σ_c (respectively, $\widetilde{C}, \widetilde{\sigma}_c$) changes after tampering, i.e., $b_2 = 1$ (respectively, $\widetilde{b}_2 = 1$) if and only if $(g_2(R, C, \sigma_c), g_3(R, C, \sigma_c)) = (C, \sigma_c)$ (respectively, $(g_2(R, \widetilde{C}, \widetilde{\sigma}_c), g_3(R, \widetilde{C}, \widetilde{\sigma}_c)) = (\widetilde{C}, \widetilde{\sigma}_c)$). Notice that for any i, whether the event that $(L, W, \sigma_w, R, C, \sigma_c) \in \mathcal{P}_i$ (respectively, $(L, W, \sigma_w, R, \widetilde{C}, \widetilde{\sigma}_c) \in \mathcal{P}_i$) occurs is determined by the random variables $L, R, f_1(L, W, \sigma_w), g_1(R, C, \sigma_c)$, b_1, and b_2 (respectively, L, R, $f_1(L, W, \sigma_w)$, $g_1(R, \widetilde{C}, \widetilde{\sigma}_c)$, b_1, and $\widetilde{b}_2$).

Lemma 3.4.

$$\Delta\left(C, \sigma_c; \widetilde{C}, \widetilde{\sigma}_c \mid L, R, f_1(L, W, \sigma_w), b_1\right) \le 4\varepsilon + 2^{-\Omega(\kappa/\log\kappa)} .$$

Proof. By Lemma 2.14, we have that one of the shares of a split-state non-malleable code is 2ε-independent of the message. In other words, there is a random variable L^* independent of S, K_w, K_c such that

$$L, S, K_w, K_c \approx_{2\varepsilon} L^*, S, K_w, K_c ,$$

Introducing fresh random variable W that is sampled independent of L, S, K_w, K_c, and observing that σ_w is a deterministic function of K_w and W, we have that

$$L, S, K_w, K_c, W, \sigma_w \approx_{2\varepsilon} L^*, S, K_w, K_c, W, \sigma_w . \tag{10}$$

Let b_1^* be a boolean random variable that is 1 if and only if $(f_2(L^*, W, \sigma_w), f_3(L^*, W, \sigma_w)) = (W, \sigma_w)$. By Lemma 2.1, the average min-entropy of W given $b_1^*, f_1(L^*, W, \sigma_w)$ is at least $N - n - 1$, which is at least k. Thus, by Lemma 2.8, we have that

$$\Delta(\mathsf{Ext}(W; S) \ ; \ U \mid b_1^*, f_1(L^*, W, \sigma_w), L^*, K_w, K_c, S) \le \varepsilon_2 , \tag{11}$$

where U is uniform in $\{0,1\}^{\kappa^*}$ and independent of all other random variables.

Applying Lemma 2.2 to Equation 10 twice, we get

$$\Delta(b_1^*, f_1(L^*, W, \sigma_w), L^* \ ; \ b_1, f_1(L, W, \sigma_w), L \mid \mathsf{Ext}(W; S), K_w, K_c, S) \le 2\varepsilon , \tag{12}$$

and

$$\Delta(b_1^*, f_1(L^*, W, \sigma_w), L^* \ ; \ b_1, f_1(L, W, \sigma_w), L \mid U, K_w, K_c, S) \le 2\varepsilon . \tag{13}$$

By triangle inequality on inequalities 11, 12, 13, we get that

$$\Delta(\mathsf{Ext}(W; S) \ ; \ U \mid b_1, f_1(L, W, \sigma_w), L, K_w, K_c, S) \le \varepsilon_2 + 4\varepsilon .$$

Notice that the distribution R is independent of W and is only correlated to (L, K_w, K_c, S) (as $(L, R) \equiv \mathsf{NMEnc}(K_w, K_c, S)$). Hence by applying Lemma 2.2, we can include R to get that

$$\Delta(\mathsf{Ext}(W; S) \ ; \ U \mid b_1, f_1(L, W, \sigma_w), L, R, K_w, K_c, S) \le \varepsilon_2 + 4\varepsilon .$$

Finally, observing that given $(b_1, f_1(L, W, \sigma_w), L, R, K_w, K_c, S)$, the random variable $\mathsf{Ext}(W; S) \oplus m$ is distributed as C, and $U \oplus m$ is distributed as $\widetilde{C}$, we get the desired result by another application of Lemma 2.2. □

The following lemma bounds the probability that the codeword is in $\mathcal{P}_2$ and doesn't decode to $\perp$.

Lemma 3.5.

$$\Pr[\mathsf{Tamper}^m_{f,g} \ne \perp \ \wedge \ (L, W, \sigma_w, R, C, \sigma_c) \in \mathcal{P}_2] \le 4\varepsilon + 2^{-\Omega(\kappa)} .$$

Proof. Since the codeword is in $\mathcal{P}_2$, we have that $\mathsf{Dec}(L, R) = \mathsf{Dec}(\underline{L}, \underline{R})$, and hence K_w, K_c remain unchanged after tampering. Moreover, one of the pairs (W, σ_w) or (C, σ_c) has been tampered with. Assume without loss of generality that (W, σ_w) has been changed after tampering. By the security of MAC, this tampering will be detected by our decoding algorithm as long as the tampering is independent of K_w. We argue using the secret sharing property of non-malleable codes that this is indeed the case since L is almost independent of K_w.

By Lemma 2.14, we have that one of the shares of a split-state non-malleable code is 2ε-independent of the message. In other words, there is a random variable L^* independent of S, K_w, K_c such that

$$L, S, K_w, K_c \approx_{2\varepsilon} L^*, S, K_w, K_c ,$$

Introducing fresh random variable W that is sampled independent of L, S, K_w, K_c, and observing that σ_w is a deterministic function of K_w and W, we have that

$$L, S, K_w, K_c, W, \sigma_w \approx_{2\varepsilon} L^*, S, K_w, K_c, W, \sigma_w .$$

Also, by definition of one-time message authentication codes, we have that

$$\Pr[\mathsf{Vrfy}_{K_w}(f_2(L^*, W, \sigma_w), f_3(L^*, W, \sigma_w)) = 1 \wedge (f_2(L^*, W, \sigma_w), f_3(L^*, W, \sigma_w)) \ne (W, \sigma_w)] = 2^{-\Omega(\kappa)} .$$

By triangle inequality, we get

$$\Pr[\mathsf{Vrfy}_{K_w}(f_2(L, W, \sigma_w), f_3(L, W, \sigma_w)) = 1 \wedge (f_2(L, W, \sigma_w), f_3(L, W, \sigma_w)) \ne (W, \sigma_w)] \le 2^{-\Omega(\kappa)} + 2\varepsilon .$$

Similarly, there exists R^* independent of S, K_w, K_c such that

$$R, S, K_w, K_c \approx_{2\varepsilon} R^*, S, K_w, K_c ,$$

and observing that C is independent of R, K_w, K_c given S, we have that

$$R, S, K_w, K_c, C, \sigma_c \approx_{2\varepsilon} R^*, S, K_w, K_c, C, \sigma_c .$$

This implies, via an argument similar as above, that

$$\Pr[\mathsf{Vrfy}_{K_c}(g_2(R, C, \sigma_c), g_3(R, C, \sigma_c)) = 1 \wedge (g_2(R, C, \sigma_c), g_3(R, C, \sigma_c) \ne (C, \sigma_c)] \le 2^{-\Omega(\kappa)} + 2\varepsilon .$$

The desired result then follows from a simple union bound by observing that if $\mathsf{Tamper}^m_{f,g} \ne \perp$, and the codeword is in $\mathcal{P}_2$, i.e., K_w, K_c remain unchanged after tampering, while W, σ_w, C, σ_c are changed after tampering, then we must have that

- $\mathsf{Vrfy}_{K_w}(f_2(L, W, \sigma_w), f_3(L, W, \sigma_w)) = 1$
- $\mathsf{Vrfy}_{K_c}(g_2(R, C, \sigma_c), f_3(R, C, \sigma_c)) = 1$,
- At least one of $(f_2(L, W, \sigma_w), f_3(L, W, \sigma_w)) \ne (W, \sigma_w)$ or $(g_2(R, C, \sigma_c), g_3(R, C, \sigma_c)) \ne (C, \sigma_c)$.

□

Lemma 3.6. *For any fixed k_w, k_c, and any $\alpha > 2\varepsilon + 2^{-d}$, if $\Pr[(L, W, \sigma_w, R, \widetilde{C}, \widetilde{\sigma}_c) \in \mathcal{P}_3] \ge \alpha$, then the statistical distance between*

$$\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c|_{(L, W, \sigma_w, R, C, \sigma_c) \in \mathcal{P}_3}$$

and

$$\mathsf{Dec}(f(L,W,\sigma_w), g(R,\widetilde{C},\widetilde{\sigma}_c)), R, \widetilde{C}, \widetilde{\sigma}_c|_{(L,W,\sigma_w,R,\widetilde{C},\widetilde{\sigma}_c)\in\mathcal{P}_3}$$

is at most $2^{-\Omega(\kappa/\log\kappa)} + \frac{4\varepsilon}{\alpha}$.

Proof. In this lemma, we wish to argue that the decoding of the tampered codeword is independent of the message m. We begin by using that since the codeword is in $\mathcal{P}_3$, the output of the tampered codeword for the inner code $\mathsf{NMEnc}(S, k_w, k_c)$ is independent of S which is the seed of the strong extractor.

We will use the non-malleability of the inner code, $(\mathsf{NMEnc}, \mathsf{NMDec})$. Consider the tampering functions f^*, g^* on (L, R) that sample $W, \widetilde{C}$ uniformly and then compute $f^*(L) = f_1(L, W, \mathsf{Tag}_{k_w}(W))$ and $g^*(R) = g_1(R, \widetilde{C}, \mathsf{Tag}_{k_c}(\widetilde{C}))$. Let $\underline{S}, \underline{K_w}, \underline{K_c} = \mathsf{NMDec}(f^*(L), g^*(R))$.[10] By the ε-augmented non-malleability of $(\mathsf{NMEnc}, \mathsf{NMDec})$, there exists a simulator NMSim such that

$$\underline{S}, \underline{K_w}, \underline{K_c}, L, S, W, \widetilde{C} \approx_\varepsilon \mathsf{Copy}(\mu, \mathsf{NMSim}), S, W, \widetilde{C}\,, \tag{14}$$

where the output of NMSim depends on $W, \widetilde{C}$ and the functions f_1, g_1. Recall (from definition 2.13) that NMSim is parsed as the joint distribution $(\mathsf{NMSim}_1, L^{\mathsf{Sim}})$. We further parse NMSim_1 as $(\underline{S^{\mathsf{Sim}}}, \underline{K_w^{\mathsf{Sim}}}, \underline{K_c^{\mathsf{Sim}}})$, when NMSim_1 is conditioned to not output same^* or $\perp$. If $\mathsf{NMSim}_1 = \perp$, set $(\underline{S^{\mathsf{Sim}}}, \underline{K_w^{\mathsf{Sim}}}, \underline{K_c^{\mathsf{Sim}}}) = (\perp, \perp, \perp)$. The event that $(L, W, \sigma_w, R, \widetilde{C}, \widetilde{\sigma}_c) \in \mathcal{P}_3$ is the same as $(\underline{S}, \underline{K_w}, \underline{K_c}) \neq (S, k_w, k_c)$. By inequality (14) and the hypothesis of the lemma, we have that $\Pr[\mathsf{NMSim}_1 \notin \{\mathsf{same}^*, (S, k_w, k_c)\}] \geq \alpha - \varepsilon$.
By Lemma 2.3, we have that

$$\begin{aligned}
&(\underline{S}, \underline{K_w}, \underline{K_c}, L, S, W, \widetilde{C})|_{(\underline{S},\underline{K_w},\underline{K_c})\neq(S,k_w,k_c)} \approx_{\frac{2\varepsilon}{\alpha}}\\
&\mathsf{Copy}(\mu, \mathsf{NMSim}), S, W, \widetilde{C}|_{\mathsf{NMSim}_1\notin\{\mathsf{same}^*,(S,k_w,k_c)\}}\\
&(\underline{S}, \underline{K_w}, \underline{K_c}, L, S, W, \widetilde{C})|_{(\underline{S},\underline{K_w},\underline{K_c})\neq(S,k_w,k_c)} \approx_{\frac{2\varepsilon}{\alpha}}\\
&(\underline{S^{\mathsf{Sim}}}, \underline{K_w^{\mathsf{Sim}}}, \underline{K_c^{\mathsf{Sim}}}, L^{\mathsf{Sim}}, S, W, \widetilde{C})|_{\mathsf{NMSim}_1\notin\{\mathsf{same}^*,(S,k_w,k_c)\}}
\end{aligned} \tag{15}$$

We have that for any w,

$$\begin{aligned}
\Pr[W = w|\mathsf{NMSim}_1 &\notin \{\mathsf{same}^*, (S, k_w, k_c)\}]\\
&\leq \frac{\Pr[W = w]}{\Pr[\mathsf{NMSim}_1 \notin \{\mathsf{same}^*, (S, k_w, k_c)\}]}\\
&\leq \frac{2^{-N}}{\alpha - \varepsilon} \leq 2^{d-N}\,,
\end{aligned}$$

where we use that $\alpha > 2\varepsilon + 2^{-d}$.
Thus $\mathbf{H}_\infty(W|\mathsf{NMSim}_1 \notin \{\mathsf{same}^*, (S, k_w, k_c)\}) \geq N - d$. For the remainder of the proof, we assume $\mathsf{NMSim}_1 \notin \{\mathsf{same}^*, (S, k_w, k_c)\}$ (or, when appropriate, $(L, W, \sigma_w, R, \widetilde{C}, \widetilde{\sigma}_c) \in \mathcal{P}_3$) but don't state them explicitly.
Let $\mathsf{Vrfy}_\ell^{\mathsf{Sim}} := \mathsf{Vrfy}_{\underline{K_w^{\mathsf{Sim}}}}(f_2(L^{\mathsf{Sim}}, W, \sigma_w), f_3(L^{\mathsf{Sim}}, W, \sigma_w))$, where $\sigma_w = \mathsf{Tag}_{k_w}(W)$. By Lemma 2.1, we have that

$$\begin{aligned}
\widetilde{\mathbf{H}}_\infty(W|\underline{S^{\mathsf{Sim}}}, \underline{K_w^{\mathsf{Sim}}}, \underline{K_c^{\mathsf{Sim}}}, \mathsf{Vrfy}_\ell^{\mathsf{Sim}}, f_1(L^{\mathsf{Sim}}, W, \sigma_w),&\\
\mathsf{Ext}(f_2(L^{\mathsf{Sim}}, W, \sigma_w); \underline{S^{\mathsf{Sim}}}))&
\end{aligned}$$

[10]For the purpose of this proof alone, we slightly abuse notation and let $\underline{S}, \underline{K_w}, \underline{K_c}$ denote the tampered seed and the MAC keys when the codeword $((L, W, \mathsf{Tag}_{k_w}(W)), (R, \widetilde{C}, \mathsf{Tag}_{k_c}(\widetilde{C})))$ is tampered using functions f, g.

is at least $N - d - \kappa^* - \kappa - n - 1 \geq \kappa^* + 4n$. Thus, by the strong extractor property of Ext,

$$\begin{aligned}
\mathsf{Ext}(W; S) \approx_{\varepsilon_2} U|S, L^{\mathsf{Sim}}, \widetilde{C}, f_1(L^{\mathsf{Sim}}, W, \sigma_w), \underline{K_w^{\mathsf{Sim}}}, \underline{K_c^{\mathsf{Sim}}}&\\
\mathsf{Ext}(f_2(L^{\mathsf{Sim}}, W, \sigma_w); \underline{S^{\mathsf{Sim}}}), \underline{S^{\mathsf{Sim}}}, \mathsf{Vrfy}_\ell^{\mathsf{Sim}}&
\end{aligned} \tag{16}$$

In order to show that the tampered codeword is independent of the message, we first need a statement similar to the inequality 16, but where the simulated output is replaced by the decoding of the tampering of $(L, W, \sigma_w, R, \widetilde{C}, \widetilde{\sigma}_c)$. We will obtain this using (15) twice and applying the triangle inequality.
For the remainder of the proof, let $L, W, \sigma_w, R, \widetilde{C}, \widetilde{\sigma}_c$ be distributed as $L, W, \sigma_w, R, \widetilde{C}, \widetilde{\sigma}_c|_{(L,W,\sigma_w,R,\widetilde{C},\widetilde{\sigma}_c)\in\mathcal{P}_3}$. Also, let $\underline{K_w}, \underline{K_c}, \underline{S}, \underline{L}, \underline{R}, \underline{W}$ be the corresponding random variables after tampering conditioned on the event that $(L, W, \sigma_w, R, \widetilde{C}, \widetilde{\sigma}_c) \in \mathcal{P}_3$. We shorthand the distribution in the LHS and RHS of 16 by A and B, respectively. Also, we define $\mathsf{Vrfy}_\ell := \mathsf{Vrfy}_{\underline{K_w}}(f_2(L, W, \sigma_w), f_3(L, W, \sigma_w))$. Applying Lemma 2.2 on inequality 15, we get that

$$A \approx_{\frac{2\varepsilon}{\alpha}} \mathsf{Ext}(W; S), S, L, \widetilde{C}, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, \underline{K_w}, \underline{K_c}, \mathsf{Vrfy}_\ell\,, \tag{17}$$

and

$$B \approx_{\frac{2\varepsilon}{\alpha}} U, S, L, \widetilde{C}, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, \underline{K_w}, \underline{K_c}, \mathsf{Vrfy}_\ell\,. \tag{18}$$

By triangle inequality applied on the inequalities (17),(18), and (16), we have that

$$\mathsf{Ext}(W; S) \approx_{\varepsilon_2 + \frac{4\varepsilon}{\alpha}} U|S, L, \widetilde{C}, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, \underline{K_w}, \underline{K_c}, \mathsf{Vrfy}_\ell\,.$$

Notice that given $(S, L, \widetilde{C}, \underline{S}, \underline{K_w}, \underline{K_c}, \underline{L})$, the random variable R is independent of W, and hence we obtain that

$$\begin{aligned}
\mathsf{Ext}(W; S) \approx_{\varepsilon_2 + \frac{4\varepsilon}{\alpha}} U|S, L, \widetilde{C}, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, \underline{K_w}, \underline{K_c}, \mathsf{Vrfy}_\ell,&\\
R, g_1(R, \widetilde{C}, \widetilde{\sigma}_c)\,.&
\end{aligned}$$

Using Lemma 2.2, we can drop $\widetilde{C}$ on both sides to get

$$\mathsf{Ext}(W; S) \approx_{\varepsilon_2 + \frac{4\varepsilon}{\alpha}} U|S, L, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, \underline{K_w}, \underline{K_c}, \mathsf{Vrfy}_\ell, R, \underline{R} \tag{19}$$

where $\underline{R} \overset{\text{def}}{=} g_1(R, \widetilde{C}, \widetilde{\sigma}_c)$.
We now observe that W is independent of $\widetilde{C}$ given $S, L, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, \underline{K_w}, \underline{K_c}, \mathsf{Vrfy}_\ell, R, \underline{R}$. To see this, we will use Lemma 2.5 with $V_0 = L, R, S, k_w, k_c$, and then $V_1 = \underline{L}$, $V_2 = \underline{R}$, $V_3 = \mathsf{Ext}(\underline{W}; \underline{S}), \mathsf{Vrfy}_\ell$, and observing that $\underline{S}, \underline{K_w}, \underline{K_c}$ is a deterministic function of $\underline{L}$ and $\underline{R}$. Thus, $\mathsf{Ext}(W; S)$ is independent of $\widetilde{C}$ given $S, L, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, \underline{K_w}, \underline{K_c}, \mathsf{Vrfy}_\ell, R, \underline{R}$.
Now, applying Lemma 2.6 to equation 19, with $X = \mathsf{Ext}(W, S)$, $Y = \widetilde{C}$, $Z = (S, L, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, \underline{K_w}, \underline{K_c}, \mathsf{Vrfy}_\ell, R, \underline{R})$, and using the above observation that $Y = \widetilde{C}$ is independent of $\mathsf{Ext}(W; S)$ given Z and of U given Z, we get that conditioned on $\widetilde{C} \oplus \mathsf{Ext}(W; S) = m$ on the LHS, and on $\widetilde{C} \oplus U = m$ on the RHS:

$$\begin{aligned}
C, S, L, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, \underline{K_w}, \underline{K_c}, \mathsf{Vrfy}_\ell, R, \underline{R} \approx_{\varepsilon_2 + \frac{4\varepsilon}{\alpha}}&\\
\widetilde{C}, S, L, \underline{L}, \mathsf{Ext}(\underline{W}; \underline{S}), \underline{S}, \underline{K_w}, \underline{K_c}, \mathsf{Vrfy}_\ell, R, \underline{R}\,,&
\end{aligned}$$

since $\widetilde{C}$ conditioned on $\widetilde{C} \oplus U = m$ is distributed identically as $\widetilde{C}$ given all other random variables on the RHS in the above inequality. The desired result follows by observing that the tampered codeword is a function of

$$\underline{L}, \underline{R}, \mathsf{Ext}(\underline{W}; \underline{S}), C, R, \mathsf{Vrfy}_\ell\,.$$

□

Proof of Theorem 3.1. The simulator $\mathsf{Sim}_{f,g}$ does the following. It samples $W, S, K_w, K_c, \sigma_w, L, R, \widetilde{C}, \widetilde{\sigma}_c$. Let $I^{\mathsf{sim}} \in \{1,2,3\}$ be a random variable indicating the partition ($\mathcal{P}_{I^{\mathsf{sim}}}$) in which the sampled codeword belongs. If $(L, W, \sigma_w, R, \widetilde{C}, \widetilde{\sigma}_c) \in \mathcal{P}_1$, then the simulator outputs $(\mathsf{same}^*, R, \widetilde{C}, \widetilde{\sigma}_c)$ and sets $I^{\mathsf{sim}} = 1$, if $(L, W, R, \widetilde{C}) \in \mathcal{P}_2$ then the simulator outputs $(\perp, R, \widetilde{C}, \widetilde{\sigma}_c)$ and sets $I^{\mathsf{sim}} = 2$, else the simulator outputs $\left(\mathsf{Dec}((L, W, \sigma_w), (R, \widetilde{C}, \widetilde{\sigma}_c)), R, \widetilde{C}, \widetilde{\sigma}_c\right)$ and sets $I^{\mathsf{sim}} = 3$.

Now, if $\Delta((\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c)|_{(L,W,\sigma_w,R,C,\sigma_c)\in\mathcal{P}_i};$ $\mathsf{Copy}(m, \mathsf{Sim}_{f,g})|_{I^{\mathsf{sim}}=i}) \leq \gamma_i$, by Lemma 2.4, we will get:

$$\Delta((\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c);\, D^m_{f,g}) \leq \sum_{i=1}^{3} \gamma_i \Pr[(L, W, \sigma_w, R, C, \sigma_c) \in \mathcal{P}_i], \tag{20}$$

where $D^m_{f,g}$ is a distribution on $(\{0,1\}^{\kappa^*} \cup \{\mathsf{same}^*, \perp\}) \times \{0,1\}^{n+\kappa^*+t}$, such that $\Pr[D^m_{f,g} = d] = \sum_{i=1}^3 \Pr[(L, W, \sigma_w, R, C, \sigma_c) \in \mathcal{P}_i] \cdot \Pr[\mathsf{Copy}(m, \mathsf{Sim}_{f,g})|_{I^{\mathsf{sim}}=i} = d]$. But, by Lemma 3.4, we get $(C, \sigma_c) \approx_{4\varepsilon+2^{-\Omega(\kappa/\log\kappa)}} (\widetilde{C}, \widetilde{\sigma}_c)|(L, R, \underline{L}, b_1)$, where b_1 is the bit indicating if $(\underline{W}, \underline{\sigma_w}) = (W, \sigma_w)$ or not, and the event that $(L, W, \sigma_w, R, C, \sigma_c) \in \mathcal{P}_i$ (or correspondingly $(L, W, \sigma_w, R, \widetilde{C}, \widetilde{\sigma}_c) \in \mathcal{P}_i$) can be determined by the random variables on the LHS and RHS of this inequality. Hence, together with the fact that $\Pr[\mathsf{Copy}(m, \mathsf{Sim}_{f,g}) = d] = \sum_{i=1}^3 \Pr[(L, W, \sigma_w, R, \widetilde{C}, \widetilde{\sigma}_c) \in \mathcal{P}_i] \cdot \Pr[\mathsf{Copy}(m, \mathsf{Sim}_{f,g})|_{I^{\mathsf{sim}}=i} = d]$, for each $d \in (\{0,1\}^{\kappa^*} \cup \{\mathsf{same}^*, \perp\}) \times \{0,1\}^{n+\kappa^*+t}$, we get:

$$\Delta(D^m_{f,g}\,;\, \mathsf{Copy}(m, \mathsf{Sim}_{f,g})) \leq 4\varepsilon + 2^{-\Omega(\kappa/\log\kappa)} \tag{21}$$

Combining inequality 20 and 21, by triangle inquality, we get:

$$\begin{aligned}&\Delta((\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c);\, \mathsf{Copy}(m, \mathsf{Sim}_{f,g}))\\ &\leq 4\varepsilon + 2^{-\Omega(\kappa/\log\kappa)} + \sum_{i=1}^{3} \gamma_i \Pr[(L, W, \sigma_w, R, C, \sigma_c) \in \mathcal{P}_i],\end{aligned} \tag{22}$$

Hence, now we consider each partition and bound the corresponding term in the RHS of the summation in the inequality 22.

First, since we know that $(\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c)|_{(L,W,\sigma_w,R,C,\sigma_c)\in\mathcal{P}_1} = (m, R, C, \sigma_c)$ and

$\mathsf{Copy}(m, \mathsf{Sim}_{f,g})|_{I^{\mathsf{sim}}=1} = (m, R, \widetilde{C}, \widetilde{\sigma}_c)$, we use Lemma 3.4 to get:

$$\begin{aligned}&\Delta((\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c)|_{(L,W,\sigma_w,R,C,\sigma_c)\in\mathcal{P}_1}; \mathsf{Copy}(m, \mathsf{Sim}_{f,g})|_{I^{\mathsf{sim}}=1})\\ &\leq \Delta((C, \sigma_c)\,;\, (\widetilde{C}, \widetilde{\sigma}_c)|(L, R, \underline{L}, b_1)) \leq 4\varepsilon + 2^{-\Omega(\kappa/\log\kappa)}\end{aligned} \tag{23}$$

Hence, $\gamma_1 = 4\varepsilon + 2^{-\Omega(\kappa/\log\kappa)}$, on the RHS of inequality 22.

Similarly, since we know that by Lemma 3.5, $\Pr[(\mathsf{Tamper}^m_{f,g} \neq \perp \wedge (L, W, \sigma_w, R, C, \sigma_c) \in \mathcal{P}_2] \leq 4\varepsilon + 2^{-\Omega(\kappa)}$, and $\mathsf{Copy}(m, \mathsf{Sim}_{f,g})|_{I^{\mathsf{sim}}=2} = (\perp, R, \widetilde{C}, \widetilde{\sigma}_c)$, we use Lemma 3.4 to get:

$$\begin{aligned}&\Pr[(L, W, \sigma_w, R, C, \sigma_c) \in \mathcal{P}_2]\\ &\cdot \Delta((\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c)|_{(L,W,\sigma_w,R,C,\sigma_c)\in\mathcal{P}_2};\\ &\qquad\qquad \mathsf{Copy}(m, \mathsf{Sim}_{f,g})|_{I^{\mathsf{sim}}=2})\\ &\leq \Pr[(\mathsf{Tamper}^m_{f,g} \neq \perp \wedge (L, W, \sigma_w, R, C, \sigma_c) \in \mathcal{P}_2]\\ &+ \Delta((C, \sigma_c)\,;\, (\widetilde{C}, \widetilde{\sigma}_c)|(L, R, \underline{L}, b_1)) \leq 8\varepsilon + 2^{-\Omega(\kappa/\log\kappa)}\,.\end{aligned} \tag{24}$$

Hence, $\gamma_2 \cdot \Pr[(L, W, \sigma_w, R, C, \sigma_c) \in \mathcal{P}_2] \leq 8\varepsilon + 2^{-\Omega(\kappa/\log\kappa)}$, on the RHS of inequality 22.

Now, for the third partition, we set $\alpha = \sqrt{\varepsilon} + 2^{-d/2} (> 2\varepsilon + 2^{-d})$ in Lemma 3.6, and get that for any fixing a, b of K_w, K_c, if $\Pr[(L, W, \sigma_w, R, \widetilde{C}, \widetilde{\sigma}_c) \in \mathcal{P}_3 | K_w = a, K_c = b] \geq \alpha = \sqrt{\varepsilon} + 2^{-d/2}$, then

$$\begin{aligned}&\Delta((\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c)|_{(L,W,\sigma_w,R,C,\sigma_c)\in\mathcal{P}_3, K_w=a, K_c=b}\\ &\;;\; \mathsf{Copy}(m, \mathsf{Sim}_{f,g})|_{I^{\mathsf{sim}}=3, K_w=a, K_c=b})\\ &\leq 2^{-\Omega(\kappa/\log\kappa)} + \frac{4\varepsilon}{\alpha}\\ &\leq 2^{-\Omega(\kappa/\log\kappa)} + 4\sqrt{\varepsilon}\end{aligned}$$

Now, since $d = \Omega(\kappa/\log\kappa)$, and once again using Lemma 3.4, we get:

$$\begin{aligned}&\Pr[(L, W, \sigma_w, R, C, \sigma_c) \in \mathcal{P}_3]\\ &\cdot \Delta((\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c)|_{(L,W,\sigma_w,R,C,\sigma_c)\in\mathcal{P}_3}\,;\\ &\qquad\qquad \mathsf{Copy}(m, \mathsf{Sim}_{f,g})|_{I^{\mathsf{sim}}=3})\\ &\leq 4\varepsilon + 2^{-\Omega(\kappa/\log\kappa)} + \Pr[I^{\mathsf{sim}} = 3] \cdot \sum_{a,b} \Pr[K_w = a\\ &, K_c = b | I^{\mathsf{sim}} = 3] \cdot \Delta((\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c)|_{I^{\mathsf{sim}}=3, K_w=a, K_c=b}\\ &\qquad\;;\; \mathsf{Copy}(m, \mathsf{Sim}_{f,g})|_{I^{\mathsf{sim}}=3, K_w=a, K_c=b})\\ &\leq (4\varepsilon + 2^{-\Omega(\kappa/\log\kappa)})\\ &\quad + \sum_{a,b} \Pr[K_w = a, K_c = b] \cdot \Pr[I^{\mathsf{sim}} = 3 | K_w = a, K_c = b]\\ &\quad \cdot \Delta((\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c)|_{I^{\mathsf{sim}}=3, K_w=a, K_c=b}\\ &\qquad\;;\; \mathsf{Copy}(m, \mathsf{Sim}_{f,g})|_{I^{\mathsf{sim}}=3, K_w=a, K_c=b})\\ &\leq (4\varepsilon + 2^{-\Omega(\kappa/\log\kappa)}) + \sum_{a,b} \Pr[K_w = a, K_c = b]\\ &\cdot O(\sqrt{\varepsilon} + 2^{-\Omega(\kappa/\log\kappa)}) \leq O(\sqrt{\varepsilon} + 2^{-\Omega(\kappa/\log\kappa)})\end{aligned} \tag{25}$$

In the above inequality 25, we make an abuse of notation, and use $(\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c)|_{I^{\mathsf{sim}}=3, K_w=a, K_c=b}$ to denote the distribution where the partition is first picked using I^{sim}, and then the distribution $(\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c)$ is drawn conditioned on the partition. This distribution is clearly identical to

$(\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c)|_{(L,W,\sigma_w,R,C,\sigma_c)\in\mathcal{P}_3}$, which we use above.

Hence, $\gamma_3 \cdot \Pr[(L, W, \sigma_w, R, C, \sigma_c) \in \mathcal{P}_3] = O(\sqrt{\varepsilon} + 2^{-\Omega(\kappa/\log\kappa)})$, on the RHS of inequality 22. Finally, we can use inequalities 23,24 and 25, in inequality 22 to get:

$$\Delta((\mathsf{Tamper}^m_{f,g}, R, C, \sigma_c);\, \mathsf{Copy}(m, \mathsf{Sim}_{f,g})) \leq O(\sqrt{\varepsilon} + 2^{-\Omega(\kappa/\log\kappa)})$$

□

4 A RATE BOOSTER FOR TWO-SOURCE NON-MALLEABLE EXTRACTORS

We begin by defining a strong two-source non-malleable extractor, introduced in [22].

Definition 4.1. A function $\mathsf{nm2Ext} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ is a strong $(n_1, n_2, k_1, k_2, \epsilon)$-strong two-source non-malleable extractor if for each (n_1, k_1)-source X and (n_2, k_2)-source Y, the following holds. Let $f : \{0,1\}^{n_1} \to \{0,1\}^{n_1}$ and $g : \{0,1\}^{n_2} \to \{0,1\}^{n_2}$

be functions with no fixed points, i.e., for each $x \in \{0,1\}^{n_1}$, $y \in \{0,1\}^{n_2}$, $f(x) \neq x$ and $g(y) \neq y$. Then,

$$\mathsf{nm2Ext}(X,Y), X, \mathsf{nm2Ext}(f(X),g(Y)) \approx_{\epsilon} U_m, X, \mathsf{nm2Ext}(f(X),g(Y))$$

We drop the adjective strong when the distribution X is excluded from the statistical distance above.
The rate is defined to be the ratio $m/(n_1+n_2)$.

We consider an $(n_1,n_2,k_1,k_2,d,\epsilon_1)$-strong two-source unbalanced non-malleable extractor, nm2Ext, with $n_2 = o(n_1)$, and an (n_1,k_1,d,l,ϵ_2)-strong seeded extractor and define the following function $\mathsf{nm2Ext}^* : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^l$:

$$\mathsf{nm2Ext}^*(x,y) \stackrel{\text{def}}{=} \mathsf{Ext}(x;\mathsf{nm2Ext}(x,y)),\ \forall\ x \in \{0,1\}^{n_1}, y \in \{0,1\}^{n_2}$$

We show that the above function is a two-source non-malleable extractor achieving rate 1/2, with appropriate instantiation of Ext.

Theorem 4.2. *If* nm2Ext *is an* $(n_1,n_2,k_1,k_2,d,\epsilon_1)$*-strong two-source unbalanced non-malleable extractor, with* $n_2 = o(n_1)$ *and* Ext *is an* (n_1,k_1,d,l,ϵ_2)*-strong seeded extractor,then* $\mathsf{nm2Ext}^*$ *is an* $(n_1,n_2,k_1,k_2,l,\epsilon_1+\epsilon_2)$*-two-source non-malleable extractor. Further, if we instantiate* Ext *with the seeded extractor in Lemma 2.8, with* $k_1, l < n_1/2$*, then the rate of* $\mathsf{nm2Ext}^*$ *is* $1/2$.

Proof. Consider an (n_1,k_1)-source X, an (n_2,k_2)-source Y and tampering functions f and g (with no fixed points). By the nm2Ext security, we get:

$$X, \mathsf{nm2Ext}(X,Y), \mathsf{nm2Ext}(f(X),g(Y)) \approx_{\epsilon_1} X, U_d, \mathsf{nm2Ext}(f(X),g(Y))$$

By applying the function Ext on the first two distributions on either side and using Lemma 2.2, we get:

$$\begin{aligned}\mathsf{Ext}(X;\mathsf{nm2Ext}(X,Y)),&\mathsf{nm2Ext}(f(X),g(Y))\\ &\approx_{\epsilon_1} \mathsf{Ext}(X;U_d), \mathsf{nm2Ext}(f(X),g(Y))\end{aligned}$$

By security of Ext, since $\widetilde{\mathbf{H}}_\infty(X|\mathsf{nm2Ext}(f(X),g(Y))) \geq k_1$, we get:

$$\mathsf{Ext}(X;U_d), \mathsf{nm2Ext}(f(X),g(Y)) \approx_{\epsilon_2} U_l, \mathsf{nm2Ext}(f(X),g(Y))$$

Hence, by triangle inequality, this gives:

$$\begin{aligned}\mathsf{Ext}(X;\mathsf{nm2Ext}(X,Y)),&\mathsf{nm2Ext}(f(X),g(Y))\\ &\approx_{\epsilon_1+\epsilon_2} U_l, \mathsf{nm2Ext}(f(X),g(Y))\end{aligned}$$

which proves that $\mathsf{nm2Ext}^*$ is a two-source non-malleable extractor.
Rate. Setting Ext to be the extractor from Lemma 2.8, with $k_1, l < n/2$ gives $\mathsf{nm2Ext}^*$ a rate of at most $(n_1/2)/(n_1+n_2) = 1/2$. □

ACKNOWLEDGMENTS

The second author thanks Venkatesan Guruswami for insightful discussions. The authors would also like to thank the anonymous referees for their valuable comments and helpful suggestions. The third author's work is supported by Microsoft Research Grant.

REFERENCES

[1] Divesh Aggarwal. 2015. Affine-evasive sets modulo a prime. *Inf. Process. Lett.* 115, 2 (2015), 382–385. https://doi.org/10.1016/j.ipl.2014.10.015

[2] Divesh Aggarwal, Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. 2016. Optimal Computational Split-state Non-malleable Codes. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II.* 393–417. https://doi.org/10.1007/978-3-662-49099-0_15

[3] Divesh Aggarwal and Jop Briët. 2016. Revisiting the Sanders-Bogolyubov-Ruzsa theorem in Fp^n and its application to non-malleable codes. In *IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, July 10-15, 2016.* IEEE, 1322–1326. https://doi.org/10.1109/ISIT.2016.7541513

[4] Divesh Aggarwal, Ivan Damgård, Jesper Buus Nielsen, Maciej Obremski, Erick Purwanto, João Ribeiro, and Mark Simkin. 2019. Stronger Leakage-Resilient and Non-Malleable Secret Sharing Schemes for General Access Structures. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II.* 510–539. https://doi.org/10.1007/978-3-030-26951-7_18

[5] Divesh Aggarwal, Yevgeniy Dodis, Tomasz Kazana, and Maciej Obremski. 2015. Non-malleable Reductions and Applications. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015.* 459–468. https://doi.org/10.1145/2746539.2746544

[6] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. 2014. Non-malleable codes from additive combinatorics. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014.* 774–783. https://doi.org/10.1145/2591796.2591804

[7] Divesh Aggarwal, Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. 2015. Leakage-resilient non-malleable codes. In *Theory of Cryptography Conference.* Springer, 398–426.

[8] Divesh Aggarwal, Tomasz Kazana, and Maciej Obremski. 2017. Inception Makes Non-malleable Codes Stronger. In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 10678)*, Yael Kalai and Leonid Reyzin (Eds.). Springer, 319–343. https://doi.org/10.1007/978-3-319-70503-3_10

[9] Divesh Aggarwal and Maciej Obremski. 2020. A constant rate non-malleable code in the split-state model. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020.* IEEE, 1285–1294. https://doi.org/10.1109/FOCS46700.2020.00122

[10] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. 2002. The EM Side-Channel(s). In *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers (Lecture Notes in Computer Science, Vol. 2523)*, Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar (Eds.). Springer, 29–45. https://doi.org/10.1007/3-540-36400-5_4

[11] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. 2015. A Rate-Optimizing Compiler for Non-malleable Codes Against Bit-Wise Tampering and Permutations. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I.* 375–397. https://doi.org/10.1007/978-3-662-46494-6_16

[12] Saikrishna Badrinarayanan and Akshayaram Srinivasan. 2019. Revisiting Non-Malleable Secret Sharing. In *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I.* 593–622. https://doi.org/10.1007/978-3-030-17653-2_20

[13] Marshall Ball, Dana Dachman-Soled, Mukul Kulkarni, and Tal Malkin. 2018. Non-malleable Codes from Average-Case Hardness: AC ^{0}, Decision Trees, and Streaming Space-Bounded Tampering. In *Advances in Cryptology – EUROCRYPT 2018*, Jesper Buus Nielsen and Vincent Rijmen (Eds.). Springer International Publishing, Cham, 618–650.

[14] Eli Biham and Adi Shamir. 1997. Differential Fault Analysis of Secret Key Cryptosystems. In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings (Lecture Notes in Computer Science, Vol. 1294)*, Burton S. Kaliski Jr. (Ed.). Springer, 513–525. https://doi.org/10.1007/BFb0052259

[15] Nishanth Chandran, Vipul Goyal, Pratyay Mukherjee, Omkant Pandey, and Jalaj Upadhyay. 2016. Block-Wise Non-Malleable Codes. In *ICALP (LIPIcs, Vol. 55).* Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 31:1–31:14.

[16] Nishanth Chandran, Bhavana Kanukurthi, and Srinivasan Raghuraman. 2016. Information-Theoretic Local Non-malleable Codes and Their Applications. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II.* 367–392. https://doi.org/10.1007/978-3-662-49099-0_14

[17] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. 2016. Non-malleable extractors and codes, with their many tampered extensions. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016.* 285–298. https://doi.org/10.1145/2897518.2897547

[18] Eshan Chattopadhyay, Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, and Sruthi Sekar. 2019. Privacy Amplification from Non-malleable Codes. In *Progress in Cryptology - INDOCRYPT 2019 - 20th International Conference on Cryptology in India, Hyderabad, India, December 15-18, 2019, Proceedings (Lecture Notes in Computer Science, Vol. 11898)*, Feng Hao, Sushmita Ruj, and Sourav Sen Gupta (Eds.). Springer, 318–337. https://doi.org/10.1007/978-3-030-35423-7_16

[19] Eshan Chattopadhyay and Xin Li. 2017. Non-malleable codes and extractors for small-depth circuits, and affine functions. In *STOC.* ACM, 1171–1184.

[20] Eshan Chattopadhyay and David Zuckerman. 2014. Non-malleable Codes against Constant Split-State Tampering. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*. 306–315. https://doi.org/10.1109/FOCS.2014.40

[21] Mahdi Cheraghchi and Venkatesan Guruswami. 2014. Capacity of non-malleable codes. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*. 155–168. https://doi.org/10.1145/2554797.2554814

[22] Mahdi Cheraghchi and Venkatesan Guruswami. 2014. Non-malleable Coding against Bit-Wise and Split-State Tampering. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*. 440–464. https://doi.org/10.1007/978-3-642-54242-8_19

[23] Sandro Coretti, Yevgeniy Dodis, Björn Tackmann, and Daniele Venturi. 2016. Non-Malleable Encryption: Simpler, Shorter, Stronger. In *Theory of Cryptography*, Eyal Kushilevitz and Tal Malkin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 306–335.

[24] Sandro Coretti, Ueli Maurer, Björn Tackmann, and Daniele Venturi. 2014. From Single-Bit to Multi-Bit Public-Key Encryption via Non-Malleable Codes. *IACR Cryptology ePrint Archive* 2014 (2014), 324. http://eprint.iacr.org/2014/324

[25] Dana Dachman-Soled, Mukul Kulkarni, and Aria Shahverdi. 2019. Tight upper and lower bounds for leakage-resilient, locally decodable and updatable non-malleable codes. *Inf. Comput.* 268 (2019). https://doi.org/10.1016/j.ic.2019.05.001

[26] Dana Dachman-Soled, Feng-Hao Liu, Elaine Shi, and Hong-Sheng Zhou. 2020. Locally Decodable and Updatable Non-malleable Codes and Their Applications. *J. Cryptol.* 33, 1 (2020), 319–355. https://doi.org/10.1007/s00145-018-9306-z

[27] Yevgeniy Dodis, Bhavana Kanukurthi, Jonathan Katz, Leonid Reyzin, and Adam Smith. 2012. Robust Fuzzy Extractors and Authenticated Key Agreement from Close Secrets. *IEEE Transactions on Information Theory* (2012).

[28] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. 2008. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM J. Comput.* 38, 1 (2008), 97–139. arXiv:cs/0602007.

[29] Nico Döttling, Jesper Buus Nielsen, and Maciej Obremski. 2017. Information Theoretic Continuously Non-Malleable Codes in the Constant Split-State Model. *Electronic Colloquium on Computational Complexity (ECCC)* 24 (2017), 78. https://eccc.weizmann.ac.il/report/2017/078

[30] Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. 2013. Non-malleable Codes from Two-Source Extractors. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*. 239–257. https://doi.org/10.1007/978-3-642-40084-1_14

[31] Stefan Dziembowski and Krzysztof Pietrzak. 2007. Intrusion-Resilient Secret Sharing. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07)*. IEEE Computer Society, Washington, DC, USA, 227–237. https://doi.org/10.1109/FOCS.2007.35

[32] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. 2010. Non-Malleable Codes. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*. 434–452. http://conference.itcs.tsinghua.edu.cn/ICS2010/content/papers/34.html

[33] Sebastian Faust, Kristina Hostáková, Pratyay Mukherjee, and Daniele Venturi. 2017. Non-Malleable Codes for Space-Bounded Tampering. In *CRYPTO (2) (Lecture Notes in Computer Science, Vol. 10402)*. Springer, 95–126.

[34] Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. 2014. Continuous Non-malleable Codes. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*. 465–488. https://doi.org/10.1007/978-3-642-54242-8_20

[35] Vipul Goyal and Ashutosh Kumar. 2018. Non-malleable secret sharing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*. 685–698. https://doi.org/10.1145/3188745.3188872

[36] Vipul Goyal, Omkant Pandey, and Silas Richelson. 2016. Textbook non-malleable commitments. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*. 1128–1141. https://doi.org/10.1145/2897518.2897657

[37] Divya Gupta, Hemanta K. Maji, and Mingyuan Wang. 2018. Non-malleable Codes Against Lookahead Tampering. In *Progress in Cryptology - INDOCRYPT 2018 - 19th International Conference on Cryptology in India, New Delhi, India, December 9-12, 2018, Proceedings (Lecture Notes in Computer Science, Vol. 11356)*, Debrup Chakraborty and Tetsu Iwata (Eds.). Springer, 307–328. https://doi.org/10.1007/978-3-030-05378-9_17

[38] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. 2007. Unbalanced Expanders and Randomness Extractors from Parvaresh-Vardy Codes. In *IEEE Conference on Computational Complexity*. 96–108.

[39] Zahra Jafargholi and Daniel Wichs. 2015. Tamper Detection and Continuous Non-malleable Codes. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*. 451–480. https://doi.org/10.1007/978-3-662-46494-6_19

[40] Thomas Johansson, Gregory Kabatianskii, and Ben J. M. Smeets. 1993. On the Relation between A-Codes and Codes Correcting Independent Errors. In *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*. 1–11. https://doi.org/10.1007/3-540-48285-7_1

[41] Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, and Sruthi Sekar. 2017. Four-State Non-malleable Codes with Explicit Constant Rate. In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II*. 344–375. https://doi.org/10.1007/978-3-319-70503-3_11

[42] Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, and Sruthi Sekar. 2018. Non-malleable Randomness Encoders and Their Applications. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*. 589–617. https://doi.org/10.1007/978-3-319-78372-7_19

[43] Xin Li. 2017. Improved Non-Malleable Extractors, Non-Malleable Codes and Independent Source Extractors. In *Symposium on Theory of Computing, STOC 2017, Montreal, Canada, June 19-23, 2017*.

[44] Xin Li. 2019. Non-Malleable Extractors and Non-Malleable Codes: Partially Optimal Constructions. In *Computational Complexity Conference, CCC 2019, New Brunswick, June 18-20, 2019*.

[45] Feng-Hao Liu and Anna Lysyanskaya. 2010. Algorithmic Tamper-Proof Security under Probing Attacks. In *Security and Cryptography for Networks, 7th International Conference, SCN 2010, Amalfi, Italy, September 13-15, 2010. Proceedings (Lecture Notes in Computer Science, Vol. 6280)*, Juan A. Garay and Roberto De Prisco (Eds.). Springer, 106–120. https://doi.org/10.1007/978-3-642-15317-4_8

[46] Feng-Hao Liu and Anna Lysyanskaya. 2012. Tamper and Leakage Resilience in the Split-State Model. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings (Lecture Notes in Computer Science, Vol. 7417)*, Reihaneh Safavi-Naini and Ran Canetti (Eds.). Springer, 517–532. https://doi.org/10.1007/978-3-642-32009-5_30

[47] Akshayaram Srinivasan and Prashant Nalini Vasudevan. 2019. Leakage Resilient Secret Sharing and Applications. In *Advances in Cryptology – CRYPTO 2019*, Alexandra Boldyreva and Daniele Micciancio (Eds.). Springer International Publishing, Cham, 480–509.