Non-malleable Randomness Encoders and their Applications

Bhavana Kanukurthi * Sai Lakshmi Bhavana Obbattu ** Sruthi Sekar***

Indian Institute Of Science, Bangalore

Abstract. Non-malleable Codes (NMCs), introduced by Dziembowski, Peitrzak and Wichs (ITCS 2010), serve the purpose of preventing "related tampering" of encoded messages. The most popular tampering model considered is the 2-split-state model where a codeword consists of 2 states, each of which can be tampered independently. While NMCs in the 2-split state model provide the strongest security guarantee, despite much research in the area we only know how to build them with poor rate ($\Omega(\frac{1}{logn})$, where n is the codeword length). However, in many applications of NMCs one only needs to be able to encode randomness i.e., security is not required to hold for arbitrary, adversarially chosen messages. For example, in applications of NMCs to tamper-resilient security, the messages that are encoded are typically randomly generated secret keys. To exploit this, in this work, we introduce the notion of "*Non-malleable Randomness Encoders*" (NMREs) as a relaxation of NMCs in the following sense: NMREs output a random message along with its corresponding non-malleable encoding.

Our main result is the construction of a 2-split state, rate-$\frac{1}{2}$ NMRE. While NMREs are interesting in their own right and can be directly used in applications such as in the construction of tamper-resilient cryptographic primitives, we also show how to use them, in a black-box manner, to build a 3-split-state (standard) NMCs with rate $\frac{1}{3}$. This improves both the number of states, as well as the rate, of existing constant-rate NMCs.

1 Introduction

How do we protect sensitive data from being tampered? Can we ensure that tampering of the data is detected? These are precisely the kind of questions answered in the rich area of Coding Theory. Encoding data using an error correcting code ensures that data stays the same so long as the errors introduced are

* Department Of Computer Science and Automation, Indian Institute Of Science, Email: bhavana.kanukurthi@gmail.com. Research supported in part by Department of Science and Technology Inspire Faculty Award.

** Department Of Computer Science and Automation, Indian Institute Of Science, Email: oslbhavana@gmail.com

*** Department Of Mathematics, Indian Institute Of Science, Email: sruthi.sekar1@gmail.com.

appropriately limited. Dziembowski, Peitrzak and Wichs [DPW10], introduced an important variant of ECCs based on the well-established intuition in eryptography that, often times, tampering data into something independent doesn't threaten the security of the underlying cryptosystem. (For example, an adversary who obtains signatures on an independently generated signing key, will not be able to forge signatures with respect to the original secret key.) Specifically, they introduced Non-malleable Codes which provide a guarantee that an adversary cannot tamper the codeword of message m into the codeword of a related message m'.

As observed in [DPW10], it is impossible to build NMCs secure against all functions. Therefore, NMCs are defined with respect to a family of tampering functions. A natural class of tampering functions that have been considered is the $t-$split state model where a codeword consists of t states, each of which is tampered independently by the adversary. An important parameter of interest for NMCs is its $\mathsf{Rate} = \frac{k}{n}$ where $k = \mathsf{message\ length}$ and $n = \mathsf{codeword\ length}$.

Prior to this work, the best known results for various t-state tampering models were given in Table 1.

Result	**States**	**Rate**
[CG14b]	n	1
[KOS17]	4	1/3
[Li17]	2	$\Omega(\frac{1}{logn})$

Table 1

As we can see, while 2-split-state NMCs provide the strongest security guarantee, despite significant effort in this direction, we only know how to build them with poor rate of $\Omega(\frac{1}{logn})$. An important observation about the definition of non-malleable codes is that they ensure non-malleability of the codeword of *any* message, even adversarially chosen ones. However, in most applications of non-malleable codes, such as tamper-resilient security, the message is not adversarially controlled. In fact, it is typically a randomly chosen secret key. With that in mind, in this work, we ask the following question:

Is there any advantage in non-malleably encoding randomness?

With this question in mind, we introduce "*Non-malleable Randomness Encoders*" (NMRE) as objects which allow you to generate randomness along with its corresponding non-malleable encoding. We then go on to show that NMREs can, infact, be built efficiently and with interesting parameters: Specficially, we build a **2-state, rate-$\frac{1}{2}$ Non-malleable Randomness Encoder**. Given the major open problem in this area of NMCs i.e., of building an explicit 2-state, constant-rate non-malleable code, we propose NMREs as a useful alternative to NMCs in applications where they suffice, of which we give some examples.

Application of NMREs Consider the key generation process of symmetric key cryptosystems. These processes typically use uniform randomness r to generate a secret key k. Using NMREs we can generate r along with its non-malleable

encoding C. Instead of storing the secret key k directly, we store C in the secret state. The advantage is that this secret state is now reslient to tampering attacks. Of course, this will require us to decode C and regenerate the secret key k whenever we need to use it. Therefore, the applicability of NMREs is for scenarios where key generation is an efficient process.

As another application of NMREs, we show that NMREs can be used, in a black-box, to improve the current state of the art of standard non-malleable codes. Specifically, we build 3-state Non-malleable Codes with a rate of $\frac{1}{3}$.

1.1 Prior Work

We now survey the main results in the area of Non-malleable Codes. For the sake of completeness, we may revisit some of the terminology introduced in the previous section. Informally, a non-malleable code (NMC) [DPW10] provides the following guarantee – a codeword of message m, if tampered, will decode to one of the following:

- $\perp$ i.e., it detects tampering.
- the original message m itself i.e., the tampering did not change the message
- something independent of m

While each of these cases may occur with varying probabilities (for example, a tampering function that maps codeword to identity always results in Case 2), the probability with which these cases occur need to be independent of the underlying message. In [DPW10], the authors observe that it is impossible to build NMCs which are secure against unrestricted tampering. Specifically, a function $f(c) \stackrel{\text{def}}{=} \mathsf{Enc}(\mathsf{Dec}(c) + 1)$ clearly tampers $m = \mathsf{Dec}(c)$ into a related $m + 1$. This necessitates the need to define non-malleable codes with respect to the class of functions they protect against. ([DPW10] show the existence of non-malleable codes w.r.t tampering families of size less than 2^{2^n}, where n is the codeword length.)

Tampering Families and Rate. One family that has been considered in several works is that of t-state tampering families: here, a codeword consists of t blocks or states and the adversary tampers each of these independently. The family of functions $\mathcal{F}$ therefore consists of t-functions $f_1, \ldots, f_t$. For $t = n$, the model is referred to as the *bit-wise tampering model*. Dziembowski et al. [DPW10] constructed non-malleable codes resilient against this family. In addition to the class of tampering functions, another important parameter is that of $\mathsf{Rate} = \frac{\mathsf{message\ length}}{\mathsf{codeword\ length}}$ they achieve. Cheraghchi and Guruswami [CG14b] built an explicit construction of an *optimal rate* NMC in the bit-wise tampering model. While building NMCs for this model is technically challenging, a disadvantage is that, from a practical stand point, requiring each bit of the codeword to be stored in an independent state makes the model less desirable. Indeed, the best possible t-split state model would be where $t = 2$. On this front, the first efficient solution was obtained for 1-bit messages by Dziembowski, Kazana and Obremski [DKO13]. The first construction for encoding arbitrary-length messages, was

an $\Omega(n^{-6/7})$-rate construction due to Aggarwal, Dodis and Lovett [ADL14]. At the same time, in [CG14a], Cheraghchi and Guruswami show a $1 - 1/t$ upper bound on best achievable rate for the t-split state family (and, specifically, $1/2$ when $t = 2$). The first constant rate construction for any $t = o(n)$, was due to Chattopadhyay and Zuckerman [CZ14]. Specifically, they build a constant rate, 10-state NMC. Recently, Kanukurthi, Obbattu and Sekar [KOS17] obtained a $4-$state construction (i.e., $t = 4$) with rate $\frac{1}{3}$. For $t = 2$, the current best known construction is due to Li [Li17] with a rate of $\Omega(1/\log n)$. In other results, Aggarwal et al. [ADKO15] demonstrated connections between various split-state models and Agrawal et al. [AGM$^+$15] build optimal NMCs which are simultaneously resilient to permutation attacks as well as bit-wise tampering attacks. On the computational front, there are constructions in the 2-split-state model such as [LL12] and the optimal construction of [AAG$^+$16].

Variants of Non-malleable codes Since the introduction of Non-malleable codes several variants of Non-malleable codes have been considered. Some of them are Continuous NMCs [FMNV14,JW15,AKO15,DNO17], Locally updatable and decodable NMCs [DLSZ14,DKS17,CKR16].

1.2 Our Results

In this work, we introduce *Non-malleable Randomness Encoders*. Informally, NMREs allow for the generation of randomness r along with its corresponding non-malleable encoding C. The non-malleability is, as for standard NMCs, defined with respect to $\mathcal{F}$, a family of tampering functions. Note that any non-malleable code NMC is, by default, a secure NMRE (simply generate randomness r at random and let the codeword be the output of NMC). The main challenge is in building a **rate-optimal, state-optimal NMRE**. We give an overview of our construction which uses Information-theoretic *one-time message authentication codes (MACs)* as well as *Randomness Extractors*.

Randomness extractors Ext are objects that allow us to generate randomness from a source W with a Min-entropy guarantee using a short *seed* (s) of true randomness. Message authentication codes $\mathsf{MAC} = (\mathsf{Tag}, \mathsf{Vrfy})$ are secret key primitives which guarantee that even given $\mathsf{Tag}(m; k)$, an adversary cannot generate m', t' such that $m' \neq m$ and $\mathsf{Vrfy}(m', t') = 1$. Our construction makes a black-box use of a 2-split-state non-malleable code NMEnc.

Recall that our goal is to construct a 2-state NMRE with constant rate. For now, consider a 3-state codeword $C = W||L||R$ where $(L, R) \leftarrow NMC(s)$ where W is the source of the extractor and s is a randomly chosen seed. We can see that this is a three-state NMRE resilient to $f_{\mathsf{ID}}, f_2, f_3$ where f_{ID} is the identity function, f_2 and f_3 are arbitrary functions. The idea is that since L, R is the output of an NMC, any independent tampering of L, R respectively renders a tampered s', if not $\perp$, to be indepdent of s. From here, extractor security can be used– recall that W remains unchanged by our choice of the function family – to argue non-malleability. (This argument isn't trivial. Particularly, to complete it, we must show how $\mathsf{Ext}(W; s')$ can be simulated to complete the proof of

non-malleability. While we don't go into the details, it can be done.) Note also that this argument crucially relies on f_1 being f_{ID}. Indeed, if we let W to be tampered to W', then there is no extractor security. (One can come up with concocted constructions of randomness extractors such that tampering w' to a related w and keeping s the same, can result in a related extractor output.) To prevent tampering of W, we use a one-time message authentication code: we let $(L, R) \leftarrow NMC(s, k, \mathsf{Tag}_k(W))$. This gives us a 3-split-state construction ($C = W||L||R$), i.e., one that is resilient to (f_1, f_2, f_3) where each f_i acts independently on each state.

We note that our techniques are similar in spirit to those of [KOS17]'s 4-state NMC. However, our goal here is to build 2-state NMREs. So, on the one hand, we can leverage the fact that the security we are trying to achieve is weaker. On the other hand, the task of bringing down the number of states to 3 while retaining good rate is challenging. To bring down the number of states in our current proposed 3-state NMRE, we wish to explore possibility of combining two of the states. Can we combine W with, say, L? Without going into too much detail regarding the definition of a NMC, an adversary breaking non-malleability can be viewed as consisting of two parts: one that specifies the tampering functions and the other that actually distinguishes the output of the tampering experiment from the simulated experiment.

When we combine W with L, to use the underlying NMC, we would need to be able to do two things: a) specify the tampering functions that act on L and R and b) use the distinguisher of the NMRE to build a distinguisher for the NMC. Indeed, the former can be done by merely hardwiring the value of W. Unfortunately, we will not be able to use the distinguisher for the NMRE for the simple reason that we won't know how W was tampered. It is for this reason that we require our NMCs to satisfy a stronger property of "augmented non-malleability". An augmented nonmalleable code is one that remains non-malleable even when the adversary, after specifying the tampering function, additionally obtains one of the states along with the decoded (tampered) message. In our proof, we carefully use the augmented non-malleability of the underlying NMC to argue non-malleability of 2-split state NMRE.

The question still remains of how to instantiate the underlying augmented NMC. We note that the Augmented Non-malleable Codes due to [ADL14] would, asymptotically, indeed give us a constant-rate solution. However, the parameters would be less desirable in terms of tradeoffs between the error and the rate. (Given that this isn't our final construction, a more detailed parameter calculation would be tedious.) To overcome these shortcomings, we instead resort to Li's 2-state construction which has the so-far best-known rate. Since Li only proves the standard non-malleability of his scheme, in Appendix A, we give a proof that it is indeed augmented non-malleable. (This follows by revisitig the connection between seedless non-malleable extractors and non-malleable codes due to [CG14b] and reproving it to achieve augmented non-malleability from strong NME.) Combining this with the outline laid out above, we get our final NMRE construction.

Building NMCs from NMRE as a black-box. Our next goal is to use NMREs in a black-box to build NMCs for arbitrary messages m. To do so, we use the "random message" encoded as a part of the NMRE to both compute the ciphertext (using a one-time pad) $c = Enc_{k_e}(m)$ as well as authenticate the ciphertext i.e., compute $t = \mathsf{Tag}_{\mathsf{k_2}}(\mathsf{Enc}_{\mathsf{k_e}}(\mathsf{m}))$. In order to build it in a black-box using the NMRE, it is important that we do not use anything pertaining to the message m in our underlying NMRE. The codeword now needs to have the codeword of NMRE and, additionally, c, t. In the proof, we show that the non-malleability of k_a, k_e essentially suffices to argue the over-all non-malleability and achieve constant rate. Further we show that c, t can stored jointly in a single state giving us a 3-state NMC for arbitrary messages with rate 1/3.

1.3 Organization of the Paper

We write preliminaries and building blocks in Sections 2 and 3. We give definition of NMRE in Section 4.1, an explicit construction of NMRE in 4.3, security proof of the construction in Section 4.4, instantiate it and analyze rate and error in rest of the Section 4 . We show how to build a 3-state augmented non-malleable code from an NMRE, prove security, instantiate and analyze in Sections 5.1,5.2 and 5.3 respectively. We add concluding remarks in Section 6. Appendix B gives details about [Li17]'s 2-state NMC being augmented.

2 Preliminaries

Notation. κ denotes security parameter throughout. $s \in_R S$ denotes uniform sampling from set S. $x \leftarrow X$ denotes sampling from a probability distribution X. $x||y$ represents concatenation of two binary strings x and y. $|x|$ denotes length of binary string x. U_l denotes the uniform distribution on $\{0,1\}^l$. All logarithms are base 2.

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

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

(they are said to be ε-close if $\mathbf{SD}\left(X_1, X_2\right) \leq \varepsilon$ and denoted by $X_1 \approx_\varepsilon X_2$). 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), define the (average) conditional min-entropy of W given E [DORS08] as

$$\widetilde{\mathbf{H}}_\infty(W \mid E) = -\log(\mathop{\mathbf{E}}_{e \leftarrow E}(2^{-\mathbf{H}_\infty(W|E=e)}))$$

(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

$\widetilde{\mathbf{H}}_\infty(W|E) \geq t$.
We now state some Lemmata about statistical distance and average entropy loss.

Proposition 1. *Let $A_1, ..., A_n$ be mutually exclusive and exhaustive events. Then, for probability distributions X_1, X_2 over some set S, we have:*

$$\mathbf{SD}\left(X_1, X_2\right) \leq \sum_{i=1}^{n} \Pr[A_i].\mathbf{SD}\left(X_1|A_i, X_2|A_i\right)$$

where $X_j|A_i$ is the distribution of X_j conditioned on the event A_i.

Lemma 1. *For any random variables A, B, C if $(A, B) \approx_\epsilon (A, C)$, then $B \approx_\epsilon C$*

Lemma 2. *For any random variables A, B if $A \approx_\epsilon B$, then for any function f, $f(A) \approx_\epsilon f(B)$*

Lemma 3. *[KOS17] Let A, B be correlated random variables over $\mathcal{A}, \mathcal{B}$. For randomized functions $F : \mathcal{A} \to \mathcal{X}$, $G : \mathcal{A} \to \mathcal{X}$ (randomness used is independent of B) if $\forall\ a \in \mathcal{A}, F(a) \approx_\epsilon G(a)$, then $(B, A, F(A)) \approx_\epsilon (B, A, G(A))$*

Lemma 4. *[DORS08] If B has at most 2^λ possible values, then $\widetilde{\mathbf{H}}_\infty(A \mid B) \geq \mathbf{H}_\infty(A, B) - \lambda \geq \mathbf{H}_\infty(A) - \lambda$. and, more generally, $\widetilde{\mathbf{H}}_\infty(A \mid B, C) \geq \widetilde{\mathbf{H}}_\infty(A, B \mid C) - \lambda \geq \widetilde{\mathbf{H}}_\infty(A \mid C) - \lambda$*

2.1 Definitions

Definition 1. *A (possibly randomized) function $\mathsf{Enc} : \{0,1\}^l \to \{0,1\}^n$ and a deterministic function $\mathsf{Dec} : \{0,1\}^n \to \{0,1\}^l \cup \{\perp\}$ is said to be a* coding scheme *if $\forall\ m \in \{0,1\}^l,\ \Pr[\mathsf{Dec}(\mathsf{Enc}(m)) = m] = 1$. l is called the message length and n is called the block length or the codeword length. Rate of a coding scheme is given by $\frac{l}{n}$.*

We now state the definition of non-malleable codes, as given in [CG14b].

Definition 2. *A coding scheme $(\mathsf{Enc}, \mathsf{Dec})$ with message and codeword spaces as $\{0,1\}^l, \{0,1\}^n$ respectively, is ϵ- non-malleable with respect to a function family $\mathcal{F} \subseteq \{f : \{0,1\}^n \to \{0,1\}^n\}$ if $\forall\ f \in \mathcal{F}$, $\exists$ a distribution Sim_f over $\{0,1\}^l \cup \{same^*, \perp\}$ such that $\forall\ m \in \{0,1\}^l$*

$$\mathsf{Tamper}_f^m \approx_\epsilon \mathsf{Copy}_{Sim_f}^m$$

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

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

$$\mathsf{Copy}_{Sim_f}^m = \begin{cases} m \ if\ \tilde{m} = same^* \\ \tilde{m} \ \text{otherwise} \end{cases}$$

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

We now generalize the definition of 2-state augmented-NMC as defined in [AAG+16], to a j-augmented NMC for t-split state family, i.e., j of the t-states is also simulatable independent of the message (where $j < t$).

Definition 3. *A coding scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *with message and codeword spaces as* $\{0,1\}^\alpha, (\{0,1\}^\beta)^t$ *respectively, is* $[\epsilon, j]$*- augmented-non-malleable (where* $j < t$*) with respect to the function family* $\mathcal{F} = \{(f_1, \cdots, f_t) : f_i : \{0,1\}^\beta \to \{0,1\}^\beta\}$ *if* $\forall\ (f_1, \cdots, f_t) \in \mathcal{F}$*,* $\exists$ *a distribution* $Sim_{f_1,\cdots,f_t}$ *over* $(\{0,1\}^\beta)^j \times (\{0,1\}^\alpha \cup \{same^*, \perp\})$ *such that* $\forall\ m \in \{0,1\}^\alpha$

$$\mathsf{Tamper}^m_{f_1,\cdots,f_t} \approx_\epsilon \mathsf{Copy}^m_{Sim_{f_1,\cdots,f_t}}$$

where $\mathsf{Tamper}^m_{f,g}$ *denotes the distribution* $(X_{i_1}, \cdots, X_{i_j}, \mathsf{Dec}(f_1(X_1), \cdots, f_t(X_t)))$*, where* $\mathsf{Enc}(m) = (X_1, \cdots, X_t)$ *and* $(X_{i_1}, \cdots, X_{i_j})$ *represents some* j *states of the total* t *states.* $\mathsf{Copy}^m_{Sim_{f_1,\cdots,f_t}}$ *is defined as*

$$(X_{i_1}, \cdots, X_{i_j}, \tilde{m}) \leftarrow Sim_{f_1,\cdots,f_t}$$

$$\mathsf{Copy}^m_{Sim_{f_1,\cdots,f_t}} = \begin{cases} (X_{i_1}, \cdots, X_{i_j}, m)\ if\ (X_{i_1}, \cdots, X_{i_j}, \tilde{m}) = (X_{i_1}, \cdots, X_{i_j}, same^*) \\ (X_{i_1}, \cdots, X_{i_j}, \tilde{m})\ \mathsf{otherwise} \end{cases}$$

$Sim_{f_1,\cdots,f_t}$ *should be efficiently samplable given oracle access to* $(f_1, \cdots, f_t)(.)$*.*

3 Building blocks

We use information-theoretic message authentication codes, strong average case extractor and an augmented non-malleable code for 2-split-state family , as building blocks to our construction. We define these building blocks below.

3.1 One-Time Message Authentication Codes

A family of pair 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 $\mu - \mathsf{secure\ one\ time\ MAC}$ if

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

3.2 Average-case Extractors

Definition 4. *[DORS08, Section 2.5] Let* $Ext : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^l$ *be a polynomial time computable function. We say that Ext is an efficient average-case* (n, t, d, l, ϵ)*-strong extractor if for all pairs of random variables* (W, I) *such that* W *is an* n*-bit string satisfying* $\widetilde{\mathbf{H}}_\infty(W|I) \geq t$*, we have* $\mathbf{SD}\left((Ext(W; X), X, I), (U, X, I)\right)$ *, where* X *is uniform on* $\{0,1\}^d$*.*

4 Non-malleable Randomness Encoders

We now formally define non-malleable randomness encoding and give a construction for the same.

4.1 Definition

We first formalize the definition of a non-malleable randomness encoder. The goal is to argue that the original message looks random, even given the modified message. But, here the message and the codeword are both generated within the tampering experiment and the experiment outputs the message along with the modified message. This is where the non-malleability definition will defer from the regular NMC definition 2. We capture the goal by saying that, we are able to simulate the modified message, such that its joint distribution with a message chosen independently uniformly at random is statistically close to the tampering experiment's output. The case where the simulator outputs $same^*$ is a technicality, which we address in the definition below.

Definition 5. *Let* $(\mathsf{NMREnc}, \mathsf{NMRDec})$ *be s.t.* $\mathsf{NMREnc} : \{0,1\}^r \to \{0,1\}^k \times (\{0,1\}^{n_1} \times \{0,1\}^{n_2})$ *is defined as* $\mathsf{NMREnc}(r) = (\mathsf{NMREnc}_1(r), \mathsf{NMREnc}_2(r)) = (m, (x, y))$ *and* $\mathsf{NMRDec} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^k$.
We say that $(\mathsf{NMREnc}, \mathsf{NMRDec})$ *is a* ϵ*-non-malleable randomness encoder with message space* $\{0,1\}^k$ *and codeword space* $\{0,1\}^{n_1} \times \{0,1\}^{n_2}$*, for the distribution* $\mathcal{R}$ *on* $\{0,1\}^r$ *with respect to the* 2*-split-state family* $\mathcal{F}$ *if the following is satisfied:*

- ***Correctness****:*

$$\Pr_{r \leftarrow \mathcal{R}}[\mathsf{NMRDec}(\mathsf{NMREnc}_2(r)) = \mathsf{NMREnc}_1(r)] = 1$$

- ***Non-malleability****: For each* $(f, g) \in \mathcal{F}$*,* $\exists$ *a distribution* $\mathsf{NMRSim}_{f,g}$ *over* $\{0,1\}^k \cup \{same^*, \perp\}$ *such that*

$$\mathsf{NMRTamper}_{f,g} \approx_\epsilon Copy(U_k, \mathsf{NMRSim}_{f,g})$$

  where $\mathsf{NMRTamper}_{f,g}$ *denotes the distribution* $(\mathsf{NMREnc}_1(\mathcal{R}), \mathsf{NMRDec}((f, g)(\mathsf{NMREnc}_2(\mathcal{R})))$[1] *and* $Copy(U_k, \mathsf{NMRSim}_{f,g})$ *is defined as:*

$$u \leftarrow U_k;\ \tilde{m} \leftarrow \mathsf{NMRSim}_{f,g}$$

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

  $\mathsf{NMRSim}_{f,g}$ *should be efficiently samplable given oracle access to* $(f, g)(.)$.

Further, the rate of this code is defined as $k/(n_1 + n_2)$

[1] Here $(f, g)(\mathsf{NMREnc}_2(\mathcal{R}))$ just denotes the tampering by the split-state tampering functions f and g on the corresponding states.

While the non-malleability condition above, in flavor, resembles the seedless non-malleable extractors (the decoder function in the above protocol behaves like a seedless non-malleable extractor), the key difference is that, here the two states being tampered are correlated (through the encoder), while in a 2-source seedless NME, the sources need to be independent.

4.2 Notation

- $\mathsf{NMEnc}, \mathsf{NMDec}$ be an $[\varepsilon_1, 1]$-augmented-non-malleable code for 2-split state family over message and codeword spaces as $\{0,1\}^{\alpha}$, $\{0,1\}^{\beta_1} \times \{0,1\}^{\beta_2}$ respectively (as in Def 3), with the message length α and the length of the 2 states, β_1, β_2, respectively. $\mathsf{NMTamper}^{\mathsf{m}}_{\mathsf{f},\mathsf{g}}, \mathsf{NMSim}_{\mathsf{f},\mathsf{g}}$ denote the tampered message distribution of m and the simulator of $\mathsf{NMEnc}, \mathsf{NMDec}$ with respect to tampering functions (f, g)
- $\mathsf{Tag}', \mathsf{Vrfy}'$ be an information theoretic ε_2-secure one time MAC over key, message, tag spaces as $\{0,1\}^{\tau_1}, \{0,1\}^{n}, \{0,1\}^{\delta_1}$ respectively.
- Ext be an $(n, t, d, l + \tau, \varepsilon_3)$ average case strong extractor.

The parameters will be chosen such that $\alpha = \tau_1 + \delta_1 + d$ and $n > 2 + l + \tau + t$. (Refer to Section 4.5 for details)

4.3 Construction Overview

We now build a non-malleable randomness encoder, where the randomness is generated as the output of an extractor. To encode the seed, we use a regular 2-state aug-NMC. As mentioned in the introduction, in order to ensure that the source is not modified, when the seed is the same, we authenticate it using a MAC and encode the MAC key and tag along with the seed. In addition, to obtain a 2-state code, we combine the source with one of the states of the underlying aug-NMC.

$\mathsf{NMREnc}(r)$:

- Parse r as $s||w||k_{a_1}$
- $k_e||k_{a_2} = \mathsf{Ext}(w; s)$
- $t_1 = \mathsf{Tag}'_{k_{a_1}}(w)$
- $(L, R) \leftarrow \mathsf{NMEnc}(k_{a_1}||t_1||s)$
- O/P: $(k_e||k_{a_2}, (L||w, R))$

$\mathsf{NMRDec}(L||w, R)$:

- $k_{a_1}||t_1||s = \mathsf{NMDec}(L, R)$
- If $k_{a_1}||t_1||s = \perp$ output $\perp$
- else if $\mathsf{Vrfy}'_{k_{a_1}}(w, t_1) = 1$
  Output $\mathsf{Ext}(w, s)$
  else Output $\perp$

Theorem 1. *Let* $\mathsf{NMEnc}, \mathsf{NMDec}$ *be an* $[\varepsilon_1, 1]$*-augmented-non-malleable code for the* 2*-split state family,* $\mathsf{Tag}', \mathsf{Vrfy}'$ *be an information theoretic* ε_2*-secure one time* MAC *given above. Let* Ext *be an* $(n, t, d, l + \tau, \varepsilon_3)$ *average case strong extractor. Let* $\alpha = \tau_1 + \delta_1 + d$ *and* $n > 2 + l + \tau + t$.
Then $(\mathsf{NMREnc}, \mathsf{NMRDec})$ *is a non-malleable randomness encoding for the uniform distribution on* $\{0,1\}^{d+n+\tau_1}$*, with respect to the* 2*-split-state family.*
Further, the above construction can be instantiated, as in Section 4.5, to achieve

a constant rate of $\frac{1}{2+\zeta}$*, for any* $\zeta > 0$ *and an error of* $2^{-\Omega(l/\log^{\rho+1} l)}$*, for any* $\rho > 0$.

Proof. We give the proof in two steps. Firstly, we prove that the proposed construction is a non-malleable randomness encoding scheme (Section 4.4). Secondly, we set the parameters to achieve the desired rate and error (Section 4.5).

4.4 Security proof

Define the 2-split-state tampering family for the above construction as

$$\mathcal{F} = \{(f,g) : f : \{0,1\}^{\beta_1} \times \{0,1\}^n \to \{0,1\}^{\beta_1} \times \{0,1\}^n, g : \{0,1\}^{\beta_2} \to \{0,1\}^{\beta_2}\}$$

Correctness of the construction follows by its definition.
To show that $(\mathsf{NMREnc}, \mathsf{NMRDec})$ satisfies non-malleability, we need to show that $\forall\ (f,g) \in \mathcal{F}$, $\exists\ \mathsf{NMRSim}_{f,g}$ such that

$$\mathsf{NMRTamper}_{f,g} \approx_\varepsilon Copy(U_k, \mathsf{NMRSim}_{f,g}).$$

Let $f, g \in \mathcal{F}$. We define the simulator $\mathsf{NMRSim}_{f,g}$ as follows:

$\mathsf{NMRSim}_{f,g}$:

1. $w \in_R \{0,1\}^n$
2. $(L, \tilde{k_{a_1}}||\tilde{t_1}||\tilde{s}) \leftarrow \mathsf{NMSim}_{f_w,g}$ // where f_w is the function f with w hardcoded.
3. $\tilde{w} = f_L(w)$ // where f_L is the function f with L hardcoded.
4. If $\tilde{k_{a_1}}||\tilde{t_1}||\tilde{s} = same^*$:
   - If $\tilde{w} = w$ output $same^*$
   - else output $\perp$
5. Else if $\mathsf{Vrfy}'_{\tilde{k_{a_1}}}(\tilde{w}, \tilde{t_1}) = 1$ output $\mathsf{Ext}(\tilde{w}; \tilde{s})$
6. Else output $\perp$

We now prove the closeness of $\mathsf{NMRTamper}_{f,g}$ and $Copy(U_k, \mathsf{NMRSim}_{f,g})$ through a sequence of hybrids:

$\mathsf{NMRTamper}_{f,g}$:

1. $r \in_R \{0,1\}^{d+n+\tau_1}$;
   Parse r as $s||w||k_{a_1}$
2. $t_1 = \mathsf{Tag}'_{k_{a_1}}(w)$
3. $(L, \tilde{k_{a_1}}||\tilde{t_1}||\tilde{s}) \leftarrow \mathsf{NMTamper}^{k_{a_1}||t_1||s}_{f_w,g}$
4. $\tilde{w} = f_L(w)$
5. $k_e||k_{a_2} = \mathsf{Ext}(w; s)$
6. If $\mathsf{Vrfy}'_{\tilde{k_{a_1}}}(\tilde{w}, \tilde{t_1}) = 1$
   output $k_e||k_{a_2}, \mathsf{Ext}(\tilde{w}; \tilde{s})$
7. Else output $k_e||k_{a_2}, \perp$.

$\mathsf{Hybrid1}_{f,g}$:

1. $r \in_R \{0,1\}^{d+n+\tau_1}$
   Parse r as $s||w||k_{a_1}$
2. $t_1 = \mathsf{Tag}'_{k_{a_1}}(w)$
3. $(L, \tilde{k_{a_1}}||\tilde{t_1}||\tilde{s}) \leftarrow \mathsf{NMSim}_{f_w,g}$
   If $\tilde{k_{a_1}}||\tilde{t_1}||\tilde{s} = same^*$,
   set $\tilde{k_{a_1}}||\tilde{t_1}||\tilde{s} = k_{a_1}||t_1||s$
4. $\tilde{w} = f_L(w)$
5. $k_e||k_{a_2} = \mathsf{Ext}(w; s)$
6. If $\mathsf{Vrfy}'_{\tilde{k_{a_1}}}(\tilde{w}, \tilde{t_1}) = 1$
   output $k_e||k_{a_2}, \mathsf{Ext}(\tilde{w}; \tilde{s})$
   Else output $k_e||k_{a_2}, \perp$.

$\mathsf{Hybrid2}_{f,g}$:

1. $s||w \in_R \{0,1\}^{d+n}$
2. $(L, \tilde{k_{a_1}}||\tilde{t_1}||\tilde{s}) \leftarrow \mathsf{NMSim}_{f_w,g}$
3. $\tilde{w} = f_L(w)$
4. $k_e||k_{a_2} = \mathsf{Ext}(w; s)$
5. If $\tilde{k_{a_1}}||\tilde{t_1}||\tilde{s} = same^*$:
   - If $\tilde{w} = w$
     output $k_e||k_{a_2}, k_e||k_{a_2}$
   - else output $k_e||k_{a_2}, \perp$

   Else if $\mathsf{Vrfy}'_{\tilde{k_{a_1}}}(\tilde{w}, \tilde{t_1}) = 1$
   output $k_e||k_{a_2}, \mathsf{Ext}(\tilde{w}; \tilde{s})$
   Else output $k_e||k_{a_2}, \perp$

$\mathsf{Hybrid3}_{f,g}$:

1. $w \in_R \{0,1\}^n$
2. $(L, k_{a_1}||\tilde{t_1}||\tilde{s}) \leftarrow \mathsf{NMSim}_{f_w,g}$
3. $\tilde{w} = f_L(w)$
4. $k_e||k_{a_2} \in_R \{0,1\}^{l+\tau}$
5. If $\tilde{k_{a_1}}||\tilde{t_1}||\tilde{s} = same^*$:
   - If $\tilde{w} = w$
     output $k_e||k_{a_2}, k_e||k_{a_2}$
   - else output $k_e||k_{a_2}, \perp$

   Else if $\mathsf{Vrfy}'_{\tilde{k_{a_1}}}(\tilde{w}, \tilde{t_1}) = 1$
   output $k_e||k_{a_2}, \mathsf{Ext}(\tilde{w}; \tilde{s})$
   Else output $k_e||k_{a_2}, \perp$

Claim 1. If $(\mathsf{NMEnc}, \mathsf{NMDec})$ is a ε_1-augmented-non-malleable code, then $\mathsf{NMRTamper}_{f,g} \approx_{\varepsilon_1} \mathsf{Hybrid1}_{f,g}$.

Proof. By augmented non-malleability of $(\mathsf{NMEnc}, \mathsf{NMDec})$, we get

$$\mathsf{NMTamper}^{k_{a_1}||t_1||s}_{f_w,g} \approx_{\varepsilon_1} Copy^{k_{a_1}||t_1||s}_{\mathsf{NMSim}_{f_w,g}}$$

By using Lemma 3, we get

$$w, k_{a_1}||t_1||s, \mathsf{NMTamper}^{k_{a_1}||t_1||s}_{f_w,g} \approx_{\varepsilon_1} w, k_{a_1}||t_1||s, Copy^{k_{a_1}||t_1||s}_{\mathsf{NMSim}_{f_w,g}}$$

Now, the outputs of $\mathsf{NMRTamper}_{f,g}$ and $\mathsf{Hybrid1}_{f,g}$ are deterministic functions of above random variables. Hence, by Lemma 2, we get

$$\mathsf{NMRTamper}_{f,g} \approx_{\varepsilon_1} \mathsf{Hybrid1}_{f,g}$$

Claim 2. If $(\mathsf{Tag}', \mathsf{Vrfy}')$ is an information theoretic ε_2-secure one time MAC, then $\mathsf{Hybrid1}_{f,g} \approx_{\varepsilon_2} \mathsf{Hybrid2}_{f,g}$

Proof. If $same^*$ is not the value sampled from NMSim_{h_1,h_2}, then the output of the two hybrids are identical. Therefore, the statistical distance is zero in this case. When $same^*$ is sampled, the key difference between $\mathsf{Hybrid1}_{f,g}$ and $\mathsf{Hybrid2}_{f,g}$ is that, corresponding to this case, we remove the two verify checks in $\mathsf{Hybrid2}_{f,g}$ and simply replace it with the equality checks. Intuitively, in this case, the statistical closeness would hold due to unforgeability of MAC. The full proof can be found in Appendix A.1.

Claim 3. If Ext is an $(n, t, d, l + \tau, \varepsilon_3)$ average case extractor, then $\mathsf{Hybrid2}_{f,g} \approx_{\varepsilon_3} \mathsf{Hybrid3}_{f,g}$.[2]

Proof. We first consider the following random variables, which capture the auxiliary information. We then use extractor security and Lemma 2 to prove the closeness of the two hybrids.
We consider the output of $\mathsf{NMSim}_{f_w,g}$, which is $(L, \tilde{k_{a_1}}||\tilde{t_1}||\tilde{s})$ and define the following random variables, dependent on this:
We start with b_{same^*}, which indicates whether $\mathsf{NMSim}_{f_w,g}$ has output $same^*$ or not

$$b_{same^*} = \begin{cases} 1 & \text{if } \tilde{k_{a_1}}||\tilde{t_1}||\tilde{s} = same^* \\ 0 & \text{otherwise} \end{cases}$$

Further, $b_\perp$ is an indicator of whether $\mathsf{NMSim}_{f_w,g}$ output $\perp$ or not.

$$b_\perp = \begin{cases} 1 & \text{if } \tilde{k_{a_1}}||\tilde{t_1}||\tilde{s} = \perp \\ 0 & \text{otherwise} \end{cases}$$

We also have:

$$eq(w) = \begin{cases} 0 \text{ if } & f_L(w) \neq w \\ 1 \text{ if } & f_L(w) = w \end{cases}$$

which is an indicator of whether $\tilde{w}$ is modified or not. And,

$$Verify(w) = \mathsf{Vrfy}'_{\tilde{k_{a_1}}}(f_L(w), \tilde{t_1})$$

which is the indicator of the MAC verification bit.
Further define:

$$Y(w, b_1, b_2) := \begin{cases} eq(w) & \text{if } b_1 = 1 \\ (Verify(w), \mathsf{Ext}(\tilde{w}; \tilde{s})) & \text{if } b_1 = 0 \wedge b_2 = 0 \\ \perp & \text{otherwise} \end{cases}$$

We now define the auxiliary information by $\hat{E} = (b_{same^*}, b_\perp, Y(W, b_{same^*}, b_\perp))$.
We now define the following function
$G(e, k)$:

[2] We refer the reader to Appendix A.2 for an alternate proof of this claim

– Parse $e = (b_{same^*}, b_\perp, y = Y(w, b_{same^*}, b_\perp))$[3].
– If $b_{same^*} = 1$:
  • If $y = 1$, output (k, k)
  • Else output $(k, \perp)$
– Else:
  • If $b_\perp = 1$, output $(k, \perp)$.
  • Else parse $y = (Verify(w), \mathsf{Ext}(\tilde{w}; \tilde{s}))$.
    ∗ if $Verify(w) = 1$ output $(k, \mathsf{Ext}(\tilde{w}; \tilde{s}))$
    ∗ else output $(k, \perp)$

The outputs of $\mathsf{Hybrid2}_{f,g}$ and $\mathsf{Hybrid3}_{f,g}$ are $G(\hat{E}, \mathsf{Ext}(W; S))$ and $G(\hat{E}, U_{l+\tau})$ respectively, where G is deterministic. So, to prove this claim it suffices to show

$$\hat{E}, \mathsf{Ext}(W; S) \approx_{\varepsilon_3} \hat{E}, U_{l+\tau} \tag{1}$$

Observe that $\hat{E}$ depends on $\mathsf{NMSim}_{f_w,g}$ and w, which are independent of the seed s. Therefore it can be captured as auxiliary information. $\hat{E}$ takes at most $2^{3+l+\tau}$ possible values. Hence, $\tilde{\mathbf{H}}_\infty(W|\hat{E}) \geq \mathbf{H}_\infty(W) - (3 + l + \tau) = n - (3 + l + \tau)$, by Lemma 4. As $n - (3 + l + \tau) > t$ (due to the way we set parameters in section 4.5), by security of average case extractor, Equation 1 holds. This proves the claim.

From above Claims 1,2 and 3, we get:

$$\mathsf{NMRTamper}_{f,g} \approx_{\varepsilon_1} \mathsf{Hybrid1}_{f,g} \approx_{\varepsilon_2} \mathsf{Hybrid2}_{f,g} \approx_{\varepsilon_3} \mathsf{Hybrid3}_{f,g} \equiv Copy(U_k, \mathsf{NMRSim}_{f,g})$$

$$\text{i.e., } \mathsf{NMRTamper}_{f,g} \approx_{\varepsilon_1+\varepsilon_2+\varepsilon_3} Copy(U_k, \mathsf{NMRSim}_{f,g})$$

4.5 Rate and Error analysis

We now present the details of the rate of the code as well as the error it achieves. We instantiate the above construction using specific MAC construction, average case extractor Ext and non-malleable code $(\mathsf{NMEnc}, \mathsf{NMDec})$, as given in the lemmata below.
As we are encoding the seed of the extractor using the underlying non-malleable code, it is important that the strong extractor we use has short seed length. This is guaranteed by the following lemma.

Lemma 5. *[GUV07] For every constant $\nu > 0$ all integers $n \geq t$ and all $\epsilon \geq 0$, there is an explicit (efficient) $(n, t, d, l, \epsilon)-$strong extractor with $l = (1 - \nu)t - \mathcal{O}(\log(n) + \log(\frac{1}{\epsilon}))$ and $d = \mathcal{O}(\log(n) + \log(\frac{1}{\epsilon}))$.*

Now, as we give some auxiliary information about the source, we require the security of the extractor to hold, even given this information. Hence, we use average case extractors, given in the following lemma.

[3] Here, we abuse the notation: b_{same^*} and $b_\perp$ represent the particular values taken by the corresponding random variables

Lemma 6. *[DORS08] For any $\mu > 0$, if* Ext *is a (worst case)$(n, t, d, l, \epsilon)-$strong extractor, then* Ext *is also an average-case $(n, t + \log(\frac{1}{\mu}), d, l, \epsilon + \mu)$ strong extractor.*

We now combine the Lemmata 5 and 6 to get an average case extractor with optimal seed length.

Corollary 1. *For any $\mu > 0$ and every constant $\nu > 0$ all integers $n \geq t$ and all $\epsilon \geq 0$, there is an explicit (efficient) $(n, t+\log(\frac{1}{\mu}), d, l, \epsilon+\mu)-$ average case strong extractor with $l = (1 - \nu)t - \mathcal{O}(\log(n) + \log(\frac{1}{\epsilon}))$ and $d = \mathcal{O}(\log(n) + \log(\frac{1}{\epsilon}))$.*

Now, we also encode the authentication keys and tags using the underlying non-malleable code. Hence, we require them to have short lengths. This is guaranteed by the following lemma (as given by Johansson, Kabatianskii, Smeets):

Lemma 7. *For any $n', \varepsilon_2 > 0$ there is an efficient ε_2-secure one time* MAC *with $\delta \leq (\log(n') + \log(\frac{1}{\varepsilon_2}))$, $\tau \leq 2\delta$, where τ, n', δ are key, message, tag length respectively.*

Further, we use the 2-split-state non-malleable code by [Li17] to instantiate our construction.

Lemma 8. *[Li17, Theorem 7.12] For any $\beta \in \mathbb{N}$ there exists an explicit non-malleable code with efficient encoder/decoder in 2-split state model with block length 2β, rate $\Omega\left(\frac{1}{\log\ \beta}\right)$ and error $= 2^{-\Omega\left(\frac{\beta}{\log\ \beta}\right)}$*

Further, we show in Appendix B (Corollary 2) that the construction corresponding to Lemma 8 is in fact an $[2^{-\Omega(\beta/\log\ \beta)}, 1]$-augmented-non-malleable code for the two split-state family with the same rate as above.

4.5.1 Setting parameters We instantiate our construction using $(\mathsf{NMEnc}, \mathsf{NMDec})$ as in Corollary 2, strong average case extractors, as in Corollary 1 and one time information theoretic MAC, as in Lemma 7.

- We set the error parameters as $\epsilon, \mu, \varepsilon_1, \varepsilon_2 = 2^{-\lambda}$ and $\varepsilon_3 = \epsilon + \mu$.
- The message length and codeword length in the construction of $(\mathsf{NMREnc}, \mathsf{NMRDec})$ above, are $l + \tau$ and $2\beta + n$ respectively. Here we take k_{a_2} to be of size $\tau = \mathcal{O}(\log l + \lambda)$.
- We estimate the length of the source (n). As we saw in the Claim 3 of the proof (Section 4.4), we leak auxiliary information of length at most $3 + l + \tau$. Hence, by Lemma 4, the average entropy of the source, given auxiliary information is $\geq n - (3 + l + \tau)$.

  To use extractor security, we require that the average entropy is at least the

entropy threshold $t + log(\frac{1}{\mu})$, i.e., $n - (3 + l + \tau) \geq t + log(\frac{1}{\mu})$.
By Corollary 1 (with output length of extractor $l + \tau$), we have
$t = (l + \tau + \mathcal{O}(\log(n) + \log(\frac{1}{\epsilon})))\frac{1}{1-\nu}$.
Hence, taking ν as a very small constant close to 0, we get: for some constant ζ close to 0

$$n = (2 + \zeta)l + \mathcal{O}(\log l + \lambda) \tag{2}$$

– We now estimate the codeword length 2β, of the underlying NMC.
The message size for this codeword is $\alpha = \tau_1 + \delta_1 + d$. By Lemma 7 and Corollary 1, we get $\alpha = \mathcal{O}(\log(l) + \lambda)$.
By using the rate in Lemma 8, we get:

$$\beta = \mathcal{O}((\log(l))^2 + \lambda \log(\lambda) + 2\lambda \log(l)) \tag{3}$$

4.5.2 Rate The rate of our construction of non-malleable randomness encoding is:

$$R = \frac{l + \tau}{2\beta + n}$$

By substituting n and β from Equations 2 and 3, respectively and τ as described above, we get:

$$R = \frac{l + \mathcal{O}(\log l + \lambda)}{\mathcal{O}((\log(l))^2 + \lambda \log(\lambda) + 2\lambda \log(l)) + (2 + \zeta)l + \mathcal{O}(\log l + \lambda)}$$

For large l, and taking $\lambda = o(\frac{l}{\log l})$, we get

$$R \geq \frac{1}{2 + \zeta}$$

Hence, the construction given achieves rate atleast $\frac{1}{2+\zeta}$, for some ζ close to 0.

4.5.3 Error Error of the protocol, as seen in the proof, is $\varepsilon_1 + \varepsilon_2 + \varepsilon_3 = 4(2^{-\lambda})$. Since, $\lambda = o(\frac{l}{\log l})$, the error will be at least $2^{-\frac{l}{\log l}}$. For any $\rho > 0$, fixing $\lambda = \frac{l}{\log^{\rho+1} l}$, the error would be at most $4.2^{-\frac{l}{\log^{\rho+1} l}}$. Setting $\kappa = \lambda - \log 5$ the error would be $2^{-\kappa} = 2^{-\Omega(l/\log^{\rho+1} l)}$.

5 Non-malleable Codes from Non-malleable Randomness Encoders

As an application of non-malleable randomness encoding, we build a 3-state 1-augmented-non-malleable code, using non-malleable randomness encoding in black-box. For achieving an explicit constant rate and a specific error, we instantiate the construction using the construction in Section 4.

5.1 Construction overview

To encode the message, we first hide the message using one part of the randomness generated in the underlying NMRE. To ensure that this ciphertext is not modified, we authenticate it using a MAC. We show that we can use NMRE's "random messages" as the keys for encryption as well authentication. The fact that the tag t does not need to be non-malleably encoded, and can instead be combined with c, is what allows us to get a 3-state NMC construction while only making a **black-box** use of the underlying NMRE. Details follow.

$\mathsf{AEnc}(m)$	$\mathsf{ADec}(\tilde{y}_1, \tilde{y}_2, \tilde{c}\|\|\tilde{t})$
– $r \in_R \{0,1\}^{r'}$ – $(k_a\|\|k_e, y_1, y_2) \leftarrow \mathsf{NMREnc}(r)$ – $c = m \oplus k_e$ – $t = \mathsf{Tag}_{k_a}(c)$ – Output $(y_1, y_2, c\|\|t)$	– $\tilde{k}_e\|\|\tilde{k}_a = \mathsf{NMRDec}(\tilde{y}_1, \tilde{y}_2)$ – If $\mathsf{Vrfy}_{\tilde{k}_a}(\tilde{c}, \tilde{t}) = 1$ Output $\tilde{c} \oplus \tilde{k}_e$ else Output $\perp$

Theorem 2. *Let* $(\mathsf{NMREnc}, \mathsf{NMRDec})$ *be a 2-state* ϵ_1*- non-malleable randomness encoding scheme for the uniform distribution on* $\{0,1\}^{r'}$*, for messages in* $\{0,1\}^{l+\tau}$ *and* $(\mathsf{Tag}, \mathsf{Vrfy})$ *be an information theoretic* ϵ_2*-secure one-time* MAC *with key, message and tag spaces being* $\{0,1\}^{\tau}, \{0,1\}^{l}, \{0,1\}^{\delta}$*. Then* $(\mathsf{AEnc}, \mathsf{ADec})$*, as defined above, is a 3-state* $[\epsilon_1 + \epsilon_2,\ 1]$*-augmented non-malleable code for messages of length* l *(with the augmented state being* $c||t$*).*
Further, instantiating the construction with $(\mathsf{NMREnc}, \mathsf{NMRDec})$ *achieving rate and error, as in Section 4.5, we can achieve a constant rate of* $\frac{1}{3+\zeta}$*, for any* $\zeta > 0$ *and an error of* $2^{-\Omega(l/\log^{\rho+1} l)}$*, for any* $\rho > 0$*.*

5.2 Security Proof

Let $(f_1, f_2, g) \in \mathcal{F}_3$(3-split state tampering family) where $f_1 : \{0,1\}^{\beta_1} \rightarrow \{0,1\}^{\beta_1}, f_2 : \{0,1\}^{\beta_2} \rightarrow \{0,1\}^{\beta_2}, g : \{0,1\}^{l+\delta} \rightarrow \{0,1\}^{l+\delta}$. We propose the following distribution as simulator for $(AEnc, ADec)$.

$\mathsf{ASim}_{f_1,f_2,g}$

- $k_e||k_a \in_R \{0,1\}^{l+\tau}$
- $\tilde{k}_e||\tilde{k}_a \leftarrow \mathsf{NMRSim}_{f_1,f_2}$
- $c = 0 \oplus k_e$
- $t = \mathsf{Tag}_{k_a}(c)$
- $\tilde{c}||\tilde{t} = g(c||t)$
- If $\tilde{k}_e||\tilde{k}_a = same^*$
  - If $\tilde{c} = c$, Output $c, t, same^*$
  - Else output $c, t, \perp$
  - Else if $\mathsf{Vrfy}_{\tilde{k_a}}(\tilde{c}, \tilde{t}) = 1$
  - Output $c, t, \tilde{c} \oplus \tilde{k}_e$
  - Else Output $c, t, \perp$

We prove that $ASim_{f_1,f_2,g}$ is the simulator of $(AEnc, ADec)$ through a sequence of hybrids.

$\mathsf{ATamper}^{m}_{f_1,f_2,g}$

- $k_e||k_a, \tilde{k}_e||\tilde{k}_a \leftarrow \mathsf{NMRTamper}_{f_1,f_2}$
- $c = m \oplus k_e$
- $t = \mathsf{Tag}_{k_a}(c)$
- $\tilde{c}||\tilde{t} = g(c||t)$
- If $\mathsf{Vrfy}_{\tilde{k_a}}(\tilde{c}, \tilde{t}) = 1$
  Output $c, t, \tilde{c} \oplus \tilde{k}_e$
  Else Output $c, t, \perp$

$\mathsf{Hybrid1}^{m}_{f_1,f_2,g}$

- $k_e||k_a \in_R \{0,1\}^{l+\tau}$
- $\tilde{k}_e||\tilde{k}_a \leftarrow \mathsf{NMRSim}_{f_1,f_2}$
- If $\tilde{k}_e||\tilde{k}_a = same^*$
  set $\tilde{k}_e||\tilde{k}_a = k_e||k_a$
- $c = m \oplus k_e$
- $t = \mathsf{Tag}_{k_a}(c)$
- $\tilde{c}||\tilde{t} = g(c||t)$
- If $\mathsf{Vrfy}_{\tilde{k_a}}(\tilde{c}, \tilde{t}) = 1$
  Output $c, t, \tilde{c} \oplus \tilde{k}_e$
  Else Output $c, t, \perp$

$\mathsf{Hybrid2}^{m}_{f_1,f_2,g}$

- $k_e||k_a \in_R \{0,1\}^{l+\tau}$
- $\tilde{k}_e||\tilde{k}_a \leftarrow \mathsf{NMRSim}_{f_1,f_2}$
- $c = m \oplus k_e$
- $t = \mathsf{Tag}_{k_a}(c)$
- $\tilde{c}||\tilde{t} = g(c||t)$
- If $\tilde{k}_e||\tilde{k}_a = same^*$
  set $\tilde{k}_e||\tilde{k}_a = k_e||k_a$
- If $\mathsf{Vrfy}_{\tilde{k_a}}(\tilde{c}, \tilde{t}) = 1$
  Output $c, t, \tilde{c} \oplus \tilde{k}_e$
  Else Output $c, t, \perp$

$\mathsf{Hybrid3}^{m}_{f_1,f_2,g}$

- $k_e||k_a \in_R \{0,1\}^{l+\tau}$
- $\tilde{k}_e||\tilde{k}_a \leftarrow \mathsf{NMRSim}_{f_1,f_2}$
- $c = m \oplus k_e$
- $t = \mathsf{Tag}_{k_a}(c)$
- $\tilde{c}||\tilde{t} = g(c||t)$
- If $\tilde{k}_e||\tilde{k}_a = same^*$
  If $\tilde{c} = c$ Output c, t, m
  Else output $c, t, \perp$
  Else if $\mathsf{Vrfy}_{\tilde{k_a}}(\tilde{c}, \tilde{t}) = 1$
  Output $c, t, \tilde{c} \oplus \tilde{k}_e$
  Else Output $c, t, \perp$

$\mathsf{Hybrid4}^{m}_{f_1,f_2,g}$

- $k_e||k_a \in_R \{0,1\}^{l+\tau}$
- $\tilde{k}_e||\tilde{k}_a \leftarrow \mathsf{NMRSim}_{f_1,f_2}$
- $c = 0 \oplus k_e$
- $t = \mathsf{Tag}_{k_a}(c)$
- $\tilde{c}||\tilde{t} = g(c||t)$
- If $\tilde{k}_e||\tilde{k}_a = same^*$
  If $\tilde{c} = c$ Output c, t, m
  Else output $c, t, \perp$
  Else if $\mathsf{Vrfy}_{\tilde{k_a}}(\tilde{c}, \tilde{t}) = 1$
  Output $c, t, \tilde{c} \oplus \tilde{k}_e$
  Else Output $c, t, \perp$

Claim 1 *If* $(\mathsf{NMREnc}, \mathsf{NMRDec})$ *is a non-malleable randomness encoding scheme, then* $\mathsf{ATamper}^{m}_{f_1,f_2,g} \approx_{\epsilon_1} \mathsf{Hybrid1}^{m}_{f_1,f_2,g}$.

Proof. By non-malleability of $(\mathsf{NMREnc}, \mathsf{NMRDec})$, we have

$$\mathsf{NMRTamper}_{f_1,f_2} \approx_{\epsilon_1} Copy(U_k, \mathsf{NMRSim}_{f_1,f_2})$$

As m is independent we have

$$m, \mathsf{NMRTamper}_{f_1,f_2} \approx_{\epsilon_1} m, Copy(U_k, \mathsf{NMRSim}_{f_1,f_2})$$

By Lemma 2 we have,

$$m, c, t, \mathsf{NMRTamper}_{f_1,f_2} \approx_{\epsilon_1} m, c, t, Copy(U_k, \mathsf{NMRSim}_{f_1,f_2})$$

The outputs of $\mathsf{ATamper}^m_{f_1,f_2,g}, \mathsf{Hybrid1}^m_{f_1,f_2,g}$ are determined by a deterministic function of above distributions. Therefore by Lemma 2 we have

$$m, \mathsf{ATamper}^m_{f_1,f_2,g} \approx_{\epsilon_1} m, \mathsf{Hybrid1}^m_{f_1,f_2,g}$$

Claim 2 $\mathsf{Hybrid1}^m_{f_1,f_2,g} \equiv \mathsf{Hybrid2}^m_{f_1,f_2,g}$

Proof. The claim trivially follows because $\mathsf{Hybrid2}^m_{f_1,f_2,g}$ is rewriting of $\mathsf{Hybrid1}^m_{f_1,f_2,g}$.

$$k_e, k_a, \mathsf{NMRSim}_{f_1,f_2} \equiv k_e, k_a, \mathsf{NMRSim}_{f_1,f_2}$$

$$m, c, t, k_e, k_a, \mathsf{NMRSim}_{f_1,f_2} \equiv m, c, t, k_e, k_a, \mathsf{NMRSim}_{f_1,f_2}$$

$$m, \mathsf{Hybrid1}^m_{f_1,f_2,g} \equiv m, \mathsf{Hybrid2}^m_{f_1,f_2,g}$$

All equations follow by Lemma 2

Claim 3 *If* $(\mathsf{Tag}, \mathsf{Vrfy})$ *is an* ϵ_2 *IT-secure-One-time Mac, then* $\mathsf{Hybrid2}^m_{f_1,f_2,g} \approx_{\epsilon_2} \mathsf{Hybrid3}^m_{f_1,f_2,g}$

Proof. Let E denote the event $\tilde{k_e}, \tilde{k_a} \neq same^*$, and $\tilde{E}$, its compliment. Given E, both the hybrids are identical. Given $\tilde{E}$ the statistical distance of the hybrids is at most

$$\Pr_{k_a}[\mathsf{Vrfy}_{k_a}(\tilde{c}, \tilde{t}) = 1 | t = \mathsf{Tag}_{k_a}(c), \tilde{c} || \tilde{t} = f(c||t)] \leq \epsilon_2$$

Therefore claim follows.

Claim 4 *By semantic security of One Time Pad encryption*

$$\mathsf{Hybrid3}^m_{f_1,f_2,g} \equiv \mathsf{Hybrid4}^m_{f_1,f_2,g}$$

Proof. By semantic security,

$$m, m \oplus k_e \equiv m, 0 \oplus k_e$$

$$m, t, m \oplus k_e, k_a \equiv m, t, 0 \oplus k_e, k_a$$

The outputs of the hybrids 3 and 4 are a randomized function of above distributions. Therefore

$$\mathsf{Hybrid3}^m_{f_1,f_2,g} \equiv \mathsf{Hybrid4}^m_{f_1,f_2,g} \equiv Copy^m_{Asim_{f_1,f_2,g}}$$

Combining the above Claims 1,2,3 and 4, we have

$$\mathsf{ATamper}^m_{f_1,f_2,g} \approx_{\epsilon_1+\epsilon_2} Copy^m_{Asim_{f_1,f_2,g}}$$

5.3 Rate and error analysis

From Section 4.5, we have a non-malleable randomness encoding $(\mathsf{NMREnc}, \mathsf{NMRDec})$ with a constant rate of $R \geq \frac{1}{2+\zeta}$, for any $\zeta > 0$ and an error of $\epsilon_1 = 2^{-\Omega(l/\log^{\rho+1} l)}$, for any $\rho > 0$.

5.3.1 Rate The rate of $(\mathsf{AEnc}, \mathsf{ADec})$ is:

$$R' = \frac{l}{\frac{1}{R}.(l+\tau)+l+\delta} = \frac{l}{(2+\zeta).(l+\tau)+l+\delta}$$

where, δ is size of tag t. Hence,

$$R' = \frac{l}{(3+\zeta)l+(2+\zeta)\tau+\delta}$$

By using Lemma 7, we know that for $\lambda = o(l/\log l)$, we get $\tau + \delta \leq 3(\log l + o(l/\log l))$. Hence, we get, for large l:

$$R' \geq \frac{1}{3+\zeta}$$

5.3.2 Error By setting $\epsilon_2 = 2^{-\lambda}$, we get that the error of $(\mathsf{AEnc}, \mathsf{ADec})$ is $\epsilon_1 + \epsilon_2 = 2^{-\Omega(l/\log^{\rho+1} l)}$, for any $\rho > 0$.

6 Conclusion

In this work, we introduced Non-malleable Randomness Encoders as a relaxation of NMCs, applicable in settings where randomness is encoded. We built a 1/2-rate, 2-state NMRE. In cases where NMREs suffice, this presents a significant advantage over using a poor-rate 2-state NMC. It would be interesting to find other applications of NMREs in addition to the ones presented in this paper i.e., to tamper-resilient security and to building 3-state (standard) with rate $\frac{1}{3}$ in a black-box. (Infact, our techniques can be generalized to show that $(t+1)$-state augmented NMCs can be constructed from t-state NMREs in black box manner.) While we know that the optimal achievable rate for 2-state NMCs is 1/2, it would be interesting to see what the optimal achievable rate for 2-state NMREs is and, more generally, for t-state NMREs. Of course, the crux of this long, compelling line of research, which is to build constant rate efficient 2-state NMCs, still remains open and would be fascinating to solve.

Acknowledgement

We thank Eshan Chattopadhyay for helpful discussions on connections between non-malleable codes and extractors.

References

AAG+16. Divesh Aggarwal, Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. 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*, pages 393–417, 2016.

ADKO15. Divesh Aggarwal, Yevgeniy Dodis, Tomasz Kazana, and Maciej Obremski. 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*, pages 459–468, 2015.

ADL14. Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 774–783, 2014.

AGM+15. Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. 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*, pages 375–397, 2015.

AKO15. Divesh Aggarwal, Tomasz Kazana, and Maciej Obremski. Inception makes non-malleable codes stronger. *IACR Cryptology ePrint Archive*, 2015:1013, 2015.

CG14a. Mahdi Cheraghchi and Venkatesan Guruswami. Capacity of non-malleable codes. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 155–168, 2014.

CG14b. Mahdi Cheraghchi and Venkatesan Guruswami. 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*, pages 440–464, 2014.

CKR16. Nishanth Chandran, Bhavana Kanukurthi, and Srinivasan Raghuraman. 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*, pages 367–392, 2016.

CZ14. Eshan Chattopadhyay and David Zuckerman. 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*, pages 306–315, 2014.

DKO13. Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. 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*, pages 239–257, 2013.

DKS17. Dana Dachman-Soled, Mukul Kulkarni, and Aria Shahverdi. Tight upper and lower bounds for leakage-resilient, locally decodable and updatable non-malleable codes. *IACR Cryptology ePrint Archive*, 2017:15, 2017.

DLSZ14. Dana Dachman-Soled, Feng-Hao Liu, Elaine Shi, and Hong-Sheng Zhou. Locally decodable and updatable non-malleable codes and their applications. *IACR Cryptology ePrint Archive*, 2014:663, 2014.

DNO17. Nico Döttling, Jesper Buus Nielsen, and Maciej Obremski. Information theoretic continuously non-malleable codes in the constant split-state model. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:78, 2017.

DORS08. Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008. arXiv:cs/0602007.

DPW10. Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 434–452, 2010.

FMNV14. Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. Continuous non-malleable codes. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, pages 465–488, 2014.

GUV07. Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. In *IEEE Conference on Computational Complexity*, pages 96–108, 2007.

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

KOS17. Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, and Sruthi Sekar. Four-state non-malleable codes with explicit constant rate. In *Theory of Cryptography Conference, TCC 2017, Full version can be found at http://www.csa.iisc.ernet.in/ crysp/files/KOStcc2017.pdf*, 2017.

Li17. Xin Li. 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*, 2017.

LL12. Feng-Hao Liu and Anna Lysyanskaya. Tamper and leakage resilience in the split-state model. *IACR Cryptology ePrint Archive*, 2012:297, 2012.

A Proofs of Claims 2 and 3 in Section 4.4

A.1 Proof of Claim 2 in Section 4.4

We define the following events:

- Let E be the event that $same^*$ is sampled from $\mathsf{NMSim}_{f_w,g}$ and $\tilde{E}$ be its compliment.
- Let F be the event that $\tilde{w} = w$ and $\tilde{F}$ its complement.

By Proposition 1 we get:

$$\mathbf{SD}\left(\mathsf{Hybrid1}_{f,g}; \mathsf{Hybrid2}_{f,g}\right) = \Pr[E] \cdot \mathbf{SD}\left(\mathsf{Hybrid1}_{f,g}|E; \mathsf{Hybrid2}_{f,g}|E\right) + \Pr[\tilde{E}] \cdot \underbrace{\mathbf{SD}\left(\mathsf{Hybrid1}_{f,g}|\tilde{E}; \mathsf{Hybrid2}_{f,g}|\tilde{E}\right)}_{=0 \text{ The hybrids are identical in "not } same^* \text{case}}$$

So, now remains the case when $\mathsf{NMSim}_{f_w,g}$ outputs $same^*$. By using unforgeability of $(\mathsf{Tag}', \mathsf{Vrfy}')$ we show the that two hybrids are statistically close.
2. $\Pr[E].\mathbf{SD}\left(\mathsf{Hybrid1}_{f,g}|E; \mathsf{Hybrid2}_{f,g}|E\right)$

$$= \sum_{m \in \{0,1\}^{l+\tau}} \sum_{\tilde{m} \in \{0,1\}^{l+\tau} \cup \{\perp\}} \Pr[E] \Big| \Pr[\mathsf{Hybrid1}_{f,g} = (m, \tilde{m})|E] - \Pr[\mathsf{Hybrid2}_{f,g} = (m, \tilde{m})|E] \Big|$$

$$= \Pr[E] \sum_{m \in \{0,1\}^{l+\tau}} \sum_{\tilde{m} \in \{0,1\}^{l+\tau} \cup \{\perp\}} \Big| \Pr[F|E].$$

$$\Big(\underbrace{\Pr[\mathsf{Hybrid1}_{f,g} = (m, \tilde{m})|E, F] - \Pr[\mathsf{Hybrid2}_{f,g} = (m, \tilde{m})|E, F]}_{=0 \text{ as given E and F both the hybrids are identical}} \Big) + \Pr[\tilde{F}|E].$$

$$\Big(\Pr[\mathsf{Hybrid1}_{f,g} = (m, \tilde{m})|E, \tilde{F}] - \Pr[\mathsf{Hybrid2}_{f,g} = (m, \tilde{m})|E, \tilde{F}] \Big) \Big|$$

$$= \Pr[E] \sum_{m \in \{0,1\}^{l+\tau}} \sum_{\tilde{m} \in \{0,1\}^{l+\tau} \cup \{\perp\}} \Bigg| \Pr[\tilde{F}|E] \Big(\Pr[\mathsf{Hybrid1}_{f,g} = (m, \tilde{m})|E, \tilde{F}] - \Pr[\mathsf{Hybrid2}_{f,g} = (m, \tilde{m})|E, \tilde{F}] \Big) \Bigg|$$

$$= \Pr[E] \Pr[\tilde{F}|E]$$

$$\Bigg(\sum_{m \in \{0,1\}^{l+\tau}} \sum_{\tilde{m} \in \{0,1\}^{l+\tau}} \Bigg| \Pr[\mathsf{Hybrid1}_{f,g} = (m, \tilde{m})|E, \tilde{F}] - \underbrace{\Pr[\mathsf{Hybrid2}_{f,g} = (m, \tilde{m})|E, \tilde{F}]}_{= 0 \text{ as given E,} \tilde{F} \text{ Hybrid 2 outputs } \perp \text{ as second component}} \Bigg|$$

$$+ \sum_{m \in \{0,1\}^{l+\tau}} \Bigg| \Pr[\mathsf{Hybrid1}_{f,g} = (m, \perp)|E, \tilde{F}] - \Pr[\mathsf{Hybrid2}_{f,g} = (m, \perp)|E, \tilde{F}] \Bigg| \Bigg)$$

$$= \Pr[\tilde{F}] \Bigg(1 + \sum_{m \in \{0,1\}^{l+\tau}} \Bigg(\Bigg(\sum_{\tilde{m} \in \{0,1\}^{l+\tau}} \Pr[\mathsf{Hybrid1}_{f,g} = (m, \tilde{m})|E, \tilde{F}] \Bigg) - \Pr[\mathsf{Hybrid1}_{f,g} = (m, \perp)|E, \tilde{F}] \Bigg) \Bigg)$$

$$= 2 \Pr[\tilde{F}] \Big(\Pr[\text{Second component of output of } \mathsf{Hybrid1}_{f,g} \neq \perp | E, \tilde{F}] \Big)$$

$$= 2 \Pr[\tilde{F}] \Pr[\mathsf{Vrfy}_{\tilde{k_{a_1}}}(\tilde{w}, \tilde{t_1}) = 1 \wedge t_1 = \mathsf{Tag}_{k_{a_1}}(w)|E, \tilde{F}]$$

$$= 2 \Pr[\tilde{F}] \Pr[\mathsf{Vrfy}_{k_{a_1}}(\tilde{w}, t_1) = 1 \wedge t_1 = \mathsf{Tag}_{k_{a_1}}(w)|\tilde{F}]$$

$$\leq 2(\varepsilon_2)$$

$$\therefore \mathsf{Hybrid1}_{f,g} \approx_{\varepsilon_2} \mathsf{Hybrid2}_{f,g}$$

A.2 Alternate proof of Claim 3 in Section 4.4

Claim 3. If Ext is an $(n, t, d, l + \tau, \varepsilon_3)$ average case extractor, then $\mathsf{Hybrid2}_{f,g} \approx_{\varepsilon_3} \mathsf{Hybrid3}_{f,g}$.

Proof. As the function modifying the state L, f_w, is dependent on W, hence $\mathsf{NMSim}_{f_w,g}$ is also dependent on W. Hence, before analyzing the auxiliary information leaked in each case, corresponding to the value of $\mathsf{NMSim}_{f_w,g}$, we define the following indicator random variables, which are also auxiliary information, w.r.t. to the source W:

$$b_{same^*} = \begin{cases} 1 & \text{if } \tilde{k_{a_1}}||\tilde{t_1}||\tilde{s} = same^* \\ 0 & \text{otherwise} \end{cases}$$

$$b_\perp = \begin{cases} 1 & \text{if } \tilde{k_{a_1}}||\tilde{t_1}||\tilde{s} = \perp \\ 0 & \text{otherwise} \end{cases}$$

By Proposition 1, we get:
$\mathbf{SD}\left(\mathsf{Hybrid2}_{f,g}, \mathsf{Hybrid3}_{f,g}\right)$

$$\begin{aligned} &\leq Pr[b_{same^*} = 1]\ \mathbf{SD}\left(\mathsf{Hybrid2}_{f,g}|b_{same^*} = 1, \mathsf{Hybrid3}_{f,g}|b_{same^*} = 1\right) \\ &+ Pr[b_{same^*} = 0 \wedge b_\perp = 1]\ \mathbf{SD}\left(\mathsf{Hybrid2}_{f,g}|b_{same^*} = 0 \wedge b_\perp = 1, \mathsf{Hybrid3}_{f,g}|b_{same^*} = 0 \wedge b_\perp = 1\right) \\ &+ Pr[b_{same^*} = 0 \wedge b_\perp = 0]\ \mathbf{SD}\left(\mathsf{Hybrid2}_{f,g}|b_{same^*} = 0 \wedge b_\perp = 0, \mathsf{Hybrid3}_{f,g}|b_{same^*} = 0 \wedge b_\perp = 0\right) \end{aligned} \tag{4}$$

Now, in order to analyze the auxiliary information leaked in each of the three cases, and use the extractor security, we first consider the conditional distribution on W, when conditioned on each of the three cases. We denote the three conditional distributions by: $W_1 := W|b_{same^*} = 1$, $W_2 := W|b_{same^*} = 0 \wedge b_\perp = 1$ and $W_3 := W|b_{same^*} = 0 \wedge b_\perp = 0$. By [Lemma 2.2a,[DORS08]], we get:

$$\Pr[\mathbf{H}_\infty(W_1) \geq \widetilde{\mathbf{H}}_\infty(W|b_{same^*}) - \lambda] \geq 1 - 2^{-\lambda}$$

which by [Lemma 2.2b, [DORS08]]further gives:

$$\Pr[\mathbf{H}_\infty(W_1) \geq n - 1 - \lambda] \geq 1 - 2^{-\lambda}$$

Similarly, we get

$$\Pr[\mathbf{H}_\infty(W_2) \geq n - 2 - \lambda] \geq 1 - 2^{-\lambda}$$
$$\Pr[\mathbf{H}_\infty(W_3) \geq n - 2 - \lambda] \geq 1 - 2^{-\lambda}$$

Now, we analyze the additional auxiliary information in each subcase:
$Case1 : b_{same^*} = 1$
In this case , the additional auxiliary information just includes a single bit, indicating whether w is modified or remains the same. So, we first define this indicator function:

$$eq(w) = \begin{cases} 0\ if\ f_L(w) \neq w \\ 1\ if\ f_L(w) = w \end{cases}$$

Let the auxiliary information be denoted by $E_1 \equiv eq(W)$. E_1 is independent of S because E_1 is determined given W and W is independent of S. Now, E_1 and W are correlated and E_1 can take at most two possible values.
Hence, $\widetilde{\mathbf{H}}_\infty(W_1|E_1) \geq \mathbf{H}_\infty(W_1) - 1 \geq n - 1 - \lambda - 1$ w.p. $\geq 1 - 2^{-\lambda}$. Let G_1 denote the event $\widetilde{\mathbf{H}}_\infty(W_1|E_1) \geq n - \lambda - 2$. As $n - \lambda - 2 > t$, by security of average case extractor, we get:

$$E_1, \mathsf{Ext}(W_1; S)|G_1 \approx_{\varepsilon_3} E_1, U_l|G_1$$

Now, clearly, in this case, the output of $\mathsf{Hybrid2}_{f,g}$ and $\mathsf{Hybrid3}_{f,g}$ are functions of above random variables. Hence, by Lemma 2, we get:

$$\mathsf{Hybrid2}_{f,g}|b_{same^*} = 1, G_1 \approx_{\varepsilon_3} \mathsf{Hybrid3}_{f,g}|b_{same^*} = 1, G_1$$

Hence, by further using Proposition 1, as $\Pr[G_1^c] \leq 2^{-\lambda}$, we get:

$$\mathsf{Hybrid2}_{f,g}|b_{same^*} = 1 \approx_{\varepsilon_3+2^{-\lambda}} \mathsf{Hybrid3}_{f,g}|b_{same^*} = 1 \tag{5}$$

$Case2 : b_{same^*} = 0$
This case is further divided into two mutually exclusive events of Case2.
$Case2a : b_\perp = 1$
Now, let G_2 denote the event $\mathbf{H}_\infty(W_2) \geq n - 2 - \lambda$. Then as $\Pr[G_2^c] \leq 2^{-\lambda}$ and using extractor security, we get:

$$\mathbf{SD}\left(\mathsf{Hybrid2}_{f,g}|b_{same^*} = 0 \wedge b_\perp = 1, \mathsf{Hybrid2}_{f,g}|b_{same^*} = 0 \wedge b_\perp = 1\right) \leq \varepsilon_3 + 2^{-\lambda} \tag{6}$$

$Case2b : b_\perp = 0$
In this case, the additional auxiliary information consists of an indicator of verification of $\tilde{w}$ and the extractor output on modified source and seed. We first define the indicator of verification bit:

$$Verify(w) = \mathsf{Vrfy}'_{\tilde{k_{a_1}}}(f_L(w), \tilde{t_1})$$

Now, let the auxiliary information be denoted by $E_2 \equiv (Verify(W), \mathsf{Ext}(\tilde{W}; \tilde{S}))$, where $\tilde{K_{a_1}}, \tilde{T_1}, \tilde{S}$ denote the distributions on the authentication key, tag spaces and the seed, when sampled from the simulator conditioned on the event Case2b. E_2 is clearly a deterministic function of $\tilde{K_{a_1}}, \tilde{W}, \tilde{T_1}, \tilde{S}$, all of which are independent of S (as we use the simulator). Hence, E_2 is independent of S. Now, E_2 and W are correlated. E_2 can take at most $2^{1+l+\tau}$ possible values.
Hence, $\widetilde{\mathbf{H}}_\infty(W_3|E_2) \geq \mathbf{H}_\infty(W_3) - (1 + l + \tau) \geq n - 2 - \lambda - (1 + l + \tau)$ w.p. $\geq 1 - 2^{-\lambda}$. Let G_3 denote the event $\widetilde{\mathbf{H}}_\infty(W_3|E_2) \geq n - (3 + \lambda + l + \tau)$. As $n - (3 + \lambda + l + \tau) > t$ (if we set parameters appropriately), by security of average case extractor and using Proposition 1, we get:

$$E_2, \mathsf{Ext}(W; S)|G_3 \approx_{\varepsilon_3} E_2, U_l|G_3$$

Now, clearly, in this case, the output of $\mathsf{Hybrid2}_{f,g}$ and $\mathsf{Hybrid3}_{f,g}$ are functions of above random variables. Hence, by Lemma 2, we get:

$$\mathsf{Hybrid2}_{f,g}|b_{same^*} = 0 \wedge b_\perp = 0, G_3 \approx_{\varepsilon_3} \mathsf{Hybrid3}_{f,g}|b_{same^*} = 0 \wedge b_\perp = 0, G_3$$

Further, since $\Pr[G_3^c] \leq 2^{-\lambda}$, using Proposition 1, we get

$$\mathsf{Hybrid2}_{f,g}|b_{same^*} = 0 \wedge b_\perp = 0 \approx_{\varepsilon_3+2^{-\lambda}} \mathsf{Hybrid3}_{f,g}|b_{same^*} = 0 \wedge b_\perp = 0 \tag{7}$$

Hence, by Proposition 1, Equations 4, 5, 6 and 7 give:

$$\mathsf{Hybrid2}_{f,g} \approx_{\varepsilon_3+2^{-\lambda}} \mathsf{Hybrid3}_{f,g}$$

B Appendix: From t-source strong non-malleable extractors to t-state 1-augmented NMC

We generalize the connection known between seedless non-malleable extractors for t independent sources and non-malleable codes for the t-split-state family ([CG14b]), to establish a connection between strong seedless non-malleable extractors for t independent sources and augmented non-malleable codes for t-split-state family. We first define strong seedless non-malleable t-source extractor.

Definition 6. *[Li17] A function* $nmExt : (\{0,1\}^n)^t \to \{0,1\}^m$ *is a* (k,ϵ)*-seedless strong non-malleable extractor for* t *independent sources w.r.t. family* $\mathcal{F} = \{(f_1, \cdots f_t) : f_i : \{0,1\}^n \to \{0,1\}^n\}$*, if it satisfies the following property: Let* $X_1, \cdots, X_t$ *be* t *independent* (n,k)*-sources and* $(f_1, \cdots, f_t) \in \mathcal{F}$ *be* t *arbitrary functions such that there exists an* f_j *with no fixed points, then for every* i*:*

$$(nmExt(X_1, \cdots, X_t), nmExt(f_1(X_1), \cdots, f_t(X_t)), X_i) \approx_\epsilon \\ (U_m, nmExt(f_1(X_1), \cdots, f_t(X_t)), X_i)$$

Now, we formulate an alternate definition of a t-source relaxed strong non-malleable extractor, generalizing the definition of seedless relaxed non-malleable extractors in [CG14b]. This definition captures the property that the output of non-malleable extractor on the modified sources along with one of the source, is simulatable independent of the output of non-malleable extractor on original sources.

Definition 7. *A function* $nmExt : (\{0,1\}^n)^t \to \{0,1\}^m$ *is a* (k,ϵ)*-seedless relaxed strong non-malleable extractor for* t *independent sources w.r.t. family* $\mathcal{F} = \{(f_1, \cdots f_t) : f_i : \{0,1\}^n \to \{0,1\}^n$ *and* $\exists$ *at least one* j *s.t.* f_j *has no fixed point}, if it satisfies the following property: Let* $X_1, \cdots, X_t$ *be* t *independent* (n,k)*-sources and* $(f_1, \cdots, f_t) \in \mathcal{F}$*, then the following hold:*

- $nmExt$ *is a* t*-source extractor for* $(X_1, \cdots, X_t)$*, i.e.,* $nmExt(X_1, \cdots, X_t) \approx_\epsilon U_m$*.*
- *There exists a distribution* $\mathcal{D}$ *over* $\{0,1\}^n \times (\{0,1\}^m \cup \{same^*\})$ *s.t. for an independent* $(X_1, Y) \sim \mathcal{D}$*,*

$$(nmExt(X_1, \cdots, X_t), X_1, nmExt(f_1(X_1), \cdots, f_t(X_t))) \approx_\epsilon \\ (nmExt(X_1, \cdots, X_t), copy((X_1, Y), (X_1, nmExt(X_1, \cdots, X_t))))$$

Remark 1. It is clear that the non-malleability condition in Def 6 (for $i = 1$) is sufficient for the conditions in Def 7 to be satisfied.

But then, this relaxed notion of strong non-malleable extractor is equivalent to the following general notion of strong non-malleable extractor (where, the tampering functions can have fixed points) upto a slight loss of parameters. (This proof follows from [Lemma 5.6, [CG14b]]).

Definition 8. *A function* $nmExt : (\{0,1\}^n)^t \to \{0,1\}^m$ *is a* (k,ϵ)*-seedless strong non-malleable extractor for* t *independent sources w.r.t. family* $\mathcal{F} = \{(f_1, \cdots f_t) : f_i : \{0,1\}^n \to \{0,1\}^n\}$*, if it satisfies the following property: Let* $X_1, \cdots, X_t$ *be* t *independent* (n,k)*-sources and* $(f_1, \cdots, f_t) \in \mathcal{F}$*, then the following hold:*

- $nmExt$ *is a* t*-source extractor for* $(X_1, \cdots, X_t)$*, i.e.,* $nmExt(X_1, \cdots, X_t) \approx_\epsilon U_m$*.*
- *There exists a distribution* $\mathcal{D}$ *over* $\{0,1\}^n \times (\{0,1\}^m \cup \{same^*\})$ *s.t. for an independent* $(X_1, Y) \sim \mathcal{D}$*,*

$$\begin{aligned}(nmExt(X_1, \cdots, X_t), X_1, nmExt(f_1(X_1), \cdots, f_t(X_t))) \approx_\epsilon \\ (nmExt(X_1, \cdots, X_t), copy((X_1, Y), (X_1, nmExt(X_1, \cdots, X_t))))\end{aligned}$$

Hence, we take the above Def 8 for strong non-malleable extractors and prove the following theorem.

Proposition 2. *Let* $nmExt : (\{0,1\}^n)^t \to \{0,1\}^k$ *be a* (n,ϵ)*-seedless strong non-malleable extractor for* t *independent sources (by Def 8). Define a coding scheme* $(\mathsf{Enc}, \mathsf{Dec})$ *with message length* k *and block length* tn *as follows. The decoder* Dec *is defined by*

$$\mathsf{Dec}(x_1, \cdots, x_t) = nmExt(x_1, \cdots, x_t)$$

The encoder Enc *is defined as:*

$$\mathsf{Enc}(m) := \begin{cases} x_1, \cdots, x_t \xleftarrow{\$} nmExt^{-1}(m) \\ o/p : (x_1, \cdots, x_t) \end{cases}$$

Then, $(\mathsf{Enc}, \mathsf{Dec})$ *is a* $[\epsilon', 1]$*-augmented non-malleable code with error* $\epsilon' = \epsilon(2^k + 1)$ *for the* t*-split state family and with rate* $= \frac{k}{tn}$*.*

Proof. Let $m \in \{0,1\}^k$ and $f = (f_1, \cdots, f_t) \in \mathcal{F}$, the t-split-state family be arbitrary. Since $\mathsf{Dec} = nmExt$ is a strong non-malleable extractor, by Def 8, $\exists$ a distribution $\mathcal{D}$ s.t. for $(X_1, Y) \sim \mathcal{D}_{f_1,\cdots,f_t}$, we have:

$$\begin{aligned}(nmExt(X_1, \cdots, X_t), X_1, nmExt(f_1(X_1), \cdots, f_t(X_t))) \approx_\epsilon \\ (nmExt(X_1, \cdots, X_t), copy((X_1, Y), (X_1, nmExt(X_1, \cdots, X_t)))) \quad (8)\end{aligned}$$

Claim. $\mathsf{Enc}(U_k)$ is ϵ-close to uniform.

Proof. By extractor security, we have:

$$\mathsf{Dec}(U_{tn}) \approx_\epsilon U_k$$

Further, as $\mathsf{Enc}(.)$ samples uniformly random element of $nmExt^{-1}(.)$, it follows that

$$\mathsf{Enc}(\mathsf{Dec}(U_{tn})) = U_{tn}$$

Hence, we get $\mathsf{Enc}(U_k) \approx_\epsilon \mathsf{Enc}(\mathsf{Dec}(U_{tn})) = U_{tn}$.

Thus, at cost of ϵ increase in error, we assume codeword is of uniform distribution.
Let $(X_1, Y) \sim \mathcal{D}_{f_1,\cdots,f_t}$. Now by Eq 8, just by substitution, we get:

$$(M, X_1, \mathsf{Dec}(f(\mathsf{Enc}(M))) \approx_\epsilon (M, copy((X_1, Y), (X_1, M)))$$

Now, for the arbitrary m that we chose, we get:

$$(m, X_1, \mathsf{Dec}(f(\mathsf{Enc}(m))) \approx_{\epsilon 2^k} (m, copy((X_1, Y), (X_1, m)))$$

which proves the theorem.

Augmented-non-malleability of 2-state construction in [Li17]:

Corollary 2. *For any $\beta \in \mathbb{N}$ there exists an explicit augmented-non-malleable code with efficient encoder/decoder in 2-split state model with block length 2β, rate $\Omega\left(\frac{1}{\log\ \beta}\right)$ and error $= 2^{-\Omega\left(\frac{\beta}{\log\ \beta}\right)}$*

Proof. As proved in [Theorem 7.9, [Li17]], the seedless 2 source non-malleable extractor constructed in [Li17] satisfies: For any (f, g) in 2-split-state family, such that atleast one of f or g has no fixed point, we have:

$$nmExt(X, Y), X, nmExt(f(X), g(Y)) \approx_\epsilon U_m, X, nmExt(f(X), g(Y))$$

which, by Remark 1, is sufficient to imply the conditions in Def 7. Hence, by Proposition 2, it is proved that the 2-split-state construction given in [Li17] is actually a 2-split-state augmented-non-malleable code.
Further, the specific non-malleable extractor of [Li17] gives error and rate parameters for the augmented-non-malleable code, exactly as obtained in Lemma 8.