Limits to Non-Malleability

Marshall Ball[1], Dana Dachman-Soled[2], Mukul Kulkarni[3], and Tal Malkin[1]

[1] Columbia University
{marshall,tal}@cs.columbia.edu
[2] University of Maryland
danadach@ece.umd.edu
[3] University of Massachusetts Amherst mukul@cs.umass.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 $\mathcal{F}$?

First, we start with some classes where positive results are well-known, and show that when these classes are extended in a natural way, non-malleable codes are no longer possible. Specifically, we show that no non-malleable codes exist for any of the following 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.

Furthermore, we rule out constructions of non-malleable codes for certain classes $\mathcal{F}$ via reductions to the assumption that a distributional problem is hard for $\mathcal{F}$, 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 NC, even assuming average-case variants of $P \not\subseteq \mathsf{NC}$.

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 [44]. 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 $\mathcal{F}$ is defined via the following "tampering" experiment:

Let $f \in \mathcal{F}$ denote a tampering function.

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

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

[44] showed that, remarkably, this definition is achievable for any $\mathcal{F}$ such that $\mathrm{loglog}|\mathcal{F}| < n - 2 \cdot \log(1/\epsilon)$, where n is the length of the codeword (the input/output of functions in $\mathcal{F}$), and ϵ parameterizes the quality of simulation possible (see Definition 3). However the definition is not achievable in general. It is easy to observe that if $\mathcal{F}$ 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. These impossibility results apply to classes that are simple and natural extensions of classes with well-known and efficient NMC constructions. 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. This result is more technically complex than the previous ones, and requires the introduction of a new notion of fine-grained black-box reductions appropriate for the non-malleability setting, as we explain below. This result allows us to study the minimal assumptions necessary for achieving NMC for complexity classes contained in P (e.g., NC^1), and to rule out such NMC constructions (with black-box reductions) from minimal average-case hardness assumptions.

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 1.4. 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 (ECC). It is easy to see that ECC are NMC against tampering that changes up to the allowed fraction of symbols, but since NMC only require correctness of decoding in the absence of errors, they can provide "security" for tampering functions that ECC cannot, in particular functions that can modify most, or even all, symbols of the codeword. This suggests that we could potentially construct NMC for tampering classes that are strictly larger than any class for which ECC could exist. However, no such construction is known: all NMC results that allow to modify more symbols, also require that the computation of the tampering function is restricted in some way. In contrast, ECC do not place any restrictions on the tampering adversary beyond the limit to the number of modified symbols.

In the current work we ask whether this trade-off is in fact necessary. Specifically, can we construct NMC that allow for modifying more symbols of the codeword than ECC *without* placing any other restrictions on the tampering? Note that for ECC 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 $d/2-1$ symbols (e.g., Reed-Solomon ECC achieve this bound). Indeed, for the case of potentially inefficient coding schemes, the above is tight: A coding scheme with distance d implies error correction against $d/2-1$ errors.

Thus, fixing a message space $\mathcal{M}$ and a codeword space $\mathcal{C}$, we consider the optimal ECC for this message and codeword space, which has some distance d (and therefore can correct $d/2-1$ errors). We then ask whether one can construct a non-malleable code with message space $\mathcal{M}$ and codeword space $\mathcal{C}$ against the class of functions that may arbitrarily tamper with $d/2$ codeword symbols.

We fully resolve our question. We show that for message space of size 2, non-malleable codes that tolerate arbitrary modification of even up to $d-1$ symbols are *possible* (via a repetition code, see Section 3). 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 affects at most one output symbol. In their recent work, Ball et al. [17] 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 this 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 $n - \log(1/(4\epsilon))$ input symbols. Here we move on to consider achieving NMC for output-local tampering functions. The prior work of [17] constructed efficient NMC for tampering functions with locality n^c, for constant $c < 1$. The size of the class of all output-local tampering functions (with locality sufficiently smaller than n) is also bounded in size and therefore NMC for this class follow from the existential results of [44]. A natural question is how large can the output-locality be?

We prove the impossibility of ϵ-non-malleable codes (see Definition 3) for the class of $(n - \log(1/(4\epsilon)))$-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 $4^n \cdot 2^{2^{n-\log(1/(4\epsilon))}}$, which in turn means that $\log\log|\mathcal{F}'| = n - \log(1/(4\epsilon)) = n - \log(1/\epsilon) + 2$. On the other hand, the aforementioned result of Dziembowski et al. [44] shows existence of an ϵ-non-malleable code for any class $\mathcal{F}$ such that $\log\log|\mathcal{F}| \le n - 2\log(1/\epsilon)$. Thus, our lower bound result is close to matching the existential upper bound.

[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. [22]

Remark: deterministic vs. randomized decoding. The standard (and original) definition of NMC requires deterministic decoding and perfect correctness, although some later work took advantage of randomized decoding.[2] We note that our lower bound for the class of input-local functions holds for standard schemes (with deterministic decoding and perfect correctness). Our lower bound (with $\epsilon = \frac{1}{4n}$) 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 NC^0 [17, 29, 15] and AC^0 [29, 15], have been presented. In fact, the construction of [15] 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 P, extending their result to obtain unconditional NMC for circuits with asymptotically larger depth would require separating P from 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 NC^1, as well as other classes $\mathcal{F}$ that are believed to be strictly contained in P.

The above question was partially addressed by Ball et al. [18, 16] 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 NC^1 yields a computational, CRS-model construction of 1-bit NMC for NC^1, assuming there is a distributional problem that is hard for NC^1, but easy for P. Moreover, such distributional problems for NC^1 can be based on worst-case assumptions.[3]

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

We provide a negative answer, showing 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 (Ψ, 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. To motivate our new notion, we begin by recalling the notions of reductions in complexity theory and cryptography, and how they are used.

Reductions in Complexity Theory. A reduction is a technique in complexity theory that is used to show that *Problem 1* is as hard as *Problem 2*. For example, the famous Cook-Levin theorem showed

[2] For the class of output-local functions (where each output depends on at most ℓ inputs) we have explicit constructions with randomized decoding (and $\epsilon = \mathsf{negl}(n)$) for $\ell < n/\log n$ [17], whereasconstructions with deterministic decoding are known for locality up to $n^{1/2-\epsilon}$ for small ϵ. [28, 15].

[3] Assuming $\oplus\mathsf{L}/\operatorname{poly} \not\subseteq \mathsf{NC}^1$ yields a distributional problem since randomized encodings for $\oplus\mathsf{L}/\operatorname{poly}$ are known to exist [13, 21, 41, 14].

that SAT (Boolean satisfiability) is as hard as any problem in NP, by showing a reduction from any problem in NP to SAT. In more detail, a reduction R, is an algorithm that receives as input an algorithm A that solves Problem 1 and uses it to solve Problem 2. Typically, R will only access A in an input/output manner as a subroutine (also known as oracle access and denoted as R^A). When a reduction R uses a solver A in this way, R is known as a black-box reduction. Intuitively, in this case, R does not care *how* A solves Problem 1, it just cares that A exhibits the correct input-output behavior. Therefore, the reduction R should still be able to solve Problem 2, even when A is computationally unbounded. In fact, in the Cook-Levin theorem, only a single oracle query is made by the reduction to the algorithm solving SAT.

When is a reduction R between two problems useful? Note that if R is in a complexity class that can solve Problem 2, then existence of such a reduction R is tautological, since R can simply ignore its oracle and solve Problem 2 on its own (so no relationship is demonstrated about the relative hardness of the problems). This is why in the Cook-Levin theorem the reduction is required to be *polynomial time*. Reductions are also useful since they allow us to draw conclusions about the relationships between different complexity classes. For example, using the Cook-Levin theorem, we conclude that if there exists a polynomial time algorithm for solving SAT then there exists a polynomial time algorithm for all of NP (i.e. $\mathsf{P} = \mathsf{NP}$). Note that there is actually a subtlety here: In order for the above to hold, we need that whenever A is polynomial time, R^A is also polynomial time. This holds trivially for the case of polynomial time, since the class P is closed under composition. However, as we will see later, this closure does not necessarily hold in some of the settings we consider. Many variations on reductions beyond the setting of the Cook-Levin Theorem and NP-completeness have been considered in complexity theory. For example, polynomial time reductions have no utility when P is the object of study. Instead, the theory of P-completeness uses NC-computable reductions to argue about whether or not problems in P can be parallelized. Another such example is the theory of fine-grained complexity, which seeks to understand the *exact* (or, more exact) complexity of problems in P. Fine-grained reductions allow one to argue that a problem cannot be solved in, for example, $O(n^{2-\epsilon})$ time (for any $\epsilon > 0$), by reduction from another problem believed to require $\Omega(n^{2-o(1)})$ time. For this reasoning to hold, such a fine-grained reduction must run in sub-quadratic time, and moreover it can make at most $n^{o(1)}$ queries to an oracle defined on instances of length n if it is to remain useful. As we will see, this tension between the length of the inputs queried to the oracle and the number of such queries will also be relevant in our setting.

Reductions in Cryptography. Reductions in cryptography are exactly like reductions in complexity theory. For example, the seminal result of [56] proves by reduction that *breaking a pseudorandom function* (Problem 1) is as hard as *breaking a pseudorandom generator* (Problem 2). In order to prove this, they present a reduction R such that that given an algorithm A that breaks the constructed pseudorandom function, R^A breaks the underlying pseudorandom generator. Note that since R only has oracle access to A, again R does not care how A works, as long as it exhibits input-output behavior that qualifies it as a valid distinguisher between a pseudorandom and random function. Thus, R is black-box. As before, R is only useful if it is polynomial time, since otherwise R can break the pseudorandom generator on its own. Furthermore, we again want to use the existence of the reduction to draw conclusions about the security relationship between the pseudorandom function and the pseudorandom generator. Here we want to show that if there exists a polynomial-time algorithm that breaks the pseudorandom function, then there exists a polynomial-time algorithm that breaks the pseudorandom generator. Therefore, we want it to be the case that whenever A

is polynomial time, R^A is also polynomial time. This trivially holds, as before, since P is closed under composition. However, in the following we will consider cases where this type of closure does not necessarily hold. For example, when A and R are in NC^1, R^A may no longer be in NC^1 (in fact R^A could have depth up to $\log^2(n)$). We therefore need to include a notion of closure under composition as one of the requirements of a black-box reduction in our setting.

A fine-grained setting: Security reductions for non-malleable codes. What would a security reduction in the setting of non-malleable codes look like? In this case, we want to show that *breaking the non-malleable code* (Problem 1) is as hard as *breaking distributional problem* (Ψ, L) (Problem 2). Here, an algorithm that breaks the non-malleable code simply consists of a *tampering function* f. A reduction R is provided black-box access to the tampering function f and must use it to break the distributional problem (Ψ, L). First, note that since we assume R is black-box, R is only allowed to use f as a subroutine (gives it inputs and obtains its output), regardless of *how* f performs its computation. Thus, as in all the cases discussed above, we require that R^f break the distributional problem (Ψ, L), even in the case that f is not contained in $\mathcal{F}$. Note that the distributional problem (Ψ, L) is *easy* for polynomial-time. Therefore, for the reduction to be non-trivial, R must be in a complexity class that does not contain P. Indeed, since we only assume that (Ψ, L) is hard for $\mathcal{F}$, R must be contained in $\mathcal{F}$ in order for us to draw any conclusions (otherwise, we cannot rule out the possibility that R simply ignores its oracle and solves (Ψ, L) on its own). Furthermore, as discussed above, the point of the reduction is to be able to conclude that if there is a tampering function f in $\mathcal{F}$ that breaks the non-malleable code, then there exists an algorithm in $\mathcal{F}$ that breaks the distributional problem (Ψ, L). Therefore, it is not enough that $R \in \mathcal{F}$, and we actually need that whenever $f \in \mathcal{F}$, $R^f \in \mathcal{F}$. We will then use the fact that R^f breaks (Ψ, L) and is used to obtain a contradiction to the hardness of (Ψ, L) for $\mathcal{F}$.

Overall, at a high level (skipping some technical details), we require two properties of a black-box reduction R from (E, D) to (Ψ, L):

- If the tampering function f succeeds in breaking the non-malleable code, the reduction, R^f, 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^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 (Ψ, 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.

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 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 19, 20 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 security parameter preserving reductions have been used in constructions of leakage resilient circuit compilers [11]. 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 $u(n)$. This notion is somewhat less restrictive than a security parameter preserving reduction. 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. This notion allows us to rule out the most general type of black-box reduction discussed above.

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 NC^1 as a special case. Our result ruling out non-security parameter preserving reductions applies to the class (non-uniform) NC as a special case. See Corollaries 2, 3, 4 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.

Do reductions for NMC take the above form? So far, in the non-malleable codes setting, results have either been *unconditional* (e.g. [44, 4]) or have been based on polynomial-hardness assumptions (e.g. [70, 2]). The results that are based on polynomial-hardness assumptions have all used black-box security reductions, in the standard polynomial-time sense [76]. Our notion is new since it captures a fine-grained setting where the underlying distributional problem is, in fact, *easy* for polynomial-time algorithms. As discussed above, this is the minimal computational hardness assumption necessary to construct non-malleable codes for classes $\mathcal{F}$ for which we cannot prove unconditionally that $\mathsf{P} \not\subseteq \mathcal{F}$. While this type of reduction implicitly arises in the work of [18], our work is the first to formally define and explore this notion of fine-grained black-box reductions in a cryptographic setting.

1.3 Technical Overview

In order to prove impossibility of constructing non-malleable codes in different scenarios, we need to show that any such code is malleable. Recall that for single-bit messages, non-malleability is equivalent to showing that when applying the tampering function to a randomly generated encoding of a random bit, the decoded value flips with probability at most $1/2 + \mathsf{negl}(n)$. Thus, proving that something is malleable, corresponds to showing that the decoded value flips with probability at least $1/2 + 1/\operatorname{poly}(n)$. We will use this fact in the following exposition.

On tampering functions that change $d/2$ symbols, where d is the distance of the code. We observe that for any coding scheme, there must be some message, x^*, such that *every* encoding of that message is at most distance $d/2$ from something that is likely to decode to something other than x^*. Our tampering function will only modify encodings of x^*, and it will do so by moving each encoding to one of these nearby points that decodes differently. We claim that if there are at least 3 messages in the message space, then the output of decoding with the tampering function described above depends on the input message (and thus cannot be simulated). Indeed, when starting with message x^* (which is encoded, tampered, and decoded) there must be some other message $y^* \neq x^*$ that is not output a majority of the times by this process. On the other hand, when starting with y^*, since the tampering does not change anything in this case, correctness of decoding means that y^* should be output a majority of times.

This argument falls apart if there are only two possible messages, and in this case a repetition code with a decoding that outputs a fixed value on invalid codewords is, in fact, non-malleable with respect to tampering functions that can change up to $n-1$ symbols.

On tampering functions where each input symbol affects at most one output symbol. Consider any two codewords c_x and c_y corresponding to distinct messages, x and y. Now consider any sequence of n codewords between c_x and c_y, made by altering one symbol at a time. There must be two adjacent codewords, c_i, c_{i+1}, (differing on a single bit) in this sequence that decode differently. Therefore, to tamper, simply output c_i if the input is an encoding of 0, and c_{i+1} otherwise. Because c_i and c_{i+1} just differ on a single bit, the tampering function has input locality 1.

On tampering functions where each output symbol depends on $n - \log(1/(4\epsilon))$ input symbols. We begin by considering a simpler argument, that only rules out tampering functions of output locality $n-1$ (each output bit can depend on at most $n-1$ input bits), where n is the bit length of the codeword. To illustrate the idea in the locality $n-1$ case, we also assume that the decoding algorithm is deterministic. We consider two cases and show that each case leads to a different tampering attack:

- **Case 1:** Given the first $n-1$ bits of the codeword, the codeword decodes to the same bit b, regardless of whether the final bit is 0 or 1. In this case, the tampering function contains a hardwired valid codeword encoding 0 and a valid codeword encoding 1. The tampering function derives the bit b, given only the first $n-1$ bits (since the decoded bit b is independent of the final bit) and replaces the codeword with the hardwired encoding of $1-b$.
- **Case 2:** Given the first $n-1$ bits of the codeword, the codeword decodes to 0 if the final bit is set to b and decodes to 1 if the final bit is set to $1-b$. In this case, the tampering function just flips the final bit, causing the decoding to flip.

The key observation is that we can extend the attacks for Cases 1 and 2 above to tampering functions of output locality $n - \log(1/(4\epsilon))$. We will sketch the special case corresponding to extending to $\epsilon = \frac{1}{4n}$ (and locality $(n - \log(n))$), to rule out non-malleable codes with negligible error. Case 1 now corresponds to the case that, for a randomly generated codeword, when the first $n - \log(n)$ bits of the codeword are fixed and the remaining $\log(n)$ bits are set at random, the decoded value *remains the same* with probability at least $1/2 + 1/(4n)$ In this case, the tampering function gets the first $n - \log(n)$ bits, randomly sets the final $\log(n)$ bits and decodes to obtain a bit b. Then, the tampering function succeeds in flipping the encoding with probability $1/2 + 1/(4n)$ by replacing the codeword with the hardwired encoding of $1-b$.

Case 2 now corresponds to the case that for a randomly generated codeword, when the first $n - \log(n)$ bits of the codeword are fixed and the remaining $\log(n)$ bits are set at random, the decoded value *flips* with probability at least $1/2 - 1/(4n)$. Note that the decoded value *never flips* when the randomly chosen $\log(n)$ bits happen to be the same as the original value, which occurs with probability $1/n$. Thus, if the final $\log(n)$ bits are chosen at random, conditioned on being different from the original value, then the decoded value must flip with probability at least $\frac{1/2-1/(4n)}{1-1/n} \geq 1/2+1/(4n)$. In this case, the tampering function ignores the first part of the codeword and simply sets the final $\log(n)$ bits at random, *conditioned on the value being different from the original value.* Then, the tampering function succeeds in flipping the value of the encoding with probability at least $1/2 + 1/(4n)$.

Our final argument for $n - \log(1/(4\epsilon))$-locality holds even for *randomized* decoding.

Impossibility of Black-Box Security Reductions. We begin by describing our proof showing the impossibility of a black-box, security-parameter preserving reduction, from NMC against the tampering class NC^1, to a distributional problem that is hard for NC^1. The proof for approximately security parameter preserving reductions is essentially the same, and so we subsequently describe the extension to impossibility of a black-box, *non*-security-parameter preserving reduction, from non-malleable codes against the tampering class NC, to a distributional problem that is hard for NC.

Our proof proceeds via the meta-reduction technique. Specifically, consider a black-box reduction R, reducing the security of a single-bit non-malleable code against NC^1 to a distributional problem that is hard for NC^1. The form that this reduction takes, is that it submits codewords c to the tampering function f and gets back (tampered) codewords y as responses. The main idea is to begin with a tampering function f, which is not in NC^1. This tampering function receives a codeword c, decodes it to obtain the bit b and then submits a randomly generated encoding of the bit $1-b$. In the proof, we assume the existence of a reduction R such that R^f breaks the underlying distributional problem (this follows from the definition of a black-box reduction). We then switch from f to a tampering function f' that *is* in NC^1, which behaves as follows: Upon receiving a codeword c, f simply responds with a (hardcoded) random codeword c' that encodes a random bit, independent of the bit that is obtained when the decoding algorithm is applied to c. This switch is desirable, since then $R^{f'}$ will be in NC^1 (note that we are guaranteed that $R^{f'}$ is in NC^1, since one of the properties of R is that whenever the tampering function f' is in NC^1, then $R^{f'}$ must also be in NC^1). It remains, however, to show that $R^{f'}$ succeeds in breaking the underlying distributional problem, which then implies that the underlying distributional problem is not hard for NC^1. In order to ensure this, we use a hybrid argument, where responses to queries from R are switched one by one, from responses according to f to responses according to f'. In each step, we must show that the reduction remains successful in breaking the underlying distributional problem. Importantly, in the i-th hybrid, the first $i-1$ responses are answered according to f, the i-th response and on are answered according to f'. Since R is in NC^1, we argue that if R can distinguish the $(i-1)$-st and i-th hybrids, then we obtain a *tampering attack in* NC^1 *on the non-malleable code.* To do this, we construct a tampering function that hardwires the input to R, the transcript (queries and responses) and entire state of R for the first $i-1$ queries made from R to f, the i-th query along with the value b that it decodes to, and the responses to the queries $i+1$ and on. Then, given an input codeword c', the tampering function inserts this value as the response to the i-th query, runs the reduction R from this point on (given the state of R at the point of the response to the i-th query) and responds with the random hardwired queries upon any future queries from R. If R distinguishes between Hybrids $i-1$ and i, then the above yields a distinguisher between randomly generated encodings of the bit b, versus randomly generated encodings of a random bit. It is not hard to see that such a distinguisher immediately yields a tampering attack, since it can be used to predict the underlying encoded value and the tampering attack can then replace the codeword with an encoding of a bit which is the opposite of the predicted bit.

In the above, note that it is crucial that the reduction is security parameter preserving. Indeed, if R queries codewords c' that are very short (say length $\log^2(n)$) then we can no longer use R to obtain a valid tampering function against the non-malleable code. This is because R has size $\text{poly}(n)$ and depth $\log(n)$, which is not in NC^1 relative to input length $\log^2(n)$. To deal with this problem, we take advantage of the fact that R must be successful even for tampering functions f that work only for very sparse input lengths $\{1, 2, 2^2, 2^{2^2}, \ldots\}$. In this way, we can essentially guarantee that the reduction queries at most a single input length ℓ which is greater than $\log(n)$ and at most $\text{poly}(n)$. We now consider two cases: Either for this input length ℓ it is the case that NC circuits can distinguish encodings of 0 and 1 with probability at least 3/4, or for this input length ℓ it is the case that NC circuits can distinguish encodings of 0 and 1 with probability at most $1 - 1/\text{poly}(n)$. If we are in the first case, then we can actually honestly run the attack using a NC circuit (in this case we just use the distinguisher to guess the value of the encoding and succeed with probability 3/4). If we are in the second case, then we can use Impagliazzo's Hard Core Set [58] to find a set of encodings such that a NC circuit can distinguish random encodings of 0 and 1 from this set with probability at most $1/2 + 1/\text{poly}(n)$. In this case, we modify the tampering function to hardcode random encodings *from the hard core set* and return these in response to the queries from R. Note that to obtain contradiction to the security of the constructed non-malleable code, we now require that when the reduction R is composed with any tampering circuit in NC, then the composed circuit is still in NC. This property holds for NC, but not NC^1, which is why our result on ruling out non-security preserving reductions holds only for NC.

1.4 Related Work

Non-Malleable Codes. Non-malleable codes (NMC) were introduced in the seminal work of Dziembowski, Pietrzak and Wichs [44]. 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 [70] 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 [2] as well as information theoretic setting with a series of advances—reduced number of states, improved rate, or adding desirable features to the scheme [43, 4, 31, 3, 9, 2, 26, 62, 67, 68, 7]. Recently efficient NMC have been constructed for tampering function classes other than split-state tampering [17, 8, 29, 46, 18, 15, 16, 19, 32] in both the computational and information-theoretic setting. Additionally, [44, 33, 49] 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 [47, 34, 24, 6, 61, 40, 48, 3, 25, 23, 64, 38, 5, 39, 63, 72, 65, 45, 27, 30, 75, 35].

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 [44, 33, 23, 38, 37], most of the effort has been focused on understanding and/or improving the bounds on the rates of NMC [2, 8, 9, 62, 68, 35]

Perhaps the closest to this work are the results of [33, 38, 37]. Cheragachi and Guruswami [33] 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 $\mathcal{F}$ of size $|\mathcal{F}|$ if loglog$|\mathcal{F}| \leq \alpha n$ for $0 \leq \alpha < 1$, moreover these NMC have optimal rate of $1 - \alpha$ with error $\varepsilon \in O(1/\mathsf{poly}(n))$. Dachman-Soled, Kulkarni, and Shahverdi [38] 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 [40] construction to match the lower bound in computational setting. Recently, Dachman-Soled and Kulkarni [37] 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. [47] showed the impossibility of constructing information-theoretically secure 2-split-state CNMC.

Black-Box Separations. Impagliazzo and Rudich [59] 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 [80] separating collision resistant hash functions from one way functions, and [55] ruling out oblivious transfer from public key encryption. The meta-reduction technique (cf. [36, 73, 53, 50, 74, 54, 1, 78, 20, 52]) 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 $\| x \|_0 := |\{x_i : x_i \neq 0 \,\text{for}\, i \in [n]\}|$, 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 ε-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 ε-close, denoted by $X \approx_\varepsilon Y$, if $\Delta(X, Y) \leq \varepsilon$.

2.2 Non-Malleable Codes

Definition 1 (Coding Scheme [44]). *A* Coding scheme*,* (E, D)*, consists of a (possibly randomized) encoding function* $\mathsf{E} : \{0,1\}^k \to \{0,1\}^n$ *and a deterministic decoding function* $\mathsf{D} : \{0,1\}^n \to \{0,1\}^k \cup \{\perp\}$ *such that* $\forall m \in \{0,1\}^k, \Pr[\mathsf{D}(\mathsf{E}(m)) = m] = 1$ *(over randomness of* E*).*

We define the distance of a *randomized* coding scheme by considering the minimum distance of all codes formed as follows: for each message $x \in \{0,1\}^k$ choosing an arbitrary codeword corresponding to that message, $c_x \in \{\mathsf{E}(x; r)\}_{r \in \{0,1\}^*}$. We take the distance of the randomized encoding scheme to be the maximum of all such minimum distances (i.e. the distance of the best sub-code).

Definition 2 (Distance of a Coding Scheme). *The* distance *of a (randomized) coding scheme,* (E, D)*, is*

$$\max_{\substack{S \subset \{c = \mathsf{E}(x;r) : x \in \{0,1\}^k, r \in \{0,1\}^*\}: \\ \forall x \in \{0,1\}^k, \exists c_x \in S: \Pr[\mathsf{D}(c_x) = x] > 1/2}} \quad \min_{\substack{c_x, c_y \in S: \\ x \neq y \in \{0,1\}^k}} \|c_x - c_y\|_0$$

Note that this definition can be extended to arbitrary alphabets. Moreover, it is clear that the minimum distance any coding scheme with K messages and codeword space Σ^n is upper bounded by the maximum of the traditional notion of minimum distance in (non-randomized) codes with the same parameters: the minimum distance between codewords from a code (over Σ^n with K distinct code words).

Definition 3 (ε-Non-malleability [44]). *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:*

$$\mathbf{Tamper}_m^f \stackrel{\text{def}}{=} \left\{ \begin{matrix} c \leftarrow \mathsf{E}(m), \tilde{c} := f(c), \tilde{m} := \mathsf{D}(\tilde{c}). \\ Output: \ \tilde{m}. \end{matrix} \right\},$$

where the randomness of the experiment comes from E*. We say a coding scheme* (E, D) *is ε-non-malleable with respect to $\mathcal{F}$ if for each $f \in \mathcal{F}$, there exists a distribution D^f over $\{0,1\}^k \cup \{\mathsf{same}^*, \perp\}$ such that for every message $m \in \{0,1\}^k$, we have*

$$\mathbf{Tamper}_m^f \approx_\varepsilon \left\{ \begin{matrix} \tilde{m} \leftarrow D^f. \\ Output: m \text{ if } \tilde{m} = \mathsf{same}^*; \\ \text{otherwise } \tilde{m}. \end{matrix} \right\}$$

Here the indistinguishability can be either statistical or computational.

Lemma 1 (Lemma 2 [43]). *Let* (E, D) *be a coding scheme with* $\mathsf{E} : \{0,1\} \to \mathcal{X}$ *and* $\mathsf{D} : \mathcal{X} \to \{0,1\}$*. Let $\mathcal{F}$ be a set of functions $f : \mathcal{X} \to \mathcal{X}$. Then* (E, D) *is ε-non-malleable with respect to $\mathcal{F}$ if and only if for every $f \in \mathcal{F}$,*

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

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

Definition 4 (ε-Malleable Code).

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

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

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 [12].

Definition 5 ([17]). *We say that a bit x_i* affects *the boolean function f, if $\exists \{x_1, x_2, \cdots x_{i-1}, x_{i+1}, \cdots x_n\} \in \{0,1\}^{n-1}$ such that, $f(x_1, x_2, \cdots x_{i-1}, 0, x_{i+1}, \cdots x_n) \neq f(x_1, x_2, \cdots x_{i-1}, 1, x_{i+1}, \cdots x_n)$.*

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 6 (Input Locality [17]). *A function $f : \{0,1\}^n \to \{0,1\}^n$ is said to have* input locality *ℓ if every input bit f_i is affects at most ℓ output bits.*

Definition 7 (Output Locality [17]). *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 8 (Input Local Functions [12]). *A function $f : \{0,1\}^n \to \{0,1\}^n$ is said to be* ℓ-input local, *$f \in \mathrm{Local}_\ell$, if it has input locality ℓ.*

Definition 9 (Output Local Functions [12]). *A function $f : \{0,1\}^n \to \{0,1\}^n$ is said to be* m-output local, *$f \in \mathrm{Local}^m$, if it has output locality m.*

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

Definition 10 (Pseudorandom Generator [41]). *Let $n, n' \in \mathbb{N}$ such that $n' > n$, and let $\mathsf{PRG} = \{\mathsf{prg}_n : \{0,1\}^n \to \{0,1\}^{n'}\}$ be a family of deterministic functions which can be computed in computational class $\mathcal{C}_1$. We say PRG is a $\mathcal{C}_1$-*pseudorandom generator *for $\mathcal{C}_2$ if for any $D := \{D_n : \{0,1\}^{n'} \to \{0,1\}\} \in \mathcal{C}_2$:*

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

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

For class $\mathcal{C}$, if $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C}$ then we simply call it $\mathcal{C}$-pseudorandom generator.

2.4 Distributional Problems

Definition 11 (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 Ψ_n is probability distribution over $\{0,1\}^n$. The decision problem is to decide whether $s \in L_n$ where, $s \leftarrow \Psi_n$. For a string $s \in \{0,1\}^n$, we define $L(s)$ to output 1, if and only if $s \in L_n$.*

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 P if $L \in \mathsf{P}$. We say it is efficiently samplable if there is a randomized poly-time algorithm that on input 1^n samples Ψ_n.

Definition 12 ($\varepsilon(n)$-Hard Distributional Problem). *Let ($\Psi = \{\Psi_n\}_{n=1}^{\infty}, L = \{L_n\}_{n=1}^{\infty}$) be a distributional problem, we say that (Ψ, L) is $\varepsilon(n)$-hard for family of boolean circuits $\mathcal{C} = \{C_n\}_{n=1}^{\infty}$, if and only if for every circuit $C_n \in \mathcal{C}$,*

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

Definition 13 ($\varepsilon(n)$-Easy Distributional Problem). *Let ($\Psi = \{\Psi_n\}_{n=1}^{\infty}, L = \{L_n\}_{n=1}^{\infty}$) be a distributional problem, we say that (Ψ, L) is $\varepsilon(n)$-easy for family of boolean circuits $\mathcal{C} = \{C_n\}_{n=1}^{\infty}$, if there exists some circuit $C_n \in \mathcal{C}$,*

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

2.5 Hardness of Boolean Functions and Composition

Definition 14 (δ-hardness of boolean function). *Let $f : \{0,1\}^n \to \{0,1\}$ be a boolean function, and U_n be uniform distribution over $\{0,1\}^n$. Also let $0 < \delta < \frac{1}{2}$, and $n \leq s \leq \frac{2^n}{n}$. We say f* is δ-hard for size *s if for any boolean circuits C of size at most s*

$$\Pr_{x \leftarrow U_n}[C(x) = f(x)] \leq 1 - \delta$$

.

Definition 15 (ε-hard-core function). *Let $f : \{0,1\}^n \to \{0,1\}$ be a boolean function, and D_S be a distribution over $\{0,1\}^n$ induced by the characteristic function of set $S \subset \{0,1\}^n$[4]. We call f* ε-hard-core *on S for size s (where $n \leq s \leq \frac{2^n}{n}$), if for any boolean circuits C of size at most s*

$$\Pr_{x \leftarrow D_S}[C(x) = f(x)] < \frac{1}{2} \cdot (1 + \varepsilon)$$

.

We also present the following theorem from [58].

Theorem 1 (Theorem 1 [58]). *Let $f : \{0,1\}^n \to \{0,1\}$ be boolean function that δ-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 ε-hard-core on S for circuits of size $s' \leq c \cdot \varepsilon^2 \cdot \delta^2 \cdot s$.*

Definition 16 (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 $\mathsf{MAJ}(C_1, \ldots, C_{p(n)}) \in \mathcal{F}_n$.*

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

[4] Characteristic function of set S outputs 1 if the input to the function is in set S.

Definition 17 (Unroll **functionality.**). *Let $F := \{f_n\}_{n=1}^{\infty} \in \mathcal{F}$, where $f_n : \{0,1\}^n \to \{0,1\}$ and $G = \{g_m\}_{m=1}^{\infty} \in \mathcal{G}$, where $g_m : \{0,1\}^m \to \{0,1\}^m$, be function families. Also let t, p be polynomials. Let $m \in \mathrm{poly}(n)$. Let F^G denote families functions $f_n : \{0,1\}^n \to \{0,1\} \in F$ which contains at most $t(n)$ oracle gates computing $g_m : \{0,1\}^m \to \{0,1\}^m \in G$ 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 (a topological ordering of) 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 $\mathsf{Unroll}(F^G)$, is an unrolling of $F^G(x)$.*

Definition 18 (Replace **Functionality.**). *Consider replacing each g_{n_i}, $i \in [t(n)]$, in $\mathsf{Unroll}(F^G)$ with a circuit g'_{n_i} that takes input $(x_1, \ldots, x_i)$ and produces output y_i. This is denoted by $\mathsf{Replace}(\mathsf{Unroll}(F^G), g'_{n_1}, \ldots, g'_{n_{t(n)}})$.*

Definition 19 **($(\mathcal{G}, t, \ell, u)$-closure of $\mathcal{F}$).** *Let $F := \{f_n\}_{n=1}^{\infty} \in \mathcal{F}$, where $f_n : \{0,1\}^n \to \{0,1\}$ and $G = \{g_m\}_{m=1}^{\infty} \in \mathcal{G}$, where $g_m : \{0,1\}^m \to \{0,1\}^m$, be function families. Also let t, ℓ, u be polynomials, and $\ell(n) \le m \le u(n)$. Let ${f_n}^{g_m}$ denote function $f_n : \{0,1\}^n \to \{0,1\}$ which has access to the output of $g_m : \{0,1\}^m \to \{0,1\}^m$ on at most $t(n)$ inputs of its choice.*

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

Definition 20 **($(\mathcal{G}, t)$-closure of $\mathcal{F}$ under Strong Composition).** *Let $F := \{f_n\}_{n=1}^{\infty} \in \mathcal{F}$, where $f_n : \{0,1\}^n \to \{0,1\}$ and $G = \{g_m\}_{m=1}^{\infty} \in \mathcal{G}$, where $g_m : \{0,1\}^m \to \{0,1\}^m$, be function families. Also let t, p be polynomials. Let $m \in \mathrm{poly}(n)$. Let F^G denote families functions $f_n : \{0,1\}^n \to \{0,1\} \in F$ which contains at most $t(n)$ oracle gates computing $g_m : \{0,1\}^m \to \{0,1\}^m \in G$*

We say that $\mathcal{F}$ is $(\mathcal{G}, t)$-closed under compositions if for every $F \in \mathcal{F}$ we have that for all $G, G' \in \mathcal{G}$ and all $g'_1, \ldots, g'_{t(n)} \in G'$, $\mathsf{Replace}(\mathsf{Unroll}(F^G), g'_1, \ldots, g'_{t(n)}) \in \mathcal{F}$.

2.6 Black Box Reductions

Definition 21 (Black-Box-Reduction). *We say R is an (F, ϵ, δ)*-black-box reduction from a (single bit) non-malleable code, $(\mathsf{E}, \mathsf{D}) = \{(\mathsf{E}_n, \mathsf{D}_n)\}_{n=1}^{\infty}$, 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 ϵ, i.e.*

$$\Pr_{b \xleftarrow{u} \{0,1\}}[\mathsf{D}_n(f_n(\mathsf{E}_n(b))) = 1 - b] > \frac{1}{2} + \epsilon(n),$$

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

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

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

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,*

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

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,*

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

We say a reduction is in NC^1 if it can be written as a family of circuits of $O(\log n)$-depth, 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 $b_1, \dots, b_d$, 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 3–are identical). Applying Lemma 1, it is sufficient to show that for every tampering function f that modifies at most $d-1$ symbols,

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

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

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

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,

$$\begin{aligned}
\Pr_{b \leftarrow \{0,1\}}[\mathsf{D}(f(\mathsf{E}(b))) = 1 - b] &= \frac{1}{2}\Pr[\mathsf{D}(f(\mathsf{E}(0))) = 1] + \frac{1}{2}\Pr[\mathsf{D}(f(\mathsf{E}(1))) = 0] \\
&= \frac{1}{2}\Pr[\mathsf{D}(f(\mathsf{E}(0))) = 1] \\
&\leq \frac{1}{2}.
\end{aligned}$$

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 Max Minimum Distance

Let (E, D) be a coding scheme with distance d. Define the class of functions $\mathcal{F}_{d/2-1} = \{f : f$ changes $< d/2$ 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 $\mathcal{F}_{d/2} = \{f : f$ changes $\leq d/2$ symbols$\}$. In Theorem 2 we show that even inefficient NMC do not exist for $\mathcal{F}_{d/2}$. Recall that distance for randomized coding schemes is upper bounded by the notion of distance for (non-randomized) codes with the same message/codeword-space parameters. Specifically, for a set of codewords S, we define the distance of S ($\mathsf{dist}(S)$) as the minimum pairwise distance over all pairs of codewords in S. Let $\mathcal{S}$, be the set that consists of all sets S that contain exactly one codeword for each message in the message space. Then the distance of the code is defined as $\max_{S \in \mathcal{S}} \mathsf{dist}(S)$. Refer to definition 2 for formal definition of distance of coding scheme, presented earlier. Intuitively, we want that a code with distance d, ensures that any 2 codewords which are at least d distance apart decode to distinct messages. Therefore, first consider a set of codewords which contains at least one codeword corresponding to each message in the message space such that decoding that specific codeword returns the corresponding message with probability greater than $1/2$. Now consider the minimum of pairwise distances for the codewords in this set (say d'). Note that, if the distance of the code is set to d', then for this particular set, we will ensures that any 2 codewords which are at least d' apart decode to distinct messages with high probability. Further, if we take the maximum over all such distances d' corresponding to each set of codewords and call that value d as the distance of the code, then for any 2 codewords which are at least d apart decode to distinct messages with high probability.

Theorem 2. *Let* (E, D) *be a coding scheme with message space of size greater than* 2*, alphabet* Σ *and distance* d*. Then, for any* $\epsilon > 0$*,* (E, D) *is not a* $\frac{1}{8} - \epsilon$*-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 $(\mathsf{E} : U \to V, \mathsf{D} : V \to U)$ be a randomized encoding scheme, where $U \subseteq \Sigma^k, V \subseteq \Sigma^n$ and $|U| > 2$.

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

1. $\|c_x - z\|_0 \leq \frac{d}{2}$
2. $\Pr[\mathsf{D}(z) \neq x] = \geq \frac{1}{2}$.

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

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

Let $\Pr_{c_{x^*}\leftarrow \mathsf{E}(x^*)}[\mathsf{D}(z(c_{x^*})) \neq x^*] = p \geq \frac{1}{2}$. Then, $\exists y^* \neq x^* \in U$ such that $\Pr_{c_{x^*}\leftarrow \mathsf{E}(x^*)}[\mathsf{D}(z(c_{x^*})) = y^*] \leq \frac{p}{|U|-1}$, but $\Pr[\mathsf{D}(f(\mathsf{E}(y^*))) = y^*] = 1$. This means that a distribution $D^f_{x^*}$ that exactly agrees with $\mathsf{D}(f(\mathsf{E}(\cdot)))$ on x^* must output same^* or x^* with probability $1-p$ and y^* with probability at most $\frac{p}{|U|-1}$. A distribution $D^f_{y^*}$ that exactly agrees with $\mathsf{D}(f(\mathsf{E}(\cdot)))$ on y^* must output same^* or y^* with probability 1. Thus, any distribution D^f can only agree with $\mathsf{D}(f(\mathsf{E}(\cdot)))$ 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 $\forall x \in U, \exists c_x \in \mathsf{E}(x)$ such that $\forall z \in V$ with $\|c_x - z\|_0 \leq \frac{d}{2}$ it is the case that $\Pr[\mathsf{D}(z) \neq x] < \frac{1}{2}$. Fix any such set of codewords corresponding to all messages $S = \{c_x : \forall z \in V\, \|c_x - z\|_0 \leq \frac{d}{2} \implies \Pr[\mathsf{D}(z) \neq x] < \frac{1}{2}\}_{x\in U}$. Note that the distance of S ($\min_{c_x \neq c_y \in S} \|c_x - c_y\|_0$) is at most d (by definition of the distance of a randomized code). Let $c_x \neq c_y \in U$ be two such codewords such that $\|c_x - c_y\| \leq d$. 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 $\Pr[\mathsf{D}(z) = x] > \frac{1}{2}$ and $\Pr[\mathsf{D}(z) = y] > \frac{1}{2}$, which is a contradiction because $x \neq y$.

We observe that a randomized coding scheme with message space $\mathcal{M}$, codeword space $\mathcal{C}$ and distance d (as defined above for randomized coding schemes), implies a (possibly inefficient) error correcting code with the same message/codeword space that can correct up to $d/2-1$ errors. To see this, note that one can take the set $S \in \mathcal{S}$ (as in the definition of distance for randomized coding schemes) achieving the maximum $\mathsf{dist}(S)$. Since $S \subseteq \mathcal{C}$ contains exactly one codeword for each message in the message space, the set S itself comprises a code with message space $\mathcal{M}$, codeword space $\mathcal{C}$ and distance d. This, in turn, implies a (possibly inefficient) error correcting code with message space $\mathcal{M}$ and codeword space $\mathcal{C}$ that can correct up to $d/2 - 1$ errors. We thus obtain the following corollary:

Corollary 1. *Fix a message space $\mathcal{M}$ and a codeword space $\mathcal{C}$. If the optimal (inefficient) error-correcting code for $(\mathcal{M}, \mathcal{C})$ can correct at most t errors, then there is no non-malleable code with message space $\mathcal{M}$ and codeword space $\mathcal{C}$ against tampering class $\mathcal{F}_{t+1}$.*

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 ℓ output symbols and ℓ 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 Σ. Then, for any $\epsilon > 0$, (E, D) is not a $1/2 - \epsilon$-NMC for $Local_1$.*

Proof. Let $U \subseteq \{0,1\}^k$, $V \subseteq \{0,1\}^n$ where $|U| > 1$. Let $(\mathsf{E} : U \to V, \mathsf{D} : V \to U)$ be non-malleable code. Take $x \neq y \in U$. Consider $c_x = \mathsf{E}(x), c_y = \mathsf{E}(y)$ for some fixed randomness. By correctness

$c_x \neq c_y$ and moreover, $\mathsf{D}(c_x) \neq \mathsf{D}(c_y)$. 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

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

Let $x = \mathsf{D}(c_j)$ and let $y = \mathsf{D}(c_{j+1})$, where $x \neq y$. Now, consider the following $f \in \text{Local}_1$,

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

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

4.3 Functions with Large Output Locality

A function $f : \{0,1\}^n \to \{0,1\}^n$ 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 $4^n \cdot 2^{2^{n-\log(n)}}$, so $\log\log|\mathcal{F}'| = n - \log(n)$. On the other hand, Dziembowski et al. [44] showed existence of a $1/n$-non-malleable code for any class $\mathcal{F}$ such that $\log\log|\mathcal{F}| \leq n - 2\log(n)$. Thus, our lower bound result is nearly tight matching the existential upper bound. In our theorem, we prove a more general statement:

Theorem 4. *Let (E, D) be a coding scheme with $\mathsf{E} : \{0,1\} \to \{0,1\}^n$ and $\mathsf{D} : \{0,1\}^n \to \{0,1\}$. Let $\mathcal{F}$ be the class of $(n - \log(1/\epsilon) + 2)$-output- local functions, where $1/8 \geq \epsilon \geq 1/2^n$. Then (E, D) is ϵ-malleable with respect to $\mathcal{F}$.*

Note that the parameters discussed above can be obtained by setting $\epsilon = \frac{1}{4n}$.

Additionally, note that non-malleable codes whose decode function D may output values in $\{0, 1, \perp\}$ *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 $\mathsf{E} : \{0,1\} \to \{0,1\}^n$ and $\mathsf{D} : \{0,1\}^n \to \{0,1\}$. Our analysis considers two cases and shows that for each case, there exists $f \in \mathcal{F}$ such that

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

By Definition 4, 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^{top} (resp. c^{bot}) denote the first $n - \log(1/\epsilon) + 2$ bits (resp. last $\log(1/\epsilon) - 2$ bits) of c. I.e.

$c^{\mathsf{top}} := c_1, \ldots, c_{n-\log(1/\epsilon)+2}$ ($c^{\mathsf{bot}} := c_{n-\log(1/\epsilon)+3}, \ldots, c_n$). 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. Assume $n \geq 2$.

Case 1:

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

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. $\mathsf{D}(c^{*,0}) = 0$ and $\mathsf{D}(c^{*,1}) = 1$.

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 $r \leftarrow U_{\log(1/\epsilon)-2}$, construct circuits f_i for each $i \in [n]$, which take input c^{top} and output c'_i. Each f_i does the following:

- Compute $d := \mathsf{D}(c^{\mathsf{top}}||r)$.
- Output $c'_i = c_i^{*,1-d}$.

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

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

where the two equalities follow from the definition of the tampering function f, and the inequality follows since we are in Case 1. This implies the ϵ-malleability of (E, D).

Case 2:

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

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 $n - \log(1/\epsilon) + 2$ circuits $(f_1, \ldots, f_{n-\log(1/\epsilon)+2})$ simply compute the identity function: I.e. f_i for $i \in [n - \log(1/\epsilon) + 2]$ takes c_i as input and produces c_i as output.

We next describe the distribution over circuits f_i for $i \in \{n - \log(1/\epsilon) + 3, \ldots, n\}$. Sample $r' \leftarrow [1/(4\epsilon) - 1]$. Construct circuits f_i for each $i \in \{n - \log(1/\epsilon) + 3, \ldots, n\}$ that take input c^{bot} and produce output c'_i. Each f_i does the following:

- Let $r := r_{n-\log(1/\epsilon)+3}, \ldots, r_n$ be the r'-th lexicographic string in the set $S_{\log(1/\epsilon)-2} \setminus \{c^{\mathsf{bot}}\}$[5].
- Output $c'_i = r_i$.

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

[5] Recall that, $t \in \mathbb{N}$, let S_t denote the set of all t-bit strings and let U_t denote the uniform distribution over t bits.

Since we are in Case 2 we have that:

$$\begin{aligned}
1/2 - \epsilon &\leq \Pr_{b \leftarrow \{0,1\}}[\mathsf{D}(c^{\mathsf{top}}||r) = 1 - b \mid c \leftarrow \mathsf{E}(b), r \leftarrow U_{\log(1/\epsilon)-2}] \\
&= \Pr_{b \leftarrow \{0,1\}}\left[\begin{array}{c} c^{\mathsf{bot}} = r \mid \\ c \leftarrow \mathsf{E}(b), r \leftarrow U_{\log(1/\epsilon)-2} \end{array}\right] \cdot \Pr_{b \leftarrow \{0,1\}}\left[\begin{array}{c} \mathsf{D}(c^{\mathsf{top}}||r) = 1 - b \mid \\ c^{\mathsf{bot}} = r \wedge c \leftarrow \mathsf{E}(b), r \leftarrow U_{\log(1/\epsilon)-2} \end{array}\right] \\
&\quad + \Pr_{b \leftarrow \{0,1\}}\left[\begin{array}{c} c^{\mathsf{bot}} \neq r \mid \\ c \leftarrow \mathsf{E}(b), r \leftarrow U_{\log n} \end{array}\right] \cdot \Pr_{b \leftarrow \{0,1\}}\left[\begin{array}{c} \mathsf{D}(c^{\mathsf{top}}||r) = 1 - b \mid \\ c^{\mathsf{bot}} \neq r \wedge c \leftarrow \mathsf{E}(b), r \leftarrow U_{\log(1/\epsilon)-2} \end{array}\right] \\
&= \Pr_{b \leftarrow \{0,1\}}\left[\begin{array}{c} c^{\mathsf{bot}} = r \mid \\ c \leftarrow \mathsf{E}(b), r \leftarrow U_{\log(1/\epsilon)-2} \end{array}\right] \cdot 0 \\
&\quad + \Pr_{b \leftarrow \{0,1\}}\left[\begin{array}{c} c^{\mathsf{bot}} \neq r \mid \\ c \leftarrow \mathsf{E}(b), r \leftarrow U_{\log(1/\epsilon)-2} \end{array}\right] \cdot \Pr_{b \leftarrow \{0,1\}}\left[\begin{array}{c} \mathsf{D}(c^{\mathsf{top}}||r) = 1 - b \mid \\ c^{\mathsf{bot}} \neq r \wedge c \leftarrow \mathsf{E}(b), r \leftarrow U_{\log(1/\epsilon)-2} \end{array}\right] \\
&= \Pr_{b \leftarrow \{0,1\}}\left[\begin{array}{c} c^{\mathsf{bot}} \neq r \mid \\ c \leftarrow \mathsf{E}(b), r \leftarrow U_{\log(1/\epsilon)-2} \end{array}\right] \cdot \Pr_{b \leftarrow \{0,1\}}\left[\begin{array}{c} \mathsf{D}(c^{\mathsf{top}}||r) = 1 - b \mid \\ c^{\mathsf{bot}} \neq r \wedge c \leftarrow \mathsf{E}(b), r \leftarrow U_{\log(1/\epsilon)-2} \end{array}\right] \\
&= (1 - 4\epsilon) \cdot \Pr_{b \leftarrow \{0,1\}}\left[\begin{array}{c} \mathsf{D}(c^{\mathsf{top}}||r) = 1 - b \mid \\ c^{\mathsf{bot}} \neq r \wedge c \leftarrow \mathsf{E}(b), r \leftarrow U_{\log(1/\epsilon)-2} \end{array}\right].
\end{aligned}$$

Note that

$$\Pr_{b \leftarrow \{0,1\}}\left[\begin{array}{c} \mathsf{D}(c^{\mathsf{top}}||r) = 1 - b \mid \\ c^{\mathsf{bot}} \neq r \wedge c \leftarrow \mathsf{E}(b), r \leftarrow U_{\log(1/\epsilon)-2} \end{array}\right] = \Pr_{b \leftarrow \{0,1\}}[\mathsf{D}(f(\mathsf{E}(b))) = 1 - b].$$

Thus, we have that

$$1/2 - \epsilon \leq (1 - 4\epsilon) \Pr_{b \leftarrow \{0,1\}}[\mathsf{D}(f(\mathsf{E}(b))) = 1 - b].$$

Since

$$\begin{aligned}
(1/2 + \epsilon) \cdot (1 - 4\epsilon) &= 1/2 + \epsilon - 2\epsilon - 4\epsilon^2 \\
&\leq 1/2 - \epsilon,
\end{aligned}$$

we have that

$$\begin{aligned}
\Pr_{b \leftarrow \{0,1\}}[\mathsf{D}(f(\mathsf{E}(b))) = 1 - b] &\geq \frac{1/2 - \epsilon}{1 - 4\epsilon} \\
&\geq 1/2 + \epsilon.
\end{aligned}$$

This implies the ϵ-malleability of (E, D).

5 On NMC via BB Reductions

For the formal definition of a *(F, ϵ, δ)-black-box reduction from a (single bit) non-malleable code, $(\mathsf{E}, \mathsf{D}) = \{(\mathsf{E}_n, \mathsf{D}_n)\}_{n=1}^{\infty}$, to a distributional problem, $(\Psi, L) = \{(\Psi_n, L_n)\}_{n=1}^{\infty}$*, see Definition 21 in Section 2.6.

A crucial component of our impossibility result will be a lookup circuit that responds to queries submitted by the reduction with hardwired responses. However, we need the lookup circuit to maintain consistency: If the reduction queries the same query multiple times, the same response should be given each time. Such a lookup circuit is trivial to implement with polynomial-size circuits. However, in our case, we require that this lookup circuit is implementable in NC^1. In the following, we first formally define such a lookup circuit and then prove that it is implementable in NC^1.

Definition 22 (Look-Up Circuit.). *A $(\ell(n),\ p(n))$ lookup circuit consists of $\ell(n)$ hardwired values of length $p(n)$ bits, denoted $y_1, \ldots, y_{\ell(n)}$. The lookup circuit receives as input $x_1, \ldots, x_{\ell(n)}$, where each x_i has length $p(n)$ bits. The circuit outputs $\ell(n)$ number of $p(n)$-bit strings: $y_{i_1}, \ldots, y_{i_{\ell(n)}}$, where for $j \in [\ell(n)]$, i_j is set to the first index $k \in [\ell(n)]$ such that $x_j = x_k$. For example, on input $x_1, x_2, x_3, x_4, \ldots,$ where $x_1 = x_3$ and $x_2 = x_4$, the circuit outputs y_1, y_2, y_1, y_2.*

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 $O(\log n)$ [10]. 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_ℓ 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_ℓ.

We now present the central technical lemma of the section.

Lemma 2. *Assume that $\mathcal{F}$ is $(\mathcal{F}, t, p(n), p(n))$-closed under composition (see Definition 19), and contains $(t(n), p(n))$ look-up circuits for polynomials $t(\cdot)$, $p(\cdot)$.*[6] *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 $\mathcal{F}$, $(\mathsf{E}, \mathsf{D}) = \{(\mathsf{E}_n, \mathsf{D}_n)\}_{n=1}^{\infty}$, 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 $\mathcal{F}$.*
2. *(Ψ, L) is $(\delta(n)/2)$-easy for $\mathcal{F}$.*

Moreover, if (E, 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 (E, D) and distributional problem (Ψ, 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 $H : \{0,1\}^{p(n)} \to \{0,1\}^*$ and $H(c)$ is the randomness used by encoding algorithm):

$$f_{p(n)}(c) := \begin{cases} \mathsf{E}_n(1; H(c)) \text{ if } \mathsf{D}_n(c) = 0 \\ \mathsf{E}_n(0; H(c)) \text{ if } \mathsf{D}_n(c) = 1 \end{cases}$$

[6] $p(n)$ corresponds to the length of the codeword outputted by E_n.

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

$$\Pr_{b \xleftarrow{u} \{0,1\}} [\mathsf{D}_n(f_{p(n)}(\mathsf{E}(b))) = 1 - b] = 1.$$

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 $f'^{,j}_{p(n)}$, a stateful simulation of the output of the tampering function $f_{p(n)}$ on the j-th query. Each $f'^{,j}_{p(n)}$ is a $(j, p(n))$ lookup circuit (with j number of inputs/outputs of length $p(n)$) that hardcodes a random codeword (sampled from $\mathsf{E}(b)$ where b is uniform) as the y_j value.

By our assumption on $\mathcal{F}$ (and R), we have that $\mathsf{Replace}(\mathsf{Unroll}(R^{f_{p(n)}}), f'^{,1}_{p(n)}, \ldots, f'^{,t(n)}_{p(n)}) \in \mathcal{F}$. We will abuse notation and denote the resulting circuit by $R^{f'_{p(n)}}$. So, it suffices to show that the behavior of $R^{f'_{p(n)}}(x)$ is close that of $R^{f^H_{p(n)}}(x)$, for any x, which will imply that $R^{f'_{p(n)}}(x) \in \mathcal{F}$ breaks the distributional problem w.h.p., since $R^{f^H_{p(n)}}(x)$ does. More accurately, if (E, D) is $\frac{\delta(n)}{2t(n)}$-non-malleable by $\mathcal{F}$, then we will show that

$$\forall n \in \mathbb{N}, \forall x \in \{0,1\}^n, \Delta(R^{f'_n}(x); R^{f_n}(x)) \leq \delta(n)/2.$$

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

To show that the outputs of $R^{f'_{p(n)}}(x)$ and $R^{f^H_{p(n)}}(x)$ are close, we will use a hybrid argument, reducing to the $\frac{\delta(n)}{2t(n)}$-non-malleablity of (E, D) at every step.

In the i-th hybrid, the function $f^{(i),j}_{p(n)}$ responding to the j-th query is a $(j, p(n))$ look-up circuit that hardcodes values $y^i_1, \ldots, y^i_j$. For $k \in [t = t(n)]$, the y^i_k values are sampled as follows: For $k \in [t - i]$, y^i_k is sampled as by $f^H_{p(n)}$. For $k > t - i$, y^i_k is a random encoding of a random bit. The concatenation of the t circuits for each query is denoted by $f^{(i)}_{p(n)}$. Clearly, $f^{(0)}_{p(n)} \equiv f_{p(n)}$ and $f^{(t)}_{p(n)} \equiv f'_{p(n)}$.

We will show that for all $x \in \{0,1\}^n$ (and any fixing of random coins r for R) $\Delta(R^{f^{(i)}_{p(n)}}(x); R^{f^{(i-1)}_{p(n)}}(x)) \leq \frac{\delta(n)}{2t(n)}$ (for $i \in [t(n)]$), which proves the claim above.($R^{f^{(0)}_{p(n)}}(x)$ 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 $f^{(i)}_{p(n)}$ and $f^{(i-1)}_{p(n)}$: $|\Pr[R^{f^{(i)}_{p(n)}}(x) = 1] - \Pr[R^{f^{(i-1)}_{p(n)}}(x) = 1]| \geq \frac{\delta(n)}{2t(n)}$.

Note that for fixed random function H (that generates the random coins used to sample the y_j values) $f^{(i)}_{p(n)}$ and $f^{(i-1)}_{p(n)}$ 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 $f^{(i)}_{p(n)}$ and $f^{(i-1)}_{p(n)}$, 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 $R^{f^{(i-1)}_{p(n)}}(x)$ 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 $R^{f^{(i)}_{p(n)}}(x)$, 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 $R^{f^{(i)}_{p(n)}}$, 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^i_{t-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,

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

By standard manipulation, the above is equivalent to:

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

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 $c_{s_{H,x}}$ 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: $\hat{f}_{H,x}$ hardcodes $c_{s_{H,x}}$ and $c_{1-s_{H,x}}$. 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

$$\Pr_{b \leftarrow \{0,1\}}[\mathsf{D}_n(\hat{f}_{H,x}(\mathsf{E}_n(b))) = 1-b].$$

$$\begin{aligned}
\Pr_{b \leftarrow \{0,1\}}[\mathsf{D}(\hat{f}_{H,x}(\mathsf{E}(b))) = 1-b] &= \Pr[b = 1-s_{H,x}] \cdot \Pr[R'_{H,x}(c) = 1 \mid c \leftarrow \mathsf{E}_n(1-s_{H,x})] \\
&\quad + \Pr[b = s_{H,x}] \cdot \Pr[R'_{H,x}(c) = 0 \mid c \leftarrow \mathsf{E}_n(s_{H,x})] \\
&= \frac{1}{2} \cdot \Pr[R'_{H,x}(c) = 1 \mid c \leftarrow \mathsf{E}_n(1-s_{H,x})] \\
&\quad + \frac{1}{2} \cdot \Pr[R'_{H,x}(c) = 0 \mid c \leftarrow \mathsf{E}_n(s_{H,x})] \\
&\geq \frac{1}{2} + \frac{\delta(n)}{2t(n)}.
\end{aligned}$$

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

Therefore, we conclude that either (E,D) is $\frac{\delta(n)}{2t(n)}$-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 NC^1 is $(\mathsf{NC}^1, t, p(n), p(n))$-closed under composition (for all polynomials $p(\cdot)$), and NC^1 contains $(t(n), p(n))$ lookup circuits for any polynomials $t(\cdot)$, $p(\cdot)$.

Corollary 2. *If there is an $(\mathsf{NC}^1, 1/2, \delta(n))$-black-box-reduction making $t(n)$ security parameter preserving queries from a (single bit) non-malleable code for NC^1, $(\mathsf{E},\mathsf{D}) = \{(\mathsf{E}_n,\mathsf{D}_n)\}_{n=1}^{\infty}$, 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 NC^1.*
2. *(Ψ, L) is $(\delta(n)/2)$-easy for 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 $\mathcal{F}$. 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, (Ψ, 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 NC^1 from a distributional problem that is hard for 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 [16]. Briefly, if a function ψ is incompressible by circuit class $\mathcal{F}$, it means that for $t \ll n$, for any *computationally unbounded* Boolean function $D : \{0,1\}^t \to \{0,1\}$ and any $F : \{0,1\}^n \to \{0,1\}^t \in \mathcal{F}$, 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.

Note 2. We can extend Lemma 2 to rule out $(u(n), \ell(n))$-*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, 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 $(\mathcal{F}, t, \ell, u)$-closed under composition and contains look-up circuits with $t(n)$ inputs, consisting of $\ell(n)$ to $u(n)$ number of bits, for polynomials $t(\cdot)$, $\ell(\cdot), u(\cdot)$.

Lemma 3. *Assume $\mathcal{F}$ is $(\mathcal{F}, t, \ell, u)$-closed under composition (see Definition 19) and contains $(t(n), p(n))$ look-up circuits for polynomials $t(\cdot)$, $p(\cdot)$. 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 $\mathcal{F}$, $(\mathsf{E},\mathsf{D}) = \{(\mathsf{E}_n,\mathsf{D}_n)\}_{n=1}^{\infty}$, 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 $\mathcal{F}$.*
2. *(Ψ, 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.*

The following corollary holds since NC^1 is $(\mathsf{NC}^1, t, \ell, u)$-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 NC^1 contains look-up circuits with $t(n)$ number of inputs of length $\ell(n)$ to $u(n)$-bits for polynomials $t(\cdot)$, $\ell(\cdot), u(\cdot)$.

Corollary 3. *Fix constants* $\gamma \leq 1$, $c \geq 1$. *If there is an* $(\mathsf{NC}^1, 1/2, \delta(n))$*-black-box-reduction making* $t(n)$ (n^γ, n^c)*-approximately length preserving queries from a (single bit) non-malleable code for* NC^1, $(\mathsf{E},\mathsf{D}) = \{(\mathsf{E}_n,\mathsf{D}_n)\}_{n=1}^\infty$, *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* NC^1.
2. (Ψ, L) *is* $(\delta(n)/2)$*-easy for* 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 20) and contain* $(t(n), u(n))$ *lookup circuits. If for every non-negligible* ϵ, *there is an* $(\mathcal{F}, \epsilon, \delta(n))$*-black-box-reduction (for some non-negligble* δ*) making* $t(n)$ *queries from an (single bit)* $\epsilon(n)$*-non-malleable code for* $\mathcal{F}$, $(\mathsf{E},\mathsf{D}) = \{(\mathsf{E}_n,\mathsf{D}_n)\}_{n=1}^\infty$, *to a distributional problem,* $(\Psi, L) = \{(\Psi_n, L_n)\}_{n=1}^\infty$, *then* (Ψ, L) *is not* $(\delta(n) - t(n) \cdot \epsilon(n))$*-hard for* $\mathcal{F}$.

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

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

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

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 $k \notin \mathcal{S}$). 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 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 $c \leftarrow \mathsf{E}_{\ell(n)}(b), b \leftarrow \{0,1\}$ 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 $\mathcal{F} := \{\mathcal{F}_n\}_{n \in \mathbb{N}}$ that outputs $\mathsf{D}_{\ell(n)}(c)$ with probability at least $3/4$[7].
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 $\epsilon'(n) \cdot 2^\ell$, where $\epsilon'(n) = \frac{1}{2 \cdot n^c \cdot t(n)}$ such that every adversary in $\mathcal{F} := \{\mathcal{F}_n\}_{n \in \mathbb{N}}$ outputs $\mathsf{D}_{\ell(n)}(c)$ with probability at most $1/2 + \epsilon'(n)$, when c is chosen at random from $\mathcal{H}$[8].

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 ϵ. Since the ϵ function remains the same, we know that δ and ℓ, $\mathcal{S}$, $\mathcal{S}'$ remain the same.

In this case, as in the previous proof, we can switch to a simulated tampering function 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^{Sim}, is a circuit in $\mathcal{F}_n$, since $\mathcal{F}_n$ strongly composes. Additionally, the behavior of R^{Sim} is identical to the behavior of $R^{\{f_k\}_k}$. Moreover, since f_k succeeds with non-negligible ϵ, by assumption on R, it means that for all $n \in \mathcal{S}'$, $R^{f_{\ell(n)}}$ agrees with (Ψ, L) with probability $1/2 + 1/n^c$. But then we must have that for infinitely many $n \in \mathcal{S}'$–input lengths $n \in \mathcal{S}''$–R^{Sim} agrees with (Ψ, L) with probability $1/2 + 1/n^c$ and $R^{\mathsf{Sim}} \in \mathcal{F}_n$. So (Ψ, L) is $(\delta'(n))$-easy for $\mathcal{F}$, where

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

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 decodes 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 $\ell^{-1}(k) \cap \mathcal{S}''$[9], and chooses the lexicographically first set $\mathcal{H}$ of size $\epsilon'(n^*) \cdot 2^{\ell(n^*)} = \epsilon'(n^*) \cdot 2^k$ for which every adversary in $\mathcal{F}_n$ outputs $\mathsf{D}_{\ell(n^*)}(c)$ 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 the ϵ function remains the same in this case as well, the δ function also remains the same. Thus, for $n \in \mathcal{S}'$, $R^{f_{\ell(n)}}$ must still agree with (Ψ, L) with probability $1/2 + 1/n^c$.

In this case, as in the previous proof, we can switch to a simulated tampering function Sim that does not decode but rather chooses a random codeword from the hardcode set $\mathcal{H}$ (which again we can hardcode in using lookup circuits as before). Moreover, for queries R makes to Sim with input lengths $k = k(n) \in O(\log(n))$, all responses for all possible queries c are hardcoded into 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 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

[7] Note that $\mathsf{D}_{\ell(n)}(c)$ 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.

[8] Again, the input c to $\mathsf{D}_{\ell(n)}$ 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.

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

the hardcore set. Finally, we must argue that for infinitely many $n \in \mathcal{S}'$–input lengths $n \in \mathcal{S}''$–R composed with 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 $n \in \mathcal{S}''$, where n is the lexicographically first element in the set $\ell^{-1}(\ell(n)) \cap \mathcal{S}''$. Since n is always contained in $\ell^{-1}(\ell(n))$, since the size of $\ell^{-1}(\ell(n)) \cap \mathcal{S}'$ is finite and since the size of $\mathcal{S}''$ is infinite, there will be infinitely many $n \in \mathcal{S}''$ for which this event occurs. Thus, for infinitely many $n \in \mathcal{S}''$ (denote this set of values by $\widetilde{\mathcal{S}}$, $R^{\{f_k\}_k}$ agrees with (Ψ, L) with probability $1/2 + 1/n^c$ and R^{Sim} is $t(n) \cdot \epsilon'(n) \leq 1/2n^c$-close to $R^{\{f_k\}_k}$. So we conclude that (Ψ, L) is $(\delta'(n))$-easy for $\mathcal{F}$, where

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

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

Corollary 4. *If for every non-negligible $\epsilon = \epsilon(\cdot)$, there is an $(\mathsf{nu}-\mathsf{NC}, \epsilon, \delta)$-black-box-reduction, for some non-negligible $\delta = \delta(\cdot)$, making $t(n)$ queries from a (single bit) non-malleable code for $\mathsf{nu}-\mathsf{NC}$, $(\mathsf{E}, \mathsf{D}) = \{(\mathsf{E}_n, \mathsf{D}_n)\}_{n=1}^{\infty}$, to a distributional problem, $(\Psi, L) = \{(\Psi_n, L_n)\}_{n=1}^{\infty}$, then (Ψ, L) is $(\delta'(n))$-easy for NC, for some non-negligible $\delta' = \delta'(\cdot)$.*

Acknowledgments

The first and fourth authors are supported in part by the Leona M. & Harry B. Helmsley Charitable Trust and the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA) via Contract No. 2019-1902070006. 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.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, DoI/NBC, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

1. Masayuki Abe, Jens Groth, and Miyako Ohkubo. Separating short structure-preserving signatures from non-interactive assumptions. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 628–646. Springer, Heidelberg, December 2011. http://dx.doi.org/10.1007/978-3-642-25385-$0_3 4 doi : 10.1007/978-3-642-25385-0_3 4$.
2. Divesh Aggarwal, Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Optimal computational split-state non-malleable codes. In Kushilevitz and Malkin [66], pages 393–417. http://dx.doi.org/10.1007/978-3-662-49099-$0_1 5 doi : 10.1007/978-3-662-49099-0_1 5$.

3. Divesh Aggarwal, Yevgeniy Dodis, Tomasz Kazana, and Maciej Obremski. Non-malleable reductions and applications. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th ACM STOC*, pages 459–468. ACM Press, June 2015. http://dx.doi.org/10.1145/2746539.2746544 doi:10.1145/2746539.2746544.
4. Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. In David B. Shmoys, editor, *46th ACM STOC*, pages 774–783. ACM Press, May / June 2014. http://dx.doi.org/10.1145/2591796.2591804 doi:10.1145/2591796.2591804.
5. Divesh Aggarwal, Nico Döttling, Jesper Buus Nielsen, Maciej Obremski, and Erick Purwanto. Continuous non-malleable codes in the 8-split-state model. In Ishai and Rijmen [60], pages 531–561. http://dx.doi.org/10.1007/978-3-030-17653-$2_18 doi: 10.1007/978-3-030-17653-2_18$.
6. Divesh Aggarwal, Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Leakage-resilient non-malleable codes. In Dodis and Nielsen [42], pages 398–426. http://dx.doi.org/10.1007/978-3-662-46494-$6_17 doi: 10.1007/978-3-662-46494-6_17$.
7. Divesh Aggarwal and Maciej Obremski. Inception makes non-malleable codes shorter as well! Cryptology ePrint Archive, Report 2019/399, 2019. https://eprint.iacr.org/2019/399.
8. Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Explicit non-malleable codes against bit-wise tampering and permutations. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 538–557. Springer, Heidelberg, August 2015. http://dx.doi.org/10.1007/978-3-662-47989-$6_26 doi: 10.1007/978-3-662-47989-6_26$.
9. Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. A rate-optimizing compiler for non-malleable codes against bit-wise tampering and permutations. In Dodis and Nielsen [42], pages 375–397. http://dx.doi.org/10.1007/978-3-662-46494-$6_16 doi: 10.1007/978-3-662-46494-6_16$.
10. Miklós Ajtai, János Komlós, and Endre Szemerédi. An o(n log n) sorting network. In David S. Johnson, Ronald Fagin, Michael L. Fredman, David Harel, Richard M. Karp, Nancy A. Lynch, Christos H. Papadimitriou, Ronald L. Rivest, Walter L. Ruzzo, and Joel I. Seiferas, editors, *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 1–9. ACM, 1983. URL: https://doi.org/10.1145/800061.808726, http://dx.doi.org/10.1145/800061.808726 doi:10.1145/800061.808726.
11. Marcin Andrychowicz, Ivan Damgård, Stefan Dziembowski, Sebastian Faust, and Antigoni Polychroniadou. Efficient leakage resilient circuit compilers. In Kaisa Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*, pages 311–329. Springer, Heidelberg, April 2015. http://dx.doi.org/10.1007/978-3-319-16715-$2_17 doi: 10.1007/978-3-319-16715-2_17$.
12. Benny Applebaum. *Cryptography in Constant Parallel Time*. Information Security and Cryptography. Springer, 2014. URL: http://dx.doi.org/10.1007/978-3-642-17367-7, http://dx.doi.org/10.1007/978-3-642-17367-7 doi:10.1007/978-3-642-17367-7.
13. Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. In *45th FOCS*, pages 166–175. IEEE Computer Society Press, October 2004. http://dx.doi.org/10.1109/FOCS.2004.20 doi:10.1109/FOCS.2004.20.
14. Benny Applebaum and Pavel Raykov. On the relationship between statistical zero-knowledge and statistical randomized encodings. In Robshaw and Katz [77], pages 449–477. http://dx.doi.org/10.1007/978-3-662-53015-$3_16 doi: 10.1007/978-3-662-53015-3_16$.
15. Marshall Ball, Dana Dachman-Soled, Siyao Guo, Tal Malkin, and Li-Yang Tan. Non-malleable codes for small-depth circuits. In Mikkel Thorup, editor, *59th FOCS*, pages 826–837. IEEE Computer Society Press, October 2018. http://dx.doi.org/10.1109/FOCS.2018.00083 doi:10.1109/FOCS.2018.00083.
16. Marshall Ball, Dana Dachman-Soled, Mukul Kulkarni, Huijia Lin, and Tal Malkin. Non-malleable codes against bounded polynomial time tampering. In Ishai and Rijmen [60], pages 501–530. http://dx.doi.org/10.1007/978-3-030-17653-$2_17 doi: 10.1007/978-3-030-17653-2_17$.
17. Marshall Ball, Dana Dachman-Soled, Mukul Kulkarni, and Tal Malkin. Non-malleable codes for bounded depth, bounded fan-in circuits. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 881–908. Springer, Heidelberg, May 2016. http://dx.doi.org/10.1007/978-3-662-49896-$5_31 doi: 10.1007/978-3-662-49896-5_31$.
18. Marshall Ball, Dana Dachman-Soled, Mukul Kulkarni, and Tal Malkin. Non-malleable codes from average-case hardness: AC^0, decision trees, and streaming space-bounded tampering. In Nielsen and Rijmen [71], pages 618–650. http://dx.doi.org/10.1007/978-3-319-78372-$7_20 doi: 10.1007/978-3-319-78372-7_20$.
19. Marshall Ball, Siyao Guo, and Daniel Wichs. Non-malleable codes for decision trees. *IACR Cryptology ePrint Archive*, 2019:379, 2019.
20. Boaz Barak and Mohammad Mahmoody-Ghidary. Merkle puzzles are optimal - an $O(n^2)$-query attack on any key exchange from a random oracle. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 374–390. Springer, Heidelberg, August 2009. http://dx.doi.org/10.1007/978-3-642-03356-$8_22 doi: 10.1007/978-3-642-03356-8_22$.

21. Nir Bitansky, Shafi Goldwasser, Abhishek Jain, Omer Paneth, Vinod Vaikuntanathan, and Brent Waters. Time-lock puzzles from randomized encodings. In Madhu Sudan, editor, *ITCS 2016*, pages 345–356. ACM, January 2016. http://dx.doi.org/10.1145/2840728.2840745 doi:10.1145/2840728.2840745.
22. Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theor. Comput. Sci.*, 288(1):21–43, 2002. URL: https://doi.org/10.1016/S0304-3975(01)00144-X, http://dx.doi.org/10.1016/S0304-3975(01)00144-X doi:10.1016/S0304-3975(01)00144-X.
23. Nishanth Chandran, Vipul Goyal, Pratyay Mukherjee, Omkant Pandey, and Jalaj Upadhyay. Block-wise non-malleable codes. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *ICALP 2016*, volume 55 of *LIPIcs*, pages 31:1–31:14. Schloss Dagstuhl, July 2016. http://dx.doi.org/10.4230/LIPIcs.ICALP.2016.31 doi:10.4230/LIPIcs.ICALP.2016.31.
24. Nishanth Chandran, Bhavana Kanukurthi, and Rafail Ostrovsky. Locally updatable and locally decodable codes. In Lindell [69], pages 489–514. http://dx.doi.org/10.1007/978-3-642-54242-$8_21 doi : 10.1007/978-3-642-54242-8_21$.
25. Nishanth Chandran, Bhavana Kanukurthi, and Srinivasan Raghuraman. Information-theoretic local non-malleable codes and their applications. In Kushilevitz and Malkin [66], pages 367–392. http://dx.doi.org/10.1007/978-3-662-49099-$0_14 doi : 10.1007/978-3-662-49099-0_14$.
26. Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. In Daniel Wichs and Yishay Mansour, editors, *48th ACM STOC*, pages 285–298. ACM Press, June 2016. http://dx.doi.org/10.1145/2897518.2897547 doi:10.1145/2897518.2897547.
27. Eshan Chattopadhyay, Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, and Sruthi Sekar. Privacy amplification from non-malleable codes. Cryptology ePrint Archive, Report 2018/293, 2018. https://eprint.iacr.org/2018/293.
28. Eshan Chattopadhyay and Xin Li. Non-malleable codes and extractors for small-depth circuits, and affine functions. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1171–1184. ACM, 2017. URL: https://doi.org/10.1145/3055399.3055483, http://dx.doi.org/10.1145/3055399.3055483 doi:10.1145/3055399.3055483.
29. Eshan Chattopadhyay and Xin Li. Non-malleable codes and extractors for small-depth circuits, and affine functions. In Hatami et al. [57], pages 1171–1184. http://dx.doi.org/10.1145/3055399.3055483 doi:10.1145/3055399.3055483.
30. Eshan Chattopadhyay and Xin Li. Non-malleable extractors and codes for composition of tampering, interleaved tampering and more. Cryptology ePrint Archive, Report 2018/1069, 2018. https://eprint.iacr.org/2018/1069.
31. Eshan Chattopadhyay and David Zuckerman. Non-malleable codes against constant split-state tampering. In *55th FOCS*, pages 306–315. IEEE Computer Society Press, October 2014. http://dx.doi.org/10.1109/FOCS.2014.40 doi:10.1109/FOCS.2014.40.
32. Binyi Chen, Yilei Chen, Kristina Hostáková, and Pratyay Mukherjee. Continuous space-bounded non-malleable codes from stronger proofs-of-space. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 467–495. Springer, Heidelberg, August 2019. http://dx.doi.org/10.1007/978-3-030-26948-$7_17 doi : 10.1007/978-3-030-26948-7_17$.
33. Mahdi Cheraghchi and Venkatesan Guruswami. Capacity of non-malleable codes. In Moni Naor, editor, *ITCS 2014*, pages 155–168. ACM, January 2014. http://dx.doi.org/10.1145/2554797.2554814 doi:10.1145/2554797.2554814.
34. Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. In Lindell [69], pages 440–464. http://dx.doi.org/10.1007/978-3-642-54242-$8_19 doi : 10.1007/978-3-642-54242-8_19$.
35. Sandro Coretti, Antonio Faonio, and Daniele Venturi. Rate-optimizing compilers for continuously non-malleable codes. Cryptology ePrint Archive, Report 2019/055, 2019. https://eprint.iacr.org/2019/055.
36. Jean-Sébastien Coron. Security proof for partial-domain hash signature schemes. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 613–626. Springer, Heidelberg, August 2002. http://dx.doi.org/10.1007/3-540-45708-$9_39 doi : 10.1007/3-540-45708-9_39$.
37. Dana Dachman-Soled and Mukul Kulkarni. Upper and lower bounds for continuous non-malleable codes. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part I*, volume 11442 of *LNCS*, pages 519–548. Springer, Heidelberg, April 2019. http://dx.doi.org/10.1007/978-3-030-17253-$4_18 doi : 10.1007/978-3-030-17253-4_18$.
38. Dana Dachman-Soled, Mukul Kulkarni, and Aria Shahverdi. Tight upper and lower bounds for leakage-resilient, locally decodable and updatable non-malleable codes. In Serge Fehr, editor, *PKC 2017, Part I*, volume 10174 of *LNCS*, pages 310–332. Springer, Heidelberg, March 2017. http://dx.doi.org/10.1007/978-3-662-54365-$8_13 doi : 10.1007/978-3-662-54365-8_13$.

39. Dana Dachman-Soled, Mukul Kulkarni, and Aria Shahverdi. Local non-malleable codes in the bounded retrieval model. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 281–311. Springer, Heidelberg, March 2018. http://dx.doi.org/10.1007/978-3-319-76581-$5_10 doi:10.1007/978-3-319-76581-5_10$.
40. 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 [42], pages 427–450. http://dx.doi.org/10.1007/978-3-662-46494-$6_18 doi:10.1007/978-3-662-46494-6_18$.
41. Akshay Degwekar, Vinod Vaikuntanathan, and Prashant Nalini Vasudevan. Fine-grained cryptography. In Robshaw and Katz [77], pages 533–562. http://dx.doi.org/10.1007/978-3-662-53015-$3_19 doi:10.1007/978-3-662-53015-3_19$.
42. Yevgeniy Dodis and Jesper Buus Nielsen, editors. *TCC 2015, Part I*, volume 9014 of *LNCS*. Springer, Heidelberg, March 2015.
43. Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Non-malleable codes from two-source extractors. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 239–257. Springer, Heidelberg, August 2013. http://dx.doi.org/10.1007/978-3-642-40084-$1_14 doi:10.1007/978-3-642-40084-1_14$.
44. Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. *J. ACM*, 65(4):20:1–20:32, April 2018. Extended abstract appeared in Innovations in Computer Science (ICS) 2010. URL: http://doi.acm.org/10.1145/3178432, http://dx.doi.org/10.1145/3178432 doi:10.1145/3178432.
45. Antonio Faonio, Jesper Buus Nielsen, Mark Simkin, and Daniele Venturi. Continuously non-malleable codes with split-state refresh. In Bart Preneel and Frederik Vercauteren, editors, *ACNS 18*, volume 10892 of *LNCS*, pages 121–139. Springer, Heidelberg, July 2018. http://dx.doi.org/10.1007/978-3-319-93387-$0_7 doi:10.1007/978-3-319-93387-0_7$.
46. Sebastian Faust, Kristina Hostáková, Pratyay Mukherjee, and Daniele Venturi. Non-malleable codes for space-bounded tampering. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 95–126. Springer, Heidelberg, August 2017. http://dx.doi.org/10.1007/978-3-319-63715-$0_4 doi:10.1007/978-3-319-63715-0_4$.
47. Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. Continuous non-malleable codes. In Lindell [69], pages 465–488. http://dx.doi.org/10.1007/978-3-642-54242-$8_20 doi:10.1007/978-3-642-54242-8_20$.
48. Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. A tamper and leakage resilient von neumann architecture. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 579–603. Springer, Heidelberg, March / April 2015. http://dx.doi.org/10.1007/978-3-662-46447-$2_26 doi:10.1007/978-3-662-46447-2_26$.
49. Sebastian Faust, Pratyay Mukherjee, Daniele Venturi, and Daniel Wichs. Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 111–128. Springer, Heidelberg, May 2014. http://dx.doi.org/10.1007/978-3-642-55220-$5_7 doi:10.1007/978-3-642-55220-5_7$.
50. Marc Fischlin and Dominique Schröder. On the impossibility of three-move blind signature schemes. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 197–215. Springer, Heidelberg, May / June 2010. http://dx.doi.org/10.1007/978-3-642-13190-$5_10 doi:10.1007/978-3-642-13190-5_10$.
51. Lance Fortnow and Salil P. Vadhan, editors. *43rd ACM STOC*. ACM Press, June 2011.
52. Georg Fuchsbauer, Momchil Konstantinov, Krzysztof Pietrzak, and Vanishree Rao. Adaptive security of constrained PRFs. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 82–101. Springer, Heidelberg, December 2014. http://dx.doi.org/10.1007/978-3-662-45608-$8_5 doi:10.1007/978-3-662-45608-8_5$.
53. Sanjam Garg, Raghav Bhaskar, and Satyanarayana V. Lokam. Improved bounds on security reductions for discrete log based signatures. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 93–107. Springer, Heidelberg, August 2008. http://dx.doi.org/10.1007/978-3-540-85174-$5_6 doi:10.1007/978-3-540-85174-5_6$.
54. Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Fortnow and Vadhan [51], pages 99–108. http://dx.doi.org/10.1145/1993636.1993651 doi:10.1145/1993636.1993651.
55. Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *41st FOCS*, pages 325–335. IEEE Computer Society Press, November 2000. http://dx.doi.org/10.1109/SFCS.2000.892121 doi:10.1109/SFCS.2000.892121.

56. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th FOCS*, pages 464–479. IEEE Computer Society Press, October 1984. http://dx.doi.org/10.1109/SFCS.1984.715949 doi:10.1109/SFCS.1984.715949.
57. Hamed Hatami, Pierre McKenzie, and Valerie King, editors. *49th ACM STOC*. ACM Press, June 2017.
58. Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *36th FOCS*, pages 538–545. IEEE Computer Society Press, October 1995. http://dx.doi.org/10.1109/SFCS.1995.492584 doi:10.1109/SFCS.1995.492584.
59. Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st ACM STOC*, pages 44–61. ACM Press, May 1989. http://dx.doi.org/10.1145/73007.73012 doi:10.1145/73007.73012.
60. Yuval Ishai and Vincent Rijmen, editors. *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*. Springer, Heidelberg, May 2019.
61. Zahra Jafargholi and Daniel Wichs. Tamper detection and continuous non-malleable codes. In Dodis and Nielsen [42], pages 451–480. http://dx.doi.org/10.1007/978-3-662-46494-6$_19doi : 10.1007/978-3-662-46494-6_19$.
62. Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, and Sruthi Sekar. Four-state non-malleable codes with explicit constant rate. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part II*, volume 10678 of *LNCS*, pages 344–375. Springer, Heidelberg, November 2017. http://dx.doi.org/10.1007/978-3-319-70503-3$_11doi : 10.1007/978-3-319-70503-3_11$.
63. Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, and Sruthi Sekar. Non-malleable randomness encoders and their applications. In Nielsen and Rijmen [71], pages 589–617. http://dx.doi.org/10.1007/978-3-319-78372-7$_19doi : 10.1007/978-3-319-78372-7_19$.
64. Aggelos Kiayias, Feng-Hao Liu, and Yiannis Tselekounis. Practical non-malleable codes from l-more extractable hash functions. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 1317–1328. ACM Press, October 2016. http://dx.doi.org/10.1145/2976749.2978352 doi:10.1145/2976749.2978352.
65. Aggelos Kiayias, Feng-Hao Liu, and Yiannis Tselekounis. Non-malleable codes for partial functions with manipulation detection. In Shacham and Boldyreva [79], pages 577–607. http://dx.doi.org/10.1007/978-3-319-96878-0$_20doi : 10.1007/978-3-319-96878-0_20$.
66. Eyal Kushilevitz and Tal Malkin, editors. *TCC 2016-A, Part II*, volume 9563 of *LNCS*. Springer, Heidelberg, January 2016.
67. Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In Hatami et al. [57], pages 1144–1156. http://dx.doi.org/10.1145/3055399.3055486 doi:10.1145/3055399.3055486.
68. Xin Li. Non-malleable extractors and non-malleable codes: Partially optimal constructions. Cryptology ePrint Archive, Report 2018/353, 2018. https://eprint.iacr.org/2018/353.
69. Yehuda Lindell, editor. *TCC 2014*, volume 8349 of *LNCS*. Springer, Heidelberg, February 2014.
70. Feng-Hao Liu and Anna Lysyanskaya. Tamper and leakage resilience in the split-state model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 517–532. Springer, Heidelberg, August 2012. http://dx.doi.org/10.1007/978-3-642-32009-5$_30doi : 10.1007/978-3-642-32009-5_30$.
71. Jesper Buus Nielsen and Vincent Rijmen, editors. *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*. Springer, Heidelberg, April / May 2018.
72. 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 [79], pages 608–639. http://dx.doi.org/10.1007/978-3-319-96878-0$_21doi : 10.1007/978-3-319-96878-0_21$.
73. Pascal Paillier and Damien Vergnaud. Discrete-log-based signatures may not be equivalent to discrete log. In Bimal K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 1–20. Springer, Heidelberg, December 2005. http://dx.doi.org/10.1007/11593447$_1doi : 10.1007/11593447_1$.
74. Rafael Pass. Limits of provable security from standard assumptions. In Fortnow and Vadhan [51], pages 109–118. http://dx.doi.org/10.1145/1993636.1993652 doi:10.1145/1993636.1993652.
75. Peter M. R. Rasmussen and Amit Sahai. Expander graphs are non-malleable codes. Cryptology ePrint Archive, Report 2018/929, 2018. https://eprint.iacr.org/2018/929.
76. Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 1–20. Springer, Heidelberg, February 2004. http://dx.doi.org/10.1007/978-3-540-24638-1$_1doi : 10.1007/978-3-540-24638-1_1$.
77. Matthew Robshaw and Jonathan Katz, editors. *CRYPTO 2016, Part III*, volume 9816 of *LNCS*. Springer, Heidelberg, August 2016.

78. Yannick Seurin. On the exact security of Schnorr-type signatures in the random oracle model. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 554–571. Springer, Heidelberg, April 2012. http://dx.doi.org/10.1007/978-3-642-29011-$4_33doi:10.1007/978-3-642-29011-4_33$.
79. Hovav Shacham and Alexandra Boldyreva, editors. *CRYPTO 2018, Part III*, volume 10993 of *LNCS*. Springer, Heidelberg, August 2018.
80. Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 334–345. Springer, Heidelberg, May / June 1998. http://dx.doi.org/10.1007/BFb0054137 doi:10.1007/BFb0054137.