Upper and Lower Bounds for Continuous Non-Malleable Codes

Dana Dachman-Soled * and Mukul Kulkarni

University of Maryland, College Park, USA
danadach@ece.umd.edu, mukul@terpmail.umd.edu

Abstract. Recently, Faust et al. (TCC'14) introduced the notion of *continuous* non-malleable codes (CNMC), which provides stronger security guarantees than standard non-malleable codes, by allowing an adversary to tamper with the codeword in a continuous way instead of one-time tampering. They also showed that CNMC with information theoretic security cannot be constructed in the 2-split-state tampering model, and presented a construction in the common reference string (CRS) model from collision-resistant hash functions and non-interactive zero-knowledge proofs.

In this work, we ask if it is possible to construct CNMC from weaker assumptions. We answer this question by presenting lower as well as upper bounds. We show that it is impossible to construct 2-split-state CNMC, with no CRS, for one-bit messages from any falsifiable assumption, thus establishing the lower bound. We additionally provide an upper bound by constructing 2-split-state CNMC for one-bit messages, assuming only the existence of a family of injective one way functions. We note that in a recent work, Ostrovsky et al. (CRYPTO'18) considered the construction of a relaxed notion of 2-split-state CNMC from minimal assumptions.

We also present a construction of 4-split-state CNMC for multi-bit messages in CRS model from the same assumptions. Additionally, we present definitions of the following new primitives: 1) *One-to-one commitments*, and 2) *Continuous Non-Malleable Randomness Encoders*, which may be of independent interest.

Keywords: Continuous non-malleable codes, black-box impossibility, split-state.

1 Introduction

Non-malleable codes (NMC). Non-malleable codes were introduced by Dziembowski, Pietrzak and Wichs [**?**] as a relaxation of error-correcting codes, and are useful in settings where privacy—but not necessarily correctness—is desired.

* This work is supported in part by NSF grants #CNS-1840893, #CNS-1453045 (CAREER), by a research partnership award from Cisco and by financial assistance award 70NANB15H328 from the U.S. Department of Commerce, National Institute of Standards and Technology.

The main application of non-malleable codes proposed in the literature is for protecting a secret key stored on a device against tampering attacks, although non-malleable codes have also found applications in other of areas of cryptography [**?**, **?**, **?**] and theoretical computer science [**?**].

Continuous Non-malleable codes (CNMC). Importantly, standard non-malleable codes achieve security only against one-time tampering. So in applications, the non-malleable encoding of a secret key must be continually decoded and re-encoded, incurring overhead in computation and in generation of randomness for re-encoding. This motivated a stronger notion of non-malleable codes, *continuous non-malleable codes (CNMC)*, introduced by Faust et al. [**?**]. This definition allows many-time tampering–i.e. the adversary can continuously tamper with the codeword and observe the effects of the tampering. Due to known impossibility results, there must also be a "self-destruct" mechanism: If, upon decode, the device detects an error, then a "self-destruct" mechanism, which erases the secret key, is triggered, rendering the device useless.

The notion of CNMC with respect to a tampering class $\mathcal{F}$ is as follows: Given a coding scheme $\Pi = (\mathsf{E}, \mathsf{D})$, where E is the encoding function and D is the decoding function, the adversary interacts with an oracle $\mathcal{O}_\Pi(C)$, parameterized by Π and an encoding of a message m, $C \leftarrow \mathsf{E}(m)$. We refer to the encoding C as the "challenge" encoding. In each round, the adversary submits a tampering function $f \in \mathcal{F}$. The oracle evaluates $C' = f(C)$. If $\mathsf{D}(C') = \perp$, the oracle outputs $\perp$ and a "self-destruct" occurs, aborting the experiment. If $C' = C$, the oracle outputs a special message "same." Otherwise, the oracle outputs C'. We emphasize that the entire tampered codeword is returned to the adversary in this case. A CNMC is secure if for every pair of messages m_0, m_1, the adversary's view in the above game is computationally indistinguishable when the message is m_0 or m_1.

Recently, Ostrovsky et al. [**?**] proposed a relaxed definition of CNMC (sufficient for many applications) along with a construction, in which the oracle $\mathcal{O}_\Pi(C)$ returns, if valid, the decoding of the tampered codeword $\mathsf{D}(C')$ (or "same") instead of the tampered codeword C' as in the standard (original) definition of [**?**]. In terms of applications, the difference between the original notion (which we consider in this paper) and the notion of [**?**], is that the notion we consider captures stronger types of side-channel attacks: Our notion provides security against an adversary who tampers and additionally learns information about the modified codeword C' through other side-channels. As a concrete example, an interesting research direction is to compose a split-state CNMC (under the original definition) with a leakage-resilient circuit compiler, such as the compiler of Ishai, Sahai and Wagner [**?**], in order to yield a compiler that simultaneously provides security against tampering with memory and leakage on computation. For more discussion and comparison of this paper with [**?**] see Section 1.2.

Split-state tampering. One of the most well-studied tampering classes for non-malleable codes is *split-state tampering*. Here, the codeword is split into sections

and the adversarial tampering function may tamper each section independently. The case of 2-split-state tampering, where the codeword is split into two sections, is of particular interest. See Section 1.4 for a discussion of prior work on NMC and CNMC against split-state tampering.

Information-theoretic impossibility. The original CNMC paper of [**?**] showed an information-theoretic impossibility result for 2-split-state CNMC. To aid the subsequent discussion, we present an outline of this result. The impossibility result considers a property of 2-split-state CNMC known as (perfect) "uniqueness." Informally, perfect uniqueness means that there do not exist triples (x, y, z) such that either (1) $y \neq z \wedge \mathsf{D}(x, y) \neq \perp \wedge \mathsf{D}(x, z) \neq \perp$ OR (2) $x \neq y \wedge \mathsf{D}(x, z) \neq \perp \wedge \mathsf{D}(y, z) \neq \perp$. First, a perfectly unique CNMC cannot be information-theoretically secure since, given L, the split-state tampering function can find the unique R such that $\mathsf{D}(L, R) \neq \perp$ and then tamper based on $m = \mathsf{D}(L, R)$. On the other hand, if the CNMC is not perfectly unique, then the following is an efficient attack (with non-uniform advice): Given a tuple L'_1, L'_2, R' such that $\mathsf{D}(L'_1, R') \neq \perp$ and $\mathsf{D}(L'_2, R') \neq \perp$, the adversary can learn L bit-by-bit by using the following tampering function in the i-th round: f_L does the following: If the i-th bit of L is equal to 0, replace L with L'_1. Otherwise, replace L' with L'_2. f_R always replaces R with R'. Now, in the i-th round, if the oracle returns (L'_1, R'), then the adversary learns that the i-th bit of L is equal to 0. If the oracle returns (L'_2, R'), then the adversary learns that the i-th bit of L is equal to 1. Once L is fully recovered, the adversary can tamper based on $m = \mathsf{D}(L, R)$.

The computational setting. The above shows that the CNMC setting is distinguished from other NMC settings, since information-theoretic (unconditional) security is impossible. Prior work has shown how to construct 2-split-state CNMC in the *CRS model* under the assumptions of collision-resistant hash functions and NIZK. On the other hand, CNMC's imply commitment schemes, which in turn imply OWF. It remains to determine where CNMC lies in terms of complexity assumptions and what are the minimal computational assumptions needed to achieve CNMC. As mentioned previously, a very recent work of Ostrovsky et al. [**?**] addressed minimizing computational assumptions under a relaxed definition of CNMC. See Section 1.2 for more details.

Black-box reductions. In general, it is not feasible to unconditionally rule out the construction of a primitive G from a cryptographic assumption H, since unconditionally ruling it out is as hard as proving $P \neq NP$. Despite this, we can still show that the proof techniques we have at hand cannot be used to construct G from assumption H. In the literature, this is typically done by showing that there is no black-box reduction from primitive G to assumption H. In this work, what we mean by a black-box reduction is a reduction that accesses the *adversary* in an input/output fashion only. However, we allow non-black-box usage of the assumption H in both the construction and the proof (see Definition 6 for a

formal definition tailored to CNMC). While there are some exceptions [**?**, **?**], the vast majority of cryptographic reductions are black-box in the adversary.

1.1 Our Results

We present upper and lower bounds for CNMC in the 2-split-state model. First, we show that with no CRS, single-bit CNMC in the 2-split-state model (with a black-box security proof) is impossible to construct from any falsifiable assumption.

Theorem 1 (Informal). *There is no black-box reduction from a single-bit, 2-split-state, CNMC scheme $\Pi = (\mathsf{E}, \mathsf{D})$ to any falsifiable assumption.*

On the other hand, in the CRS model, we show how to achieve single-bit CNMC in the 2-split-state model from injective one-way functions.

Theorem 2. *Assuming the existence of an injective one-way function family, there is a construction of a 2-split-state CNMC for encoding single bit, in the CRS model. Moreover, the corresponding reduction is black-box.*

Actually, we show a somewhat more general result: First, we define a (to the best of our knowledge) new type of commitment scheme called *one-to-one commitment schemes in the CRS model*. Informally, these commitment schemes have the additional property that with all but negligible probability over Σ produced by CRS generation, for every string *com*, there is at most a *single* string d that will be accepted as a valid decommitment for *com* (See Definition 9 for a formal definition). We also define the notion of a 2-split-state CNM Randomness Encoder, which is the continuous analogue of the non-malleable randomness encoder recently introduced by [**?**] (See Definition 5). We then show the following:

Theorem 3. *Assuming the existence of one-to-one commitment schemes in the CRS model, there is a construction of a 2-split-state CNM Randomness Encoder in the CRS model. Moreover, the corresponding reduction is black-box.*

One-to-one commitment schemes in the CRS model can be constructed from any injective one-way function family. Furthermore, we show (see the full version of this paper [**?**])that 2-split-state CNM Randomness Encoders in the CRS model imply 2-split-state CNMC for encoding single bit, in the CRS model. We therefore obtain Theorem 2 as a corollary. Moreover, CNMC with *perfect* uniqueness in the CRS model implies one-to-one commitment schemes in the CRS model in a straightforward way (refer to the full version of this paper [**?**])

We leave open the question of constructing CNMC in the CRS model from (non-injective) one-way functions and/or showing a black-box separation between the two primitives. Finally, we extend the techniques from our single-bit construction above to achieve the following:

Theorem 4. *Assuming the existence of one-to-one commitment schemes in the CRS model, there is a construction of a multi-bit, 4-split-state CNMC in the CRS model. Moreover, the corresponding reduction is black-box.*

Are prior CNMC reductions "black-box"? Prior CNMC reductions often proceed in a sequence of hybrids, where in the final hybrid, the description of the adversary is incorporated in the definition of a leakage function. It is then shown that the leakage-resilience properties of an underlying encoding imply that the view of the adversary is statistically close when the encoded message is set to m_0 or m_1. While this may seem like non-black-box usage of the adversary, we note that typically the leakage-resilience of the underlying encoding is information-theoretic. When converting a hybrid-style proof to a reduction, the reduction will choose one of the hybrid steps at random and use the fact that a distinguisher between some pair of consecutive hybrids implies an adversary breaking an underlying assumption. Therefore, reductions of the type discussed above are still black-box in the adversary, pairs of consecutive hybrids whose indistinguishability is implied by a *computational* assumption yield a reduction in which the adversary is used in a black-box manner.

1.2 Comparison with Ostrovsky et al. [?]

The CNMC notion considered in this work is the original continuous non-malleable codes notion, first introduced in [?] and then further studied in several follow-up works (including [?, ?, ?]). Recently, Ostrovsky et. al. [?] introduced a relaxed notion of CNMC,[1] which is sufficient for many applications. In the work of Ostrovsky et. al. [?], they refer to the original notion as "continuous super-non-malleability" (since it is analogous to "super-non-malleability", a notion that was introduced in the non-continuous setting [?]). They then presented a construction achieving the relaxed definition (which they simply call "continuous non-malleability"), against 2-split-state tampering functions, assuming the existence of injective one-way functions in the plain model (without CRS).

The difference between the two CNMC notions is that in the original CNMC notion, the tampering oracle returns the entire modified codeword C' if $C' = f(C) \neq C$ and $\mathsf{D}(C) \neq \perp$, whereas the relaxation only requires the oracle to return $\mathsf{D}(C')$ but not C' itself. The original notion captures stronger types of tampering attacks; specifically, it provides security against an adversary who learns arbitrary additional information about the modified codeword C' through other side-channels.

Our result and the result of [?] are complementary and together give a full picture of the landscape of assumptions required for CNMC. Our work shows that it is *necessary* to rely on setup assumptions (CRS) in order to achieve the original, stronger security definition of CNMC. Moreover, if one is willing to assume the existence of a CRS, we show that this type of CNMC can be achieved from nearly minimal computational assumptions. In contrast, if one is not willing to assume the existence of a CRS, the work of [?] achieves weaker security guarantees in the plain model (with no setup assumptions) from the

[1] A similar relaxed definition was previously given for a variant of CNMC, known as R- CNMC [?], but in this setting it was shown that it is actually impossible to achieve the stronger notion.

same computational assumptions. We also note that the work of Ostrovsky et. al. [**?**] explicitly lists the question we address in this work as an interesting open problem. They state:

> Interesting open questions related to our work are, for instance, whether continuous non-malleability can be achieved, under minimal assumptions, together with additional properties, such as strong non-malleability, **super-non-malleability**, augmented non-malleability, and locality ...

1.3 Technical Overview

Lower bound. Recall that prior work has shown that if a CNMC is not perfectly unique, then there is an efficient attack (with non-uniform advice). Thus, it remains to show that there is no black-box reduction from a single-bit, *perfectly unique* CNMC scheme to any falsifiable assumption. We use the meta-reduction approach, which is to prove impossibility by showing that given only black-box access to the split-state adversary, $A = (A_L, A_R)$, the reduction cannot distinguish between the actual adversary and a *simulated* (efficient) adversary (which is possibly stateful). Since the view of the reduction is indistinguishable in the two cases, the reduction must also break the falsifiable assumption when interacting with the simulated adversary. But this in turn means that there is an efficient adversary (obtained by composing the reduction and the simulated adversary), which contradicts the underlying falsifiable assumption. Consider the following stateless, inefficient, split-state adversary $A = (A_L, A_R)$, which leverages the uniqueness property of the CNMC scheme: The real adversary, given L (resp. R), recovers the corresponding unique valid codeword (L, R) (if it exists) and decodes to get the bit b. If $b = 0$, the real adversary encodes a random bit b' using internal randomness that is tied to (L, R), and outputs the left/right side as appropriate. If $b = 1$ or there is no corresponding valid codeword, the real adversary outputs the left/right side of a random encoding of a random bit, b'' (generated using internal randomness that is tied to L or R respectively). The simulated adversary is stateful and keeps a table containing all the L and R values that it has seen. Whenever a L (resp. R) query is made, the simulated adversary first checks the table to see if a matching query to R (resp. L) such that $\mathsf{D}(L, R) \neq \perp$ was previously made. If not, the simulated adversary chooses a random encoding, (L', R'), of a random bit b', stores it in the table along with the L/R query that was made and returns either L' or R' as appropriate. If yes, the simulated adversary finds the corresponding R (resp. L) along with the pair (L', R') stored in the table. The simulated adversary then decodes (L, R) to find out b. If $b = 0$, the simulated adversary returns either L' or R' as appropriate. Otherwise, the simulated adversary returns the left/right side of an encoding of a random bit b''. The uniqueness property allows us to prove that the input/output behavior of the real adversary is identical to that of the simulated adversary. See Section 3 for additional details. For a discussion on why our impossibility result does not hold for the relaxed CNMC notion considered by [**?**], see the full version of this paper [**?**].

Upper bound. For the upper bound, we construct a new object called a continuous non-malleable randomness encoder (see Definition 5), which is the continuous analogue of the non-malleable randomness encoder recently introduced by [**?**]. Informally, a continuous non-malleable randomness encoder is just a non-malleable code for randomly chosen messages. It is then straightforward to show that a continuous non-malleable randomness encoder implies a single-bit continuous non-malleable code (see the full version of this paper [**?**] for details).

At a high level, the difficulty in proving continuous non-malleability arises from the need of the security reduction to simulate the interactive tampering oracle, without knowing the message underlying the "challenge" encoding. The approach of prior work such as [**?**] was to include a NIZK Proof of Knowledge in each part of the codeword to allow the simulator to extract the second part of the encoding, given the first. This then allowed the simulator (with some additional leakage) to respond correctly to a tampering query, while knowing only one of the two split-states of the original encoding. In our setting, we cannot use NIZK, since our goal is to reduce the necessary complexity assumptions; therefore, we need a different extraction technique.[2] Our main idea is as follows: To respond to the i-th tampering query, we run the adversarial tampering function on random (simulated) codewords (L', R') that are consistent with the output seen thus far (denoted Out_A^{i-1}) and keep track of frequent outcomes (occurring with non-negligible probability) of the tampering function, $\widehat{L}, \widehat{R}$. I.e. S_L (resp. S_R) is the set of values of $\widehat{L}$ (resp. $\widehat{R}$) such that with non-negligible probability over choice of L' (resp. R'), it is the case that $\widehat{L} = f_L(L')$ (resp. $\widehat{R} = f_R(R')$). We then show that if the outcome of the tampering function applied to the actual "challenge" split-state L or R is not equal to one of these frequent outcomes (i.e. $f_L(L) \notin S_L$ or $f_R(R) \notin S_R$), then w.h.p. the decode function D outputs $\perp$. This will allow us to simulate the experiment with only a small amount of leakage (to determine which of the values in S_L/S_R should be outputted). Note that, while the sets S_L/S_R are small, and so only a few bits are needed to specify the outcome, conditioned on the outcome being in S_L/S_R, the CNMC experiment runs for an *unbounded* number of times, and so even outputting a small amount of information in each round can ultimately lead to unbounded leakage. To solve this problem, we also consider the *most frequent* outcome in the sets S_L/S_R. This is the value of $\widehat{L}$ (resp. $\widehat{R}$) that occurs with the highest probability when $f_L(L')$ (resp. $f_R(R')$) is applied to consistent L' (resp. R'). Note that if a value $\widehat{L}'$ (resp. $\widehat{R}'$) is *not* the most frequent value, then it occurs with probability at most $1/2$. We argue that, for each round i of the CNMC experiment, the probability that a value $\widehat{L}'$ (resp. $\widehat{R}'$) that is not the most frequent value is outputted by f_L (resp. f_R) *and* self-destruct does not occur is at most $1/2$. This allows us to bound, w.h.p., the number of times in the entire tampering experiment that

[2] Note that our extraction technique is inefficient. This is ok, since the goal of the extraction technique is simply to show that the view of the adversary can be simulated given a small amount of leakage on each of the two split-states. Then, information-theoretic properties of the encoding are used to show that the view of the adversary must be independent of the random encoded value.

the value outputted by f_L (resp. f_R) is not the most frequent value. Thus, when the value outputted by f_L (resp. f_R) *is* the most frequent value, the leakage function outputs *nothing*, since the most frequent value can be reconstructed from the given information. In contrast, if the value outputted by f_L (resp. f_R) is *not* the most frequent value, but is in the sets S_L/S_R, then it has a small description and, moreover, this event occurs a bounded number of times. Therefore, we can afford to leak this information up to some upperbounded number of rounds, while the total amount of leakage remains small relative to the length of the encoding. Looking ahead, our construction will use a two-source extractor, whose properties will guarantee that even given the leakage (which contains all the information needed to simulate the CNMC experiment), the decoded value remains uniform random.

To show that if the outcome of the tampering function is not in S_L or S_R, then decode outputs $\perp$ w.h.p., we first use the "uniqueness" property, which says that for every $\widehat{L} = f_L(L)$ (resp. $\widehat{R} = f_R(R)$), there is at most a single "match", $\widehat{R}'$ (resp. $\widehat{L}'$), such that $\mathsf{D}_\Sigma(\widehat{L}, \widehat{R}') \neq \perp$ (resp. $\mathsf{D}_\Sigma(\widehat{L}', \widehat{R}) \neq \perp$). Given the "uniqueness" property, it is sufficient to show that for every setting of L, Out_A^{i-1}

$$\Pr[f_R(R) = \widehat{R}' \wedge \widehat{R}' \notin S_R \mid L \wedge \mathsf{Out}_A^{i-1}] \leq \mathsf{negl}(n) \tag{1}$$

and that for every setting of $R \wedge \mathsf{Out}_A^{i-1}$

$$\Pr[f_L(L) = \widehat{L}' \wedge \widehat{L}' \notin S_L \mid R \wedge \mathsf{Out}_A^{i-1}] \leq \mathsf{negl}(n). \tag{2}$$

To prove the above, we first argue that for the "challenge" codeword, (L, R), the split-states L and R are conditionally independent, given Out_A^{i-1} (assuming no $\perp$ has been outputted thus far) and an additional simulated part of the codeword. This means that the set of frequent outcomes S_L (resp. S_R) conditioned on Out_A^{i-1} is the same as the set of frequent outcomes S_L (resp. S_R) conditioned on *both* Out_A^{i-1} and R (resp. L). So for any $\widehat{R} \notin S_R$,

$$\Pr[f_R(R) = \widehat{R} \mid L \wedge \mathsf{Out}_A^{i-1}] \leq \mathsf{negl}(n)$$

and for any $\widehat{L} \notin S_L$,

$$\Pr[f_L(L) = \widehat{L} \mid R \wedge \mathsf{Out}_A^{i-1}] \leq \mathsf{negl}(n).$$

Since $\widehat{R}'$ (resp. $\widehat{L}'$) is simply a particular setting of $\widehat{R} \notin S_R$ (resp. $\widehat{L} \notin S_L$), we have that (1) and (2) follow.

For the above analysis, we need the encoding scheme to possess the following property: The L, R sides of the "challenge" codeword are conditionally independent given Out_A^{i-1} (and an additional simulated part of the codeword), but any tampered split-state $f_L(L)$ or $f_R(R)$ created by the adversary has at most a single "match," $\widehat{R}'$ or $\widehat{L}'$.

To explain how we achieve this property, we briefly describe our construction. Our construction is based on a non-interactive, equivocal commitment scheme in the CRS model and a two-source (inner product) extractor. Informally, an

equivocal commitment scheme is a commitment scheme with the normal binding and hiding properties, but for which there exists a simulator that can output simulated commitments which can be opened to both 0 and 1. In the CRS model, the simulator also gets to sample a simulated CRS. Moreover, the CRS and commitments produced by the simulator are indistinguishable from real ones.

To encode a random value m, random vectors c_L, c_R such that $\langle c_L, c_R \rangle = m$ are chosen. We generate a commitment *com* to $c_L || c_R$. The commitment scheme has the additional property that adversarially produced commitments are statistically binding (even if an equivocal commitment has been released) and have at most a *single* valid decommitment string. The left (resp. right) split-state L (resp. R) consists of *com* and an opening of *com* to the bits of c_L (resp. c_R). The special properties of the commitment scheme guarantee the "perfect uniqueness" property of the code. In the security proof, we replace the statistically binding commitment *com* in the "challenge" codeword with an equivocal commitment. Thus, each split-state of the challenge encoding, L (resp. R), contains no information about c_R (resp. c_L). Moreover, assuming "$\perp$" is not yet outputted, the output received by the adversary in the experiment at the point that the i-th tampering function is submitted, denoted Out_A^{i-1} is of the form $(f_L^1(L) = v_1, f_R^1(R) = w_1), \ldots, (f_L^{i-1}(L)) = v_{i-1}, f_R^{i-1}(R) = w_{i-1})$, where for $j \in [i-1]$, v_j is equal to the left value outputted in response to the j-th query and w_j is equal to the right value outputted in response to the j-th query. (note that v_j/w_j can be set to "same" if the tampering function leaves L/R unchanged). This allows us to argue that the distribution of $L \mid \mathsf{Out}_A^{i-1}, R$ (resp. $R \mid \mathsf{Out}_A^{i-1}, L$) is identical to the distribution of $L \mid \mathsf{Out}_A^{i-1}$ (resp. $R \mid \mathsf{Out}_A^{i-1}$) which implies that the left and right hand sides are conditionally independent given Out_A^{i-1} and the equivocal commitment, as desired. See Section 4 for additional details.

Extension to 4-state CNMC in CRS model from OWF. To encode a message m we now generate random $(c_{L,1}, c_{R,1}, c_{L,2}, c_{R,2})$ conditioned on $\langle c_{L,1}, c_{R,1} \rangle + \langle c_{L,2}, c_{R,2} \rangle = m$ (where addition is over a finite field). Now, we generate a commitment *com* to $c_{L,1}||c_{R,1}||c_{L,2}||c_{R,2}$. Each of the four split states now consists of *com* and an opening of *com* to the bits of $c_{L,b}$ (resp. $c_{R,b}$). The analysis is similar to the previous case and requires the property that at each point in the experiment the distribution of $\langle c_{L,1}, c_{R,1} \rangle$ (resp. $\langle c_{L,2}, c_{R,2} \rangle$) is uniform random, conditioned on the output thus far. Our techniques are somewhat similar to those used in [**?**] in their construction of $2t$-split-state continuously non-malleable codes from t-split-state one-way continuously non-malleable codes. See the full version of this paper [**?**] for additional details.

1.4 Additional Related Work

Non-Malleable Codes. The notion of non-malleable codes (NMC) was formalized in the seminal work of Dziembowski, Pietrzak and Wichs [**?**]. Split-state classes of tampering functions subsequently received a lot of attention with a long line

of works, including [?, ?, ?, ?, ?, ?, ?, ?, ?, ?]. Other works focused on various other classes of tampering functions, including [?, ?, ?, ?, ?, ?]. NMC have also been considered in several other models for various applications such as in [?, ?, ?]. Other works on non-malleable codes include [?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?].

Continuous Non-Malleable Codes *Continuous Non-Malleable codes (CNMC)* were introduced by Faust etal. [?]. They gave a construction based on collision resistant hash functions and non-interactive zero knowledge proof systems in the CRS model. They also showed the impossibility of constructing 2-split state CNMC information theoretically. Subsequently, Jafargholi and Wichs [?] presented a general study of CNMCs and its variants with some existential results. Aggarwal et al. [?] gave the first information theoretic construction in the 8-split-state model. Recently, Damgård et al. [?] gave the first construction of information theoretic CNMC against permutations. Faonio et al. [?] considered a variant of CNMC against split-state tampering where the codeword is refreshed (to avoid self-destruct) in the CRS model. For a discussion related to the recent work of Ostrovsky et al. [?], see Section 1.2 and refer to the full version of this paper [?] for further details.

Non-Malleable Randomness Encoders (NMRE) NMRE were introduced recently by Kanukurthi et al. [?] as a building block for constructing efficient (constant-rate) split-state NMC. In this work, we present the stronger variant *Continuous NMRE* which allows continual tampering in split-state model.

Bounds on Non-Malleable Codes. Cheragachi and Guruswami [?] studied the "capacity" of non-malleable codes and their work has been instrumental in asserting the claims of efficient constructions for non-malleable codes since then (cf. [?, ?, ?]). A similar study was presented in [?] for locally decodable and updatable NMC. This work studies bounds for *continuous non-malleable codes* in terms of *complexity assumptions*.

Black-Box Separations. Impagliazzo and Rudich ruled out black-box reductions from key agreement to one-way function in their seminal work [?]. Their oracle separation technique was subsequently used to rule out black-box reductions between various other primitives (cf. [?, ?] and many more). The meta-reduction technique (cf. [?, ?, ?, ?, ?, ?, ?, ?, ?, ?]) has been useful for ruling out larger classes of reductions—where the construction is arbitrary (non-black-box), but the reduction uses the adversary in a black-box manner. The meta-reduction technique is often used to provide evidence that construction of some cryptographic primitive is impossible under "standard assumptions" (e.g. falsifiable assumptions or non-interactive assumptions).

2 Definitions and Preliminaries

We present some standard notations and definitions, along with important lemmas related to randomness extractors, and the definition of strong one-time signature schemes in the full version of this paper [?] due to lack of space.

We present some more definitions in the following secrions.

2.1 CNMC

Definition 1 (Coding Scheme [?]). *A coding scheme,* $\mathsf{Code} = (\mathsf{E}, \mathsf{D})$*, consists of two functions: a randomized encoding function* $\mathsf{E} : \{0,1\}^\lambda \to \{0,1\}^n$*, and a deterministic decoding function* $\mathsf{D} : \{0,1\}^n \to \{0,1\}^\lambda \cup \{\perp\}$ *such that, for each* $m \in \{0,1\}^\lambda$*,* $\Pr[\mathsf{D}(\mathsf{E}(m)) = m] = 1$ *(over the randomness of encoding function).*

Definition 2 (Split-State Encoding Scheme in the CRS model [?]). *A split-state encoding scheme in* common reference string *(CRS) model is a tuple of algorithms,* $\mathsf{Code} = (\mathsf{CRSGen}, \mathsf{E}, \mathsf{D})$ *specified as follows:*

- CRSGen *takes the security parameter as input and outputs the CRS,* $\Sigma \leftarrow \mathsf{CRSGen}(1^\lambda)$*.*
- E *takes a message* $x \in \{0,1\}^\lambda$ *as input along with the CRS* Σ*, and outputs a codeword consisting of two parts* (X_0, X_1) *such that* $X_0, X_1 \in \{0,1\}^n$*.*
- D *takes a codeword* $(X_0, X_1) \in \{0,1\}^{2n}$ *as input along with the CRS* Σ *and outputs either a message* $x' \in \{0,1\}^\lambda$ *or a special symbol* $\perp$*.*

Consider the following oracle, $\mathcal{O}_{\mathsf{CNM}}((X_0, X_1), (\mathsf{T}_0, \mathsf{T}_1))$ which is parametrized by the CRS Σ and "challenge" codeword (X_0, X_1) and takes functions $\mathsf{T}_0, \mathsf{T}_1 : \{0,1\}^n \to \{0,1\}^n$ as inputs.

$\underline{\mathcal{O}_{\mathsf{CNM}}(\Sigma, (X_0, X_1), (\mathsf{T}_0, \mathsf{T}_1))}$:

$(X'_0, X'_1) = (\mathsf{T}_0(X_0), \mathsf{T}_1(X_1))$
If $(X'_0, X'_1) = (X_0, X_1)$ return same^*
If $\mathsf{D}_\Sigma(X'_0, X'_1) = \perp$, return $\perp$ and "self destruct"
Else return (X'_0, X'_1).

"Self destruct" here means that once $\mathsf{D}_\Sigma(X'_0, X'_1)$ outputs $\perp$, the oracle answers all the future queries with $\perp$.

Definition 3 (Continuous Non Malleability [?]). *Let* $\mathsf{Code} = (\mathsf{CRSGen}, \mathsf{E}, \mathsf{D})$ *be a split-state encoding scheme in the CRS model. We say that* Code *is q-continuously non-malleable code, if for all messages* x*,* $y \in \{0,1\}^\lambda$ *and all PPT adversary* $\mathcal{A}$ *it holds that*

$$\left\{\mathsf{CTamper}_{\mathcal{A},x}(\lambda)\right\}_{\lambda\in\mathbb{N}} \approx_c \left\{\mathsf{CTamper}_{\mathcal{A},y}(\lambda)\right\}_{\lambda\in\mathbb{N}} \qquad \text{where,}$$

$$\mathsf{CTamper}_{\mathcal{A},x}(\lambda) \stackrel{def}{=} \left\{ \begin{array}{c} \Sigma \leftarrow \mathsf{CRSGen}(1^\lambda);\ (X_0, X_1) \leftarrow \mathsf{E}_\Sigma(x); \\ \mathsf{out}_{\mathcal{A}} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{CNM}}(\Sigma,(X_0,X_1),(\cdot,\cdot))};\ \text{OUTPUT} : \mathsf{out}_{\mathcal{A}} \end{array} \right\}$$

and $\mathcal{A}$ *asks total of* q *queries to* $\mathcal{O}_{\mathsf{CNM}}$*.*

The following is an equivalent formulation

Definition 4 (Continuous Non Malleability [?], equivalent formulation).

Let $\mathsf{Code} = (\mathsf{CRSGen}, \mathsf{E}, \mathsf{D})$ *be a split-state encoding scheme in the CRS model. We say that* Code *is* q*-continuously non-malleable code, if for all messages* m_0*,* $m_1 \in \{0,1\}^\lambda$*, all PPT adversary* $\mathcal{A}$ *and all PPT distinguishers* D *it holds that*

$$\Pr[D(\mathsf{out}^b_{\mathcal{A}}) = b] \leq 1/2 + \mathsf{negl}(\lambda)$$

where $b \leftarrow \{0,1\}$ *and*

$$\mathsf{out}^b_{\mathcal{A}} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{CNM}}(\Sigma,(X^b_0,X^b_1),(\cdot,\cdot))} : \Sigma \leftarrow \mathsf{CRSGen}(1^\lambda);\ (X^b_0, X^b_1) \leftarrow \mathsf{E}_\Sigma(m_b)$$

and $\mathcal{A}$ *asks total of* q *queries to* $\mathcal{O}_{\mathsf{CNM}}$*.*

2.2 Continuous Non-Malleable Randomness Encoder

The following definition is an adaptation of the notion of Non-Malleable Randomness Encoders [**?**] to the continuous setting.

Definition 5. *Let* $\mathsf{Code} = (\mathsf{CRSGen}, \mathsf{CNMREnc}, \mathsf{CNMRDec})$ *be such that* CRSGen *takes security parameter* λ *as input and outputs a string of length* $\Sigma_1 = \mathrm{poly}(\lambda)$ *as CRS.* $\mathsf{CNMREnc} : \{0,1\}^{\Sigma_1} \times \{0,1\}^r \to \{0,1\}^\lambda \times (\{0,1\}^{n_1}, \{0,1\}^{n_2})$ *is defined as* $\mathsf{CNMREnc}(r) = (\mathsf{CNMREnc}_{1,\Sigma}(r), \mathsf{CNMREnc}_{2,\Sigma}(r)) = (m, (x_0, x_1))$ *and* $\mathsf{CNMRDec} : \{0,1\}^{\Sigma_1} \times \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^\lambda$*.*

We say that $(\mathsf{CRSGen}, \mathsf{CNMREnc}, \mathsf{CNMRDec})$ *is a* continuous non-malleable randomness encoder *with message space* $\{0,1\}^\lambda$ *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 holds true:*

- *Correctness:*

$$\Pr_{r \leftarrow \mathcal{R}}[\mathsf{CNMRDec}_\Sigma(\mathsf{CNMREnc}_{2,\Sigma}(r)) = \mathsf{CNMREnc}_{1,\Sigma}(r)] = 1$$

- *Continuous Non-Malleability:*

$$(\Sigma, \mathsf{CNMREnc}_{1,\Sigma}(R), \mathsf{out}_{\Sigma,\mathcal{A}}(R)) \approx_c (\Sigma, U_\lambda, \mathsf{out}_{\Sigma,\mathcal{A}}(R))$$

  where $\Sigma \leftarrow \mathsf{CRSGen}(1^\lambda)$*,* R *is a uniform random variable over* $\{0,1\}^r$*,* U_λ *is a uniform random variable over* $\{0,1\}^\lambda$ *and* $\mathsf{out}_{\Sigma,\mathcal{A}}(R)$ *is defined as follows:*

$$\mathsf{out}_{\Sigma,\mathcal{A}}(R) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{CNM}}(\Sigma,(X_0,X_1),(\cdot,\cdot))} : (X_0, X_1) \leftarrow \mathsf{CNMREnc}_{2,\Sigma}(R)$$

  where $\mathcal{O}_{\mathsf{CNM}}$ *runs with* $\mathsf{CNMRDec}$ *as decoding algorithm.*

2.3 Falsifiable Assumptions and Black-Box Reductions

Definition 6. *A falsifiable assumption consists of* PPT *interactive challenger* $\mathcal{C}(1^\lambda)$ *that runs in time* $\mathrm{poly}(\lambda)$ *and a constant* $0 \leq \delta < 1$*. The challenger* $\mathcal{C}$

interacts with a machine $\mathcal{A}$ and may output special symbol win*. If this occurs, $\mathcal{A}$ is said to win $\mathcal{C}$. For any adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ over $\mathcal{C}$ is defined as:*

$$\mathrm{Adv}_{\mathcal{A}}^{(\mathcal{C},\delta)} = |\Pr\left[\mathcal{A}(1^\lambda)\ wins\ \mathcal{C}(1^\lambda)\right] - \delta|,$$

where the probability is taken over the random coins of $\mathcal{A}$ and $\mathcal{C}$. The assumption associated with the tuple $(\mathcal{C}, \delta)$ states that for every (non-uniform) adversary $\mathcal{A}(1^\lambda)$ running in time $\mathrm{poly}(\lambda)$*,*

$$\mathrm{Adv}_{\mathcal{A}}^{(\mathcal{C},\delta)} = \mathsf{negl}(\lambda).$$

If the advantage of $\mathcal{A}$ is non-negligible in λ then $\mathcal{A}$ is said to break *the assumption.*

Definition 7. *Let $\Pi = (\mathsf{E}, \mathsf{D})$ be a split-state CNMC. We say that the non-malleability of Π can be proven via a* black-box reduction *to a falsifiable assumption, if there is an oracle access machine $\mathcal{M}^{(\cdot)}$ such that for every (possibly inefficient) Π-adversary $\mathcal{P}^*$, the machine $\mathcal{M}^{\mathcal{P}^*}$ runs in time* $\mathrm{poly}(\lambda)$ *and breaks the assumption.*

2.4 Equivocal Commitment Scheme

Definition 8 (Commitment Scheme). *A (non-interactive) commitment scheme in the CRS model for the message space $\mathcal{M}$, is a triple* $(\mathsf{CRSGen}, \mathsf{Commit}, \mathsf{Open})$ *such that:*

- $\Sigma \leftarrow \mathsf{CRSGen}(1^k)$ *generates the CRS.*
- *For all $m \in \mathcal{M}$, $(c, d) \leftarrow \mathsf{Commit}_\Sigma(m)$ is the* commitment/opening *pair for the message m. Specifically; c is the* commitment value *for m, and d is the* opening*.*
- $\mathsf{Open}_\Sigma(c, d) \to \tilde{m} \in \mathcal{M} \cup \{\perp\}$*, where $\perp$ is returned when c is not a valid commitment to any message.*

The commitment scheme must satisfy the standard correctness requirement,

$$\forall k \in \mathbb{N}, \forall m \in \mathcal{M} \text{ and } \Sigma \in \mathcal{CRS},\ \Pr\left[\mathsf{Open}_\Sigma(\mathsf{Commit}_\Sigma(m)) = m\right] = 1$$

where, $\mathcal{CRS}$ is the set of all possible valid CRS's generated by $\mathsf{CRSGen}(1^k)$ *and where the probability is taken over the randomness of* Commit*.*

The commitment scheme provides the following 2 security properties:

Hiding: It is computationally hard for any adversary $\mathcal{A}$ to generate two messages $m_0, m_1 \in \mathcal{M}$ such that $\mathcal{A}$ can distinguish between their corresponding commitments. Formally, for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ it should hold that:

$$\Pr\left[b = b' \;\middle|\; \begin{array}{c} \Sigma \leftarrow \mathsf{CRSGen}(1^k), (m_0, m_1, \alpha) \leftarrow \mathcal{A}_1(\Sigma), \\ b \leftarrow_r \{0,1\}, (c, d) \leftarrow \mathsf{Commit}_\Sigma(m_b), b' \leftarrow \mathcal{A}_2(c, \alpha) \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(k)$$

Binding: It is computationally hard for any adversary $\mathcal{A}$ to find a triple (c, d, d') such that *both* (c, d) and (c, d') are valid commitment/opening pairs for some $m, m' \in \mathcal{M}$ respectively, and $m \neq m'$. Formally, for any PPT adversary $\mathcal{A}$ it should hold that:

$$\Pr\left[\begin{array}{l|l} m \neq m' \wedge & \Sigma \leftarrow \mathsf{CRSGen}(1^k), (c, d, d') \leftarrow \mathcal{A}(\Sigma), \\ m, m' \neq \perp & m \leftarrow \mathsf{Open}_{\Sigma}(c, d), m' \leftarrow \mathsf{Open}_{\Sigma}(c, d') \end{array}\right] \leq \mathsf{negl}(k)$$

Definition 9 (One-to-One Commitment Scheme in the CRS Model). *Let* $(\mathsf{CRSGen}, \mathsf{Commit}, \mathsf{Open})$ *be a bit-commitment scheme in CRS model. We say that* $(\mathsf{CRSGen}, \mathsf{Commit}, \mathsf{Open})$ *is a* one-to-one commitment scheme *if with all but negligible probability over* $b \leftarrow \{0,1\}$, $\Sigma \leftarrow \mathsf{CRSGen}(1^\lambda)$, $(com, d) \leftarrow \mathsf{Commit}_{\Sigma}(b)$, $d' = d$ *is the unique string such that* $\mathsf{Open}(com, d') \neq \perp$.

Definition 10. *Let* $(\mathsf{CRSGen}, \mathsf{Commit}, \mathsf{Open})$ *be a bit-commitment scheme in CRS model. We say that* $(\mathsf{CRSGen}, \mathsf{Commit}, \mathsf{Open})$ *is a* non-interactive equivocable bit-commitment scheme *in the CRS model if there exists an efficient probabilistic algorithm* S_{Eq} *which on input* 1^λ *outputs a 4-tuple* $(\Sigma', c', d'_0, d'_1)$ *satisfying the following:*

- $\Pr[\mathsf{Open}_{\Sigma'}(c', d'_b) = b] = 1$ *for* $b \in \{0,1\}$.
- *For* $b \in \{0,1\}$, *it holds that* $\mathsf{out}_{\mathsf{Commit}}(b) \approx_\varepsilon \mathsf{out}_{\mathsf{S}_{Eq}}(b)$ *where the random variables* $\mathsf{out}_{\mathsf{Commit}}(b)$ *and* $\mathsf{out}_{\mathsf{S}_{Eq}}(b)$ *are defined as follows:*

$$\left\{\begin{array}{c} \Sigma \leftarrow \mathsf{CRSGen}(1^\lambda);\ (c, d) \leftarrow \mathsf{Commit}_{\Sigma}(b); \\ \mathsf{out}_{\mathsf{Commit}}(b) : (\Sigma, c, d) \end{array}\right\} \approx \left\{\begin{array}{c} (\Sigma', c', d'_0, d'_1) \leftarrow \mathsf{S}_{Eq}(1^\lambda); \\ \mathsf{out}_{\mathsf{S}_{Eq}}(b) : (\Sigma', c', d'_b) \end{array}\right\}$$

We now present variant of the commitment scheme presented by Naor in [**?**], specifically we present the same construction in CRS model. This is also presented in [**?**].

Let $n > 0$ be an integer, let $G : \{0,1\}^n \to \{0,1\}^{3n}$ be a PRG.

- $\mathsf{CRSGen}(1^n)$: Output a uniform random string Σ of length $3n$.
- $\mathsf{Commit}_{\Sigma}(b)$: Choose uniform random seed $s \in \{0,1\}^n$ and compute $t = G(s)$. If $b = 0$, set $c := t$. If $b = 1$, set $c := t \oplus \Sigma$. Output c. Output decommitment $d = s$.
- $\mathsf{Open}_{\Sigma}(c, d)$: If $c = G(d)$, then output 0. Else if, $c = G(d) \oplus \Sigma$, then output 1. Output $\perp$ otherwise.

Claim 2.1. The scheme presented above is an *equivocal* commitment scheme.

The proof of claim 2.1 can be found in the full version of this paper [**?**].

2.5 One-to-one Equivocal Commitment

The scheme presented in Section 2.4 is not necessarily a one-to-one commitment scheme, since for PRG G, there may exist two different seeds s and s' such that $G(s) = G(s')$. In this case both s, s' are valid decommitments of the same bit.

We therefore, present a modification of the above scheme that allows us to achieve an equivocal commitment scheme with the one-to-one property: for every statistically binding commitment, there is at most a single opening string that will be accepted by the receiver during the decommitment phase. As an underlying ingredient, we use any commitment scheme $\Pi = (\mathsf{CRSGen}_\Pi, \mathsf{Commit}_\Pi, \mathsf{Open}_\Pi)$ (not necessarily equivocal) with the above property. Such a commitment scheme can be constructed straightforwardly e.g. from *injective* one-way functions. Let $n > 0$ be an integer, let $G : \{0,1\}^n \to \{0,1\}^{3n}$ be PRG.

- $\mathsf{CRSGen}(1^n)$: Run $\mathsf{CRSGen}_\Pi(1^n)$ to generate Σ_Π. Output $\Sigma = \Sigma_\Pi, \Sigma_1, \Sigma_2$ where Σ_1, Σ_2 are uniform random strings of length $3n$.
- $\mathsf{Commit}(\Sigma, b)$: Choose uniform random seeds $s_1, s_2 \in \{0,1\}^n$ and compute $t_1 = G(s_1)$, $t_2 = G(s_2)$. Choose $\beta \in \{0,1\}$. Set $c^1 = t_1 \oplus (b \cdot \Sigma_1)$. Set $c^2 = t_2 \oplus (\beta \cdot \Sigma_2)$. Generate $(com_\beta, s_\Pi) \leftarrow \mathsf{Commit}(\Sigma_\Pi, s_1||s_2)$ and $(com_{1-\beta}, \cdot) \leftarrow \mathsf{Commit}(\Sigma_\Pi, 0^{2n})$. Output commitment (c^1, c^2, com_0, com_1) along with decommitment information $(\beta||s_1||s_2||s_\Pi)$. In the following, we sometimes write $\mathsf{Commit}(\Sigma, b; \beta)$, explicitly including the randomness β in the input.
- $\mathsf{Open}(\Sigma, c, s)$: Parse $c = (c^1, c^2, com_0, com_1)$ and $s = \beta||s_1||s_2||s_\Pi$. If $c^2 = G(s_2)$, check that $\beta = 0$. If $c^2 = G(s_2) \oplus \Sigma_2$, check that $\beta = 1$. Run $\mathsf{Open}(\Sigma_\Pi, com_\beta, s_\Pi)$ and check that it outputs $s_1||s_2$. Otherwise, output $\perp$. If $c^1 = G(s_1)$, output 0. If $c^1 = G(s_1) \oplus \Sigma_1$, output 1. Output $\perp$ otherwise.

Clearly, by the binding of the original commitment scheme and the one-to-one property of Π, the modified scheme has the one-to-one property.

To create equivocal commitments/openings one can do the following: Run $\mathsf{CRSGen}_\Pi(1^n)$ to generate Σ_Π. Choose uniform random seeds $s_1^0, s_1^1, s_2^0, s_2^1 \in \{0,1\}^n$ and compute $t_1^0 = G(s_1^0)$, $t_2^0 = G(s_2^0)$, $t_1^1 = G(s_1^1)$, $t_2^1 = G(s_2^1)$. Choose $\beta \leftarrow \{0,1\}$ Generate $com_\beta = \mathsf{Commit}(\Sigma_\Pi, s_1^0||s_2^\beta)$ and $com_{1-\beta} = \mathsf{Commit}(\Sigma_\Pi, s_1^1||s_2^{1-\beta})$. Set $c^1 = t_1^0$. Set $c^2 = t_2^0$. Set $\Sigma_1 = c^1 \oplus t_1^1$. Set $\Sigma_2 = c^2 \oplus t_2^1$. Output (c^1, c^2, com_0, com_1).

To open the commitment to a 0, output $(\beta||s_1^0||s_2^\beta||open_\beta)$, where $open_\beta$ is the decommitment information for com_β.

To open the commitment to a 1, output $(1-\beta||s_1^1||s_2^{1-\beta}||open_{1-\beta})$, where $open_{1-\beta}$ is the decommitment information for $com_{1-\beta}$.

We note the following important property: For any commitment string c, and any CRS Σ, any two valid openings for c $s = \beta||s_1||s_2||s_\Pi$, $s' = \beta'||s_1'||s_2'||s_\Pi'$, it must be the case that $\beta \neq \beta'$.

2.6 Equivocal Commitment (with extra properties) in the CRS model

Let $\Pi' = (\mathsf{Gen}'_{Com}, \mathsf{Com}', \mathsf{Open}', \mathsf{S}'_{Eq})$, be an equivocal, one-to-one bit commitment scheme in the CRS model (given in Section 2.5). Let $(\mathsf{Gen}_{\mathsf{Sign}}, \mathsf{Sign}, \mathsf{Verify})$ be a strong, one-time signature scheme (for definition, see the full version [**?**]). We construct $\Pi = (\mathsf{Gen}_{Com}, \mathsf{Com}, \mathsf{Open}, \mathsf{S}_{Eq})$, which is an equivocal commitment scheme, with several additional properties that we describe at the end of the section and which will be useful for our constructions in Section 4.

Key generation Gen_{Com} is as follows: On input security parameter 1^λ, run Gen'_{Com} $2t\cdot\ell$ times to generate t pairs of vectors of CRS's $[(\Sigma_{Eq}^{0,i,j}, \Sigma_{Eq}^{1,i,j})]_{i\in[\ell],j\in[t]}$, where t is the length of the verification key vk output by $\mathsf{Gen}_{\mathsf{Sign}}$.
Commitment Com is as follows: To commit to a message $m := m_1, \ldots, m_\ell$ of length ℓ, generate a key pair $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Gen}_{\mathsf{Sign}}$. For $i \in [\ell]$, choose $\beta_i \leftarrow \{0,1\}$ at random. For $i \in [\ell], j \in [t]$, generate $(com_{i,j}, d_{i,j}) \leftarrow \mathsf{Com}'(\Sigma^{\mathsf{vk}_j,i,j}, m_i; \beta_i)$, where for each $i \in [\ell]$, $[com_{i,j}]_{j\in[t]}$ is the (bit-by-bit) commitment and $[d_{i,j}]_{j\in[t]}$ is the (bit-by-bit) decommitment information. Generate $\sigma \leftarrow \mathsf{Sign}_{\mathsf{sk}}([com_{i,j}]_{i\in[\ell],j\in[t]})$. Output commitment $com = (\mathsf{vk}, [com_{i,j}]_{i\in[\ell],j\in[t]}, \sigma)$. A sender can decommit separately to any set of bits of the message m. Decommitment information for a set S of message bits consists of $d[S] = [d_{i,j}]_{i\in S,j\in[t]}$, where $d_{i,j}$ is the decommitment information corresponding to the j-th bit of the i-th instance. We also denote the decommitment for the m_i as $d_i := d_{i,1}, d_{i,2}, \ldots d_{i,t}$.
Decommitment Open w.r.t. a set S: Given a set S, a commitment com, and an opening $[d_{i,j}]_{i\in S,j\in[t]}$, Open does the following: Parse commitment as $(\mathsf{vk}, [com_{i,j}]_{i\in[\ell],j\in[t]}, \sigma)$. (1) Check that $\mathsf{Verify}_{\mathsf{vk}}([com_{i,j}]_{i\in[\ell],j\in[t]}, \sigma) = 1$ (2) For $i \in S$, $j \in [t]$, check that $d_{i,j}$ is a valid decommitment for $com_{i,j}$ w.r.t. CRS $\Sigma^{\mathsf{vk}_j,i,j}$.
Equivocal CRS generation and commitment S_{Eq} is as follows: On input security parameter 1^λ, generate a key pair $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Gen}_{\mathsf{Sign}}$. Run S'_{Eq} $t\cdot\ell$ times to generate $[\Sigma^{\mathsf{vk}_j,i,j}]_{i\in[\ell],j\in[t]}$, equivocal commitments $[com_{i,j}]_{i\in[\ell],j\in[t]}$ and decommitments $[(d^0_{i,j}, d^1_{i,j})]_{i\in[\ell],j\in[t]}$. Note that for each $i \in [\ell]$, all equivocal commitments use the same value of $\beta := \beta_i$. Run Gen'_{Com} $t\cdot\ell$ times to generate $[\Sigma^{1-\mathsf{vk}_j,i,j}]_{i\in[\ell],j\in[t]}$. Set $\Sigma_{Eq} := [(\Sigma_{Eq}^{0,i,j}, \Sigma_{Eq}^{1,i,j})]_{i\in[\ell],j\in[t]}$. Compute $\sigma \leftarrow \mathsf{Sign}_{\mathsf{sk}}([com_{i,j}]_{i\in[\ell],j\in[t]})$. Output $(\Sigma = \Sigma_{Eq}, \overline{com} = (\mathsf{vk}, [com_{i,j}]_{i\in[\ell],j\in[t]}, \sigma)$, $d^0 = [d^0_{i,j}]_{i\in[\ell],j\in[t]}, d^1 = [d^1_{i,j}]_{i\in[\ell],j\in[t]})$.
Additional Check functionality: Given a Σ and commitments $com = (\mathsf{vk}, [com_{i,j}]_{i\in[\ell],j\in[t]}, \sigma)$, $com' = (\mathsf{vk}', [com'_{i,j}]_{i\in[\ell],j\in[t]}, \sigma')$, $\mathsf{Check}_\Sigma(com, com')$ outputs 1 if (1) $\mathsf{vk} = \mathsf{vk}'$; (2) $\mathsf{Verify}_{\mathsf{vk}}([com'_{i,j}]_{i\in[\ell],j\in[t]}, \sigma') = 1$.

Additional properties:

1. With overwhelming probability over generation of Σ, for every set $S \subseteq [\ell]$ and every string com, there is at most a *single* string $d[S]$ such that $\mathsf{Open}_\Sigma(S, com, d[S]) = 1$. This property is achieved by using the equivocal, one-to-one, commitment scheme given in Section 2.5 as the underlying commitment scheme.
2. Given a pair (Σ, com), a PPT adversary outputs com' such that $com \neq com'$ but $\mathsf{Check}_\Sigma(com, com') = 1$ with negligible probability. This property follows from the security of the one-time signature scheme.
3. Given equivocal commitment $(\Sigma_{Eq}, \overline{com})$, for every string com', if $\mathsf{Check}_{\Sigma_{Eq}}(\overline{com}, com') = 0$ then (with overwhelming probability over generation of Σ_{Eq}) com' has at most one valid opening. Specifically, for every set $S \subseteq [\ell]$, there is at most a *single* string $d[S]$ such that $\mathsf{Open}_{\Sigma_{Eq}}(S, com', d[S]) = 1$. Again, this property is achieved by using the equivocal, one-to-one, commitment scheme given in Section 2.5 as the underlying commitment scheme.

We elaborate on the third property, since it is less straightforward than the first two. First, note that the third property is a type of "simulation soundness" property, which essentially says that given an equivocal commitment, the only way to construct a *different* commitment with more than one valid opening is by forging a signature. This type of construction, where the CRS is indexed by bits of a signature verification key, has been used in various settings in the literature, such as in the construction of one-time simulation-sound NIZK, as well as CCA-secure encryption and non-malleable encryption [**?**, **?**, **?**, **?**, **?**]. In more detail, assume the adversary is given an equivocal commitment $(\Sigma_{Eq}, \overline{com})$, where $\Sigma_{Eq} = [(\Sigma_{Eq}^{0,i,j}, \Sigma_{Eq}^{1,i,j})]_{i\in[\ell],j\in[t]}$, and $\overline{com} = (\mathsf{vk}, [com_{i,j}]_{i\in[\ell],j\in[t]}, \sigma)$. It is sufficient to show that any commitment output by the adversary $com' = (\mathsf{vk}', [com'_{i,j}]_{i\in[\ell],j\in[t]}, \sigma')$, where $\mathsf{vk}' \neq \mathsf{vk}$ can have at most a single valid opening relative to any set $S \subseteq [\ell]$. Assume $\mathsf{vk}' \neq \mathsf{vk}$ and that com' has two valid openings relative to a set S. These openings must be of the form $[s^{i,j,0}]_{i\in S,j\in[t]} = [\beta_i||s_1^{i,j,0}||s_2^{i,j,0}||s_{\Pi}^{i,j,0}]_{i\in S,j\in[t]}$ $[s^{i,j,0}]_{i\in S,j\in[t]} = [1-\beta_i||s_1^{i,j,1}||s_2^{i,j,1}||s_{\Pi}^{i,j,1}]_{i\in S,j\in[t]}$. Since $\mathsf{vk}' \neq \mathsf{vk}$ there must be at least one $j \in [t]$ such that $\mathsf{vk}'_j = 1 - \mathsf{vk}_j$. But $[\Sigma^{1-\mathsf{vk}_j,i,j}]_{i\in[\ell],j\in[t]}$ were generated via Gen'_{Com}, so it is guaranteed with overwhelming probability that any string $com'_{i,j}$ relative to $\Sigma^{1-\mathsf{vk}_j,i,j}$ has at most a single valid decommitment. Therefore $\beta_i||s_1^{i,j,0}||s_2^{i,j,0}||s_{\Pi}^{i,j,0}$ and $1-\beta_i||s_1^{i,j,1}||s_2^{i,j,1}||s_{\Pi}^{i,j,1}$ cannot both be valid decommitments, leading to contradiction.

3 Impossibility of CNMC with no CRS

In this section we present Theorem 5, stating the impossibility of constructing CNMC without CRS.

Theorem 5. *There is no black-box reduction from a single-bit CNMC scheme $\Pi = (\mathsf{E}, \mathsf{D})$ to any falsifiable assumption, unless the assumption is false.*

We know from prior work that continuous NMC are impossible in the info-theoretic setting. Assume we have a construction of single-bit, continuous NMC from some falsifiable assumption with no CRS. We only allow black-box usage of the adversary in the reduction. However, the underlying assumption can be used in a non-black-box way in the construction/proof.

Preliminaries. Given adversary $A = (A_L, A_R)$, we say that A has advantage α in the *simplified no-Σ CNMC game* against construction $\Pi = (\mathsf{E}, \mathsf{D})$ if:

$$\Big|\Pr[\mathsf{D}(A_L(L), A_R(R)) \neq \perp \mid (L,R) \leftarrow \mathsf{E}(1^n, 0)] - \Pr[\mathsf{D}(A_L(L), A_R(R)) \neq \perp \mid (L,R) \leftarrow \mathsf{E}(1^n, 1)]\Big| = \alpha,$$

Clearly, if $A = (A_L, A_R)$ has non-negligible advantage in the *simplified no-Σ CNMC game*, it can be used to break the CNMC security of $\Pi = (\mathsf{E}, \mathsf{D})$.

Definition 11. *A tuple* (x, y, z) *is* bad *relative to CNMC scheme* $\Pi = (\mathsf{E}, \mathsf{D})$ *if either:*

- $y \neq z \wedge \mathsf{D}(x, y) \neq \perp \wedge \mathsf{D}(x, z) \neq \perp$ *OR*
- $x \neq y \wedge \mathsf{D}(x, z) \neq \perp \wedge \mathsf{D}(y, z) \neq \perp$.

Definition 12. *A single-bit CNMC* $\Pi = (\mathsf{E}, \mathsf{D})$ *in the standard (no CRS model) is* perfectly unique *if there exist no bad tuples relative to* $\Pi = (\mathsf{E}, \mathsf{D})$.

We next present the following two lemmas, which, taken together, imply Theorem 5.

Lemma 1. *If a single-bit CNMC scheme* $\Pi = (\mathsf{E}, \mathsf{D})$ *is* not *perfectly unique then it is insecure.*

This is immediate, since if a bad tuple exists, it can be given to the adversary as non-uniform advice. Then the same attack from the literature (reviewed in the introduction) can be run.

Lemma 2. *There is no BB reduction from a single-bit CNMC scheme* $\Pi = (\mathsf{E}, \mathsf{D})$ *which is perfectly unique to any falsifiable assumption.*

The basic idea is that, given only black-box access to the split-state adversary, $A = (A_L, A_R)$, the reduction cannot tell the difference between the actual adversary and a *simulated* adversary. The simulated adversary simply waits to get matching L and R queries from the reduction, decodes, and re-encodes a fresh value that is related to the decoded value. The challenges are that the L and R queries are not received simultaneously. In fact, there could be many queries interleaved between a L and R match. So the simulated adversary must return a value upon seeing the L or R half *before* seeing the other half and *before* knowing whether the encoded value is a 0 or a 1. Therefore, the simulated adversary does the following: It keeps a table containing all the L and R values that it has seen. Whenever a L or R query is made, the simulated adversary first checks the table to see if a matching query was previously made. If not, the simulated adversary chooses a random encoding, (L', R'), of a random bit b', stores it in the table along with the L/R query that was made and returns either L' or R' as appropriate. If yes, the simulated adversary finds the corresponding L/R along with the pair (L', R') stored in the table. The simulated adversary then decodes (L, R) to find out b. If $b = 0$, the simulated adversary returns either L' or R' as appropriate. Otherwise, the simulated adversary returns the left/right side of an encoding of a random bit b''. We prove that the view generated by the reduction interacting with this adversary is identical to the view of the reduction interacting with the following real adversary: The real adversary, given L or R, recovers the corresponding unique valid codeword (L, R) (if it exists) and decodes to get the bit b. If $b = 0$, the real adversary encodes a random bit $b' = \mathsf{RO}_1(L||R)$ using randomness $r = \mathsf{RO}_2(L||R)$ (where $\mathsf{RO}_1, \mathsf{RO}_2$ are random oracles internal to the real adversary that are used to generate consistent randomness across invocations) and outputs the left/right

side as appropriate. Otherwise (i.e. if the corresponding unique codeword does not exist or if $\mathsf{D}(L, R) = 1$), the real adversary outputs the left/right side of encoding of a random bit, $b'' = \mathsf{RO}_3(L)$ (or $b'' = \mathsf{RO}_3(R)$) using randomness $r'' = \mathsf{RO}_4(L)$ (or $r'' = \mathsf{RO}_4(R)$) (where $\mathsf{RO}_3, \mathsf{RO}_4$ are random oracles internal to the real adversary that are used to generate consistent randomness across invocations). Note that since the CNMC is perfectly unique, the real adversary obtains non-negligible advantage of $1 - \mathsf{negl}(n)$ in the simplified no-Σ CNMC game.

Proof. We will construct a *meta-reduction* as follows:

Consider the following inefficient, split state adversary $A = (A_L, A_R)$ with internal random oracles $\mathsf{RO}_1, \mathsf{RO}_2$, RO_3, and RO_4:

A_L: On input L, find the unique R such that $\mathsf{D}(L, R) \neq \perp$ (if it exists). Let $b := \mathsf{D}(L, R)$. If $b = 0$, encode $b' = \mathsf{RO}_1(L||R)$ using randomness $r = \mathsf{RO}_2(L||R)$ to obtain $(L', R') := \mathsf{E}(b'; r)$ and output L'. If such R does not exist or if $b = 1$, compute a random encoding of a random bit $b'' = \mathsf{RO}_3(L)$ using randomness $r'' = \mathsf{RO}_4(L)$ to obtain $(L'', R'') := \mathsf{E}(b'', r'')$ and output L''.

A_R: On input R, find the unique L such that $\mathsf{D}(L, R) \neq \perp$ (if it exists). Let $b := \mathsf{D}(L, R)$. If $b = 0$, encode $b' = \mathsf{RO}_1(L||R)$ using randomness $r = \mathsf{RO}_2(L||R)$ to obtain $(L', R') := \mathsf{E}(b'; r)$ and output R'. If such L does not exist or if $b = 1$, compute a random encoding of a random bit $b'' = \mathsf{RO}_3(R)$ using randomness $r'' = \mathsf{RO}_4(R)$ to obtain $(L'', R'') := \mathsf{E}(b'', r'')$ and output R''.

Clearly, A succeeds with advantage $1 - \mathsf{negl}(n)$ in the simplified no-Σ CNMC game.

The following adversary A' simulates the above efficiently: Let T be a table that records internal randomness. T is initialized to empty. A' is a stateful adversary that proceeds as follows:

1. On input L, check if the corresponding R such that $\mathsf{D}(L, R) \neq \perp$ has been queried. If yes, decode to get bit $b := \mathsf{D}(L, R)$. If $b = 0$, check the table T to recover (R, L', R'). Output L'. Otherwise, if $L \in T$ then output L'' corresponding to entry (L, L'', R''). If $L \notin T$, choose a random encoding of a random bit b'': $(L'', R'') \leftarrow \mathsf{E}(b'')$. Store (L, L'', R'') in T. and output L''.
2. On input R, check if the corresponding L such that $\mathsf{D}(L, R) \neq \perp$ has been queried. If yes, decode to get bit $b := \mathsf{D}(L, R)$. If $b = 0$, check the table T to recover (L, L', R'). Output R'. Otherwise, if $R \in T$ then output R'' corresponding to entry (R, L'', R''). If $R \notin T$, choose a random encoding of a random bit b'': $(L'', R'') \leftarrow \mathsf{E}(b'')$. Store (R, L'', R'') in T and output R''.

By properties of the random oracle, the view of the reduction **Red** when interacting with A versus A' are equivalent.

Since the reduction succeeds when interacting with Real adversary A with non-negligible probability p and since the view of the reduction is identical when interacting with A or A', **Red** interacting with A' must also succeed with non-negligible probability p. But **Red** composed with A' yields an efficient adversary, leading to an efficient adversary breaking the underlying falsifiable assumption, which is a contradiction.

4 2-State CNMC for One-Bit Messages

In this section we prove the following theorem:

Theorem 6. *Assuming the existence of one-to-one commitment schemes in the CRS model, there is a construction of a 2-split-state CNM Randomness Encoder in the CRS model.*

The corollary is immediate, given the transformation in the full version [**?**].

Corollary 1. *Assuming the existence of one-to-one commitment schemes in the CRS model, there is a construction of a single-bit, 2-split-state CNMC in the CRS model.*

Notation and parameters. λ is security parameter and length of encoded randomness. $\ell = \ell(\lambda) \in \Theta(\lambda^2)$ and we assume for simplicity that $\lambda|\ell$. Sets $S_L, S_R \subseteq [2\ell]$ are defined as follows: $S_L = [\ell], S_R = [2\ell] \setminus [\ell]$. $y_o = y_o(\ell) \in \Theta(\ell^{1/2})$, $y_t = y_t(\ell) \in \Theta(\ell^{1/2})$.
The construction of the 2-state CNM Randomness Encoder is given in Figure 1.

Let $(\mathsf{CRSGen}_{\mathsf{Com}}, \mathsf{Com}, \mathsf{Open}, \mathsf{S}_{Eq})$ be the non-interactive, equivocal, one-to-one commitment in the CRS model given in Section 2.6.

$\mathsf{CRSGen}(1^\lambda)$: $\Sigma \leftarrow \mathsf{CRSGen}_{\mathsf{Com}}(1^\lambda)$. Output Σ.

$\mathsf{E}_\Sigma(c_L||c_R||r_{com})$:

1. Parse c_L, c_R as strings in $\mathbb{F}_{2^\lambda}^{\frac{\ell}{\lambda}}$.
2. $(com, d = d_1, \ldots, d_{2\ell}) \leftarrow \mathsf{Com}_\Sigma(c_L||c_R; r_{com})$
3. Let $d[S_L]$ (resp. $d[S_R]$) correspond to the decommitment of com to the bits corresponding to S_L (resp. S_R).
4. $\mathsf{E}_{2,\Sigma}$ outputs $L = (com, d[S_L])$; $R = (com, d[S_R])$. $\mathsf{E}_{1,\Sigma}$ outputs $\langle c_L, c_R \rangle$.

$\mathsf{D}_\Sigma(\widetilde{L}, \widetilde{R})$:

1. Parse $\widetilde{L} = (\widetilde{com}, \widetilde{d}[S_L])$, $\widetilde{R} = (\widetilde{com}', \widetilde{d}[S_R])$.
2. Check that $\widetilde{com} = \widetilde{com}'$.
3. Let $\widetilde{c}_L = \mathsf{Open}_\Sigma(S_L, \widetilde{com}, \widetilde{d}[S_L])$ and $\widetilde{c}_R = \mathsf{Open}_\Sigma(S_R, \widetilde{com}, \widetilde{d}[S_R])$. Check that $\widetilde{c}_L \neq \perp$ and $\widetilde{c}_R \neq \perp$.
4. If all the above checks pass, output $\langle \widetilde{c}_L, \widetilde{c}_R \rangle$. Otherwise, output $\perp$.

Fig. 1. Construction of 2-State, Continuous, Non-Malleable Randomness Encoder.

To prove Theorem 6, we show that the construction above is a secure CNM Randomness Encoder, via the following sequence of hybrids.

Hybrid 0: This is the "Real" security experiment.

Hybrid 1: The experiment is identical to Hybrid 0 except we modify the decode algorithm from D_Σ to D^1_Σ to abort if the tampered codeword submitted is different from the challenge codeword and the Check function outputs 1. Specifically, let $(L := (com, d[S_L]), R = (com, d[S_R]))$ be the "challenge" codeword (i.e. the codeword generated by the security experiment).

$\mathsf{D}^1_\Sigma(\widetilde{L}, \widetilde{R})$:

1. Parse $\widetilde{L} = (\widetilde{com}, \widetilde{d}[S_L])$, $\widetilde{R} = (\widetilde{com}', \widetilde{d}[S_R])$.
2. If $\widetilde{L} \neq L$ and $\mathsf{Check}_\Sigma(com, \widetilde{com}) = 1$ or $\widetilde{R} \neq R$ and $\mathsf{Check}_\Sigma(com, \widetilde{com}') = 1$ then output $\perp$.
3. Check that $\widetilde{com} = \widetilde{com}'$.
4. Let $\widetilde{c}_L = \mathsf{Open}_\Sigma(S_L, \widetilde{com}, \widetilde{d}[S_L])$ and $\widetilde{c}_R = \mathsf{Open}_\Sigma(S_R, \widetilde{com}, \widetilde{d}[S_R])$. Check that $\widetilde{c}_L \neq \perp$ and $\widetilde{c}_R \neq \perp$.
5. If all the above checks pass, output $\langle \widetilde{c}_L, \widetilde{c}_R \rangle$. Otherwise, output $\perp$.

Fig. 2. Decode in Hybrid 1.

Hybrid 2: The experiment is identical to Hybrid 1, except we switch to equivocal commitments in the codeword (L, R) that is given to the adversary. Specifically, CRSGen is replaced with CRSGen^2 and the challenge codeword is generated as shown in Figure 3.

$\mathsf{CRSGen}^2(1^\lambda)$: $(\Sigma_{Eq}, \overline{com}, d^0 = d^0_1 \dots d^0_{2\ell}, d^1 = d^1_1 \dots d^1_{2\ell}) \leftarrow \mathsf{S}_{Eq}(1^\lambda)$. Output Σ_{Eq}.
Challenge codeword:

1. Sample c_L, c_R uniform randomly from $\mathbb{F}_{2^\lambda}^{\ell}$.
2. Set $d[S_L] := [d_i^{c_L[i]}]_{i \in S_L}$; Set $d[S_R] := [d_i^{c_R[i]}]_{i \in S_R}$;
3. Output $L = (\overline{com}, d[S_L])$; $R = (\overline{com}, d[S_R])$.

Fig. 3. Gen and Challenge Codeword generation in Hybrid 2.

Hybrid 3: The experiment is identical to Hybrid 2, except we modify D^1 to D^3, which aborts if the outcome of $f^i_L(L)$ or $f^i_R(R)$ is not a "likely value."

Specifically, given $(\Sigma_{Eq}, \overline{com}, d^0 = d^0_1 \dots d^0_{2\ell}, d^1 = d^1_1 \dots d^1_{2\ell})$ and the adversary's current output $\mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1}$, we define the sets $\mathcal{S}_L$, $\mathcal{S}_R$, $\mathcal{S}'_L$, $\mathcal{S}'_R$ as:

- $\mathcal{S}_L$ contains all values of $\widehat{L}'$ that occur with probability at least $\epsilon = 1/2^{y_o/3}$, where values of $\widehat{L}'$ are sampled as follows: Sample $\widehat{c}_L$ conditioned on the output of the experiment in Hybrid 2 thus far being equal to $\mathsf{Out}_A^{i-1} =$

$\widehat{Out}_A^{i-1}$. Compute equivocal decommitment of $\overline{com}$: $\widehat{d}[S_L] := [d_i^{\widehat{c}_L[i]}]_{i \in S_L}$. Apply f_L^i to $\widehat{L} = (\overline{com}, \widehat{d}[S_L])$ to obtain $\widehat{L}'$ (or "same" if the output is $\widehat{L}$ itself).

- $\mathcal{S}_R$ contains all values of $\widehat{R}'$ that occur with probability at least $\epsilon = 1/2^{y_o/3}$, where values of $\widehat{R}'$ are sampled as follows: Sample $\widehat{c}_R$ conditioned on the output of the experiment in Hybrid 2 thus far being equal to $\mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1}$. Compute equivocal decommitment of $\overline{com}$: $\widehat{d}[S_R] := [d_i^{\widehat{c}_R[i]}]_{i \in S_R}$. Apply f_R^i to $\widehat{R} = (\overline{com}, \widehat{d}[S_R])$ to obtain $\widehat{R}'$ (or "same" if the output is $\widehat{R}$ itself).
- Let $\mathcal{S}'_L \subseteq \mathcal{S}_L$ be the set of $\widehat{L}'$ such that there is a "matching" $\widehat{R}' \in \mathcal{S}_R$ such that $\mathsf{D}^1_{\Sigma_{Eq}}(\widehat{L}', \widehat{R}') \neq \bot$.
- Let $\mathcal{S}'_R \subseteq \mathcal{S}_R$ be the set of $\widehat{R}'$ such that there is a "matching" $\widehat{L}' \in \mathcal{S}_L$ such that $\mathsf{D}^1_{\Sigma_{Eq}}(\widehat{L}', \widehat{R}') \neq \bot$.

Note that the decode oracle is now stateful and depends on the current round of interaction, as well as the outputs returned in previous rounds. Specifically, note that the sets $\mathcal{S}'_L$, $\mathcal{S}'_R$ change in each round i, since the likely outputs depend on the tampering function (f_L^i, f_R^i) submitted by the adversary in round i, and are conditioned on the output $\mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1}$ seen by the adversary thus far in rounds $1, \ldots, i-1$.

$\mathsf{D}^3_{\Sigma_{Eq}}((f_L^i, f_R^i), \widetilde{L}, \widetilde{R})$:

1. Check that $\widetilde{L} \in \mathcal{S}'_L$ and that $\widetilde{R} \in \mathcal{S}'_R$. If not, output $\bot$.
2. Parse $\widetilde{L} = (\widetilde{com}, \widetilde{d}[S_L])$, $\widetilde{R} = (\widetilde{com}', \widetilde{d}[S_R])$.
3. Check that $\widetilde{com} = \widetilde{com}'$.
4. Let $\widetilde{c}_L = \mathsf{Open}_\Sigma(S_L, \widetilde{com}, \widetilde{d}[S_L])$ and $\widetilde{c}_R = \mathsf{Open}_\Sigma(S_R, \widetilde{com}, \widetilde{d}[S_R])$. Check that $\widetilde{c}_L \neq \bot$ and $\widetilde{c}_R \neq \bot$.
5. If all the above checks pass, output $\langle \widetilde{c}_L, \widetilde{c}_R \rangle$. Otherwise, output $\bot$.

Fig. 4. Decode in Hybrid 3.

Hybrid 4: The experiment is identical to Hybrid 3, except we modify D^3 to D^4 which aborts if there are more than y_t number of queries f_L^i (resp. f_R^i) such that the outcome of $f_L^i(L)$ (resp. $f_R^i(R)$) is not the most "likely value". Specifically, at the beginning of the experiment, we initialize counters $\mathsf{count}_L, \mathsf{count}_R$ to 0. We also define L^* (resp. R^*) to be the element of $\mathcal{S}'_L$ (resp. $\mathcal{S}'_R$) that occurs most frequently. More precisely, we consider the sets

$$\mathcal{L}^* := \mathsf{argmax}_{L' \in \mathcal{S}'_L} \Pr[f_L^i(\widehat{L}) = L' \mid \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1}].$$

$$\mathcal{R}^* := \mathsf{argmax}_{R' \in \mathcal{S}'_R} \Pr[f_R^i(\widehat{R}) = R' \mid \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1}].$$

Then L^* (resp. R^*) is defined to be the lexicographically first element in $\mathcal{L}^*$ (resp. $\mathcal{R}^*$).

$\mathsf{D}^4_{\Sigma_{Eq}}((f^i_L, f^i_R), \widetilde{L}, \widetilde{R})$:

1. Check that $\widetilde{L} \in \mathcal{S}'_L$ and that $\widetilde{R} \in \mathcal{S}'_R$. If not, output $\perp$.
2. If $\widetilde{L} \neq L^*$, then set $\mathsf{count}_L := \mathsf{count}_L + 1$.
3. If $\widetilde{R} \neq R^*$, then set $\mathsf{count}_R := \mathsf{count}_R + 1$.
4. If $\mathsf{count}_L > y_t$ or $\mathsf{count}_R > y_t$, output $\perp$.
5. Parse $\widetilde{L} = (\widetilde{com}, \widetilde{d}[S_L])$, $\widetilde{R} = (\widetilde{com}', \widetilde{d}[S_R])$.
6. Check that $\widetilde{com} = \widetilde{com}'$.
7. Let $\widetilde{c}_L = \mathsf{Open}_\Sigma(S_L, \widetilde{com}, \widetilde{d}[S_L])$ and $\widetilde{c}_R = \mathsf{Open}_\Sigma(S_R, \widetilde{com}, \widetilde{d}[S_R])$. Check that $\widetilde{c}_L \neq \perp$ and $\widetilde{c}_R \neq \perp$.
8. If all the above checks pass, output $\langle \widetilde{c}_L, \widetilde{c}_R \rangle$. Otherwise, output $\perp$.

Fig. 5. Decode in Hybrid 4.

Claim 4.1. Hybrids 0 and 1 are computationally indistinguishable.

This follows from the additional properties of the equivocal commitment scheme given in Section 2.6.

Claim 4.2. Hybrids 1 and 2 are computationally indistinguishable.

This follows from the security of the equivocal commitment scheme.

Claim 4.3. Hybrids 2 and 3 are $\epsilon \cdot 2q$-close, where $\epsilon = 1/2^{y_o/3}$ and $y_o \in O(\ell^{1/2})$.

Proof. To prove indistinguishability of Hybrids 2 and 3, it is sufficient to show that for each $i \in [q]$, $\Pr[f^i_L(L) \notin \mathcal{S}'_L \wedge \mathsf{D}^1_{\Sigma_{Eq}}(f^i_L(L), f^i_R(R)) \neq \perp] \leq \epsilon$ and $\Pr[f^i_L(R) \notin \mathcal{S}'_R \wedge \mathsf{D}^1_{\Sigma_{Eq}}(f^i_L(L), f^i_R(R)) \neq \perp] \leq \epsilon$. The result then follows by a union bound over the q LHS and q RHS queries.

To bound the above, we in fact show something stronger: (1) for each $i \in [q]$, each value of $\mathsf{Out}^{i-1}_A = \widehat{Out}^{i-1}_A$ (which does not contain a $\perp$ output) and each value of $R = \widehat{R}$,

$$\Pr[f^i_L(L) \notin \mathcal{S}'_L \wedge \mathsf{D}^1_{\Sigma_{Eq}}(f^i_L(L), f^i_R(R)) \neq \perp \mid R = \widehat{R} \wedge \mathsf{Out}^{i-1}_A = \widehat{Out}^{i-1}_A)] \leq \epsilon;$$

and (2) for each $i \in [q]$, each value of $\mathsf{Out}^{i-1}_A = \widehat{Out}^{i-1}_A$ (which does not contain a $\perp$ output) and each value of $L = \widehat{L}$,

$$\Pr[f^i_R(R) \notin \mathcal{S}'_R \wedge \mathsf{D}^1_{\Sigma_{Eq}}(f^i_L(L), f^i_R(R)) \neq \perp \mid L = \widehat{L} \wedge \mathsf{Out}^{i-1}_A = \widehat{Out}^{i-1}_A)] \leq \epsilon.$$

We first fix $(\Sigma_{Eq}, \overline{com}, d^0 = d^0_1 \ldots d^0_{2\ell}, d^1 = d^1_1 \ldots d^1_{2\ell})$. Note that for fixed $\Sigma_{Eq}, \overline{com}, d^0 = d^0_1 \ldots d^0_{2\ell}, d^1 = d^1_1 \ldots d^1_{2\ell}$, there is a bijection ϕ_L (resp. ϕ_R)

between c_L (resp. c_R) and $(\overline{com}, d[S_L])$ (where $d[S_L] := [d_i^{c_L[i]}]_{i \in S_L}$). Therefore the probability of a particular value of c_L (resp. c_R) occurring is equivalent to the probability of $L = \phi_L(c_L)$ (resp. $R = \phi_R(c_R)$) occurring. Additionally, Let ρ_L (resp. ρ_R) be the function that given $f_R^i(R)$ (resp. $f_L^i(L)$) returns the unique L' (resp. R') if it exists such that, $\mathsf{D}^1_{\Sigma_{Eq}}(L', f_R^i(R)) \neq \perp$ (resp. $\mathsf{D}^1_{\Sigma_{Eq}}(f_L^i(L), R') \neq \perp$). Note that L' (resp. R') is equal to "same" if and only if $f_R^i(R) =$ "same" (resp. $f_L^i(L) =$ "same"). To see why this is so, recall that in D^1, $\perp$ is outputted if $\widetilde{L} \neq L$ and $\mathsf{Check}_\Sigma(com, \widetilde{com}) = 1$ or $\widetilde{R} \neq R$ and $\mathsf{Check}_\Sigma(com, \widetilde{com'}) = 1$. Now, if L' is equal to same, then it must be that $\mathsf{Check}_\Sigma(com, \widetilde{com}) = 1$. Therefore, by the above, the only value of $f_R^i(R)$, for which $\perp$ will not be output is $f_R^i(R) =$ "$same$". The same is true for the case that $f_R^i(R) =$ "$same$".

We first show that for $i \in [q]$, c_L, c_R are conditionally independent given $\mathsf{Out}_A^i = \widehat{Out}_A^i$. This follows from the fact that the information contained in $\widehat{Out}_A^i$ is of the form $(f_L^1(\phi_L(c_L)) = v_1, f_R^1(\phi_R(c_R)) = w_1), \ldots, (f_L^i(\phi_L(c_L)) = v_i, f_R^i(\phi_R(c_R)) = w_i)$, where for $j \in [i]$, v_j is equal to the L' value outputted in response to the j-th query and w_j is equal to the R' value outputted in response to the j-th query. (note that v_j/w_j can be set to "same" if the tampering function leaves L/R unchanged). Thus, the distribution of c_L, c_R conditioned on $(f_L^1(\phi_L(c_L)) = v_1, f_R^1(\phi_R(c_R)) = w_1), \ldots, (f_L^i(\phi_L(c_L)) = v_i, f_R^i(\phi_R(c_R)) = w_i)$ is equal to $(U_\ell \mid (f_L^1(\phi_L(U_\ell)) = v_1, \ldots, f_L^i(\phi_L(U_\ell)) = v_i)) \times (U_\ell \mid (f_R^1(\phi_R(U_\ell)) = w_1, \ldots, f_R^i(\phi_R(U_\ell)) = w_i))$. Moreover, due to the discussion above, L, R are also conditionally independent given $\mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1}$. Therefore, to show (1), we note that for every $(\widehat{L}, \widehat{R}, \widehat{Out}_A^{i-1})$, $\Pr[L = \widehat{L} \mid R = \widehat{R} \wedge \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1})] = \Pr[L = \widehat{L} \mid \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1})]$. So we have that for every fixed $R = \widehat{R}$ (for which $\Pr[R = \widehat{R} \wedge \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1})] > 0$), and every $L' \notin \mathcal{S}'_L$, $\Pr[f^i(L) = L' \mid R = \widehat{R} \wedge \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1})] \leq \epsilon$. Therefore,

$$\begin{aligned}
&\Pr[f_L^i(L) \notin \mathcal{S}'_L \wedge \mathsf{D}^1_{\Sigma_{Eq}}(f_L^i(L), f_R^i(R)) \neq \perp \mid R = \widehat{R} \wedge \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1})] \\
&\quad = \Pr[f_L^i(L) \notin \mathcal{S}'_L \wedge \left(f_L^i(L) = \rho_L(f_R^i(R))\right) \mid R = \widehat{R} \wedge \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1})] \\
&\quad \leq \epsilon.
\end{aligned}$$

The proof for (2) is analogous.

Claim 4.4. Hybrids 3 and 4 are statistically indistinguishable.

Proof. To prove indistinguishability of Hybrids 3 and 4, we must show that the probability that the event (1) $f_L^i(L)$ is not most frequent and $\mathsf{D}^3_{\Sigma_{Eq}}(f_L^i(L), f_R^i(R)) \neq \perp$ or event (2) $f_R^i(R)$ is not most frequent and $\mathsf{D}^3_{\Sigma_{Eq}}(f_L^i(L), f_R^i(R)) \neq \perp$ occurs more than y_t times in a single execution is at most $(1/2)^{y_t}$.

We first analyze the event (1). Recall that set $\mathcal{S}'_L$ contains values, L', that occur with probability p in some experiment. By "most frequent value" in $\mathcal{S}'_L$,

we mean the value L' in $\mathcal{S}'_L$ with the maximum associated probability p. Note that if L' is not the most frequent value, the associated probability p is at most $1/2$, since otherwise, the probabilities will sum to more than 1. More precisely, if $f_L^i(L) = L'$ is not the most frequent query in $\mathcal{S}'_L$ then, by definition of the set $\mathcal{S}'_L$ and the above argument, $\Pr[f_L^i(\widehat{L}) = L' \mid \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1}] \leq 1/2$. Recall that in the proof of the previous claim, we have shown that for $i \in \{0, \ldots, q\}$, L, R are conditionally independent given Out_A^i. Therefore, $\Pr[f_L^i(L) = L' \mid \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1} \wedge R = \widehat{R}] \leq 1/2$. This implies that for every fixed $R = \widehat{R}$ (for which $\Pr[R = \widehat{R} \wedge \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1}] > 0$),

$$\begin{aligned}
&\Pr[f_L^i(L) \neq L^* \wedge \mathsf{D}^3_{\Sigma_{Eq}}(f_L^i(L), f_R^i(R)) \neq \perp \mid R = \widehat{R} \wedge \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1})] \\
&\quad \leq \Pr[f_L^i(L) \neq L^* \wedge f_L^i(L) = \rho_L(f_R^i(R)) \mid R = \widehat{R} \wedge \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1})] \\
&\quad \leq 1/2.
\end{aligned}$$

We consider the number of adversarial queries such that both $f_L^i(L) = L'$ is not the most frequent value $(L^*) \in \mathcal{S}'_L$ *and* $\mathsf{D}^3_{\Sigma_{Eq}}(f_L^i(L), f_R^i(R)) \neq \perp$. (note that the total number of adversarial queries can be much higher). By the above argument, the probability that there are y_t number of rounds i such that both $f_L^i(L) = L'$ is not the most frequent value $(L^*) \in \mathcal{S}'_L$ *and* $\mathsf{D}^3_{\Sigma_{Eq}}(f_L^i(L), f_R^i(R)) \neq \perp$ is at most $(1/2)^{y_t} \in \mathsf{negl}(\lambda)$. Thus, we have concluded the proof for event (1). The proof for event (2) is analogous.

We finally show the main technical claim of this section, which completes the proof of Theorem 6.

Claim 4.5. In Hybrid 4, the encoded randomness $\langle c_L, c_R \rangle$ is statistically close to uniform, given the view of the adversary.

Proof. Towards proving the claim, we consider the following leakage functions:

Leakage function on c_L: Fix $\Sigma_{Eq}, \overline{com}, d^0, d^1$, universal hash $h : \{0,1\}^\alpha \to \{0,1\}^{y_o} \in \mathcal{H}$ (where α is the length of a single split-state of the encoding) and adversary A. On input c_L, set output Out_A to "" and Out_L to "". Set $L = (\overline{com}, [d_i^{c_L[i]}]_{i \in [\ell]})$. Repeat the following in rounds $i = 1, 2, \ldots$:

1. Obtain the next tampering function (f_L, f_R) from adversary A. If A terminates then terminate with output Out_L.
2. Set $L' := f_L(L)$. If $L' \in \mathcal{S}'_L$, then:
   (a) Find the unique $\widehat{R}' \in \mathcal{S}'_R$ such that $\mathsf{D}^1_{\Sigma_{Eq}}(L', \widehat{R}') \neq \perp$. Return $(L', \widehat{R}')$ to the adversary. Set $\mathsf{Out}_A = \mathsf{Out}_A || (L', \widehat{R}')$.
   (b) If L' is not the most frequent output in $\mathcal{S}'_L$, set $\mathsf{Out}_L := \mathsf{Out}_L || (i || h(L'))$ If $|\mathsf{Out}_L| > (\log(q) + y_o) \cdot y_t$ then terminate with output $\mathsf{Out}_L := \mathsf{Out}_L || (i || \perp)$.
3. If $L' \notin \mathcal{S}'_L$, output $\perp$ to the adversary and terminate with output $\mathsf{Out}_L := \mathsf{Out}_L || (i || \perp)$.

The leakage function for the RHS is analogous.

We now show that given Out_L and Out_R we can reconstruct the full output sequence for the adversary's view with probability $1 - \frac{2q}{\epsilon^2 \cdot 2^{y_o}} = 1 - \frac{2q}{2^{y_0/3}}$ in the following way:

Fix $\Sigma_{Eq}, \overline{com}, d^0 = d_1^0 \dots d_{2\ell}^0, d^1 = d_1^1 \dots d_{2\ell}^1$, universal hash $h \leftarrow \mathcal{H}$ and adversary A. Set output Out_A to "" and Out_L to "". Repeat the following in rounds $i = 1, 2, \dots, q$:

1. Obtain the next tampering function (f_L, f_R) from adversary A given its current view, Out_A.
2. If $(i, \perp) \in \mathsf{Out}_L$ or $(i, \perp) \in \mathsf{Out}_R$, set $\mathsf{Out}_A = \mathsf{Out}_A || \perp$ and abort.
3. If $(i, y) \in \mathsf{Out}_L$, for some $y \neq \perp$, set $L' = \widehat{L}'$ such that $\widehat{L}' \in \mathcal{S}'_L$ and $h(\widehat{L}') = y$.
4. If $(i, \cdot) \notin \mathsf{Out}_L$, set $L' = \widehat{L}'$ such that $\widehat{L}' \in \mathcal{S}'_L$ is the most frequent value.
5. If $(i, y) \in \mathsf{Out}_R$, for some $y \neq \perp$, set $R' = \widehat{R}'$ such that $\widehat{R}' \in \mathcal{S}'_R$ and $h(\widehat{R}') = y$.
6. If $(i, \cdot) \notin \mathsf{Out}_R$, set $R' = \widehat{R}'$ such that $\widehat{R}' \in \mathcal{S}'_R$ is the most frequent value.
7. If L' = "same" and R' = "same" output "same" and set $\mathsf{Out}_A = \mathsf{Out}_A ||$ "same".
8. Else if one of L', R' is "same" and not the other, set $\mathsf{Out}_A = \mathsf{Out}_A || \perp$ and abort.
9. Else Parse $L' := (com, d[S_L])$ and $R' := (com', d[S_R])$. If $com \neq com'$, set $\mathsf{Out}_A = \mathsf{Out}_A || \perp$ and abort.
10. Otherwise, set $\mathsf{Out}_A = \mathsf{Out}_A || (L', R')$.

It can be determined by inspection that the incorrect value is output only if in one of the at most $2q$ instances, there are two distinct values $\widehat{L}', \widehat{L}'' \in \mathcal{S}'_L$ or $\widehat{R}', \widehat{R}'' \in \mathcal{S}'_R$ such that $h(\widehat{L}') = h(\widehat{L}'')$ or $h(\widehat{R}') = h(\widehat{R}'')$. Due to universality of h and the fact that $|\mathcal{S}'_L| = |\mathcal{S}'_R| = 1/\epsilon$, this can occur with probability at most $\frac{2q}{\epsilon^2 \cdot 2^{y_o}}$, as claimed. [3]

Since $|\mathsf{Out}_L| \leq (\log(q) + y_o) \cdot y_t \leq 2y_o \cdot y_t \leq c \cdot \ell$ for constant $c < 1$ and $|\mathsf{Out}_R| \leq (\log(q) + y_o) \cdot y_t \leq 2y_o \cdot y_t \leq c \cdot \ell$ for constant $c < 1$, we can use the properties of the inner product extractor (check the full version of this paper [**?**] for more details.) to argue that $\langle c_L, c_R \rangle$ is statistically close to uniform random, given $\mathsf{Out}_L, \mathsf{Out}_R$. Moreover, since we have shown that the view of the adversary in the Hybrid 4 can be fully reconstructed given $\mathsf{Out}_L, \mathsf{Out}_R$, we have that, in the Hybrid 4, the encoded randomness $\langle c_L, c_R \rangle$ is statistically close to uniform, given the adversary's view in the CNMC experiment.

5 Acknowledgments

We thank the anonymous PKC 2019 reviewers for pointing out an error and fix to our lower bound proof. We also thank them for extensive comments that helped to significantly improve our presentation.

[3] Recall that $\mathcal{S}'_L \subseteq \mathcal{S}_L$, and $\mathcal{S}_L$ contains all the values of $\hat{L}'$ which occur with probability at least ϵ. Therefore $|\mathcal{S}_L| \leq 1/\epsilon$ (and thus $|\mathcal{S}'_L| \leq 1/\epsilon$), since otherwise the sum of the probabilities would exceed 1. A similar argument is true for $\mathcal{S}'_R$.