Limits to Non-Malleability

Marshall Ball\textsuperscript{1}, Dana Dachman-Soled\textsuperscript{2}, Mukul Kulkarni\textsuperscript{2}, and Tal Malkin\textsuperscript{1}

\textsuperscript{1} Columbia University
{marshall,tal}@cs.columbia.edu
\textsuperscript{2} University of Maryland
danadach@ece.umd.edu, mukul@umd.edu

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 symbols is a function of $n - \log n$ input symbols.

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:

Let $f \in F$ denote a tampering function.

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. Additionally, we rule out
constructions of NMC for a wide range of complexity classes with security reductions that are simply 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, we consider a stronger setting of message spaces of size two or three.

1.1 Strictly Impossible

We identify 3 tampering regimes where achieving 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).

Given the above, we fully resolve our question, showing that for message space of size 2, it is possible to obtain non-malleable codes that tolerate arbitrary modification of \(d−1\) symbols (via a repetition code, refer 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 even \(d/2\) 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 \cite{BDKM16} constructed efficient NMC for tampering functions with locality $n'$, 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 \cite{DPW18}. A natural question is how large can the output-locality be?

In the current work, we prove the impossibility of non-malleable codes for the class of $(n - \log n)$-output-local tampering functions. Note that in addition to the above motivation, parity over $n$ bits is average-case hard for this class.\footnote{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. \cite{BdW02}} 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 yet cannot construct non-malleable codes for. Furthermore, the class $\mathcal{F}'$ constructed in our lower bound proof in fact has size $4^n \cdot 2^{2^{n - \log(n)}}$, which in turn means that $\log \log |\mathcal{F}'| = n - \log(n)$. On the other hand, the aforementioned result of Dziembowski et al. \cite{DPW18} shows that there exists 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.

Note that the lower bound for tampering class of input-local requires that the coding scheme has perfect correctness and deterministic decoding. Whereas, for tampering class of $n - \log n$ output-local function lower bound is applicable for coding schemes with perfect correctness and randomized decoding. For remaining result, lower bound is applicable even with imperfect correctness of the coding scheme with randomized decoding.

### 1.2 Impossibility of Black-Box Security Reductions.

In recent work, unconditional constructions of non-malleable codes for progressively larger tampering classes, such as $\text{NC}^0$ \cite{BDKM16, BDG18, CL17a} and $\text{AC}^0$ \cite{CL17a, BDG18}, have been presented. In fact, the construction of \cite{BDG18} remains secure for circuit depths as large as $\Theta(\log(n)/\log \log(n))$. Moreover, due to the impossibility of efficient NMC for all of $\text{P}$, extending their result to obtain unconditional NMC for circuits with asymptotically larger depth would require separating $\text{P}$ from $\text{NC}^1$, 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 $\text{P}$.

The above question was partially addressed by Ball et al. \cite{BDKM18, BDSK+18} 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 $\text{P}$. Moreover, such distributional problems for $\text{NC}^1$ can be based on worst-case assumptions (see further discussion in Section 5).

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 $\text{P}$. Note that, as observed above, this assumption is minimal, since the decoding function of a 1-bit non-malleable code for $\text{NC}^1$ with respect to the distribution formed by encoding a random 1 bit message yields a distributional problem with the above properties.

In the current work, 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 $(\mathcal{E}, \mathcal{D})$ for a complexity class $\mathcal{F}$ to a distributional problem $(D, L)$ 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 $(D, L)$. At a high level (skipping some technical details), we require two properties of a black-box reduction $R$ from $(E, D)$ to $(D, L)$:

- First, whenever the tampering function $f$ succeeds in breaking the non-malleable code, the reduction should succeed, regardless of whether $f \in \mathcal{F}$. This represents the fact that $R$ uses $f$ in a black-box manner.
- Second, for any $f \in \mathcal{F}$, $R^f$ 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 $R^f \in \mathcal{F}$. 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 16, 17 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. In particular, our results on security parameter preserving and approximate security parameter preserving reductions apply to the class $\text{NC}^1$ as a special case. Our result ruling out non-security parameter preserving reductions applies to the class (non-uniform) $\text{NC}$ as a special case. See Corollaries 1, 2, 3 for the formal statements.

For simplicity we assume perfect correctness and deterministic decoding, but we expect the results 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 constructing NMC secure for all polynomial tampering functions. Since then NMC have been studied in the information-theoretic as well as computational settings. Liu and Lysyanskaya [LL12] introduced the split-state classes of tampering functions and constructed computationally secure NMC for split-state tampering. This subsequently received a lot of attention in both computational [AAG+16] as well as information theoretic setting with a series of advances: achieving reduced number of states, improved rate, or adding desirable features to the scheme [DKO13, ADL14, CZ14, ADKO15a, AG+15b, AAG+16, KOS17, Li18]. Recently efficient NMC have been constructed for tampering function classes other than split-state tampering [BDKM16, AG+15a, CL17a, FHMV17, BDKM18, BDG+18, BDSK18]. Those results include both computational as well as information-theoretically secure constructions. Additionally, [DPW18, CG14a, FMVW14] also 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, FMNV15, ADKO15a, CRK16, CGM+16, KLT16, DKS17, DNO17, ADN+17, DKS18, KOS18, OPVV18, 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 Gurusswami [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 \( \varepsilon \in O(1/poly(n)) \).

Dachman-Soled, Kulkarni, and Shahverdi [DKS17] studied the bounds on the locality 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, Pas11, GW11, AGO11, Seu12, BM09, FKPR14]) 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 some cryptographic primitive is impossible under “standard assumptions” (e.g. falsifiable assumptions or non-interactive assumptions).

2 Preliminaries

2.1 Notation

Firstly, we present some standard notations that will be used in what follows. For any positive integer \( n \), \( [n] := \{1, \ldots, n\} \). For a vector \( x \in \chi \) of length \( n \), we denote its hamming weight by \( \|x\| := |\{x_i : x_i \neq 0 \text{ for } i \in [n]\}| \), 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 \( \varepsilon \)-far if \( d(x, y) \geq \varepsilon \).) 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 \( \varepsilon \)-close, denoted by \( X \approx_\varepsilon Y \), if \( \Delta(X, Y) \leq \varepsilon \).

2.2 Non-Malleable Codes

We next present some standard definitions related to non-malleable codes. We denote the length of the input message by \( k \) while \( n \) is the length of the codeword.
Definition 1 (Coding Scheme [DPW18]). A coding scheme, \((E, D)\), consists of a (possibly randomized) encoding function \(E : \{0, 1\}^k \to \{0, 1\}^n\) and a deterministic decoding function \(D : \{0, 1\}^n \to \{0, 1\}^k \cup \{\bot\}\) such that \(\forall m \in \{0, 1\}^k, \Pr[D(E(m)) = m] = 1\) (over randomness of \(E\)).

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

\[
\text{Tamper}_{f, m} \overset{\text{def}}{=} \left\{ \begin{array}{l} \epsilon \leftarrow E(m), \tilde{c} := f(\epsilon), \tilde{m} := D(\tilde{c}), \\
\text{Output} : \tilde{m}. \end{array} \right. \]

where the randomness of the experiment comes from \(E\). We say a coding scheme \((E, D)\) is \(\varepsilon\)-non-malleable with respect to \(\mathcal{F}\) if for each \(f \in \mathcal{F}\), 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

\[
\text{Tamper}_{m}^{f} \approx_{\varepsilon} \left\{ \begin{array}{l} \tilde{m} \leftarrow D_{f} \\
\text{Output} : m \text{ if } \tilde{m} = \text{same}; \\
\text{otherwise} \tilde{m}. \end{array} \right. \]

Here the indistinguishability can be either statistical or computational.

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

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

where the probability is over the uniform choice of \(b\) and the randomness of \(E\).

We next present the converse of non-malleability property as,

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

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

We next define a class of local functions, where the number of input bits that can affect any output bit (input locality) is restricted. 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 \(f : \{0, 1\}^n \to \{0, 1\}^n\) 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 \( f : \{0,1\}^n \to \{0,1\}^n \) 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 \( f : \{0,1\}^n \to \{0,1\}^n \) is said to be \( \ell \)-input local, \( f \in \text{Local}_\ell \), if it has input locality \( \ell \).

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

Recall that \( \text{NC}^1 \) is the class of functions where each output bit can be computed by a efficiently and uniformly generated \( \text{poly}(n) \) size boolean circuit with \( O(\log n) \) depth and constant fan-in, where \( n \) is the input size. \( \text{NC} \) is the class of functions where each output is computed by a uniformly and efficiently generated \( \text{poly log}(n) \) depth \( \text{poly}(n) \) size circuit. \( \text{nu-NC} \) is the class of functions computed by a \( \text{poly log}(n) \) depth \( \text{poly}(n) \) size circuit.

Definition 9 (Pseudorandom Generator [DVV16]). Let \( n, n' \in \mathbb{N} \) such that \( n' > n \), and let \( \text{PRG} = \{\text{prg}_n : \{0,1\}^n \to \{0,1\}^{n'}\} \) be a family of deterministic functions which can be computed in computational class \( C_1 \). We say \( \text{PRG} \) is a \( C_1 \)-pseudorandom generator for \( C_2 \) if for any \( D := \{D_n : \{0,1\}^n \to \{0,1\}\} \in C_2 \):

\[
|\Pr[D_n(\text{prg}_n(x)) = 1] - \Pr[D_n(r) = 1]| \leq \text{negl}(n)
\]

where, \( x \leftarrow \{0,1\}^n \) and \( r \leftarrow \{0,1\}^{n'} \) are sampled uniform randomly.

For class \( C \), if \( C_1 = C_2 = C \) then we simply call it \( C \)-pseudorandom generator.

2.4 Black-Box Reductions

We now present definitions related to black-box reductions.

Definition 10 (Distributional Problem). A distributional problem is a decision problem along with ensembles \( (\Psi = \{\Psi_n\}_{n=1}^\infty, L = \{L_n\}_{n=1}^\infty) \) for \( n \in \mathbb{N} \), where \( \Psi_n \) is probability distribution over \( \{0,1\}^n \). The decision problem is to decide whether \( s \in L_n \) where, \( s \leftarrow \Psi_n \).

Note that length of \( s \) need not be \( n \).

We say distributional problem \( (\Psi = \{\Psi_n\}_{n=1}^\infty, L = \{L_n\}_{n=1}^\infty) \) is in \( \text{P} \) if \( L \in \text{P} \). We say it is efficiently samplable if there is a randomized poly-time algorithm that on input \( 1^n \) samples \( \Psi_n \).

We next present definitions related to hardness of distributional problems.

Definition 11 (\( \varepsilon(n) \)-Hard Distributional Problem). Let \( (\Psi = \{\Psi_n\}_{n=1}^\infty, L = \{L_n\}_{n=1}^\infty) \) be a distributional problem, we say that \( (\Psi, L) \) is \( \varepsilon(n) \)-hard for family of boolean circuits \( C = \{C_n\}_{n=1}^\infty \), if and only if for every circuit \( C_n \in C \),

\[
\Pr_{x \leftarrow \Psi_n} [C_n(x) = L_n(x)] \leq \frac{1}{2} + \varepsilon(n)
\]

We also present the definition of \( \varepsilon \)-easy distributional problem below,

Definition 12 (\( \varepsilon(n) \)-Easy Distributional Problem). Let \( (\Psi = \{\Psi_n\}_{n=1}^\infty, L = \{L_n\}_{n=1}^\infty) \) be a distributional problem, we say that \( (\Psi, L) \) is \( \varepsilon(n) \)-easy for family of boolean circuits \( C = \{C_n\}_{n=1}^\infty \), if there exists some circuit \( C_n \in C \),

\[
\Pr_{x \leftarrow \Psi_n} [C_n(x) = L_n(x)] > \frac{1}{2} + \varepsilon(n)
\]
2.5 Hardness of Boolean Functions

In this section we present some standard definitions related to boolean functions and their hardness.

Definition 13 ($\delta$-hardness of boolean function). Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be a boolean function, and $U_n$ be uniform distribution over $\{0, 1\}^n$. Also let $0 < \delta < \frac{1}{2}$, and $n \leq s \leq \frac{2}{\delta}$. We say $f$ is $\delta$-hard for size $s$ if for any boolean circuits $C$ of size at most $s$
\[
\Pr_{x \leftarrow U_n}[C(x) = f(x)] \leq 1 - \delta
\].

We also present the following theorem from [Imp95].

Theorem 1 (Theorem 1 [Imp95]). Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be boolean function that $\delta$-hard for size $s$. Also, let $\varepsilon > 0$. Then $\exists$ set $S \subseteq \{0, 1\}^n$ and constant $c$, such that $|S| \geq \delta \cdot 2^n$ and $f$ is $\varepsilon$-hard-core on $S$ for circuits of size $s' \leq c \cdot \varepsilon^2 \cdot \delta^2 \cdot s$.

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

Definition 14 (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 $F^G$ 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 left to the right correspond to the oracle queries $x_1, \ldots, x_{t(n)}$ 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 $\text{Unroll}(F^G)$, is an unrolling of $F^G(x)$.

Definition 15 (Replace Functionality.). Consider replacing each $g_{n_i}, i \in [t(n)]$, in $\text{Unroll}(F^G)$ with a circuit $g'_{n_i}$ that takes input $(x_1, \ldots, x_i)$ and produces output $y_i$. This is denoted by $\text{Replace}(\text{Unroll}(F^G)), g'_{n_1}, \ldots, g'_{n_{t(n)}}$.

Definition 16 ($((G, t, \ell, u)$-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 $G \in \mathcal{G}$, $\text{Unroll}(F') \in \mathcal{F}$, we have that for all $G' \in \mathcal{G}$ and all $g'_{n_1}, \ldots, g'_{n_{t(n)}} \in G'$, $\text{Replace}(\text{Unroll}(F'), g'_{n_1}, \ldots, g'_{n_{t(n)})} \in \mathcal{F}$.

Definition 17 ($((G, t)$-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 $F^G$ 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 $g'_{t_1}, \ldots, g'_{t_{t(n)}} \in G'$, $\text{Replace}(\text{Unroll}(F'), g'_{t_1}, \ldots, g'_{t_{t(n)})) \in \mathcal{F}$.

Definition 18 (Hard Core Set (HCS) Amenable). We say that $\mathcal{F} = \{f_n\}_{n=1}^\infty$ 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 definition of black-box reduction.

8
Definition 19 (Black-Box-Reduction). We say $R$ is an $(F,ε,δ)$-black-box reduction from a (single bit) non-malleable code, $(E,D) = \{(E_n,D_n)\}_{n=1}^\infty$, to a distributional problem, $(Ψ,L) = \{(Ψ_n,L_n)\}_{n=1}^\infty$, 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 $ε(n)$-malleability, for non-negligible $ε$, i.e.,

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

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

$$\Pr_{x \sim Ψ_n} [L_n(x) = R^{(f_k)}_{k=1}(x)] > \frac{1}{2} + δ(n).$$

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

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

$$\Pr_{x \sim Ψ_n} [L_n(x) = R^f_n(x)] > \frac{1}{2} + δ(n).$$

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,

$$\Pr_{x \sim Ψ_n} [L_n(x) = R^{(f_k)}_{k=p(n)}(x)] > \frac{1}{2} + δ(n).$$

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

3 NMC against $d - 1$ arbitrary errors for message space of size 2

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, the $D$ outputs 0. Note that this code has distance $d$.

We next prove that $(E,D)$ is a 0-non-malleable code (i.e. the distributions in the tampering experiment–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,

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

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

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 if one only needs to protect a single bit, which is the minimal non-trivial setting.

4.1 Expanded Adversarial Error-Correcting Code Channels

In this section we explore the difference between error-correcting codes (ECC) and non-malleable codes (NMC). Specifically, we ask the following question: Assume that we only desire the non-malleability but do not insist on the error-correction, then is it possible to obtain improved parameters (e.g. tampering with a larger fraction of codeword symbols), beyond what is possible in the ECC setting, without imposing additional restrictions on the tampering function. We answer this question negatively for message space of size greater than 2, thereby showing that in order to obtain improved parameters beyond ECC, imposing additional restrictions on the tampering function is necessary.

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

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

**Theorem 2 (Lower Bound for changing half the symbols).** Let \((E, D)\) be a coding scheme with alphabet \(\Sigma\) and distance \(d\). Then \((E, D)\) is not a \(\frac{1}{2}\)-NMC for \(F_{d/2}\) when the message space has cardinality greater than 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. \(\|c_x - z\|_0 \leq \frac{d}{2}\)
2. \(\Pr[D(z) \neq x] \geq \frac{1}{2}\).

Assuming the claim, consider the following tampering function \(f \in F_{d/2}\). 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 $\Pr_{c_x \leftarrow E(x)}[D(z(c_x)) \neq x^*] = p \geq \frac{1}{2}$. Then, $\exists y^* \neq x^* \in U$ such that $\Pr_{c_x \leftarrow E(x^*)}[D(z(c_x))] = y^* \leq \frac{p}{|U| - 1}$. So, $\Pr_{c_x \leftarrow E(x)}[D(z(c_x)) \neq x^*] = p \geq \frac{1}{2}$, but $\Pr[D(f(E(y^*))) = y^*] = 1$. This means that a distribution $D^{f^*}$, that exactly agrees with $D(f(E(\cdot)))$ on $x^*$ must output same* or $x^*$ with probability $1 - p \leq \frac{1}{2}$ and $y^*$ with probability at most $\frac{p}{|U| - 1}$. However, a distribution $D^{f^*}$, that exactly agrees with $D(f(E(\cdot)))$ on $y^*$ must output same* 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, since $p \geq 1/2$ and $|U| > 2$.

Next we prove the claim 4.1.

**Proof (Proof of Claim).** Suppose for the sake of contradiction that $\forall x \in U, \exists c_x \in E(x)$ such that $\forall z \in V$ with $\|z - c_x\|_0 \leq \frac{d}{2}$ it is the case that $\Pr[D(z) \neq x] < \frac{1}{2}$. 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 $\Pr[D(z) = x] > \frac{1}{2}$ and $\Pr[D(z) = y] > \frac{1}{2}$, which is a contradiction because $x \neq y$.

### 4.2 Input-Local Functions

In this section, we rule out non-malleable codes for input-local functions. These are functions which are restricted in that each input symbol may only affect $\ell$ output symbols, where $\ell$ is some locality parameter. We show that even for $\ell = 1$, non-malleability is impossible to achieve. Contrast this with the case of output-locality (where each output depends on at most $\ell$ inputs) where we have explicit constructions for $\ell < n/\log n$.\(^4\) [BDKM16]

Opening up our proof, we can alternately view this as building on the previous impossibility result. Specifically, if one allows fixing codeword symbols to constants, then one cannot achieve non-malleability against functions with just a single symbol whose value depends on the input.

**Theorem 3.** There is no $1/2$-NMC for $Local_1$.

**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

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

Let $x = D(c_j)$ and let $y = D(c_{j+1})$, where $x \neq y$. Now, consider the following $f \in Local_1$,

$$f(c) = \begin{cases} c_j & \text{if } c \in E(y) \\ c_{j+1} & \text{otherwise} \end{cases}$$

(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 $D^{f^*}$.

Thus, even inefficient NMC do not exist for $Local_1$.

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

In this section we consider the class of $(n - \log n)$-output-local tampering functions $F$. Recall that $(n - \log(n))$-output-local functions are functions $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ for which each output bit depends on at most

\(^4\) These constructions however use a randomized decoding algorithm. For the case of deterministic decoding, constructions are only known for locality up to $n^{1/2 - \epsilon}$ for small $\epsilon$. [CL17b]
$n - \log(n)$ input bits. The reason this class is of interest is two-fold: First, parity over $n$ bits is average-case hard for the class $\mathcal{F}$. This can be seen since $\mathcal{F}$ can also be viewed as the class where each output bit is represented by a decision tree of depth at most $n - \log(n)$, while decision trees of depth less than $n$ have 0 correlation with parity over $n$ bits. 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 yet cannot construct non-malleable codes for. Second, we note that the particular class $\mathcal{F}'$ that we use in our lower bound proof is actually a subclass of all $(n - \log n)$-local tampering functions $\mathcal{F}$. In particular, each $f \in \mathcal{F}'$ has the following properties: First, $f_1, \ldots, f_{n - \log n}$ (the functions that output the first $n - \log n$ bits) are the same, except for two different bits from $\{0, 1\}$ can potentially be hardcoded in each. Second, $f_{n - \log n + 1}, \ldots, f_n$ are the same, except for a different value from $\{0, 1\}$ can potentially be hardcoded in each. Finally, the set of input bits upon which $f_1, \ldots, f_{n - \log n}$ depend is fixed and the set of input bits upon which $f_{n - \log n + 1}, \ldots, f_n$ depend is fixed. Taken together, this means that the total number of functions $f$ in $\mathcal{F}$ is at most $4^n \cdot 2^{2n - \log n}$, which in turn means that $\log \log |\mathcal{F}'| = n - \log(n)$. On the other hand, a result of Dziembowski et al. [DPW18] shows that there exists 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 nearly tight matching the existential upper bound.

We prove the following theorem:

**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 former implies ruling out the latter 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

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

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. $c^{\text{top}} := c_1, \ldots, c_{n - \log n}$ ($c^{\text{bot}} := c_{n - \log n + 1}, \ldots, c_n$). 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:**

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

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 $c'_i = c_i^{*,1-d}$.

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


\[
\Pr_{b \leftarrow \{0,1\}}[D(f(E(b))) = 1 - b] = \Pr_{b \leftarrow \{0,1\}}[f(E(b)) \text{ outputs } c^{*1-b}]
\]
\[
= \Pr_{b \leftarrow \{0,1\}}[D(c^{\top}||r) = b \mid c \leftarrow E(b), r \leftarrow U_{\log n}] \\
\geq 1/2 + \epsilon \\
= 1/2 + 1/4n
\]

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

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 \(c_{\bot}\) 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 \(S_{\log n} \setminus \{c_{\bot}\}\).
- 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:
\[
1/2 - \epsilon \leq \Pr_{b \leftarrow \{0,1\}}[D(c^{\top}||r) = 1 - b \mid c \leftarrow E(b), r \leftarrow U_{\log n}]
\]
\[
= \Pr_{b \leftarrow \{0,1\}}[c_{\bot} = r \mid c \leftarrow E(b), r \leftarrow U_{\log n}] \cdot \Pr_{b \leftarrow \{0,1\}}[D(c^{\top}||r) = 1 - b \mid c_{\bot} = r \land c \leftarrow E(b), r \leftarrow U_{\log n}]
\]
\[
+ \Pr_{b \leftarrow \{0,1\}}[c_{\bot} \neq r \mid c \leftarrow E(b), r \leftarrow U_{\log n}] \cdot \Pr_{b \leftarrow \{0,1\}}[D(c^{\top}||r) = 1 - b \mid c_{\bot} \neq r \land c \leftarrow E(b), r \leftarrow U_{\log n}]
\]
\[
= \Pr_{b \leftarrow \{0,1\}}[c_{\bot} = r \mid c \leftarrow E(b), r \leftarrow U_{\log n}] \cdot 0
\]
\[
+ \Pr_{b \leftarrow \{0,1\}}[c_{\bot} \neq r \mid c \leftarrow E(b), r \leftarrow U_{\log n}] \cdot \Pr_{b \leftarrow \{0,1\}}[D(c^{\top}||r) = 1 - b \mid c_{\bot} \neq r \land c \leftarrow E(b), r \leftarrow U_{\log n}]
\]
\[
= (1 - 1/n) \cdot \Pr_{b \leftarrow \{0,1\}}[D(c^{\top}||r) = 1 - b \mid c_{\bot} \neq r \land c \leftarrow E(b), r \leftarrow U_{\log n}].
\]

Note that
\[
\Pr_{b \leftarrow \{0,1\}}[D(c^{\top}||r) = 1 - b \mid c_{\bot} \neq r \land c \leftarrow E(b), r \leftarrow U_{\log n}] = \Pr_{b \leftarrow \{0,1\}}[D(f(E(b))) = 1 - b].
\]
Thus, we have that
\[\frac{1}{2} - \epsilon \leq (1 - 1/n) \Pr_{b \sim \{0,1\}} [D(f(E(b))) = 1 - b].\]

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

This implies the 1/4n-malleability of (E, D). \(^5\)

5 On NMC from Average-Case Hardness via BB Reductions

We demonstrate a barrier to explicitly constructing NMC for NC, even assuming the existence of a language in P that is hard on-average for NC. In particular, we will rule out NMC for NC, even for encoding a single bit, with reductions in NC that make black box usage of the adversary in the security proof. Recall that efficient non-malleable codes for NC imply (efficiently samplable) distribution problems in P that are hard for NC. So, we are, in effect, ruling out such codes with black-box reductions to minimal assumptions.

In fact, our results do not hold just with respect to NC, but for any circuit class that contains majority and closed under certain kinds of composition.

For the case of non-malleable codes where the security proof makes black-box usage of the adversary but is restricted making queries on inputs of the same (or closely related) security parameter as the reduction’s input, we can rule out classes closed which only support weaker composition restraints, including NC\(^1\).

**Black-box reductions for non-malleable codes.** 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 2.4.

**Barrier for security parameter-preserving reductions.** We now prove the central lemma of the section: ruling out security parameter-preserving reductions using the meta-reduction technique.

**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 \(x_i = x_{\ell(n)}\) is equal to \(x_i\). The circuit then 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 \(O(\log n)\) [AKSS83]. 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 \(y_{\ell}\).

\(^5\) 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 \(\epsilon = \frac{1}{4n}\) to \(\epsilon = \frac{1}{8\sqrt{n}}\) while keeping the lower bound same \((\log \log |F| \leq n - \log(n))\).
Lemma 2. Assume that \( \mathcal{F} \) is \((\mathcal{F}, t, p(n), p(n))\)-closed under composition, 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 (single bit) non-malleable code for \(\mathcal{F}\), \((E, D) = \{(E_n, D_n)\}_{n=1}^\infty\), 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 \(\mathcal{F}\).
2. \((\Psi, L)\) is \((\delta(n)/2)\)-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.

Proof. Let \(R\) be such a security parameter-preserving \((\mathcal{F}, 1/2, \delta(n))\)-reduction, for a non-malleable code \((E, D)\) and distributional problem \((\Psi, L)\). Moreover, for security parameter \(n\), let \(p(n)\) be the length of the codeword generated by \(E\), 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 \(H : \{0, 1\}^{p(n)} \rightarrow \{0, 1\}^\star\)):

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

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

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

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

\[
\Pr_{x \in \Psi_n} \left[ L_n(x) = R^\Psi_{p(n)}(x) \right] \geq \frac{1}{2} + \delta(n).
\]

Now, for the \(j\)-th oracle query, we define \(f_{p(n)}^{(j)}\), 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 hardcodes a random codeword (sampled from \(E(b)\) where \(b\) is uniform) as the \(y_j\) value.

By our assumption on \(\mathcal{F}\) (and \(R\)), we have that \(\text{Replace(Unroll}(R^\Psi_{p(n)}), f_{p(n)}^{(j)}, \ldots, f_{p(n)}^{(t(n))}) \in \mathcal{F}\). We will abuse notation and denote the resulting circuit by \(R^\Psi_{p(n)}\). So, it suffices to show that the behavior of \(R^\Psi_{p(n)}(x)\) is close that of \(R^H_{p(n)}(x)\), for any \(x\), which will imply that \(R^\Psi_{p(n)}(x) \in \mathcal{F}\) breaks the distributional problem w.h.p., since \(R^H_{p(n)}(x)\) does. More accurately, if \((E, D)\) is \(\frac{\delta(n)}{2^{t(n)}}\)-non-malleable by \(\mathcal{F}\), then we will show that

\[
\forall n \in \mathbb{N}, \forall x \in \{0, 1\}^n, \Delta(R^\Psi_{p(n)}(x); R^H_{p(n)}(x)) \leq \delta(n)/2,
\]

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

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

In the \(i\)-th hybrid, the function \(f_{p(n)}^{(i), j}\) 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 \(y_1^i, \ldots, y_j^i\). For \(k \in [t - i]\), the \(y_k^i\) values are sampled as follows: For \(k \in [t - i]\), \(y_k^i\) is sampled as if \(f^H_{p(n)}\). For \(k > t - i\), \(y_k^i\) 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, \(f_{p(n)}^{(0)} = f_{p(n)}\) and \(f_{p(n)}^{(t)} = f'_{p(n)}\).

Note that polynomial \(p(n)\) corresponds to the length of the codeword outputted by \(E_n\).
We will show that for all \( x \in \{0, 1\}^n \) (and any fixing of random coins \( r \) for \( R \)) \( \Delta(R^{f(i)}_p(x); R^{f(i-1)}_p(x)) \leq \epsilon(n) \) (for \( i \in [t(n)] \)), which proves the claim above. \( (R^{f(i)}_p(x) \) 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 \( f^{(i)}_p \) and \( f^{(i-1)}_p \): \( |\Pr[R^{f^{(i)}_p}(x) = 1] - \Pr[R^{f^{(i-1)}_p}(x) = 1]| \geq \frac{\delta(n)}{2t(n)}. \)

Note that for fixed random function \( H \) (that generates the random coins used to sample the \( y_j \) values) \( f^{(i)}_p \) and \( f^{(i-1)}_p \) differ solely on the response to \((t - i)\)-th query. So, fix \( x, H \) and all but the \((t - i)\)-th value \( y_{n-i} \) 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 \( f^{(i)}_p \) and \( f^{(i-1)}_p \), 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 \( R^{f^{(i-1)}_p}(x) \) 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 \( R^{f^{(i)}_p}(x) \), 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 \( R^{f^{(i)}_p} \), with queries/responses to \( f^{(i),j}, j \in [t - i - 1] \) hardcoded, the \((t - i)\)-th query hardcoded, the \((t - i)\)-th value \( y_{n-i} \), as the input to the circuit, and for \( j > t - i \), the \( f^{(i),j} \) 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:

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

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 \( s_{H,x} \) 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 \( s_{H,x} \);
- Otherwise, output \( c_{1 - s_{H,x}} \).

We now analyze

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

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

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

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 \(\frac{\delta(n)}{2t(n)}\)-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 in the class of tampering functions \(F\). Specifically, in the hybrid arguments, we require that \(R'_{H,x}(\cdot)\) is in \(F\). In particular, our proof approach does not extend to proving impossibility of constructing a (single bit) non-malleable code for \(F\), from a distributional problem, \((\Psi, L) = \{(\Psi_n, L_n)\}_{n=1}^\infty\) that is hard for some larger class \(\hat{F}\). 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 \(\hat{F}\) from an “incompressibility”-type assumption, as those used in the recent work of [BDKS’18]. Briefly, if a function \(\psi\) is incompressible by circuit class \(\hat{F}\), it means that for \(t \ll n\), for any *computationally unbounded* Boolean function \(D : \{0,1\}^t \rightarrow \{0,1\}\) and any \(F : \{0,1\}^n \rightarrow \{0,1\}^t \in \hat{F}\), 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 \(R \in \hat{F}\) and the query made to \(D\) has length \(t \ll n\). Since \(R\) composed with \(D\) is clearly outside the tampering class \(\hat{F}\), our proof approach does not apply in the incompressibility setting.

**Note 2.** We can extend Lemma 2 to rule out \((u(n), \ell(n))-\text{approximately}\) security parameter preserving reductions by allowing our reduction access to a greater range of inefficient/simulated tampering functions (defined in the same manner as above): \(\{f_k\}_{k = \ell(n)}^{u(n)}\) and \(\{f_k'\}_{k = \ell(n)}^{u(n)}\). In this case, we can, WLOG, confine 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 \(F\) is \((\hat{F}, t, \ell, u)\)-closed under composition and contains look-up circuits with \(\ell(n)\) to \(u(n)\)-bit keys and values and \(t(n)\) inputs, for polynomials \(t(\cdot), \ell(\cdot), u(\cdot)\).

**Lemma 3.** Assume \(\hat{F}\) is \((\hat{F}, t, \ell, u)\)-closed under composition 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 \((\hat{F}, 1/2, \delta(n))\)-black-box-reduction making \(t(n)\) length-preserving queries from a (single bit) non-malleable code for \(\hat{F}\), \((E, D) = \{(E_n, D_n)\}_{n=1}^\infty\), to a distributional problem, \((\Psi, L) = \{(\Psi_n, L_n)\}_{n=1}^\infty\), then one of the following must hold:
1. \( (E, D) \) is \( \frac{\delta(n)}{\ell(n)} \)-malleable by \( F \).
2. \( (\Psi, L) \) is \( (\delta(n)/2) \)-easy for \( F \).

Moreover, if \( (E, D) \) is efficient, then for the conclusion to hold it suffices that \( F \) contains such look-up circuits generated that are generated uniformly at random.

The following corollary holds since \( \text{NC}^1 \) is \( (\text{NC}^1, t, \ell, u) \)-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 \( t(\cdot), \ell(\cdot), u(\cdot) \).

**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 \( \text{NC}^1 \), \( (E, D) = \{(E_n, D_n)\}_{n=1}^\infty \) to a distributional problem, \( (\Psi, L) = \{(\Psi_n, L_n)\}_{n=1}^\infty \), then one of the following must hold:

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

We extend to non-security parameter preserving reductions, but require a stronger compositional property for the tampering class \( F \). 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.** Assume \( F \) is \( (F, t) \)-closed under strong composition and is HCS-amenable. If for every non-negligible \( \epsilon = \epsilon(\cdot) \), there is an \( (F, \epsilon, \delta) \)-black-box-reduction, for some non-negligible \( \delta = \delta(\cdot) \), making \( t(n) \) queries from a (single bit) non-malleable code for \( F \), \( (E, D) = \{(E_n, D_n)\}_{n=1}^\infty \) to a distributional problem, \( (\Psi, L) = \{(\Psi_n, L_n)\}_{n=1}^\infty \), then \( (\Psi, L) \) is \( (\delta(n)) \)-easy for \( F \), for some non-negligible \( \delta' = \delta'(\cdot) \).

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

\[
\epsilon(n) := \begin{cases} 
\frac{1}{4} & \text{if } n \in S \\
0 & \text{if } n \notin S 
\end{cases}
\]

Assume there is some reduction \( R \) that succeeds with non-negligible probability \( \delta \) 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 \( \ell^{-1}(k) \cap S' \) is finite (otherwise we can hardcoded 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 WLOG that \( \ell(n) < n \), since otherwise our previous proof holds.

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

1. For infinitely many \( n \in S' \) (this set of values is denoted by \( S'' \subseteq S' \)), there is some adversary in \( F_n \) that outputs \( D_{\ell(n)}(c) \) with probability at least \( 3/4^7 \).
2. For infinitely many \( n \in S' \) (this set of values is denoted by \( S'' \subseteq S' \)), there is some hardcore set \( \mathcal{H} \) of size \( \epsilon'(n) \cdot 2^\ell \), where \( \epsilon'(n) \leq \frac{1}{2 n^c - \ell(n)} \) such that every adversary in \( F_n \) outputs \( D_{\ell(n)}(c) \) with probability at most \( 1/2 + \epsilon'(n) \), when \( c \) is chosen at random from \( \mathcal{H}' \).

\footnote{Note that \( D_{\ell(n)}(c) \) takes inputs of length \( \ell(n) \), whereas \( F_n \) takes inputs of length \( n \). We can easily resolve this discrepancy by padding inputs of length \( \ell(n) \) up to \( n \).}

\footnote{Again, the input \( c \) to \( D_{\ell(n)} \) has length \( \ell(n) \) while \( F_n \) takes inputs of length \( n \). As above, we resolve the discrepancy by padding inputs of length \( \ell(n) \) up to \( n \).}
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 \( k \in S \). Clearly, \( f_k \) succeeds with non-negligible probability \( \epsilon \). Since the \( \epsilon \) function remains the same, we know that \( \delta \) and \( \ell, S, S' \) 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 \( k = k(n) \in \mathcal{O}(\log(n)) \).

Note that since we are in Case 1, for infinitely many input lengths–input lengths \( n \in S'' \) 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 \( n \in S', R^{f_{\ell(n)}} \) agrees with \( (\Psi, L) \) with probability \( 1/2 + 1/n^c \). But then we must have that for infinitely many \( n \in S' \)–input lengths \( n \in S'' \)– \( R^{\text{Sim}} \) agrees with \( (\Psi, L) \) with probability \( 1/2 + 1/n^c \) and \( R^{\text{Sim}} \in \mathcal{F}_n \). So \( (\Psi, L) \) is \( (\delta'(n)) \)-easy for \( \mathcal{F} \), where

\[
\delta'(n) := \begin{cases} 
\frac{1}{2 n^c} & \text{if } n \in S'' \\
0 & \text{if } n \notin S''
\end{cases}
\]

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 decodes to 0 or 1 as appropriate. Specifically, the hardcore set \( H \) is defined as follows: \( f_k \) sets \( n^* \) to be equal to the lexicographically first element in the (finite) set \( \ell^{-1}(k) \cap S'' \), and chooses the lexicographically first set \( H \) of size \( \epsilon'(n^*) \cdot 2^{ \epsilon'(n^*) } = \epsilon'(n^*) \cdot 2^2 \) for which every adversary in \( \mathcal{F}_n \) outputs \( \ell_{\ell(n)}(c) \) with probability at most \( 1/2 + \epsilon'(n^*) \), when \( c \) is chosen at random from \( H \). If \( \ell^{-1}(k) \cap S' = \emptyset \) or there is no such hardcore set \( H \), 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 \( k \in S \). Since the \( \epsilon \) function remains the same in this case as well, the \( \delta \) function also remains the same. Thus, for \( n \in S', R^{f_{\ell(n)}} \) must still agree with \( (\Psi, L) \) with probability \( 1/2 + 1/n^c \).

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 \( H \) (which again we can hardcode in using lookup circuits as before). Moreover, for queries \( R \) makes to \( \text{Sim} \) in input lengths \( k = k(n) \in \mathcal{O}(\log(n)) \), all responses for all possible queries \( c \) are hard-coded into \( \text{Sim} \). Now, for infinitely many \( n \in S' \)–input lengths \( n \in S'' \)– \( R^{\text{Sim}} \)'s behavior should be \( \ell(n) \cdot \epsilon'(n) \)-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 \( n \in S' \)–input lengths \( n \in S'' \)– \( R^{\text{Sim}} \) composed with \( \text{Sim} \) is in the class \( \mathcal{F} \). But due to the fact that \( \mathcal{F} \) is \( (\mathcal{F}, t) \)-closed under strong composition, this occurs whenever the reduction is instantiated with security parameter \( n \in S' \), where \( n = n^* \) is the lexicographically first element in the set \( \ell^{-1}(\ell(n)) \cap S'' \). Since \( n \) is always contained in \( \ell^{-1}(\ell(n)) \), since the size of \( \ell^{-1}(\ell(n)) \cap S' \) is finite and since the size of \( S'' \) is infinite, there will be infinitely many \( n \in S'' \) for which this event occurs. Thus, for infinitely many \( n \in S'' \) (denote this set of values by \( \tilde{S} \), \( R^{f_k}_k \) agrees with \( (\Psi, L) \) with probability \( 1/2 + 1/n^c \) and \( R^{\text{Sim}} \) is \( t(n) \cdot \epsilon'(n) \leq 1/2 n^c \)-close to \( R^{f_k}_k \). So we conclude that \( (\Psi, L) \) is \( (\delta'(n)) \)-easy for \( \mathcal{F} \), where

\[
\delta'(n) := \begin{cases} 
\frac{1}{2 n^c} & \text{if } n \in \tilde{S} \\
0 & \text{if } n \notin \tilde{S}
\end{cases}
\]

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

**Corollary 3.** If for every non-negligible \( \epsilon = \epsilon(\cdot) \), there is an \( (\text{NC}, \mathcal{F}, \delta) \)-black-box-reduction, for some non-negligible \( \delta = \delta(\cdot) \), making \( t(n) \) queries from a (single bit) non-malleable code for \( \text{NC} \), \( (E, D) = \{ (E_n, D_n) \}_{n=1}^{\infty} \), to a distributional problem, \( (\Psi, L) = \{ (\Psi_n, L_n) \}_{n=1}^{\infty} \), then \( (\Psi, L) \) is \( (\delta'(n)) \)-easy for \( \text{NC} \), for some non-negligible \( \delta' = \delta'(\cdot) \).

---

Note that it is finite since \( \ell^{-1}(k) \cap S' \) is finite and \( S'' \subseteq S' \).
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


Dana Dachman-Soled, Feng-Hao Liu, Elaine Shi, and Hong-Sheng Zhou. Locally decodable and updatable non-malleable codes and their applications. In Dodis and Nielsen [DN15], pages 427–450.


OPVV18. Rafail Ostrovsky, Giuseppe Persiano, Daniele Venturi, and Ivan Visconti. Continuously non-malleable codes in the split-state model from minimal assumptions. In Shacham and Boldyreva [SB18], pages 608-639.


