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 continuous way instead of one-time tampering. They also showed that CNMC with information theoretic security cannot be constructed in 2-split-state tampering model, and presented a construction of the same in CRS (common reference string) model using 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. Specifically, 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 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.

1 Introduction

Non-malleable codes (NMC). Non-malleable codes were introduced by Dziembowski, Pietrzak and Wichs [34] as a relaxation of error-correcting codes, and are useful in settings where privacy—but not necessarily correctness—is desired. 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 [25, 24, 44] and theoretical computer science [21].

Continuous Non-malleable codes (CNMC). Importantly, standard non-malleable codes achieve security only against one-time tampering. This means that in applications, the non-malleable encoding of a secret key must be continually decoded and re-encoded, each time the device is run, incurring overhead in computation and in generation of randomness for re-encoding. This motivated a stronger notion of non-malleable codes, known as continuous non-malleable codes (CNMC), introduced by Faust et al. [36]. This definition allows many-time tampering, which means that 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. This means that 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 gets to interact 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 gets to submit 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

* This work is supported in part by an NSF CAREER Award #CNS-1453045, by a Cisco University Research Program Fund award and by financial assistance award 70NANB15H328 from the U.S. Department of Commerce, National Institute of Standards and Technology.

experiment. Otherwise, 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.

Split-state tampering. One of the most well-studied tampering classes for non-malleable codes is known as 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 the related work section for a discussion of prior work on NMC and CNMC against split-state tampering.

Information-theoretic impossibility. The original CNMC paper of [36] 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.

Black-box reductions. In general, it is not feasible to unconditionally rule out the construction of a primitive G from a cryptographic assumption H. This is because the logical statement $H \to G$ is true if H is false and, typically, if $P = NP$ then H will be false. So unconditionally ruling out the construction of primitive G from assumption H 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 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 7 for a formal definition tailored to CNMC). While there are some exceptions [12, 14], 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 CNM Randomness Encoder 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 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.*

Note that, one-to-one commitment schemes in the CRS model can be constructed from any injective one-way function family, allowing us to 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 (see Appendix B).

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, since all 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 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. In our setting, the simulated adversary is stateful and 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 to $\widehat{R}$ or

$\widehat{L}$ such that $\mathsf{D}(L, \widehat{R}) \neq \perp$ or $\mathsf{D}(\widehat{L}, R) \neq \perp$ 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 $\widehat{R}$ or $\widehat{L}$ along with the pair (L', R') stored in the table. The simulated adversary then decodes $(L, \widehat{R})$ or $(\widehat{L}, R)$ 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 guarantees that we can provide a "real" adversary (which is stateless but inefficient) whose input/output behavior is identical to that of the simulated adversary. See section 3.1 for additional details.

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 [48]. 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 Appendix A).

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 [36] 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.[1] Our main idea is very simple: To respond to the i-th tampering query, we simply 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. decode 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).

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}

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

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

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

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 $\widehat{R} \notin S_R$,

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

[1] 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 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.

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

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

is negligible. 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} (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. 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 [32] in their construction of $2t$-split-state continuously non-malleable codes from t-split-state one-way continuously non-malleable codes. See section 5 for additional details.

1.3 Related Work

Non-Malleable Codes. The notion of non-malleable codes (NMC) was formalized in the seminal work of Dziembowski, Pietrzak and Wichs [34]. Split-state classes of tampering functions introduced by Liu and Lysyanskaya [54], have subsequently received a lot of attention with a sequence of improvements achieving reduced number of states, improved rate, or adding desirable features to the scheme [33, 4, 20, 3, 8, 2, 47, 52]. Recently [10, 7, 19, 35, 11, 9] gave efficient constructions of NMC for "non-compartmentalized" tampering function classes. NMC have also been considered in several other models for various practical applications in [29, 18, 16]. Other works on non-malleable codes include [36, 23, 17, 6, 46, 15, 29, 37, 3, 18, 49, 27, 32, 28].

Continuous Non-Malleable Codes Continuous Non-Malleable codes (CNMC) were introduced by Faust et.al. in [36]. They gave a construction of CNMC based on existence of collision resistant hash functions (CRHFs) and non-interactive zero knowledge proof systems (NIZKs) in common reference string (CRS) model. They also showed the impossibility of constructing 2-split state CNMC information theoretically. Subsequent to the original Faust et.al construction, Jafargholi and Wichs [46] presented a general study of CNMCs and its variants with some existential results. Recently, Aggarwal et. al. [5] gave first information theoretic construction in 8-split-state model.

Non-Malleable Randomness Encoders (NMRE) NMRE were introduced recently by Kanukurthi et. al. [48] 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 [22] studied the "capacity" of non-malleable codes in order to understand the optimal bounds on the efficiency of non-malleable codes. This work has been instrumental in asserting the claims of efficient constructions for non-malleable codes since then [2, 7, 8] etc. Similar study was presented in [27] 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 [45]. Their oracle separation technique was subsequently used to rule out black-box reductions between various primitives such as collision resistant hash functions to one way functions [60], oblivious transfer to public key encryption [43] and many more. The meta-reduction technique (cf. [26, 57, 41, 38, 58, 42, 1, 59, 13, 40]) 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

Let $\mathbb{N}$ be the set of all natural numbers, i.e., $\mathbb{N} = \{1, 2, 3, \ldots\}$. For $n \in \mathbb{N}$, we write $[n] = \{1, \ldots, n\}$. For a set S, $x \leftarrow S$ denotes, sampling an element x uniformly at random from the set S. For an algorithm A, $y \leftarrow A(x)$ is the output obtained on execution of A on input x. If $A(\cdot, \cdot)$ is a randomized algorithm, then $y \leftarrow A(x, r)$, is the output random variable for input x and randomness r. We also write, $A(x)$ instead of $A(x, r)$ if it is clear from the context for the brevity. A function $\delta(\cdot)$ is called *negligible* if for all sufficiently large n and for every polynomial $p(\cdot)$, it holds that $\delta(n) < 1/p(n)$. In this paper, we will denote a negligible function by $\mathsf{negl}(\cdot)$.

For a random variable X, we sometimes also denote the corresponding probability distribution by X. An ensemble of probability distributions $\{X_\lambda\}_{\lambda \in \mathbb{N}}$ is a sequence of probability distributions. For two probability ensembles $\{X\}_\lambda$ and $\{Y\}_\lambda$, defined over a domain S with finite support we say that $\{X\}_\lambda$ and $\{Y\}_\lambda$ are *statistically indistinguishable* if there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$,

$$\frac{1}{2} \sum_{s \in S} \left| \Pr\left[X_\lambda = s\right] - \Pr\left[Y_\lambda = s\right] \right| \leq \mathsf{negl}(\lambda)$$

We denote statistical indistinguishability by $X_\lambda \approx_s Y_\lambda$.

Similarly, we say that two probability ensembles $\{X\}_\lambda$ and $\{Y\}_\lambda$, defined over a domain S with finite support we say that $\{X\}_\lambda$ and $\{Y\}_\lambda$ are *computationally indistinguishable* if for all probabilistic polynomial time distinguishers $\mathcal{D}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$,

$$\left| \Pr_{x \leftarrow X_\lambda}[\mathcal{D}(x) = 1] - \Pr_{y \leftarrow Y_\lambda}[\mathcal{D}(y) = 1] \right| \leq \mathsf{negl}(\lambda)$$

We denote computational indistinguishability by $X_\lambda \approx_c Y_\lambda$.

If S is a set, we denote by U_S the uniform distribution over S. For $\lambda \in \mathbb{N}$, we denote by U_λ the uniform distribution over λ-bit strings.

Remark 1. If a distribution $\mathcal{D}$ with support $\mathcal{S}$ of size 2^ℓ is statistically $2^{-\lambda}$-close to the uniform distribution over $\mathcal{S}$, denoted $U_\mathcal{S}$, then for every $x \in \mathcal{S}$, $\left|\Pr_{X \leftarrow \mathcal{D}}[X = x] - 1/2^\ell\right| \leq 1/2^\lambda$. This implies that $1/2^\ell - 1/2^\lambda \leq \Pr_{X \leftarrow \mathcal{D}}[X = x] \leq 1/2^\ell + 1/2^\lambda$.

2.1 Randomness Extractors

Lemma 1 (Inner-Product Two-Source Extractor [61]). *Let $\boldsymbol{X}, \boldsymbol{Y}$ be independent variables, where $\boldsymbol{X}, \boldsymbol{Y}$ have their support in $\{0,1\}^{\ell} = \mathbb{F}_{2^{\lambda}}^{\frac{\ell}{\lambda}}$ and $\lambda|\ell$, Let U_{λ} be uniform and independent on $\mathbb{F}_{2^{\lambda}}$. Then*

$$\Delta(\langle \boldsymbol{X}, \boldsymbol{Y} \rangle, U_{\lambda}) \leq 2^{-s}$$

for some $s \geq 1 + \frac{1}{2}(k_X + k_Y - \ell - \lambda)$, where $k_X := \mathbb{H}_{\infty}(\boldsymbol{X})$, $k_Y := \mathbb{H}_{\infty}(\boldsymbol{Y})$

Next we present the worst-case version:

Lemma 2 (Inner-Product Two-Source Extractor, worst-case version [51]). *Let $\boldsymbol{X}, \boldsymbol{Y}, Z$ be correlated variables, where $\boldsymbol{X}, \boldsymbol{Y}$ have their support in $\{0,1\}^{\ell} = \mathbb{F}_{2^{\lambda}}^{\frac{\ell}{\lambda}}$ and $\lambda|\ell$, and are independent conditioned on Z. Let U_{λ} be uniform and independent on $\mathbb{F}_{2^{\lambda}}$. Then*

$$\Delta((Z, \langle \boldsymbol{X}, \boldsymbol{Y} \rangle), (Z, U_{\lambda})) \leq 2^{-s}$$

for some $s \geq 1 + \frac{1}{2}(k_X + k_Y - \ell - \lambda)$, where $k_X := \widetilde{\mathbb{H}}_{\infty}(\boldsymbol{X}|Z)$, $k_Y := \widetilde{\mathbb{H}}_{\infty}(\boldsymbol{Y}|Z)$

2.2 CNMC

Here, we present the definition of coding scheme, and continuous non-malleable codes.

Definition 1 (Coding Scheme [34]). *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).*

Next, we present the definition of continuous non malleable codes in CRS model for codes with split-state encoding schemes.

Definition 2 (Split-State Encoding Scheme in the CRS model [36]). *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 $\perp$.*

Before defining the continuous non malleable codes 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 [36]). *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*

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

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 [36], 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*

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

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* $\mathcal{O}_{\mathsf{CNM}}$.

2.3 Continuous Non-Malleable Randomness Encoder

The following definition is an adaptation of the notion of Non-Malleable Randomness Encoders [48] 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.*

Next, we present definitions related to falsifiable assumptions and black-box reductions, strong one-time signature schemes, and equivocal commitment scheme.

2.4 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* $0 \le \delta < 1$. *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.*

2.5 (Strong) One-Time Signature Schemes

A digital signature scheme consists of a triple of ppt algorithms $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ such that:

- Gen takes the security parameter 1^λ as input and generates a pair of keys: a public verification key vk, and a secret signing key sk.
- Sign takes as input a secret key sk and a message m, and generates a signature σ. We write this as $\sigma \leftarrow \mathsf{Sign}_{\mathsf{sk}}(m)$.
- Verify takes as input a verification key vk, a message m, and a (purported) signature σ and outputs a single bit indicating acceptance or not.

For correctness, we require that for all $(\mathsf{vk}, \mathsf{sk})$ output by $\mathsf{Gen}(1^\lambda)$, for all messages m, and for all $\sigma \leftarrow \mathsf{Sign}_{\mathsf{sk}}(m)$, we have $\mathsf{Verify}_{\mathsf{vk}}(m, \sigma) = 1$.

2.6 Equivocal Commitment Scheme

We start by defining the basic commitment schemes and then present the notion of *equivocal bit-commitment schemes* introduced by Di Crescenzo et al. in [30].

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:*

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

The commitment scheme must satisfy the standard correctness requirement,

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

where, $\mathcal{CRS}$ *is the set of all possible valid CRS's generated by* $\mathsf{CRSGen}(1^\lambda)$ *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:

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

Binding: It is computationally hard for any adversary $\mathcal{A}$ to find a triple (com, d, d') such that *both* (com, d) and (com, 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:

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

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 (Non-Interactive Equivocable Bit-Commitment Scheme). *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* $(\Sigma', com', d'_0, d'_1)$ *satisfying the following:*

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

We now present variant of the commitment scheme presented by Naor in [55], specifically we present the same construction in CRS model. This is also presented in [30].

Let $\lambda > 0$ be an integer, let $G : \{0,1\}^\lambda \to \{0,1\}^{3\lambda}$ be a pseudo-random generator.

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

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

In order to prove claim 2.1 we need to show an efficient simulator S_{Eq} which outputs $(\Sigma', com', d'_0, d'_1)$ on input 1^λ. Following is the description of S_{Eq}: On input 1^λ, S_{Eq} chooses two uniform random seeds $s_0, s_1 \in \{0,1\}^\lambda$ and computes $u = G(s_0)$ and $v = G(s_1)$. Set $\Sigma' = u \oplus v$, $com' = u$, and for $b \in \{0,1\}$, set $d'_b = s_b$.

Clearly S_{Eq} can open both 0 and 1 by choosing d'_0 or d'_1 respectively. Moreover, for any algorithm distinguishing between real transcript and interaction with S_{Eq} we can present a distinguisher which breaks the security of G with same advantage. This can be achieved by replacing the string v by the challenge string in the pseudo-random generator experiment.

2.7 One-to-one Equivocal Commitment

We 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.

Let $\lambda > 0$ be an integer, let $G_1 : \{0,1\}^{\lambda'} \to \{0,1\}^{3\lambda'}$ and $G_2 : \{0,1\}^{\lambda} \to \{0,1\}^{t \cdot \lambda'}$ be pseudo-random generators.

- $\mathsf{CRSGen}(1^\lambda)$: Run $\mathsf{CRSGen}_\Pi(1^\lambda)$ to generate Σ_Π. Output $\Sigma = \Sigma_\Pi, \Sigma_1, \Sigma_2$ where Σ_1, Σ_2 are uniform random strings of length 3λ.
- $\mathsf{Commit}_\Sigma(b)$: Choose uniform random seeds $s_1, s_2 \in \{0,1\}^\lambda$ 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 $(com_\beta, d_\beta) = \mathsf{Commit}_{\Sigma_\Pi}(s_1||s_2)$ and $(com_{1-\beta}, d_{1-\beta}) = \mathsf{Commit}_{\Sigma_\Pi}(0^{2n})$. Output commitment $com := (c^1, c^2, com_0, com_1)$. Output decommitment information $s := (s_1, s_2, d_\beta)$.
- $\mathsf{Open}_\Sigma(com, s)$: Parse $com = (c^1, c^2, com_0, com_1)$ and $s = s_1||s_2||d$. If $c^2 = G(s_2)$, set $\beta = 0$. If $c^2 = G(s_2) \oplus \Sigma_2$, set $\beta = 1$. Run $\mathsf{Open}_{\Sigma_\Pi}(com_\beta, d)$ 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 unique string decommitment property of Π, the modified scheme has the unique string decommitment property.

To create equivocal commitments/openings one can do the following: Run $\mathsf{CRSGen}_\Pi(1^\lambda)$ to generate Σ_Π. Choose uniform random seeds $s_1^0, s_1^1, s_2^0, s_2^1 \in \{0,1\}^\lambda$ 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 $(com_\beta, d_\beta) = \mathsf{Commit}_{\Sigma_\Pi}(s_1^0||s_2^\beta)$ and $(com_{1-\beta}, d_{1-\beta}) = \mathsf{Commit}_{\Sigma_\Pi}(s_1^1||s_2^{1-\beta})$. 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 commitment $com' := (c^1, c^2, com_0, com_1)$.

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

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

2.8 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 ℓ-bit commitment scheme in the CRS model (given in Section 2.7). Let $(\mathsf{Gen}_\mathsf{Sign}, \mathsf{Sign}, \mathsf{Verify})$ be a strong, one-time signature scheme. 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 Sections 4 and 5.

Key generation Gen_{Com} is as follows: On input security parameter 1^λ, run Gen'_{Com} $2t$ times to generate t pairs of CRS's $[(\Sigma_{Eq}^{0,i}, \Sigma_{Eq}^{1,i})]_{i\in[t]}$, 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 of length ℓ, generate a key pair $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Gen}_\mathsf{Sign}$. For $i \in [t]$, generate $(com_i, d_i) \leftarrow \mathsf{Com}'(\Sigma^{\mathsf{vk}_i, i}, m)$, where com_i is the commitment and d_i is the decommitment information. Generate $\sigma \leftarrow \mathsf{Sign}_\mathsf{sk}([com_i]_{i\in[t]})$. Output commitment $c = (\mathsf{vk}, [com_i]_{i\in[\ell]}, \sigma)$. 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 $d[S] = [d_{i,j}]_{i\in[t], j\in[S]}$, where $d_{i,j}$ is the decommitment information contained in d_i corresponding to the j-th bit.

Decommitment Open w.r.t. a set S: Given a set S, a commitment com, and an opening $[d_{i,j}]_{i\in[t], j\in S}$, Open does the following: Parse commitment as $(\mathsf{vk}, [com_{i,j}]_{i\in[t], j\in[\ell]}, \sigma)$. (1) Check that $\mathsf{Verify}_\mathsf{vk}([com_{i,j}]_{i\in[t], j\in[\ell]}, \sigma) =$

1 (2) For $i \in [t]$, $j \in S$, check that $d_{i,j}$ is a valid decommitment for $com_{i,j}$ w.r.t. CRS $\Sigma^{\mathsf{vk}_i,i}$.

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 times to generate $[\Sigma^{\mathsf{vk}_i,i}]_{i\in[t]}$, equivocal commitments $[com_i]_{i\in[t]}$ and decommitments $[(d^0_{i,j}, d^1_{i,j})]_{i\in[t],j\in[\ell]}$. Run Gen'_{Com} t times to generate $[\Sigma^{1-\mathsf{vk}_i,i}]_{i\in t}$. Set $\Sigma_{Eq} := [(\Sigma^{0,i}_{Eq}, \Sigma^{1,i}_{Eq})]_{i\in[t]}$. Compute $\sigma \leftarrow \mathsf{Sign}_{\mathsf{sk}}([com_i]_{i\in[t]})$. Output $(\Sigma = \Sigma_{Eq}, c = (\mathsf{vk}, [com_i]_{i\in[\ell]}, \sigma),$
$d^0 = [d^0_{i,j}]_{i\in[t],j\in[\ell]}, d^1 = [d^1_{i,j}]_{i\in[t],j\in[\ell]})$.

Additional Check functionality: Given a Σ and commitments $com = (\mathsf{vk}, [com_i]_{i\in[\ell]}, \sigma)$, $com' = (\mathsf{vk}', [com'_i]_{i\in[\ell]}, \sigma')$, $\mathsf{Check}_\Sigma(com, com')$ outputs 1 if (1) $\mathsf{vk} = \mathsf{vk}'$; (2) $\mathsf{Verify}_{\mathsf{vk}}([com'_i]_{i\in[\ell]}, \sigma') = 1$.

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 2.7 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 2.7 as the underlying commitment scheme.

3 Impossibility of CNMC with no CRS

In this section we present theorem 5, stating 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.*

3.1 Proof of Theorem 5 Impossibility of CNMC with no CRS

In this section we present the proof of theorem 5.

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:

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

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:*

– $y \neq z \wedge \mathsf{D}(x, y) \neq \bot \wedge \mathsf{D}(x, z) \neq \bot$ *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 3. *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 4. *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 codeword (L, R) and decodes to get the bit b. If $b = 0$, the real adversary encodes a random bit $b' = RO_1(L||R)$ using randomness $r = RO_2(L||R)$ (where RO_1, 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, the real adversary outputs the left/right side of a random encoding of a random bit, b''. Note that since the CNMC is perfectly unique, the real adversary obtains non-negligible advantage of $1 - \mathsf{negl}(\lambda)$ 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 RO_1, RO_2:

- A_L: On input L, find the unique R such that $\mathsf{D}(L, R) \neq \perp$. Let $b := \mathsf{D}(L, R)$. If $b = 0$, encode $b' = RO_1(L||R)$ using randomness $r = RO_2(L||R)$ to obtain $(L', R') := \mathsf{D}(b'; r)$ and output L'. Otherwise, compute a random encoding of a random bit b'', $(L'', R'') \leftarrow \mathsf{D}(b'')$ and output L''.
- A_R: On input R, find the unique L such that $\mathsf{D}(L, R) \neq \perp$. Let $b := \mathsf{D}(L, R)$. If $b = 0$, encode $b' = RO_1(L||R)$ using randomness $r = RO_2(L||R)$ to obtain $(L', R') := \mathsf{D}(b'; r)$ and output R'. Otherwise, compute a random encoding of a random bit b'', $(L'', R'') \leftarrow \mathsf{D}(b'')$ 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: Keeps two lists $List_L$, $List_R$. Let T be a table that records internal randomness. 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'.
   If not, 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'.
   If not, choose a random encoding of a random bit b'': $(L'', R'') \leftarrow \mathsf{E}(b'')$. Store (L, 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 with non-negligible probability p and since the view of the reduction is identical when interacting with Real or Sim, Reduction interacting with Sim must also succeed with non-negligible probability p. But Reduction composed with Sim are efficient, 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 Appendix A.

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 presented in Figure 1.

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).

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.

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

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 2.8.

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

$\mathsf{E}_\Sigma(c_L||c_R||r_L||r_R)$:

1. Parse c_L, c_R as strings in $\mathbb{F}_{2^\lambda}^{\frac{\ell}{\lambda}}$.
2. $(com, d) \leftarrow \mathsf{Com}_\Sigma(c_L||c_R)$
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. $\mathsf{E}_{1,\Sigma}$ outputs $L = (com, d[S_L])$; $R = (com, d[S_R])$. $\mathsf{E}_{2,\Sigma}$ 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.

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

1. Parse $\widetilde{L} = (\widetilde{com}, \widetilde{d}[S_L])$, $\widetilde{R} = (\widetilde{com}', \widetilde{d}[S_R])$.
2. $\underline{\text{If } L \neq L' \text{ and } \mathsf{Check}_\Sigma(com, \widetilde{com}) = 1 \text{ or } R \neq R' \text{ and } \mathsf{Check}_\Sigma(com, \widetilde{com}') = 1 \text{ then output } \perp.}$
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.

$\mathsf{CRSGen}^2(1^\lambda)$: $\underline{\text{Run } (\Sigma_{Eq}, \overline{com}, d^0, d^1) \leftarrow \mathsf{S}_{Eq}(1^\lambda). \text{ Output } \Sigma_{Eq}.}$
Challenge codeword:

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

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

– $\mathcal{S}_L$ 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 $\mathsf{D}^1_{\Sigma_{Eq}}(\widehat{L}', \widehat{R}') \neq \perp$.
– 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 $\mathsf{D}^1_{\Sigma_{Eq}}(\widehat{L}', \widehat{R}') \neq \perp$.

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

1. $\underline{\text{Check that } \widetilde{L} \in \mathcal{S}'_L \text{ and that } \widetilde{R} \in \mathcal{S}'_R\text{. If not, output } \perp.}$
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 $\widetilde{c}_L \neq \perp$ 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. 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.

$\mathsf{D}^4_{\Sigma_{Eq}}((f_L^i, f_R^i), \widetilde{L}, \widetilde{R})$:

1. Check that $\widetilde{L} \in \mathcal{S}'_L$ and that $\widetilde{R} \in \mathcal{S}'_R$. If not, output $\perp$.
2. $\underline{\text{If } \widetilde{L} \neq L^*\text{, then set } \mathsf{count}_L := \mathsf{count}_L + 1.}$
3. $\underline{\text{If } \widetilde{R} \neq R^*\text{, then set } \mathsf{count}_R := \mathsf{count}_R + 1.}$
4. $\underline{\text{If } \mathsf{count}_L > y_t \text{ or } \mathsf{count}_R > y_t\text{, output } \perp.}$
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 2.8.

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 $i \in [q]$, $\Pr[f_L^i(L) \notin \mathcal{S}'_L \wedge \mathsf{D}^1_{\Sigma_{Eq}}(f_L^i(L), f_R^i(R)) \neq \bot] \leq \epsilon$ and $\Pr[f_L^i(R) \notin \mathcal{S}'_R \wedge \mathsf{D}^1_{\Sigma_{Eq}}(f_L^i(L), f_R^i(R)) \neq \bot] \leq \epsilon$. 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 $\mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1}$ (which does not contain a $\bot$ output) and each value of $R = \widehat{R}$,

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

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

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

We first fix $(\Sigma_{Eq}, \overline{com}, d^0, d^1)$. Note that for fixed $\Sigma_{Eq}, \overline{com}, d^0, d^1$, 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. $f_R^i(R)$) returns the unique L' (resp. R') if it exists such that, $\mathsf{D}^1_{\Sigma_{Eq}}(L', f_R^i(R)) \neq \bot$ (resp. $\mathsf{D}^1_{\Sigma_{Eq}}(f_L^i(L), R') \neq \bot$). Note that L' (resp. R') is equal to "same" if and only if $f_R^i(R) =$ "same" (resp. $f_L^i(L) =$ "same").

We first show that for $i \in \{0, \ldots, q\}$, c_L, c_R are conditionally independent given $\mathsf{Out}_A^i = \widehat{Out}_A^{i-1}$. This follows from the fact that the information contained in $\widehat{Out}_A^{i-1}$ is of the form $(f_L^1(\phi_L(c_L)) = v_1, f_R^1(\phi_R(c_R)) = w_1), \ldots, (f_L^{i-1}(\phi_L(c_L)) = v_i, f_R^{i-1}(\phi_R(c_R)) = w_i)$, where for $j \in [i-1]$, 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 $(f_L^1(\phi_L(c_L)) = v_1, f_R^1(\phi_R(c_R)) = w_1), \ldots, (f_L^{i-1}(\phi_L(c_L)) = v_i, f_R^{i-1}(\phi_R(c_R)) = w_i)$ is equal to $(U_\ell \mid (f_L^1(\phi_L(U_\ell)) = v_1, \ldots, f_L^{i-1}(\phi_L(U_\ell)) = v_i)) \times (U_\ell \mid (f_R^1(\phi_R(U_\ell)) = w_1, \ldots, f_R^{i-1}(\phi_R(U_\ell)) = w_i))$. 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,

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

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 $\mathsf{D}^1_{\Sigma_{Eq}}(f_L^i(L), f_R^i(R)) \neq \bot$ or event (2) $f_R^i(R)$ is not most frequent and $\mathsf{D}^1_{\Sigma_{Eq}}(f_L^i(L), f_R^i(R)) \neq \bot$ occurs more than y_t times in a single execution is at most $(1/2)^{y_t}$.

We first analyze the event (1). If $f_L^i(L) = L'$ is not the most frequent query in $\mathcal{S}'_L$ then, by definition, $\Pr[f_L^i(\widehat{L}) = L' \mid \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1}] \leq 1/2$. 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 $\Pr[R = \widehat{R} \wedge \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1}] > 0$),

$$\begin{aligned}
&\Pr[f_L^i(L) \text{ is not most frequent } \wedge \mathsf{D}^1_{\Sigma_{Eq}}(f_L^i(L), f_R^i(R)) \neq \bot \mid R = \widehat{R} \wedge \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1})] \\
&\quad = \Pr[f_L^i(L) \text{ is not most frequent } \wedge f_L^i(L) = \rho_L(f_R^i(R)) \mid R = \widehat{R} \wedge \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1})] \\
&\quad \leq 1/2.
\end{aligned}$$

The probability that this event occurs y_t times for y_t distinct values of $i \in [q]$ is at most $(1/2)^{y_t} \in \mathsf{negl}(\lambda)$. 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 $\mathsf{D}^1_{\Sigma_{Eq}}(L', \widehat{R}') \neq \bot$. 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 $\mathsf{Out}_L := \mathsf{Out}_L || (i || \bot)$.
3. If $L' \notin \mathcal{S}'_L$, output $\bot$ to the adversary and terminate with output $\mathsf{Out}_L := \mathsf{Out}_L || (i || \bot)$.

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 $\Sigma_{Eq}, \overline{com}, d^0, d^1$, universal hash $h \leftarrow \mathcal{H}$ and adversary A. Set output Out_A to “” and Out_L to “”. Repeat the following in rounds $i = 1, 2, \ldots$:

1. Obtain the next tampering function (f_L, f_R) from adversary A given its current view, Out_A.
2. If $(i, \bot) \in \mathsf{Out}_L$ or $(i, \bot) \in \mathsf{Out}_R$, set $\mathsf{Out}_A = \mathsf{Out}_A || \bot$ and abort.
3. If $(i, y) \in \mathsf{Out}_L$, for some $y \neq \bot$, 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 $y \neq \bot$, 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 $\mathsf{Out}_A = \mathsf{Out}_A || \bot$ and abort.
9. Else Parse $L' := (com, d[S_L])$ and $R' := (com', d[S_R])$. If $com \neq com'$, set $\mathsf{Out}_A = \mathsf{Out}_A || \bot$ 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.

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 given in Lemma 1 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 4-State CNMC for Multi-Bit Messages

In this section we prove the following theorem:

Theorem 7. *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.*

Notation and parameters. λ is security parameter and length of encoded message. $\ell = \ell(\lambda) \in \Theta(\lambda^2)$ and we assume for simplicity that $\lambda | \ell$. $k = 2\lambda$. Sets $S_{L,1}, S_{R,1}, S_{L,2}, S_{R,2} \subseteq [4\ell]$ are defined as follows: $S_{L,1} = [\ell], S_{R,1} = [2\ell] \setminus [\ell]$, $S_{L,2} = [3\ell] \setminus [2\ell], S_{R,2} = [4\ell] \setminus [3\ell]$. $y_o = y_o(\ell) \in \Theta(\ell^{1/2})$, $y_t = y_t(\ell) \in \Theta(\ell^{1/2})$.

The construction of the multi-bit, 4-state CNMC is presented in Figure 6.

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 2.8.

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

$\mathsf{E}'_\Sigma(m)$ for $m \in \{0,1\}^\lambda$:

1. Choose at random $m^1, m^2 \in \mathbb{F}_{2^\lambda}$ such that $m^1 + m^2 = m$.
2. For $b \in \{1,2\}$, Choose $(c_{L,b}, c_{R,b})$ at random, from $\mathbb{F}_{2^\lambda}^{\frac{\ell}{\lambda}}$, conditioned on $\langle c_{L,b}, c_{R,b} \rangle = m^b$.
3. $(com, d) \leftarrow \mathsf{Com}_\Sigma([c_{L,b} || c_{R,b}]_{b \in \{1,2\}})$
4. For $b \in \{1,2\}$, let $d[S_{L,b}]$ (resp. $d[S_{R,b}]$) correspond to the decommitment of com to the bits corresponding to $S_{L,b}$ (resp. $S_{R,b}$).
5. Output $L_1 = (com, d[S_{L,1}])$; $R_1 = (com, d[S_{R,1}])$; $L_2 = (com, d[S_{L,2}])$; $R_2 = (com, d[S_{R,2}])$.

$\mathsf{D}'_\Sigma(\widetilde{L}_1, \widetilde{R}_1, \widetilde{L}_2, \widetilde{R}_2)$: //For simplicity of notation, we assume D' can take its inputs in any order.

1. For $b \in \{1,2\}$, parse $\widetilde{L}_b = (\widetilde{com}_b, d[S_{L,b}])$, $\widetilde{R}_b = (\widetilde{com}'_b, d[S_{R,b}])$.
2. Check that $\widetilde{com}_1 = \widetilde{com}'_1 = \widetilde{com}_2 = \widetilde{com}'_2$.
3. For $b \in \{1,2\}$, let $\widetilde{c}_{L,b} = \mathsf{Open}_\Sigma(S_{L,b}, \widetilde{com}, \widetilde{d}[S_{L,b}])$ and $\widetilde{c}_{R,b} = \mathsf{Open}_\Sigma(S_{R,b}, \widetilde{com}, \widetilde{d}[S_{R,b}])$. Check that $\widetilde{c}_{L,b} \neq \perp$ and $\widetilde{c}_{R,b} \neq \perp$.
4. For $b \in \{1,2\}$, compute $\widetilde{m}_b = \langle c_{L,b}, c_{R,b} \rangle$.
5. If all the above checks pass, output $\widetilde{m}_1 + \widetilde{m}_2$; otherwise, output $\perp$.

Fig. 6. Construction of 4-state Continuous, Non-Malleable Code.

To prove Theorem 7, we show that the construction above is a secure multi-bit CNMC, via the following sequence of hybrids.

Hybrid 0: This is the Experiment from Definition 4.

Hybrids 1 and 2 are analogous to the first and second hybrids in the previous section. We therefore give an abbreviated description.

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 $(\widetilde{L}_1, \widetilde{R}_1, \widetilde{L}_2, \widetilde{R}_2)$ is different from the challenge codeword (L_1, R_1, L_2, R_2) but the corresponding commitment is not statistically binding.

Hybrid 2: The experiment is identical to Hybrid 1, except we switch to equivocal commitments in (L_1, R_1) (resp. (L_2, R_2)) that is given to the adversary. We denote the corresponding CRS's, and equivocal commitment and decommitments by $\Sigma_{Eq}, \overline{com}, d^0, d^1$.

We now define some terminology which will be needed for the next sequence of hybrids. Given $(\Sigma_{Eq}, \overline{com}, d^0, d^1)$ an output $(\mathsf{Out}^i_{A,1}, \mathsf{Out}^i_{A,2})$, for $b \in \{1,2\}$, we define the sets $\mathcal{S}_{L,b}$, $\mathcal{S}_{R,b}$, $\mathcal{S}'_{L,b}$, $\mathcal{S}'_{R,b}$ as follows:

– $\mathcal{S}_{L,b}$ contains all values of $\widehat{L}'_b$ that occur with probability at least $\epsilon = 1/2^{y_\circ/3}$, where values of $\widehat{L}'_b$ are sampled as follows: Sample $\widehat{c}_{L,b}$ conditioned on $\mathsf{Out}^i_{A,b}$. Compute equivocal decommitment of $\overline{com}$: $\widehat{d}[S_{L,b}] := [d_i^{\widehat{c}_{L,b}[i]}]_{i \in S_{L,b}}$. Apply $f^i_{L,b}$ to $\widehat{L}_b = (\overline{com}_b, \widehat{d}[S_{L,b}])$ to obtain $\widehat{L}'_b$ (or "same" if the output is $\widehat{L}_b$ itself).

– $\mathcal{S}_{R,b}$ contains all values of $\widehat{R}'_b$ that occur with probability at least $\epsilon = 1/2^{y_\circ/3}$, where values of $\widehat{R}'_b$ are sampled as follows: Sample $\widehat{c}_{R,b}$ conditioned on $\mathsf{Out}^i_{A,b}$. Compute equivocal decommitment of $\overline{com}$: $\widehat{d}[S_{R,b}] := [d_i^{\widehat{c}_{R,b}[i]}]_{i \in S_{R,b}}$. Apply $f^i_{R,b}$ to $\widehat{R}_b = (\overline{com}_b, \widehat{open}_{R,b})$ to obtain $\widehat{R}'_b$ (or "same" if the output is $\widehat{R}_b$ itself).

– Let $\mathcal{S}'_{L,b} \subseteq \mathcal{S}_{L,b}$ be the set of $\widehat{L}'_b$ such that there is a "matching" $\widehat{R}'_b \in \mathcal{S}_{R,b}$ such that $\mathsf{D'}^1_{\Sigma_{Eq}}(\widehat{L}'_b, \widehat{R}'_b, \cdot, \cdot) \neq \perp$.

– Let $\mathcal{S}'_{R,b} \subseteq \mathcal{S}_{R,b}$ be the set of $\widehat{R}'_b$ such that there is a "matching" $\widehat{L}'_b \in \mathcal{S}_{L,b}$ such that $\mathsf{D'}^1_{\Sigma_{Eq}}(\widehat{L}'_b, \widehat{R}'_b, \cdot, \cdot) \neq \perp$.

We also give two alternate decoding procedures in Figures 7 and 8.

$\mathsf{D'}^3_{\Sigma_{Eq}}([(f^i_{L,b}, f^i_{R,b}), \widetilde{L}_b, \widetilde{R}_b]_{b \in \{1,2\}})$:

1. <u>For $b \in \{1,2\}$, check that $\widetilde{L}_b \in \mathcal{S}'_{L,b}$ and that $\widetilde{R}_b \in \mathcal{S}'_{R,b}$. If not, output $\perp$.</u>
2. For $b \in \{1,2\}$, parse $\widetilde{L}_b = (\widetilde{com}_b, d[S_{L,b}])$, $\widetilde{R}_b = (\widetilde{com}'_b, d[S_{R,b}])$.
3. Check that $\widetilde{com}_1 = \widetilde{com}'_1 = \widetilde{com}_2 = \widetilde{com}'_2$.
4. For $b \in \{1,2\}$, let $\widetilde{c}_{L,b} = \mathsf{Open}_\Sigma(S_{L,b}, \widetilde{com}, \widetilde{d}[S_{L,b}])$ and $\widetilde{c}_{R,b} = \mathsf{Open}_\Sigma(S_{R,b}, \widetilde{com}, \widetilde{d}[S_{R,b}])$. Check that $\widetilde{c}_{L,b} \neq \perp$ and $\widetilde{c}_{R,b} \neq \perp$.
5. For $b \in \{1,2\}$, compute $\widetilde{m}_b = \langle \widetilde{c}_{L,b}, \widetilde{c}_{R,b} \rangle$.
6. If all the above checks pass, output $\widetilde{m}_1 + \widetilde{m}_2$; otherwise, output $\perp$.

Fig. 7. Algorithm $\mathsf{D'}^3_{\Sigma_{Eq}}$.

For the following decode algorithm, we assume that at the beginning of the experiment, for $b \in \{1,2\}$, counters $\mathsf{count}_{L,b}, \mathsf{count}_{R,b}$ are initialized to 0. We also define L^*_b (resp. R^*_b) to be the element of $\mathcal{S}'_{L,b}$ (resp. $\mathcal{S}'_{R,b}$) that occurs most frequently.

$\mathsf{D'}^4_{\Sigma_{Eq}}([(f^i_{L,b}, f^i_{R,b}), \widetilde{L}_b, \widetilde{R}_b]_{b \in \{1,2\}})$:

1. Check that $\widetilde{L} \in \mathcal{S}'_L$ and that $\widetilde{R} \in \mathcal{S}'_R$. If not, output $\perp$.
2. <u>For $b \in \{1,2\}$, if $\widetilde{L}_b \neq L^*_b$, then set $\mathsf{count}_{L,b} := \mathsf{count}_{L,b} + 1$.</u>
3. <u>For $b \in \{1,2\}$, if $\widetilde{R}_b \neq R^*_b$, then set $\mathsf{count}_{R,b} := \mathsf{count}_{R,b} + 1$.</u>
4. <u>For $b \in \{1,2\}$, if $\mathsf{count}_{L,b} > y_t$ or $\mathsf{count}_{R,b} > y_t$, output $\perp$.</u>
5. For $b \in \{1,2\}$, parse $\widetilde{L}_b = (\widetilde{com}_b, d[S_{L,b}])$, $\widetilde{R}_b = (\widetilde{com}'_b, d[S_{R,b}])$.
6. Check that $\widetilde{com}_1 = \widetilde{com}'_1 = \widetilde{com}_2 = \widetilde{com}'_2$.
7. For $b \in \{1,2\}$, let $\widetilde{c}_{L,b} = \mathsf{Open}_\Sigma(S_{L,b}, \widetilde{com}, \widetilde{d}[S_{L,b}])$ and $\widetilde{c}_{R,b} = \mathsf{Open}_\Sigma(S_{R,b}, \widetilde{com}, \widetilde{d}[S_{R,b}])$. Check that $\widetilde{c}_{L,b} \neq \perp$ and $\widetilde{c}_{R,b} \neq \perp$.
8. For $b \in \{1,2\}$, compute $\widetilde{m}_b = \langle \widetilde{c}_{L,b}, \widetilde{c}_{R,b} \rangle$.
9. If all the above checks pass, output $\widetilde{m}_1 + \widetilde{m}_2$; otherwise, output $\perp$.

Fig. 8. Algorithm $\mathsf{D'}^4_{\Sigma_{Eq}}$.

We next present a sequence of intermediate hybrids $H^2 = H^{2,0,b}, H^{2,1,a} \ldots, H^{2,q,b} = H^3$, defined as follows:

Hybrid $H^{2,i,a}$ for $i \in [q]$: The experiment is identical to the previous hybrid, except we respond to the i-th query to the decoding oracle using $\mathsf{D}^3_{\Sigma_{Eq}}$.

Hybrid $H^{2,i,b}$ for $i \in [q]$: The experiment is identical to the previous hybrid, except we respond to the i-th query to the decoding oracle using $\mathsf{D}^4_{\Sigma_{Eq}}$.

Claim 5.1. Hybrids 0 and 1 are computationally indistinguishable.

This follows from the security of the one-time signature scheme and "uniqueness of opening" property of the underlying commitment.

Claim 5.2. Hybrids 1 and 2 are computationally indistinguishable.

This follows from the security of the equivocal, non-malleable commitment scheme.

We say that an output pair $(\mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1}, \mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2})$ is *good* if the marginal distribution over m^1 is statistically 2^{-k}-close to uniform random conditioned on $(\mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1}, \mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2})$ and the marginal distribution over m^2 is statistically 2^{-k}-close to uniform random conditioned on $(\mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1}, \mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2})$.

Claim 5.3. For $i \in \{0, \ldots, q\}$, in Hybrid $H^{2,i,b}$, the outcome $(\mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1}, \mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2})$ is *good* with probability $1 - 2^{-k}/q$.

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

Leakage function on $c_{L,b}$, for $b \in \{1, 2\}$: 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 $L_b = (\overline{com}, [d_i^{c_{L,b}[i]}]_{i\in[\ell]})$. Repeat the following in rounds $i = 1, 2, \ldots$:

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

The leakage function for $c_{R,b}$ is analogous.

We now show that given $\mathsf{Out}_{L,1}$, $\mathsf{Out}_{R,1}$, $\mathsf{Out}_{L,2}$, $\mathsf{Out}_{R,2}$ we can reconstruct the full output sequence for the adversary's view with probability $1 - \frac{4q}{\epsilon^2 \cdot 2^{y_o}} = 1 - \frac{4q}{2^{y_0/3}}$ in the following way:

Fix $\Sigma_{Eq}, \tau, \overline{com}$, universal hash $h \leftarrow \mathcal{H}$ and adversary A. Set output $\mathsf{Out}_{A,1}$ to "" and $\mathsf{Out}_{A,2}$ to "". Repeat the following in rounds $i = 1, 2, \ldots$:

1. Obtain the next tampering function (f_L, f_R) from adversary A given its current view, $\mathsf{Out}_A = (\mathsf{Out}_{A,1}, \mathsf{Out}_{A,2})$.
2. If for $b \in \{1, 2\}$, $(i, \perp) \in \mathsf{Out}_{L,b}$ or $(i, \perp) \in \mathsf{Out}_{R,b}$, then for $b \in \{1, 2\}$, set $\mathsf{Out}_{A,b} = \mathsf{Out}_{A,b}||(i, \perp)$ and abort.
3. If for $b \in \{1, 2\}$, $(i, y) \in \mathsf{Out}_{L,b}$, for some $y \neq \perp$, set $L'_b = \widehat{L}'_b$ such that $\widehat{L}'_b \in \mathcal{S}'_{L,b}$ and $h(\widehat{L}'_b) = y$.
4. If for $b \in \{1, 2\}$, $(i, \cdot) \notin \mathsf{Out}_{L,b}$, set $L'_b = \widehat{L}'_b$ such that $\widehat{L}'_b \in \mathcal{S}'_{L,b}$ is the most frequent value.
5. If for $b \in \{1, 2\}$, $(i, y) \in \mathsf{Out}_{R,b}$, for some $y \neq \perp$, set $R'_b = \widehat{R}'_b$ such that $\widehat{R}'_b \in \mathcal{S}'_{R,b}$ and $h(\widehat{R}'_b) = y$.
6. If for $b \in \{1, 2\}$, $(i, \cdot) \notin \mathsf{Out}_{R,b}$, set $R' = \widehat{R}'_b$ such that $\widehat{R}'_b \in \mathcal{S}'_{R,b}$ is the most frequent value.
7. If for all $b \in \{1, 2\}$, L'_b = "same" and R'_b = "same" output "same" and for $b \in \{1, 2\}$, set $\mathsf{Out}_{A,b} = \mathsf{Out}_{A,b}||$"same".
8. Else if at least one of $[L'_b, R']_{b\in\{1,2\}}$ is "same" but not all, then for $b \in \{1, 2\}$, set $\mathsf{Out}_{A,b} = \mathsf{Out}_{A,b}||\perp$ and abort.

9. Else for $b \in \{1,2\}$, parse $L'_b := (com_b, open_{L,b})$ and $R'_b := (com'_b, open_{R_b})$. Check that $com_1 = com'_1 = com_2 = com'_2$. If not, set $\mathsf{Out}_{A,b} = \mathsf{Out}_{A,b}||\bot$ and abort.
10. Otherwise, for $b \in \{1,2\}$ set $\mathsf{Out}_{A,b} = \mathsf{Out}_{A,b}||(L'_b, R'_b)$.

It can be determined by inspection that the incorrect value is output only if in one of the at most q instances, for some $b \in \{1,2\}$, there are two distinct values $\widehat{L}'_b, \widehat{L}''_b \in \mathcal{S}'_{L,b}$ or $\widehat{R}'_b, \widehat{R}''_b \in \mathcal{S}'_{R_b}$ such that $h(\widehat{L}'_b) = h(\widehat{L}''_b)$ or $h(\widehat{R}'_b) = h(\widehat{R}''_b)$. Due to universality of h and the fact that for $b \in \{1,2\}$, $|\mathcal{S}'_{L,b}| = |\mathcal{S}'_{R,b}| \leq 1/\epsilon$, this can occur with probability at most $\frac{4q}{\epsilon^2 \cdot 2^{y_o}}$, as claimed.

Since for $b \in \{1,2\}$, $|\mathsf{Out}_{L,b}| \leq (\log(q)+y_o)\cdot y_t \leq 2y_o \cdot y_t \leq c\cdot\ell$, and $|\mathsf{Out}_{R,b}| \leq (\log(q)+y_o)\cdot y_t \leq 2y_o\cdot y_t \leq c\cdot\ell$, for constant $c < 1$, we have that with probability $1 - 2^{-k}/q$ over choice of $(\mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1}, \mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2})$, the min-entropy of $c_{L,b}$ (resp. $c_{R,b}$) conditioned on $(\mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1}, \mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2})$ is at least $c'\cdot\ell$ for constant $c' < 1$. We can use the properties of the inner product extractor given in Lemma 2 to argue that $\langle c_{L,b}, c_{R,b}\rangle$ is statistically close to uniform random, given $\mathsf{Out}_{L,1}$, $\mathsf{Out}_{R,1}$, $\mathsf{Out}_{L,2}$, $\mathsf{Out}_{R,2}$. Moreover, since we have shown that the view of the adversary $Out^i_{A,1}, Out^i_{A,2}$ can be fully reconstructed given $\mathsf{Out}_L, \mathsf{Out}_R$, we have that $\langle c_{L,b}, c_{R,b}\rangle$ is statistically close to uniform, given the adversary's view in the CNMC experiment.

Claim 5.4. For $i \in \{0,\dots,q-1\}$, Hybrids $H^{2,i,b}$ and $H^{2,i+1,a}$ are $4(\epsilon' + 2^{-k})$-close, where $\epsilon' = (1+2^{-\lambda})\epsilon$.

Proof. To prove indistinguishability of Hybrids 2 and 3, it is sufficient to show that for and $b \in \{1,2\}$, $\Pr[f^i_{L,b}(L_b) \notin \mathcal{S}'_{L,b} \wedge \mathsf{D}'^1_{\Sigma_{Eq}}(f^i_{L,b}(L_b), f^i_{R,b}(R_b), \cdot,\cdot) \neq \bot] \leq \epsilon'$ and $\Pr[f^i_{R,b}(R_b) \notin \mathcal{S}'_{R,b} \wedge \mathsf{D}'^1_{\Sigma_{Eq}}(f^i_{L,b}(L), f^i_{R,b}(R_b), \cdot,\cdot) \neq \bot] \leq \epsilon'$. The result then follows by a union bound.

Given Claim 5.3, to bound the above it is sufficient to show: (1) for $b \in \{1,2\}$, each *good* pair $\mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1}$ (which does not contain $\bot$), $\mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2}$ (which does not contain $\bot$), and each value of $R_b = \widehat{R}_b$,

$$\Pr[f^i_{L,b}(L_b) \notin \mathcal{S}'_{L,b} \wedge \mathsf{D}'^1_{\Sigma_{Eq}}(f^i_{L,b}(L_b), f^i_{R,b}(R_b), \cdot,\cdot) \neq \bot \mid R_b = \widehat{R}_b \wedge \mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1} \wedge \mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2}] \leq \epsilon';$$

and (2) for $b \in \{1,2\}$, each *good* pair $\mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1}$ (which does not contain $\bot$), $\mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2}$ (which does not contain $\bot$), and each value of $L_b = \widehat{L}_b$,

$$\Pr[f^i_{R,b}(R_b) \notin \mathcal{S}'_{R,b} \wedge \mathsf{D}'^1_{\Sigma_{Eq}}(f^i_{L,b}(L_b), f^i_{R,b}(R_b), \cdot,\cdot) \neq \bot \mid L_b = \widehat{L}_b \wedge \mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1} \wedge \mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2}] \leq \epsilon'.$$

We first fix $(\Sigma_{Eq}, \overline{com}, d^0, d^1)$. Note that fixed $\Sigma_{Eq}, \overline{com}, d^0, d^1$ and $b \in \{1,2\}$, there is a bijection $\phi_{L,b}$ (resp. $\phi_{R,b}$) between $c_{L,b}$ (resp. $c_{R,b}$) and $(\overline{com}, d[S_{L,b}])$, where $d[S_{L,b}] = [d_i^{c_{L,b}[i]}]_{i\in[\ell]}$. Therefore the probability of a particular value of $c_{L,b}$ (resp. $c_{R,b}$) occurring is equivalent to the probability of $L_b = \phi_{L,b}(c_{L,b})$ (resp. $R_b = \phi_{R,b}(c_{R,b})$). Additionally, Let $\rho_{L,b}$ (resp. $\rho_{R,b}$) be the function that given $f^i_{R,b}(R_b)$ (resp. $f^i_{L,b}(L_b)$) returns the unique L'_b (resp. R'_b) if it exists such that, $\mathsf{D}'^1_{\Sigma_{Eq}}(L'_b, f^i_{R,b}(R_b), \cdot,\cdot) \neq \bot$ (resp. $\mathsf{D}'^1_{\Sigma_{Eq}}(f^i_{L,b}(L_b), R'_b, \cdot,\cdot) \neq \bot$). Note that L'_b (resp. R'_b) is equal to "same" if and only if $f^i_{R,b}(R_b) =$ "same" (resp. $f^i_{L,b}(L_b) =$ "same").

Now, note that for $b = 1$ and every $(\widehat{c}_{L,1}, \widehat{c}_{R,1})$ and every *good* pair $\widehat{Out}_{A,1}, \widehat{Out}_{A,2})$:

$$
\begin{aligned}
&\Pr[c_{L,1} = \widehat{c}_{L,1} \mid c_{R,1} = \widehat{c}_{R,1} \wedge \mathsf{Out}_{A,1}^{i-1} = \widehat{Out}_{A,1}, \mathsf{Out}_{A,2}^{i-1} = \widehat{Out}_{A,2}] \\
&\quad = \sum_{\widehat{c}_{L,2}, \widehat{c}_{R,2}} \Big(\Pr[c_{L,2} = \widehat{c}_{L,2}, c_{R,2} = \widehat{c}_{R,2} \mid \mathsf{Out}_{A,2}^{i-1} = \widehat{Out}_{A,2}] \\
&\qquad\qquad \cdot \Pr[c_{L,1} = \widehat{c}_{L,1}, c_{R,1} = \widehat{c}_{R,1} \mid \mathsf{Out}_{A,1}^{i-1} = \widehat{Out}_{A,1} \wedge \langle c_{L,2}, c_{R,2} \rangle = \langle \widehat{c}_{L,2}, \widehat{c}_{R,2} \rangle] \Big) \\
&\quad = \sum_{m^2} \Pr[\langle c_{L,2}, c_{R,2} \rangle = m^2 \mid \mathsf{Out}_{A,2}^{i-1} = \widehat{Out}_{A,2}] \cdot \Pr[c_{L,1} = \widehat{c}_{L,1}, c_{R,1} = \widehat{c}_{R,1} \mid \mathsf{Out}_{A,1}^{i-1} = \widehat{Out}_{A,1} \wedge m^2] \\
&\quad \in \sum_{\widehat{m}^2} (2^{-\lambda} \pm 2^{-k}) \cdot \Pr[c_{L,1} = \widehat{c}_{L,1} \mid c_{R,1} = \widehat{c}_{R,1} \wedge \mathsf{Out}_{A,1}^{i-1} = \widehat{Out}_{A,1} \wedge m^2 = \widehat{m}^2] \qquad (3) \\
&\quad = (2^{-\lambda} \pm 2^{-k}) \sum_{m^2} \Pr[c_{L,1} = \widehat{c}_{L,1} \mid c_{R,1} = \widehat{c}_{R,1} \wedge \mathsf{Out}_{A,1}^{i-1} = \widehat{Out}_{A,1} \wedge m^2 = \widehat{m}^2] \\
&\quad = (1 \pm 2^{\lambda-k}) \cdot \Pr[c_{L,1} = \widehat{c}_{L,1} \mid \wedge \mathsf{Out}_{A,1}^{i-1} = \widehat{Out}_{A,1}] \\
&\quad = (1 \pm 2^{-\lambda}) \cdot \Pr[c_{L,1} = \widehat{c}_{L,1} \mid c_{R,1} \wedge \mathsf{Out}_{A,1}^{i-1} = \widehat{Out}_{A,1}],
\end{aligned}
$$

where (3) follows from Claim 5.3. An analogous statement holds for $b = 2$.

Morever, by the same reasoning as in the proof of Claim 4.3 (where we showed conditional independence of $c_{L,b}, c_{R,b}$) we have that for every $(\widehat{c}_{L,b}, \widehat{c}_{R,b}, \widehat{Out}_{A,b})$:

$$\Pr[c_{L,b} = \widehat{c}_{L,b} \mid c_{R,b} \wedge \mathsf{Out}_{A,b}^{i-1} = \widehat{Out}_{A,b}] = \Pr[c_{L,b} = \widehat{c}_{L,b} \mid \mathsf{Out}_{A,b}^{i-1} = \widehat{Out}_{A,b}].$$

Therefore, for every every $(\widehat{c}_{L,1}, \widehat{c}_{R,1})$ and every *good* pair $\widehat{Out}_{A,1}, \widehat{Out}_{A,2})$:

$$\Pr[L_b = \widehat{L}_b \mid R_b = \widehat{R}_b \wedge \mathsf{Out}_{A,1}^{i-1} = \widehat{Out}_{A,1}^{i-1} \wedge \mathsf{Out}_{A,2}^{i-1} = \widehat{Out}_{A,2}^{i-1}] \in (1 \pm 2^{-\lambda}) \Pr[L_b = \widehat{L}_b \mid \mathsf{Out}_{A,b}^{i-1} = \widehat{Out}_{A,b}^{i-1}].$$

So for every $R_b = \widehat{R}_b$, every *good* pair $\widehat{Out}_{A,1}, \widehat{Out}_{A,2})$ and every $L'_b \notin \mathcal{S}'_{L,b}$:

$$\Pr[f_{L,b}^i(L_b) = L'_b \mid R_b = \widehat{R}_b \wedge \mathsf{Out}_{A,1}^{i-1} = \widehat{Out}_{A,1}^{i-1} \wedge \mathsf{Out}_{A,2}^{i-1} = \widehat{Out}_{A,2}^{i-1}] \leq (1 + 2^{\lambda})\epsilon \leq \epsilon'.$$

Therefore,

$$
\begin{aligned}
&\Pr[f_{L,b}^i(L_b) \notin \mathcal{S}'_{L,b} \wedge \mathsf{D'}_{\Sigma_{Eq}}^1(f_{L,b}^i(L_b), f_{R,b}^i(R_b), \cdot, \cdot) \neq \perp \mid R_b = \widehat{R}_b \wedge \mathsf{Out}_{A,1}^{i-1} = \widehat{Out}_{A,1}^{i-1} \wedge \mathsf{Out}_{A,2}^{i-1} = \widehat{Out}_{A,2}^{i-1}] \\
&\quad = \Pr[f_{L,b}^i(L_b) \notin \mathcal{S}'_{L,b} \wedge f_{L,b}^i(L_b) = \rho_L(f_{R,b}^i(R_b)) \mid R_b = \widehat{R}_b \wedge \mathsf{Out}_{A,1}^{i-1} = \widehat{Out}_{A,1}^{i-1} \wedge \mathsf{Out}_{A,2}^{i-1} = \widehat{Out}_{A,2}^{i-1}] \\
&\quad \leq \epsilon'.
\end{aligned}
$$

The proof for (2) is analogous.

Claim 5.5. For $i \in \{1, \ldots, q\}$, Hybrids $H^{2,i,a}$ and $H^{2,i,b}$ are $4(\epsilon' + 2^{-k})$-close, where $\epsilon' = 2 \cdot \epsilon$.

Proof. To prove indistinguishability, we must show that for $i \in \{1, \ldots, q\}$, $b \in \{0, 1\}$ the probability that the event (1) $f_{L,b}^i(L_b)$ is not most frequent and $\mathsf{D'}_{\Sigma_{Eq}}^1(f_{L,b}^i(L_b), f_{R,b}^i(R_b), \cdot, \cdot) \neq \perp$ occurs more than y_t times in $H^{2,i,a}$ is at most $((1 + 2^{-\lambda})/2)^{y_t} + 2^{-k}$ and the probability that the event (2) $f_{R,b}^i(R_b)$ is not most frequent and $\mathsf{D'}_{\Sigma_{Eq}}^1(f_{L,b}^i(L), f_{R,b}^i(R_b), \cdot, \cdot) \neq \perp$ occurs more than y_t times in $H^{2,i,a}$ is at most $((1 + 2^{-\lambda})/2)^{y_t} + 2^{-k}$.

We first analyze the event (1). If $f_{L,b}^i(L_b) = L'_b$ is not the most frequent query in $\mathcal{S}'_{L,b}$ then, by definition,

$$\Pr[f_{L,b}^i(\widehat{L}_b) = L'_b \mid \mathsf{Out}_A^{i-1} = \widehat{Out}_A^{i-1}] \leq 1/2. \qquad (4)$$

Recall that, by the arguments in the proof of the previous Claim, in $H^{2,i-1,b}$ (and hence also $H^{2,i,a}$), for *good* pairs ($\mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1} \wedge \mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2}$):

$$\Pr[f^i_{L,b}(L_b) = L'_b \mid R_b = \widehat{R}_b \wedge \mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1} \wedge \mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2}] \in (1 \pm 2^{-\lambda}) \Pr[f^i_{L,b}(L_b) = L'_b \mid \mathsf{Out}^{i-1}_{A,b} = \widehat{Out}^{i-1}_{A,b}].$$

Combining with (4):

$$\Pr[f^i_{L,b}(L_b) = L'_b \mid R_b = \widehat{R}_b \wedge \mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1} \wedge \mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2}] \leq (1 + 2^{-\lambda})/2.$$

This implies that for every fixed $R_b = \widehat{R}_b$ (for which $\Pr[R_b = \widehat{R}_b \wedge \mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1} \wedge \mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2}] > 0$),

$$\begin{aligned}
&\Pr[f^i_{L,b}(L_b) \text{ is not most frequent } \wedge \mathsf{D}'(f^i_{L,b}(L_b), f^i_{R,b}(R_b)) \neq \perp \mid R_b = \widehat{R}_b \wedge \mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1} \wedge \mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2}] \\
&\quad = \Pr[f^i_{L,b}(L_b) \text{ is not most frequent } \wedge f^i_{L,b}(L) = \rho_L(f^i_{R,b}(R_b)) \mid R_b = \widehat{R}_b \wedge \mathsf{Out}^{i-1}_{A,1} = \widehat{Out}^{i-1}_{A,1} \wedge \mathsf{Out}^{i-1}_{A,2} = \widehat{Out}^{i-1}_{A,2}] \\
&\quad \leq (1 + 2^{-\lambda})/2.
\end{aligned}$$

The probability that this occurs y_t times for y_t distinct values of $j \leq [i]$, where all outcomes of $\mathsf{Out}^j_{A,1} = \widehat{Out}^j_{A,1} \wedge \mathsf{Out}^j_{A,2} = \widehat{Out}^j_{A,2}$ are *good* for all j is at most $((1 + 2^{-\lambda})/2)^{y_t}$. Since by Claim 5.3, with probability $1 - 2^{-k}$, all outcomes $\mathsf{Out}^j_{A,1} = \widehat{Out}^j_{A,1} \wedge \mathsf{Out}^j_{A,2} = \widehat{Out}^j_{A,2}$ are *good* for all $j \in [q]$, the upperbound for event (1) follows.

The proof for event (2) is analogous.

Claim 5.6. In Hybrid 3, for all (even inefficient) distinguishers D, it holds that

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

Proof. We first compute

$$\begin{aligned}
O &= \sum_{m'^2} \Pr[m^2 = m'^2 \wedge m^1 = m_0 + m'^2 \mid \mathsf{Out}_{A,1}, \mathsf{Out}_{A,2}] + \sum_{m'^2} \Pr[m^2 = m'^2 \wedge m^1 = m_1 + m'^2 \mid \mathsf{Out}_{A,1}, \mathsf{Out}_{A,2}] \\
&= \sum_{m'^2} \Pr[m^2 = m'^2 \mid \mathsf{Out}_{A,1}, \mathsf{Out}_{A,2}] \cdot \Pr[m^1 = m_0 + m'^2 \mid \mathsf{Out}_{A,1}, \mathsf{Out}_{A,2}] \\
&\qquad + \sum_{m'^2} \Pr[m^2 = m'^2 \mid \mathsf{Out}_{A,1}, \mathsf{Out}_{A,2}] \cdot \Pr[m^1 = m_1 + m'^2 \mid \mathsf{Out}_{A,1}, \mathsf{Out}_{A,2}] \\
&\geq (2^{-\lambda} - 2^{-k}) \cdot (\sum_{m'^2} \Pr[m^1 = m_0 + m'^2 \mid \mathsf{Out}_{A,1}, \mathsf{Out}_{A,2}] + \sum_{m'^2} \Pr[m^1 = m_1 + m'^2 \mid \mathsf{Out}_{A,1}, \mathsf{Out}_{A,2}]) \\
&= 2 \cdot (2^{-\lambda} - 2^{-k}) \\
&= 2 \cdot 2^{-\lambda} - 2 \cdot 2^{-k}.
\end{aligned}$$

where the first inequality follows from Claim 5.3.

So

$$
\begin{aligned}
\Pr[\text{ message is } m_b \mid \mathsf{Out}_{A,1}, \mathsf{Out}_{A,2}] &= \frac{\sum_{m'_2} \Pr[m^2 = m'^2 \mid \mathsf{Out}_{A,1}, \mathsf{Out}_{A,2}] \cdot \Pr[m^1 = m_b + m^2 \mid \mathsf{Out}_{A,1}, \mathsf{Out}_{A,2}]}{O} \\
&\leq (2^{-\lambda} + 2^{-k}) \cdot \frac{\sum_{m'_2} \Pr[m^1 = m_b + m^2 \mid \mathsf{Out}_{A,1}, \mathsf{Out}_{A,2}]}{O} \\
&= \frac{2^{-\lambda} + 2^{-k}}{O} \\
&= \frac{2^{-\lambda} + 2^{-k}}{2 \cdot 2^{-\lambda} - 2 \cdot 2^{-k}} \\
&\leq \frac{2^{-\lambda} + 3 \cdot 2^{-k}}{2 \cdot 2^{-\lambda}} \\
&= 1/2 + \frac{3 \cdot 2^{-k}}{2 \cdot 2^{-\lambda}} \\
&= 1/2 + O(2^{-\lambda}).
\end{aligned}
$$

References

1. Abe, M., Groth, J., Ohkubo, M.: Separating short structure-preserving signatures from non-interactive assumptions. In Lee, D.H., Wang, X., eds.: ASIACRYPT 2011. Volume 7073 of LNCS., Springer, Heidelberg (December 2011) 628–646
2. Aggarwal, D., Agrawal, S., Gupta, D., Maji, H.K., Pandey, O., Prabhakaran, M.: Optimal computational split-state non-malleable codes. [50] 393–417
3. Aggarwal, D., Dodis, Y., Kazana, T., Obremski, M.: Non-malleable reductions and applications. In Servedio, R.A., Rubinfeld, R., eds.: 47th ACM STOC, ACM Press (June 2015) 459–468
4. Aggarwal, D., Dodis, Y., Lovett, S.: Non-malleable codes from additive combinatorics. In Shmoys, D.B., ed.: 46th ACM STOC, ACM Press (May / June 2014) 774–783
5. Aggarwal, D., Dottling, N., Nielsen, J.B., Obremski, M., Purwanto, E.: Continuous non-malleable codes in the 8-split-state model. Cryptology ePrint Archive, Report 2017/357 (2017) https://eprint.iacr.org/2017/357.
6. Aggarwal, D., Dziembowski, S., Kazana, T., Obremski, M.: Leakage-resilient non-malleable codes. [31] 398–426
7. Agrawal, S., Gupta, D., Maji, H.K., Pandey, O., Prabhakaran, M.: Explicit non-malleable codes against bit-wise tampering and permutations. In Gennaro, R., Robshaw, M.J.B., eds.: CRYPTO 2015, Part I. Volume 9215 of LNCS., Springer, Heidelberg (August 2015) 538–557
8. Agrawal, S., Gupta, D., Maji, H.K., Pandey, O., Prabhakaran, M.: A rate-optimizing compiler for non-malleable codes against bit-wise tampering and permutations. [31] 375–397
9. Ball, M., Dachman-Soled, D., Guo, S., Malkin, T., Tan, L.Y.: Non-malleable codes for small-depth circuits. Cryptology ePrint Archive, Report 2018/207 (2018) https://eprint.iacr.org/2018/207.
10. Ball, M., Dachman-Soled, D., Kulkarni, M., Malkin, T.: Non-malleable codes for bounded depth, bounded fan-in circuits. In Fischlin, M., Coron, J.S., eds.: EUROCRYPT 2016, Part II. Volume 9666 of LNCS., Springer, Heidelberg (May 2016) 881–908
11. Ball, M., Dachman-Soled, D., Kulkarni, M., Malkin, T.: Non-malleable codes from average-case hardness: AC^0, decision trees, and streaming space-bounded tampering. [56] 618–650
12. Barak, B.: How to go beyond the black-box simulation barrier. In: 42nd FOCS, IEEE Computer Society Press (October 2001) 106–115
13. Barak, B., Mahmoody-Ghidary, M.: Merkle puzzles are optimal - an $O(n^2)$-query attack on any key exchange from a random oracle. In Halevi, S., ed.: CRYPTO 2009. Volume 5677 of LNCS., Springer, Heidelberg (August 2009) 374–390
14. Bitansky, N., Paneth, O.: From the impossibility of obfuscation to a new non-black-box simulation technique. In: 53rd FOCS, IEEE Computer Society Press (October 2012) 223–232
15. Chandran, N., Goyal, V., Mukherjee, P., Pandey, O., Upadhyay, J.: Block-wise non-malleable codes. Cryptology ePrint Archive, Report 2015/129 (2015) http://eprint.iacr.org/2015/129.
16. Chandran, N., Goyal, V., Mukherjee, P., Pandey, O., Upadhyay, J.: Block-wise non-malleable codes. In Chatzigiannakis, I., Mitzenmacher, M., Rabani, Y., Sangiorgi, D., eds.: ICALP 2016. Volume 55 of LIPIcs., Schloss Dagstuhl (July 2016) 31:1–31:14

17. Chandran, N., Kanukurthi, B., Ostrovsky, R.: Locally updatable and locally decodable codes. [53] 489–514
18. Chandran, N., Kanukurthi, B., Raghuraman, S.: Information-theoretic local non-malleable codes and their applications. [50] 367–392
19. Chattopadhyay, E., Li, X.: Non-malleable codes and extractors for small-depth circuits, and affine functions. In Hatami, H., McKenzie, P., King, V., eds.: 49th ACM STOC, ACM Press (June 2017) 1171–1184
20. Chattopadhyay, E., Zuckerman, D.: Non-malleable codes against constant split-state tampering. In: 55th FOCS, IEEE Computer Society Press (October 2014) 306–315
21. Chattopadhyay, E., Zuckerman, D.: Explicit two-source extractors and resilient functions. [62] 670–683
22. Cheraghchi, M., Guruswami, V.: Capacity of non-malleable codes. In Naor, M., ed.: ITCS 2014, ACM (January 2014) 155–168
23. Cheraghchi, M., Guruswami, V.: Non-malleable coding against bit-wise and split-state tampering. [53] 440–464
24. Coretti, S., Dodis, Y., Tackmann, B., Venturi, D.: Non-malleable encryption: Simpler, shorter, stronger. In Kushilevitz, E., Malkin, T., eds.: TCC 2016-A, Part I. Volume 9562 of LNCS., Springer, Heidelberg (January 2016) 306–335
25. Coretti, S., Maurer, U., Tackmann, B., Venturi, D.: From single-bit to multi-bit public-key encryption via non-malleable codes. [31] 532–560
26. Coron, J.S.: Security proof for partial-domain hash signature schemes. In Yung, M., ed.: CRYPTO 2002. Volume 2442 of LNCS., Springer, Heidelberg (August 2002) 613–626
27. Dachman-Soled, D., Kulkarni, M., Shahverdi, A.: Tight upper and lower bounds for leakage-resilient, locally decodable and updatable non-malleable codes. In Fehr, S., ed.: PKC 2017, Part I. Volume 10174 of LNCS., Springer, Heidelberg (March 2017) 310–332
28. Dachman-Soled, D., Kulkarni, M., Shahverdi, A.: Local non-malleable codes in the bounded retrieval model. In Abdalla, M., Dahab, R., eds.: PKC 2018, Part II. Volume 10770 of LNCS., Springer, Heidelberg (March 2018) 281–311
29. Dachman-Soled, D., Liu, F.H., Shi, E., Zhou, H.S.: Locally decodable and updatable non-malleable codes and their applications. [31] 427–450
30. Di Crescenzo, G., Ishai, Y., Ostrovsky, R.: Non-interactive and non-malleable commitment. In: 30th ACM STOC, ACM Press (May 1998) 141–150
31. Dodis, Y., Nielsen, J.B., eds.: TCC 2015, Part I. In Dodis, Y., Nielsen, J.B., eds.: TCC 2015, Part I. Volume 9014 of LNCS., Springer, Heidelberg (March 2015)
32. Döttling, N., Nielsen, J.B., Obremski, M.: Information theoretic continuously non-malleable codes in the constant split-state model. Cryptology ePrint Archive, Report 2017/357 (2017) http://eprint.iacr.org/2017/357.
33. Dziembowski, S., Kazana, T., Obremski, M.: Non-malleable codes from two-source extractors. In Canetti, R., Garay, J.A., eds.: CRYPTO 2013, Part II. Volume 8043 of LNCS., Springer, Heidelberg (August 2013) 239–257
34. Dziembowski, S., Pietrzak, K., Wichs, D.: Non-malleable codes. In Yao, A.C.C., ed.: ICS 2010, Tsinghua University Press (January 2010) 434–452
35. Faust, S., Hostáková, K., Mukherjee, P., Venturi, D.: Non-malleable codes for space-bounded tampering. In Katz, J., Shacham, H., eds.: CRYPTO 2017, Part II. Volume 10402 of LNCS., Springer, Heidelberg (August 2017) 95–126
36. Faust, S., Mukherjee, P., Nielsen, J.B., Venturi, D.: Continuous non-malleable codes. [53] 465–488
37. Faust, S., Mukherjee, P., Nielsen, J.B., Venturi, D.: A tamper and leakage resilient von neumann architecture. In Katz, J., ed.: PKC 2015. Volume 9020 of LNCS., Springer, Heidelberg (March / April 2015) 579–603
38. Fischlin, M., Schröder, D.: On the impossibility of three-move blind signature schemes. In Gilbert, H., ed.: EUROCRYPT 2010. Volume 6110 of LNCS., Springer, Heidelberg (May / June 2010) 197–215
39. Fortnow, L., Vadhan, S.P., eds.: 43rd ACM STOC. In Fortnow, L., Vadhan, S.P., eds.: 43rd ACM STOC, ACM Press (June 2011)
40. Fuchsbauer, G., Konstantinov, M., Pietrzak, K., Rao, V.: Adaptive security of constrained PRFs. In Sarkar, P., Iwata, T., eds.: ASIACRYPT 2014, Part II. Volume 8874 of LNCS., Springer, Heidelberg (December 2014) 82–101
41. Garg, S., Bhaskar, R., Lokam, S.V.: Improved bounds on security reductions for discrete log based signatures. In Wagner, D., ed.: CRYPTO 2008. Volume 5157 of LNCS., Springer, Heidelberg (August 2008) 93–107
42. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. [39] 99–108
43. Gertner, Y., Kannan, S., Malkin, T., Reingold, O., Viswanathan, M.: The relationship between public key encryption and oblivious transfer. In: 41st FOCS, IEEE Computer Society Press (November 2000) 325–335
44. Goyal, V., Pandey, O., Richelson, S.: Textbook non-malleable commitments. [62] 1128–1141
45. Impagliazzo, R., Rudich, S.: Limits on the provable consequences of one-way permutations. In: 21st ACM STOC, ACM Press (May 1989) 44–61

46. Jafargholi, Z., Wichs, D.: Tamper detection and continuous non-malleable codes. [31] 451–480
47. Kanukurthi, B., Obbattu, S.L.B., Sekar, S.: Four-state non-malleable codes with explicit constant rate. In Kalai, Y., Reyzin, L., eds.: TCC 2017, Part II. Volume 10678 of LNCS., Springer, Heidelberg (November 2017) 344–375
48. Kanukurthi, B., Obbattu, S.L.B., Sekar, S.: Non-malleable randomness encoders and their applications. [56] 589–617
49. Kiayias, A., Liu, F.H., Tselekounis, Y.: Practical non-malleable codes from l-more extractable hash functions. In Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S., eds.: ACM CCS 16, ACM Press (October 2016) 1317–1328
50. Kushilevitz, E., Malkin, T., eds.: TCC 2016-A, Part II. In Kushilevitz, E., Malkin, T., eds.: TCC 2016-A, Part II. Volume 9563 of LNCS., Springer, Heidelberg (January 2016)
51. Lee, C.J., Lu, C.J., Tsai, S.C., Tzeng, W.G.: Extracting randomness from multiple independent sources. IEEE Transactions on Information Theory **51**(6) (2005) 2224–2227
52. Li, X.: Non-malleable extractors and non-malleable codes: Partially optimal constructions. Cryptology ePrint Archive, Report 2018/353 (2018) https://eprint.iacr.org/2018/353.
53. Lindell, Y., ed.: TCC 2014. In Lindell, Y., ed.: TCC 2014. Volume 8349 of LNCS., Springer, Heidelberg (February 2014)
54. Liu, F.H., Lysyanskaya, A.: Tamper and leakage resilience in the split-state model. In Safavi-Naini, R., Canetti, R., eds.: CRYPTO 2012. Volume 7417 of LNCS., Springer, Heidelberg (August 2012) 517–532
55. Naor, M.: Bit commitment using pseudo-randomness. In Brassard, G., ed.: CRYPTO'89. Volume 435 of LNCS., Springer, Heidelberg (August 1990) 128–136
56. Nielsen, J.B., Rijmen, V., eds.: EUROCRYPT 2018, Part III. In Nielsen, J.B., Rijmen, V., eds.: EUROCRYPT 2018, Part III. Volume 10822 of LNCS., Springer, Heidelberg (April / May 2018)
57. Paillier, P., Vergnaud, D.: Discrete-log-based signatures may not be equivalent to discrete log. In Roy, B.K., ed.: ASIACRYPT 2005. Volume 3788 of LNCS., Springer, Heidelberg (December 2005) 1–20
58. Pass, R.: Limits of provable security from standard assumptions. [39] 109–118
59. Seurin, Y.: On the exact security of Schnorr-type signatures in the random oracle model. In Pointcheval, D., Johansson, T., eds.: EUROCRYPT 2012. Volume 7237 of LNCS., Springer, Heidelberg (April 2012) 554–571
60. Simon, D.R.: Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Nyberg, K., ed.: EUROCRYPT'98. Volume 1403 of LNCS., Springer, Heidelberg (May / June 1998) 334–345
61. Wichs, D.: Cryptographic resilience to continual information leakage. PhD Thesis (2011) http://www.ccs.neu.edu/home/wichs/thesis.pdf.
62. Wichs, D., Mansour, Y., eds.: 48th ACM STOC. In Wichs, D., Mansour, Y., eds.: 48th ACM STOC, ACM Press (June 2016)

A CNM Randomness Encoder to Single-Bit CNMC

Let $\Pi = (\mathsf{CRSGen}, \mathsf{E} = (\mathsf{E}_1, \mathsf{E}_2), \mathsf{D})$ be a CNM Randomness Encoder. To construct a CNMC for a single bit from Π, we must show how to use E to encode a message $b \in \{0,1\}$. In the case of the CNM Randomness Encoder given in Section 4, this can be done by choosing random $c_L, c_R \in \mathbb{F}_{2^\lambda}^{\frac{\ell}{\lambda}}$, conditioned on the parity of $\langle c_L, c_R \rangle$ being equal to b. and then running $\mathsf{E}(c_L || c_R || r_{enc} || r_L || r_R)$ In general, one can run $\mathsf{E}(r)$ repeatedly until $\mathsf{E}_2(r)$ outputs a random message m with parity equal to b. (this will give an encode algorithm that runs in polynomial time with all but negligible probability).

Now, it can be immediately seen that an adversary who breaks the security formulation of CNMC given in Definition 4 must also break the security of the CNM Randomness Encoder, given in Definition 5.

B Perfectly Unique CNMC Implies One-to-one Commitments

Let $\Pi = (\mathsf{CRSGen}_\Pi, \mathsf{E}, \mathsf{D})$ be a CNMC such that with at all but negligible probability over choice of CRS, Σ, choice of $b \leftarrow \{0,1\}$ and randomness for generating $(L, R) \leftarrow \mathsf{E}_\Sigma(b)$, there exists a *single* $R' = R$ such that $\mathsf{D}_\Sigma(L, R') \neq \perp$. Then we will construct a one-round one-to-one commitment (in the CRS model) as follows.

- $\mathsf{CRSGen}(1^n)$: Run $\mathsf{CRSGen}_\Pi(1^n)$ to generate Σ. Output Σ.

- $\mathsf{Commit}(\Sigma, b)$: Generate $(L, R) \leftarrow \mathsf{E}_\Sigma(b)$. Output $c = L$. Output decommitment $d = R$.
- $\mathsf{Open}(\Sigma, c, d)$: Parse $c = L$ and $d = R$. Output $b = \mathsf{D}_\Sigma(L, R)$.

It can immediately be seen that the scheme is a one-to-one commitment by the perfect uniqueness property of the underlying CNMC.

The hiding property of the commitment scheme is implied by non-malleability. This is because if the adversary can predict b by seeing L then the adversary can tamper with L in a way that depends on b (e.g. leave L untouched if $b = 0$ and set L to a random value if $b = 1$), thus breaking non-malleability.