Abstract. There have been many successes in constructing explicit non-malleable codes for various classes of tampering functions in recent years, and strong existential results are also known. In this work we ask the following question: When can we rule out the existence of a non-malleable code for a tampering class \( F \)?

We show that non-malleable codes are impossible to construct for three different tampering classes:

- Functions that change \( d/2 \) symbols, where \( d \) is the distance of the code;
- Functions where each input symbol affects only a single output symbol;
- Functions where each of the \( n \) output bits is a function of \( n - \log n \) input bits.

We additionally rule out constructions of non-malleable codes for certain classes \( F \) via reductions to the assumption that a distributional problem is hard for \( F \), that make black-box use of the tampering functions in the proof. In particular, this yields concrete obstacles for the construction of efficient codes for \( \text{NC} \), even assuming average-case variants of \( P \not\subseteq \text{NC} \).

1 Introduction

Since the introduction of non-malleable codes (NMC) by Dziembowski, Pietrzak, and Wichs in 2010, there has been a long line of work constructing non-malleable codes for various classes [DPW18]. A plethora of upper bounds, explicit and implicit (to varying degrees), have been shown for a wealth of classes of tampering functions. However, to our knowledge, relatively little is known about when non-malleability is impossible. In this work, we initiate the study of the limits to non-malleability.

Non-malleability for a class \( F \) is defined via the following “tampering” experiment:

1. Encode message \( m \) using a (public) randomized encoding algorithm: \( c \leftarrow E(m) \),
2. Tamper the codeword: \( \tilde{c} = f(c) \),
3. Decode the tampered codeword (with public decoder): \( \tilde{m} = D(\tilde{c}) \).

Roughly, the encoding scheme, \((E, D)\), is non-malleable for a class \( F \), if for any \( f \in F \) the result of the above experiment, \( \tilde{m} \), is either identical to the original message, or completely unrelated. More precisely, the outcome of a \( F \)-tampering experiment should be simulatable without knowledge of the message \( m \) (using a special flag “same” to capture the case of unchanged message).

[DPW18] showed that, remarkably, this definition is achievable for any \( F \) such that \( \log \log |F| < n - 5 \cdot \log(n) \). However the definition is not achievable in general. It is easy to observe that if \( F \) is the class of all functions, there is a trivial tampering attack: decode, maul, and re-encode. This same observation rules out the possibility of efficient codes against efficient tampering, as this attack only requires that decoding and outputting constants conditioned on the result is in the tampering class. By a similar argument, the decoding function of a non-malleable code with respect to the distribution formed by encoding a random one-bit message can be seen as existence of hard decision problem for the tampering class. (This, in turn, informs us of where to hope for unconditional constructions.)

In this work, we give a variety of impossibility results for non-malleable codes, in disparate tampering regimes. We present 3 unconditional impossibility results for various classes, which hold even for inefficient
NMC. Additionally, we rule out constructions of NMC for a wide range of complexity classes with security reductions that are only given black-box access to the tampering function.

To our knowledge, the only previously-known impossibility results beyond the simple observations above, are related to other variants of NMC. These include bounds on locality of locally decodable and updatable NMC, bounds on continuous NMC, and impossibility of “look-ahead” or “block-wise” NMC (which also follows from a simple observation). There are also several bounds related to the rate of NMC. We discuss these and other related works in Section 1.3. In contrast, our results hold regardless of rate. In fact, our lower bounds rule out even message spaces of size two or three.

1.1 Strictly Impossible

We identify 3 tampering regimes where non-malleability is strictly impossible.

On tampering functions that change $d/2$ symbols, where $d$ is the distance of the code. It is common to present non-malleable codes as a strict relaxation of error correcting codes. Non-malleable codes only guarantee correctness of decoding in the absence of errors, and consequently provide “security” for a wider range of tampering functions, in particular tampering functions that can modify more symbols of the codeword. However, note that in all known results, there is a trade-off: Non-malleable codes allow for modifying more (potentially all) symbols of the codeword than error correcting codes but require that the computation of the tampering function is restricted in some way, while error correcting codes can tolerate modification of fewer codeword symbols, but do not place any other restrictions on the tampering adversary.

In the current work, we ask whether this is in fact necessary. Specifically, can we construct non-malleable codes that allow for modifying more symbols of the codeword than error correcting codes without placing any other restrictions on the tampering? Note that for error correcting codes it is known that if the distance of the code is $d$, it is not possible to correct when $d/2$ symbols are modified, but there are constructions that allow for error correction after arbitrary modification of at most $(d-1)/2$ symbols (e.g., Reed-Solomon error-correcting codes achieve this bound).

We fully resolve our question, showing that for message space of size 2, non-malleable codes that tolerate arbitrary modification of $d - 1$ symbols are possible (via a repetition code, refer to Section 3 for details). On the other hand, for message space of size greater than 2, it is impossible to construct non-malleable codes with distance $d$ for tampering functions that arbitrarily modify $d/2$ codeword symbols. This indicates that for message space larger than 2, in order to obtain improved parameters beyond what is possible with error correcting codes, imposing some additional restrictions on the tampering function is necessary.

On tampering functions where each input symbol effects at most one output symbol. In their recent work, Ball et al. [BDKM16] presented unconditional NMC for the class of output-local functions, where each output symbol depends on a bounded number of input symbols. As an intermediate step, they also considered the class of functions that are both input and output local. The class of input-local functions is the class of functions where each input symbol affects a bounded number of output symbol. A natural question is whether non-malleable codes can be constructed for the class of input-local functions, where for codeword length $n$, each input bit affects $\ll n$ output bits.

In the current work, we answer this question negatively in a very strong sense: We show that even achieving NMC for 1-input local functions (where each input bit affects at most one output bit) is impossible. In fact, our proof shows an even stronger result: the impossibility holds even if each input symbol can only affect the same single output symbol. That is, it is impossible to construct NMC for the tampering class that allows to change only one codeword symbol in a way that depends on the input (while the other symbols may be changed into constant values). Stated like this, this result can also be viewed as an extension of our first result above on the maximum number of symbols that can be modified in a non-malleable code.

On tampering functions where each output symbol depends on $n - \log n$ input symbols. Here we move on to consider achieving NMC for output-local tampering functions. The prior work of [BDKM16] constructed efficient NMC for tampering functions with locality $n^\epsilon$, for constant $\epsilon$. The size of the class of all output-local tampering functions (with locality sufficiently smaller than $n$) is also bounded in size and therefore
non-malleable codes for this class follow from the existential results of [DPW18]. A natural question is how large can the output-locality be?

We prove the impossibility of non-malleable codes for the class of \((n - \log n)\)-output-local tampering functions. In addition to the above motivation, parity over \(n\) bits is average-case hard for this class.\(^1\) Therefore, our impossibility result can be viewed as a “separation” between average-case hardness and non-malleability, as it exhibits a class for which we have average-case hardness bounds, but cannot construct non-malleable codes for. Furthermore, the class \(\mathcal{F}’\) constructed in our lower bound proof has size only \(4^n \cdot 2^{2^{n-\log(n)}}\), which in turn means that \(\log \log |\mathcal{F}'| = n - \log(n)\). On the other hand, the aforementioned result of Dziembowski et al. [DPW18] shows existence of a \(1/n\)-non-malleable code for any class \(\mathcal{F}\) such that \(\log \log |\mathcal{F}| \leq n - 5 \log(n)\). Thus, our lower bound result is close to matching the existential upper bound.

The lower bound for the class of input-local functions holds for coding scheme that enjoy deterministic decoding and perfect correctness. Both properties are required in the standard definition of non-malleable codes.\(^2\) The lower bound for the class of \(n - \log(n)\) output-local functions holds even for coding schemes that have randomized decoding and perfect correctness. The lower bound for the class of functions that change \(d/2\) symbols holds even for coding schemes with randomized decoding and imperfect correctness.

1.2 Impossibility of Black-Box Security Reductions.

In recent work, unconditional constructions of non-malleable codes for progressively larger tampering classes, such as \(\text{NC}^0\) [BDKM16, CL17b, BDG’18] and \(\text{AC}^0\) [CL17b, BDG’18], have been presented. In fact, the construction of [BDG’18] remains secure for circuit depths as large as \(\Theta(\log(n) / \log(n))\). Moreover, due to the impossibility of efficient NMC for all of \(\mathcal{P}\), extending their result to obtain unconditional NMC for circuits with asymptotically larger depth would require separating \(\mathcal{P}\) from \(\text{NC}^0\), a problem that is well out of reach with current complexity-theoretic techniques. However, rather than ruling out such constructions entirely, in this regime we ask what are the minimal assumptions necessary for achieving non-malleable codes for \(\text{NC}^1\), as well as other classes \(\mathcal{F}\) that are believed to be strictly contained in \(\mathcal{P}\).

The above question was partially addressed by Ball et al. [BDKM18, BDK+19] in their recent work, where they presented a general framework for construction of NMC for various classes \(\mathcal{F}\) in the CRS model and under cryptographic assumptions. Instantiating their framework for \(\text{NC}^1\) yields a computational, CRS-model construction of 1-bit NMC for \(\text{NC}^1\), assuming there is a distributional problem that is hard for \(\text{NC}^1\), but easy for \(\mathcal{P}\). Moreover, such distributional problems for \(\text{NC}^1\) can be based on worst-case assumptions.\(^3\)

In this work, we ask whether 1-bit non-malleable codes for \(\text{NC}^1\) in the standard (no-CRS) model can be constructed from the assumption that there are distributional problems that are hard for \(\text{NC}^1\) but easy for \(\mathcal{P}\). Recall that this assumption is minimal, since the decoding function of a 1-bit non-malleable code for \(\text{NC}^1\) w.r.t. the distribution of random encodings of 1 bit messages yields such a distributional problem.

We provide a negative answer, proving that, under black-box reductions (restricting use of the tampering function in the security proof to be black-box), this is impossible.

Specifically, we define a notion of black-box reductions for the setting of 1-bit non-malleable codes \((E, D)\) against a complexity class \(\mathcal{F}\) to a distributional problem \((\Psi, L)\) that is hard for \(\mathcal{F}\). This type of reduction is required to use the “adversary” —i.e. the tampering function in our setting—in a black-box manner, but is not restricted in its use of the underlying assumptions. Thus, the reduction \(R\) is provided black-box access to the tampering function \(f\) and must use it to contradict the assumption on the distributional problem \((\Psi, L)\). At a high level (skipping some technical details), we require two properties of a black-box reduction \(R\) from \((E, D)\) to \((\Psi, L)\):

\(^1\) Note that, even arbitrary decision trees of depth \(n - 1\) have no advantage in computing the parity of \(n\) bits with respect to the uniform distribution. [BdW02]

\(^2\) For the class of output-local functions (where each output depends on at most \(\ell\) inputs) we have explicit constructions with randomized decoding for \(\ell < n/\log n\) [BDKM16], whereas constructions with deterministic decoding are known for locality up to \(n^{1/2-\epsilon}\) for small \(\epsilon\). [CL17a, BDG’18].

\(^3\) Assuming \(\oplus L/poly \not\subset \text{NC}^1\) yields a distributional problem since randomized encodings for \(\oplus L/poly\) are known to exist [AIK04, BGJ’16, DVV16, AR16].
If the tampering function \( f \) succeeds in breaking the non-malleable code, the reduction, \( R' \), should succeed, regardless of whether \( f \in \mathcal{F} \). This represents the fact that \( R \) uses \( f \) in a black-box manner.

For any \( f \in \mathcal{F} \), \( R' \) must also be in \( \mathcal{F} \), and in particular, \( R \) itself must be in \( \mathcal{F} \). This represents the fact that the black-box reduction \( R \) should allow one to obtain a contradiction to the assumption that \((D,L)\) is hard for \( \mathcal{F} \), in the case that \((E,D)\) is malleable by \( \mathcal{F} \).

Note that for arbitrary classes \( \mathcal{F} \) (unlike the usual polynomial-time adversaries typically used in cryptography), the fact that \( R \in \mathcal{F} \) and \( f \in \mathcal{F} \) does not necessarily imply that \( R' \in \mathcal{F} \). This introduces some additional complexity in our definitions and also requires us to restrict our end results to classes \( \mathcal{F} \) that behave appropriately under composition.

Indeed, we present general impossibility results for constructing 1-bit non-malleable codes for a class \( \mathcal{F} \) from a distributional problem that is hard for \( \mathcal{F} \) but easy for \( \mathcal{P} \). We present three types of results: Results ruling out security parameter preserving reductions for tampering class \( \mathcal{F} \) that behave nicely under composition, results ruling out “approximate” security parameter preserving reductions for tampering class \( \mathcal{F} \) with slightly stronger compositional properties and results ruling out non-security parameter preserving reductions for tampering class \( \mathcal{F} \) that are fully closed under composition. See Definitions 17, 18 and Lemmas 2, 3, 4 for the formal statements.

Briefly, security parameter preserving reductions have the property that the reduction only queries the adversary (in our case the tampering function) on the same security parameter that it receives as input. The notion of “approximate” security parameter preserving reductions is new to this work. Such reductions are parameterized by polynomial functions \( \ell(\cdot) \), \( u(\cdot) \) and on input security parameter \( n \), the reduction may query the adversary on any security parameter in the range \( \ell(n) \) to \( u(n) \). Finally, in a non-security parameter preserving reduction, the reduction receives security parameter \( n \) as input and may query the adversary on arbitrary security parameter \( n' \). Note that \( n'(n) \) must be in \( O(n^c) \) for some constant \( c \), since the reduction must be polynomial time.

We can instantiate the tampering class \( \mathcal{F} \) from our generic lemma statements with various classes of interest: Our results on security parameter preserving and approximate security parameter preserving reductions apply to the class \( \mathsf{NC}^1 \) as a special case. Our result ruling out non-security parameter preserving reductions applies to the class (non-uniform) \( \mathsf{NC} \) as a special case. See Corollaries 1, 2, 3 for the formal statements. As the proofs are already quite involved, we make the simplifying assumption of deterministic decoding and perfect correctness. However, this is not inherent to the proof and we expect the results to extend to coding schemes with imperfect correctness and randomized decoding.

### 1.3 Related Work

**Non-Malleable Codes.** Non-malleable codes (NMC) were introduced in the seminal work of Dziembowski, Pietrzak and Wichs [DPW18]. In the same paper they pointed out the simple impossibility result for NMC for all polynomial tampering functions. Since then NMC have been studied in the information-theoretic as well as computational settings. Liu and Lysyanskaya [LL12] studied the split-state classes of tampering functions and constructed computationally secure NMC for split-state tampering. A long line of work followed in both the computational [AAG+]16 as well as information theoretic setting with a series of advances—reduced number of states, improved rate, or adding desirable features to the scheme [DKO13, ADL14, CZ14, ADKO15a, AGM+15b, AAG+16, KOS17, Li18]. Recently efficient NMC have been constructed for tampering function classes other than split-state tampering [BDKM16, AGM+15a, CL17b, FHMV17, BDKM18, BDG+18, BDK+19, BGW19] in both the computational and information-theoretic setting. Additionally, [DPW18, CG14a, FMW18] present various inefficient, existential or randomized constructions for more general classes of tampering functions. These results are sometimes presented as efficient constructions in a random-oracle or CRS model. Other works on non-malleable codes include [FMNV14, CG14b, CKO14, ADKO15b, JW15, DLSZ15, FMV15, ADKO15a, CR16, CGM+16, KLT16, DKS17, ADN+19, DKS18, KOS18, OPV18, KLT18, FNSV18, CKOS18, CL18, RS18, CFV19].
Bounds on Non-Malleable Codes. Surprisingly, understanding the limitations and bounds on NMC has received relatively less attention. While there have been a few previous works exploring the lower and upper bounds on NMC and its variants [DPW18, CG14a, CGM+16, DKS17, DK19], most of the effort has been focused on understanding and/or improving the bounds on the rates of NMC [AAG+16, AGM+15a, AGM+15b, KOS17, Li18, CFV19]

Perhaps the closest to this work are the results of [CG14a, DKS17, DK19]. Cheragachi and Gururusswami [CG14a] studied the “capacity” of non-malleable codes in order to understand the optimal bounds on the efficiency of non-malleable codes. They showed that information theoretically secure efficient NMC exist for tampering families $F$ of size $|F|$ if $\log \log |F| \leq \alpha n$ for $0 \leq \alpha < 1$, moreover these NMC have optimal rate of $1 - \alpha$ with error $\epsilon \in O(1/\text{poly}(n))$. Dachman-Soled, Kulkarni, and Shahverdi [DKS17] studied the bounds on the efficiency of locally decodable and updatable NMC. They showed that for any locally decodable and updatable NMC which allows rewind attacks, the locality parameter of the scheme must be $\omega(1)$, and gave an improved version of [DLSZ15] construction to match the lower bound in computational setting. Recently, Dachman-Soled and Kulkarni [DK19] studied the bounds on continuous non-malleable codes (CNMC), and showed that 2-split-state CNMC cannot be constructed from any falsifiable assumption without CRS. They also gave a construction of 2-split-state CNMC from injective one-way functions in CRS model. Faust et al. [FMNV14] showed the impossibility of constructing information-theoretically secure 2-split-state CNMC.

Black-Box Separations. Impagliazzo and Rudich [IR89] showed the impossibility of black-box reductions from key agreement to one-way function. Their oracle separation technique subsequently led to sequence of works, ruling out black-box reductions between different primitives. Notable examples are [Sim98] separating collision resistant hash functions from one way functions, and [GKM+00] ruling out oblivious transfer from public key encryption. The meta-reduction technique (cf. [Cor02, PV05, GBL08, FS10, Pas11, GW11, AGO11, Sen12, BM09, FPRI14]) has been used 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 a cryptographic primitive is impossible under “standard assumptions” (e.g. falsifiable or non-interactive assumptions).

2 Preliminaries

2.1 Notation

For any positive integer $n$, $[n] := \{1, \ldots, n\}$. For a vector $x \in \chi$ of length $n$, we denote its hamming weight by $||x|| := |\{i \in [n] : x_i \neq 0\}|$, where $|S|$ is the cardinality of set $S$, and $x_i$ denotes the $i$-th element of $x$. For $x, y \in \{0, 1\}^n$ define their distance to be $d(x, y) := ||x - y||_0$. (I.e. $x$ and $y$ are $\epsilon$-far if $d(x, y) \geq \epsilon$.) We denote the statistical distance between two random variables, $X$ and $Y$, over a domain $S$ to be $\Delta(X, Y) := 1/2 \sum_{s \in S} |\Pr[X = s] - \Pr[Y = s]|$, where $|\cdot|$ denotes the absolute value. We say $X$ and $Y$ are $\epsilon$-close, denoted by $X \approx_\epsilon Y$, if $\Delta(X, Y) \leq \epsilon$.

2.2 Non-Malleable Codes

Definition 1 (Coding Scheme [DPW18]). A Coding scheme, $(E, D)$, consists of a (possibly randomized) encoding function $E : \{0, 1\}^k \rightarrow \{0, 1\}^n$ and a deterministic decoding function $D : \{0, 1\}^n \rightarrow \{0, 1\}^k \cup \{\perp\}$ such that $\forall m \in \{0, 1\}^k, \Pr[D(E(m)) = m] = 1$ (over randomness of $E$).

Definition 2 ($\epsilon$-Non-malleability [DPW18]). Let $F$ be some family of functions. For each function $f \in F$, and $m \in \{0, 1\}^k$, define the tampering experiment:

$$\text{Tamper}_m^f \overset{\text{def}}{=} \left\{ \begin{array}{l} c \leftarrow E(m), \tilde{c} := f(c), \tilde{m} := D(\tilde{c}) , \text{Output} : \tilde{m}. \end{array} \right\}$$
where the randomness of the experiment comes from $E$. We say a coding scheme $(E, D)$ is \(\varepsilon\)-non-malleable with respect to $F$ if for each $f \in F$, there exists a distribution $D_f$ over $\{0, 1\}^k \cup \{\text{same}^*, \bot\}$ such that for every message $m \in \{0, 1\}^k$, we have

$${\text{Tamper}}_m^\varepsilon \approx \begin{cases} \hat{m} \leftarrow D_f, & \text{Output: } m \text{ if } \hat{m} = \text{same}^*; \\ \text{otherwise } \hat{m}. & \end{cases}$$

Here the indistinguishability can be either statistical or computational.

Lemma 1 (Lemma 2 [DKO13]). Let $(E, D)$ be a coding scheme with $E : \{0, 1\} \rightarrow X$ and $D : X \rightarrow \{0, 1\}$. Let $F$ be a set of functions $f : X \rightarrow X$. Then $(E, D)$ is \(\varepsilon\)-non-malleable with respect to $F$ if and only if for every $f \in F$,

$$\Pr_{b \leftarrow \{0, 1\}} [D(f(E(b))) = 1 - b] \leq \frac{1}{2} + \varepsilon,$$

where the probability is over the uniform choice of $b$ and the randomness of $E$.

Definition 3 (\(\varepsilon\)-Malleable Code).

Let $(E, D)$ be a coding scheme with $E : \{0, 1\} \rightarrow X$ and $D : X \rightarrow \{0, 1\}$. Let $F$ be a set of functions $f : X \rightarrow X$. Then $(E, D)$ is \(\varepsilon\)-malleable with respect to $F$, if $\exists f \in F$ such that,

$$\Pr_{b \leftarrow \{0, 1\}} [D(f(E(b))) = 1 - b] \geq \frac{1}{2} + \varepsilon,$$

where the probability is over the uniform choice of $b$ and the randomness of $E$.

2.3 Input/Output Local Functions

We next define input and output local functions. In input local functions, each input bit can affect a bounded number of output bits. In output local functions, each output bit is affected by a bounded number of input bits. Loosely speaking, an input bit $x_i$ affects the output bit $y_j$ if for any boolean circuit computing $f$, there is a path in the underlying DAG from $x_i$ to $y_j$. The formal definitions are below, and our notation follows that of [App14].

Definition 4 ([BDKM16]). We say that a bit $x_i$ affects the boolean function $f$, if $\exists \{x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\} \in \{0, 1\}^{n-1}$ such that,

$f(x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \neq f(x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$.

Given a function $f = (f_1, \ldots, f_n)$ (where each $f_j$ is a boolean function), we say that input bit $x_i$ affects output bit $y_j$, or that output bit $y_j$ depends on input bit $x_i$, if $x_i$ affects $f_j$.

Definition 5 (Input Locality [BDKM16]). A function $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ is said to have input locality $\ell$ if every input bit $f_i$ is affects at most $\ell$ output bits.

Definition 6 (Output Locality [BDKM16]). A function $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ is said to have output locality $m$ if every output bit $f_i$ is dependent on at most $m$ input bits.

Definition 7 (Input Local Functions [App14]). A function $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ is said to be $\ell$-input local, if it has input locality $\ell$.

Definition 8 (Output Local Functions [App14]). A function $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ is said to be $m$-output local, if it has output locality $m$. 

6
Let poly(input size. NC) be a circuit (if for any boolean circuits C, if there exists some circuit C ∈ C, if there is a randomized poly-time algorithm that on input 1^n samples ψ_n.

Definition 11 (ϵ(n)-Hard Distributional Problem). Let (Ψ = {Ψ_n}_n=1, L = {L_n}_n=1) be a distributional problem, we say that (Ψ, L) is ϵ(n)-hard for family of boolean circuits C = {C_n}_n=1, if and only if for every circuit C_n ∈ C,

\[ \Pr_{x \sim \psi_n} [C_n(x) = L_n(x)] \leq \frac{1}{2} + \epsilon(n) \]

Definition 12 (ϵ(n)-Easy Distributional Problem). Let (Ψ = {Ψ_n}_n=1, L = {L_n}_n=1) be a distributional problem, we say that (Ψ, L) is ϵ(n)-easy for family of boolean circuits C = {C_n}_n=1, if there exists some circuit C_n ∈ C,

\[ \Pr_{x \sim \psi_n} [C_n(x) = L_n(x)] \geq \frac{1}{2} - \epsilon(n) \]

2.5 Hardness of Boolean Functions and Composition

Definition 13 (δ-hardness of boolean function). Let f : {0,1}^n → {0,1} be a boolean function, and U_n be uniform distribution over {0,1}^n. Also let 0 < δ < \frac{1}{2}, and n ≤ \frac{2\epsilon}{\delta^2}. We say f is δ-hard for size s if for any boolean circuits C of size at most s

\[ \Pr_{x \sim \{0,1\}^n} [C(x) = f(x)] \leq 1 - \delta \]

Theorem 1 (Theorem 1 [Imp95]). Let f : {0,1}^n → {0,1} be boolean function that δ-hard for size s. Also, let ε > 0. Then ∃ set S ⊆ {0,1}^n and constant c, such that |S| ≥ δ · 2^n and f is ε-hard-core on S for circuits of size s' ≤ c · ε^2 · δ^2 · s.
Definition 14 (Hard Core Set (HCS) Amenable). We say that $\mathcal{F} = \{\mathcal{F}_n\}_{n=1}^\infty$ is HCS-Amenable if for any polynomial $p(\cdot)$, it holds that if $C_1, \ldots, C_{p(n)} \in \mathcal{F}_n$ then $\text{MAJ}(C_1, \ldots, C_{p(n)}) \in \mathcal{F}_n$.

We now present definitions of functionalities Unroll and Replace which will then allow us to define the appropriate notions of composition and closure for function classes.

Definition 15 (Unroll functionality.). Let $F := \{f_n\}_{n=1}^\infty \in \mathcal{F}$, where $f_n : \{0,1\}^n \rightarrow \{0,1\}$ and $G = \{g_m\}_{m=1}^\infty \in \mathcal{G}$, where $g_m : \{0,1\}^m \rightarrow \{0,1\}^m$, be function families. Also let $t, p$ be polynomials. Let $m \in \text{poly}(n)$. Let $G^F$ denote families functions $f_n : \{0,1\}^n \rightarrow \{0,1\} \in F$ which contains at most $t(n)$ oracle gates computing $g_m : \{0,1\}^m \rightarrow \{0,1\}^m \in G$ and get string of length $p(n)$ as non-uniform advice. On an $n$-bit input, consider the DAG whose left side consists of the circuit of $f_n$ and whose right side consists of circuits $g_{n_1}, \ldots, g_{n(t(n))}$. The values of wires going from the right to the left correspond to the responses $x_1, \ldots, x_{t(n)}$ of lengths $n_1, \ldots, n_{t(n)}$, made in each of the $t(n)$ queries. For $i \in [t(n)]$, circuit $g_{n_i}$ takes as input $x_i$ and returns $y_i$. The values of wires going from the right to the left correspond to the responses $y_1, \ldots, y_{t(n)}$. We say that this DAG, denoted $\text{Unroll}(G^F)$, is an unrolling of $G^F(x)$.

Definition 16 (Replace Functionality.). Consider replacing each $g_{n_i}, i \in [t(n)]$, in $\text{Unroll}(G^F)$ with a circuit $g'_{n_i}$ that takes input $(x_1, \ldots, x_i)$ and produces output $y_i$. This is denoted by $\text{Replace}(\text{Unroll}(G^F), g'_{n_1}, \ldots, g'_{n_{t(n)}})$.

Definition 17 ($(G, t, \ell, u)$-closure of $\mathcal{F}$). Let $F := \{f_n\}_{n=1}^\infty \in \mathcal{F}$, where $f_n : \{0,1\}^n \rightarrow \{0,1\}$ and $G = \{g_m\}_{m=1}^\infty \in \mathcal{G}$, where $g_m : \{0,1\}^m \rightarrow \{0,1\}^m$, be function families. Also let $t, \ell, u$ be polynomials, and $\ell(n) \leq m \leq u(n)$. Let $f_n^{g_m}$ denote function $f_n : \{0,1\}^n \rightarrow \{0,1\}$ which has access to the output of $g_m : \{0,1\}^m \rightarrow \{0,1\}^m$ on at most $t(n)$ inputs of its choice.

We say that $F$ is $(G, t, \ell, u)$-closed under compositions if for every $F \in \mathcal{F}$ such that for all $G \in \mathcal{G}$, $\text{Unroll}(G^F) \in \mathcal{F}$, we have that for all $G' \in \mathcal{G}$ and all $g'_{n_1}, \ldots, g'_{n_{t(n)}} \in G'$, $\text{Replace}(\text{Unroll}(G^F), g'_{n_1}, \ldots, g'_{n_{t(n)}}) \in \mathcal{F}$.

Definition 18 ($(G, t)$-closure of $\mathcal{F}$ under Strong Composition). Let $F := \{f_n\}_{n=1}^\infty \in \mathcal{F}$, where $f_n : \{0,1\}^n \rightarrow \{0,1\}$ and $G = \{g_m\}_{m=1}^\infty \in \mathcal{G}$, where $g_m : \{0,1\}^m \rightarrow \{0,1\}^m$, be function families. Also let $t, p$ be polynomials. Let $m \in \text{poly}(n)$. Let $G^F$ denote families functions $f_n : \{0,1\}^n \rightarrow \{0,1\} \in F$ which contains at most $t(n)$ oracle gates computing $g_m : \{0,1\}^m \rightarrow \{0,1\}^m \in G$.

We say that $F$ is $(G, t)$-closed under compositions if for every $F \in \mathcal{F}$ we have that for all $G, G' \in \mathcal{G}$ and all $g'_{1}, \ldots, g'_{t(n)} \in G'$, $\text{Replace}(\text{Unroll}(G^F), g'_{1}, \ldots, g'_{t(n)}) \in \mathcal{F}$.

2.6 Black Box Reductions

Definition 19 (Black-Box-Reduction). We say $R$ is an $(F, \epsilon, \delta)$-black-box reduction from a (single bit) non-malleable code, $(E, D) = \{(E_n, D_n)\}_{n=1}^\infty$, to a distributional problem, $(\Psi, L) = \{(\Psi_n, L_n)\}_{n=1}^\infty$, if the following hold:

1. For every set of circuits $\{f_n\}_{n=1}^\infty$ parameterized by input length $n$ such that $f_n$ achieves $\epsilon(n)$-malleability, for non-negligible $\epsilon$, i.e.

$$\Pr_{b \in \{0,1\}}[D_n(f_n(E_n))] > \frac{1}{2} + \epsilon(n),$$

then $R^f$ solves $\{(\Psi_n, L_n)\}_{n=1}^\infty$ with advantage $\delta(n)$, where $\delta$ is non-negligible. I.e.

$$\Pr_{x \sim \Psi_n}[L_n(x) = R(f_n)_{n=1}^\infty(x)] > \frac{1}{2} + \delta(n).$$

2. If $\{f_n\}_{n=1}^\infty \in F$ then $R(f_n)_{n=1}^\infty(x) \in F$. 

8
We say a reduction $R$ is length-preserving if, on input of length $n$ is only allowed to make queries to oracles with security parameter $n$. Namely,

$$\Pr_{x \leftarrow \Psi_n} [L_n(x) = R^{f_n}(x)] > \frac{1}{2} + \delta(n).$$

We say a reduction $R$ is approximately length-preserving if there are polynomials $p(\cdot), q(\cdot)$ such that $R$, on input of length $n$ is only allowed to make queries to oracles with security parameter $k \in [p(n), q(n)]$. Namely,

$$\Pr_{x \leftarrow \Psi_n} [L_n(x) = R^{f \{j \leq p(n)}(x)] > \frac{1}{2} + \delta(n).$$

We say a reduction is in $\text{NC}^1$ if it can be written as a family of circuits of $O(\log n)$-depth, $\text{poly}(n)$-size.

### 3 2-Message NMC against $d - 1$ arbitrary errors

In this section, we show that when the message space has size 2 (i.e. single bit messages), non-malleable codes are possible against $d - 1$ arbitrary errors, whereas error correcting codes can tolerate at most $(d - 1)/2$ arbitrary errors. In the next section, we will show that if the message space is increased to 3 or more, then non-malleable codes are impossible even against $d/2$ errors.

The construction is simply a repetition code $(E, D)$. On input a bit $b$, $E$ outputs the string $b^d$ (the bit $b$ repeated $d$ times). On input a string $b_1, \ldots, b_d$, $D$ outputs 1 if there is some $i \in [d]$ such that $b_i = 1$. Otherwise, $D$ outputs 0. Note that this code has distance $d$.

We next prove that $(E, D)$ is a 0-non-malleable code (i.e. the two distributions in the security definition for non-malleable codes–see Definition 2–are identical). Applying Lemma 1, it is sufficient to show that for every tampering function $f$ that modifies at most $d - 1$ symbols,

$$\Pr_{b \leftarrow \{0, 1\}} [D(f(E(b))) = 1 - b] \leq \frac{1}{2},$$

We will use the fact that for the decode algorithm defined above,

$$\Pr[D(f(E(1))) = 0] = 0,$$

since a tampering function that modifies at most $d - 1$ bits cannot flip a 1 codeword to a tampered codeword that decodes to 0 under $D$.

Therefore,

$$\Pr_{b \leftarrow \{0, 1\}} [D(f(E(b))) = 1 - b] = \frac{1}{2} \Pr[D(f(E(0))) = 1] + \frac{1}{2} \Pr[D(f(E(1))) = 0]$$

$$= \frac{1}{2} \Pr[D(f(E(0))) = 1] \leq \frac{1}{2}.$$

This completes the proof.

### 4 Unconditional Negative Results

In this section we demonstrate that non-malleable codes are impossible to construct for 3 different classes. The first impossibility result holds for message spaces of size greater than 2 (which is tight, given the result in Section 3), the second and third impossibility results hold even for a single bit.
4.1 Functions that Modify Half the Symbols

Let \((E, D)\) be a coding scheme with distance \(d\). Define the class of functions \(\mathcal{F}_{d/2-1} = \{f : f\text{ changes }< d/2\text{ codeword symbols}\}\). We know that ECC exist, and thus NMC also exist, for \(\mathcal{F}_{d/2-1}\) (e.g. Reed Solomon Codes achieve this bound).

We now define the slightly larger class \(\mathcal{F}_{d/2} = \{f : f\text{ changes }\leq d/2\text{ symbols}\}\). In theorem 2 we show that even inefficient NMC do not exist for \(\mathcal{F}_{d/2}\).

**Theorem 2.** Let \((E, D)\) be a coding scheme with message space of size greater than 2, alphabet \(\Sigma\) and distance \(d\). Then, for any \(\epsilon > 0\), \((E, D)\) is not a \(1/2 - \epsilon\)-NMC for \(\mathcal{F}_{d/2}\).

**Proof.** We begin with some notation. Given \(\alpha, \beta \in \Sigma^n\), we denote by \(\|\alpha - \beta\|_0\) the number of positions \(i \in [n]\) such that \(\alpha_i \neq \beta_i\).

Let \((E : U \rightarrow V, D : V \rightarrow U)\) be a randomized encoding scheme, where \(U \subseteq \Sigma^k, V \subseteq \Sigma^n\) and \(|U| > 2\).

**Claim.** \(\exists x \in U\) such that \(\forall c_x \in E(x)\) there is a \(z = z(c_x) \in V\):

1. \(\|c_x - z\|_0 \leq d/2\)
2. \(\Pr[D(z) \neq x] \geq 1/2\).

Assuming the claim, consider the following tampering function \(f \in \mathcal{F}_{d/2}\). Let \(z_c\) be the \(z\) for each \(c \in E(x^*)\) guaranteed to exist for some \(x^* \in U\) by the above claim.

\[
f(c) := \begin{cases} z_c & \text{if } c \in E(x^*) \\ c & \text{otherwise} \end{cases}
\]

Let \(\Pr_{c_x \leftarrow E(x^*)}[D(z(c_x^*)) \neq x^*] = p \geq 1/2\). Then, \(\exists y^* \neq x^* \in U\) such that \(\Pr_{c_x \leftarrow E(x^*)}[D(z(c_x^*)) = y^*] \leq \frac{p}{|U|-1}\), but \(\Pr[D(f(E(y^*))) = y^*] = 1\). This means that a distribution \(D^f_x\), that exactly agrees with \(D(f(E(\cdot)))\) on \(x^*\) must output \(x^*\) or \(y^*\) with probability \(1 - p\) and \(y^*\) with probability at most \(\frac{p}{|U|-1}\). A distribution \(D^f_{y^*}\), that exactly agrees with \(D(f(E(\cdot)))\) on \(y^*\) must output \(x^*\) or \(y^*\) with probability \(1\). Thus, any distribution \(D^f\) can only agree with \(D(f(E(\cdot)))\) for both \(x^*\) and \(y^*\) at most \((1 - p) + \frac{p}{|U|-1} \leq 3/4\) fraction of the time (and must have statistical distance at least 1/8 from one of them), since \(p \geq 1/2\) and \(|U| > 2\).

Next we prove the claim.

**Proof.** Suppose for the sake of contradiction that \(\forall x \in U, \exists c_x \in E(x)\) such that \(\forall z \in V\) with \(\|c_x - z\|_0 \leq d/2\) it is the case that \(\Pr[D(z) \neq x] < 1/2\). Take \(x \neq y \in U\) and corresponding \(c_x, c_y\) from above. Then, \(\exists z \in V\) such that \(\|z - c_x\|_0 \leq d/2\) and \(\|z - c_y\|_0 \leq d/2\). But then by assumption it follows that \(\Pr[D(z) = x] > 1/2\) and \(\Pr[D(z) = y] > 1/2\), which is a contradiction because \(x \neq y\).

4.2 Input-Local Functions

We rule out non-malleable codes for input-local functions (see Section 2.3 for formal definition), where each input symbol affects \(\ell\) output symbols and \(\ell\) is the locality parameter. We show that even for \(\ell = 1\), non-malleability is impossible to achieve. The specific tampering functions used in our proof fix all but one of the codeword symbols to constant values. So we can alternately view this result as building on the previous impossibility result: If one allows fixing codeword symbols to constants, then one cannot achieve non-malleability against functions where even a single output symbol’s value depends on the input.

**Theorem 3.** Let \((E, D)\) be a coding scheme with message space of at least 2 and alphabet \(\Sigma\). Then, for any \(\epsilon > 0\), \((E, D)\) is not a \(1/2 - \epsilon\)-NMC for \(\text{Local}_1\).
Proof. Let $U \subseteq \{0,1\}^k$, $V \subseteq \{0,1\}^n$ where $|U| > 1$. Let $(E : U \rightarrow V, D : V \rightarrow U)$ be non-malleable code. Take $x \neq y \in U$. Consider $c_x = E(x), c_y = E(y)$ for some fixed randomness. By correctness $c_x \neq c_y$ and moreover, $D(c_x) \neq D(c_y)$ with probability 1. Also let $d := d(c_x, c_y)$ be the distance between $c_x$ and $c_y$, note that $0 < d \leq n$. Consider $d + 1$ codewords starting with, $c_0 = c_x, c_1, \ldots, c_d = c_y$ where $\forall i \in \{0, \ldots, d - 1\}, d(c_i, c_{i+1}) = 1$. Notice that

$$D(c_0) \neq D(c_d) \iff \exists j \in \{0, \ldots, d - 1\} : D(c_j) \neq D(c_{j+1}).$$

Let $x = D(c_j)$ and let $y = D(c_{j+1})$, where $x \neq y$. Now, consider the following $f \in \text{Local}_1$, $f(c) = \begin{cases} c_j & \text{if } c \in E(y) \\ c_{j+1} & \text{otherwise} \end{cases}$

(Note that all symbols except a single one are constant.) Because they have disjoint support, either $D(f(E(x)))$ or $D(f(E(y)))$ will be at least $1/2$-far from any distribution $D^f$.

### 4.3 Functions with Output Locality $n - \log n$

A function $f : \{0,1\}^n \rightarrow \{0,1\}^n$ is $(n - \log(n))$-output-local if each output bit depends on at most $n - \log(n)$ input bits (see Section 2.3 for formal definition). The particular class $\mathcal{F}'$ that we use in our lower bound proof is a subclass of all $(n - \log(n))$-local tampering functions $\mathcal{F}$. Each $f \in \mathcal{F}'$ has the following structure: First, $f_1, \ldots, f_{n-\log(n)}$ (the functions that output the first $n - \log(n)$ bits) are all the same, except that two different bits from $\{0,1\}$ are hardcoded in each. Second, $f_{n-\log(n)+1}, \ldots, f_n$ are also the same, except that a different value from $\{0,1\}$ is hardcoded in each. Finally, the set of input bits upon which $f_1, \ldots, f_{n-\log(n)}$ depend and the set of input bits upon which $f_{n-\log(n)+1}, \ldots, f_n$ depend are fixed. Taken together, this means that the total number of functions $f$ in $\mathcal{F}$ is at most $4^n \cdot 2^{n-\log(n)}$, so $\log |\mathcal{F}'| = n - \log(n)$. On the other hand, Dziembowski et al. [DPW18] showed existence of a $1/n$-non-malleable code for any class $F$ such that $\log |\mathcal{F}| \leq n - 5 \log(n)$. Thus, our lower bound result is nearly tight matching the existential upper bound.

**Theorem 4.** Let $(E, D)$ be a coding scheme with $E : \{0,1\} \rightarrow \{0,1\}^n$ and $D : \{0,1\}^n \rightarrow \{0,1\}$. Let $\mathcal{F}$ be the class of $(n - \log(n))$-output- local functions. Then $(E, D)$ is $1/4n$-malleable with respect to $\mathcal{F}$.

Note that non-malleable codes whose decode function $D$ may output values in $\{0,1, \bot\}$ *imply* non-malleable codes whose decode function $D$ may only output values in $\{0,1\}$. Thus, ruling out the latter *implies* ruling out the former and only makes our result stronger.

**Proof.** Fix an arbitrary $(E, D)$ with $E : \{0,1\} \rightarrow \{0,1\}^n$ and $D : \{0,1\}^n \rightarrow \{0,1\}$. Our analysis considers two cases and shows that for each case, there exists $f \in \mathcal{F}$ such that

$$\Pr_{b \sim \{0,1\}} [D(f(E(b))) = 1 - b] \geq \frac{1}{2} + 1/4n.$$ 

By Definition 3, this is sufficient to prove Theorem 4.

We begin with some notation and the proceed to the case analysis. For codeword $c = c_1, \ldots, c_n$, let $c^{\text{top}}$ (resp. $c^{\text{bot}}$) denote the first $n - \log n$ bits (resp. last $\log n$ bits) of $c$. I.e. $c^{\text{top}} := c_1, \ldots, c_{n-\log n}$ (resp. $c^{\text{bot}} := c_{n-\log n+1}, \ldots, c_n$). For $t \in \mathbb{N}$, let $S_t$ denote the set of all $t$-bit strings and let $U_t$ denote the uniform distribution over $t$ bits. Let $\epsilon = 1/4n$. Assume $n \geq 2$.

**Case 1:**

$$\Pr_{b \sim \{0,1\}} [D(c^{\text{top}}) \perpendicular b | c \sim E(b), r \sim U_{\log n}] \geq 1/2 + \epsilon.$$

Let $c^{*,0} = c_1^{*,0}, \ldots, c_n^{*,0}$ (resp. $c^{*,1} = c_1^{*,1}, \ldots, c_n^{*,1}$) be the lexicographically first string that decodes to 0 (resp. 1) under $D$ (i.e. $D(c^{*,0}) = 0$ and $D(c^{*,1}) = 1$).
In this case we consider the following distribution over tampering circuits \( f = f_1, \ldots, f_n \), where \( f_i \) outputs the \( i \)-th bit of \( f \):

Sample \( r \leftarrow U_{\log n} \), construct circuits \( f_i \) for each \( i \in [n] \), which take input \( c^{\text{top}} \) and output \( c'_i \). Each \( f_i \) does the following:

- Compute \( d := D(c^{\text{top}}||r) \).
- Output \( c'_i = c^{\ast,1-d}_i \).

We now analyze \( \Pr_{b \leftarrow \{0,1\}} [D(f(E(b))) = 1 - b] \).

\[
\Pr_{b \leftarrow \{0,1\}} [D(f(E(b))) = 1 - b] = \Pr_{b \leftarrow \{0,1\}} [f(E(b)) \text{ outputs } c^{\ast,1-b}_i]
\]

\[
= \Pr_{b \leftarrow \{0,1\}} [D(c^{\text{top}}||r) = b \mid c \leftarrow E(b), r \leftarrow U_{\log n}]
\]

\[
\geq 1/2 + \epsilon
\]

\[
= 1/2 + 1/4n,
\]

where the two equalities follow from the definition of the tampering function \( f \), the first inequality follows since we are in Case 1 and the last inequality follows from the definition of \( \epsilon \). This implies the \( 1/4n \)-malleability of \( (E,D) \).

**Case 2:**

\[
\Pr_{b \leftarrow \{0,1\}} [D(c^{\text{top}}||r) = 1 - b \mid c \leftarrow E(b), r \leftarrow U_{\log n}] \geq 1/2 - \epsilon.
\]

In this case we consider the following distribution over tampering circuits \( f = f_1, \ldots, f_n \), where \( f_i \) outputs the \( i \)-th bit of \( f \):

The first \( n - \log n \) circuits \( (f_1, \ldots, f_{n-\log n}) \) simply compute the identity function: i.e. \( f_i \) for \( i \in [n - \log n] \) takes \( c_i \) as input and produces \( c_i \) as output.

We next describe the distribution over circuits \( f_i \) for \( i \in \{n - \log n + 1, \ldots, n\} \). Sample \( r' \leftarrow [n - 1] \). Construct circuits \( f_i \) for each \( i \in \{n - \log n + 1, \ldots, n\} \) that take input \( c^{\text{bot}} \) and produce output \( c'_i \). Each \( f_i \) does the following:

- Let \( r := r_{n-\log n+1}, \ldots, r_n \) be the \( r' \)-th lexicographic string in the set \( S_{\log n} \setminus \{c^{\text{bot}}\} \).
- Output \( c'_i = r_i \).

We now analyze \( \Pr_{b \leftarrow \{0,1\}} [D(f(E(b))) = 1 - b] \).
Since we are in Case 2 we have that:

\[ 1/2 - \epsilon \leq \Pr_{b\leftarrow\{0,1\}} \left[ D(c^{\text{top}}|r) = 1 - b \mid c \leftarrow E(b), r \leftarrow U_{\log n} \right] \]

\[ = \Pr_{b\leftarrow\{0,1\}} \left[ c^{\text{bot}} = r \mid c \leftarrow E(b), r \leftarrow U_{\log n} \right] \cdot \Pr_{b\leftarrow\{0,1\}} \left[ D(c^{\text{top}}|r) = 1 - b \mid c^{\text{bot}} = r \land c \leftarrow E(b), r \leftarrow U_{\log n} \right] \]

\[ + \Pr_{b\leftarrow\{0,1\}} \left[ c^{\text{bot}} \neq r \mid c \leftarrow E(b), r \leftarrow U_{\log n} \right] \cdot \Pr_{b\leftarrow\{0,1\}} \left[ D(c^{\text{top}}|r) = 1 - b \mid c^{\text{bot}} \neq r \land c \leftarrow E(b), r \leftarrow U_{\log n} \right] \]

\[ = \Pr_{b\leftarrow\{0,1\}} \left[ c^{\text{bot}} = r \mid c \leftarrow E(b), r \leftarrow U_{\log n} \right] \cdot 0 \]

\[ + \Pr_{b\leftarrow\{0,1\}} \left[ c^{\text{bot}} \neq r \mid c \leftarrow E(b), r \leftarrow U_{\log n} \right] \cdot \Pr_{b\leftarrow\{0,1\}} \left[ D(c^{\text{top}}|r) = 1 - b \mid c^{\text{bot}} \neq r \land c \leftarrow E(b), r \leftarrow U_{\log n} \right] \]

\[ = (1 - 1/n) \cdot \Pr_{b\leftarrow\{0,1\}} \left[ D(c^{\text{top}}|r) = 1 - b \mid c^{\text{bot}} \neq r \land c \leftarrow E(b), r \leftarrow U_{\log n} \right] \cdot \Pr_{b\leftarrow\{0,1\}} \left[ D(f(E(b))) = 1 - b \right]. \]

Note that

\[ \Pr_{b\leftarrow\{0,1\}} \left[ D(c^{\text{top}}|r) = 1 - b \mid c^{\text{bot}} \neq r \land c \leftarrow E(b), r \leftarrow U_{\log n} \right] = \Pr_{b\leftarrow\{0,1\}} \left[ D(f(E(b))) = 1 - b \right]. \]

Thus, we have that

\[ 1/2 - \epsilon \leq (1 - 1/n) \Pr_{b\leftarrow\{0,1\}} \left[ D(f(E(b))) = 1 - b \right]. \]

Since for \( \epsilon \leq 1/4n \), we have that

\[ (1/2 + \epsilon) \cdot (1 - 1/n) = 1/2 - 1/2n - \epsilon - \epsilon/n \]

\[ \leq 1/2 - 1/4n = 1/2 - \epsilon, \]

we have that

\[ \Pr_{b\leftarrow\{0,1\}} \left[ D(f(E(b))) = 1 - b \right] \geq \frac{1/2 - \epsilon}{1 - 1/n} \]

\[ \geq 1/2 + \epsilon \]

\[ = 1/2 + 1/4n. \]

This implies the 1/4n-malleability of \((E, D)\). ⁴

5 On NMC via BB Reductions

For the formal definition of a \((F, \epsilon, \delta)\)-black-box reduction from a (single bit) non-malleable code, \((E, D) = \{(E_n, D_n)\}_{n=1}^{\infty}, \) to a distributional problem, \((\Psi, L) = \{(\Psi_n, L_n)\}_{n=1}^{\infty}, \) see Definition 19 in Section 2.6.

We begin with a useful definition.

⁴ The same proof for \(1/8\sqrt{n}\)-NMC gives a tighter upper bound of \(\log \log |F| \leq n - 2\log(n)\). By changing \( \epsilon = 1/4n \) to \( \epsilon = 1/4\sqrt{n} \) while keeping the lower bound same (\(\log \log |F| \leq n - \log(n)\)).
Definition 20 (Look-Up Circuit.). A look-up circuit with \( p(n) \)-bit keys and values and \( \ell(n) \) inputs has values \( y_1, \ldots, y_{\ell(n)} \) hardwired and gets as input \( x_1, \ldots, x_{\ell(n)} \), where each \( x_i \) and \( y_i \) is \( p(n) \) bits. The circuit finds the first \( i \in [\ell(n)] \) such that \( x_{\ell(n)} \) is equal to \( x_i \) and outputs hardcoded value \( y_i \).

Proposition 1. For \( p(n), \ell(n) = O(n^c) \) for some fixed constant \( c \), there exist polynomial size look-up circuits of depth \( O(\log n) \).

Proof (Sketch). The inputs, \( x_1, \ldots, x_{\ell-1} \), can be put in sorted order via a circuit of size \( O(n^c \log n) \) and depth \( O(\log n) \) [AKS83]. Then each sorted \( x_i \) can determine if it is the first of that value (if \( x_1, \ldots, x_{\ell-1} \) are in sorted order then \( x_j \) is determining that there does not exist \( x_i = x_j \) such that \( i < j \), by comparing only to one neighboring value. This can be done in parallel. Finally, compare \( x_\ell \) to all \( x_i \) that pass this test in parallel. If there is such an \( x_i \) such that \( x_i = x_\ell \), the circuit will output \( y_i \). Otherwise, the circuit will output \( y_\ell \).

We now present the central technical lemma of the section.

Lemma 2. Assume that \( \mathcal{F} \) is \( (\mathcal{F}, t(n), p(n)) \)-closed under composition (see Definition 17), and contains look-up circuits with \( p(n) \)-bit keys and values and \( t(n) \) inputs, for polynomials \( t(\cdot), p(\cdot) \). If there is an \( (\mathcal{F}, 1/2, \delta(n)) \)-black-box-reduction making \( t(n) \) security parameter-preserving queries from a \( (\text{single bit}) \) non-malleable code for \( \mathcal{F} \), \( (\mathcal{E}, \mathcal{D}) = \{(\mathcal{E}_n, \mathcal{D}_n)\}_{n=1}^{\infty} \), to a distributional problem, \((\Psi, L) = \{\{= \Psi_n, L_n\}\}_{n=1}^{\infty} \), then one of the following must hold:

1. \( (\mathcal{E}, \mathcal{D}) \) is \( \frac{\delta(n)}{2H(n)} \)-malleable by \( \mathcal{F} \).
2. \((\Psi, L)\) is \( \frac{\delta(n)}{2} \)-easy for \( \mathcal{F} \).

Moreover, if \( (\mathcal{E}, \mathcal{D}) \) is efficient, then it suffices that \( \mathcal{F} \) contains such look-up circuits generated in uniform polynomial time.

Proof. Let \( R \) be such a security parameter-preserving \( (\mathcal{F}, 1/2, \delta(n)) \)-reduction, for a non-malleable code \((\mathcal{E}, \mathcal{D})\) and distributional problem \((\Psi, L)\). Moreover, for security parameter \( n \), let \( p(n) \) be the length of the codeword generated by \( \mathcal{E}_n \), where \( p(\cdot) \) is a polynomial.

Consider the following tampering functions \( \{f_{p(n)}\}_{p(n)} \) whose behavior on a given codeword \( c \) is defined as follows (where \( H \) is a random function \( H : \{0, 1\}^{p(n)} \rightarrow \{0, 1\}^* \)):

\[
f_{p(n)}(c) := \begin{cases} E_n(1; H(c)) & \text{if } D_n(c) = 0 \\ E_n(0; H(c)) & \text{if } D_n(c) = 1 \end{cases}
\]

Since, NMC are perfectly correct, we have (for any choice of \( H \))

\[
\Pr_{b \leftarrow \{0,1\}}[D_n(f_{p(n)}(E(b))) = 1 - b] = 1.
\]

Therefore, by our assumption on \( R \) we have that for all \( n \),

\[
\Pr_{x \leftarrow \Psi_n}[L_n(x) = R^{f_{p(n)}(x)}] \geq \frac{1}{2} + \delta(n).
\]

Now, for the \( j \)-th oracle query, we define \( f_{p(n)}^{j}(x) \), a stateful simulation of the output of the tampering function \( f_{p(n)} \) on the \( j \)-th query. Each \( f_{p(n)}^{j} \) is a lookup circuit with \( p(n) \)-bit keys and values and \( j \) inputs that hardcoded a random codeword (sampled from \( \mathcal{E}(b) \)) as \( b \) uniform as the \( y_j \) value.

By our assumption on \( \mathcal{F} \) (and \( R \)), we have that \( \text{Replace(}\text{Unroll}(R^{f_{p(n)}}, f_{p(n)}^{1}, \ldots, f_{p(n)}^{j}), \ldots) \in \mathcal{F} \). We will abuse notation and denote the resulting circuit by \( R^{f_{p(n)}(x)} \).

\( ^5 p(n) \) corresponds to the length of the codeword outputted by \( E_n \).
is close that of $R^H_{\mathcal{F}}(x)$, for any $x$, which will imply that $R^H_{\mathcal{F}(n)}(x) \in \mathcal{F}$ breaks the distributional problem w.h.p., since $R^H_{\mathcal{F}(n)}(x)$ does. More accurately, if $(\mathcal{E}, \mathcal{D})$ is $\frac{\delta(n)}{2t(n)}$-non-malleable by $\mathcal{F}$, then we will show that

$$\forall n \in \mathbb{N}, \forall x \in \{0, 1\}^n, \Delta(R^H_{\mathcal{F}(n)}(x); R^H_{\mathcal{F}(x)}(x)) \leq \frac{\delta(n)}{2t(n)}.$$ 

By the above, this then implies that $(\mathcal{E}, L)$ is $\frac{\delta(n)}{2}$-easy for $\mathcal{F}$.

To show that the outputs of $R^H_{\mathcal{F}(n)}(x)$ and $R^H_{\mathcal{F}(n)}(x)$ are close, we will use a hybrid argument, reducing to the $\frac{\delta(n)}{2t(n)}$-non-malleability of $(\mathcal{E}, D)$ at every step.

In the $i$-th hybrid, the function $f_{p(n)}^{(i)}$ responding to the $j$-th query is a look-up circuit with with $p(n)$-bit keys and values and $j$ inputs that hardcodes values $y_1^j, \ldots, y^j$. For $k \in [t = t(n)]$, the $y^j_k$ values are sampled as follows: For $k \in [t - i]$, $y^j_k$ is sampled as by $f_{p(n)}^H$. For $k > t - i$, $y^j_k$ is a random encoding of a random bit. The concatenation of the $t$ circuits for each query is denoted by $f_{p(n)}^{(i)}$. Clearly, $f^{(0)}_{p(n)} = f_{p(n)}^H$ and $f^{(i)}_{p(n)} = f^{(i+1)}_{p(n)}$.

We will show that for all $x \in \{0, 1\}^n$ (and any fixing of random coins $r$ for $R$) $\Delta(R^H_{\mathcal{F}(n)}(x); R^H_{\mathcal{F}(n)}(x)) \leq \epsilon(n)$ (for $i \in [t(n)]$), which proves the claim above.$(R^H_{\mathcal{F}(n)}(x)$ has advantage $\delta(n)$ and in each of the subsequent $t(n)$ hybrids we lose at most an $\epsilon(n)$ factor.)

Suppose not, then there exists an $x$ (and random coins $r$, if $R$ is randomized) such that $R$’s behavior differs with respect to $f_{p(n)}^{(i)}$ and $f_{p(n)}^{(i-1)}$: $|\Pr[R^H_{\mathcal{F}(n)}(x) = 1] - \Pr[R^H_{\mathcal{F}(n)}(x) = 1]| \geq \frac{\delta(n)}{2t(n)}$.

Note that for fixed random function $H$ (that generates the random coins used to sample the $y_j$ values) $f_{p(n)}^{(i)}$ and $f_{p(n)}^{(i-1)}$ differ solely on the response to $(t - i)$-th query. So, fix $x$, $H$ and all but the $(t - i)$-th value $y^i_{t-i}$ and “hardcode” all other $y_k$ values in both cases. The reason that we can hardcode the $y_j$ values except for the $(t - i)$-th response is the following: Clearly, up to the $(t - i)$-th query, the responses can be fully hardcoded since $x$ is fixed and so all the queries and responses can also be fixed. The $y_j$ values hardcoded in the $(t - i + 1)$-st lookup circuit and on can also be fixed, since in both $f_{p(n)}^{(i)}$ and $f_{p(n)}^{(i-1)}$ the $(t - i + 1)$-st value of $y_j$ and on is a random codeword, that does not depend on the value encoded in the query submitted by the reduction. Let $s_{H,x}$ denote the value encoded in the $(t - i)$-th query in this hardcoded variant of the hybrid. Note that the value of $s_{H,x}$ is also fixed.

1. In $R^H_{\mathcal{F}(n)}(x)$ all values up to the $(t - i)$-th response are hardcoded. The $(t - i)$-th response, which will be a random encoding of bit $1 - s_{H,x}$, is not hardcoded. All the other responses are computed by lookup circuits with hardwired $y_j$ values.
2. In $R^H_{\mathcal{F}(n)}(x)$, all values up to the $(t - i)$-th response are hardcoded. The $(t - i)$-th response, which will be a random encoding of a random bit, is not hardcoded. All the other responses are computed by lookup circuits with hardwired $y_j$ values.

Thus, we will treat the above as a new function $R'_{H,x}(\cdot)$ that takes as input just the response to the $(t - i)$-th query and returns some value. Note that $R'_{H,x}(\cdot)$ is in $\mathcal{F}$, since it can be viewed as the circuit $R^H_{\mathcal{F}(n)}$ with queries/responses to $f^{(i)}$, $j \in [t - i + 1]$ hardcoded, the $(t - i)$-th query hardcoded, the $(t - i)$-th value $y^i_{t-i}$ as the input to the circuit, and for $j > t - i$, the $f^{(i)}$ functions as lookup circuits contained in $\mathcal{F}$. Moreover, by the above, $R'_{H,x}(\cdot)$ distinguishes random codewords that encode the bit $1 - s_{H,x}$ from random codewords that encode a random bit with advantage $\epsilon(n)$. Specifically,

$$\Pr[R'_{H,x}(c) = 1 | c \leftarrow E_n(1 - s_{H,x})] - \Pr[R'_{H,x}(c) = 1 | c \leftarrow E_n(b), b \leftarrow \{0, 1\}] \geq \frac{\delta(n)}{2t(n)}.$$ 

By standard manipulation, the above is equivalent to:

$$\frac{1}{2} \cdot \Pr[R'_{H,x}(c) = 1 | c \leftarrow E_n(1 - s_{H,x})] + \frac{1}{2} \cdot \Pr[R'_{H,x}(c) = 0 | c \leftarrow E_n(s_{H,x})] \geq \frac{1}{2} + \frac{\delta(n)}{2t(n)}.$$ 

15
This implies that we can use $R'_{H,x}$ to construct a distribution over tampering functions in $\mathcal{F}$ that successfully break $(E, D)$. Details follows.

Let $c_{s_{H,x}}$ be a codeword encoding bit $s_{H,x}$ and let $c_{1-s_{H,x}}$ be a codeword encoding bit $1-s_{H,x}$. Define $\hat{f}_{H,x}$ as follows: on input (codeword) $c$,

- If $R'_{H,x}(c) = 1$, output $c_{s_{H,x}}$;
- Otherwise, output $c_{1-s_{H,x}}$.

We now analyze

\[
\Pr_{b \leftarrow \{0,1\}} [D_n(\hat{f}_{H,x}(E_n(b))) = 1 - b] = \Pr[b = 1 - s_{H,x}] \cdot \Pr[R'_{H,x}(c) = 1 | c \leftarrow E_n(1 - s_{H,x})] \\
+ \Pr[b = s_{H,x}] \cdot \Pr[R'_{H,x}(c) = 0 | c \leftarrow E_n(s_{H,x})] \\
= \frac{1}{2} \cdot \Pr[R'_{H,x}(c) = 1 | c \leftarrow E_n(1 - s_{H,x})] \\
+ \frac{1}{2} \cdot \Pr[R'_{H,x}(c) = 0 | c \leftarrow E_n(s_{H,x})] \\
\geq \frac{1}{2} + \frac{\delta(n)}{2^t(n)}.
\]

But, the above implies that $(E, D)$ is $\frac{\delta(n)}{2^t(n)}$-malleable for $\mathcal{F}$.

Therefore, we conclude that either $(E, D)$ is $\frac{\delta(n)}{2^t(n)}$-malleable for $\mathcal{F}$ or the distributional problem, $(\Psi, L) = \{(\Psi_n, L_n)\}_{n=1}^{\infty}$ is $(\delta(n)/2)$-easy for $\mathcal{F}$.

The following corollary holds since $\text{NC}^1$ is $(\text{NC}^1, t, p(n), p(n))$-closed under composition (for all polynomials $p(\cdot)$), and $\text{NC}^1$ contains lookup circuits with $p(n)$-bit keys and values, for any polynomial $p(\cdot)$.

**Corollary 1.** If there is an $(\text{NC}^1, 1/2, \delta(n))$-black-box-reduction making $t(n)$ security parameter preserving queries from a (single bit) non-malleable code for $\text{NC}^1$, $(E, D) = \{(E_n, D_n)\}_{n=1}^{\infty}$, to a distributional problem, $(\Psi, L) = \{(\Psi_n, L_n)\}_{n=1}^{\infty}$, then one of the following must hold:

1. $(E, D)$ is $\frac{\delta(n)}{2^t(n)}$-malleable by $\text{NC}^1$.
2. $(\Psi, L)$ is $(\delta(n)/2)$-easy for $\text{NC}^1$.

**Note 1.** The proof of Lemma 2 (as well as the other proofs in this section), does not extend to cases in which the reduction $R$ is outside the class of tampering functions $\mathcal{F}$. Specifically, in the hybrid arguments, we require that $R'_{H,x}(\cdot)$ is in $\mathcal{F}$. In particular, our proof approach does not extend to proving impossibility of constructing a (single bit) non-malleable code for $\mathcal{F}$, from a distributional problem, $(\Psi, L)$ that is hard for some larger class $\mathcal{F}$. E.g., our techniques do not allow us to rule out constructions of non-malleable codes for $\text{NC}^1$ from a distributional problem that is hard for $\text{NC}^2$. Our techniques also do not rule out constructions of non-malleable codes for $\mathcal{F}$ from an “incompressibility”-type assumption, such as those used in the recent work of [BDK+19]. Briefly, if a function $\psi$ is incompressible by circuit class $\mathcal{F}$, it means that for $t \ll n$, for any computationally unbounded Boolean function $D : \{0,1\}^t \rightarrow \{0,1\}$ and any $F : \{0,1\}^n \rightarrow \{0,1\}^t \in \mathcal{F}$, the output of $D \circ F(x_1, \ldots, x_n)$ is uncorrelated with $\psi(x_1, \ldots, x_n)$ (over uniform choice of $x_1, \ldots, x_n$). In our case, this would mean that the reduction $R$ is allowed oracle access to a computationally unbounded Boolean function $D$, since the hardness assumption would still be broken by the reduction as long as $R \in \mathcal{F}$ and the query made to $D$ has length $t \ll n$. Since $R$ composed with $D$ is clearly outside the tampering class $\mathcal{F}$, our proof approach does not apply in the incompressibility setting.
Lemma 4. Let conflate the security parameter queried to the oracle with the length of the query made to the oracle. In this case, we can, WLOG, conflate the security parameter queried to the oracle with the length of the query made to the oracle. However, we now require for our proof that \( \mathcal{F} \) is \( (\mathcal{F}, \ell, \ell, u) \)-closed under composition and contains look-up circuits with \( \ell(n) \) to \( u(n) \)-bit keys and values and \( t(n) \) inputs, for polynomials \( \ell(\cdot), \ell(\cdot), u(\cdot) \).

Lemma 3. Assume \( \mathcal{F} \) is \( (\mathcal{F}, \ell, \ell, u) \)-closed under composition (see Definition 17) and contains look-up circuits with \( p(\ell) \)-bit keys and values and \( t(n) \) inputs, for polynomials \( \ell(\cdot), p(\cdot) \). If there is an \( (\mathcal{F}, 1/2, \delta(n)) \)-black-box-reduction making \( t(n) \) number of \( (\ell(n), u(n)) \)-approximately length preserving queries, from a (single bit) non-malleable code for \( \mathcal{F} \), \( (E, D) = \{(E_n, D_n)\}_n \), to a distributional problem, \( (\Psi, L) = \{(\Psi_n, L_n)\}_n \), then one of the following must hold:

1. \( (E, D) \) is \( \frac{\delta(n)}{2^{t(n)}} \)-malleable by \( \mathcal{F} \).
2. \( (\Psi, L) \) is \( \frac{\delta(n)}{t(n)^2} \)-easy for \( \mathcal{F} \).

Moreover, if \( (E, D) \) is efficient, then for the conclusion to hold it suffices that \( \mathcal{F} \) contains such look-up circuits generated that are generated uniform polynomial time.

The following corollary holds since \( \text{NC}^1 \) is \( (\text{NC}^1, \ell, u) \)-closed under composition, where \( \ell(n) = n^\gamma \), for any constant \( \gamma \leq 1 \), \( u(n) = n^c \), for any constant \( c \geq 1 \) and \( \text{NC}^1 \) contains look-up circuits with \( \ell(n) \) to \( u(n) \)-bit keys and values and \( t(n) \) inputs, for polynomials \( \ell(\cdot), \ell(\cdot), u(\cdot) \).

Corollary 2. Fix constants \( \gamma \leq 1 \), \( c \geq 1 \). If there is an \( (\text{NC}^1, 1/2, \delta(n)) \)-black-box-reduction making \( t(n) \) \( (n^\gamma, n^c) \)-approximately length preserving queries from a (single bit) non-malleable code for \( \text{NC}^1 \), \( (E, D) = \{(E_n, D_n)\}_n \), to a distributional problem, \( (\Psi, L) = \{(\Psi_n, L_n)\}_n \), then one of the following must hold:

1. \( (E, D) \) is \( \frac{\delta(n)}{2^{t(n)}} \)-malleable by \( \text{NC}^1 \).
2. \( (\Psi, L) \) is \( \frac{\delta(n)}{t(n)^2} \)-easy for \( \text{NC}^1 \).

We extend to non-security parameter preserving reductions, but require a stronger compositional property for the tampering class \( \mathcal{F} \). As for approximate security parameter preserving reductions, WLOG we may conflate the security parameter queried to the oracle with the length of the query made to the oracle.

Lemma 4. Let \( \mathcal{F} \) be closed under strong composition (see Definition 18) and contain dictionaries of size \( t(n) \) with keys and values of length at most \( u(n) \). If for every non-negligible \( \epsilon \), there is an \( (\mathcal{F}, \epsilon, \delta(n)) \)-black-box-reduction (for some non-negligible \( \delta \)) making \( t(n) \) queries from an (single bit) \( \epsilon(n) \)-non-malleable code for \( \mathcal{F} \), \( (E, D) = \{(E_n, D_n)\}_n \), to a distributional problem, \( (\Psi, L) = \{(\Psi_n, L_n)\}_n \), then \( (\Psi, L) \) is not \( (\delta(n) - t(n) \cdot \epsilon(n)) \)-hard for \( \mathcal{F} \).

Proof. Let \( S := \{1, 2^1, 2^2, 2^2, \ldots\} \). Let \( \epsilon(n) \) be the following non-negligible function:

\[
\epsilon(n) := \begin{cases} 
\frac{1}{n} & \text{if } n \in S \\
0 & \text{if } n \notin S
\end{cases}
\]

Assume there is some reduction \( R \) that succeeds with non-negligible probability \( \delta := \delta(n) \) for this \( \epsilon \). Since \( \delta \) is non-negligible, there must be an infinite set \( S' \) such that \( \delta(n) \geq 1/n^c \) for some constant \( c \) and for all \( n \in S' \).

WLOG, we may assume that the reduction \( R \), on input of length \( n \), queries at most a single input length \( \ell(n) \in \omega(\log(n)) \), whereas all other queries are of input length \( O(\log(n)) \) (since we may assume the oracle simply returns strings of all 0’s on any input of length \( k \notin S \)). Additionally, we may assume that (1) \( \ell(n) \) is polynomial in \( n \) (since otherwise the reduction does not have time to even write down the query) and (2) for
any \( k \in \mathbb{N} \), the size of the set \( \ell^{-1}(k) \cap \mathcal{S}' \) is finite (otherwise we can hardcode all possible query/responses for a particular input length \( k \) into the reduction—which is constant size since \( k \) is constant—and obtain a circuit that breaks the underlying hard problem on an infinite number of input lengths). Moreover, we assume \( \text{WLOG} \) that \( \ell(n) < n \), since otherwise our previous proof holds.

Since by assumption \( \mathcal{F} \) is HCS-amenable, it means that Impagliazzo’s hardcore set holds for adversaries in \( \mathcal{F} \). Specifically, for random codewords \( c \leftarrow \mathcal{E}_{\ell(n)}(b), b \leftarrow \{0, 1\} \) of length \( \ell = \ell(n) \) s.t. \( \ell(n) < n \), there are two possible cases:

1. For infinitely many \( n \in \mathcal{S}' \) (this set of values is denoted by \( \mathcal{S}'' \subseteq \mathcal{S}' \)), there is some adversary in \( \mathcal{F}_n \) that outputs \( \mathcal{D}_{\ell(n)}(c) \) with probability at least 3/4.

2. For infinitely many \( n \in \mathcal{S}' \) (this set of values is denoted by \( \mathcal{S}'' \subseteq \mathcal{S}' \)), there is some hardcore set \( \mathcal{H} \) of size at least \( \epsilon'(n) \cdot 2^k \), where \( \epsilon'(n) = \frac{1}{2^{\ell(n)}} \) such that every adversary in \( \mathcal{F}_n \) outputs \( \mathcal{D}_{\ell(n)}(c) \) with probability at most \( 1/2 + \epsilon'(n) \), when \( c \) is chosen at random from \( \mathcal{H} \).

In Case 1, we set the tampering function \( \{ f_k \}_k \) to use the circuit described above to decode a random codeword with prob 3/4 and then chooses a random encoding of 0 or 1 appropriately. Additionally, \( f_k \) only responds if \( k \in \mathcal{S} \). Clearly, \( f_k \) succeeds with non-negligible probability \( \epsilon \). Since the \( \mathcal{F} \) function remains the same, we know that \( \delta \) and \( \ell, \mathcal{S}, \mathcal{S}' \) remain the same.

In this case, as in the previous proof, we can switch to a simulated tampering function \( \text{Sim} \), which responds with \( f_{\ell(n)} \) on query input length \( \ell(n) \) and hardcodes all responses for all possible queries \( R \) makes to \( f_k \) with input lengths \( k = k(n) \in O(\log(n)) \).

Note that since we are in Case 1, for infinitely many input lengths—input lengths \( n \in \mathcal{S}'' \)—to \( R, R^\text{Sim} \), is a circuit in \( \mathcal{F}_n \), since \( \mathcal{F}_n \) strongly composes. Additionally, the behavior of \( R^\text{Sim} \) is identical to the behavior of \( R(f_k)_k \). Moreover, since \( f_k \) succeeds with non-negligible \( \epsilon \), by assumption on \( R \), it means that for all \( n \in \mathcal{S}', R^\text{Sim} \) agrees with \( (\Psi, L) \) with probability 1/2 + 1/n\(^\epsilon \). But then we must have that for infinitely many \( n \in \mathcal{S}' \)—input lengths \( n \in \mathcal{S}'' \), \( R^\text{Sim} \) agrees with \( (\Psi, L) \) with probability 1/2 + 1/n\(^\epsilon \) and \( R^\text{Sim} \in \mathcal{F}_n \). So \( (\Psi, L) \) is \((\delta'(n))\)-easy for \( \mathcal{F} \), where

\[ \delta'(n) := \begin{cases} \frac{1}{n} & \text{if } n \in \mathcal{S}'' \\ 0 & \text{if } n \notin \mathcal{S}'' \end{cases} \]

In Case 2, we set the tampering function \( \{ f_k \}_k \) to decode the query submitted by the reduction \( R \) and respond with a random encoding from the hardcore set described above (if it exists), which decides to 0 or 1 as appropriate. Specifically, the hardcore set \( \mathcal{H} \) is defined as follows: \( f_k \) sets \( n^* \) to be equal to the lexicographically first element in the (finite) set \( \ell^{-1}(k) \cap \mathcal{S}'' \), and chooses the lexicographically first set \( \mathcal{H} \) of size \( \epsilon(n^*) \cdot 2^\ell(n^*) \) for which every adversary in \( \mathcal{F}_n \) outputs \( \mathcal{D}_{\ell(n^*)}(c) \) with probability at most \( 1/2 + \epsilon(n^*) \), when \( c \) is chosen at random from \( \mathcal{H} \). If \( \ell^{-1}(k) \cap \mathcal{S}' = \emptyset \) or there is no such hardcore set \( \mathcal{H} \), then \( f_k \) applies the trivial breaking strategy described above (decoding the input and responding with a random encoding of 0 or 1 as appropriate). Moreover, \( f_k \) responds only if \( k \in \mathcal{S} \). Since \( \delta \) function remains the same in this case as well, the \( \delta \) function also remains the same. Thus, for \( n \in \mathcal{S}' \), \( R^\text{Sim} \) must still agree with \( (\Psi, L) \) with probability 1/2 + 1/n\(^\epsilon \).

In this case, as in the previous proof, we can switch to a simulated tampering function \( \text{Sim} \) that does not decode but rather chooses a random codeword from the hardcore set \( \mathcal{H} \) (which again we can hardcode in using lookup circuits as before). Moreover, for queries \( R \) makes to \( \text{Sim} \) with input lengths \( k = k(n) \in O(\log(n)) \), all responses for all possible queries \( c \) are hardcoded into \( \text{Sim} \). Now, for infinitely many \( n \in \mathcal{S}' \)—input lengths \( n \in \mathcal{S}'' \)—\( R \)'s behavior should be \( t(n) \cdot \epsilon(n) \)-close when interacting with \( \{ f_k \}_k \) versus \( \text{Sim} \), since otherwise in each hybrid step we can construct a distinguishing circuit in \( \mathcal{F}_n \) (as in the previous proof) contradicting the guaranteed hardness of the hardcore set. Finally, we must argue that for infinitely many \( n \in \mathcal{S}' \)—input lengths

---

6. Note that \( \mathcal{D}_{\ell(n)}(c) \) takes inputs of length \( \ell(n) \), whereas \( \mathcal{F}_n \) takes inputs of length \( n \). We can easily resolve this discrepancy by padding inputs of length \( \ell(n) \) up to \( n \).

7. Again, the input \( c \) to \( \mathcal{D}_{\ell(n)} \) has length \( \ell(n) \) while \( \mathcal{F}_n \) takes inputs of length \( n \). As above, we resolve the discrepancy by padding inputs of length \( \ell(n) \) up to \( n \).

8. Note that it is finite since \( \ell^{-1}(k) \cap \mathcal{S}' \) is finite and \( \mathcal{S}'' \subseteq \mathcal{S}' \).
n ∈ S''−R composed with Sim is in the class F. But due to the fact that F is (F, t)-closed under strong composition, this occurs whenever the reduction is instantiated with security parameter n ∈ S'', where n is the lexicographically first element in the set ℓ−1(ℓ(n)) ∩ S''. Since n is always contained in ℓ−1(ℓ(n)), since the size of ℓ−1(ℓ(n)) ∩ S' is finite and since the size of S'' is infinite, there will be infinitely many n ∈ S'' for which this event occurs. Thus, for infinitely many n ∈ S'' (denote this set of values by ˜S, R(fn)k agrees with (Ψ, L) with probability 1/2 + 1/nε and RSim is t(n) · ε(n) ≤ 1/2nε-close to R(fn)k. So we conclude that (Ψ, L) is (δ′(n))-easy for F, where

\[ δ′(n) := \begin{cases} \frac{1}{n^\nu} & \text{if } n \in ˜S \\ 0 & \text{if } n \notin ˜S \end{cases} \]

The following corollary holds since NC is (NC, t)-closed under strong composition and Impagliazzo’s HCS holds for NC.

**Corollary 3.** If for every non-negligible \( \epsilon = \epsilon(\cdot) \), there is an (nu − NC, ε, δ)-black-box-reduction, for some non-negligible \( \delta = \delta(\cdot) \), making t(n) queries from a (single bit) non-malleable code for nu − NC, \((E, D) = \{(E_n, D_n)\}_{n=1}^\infty \), to a distributional problem, \((Ψ, L) = \{(Ψ_n, L_n)\}_{n=1}^\infty \), then \((Ψ, L)\) is \((δ′(n))-easy for NC, for some non-negligible \( \delta′ = δ′(\cdot) \).

## Acknowledgments

The first and fourth authors are supported in part by the Leona M. & Harry B. Helmsley Charitable Trust. The first author is additionally supported in part by an IBM Research PhD Fellowship. The second and third authors are supported in part by NSF grants #CNS-1840893, #CNS-1453045 (CAREER), by a research partnership award from Cisco and by financial assistance award 70NANB15H328 from the U.S. Department of Commerce, National Institute of Standards and Technology. This work was performed, in part, while the first author was visiting IDC Herzliya’s FACT center and supported in part by ISF grant no. 1790/13 and the Check Point Institute for Information Security.

## References


