Coding for interactive communication beyond threshold adversaries

Slava Radune[*]and Anat Paskin-Cherniavsky[†]

July 5, 2017

Abstract

We revisit the setting of coding for interactive communication, CIC, (initiated by Schulman 96') for non-threshold tampering functions. In a nutshell, in the (special case of) the communication complexity setting, Alice and Bob holding inputs x, y wish to compute a function $g(x, y)$ on their inputs over the identity channel using an interactive protocol. The goal here is to minimize the total communication complexity (CC). A "code" for interactive communication is a compiler transforming any π_0 working in the communication complexity setting into a protocol π evaluating the same function over any channel f picked from a family $\mathcal{F}$. Here f is a function modifying the entire communication transcript. The goal here is to minimize the code's *rate*, which is the CC overhead $CC(\pi)/CC(\pi_0)$ incurred by the compiler.

All previous work in coding for interactive communication considered error correction (that is, $g(x, y)$ must be recovered correctly with high probability), which puts a limit of corrupting up to a 1/4 of the symbols (Braverman and Rao 11'). In this work, we initiate the study of CIC for non-threshold families. We first come up with a robustness notion both meaningful and achievable by CIC for interesting *non-threshold* families. As a test case, we consider $\mathcal{F}_{\text{bit}}$, where each bit of the codeword is modified independently of the other bits (and all bits can be modified). Our robustness notion is an enhanced form of error-detection, where the output of the protocol is distributed over $\{\perp, f(x, y)\}$, and the distribution does not depend on x, y. This definition can be viewed as enhancing error detection by non malleability (as in the setting of non-malleable codes introduced by Dzembowski et. al. 10'). We devise CIC for several interesting tampering families (including $\mathcal{F}_{\text{bit}}$). As a building block, we introduce the notion of MNMC (non malleable codes for multiple messages), which may be of independent interest.

Keywords: Error correcting codes, coding for interactive communication, non malleable codes.

1 Introduction

Classical coding theory studies error correcting codes (ECC) $C \subseteq \Sigma^m$ for some finite alphabet, such that any pair $x \neq y \in C$ differs in at least a $d(C)$ fraction of the symbols. In the most useful and hardest case, $\Sigma = \{0, 1\}$. Such codes are useful for transmitting data from Alice to Bob over a noisy channel. In this scenario, C is viewed as injective mapping $\text{Enc}_C : \Sigma^n \to \Sigma^m$, where to send a message m, $C(m)$ is sent, and decoded at the other end (as $C^{-1}(c)$).

ECC's were first studied in the foundational work of Shannon [Sha48] for random noise, and for adversarial noise by Hamming [Ham50]. Explicit constructions with constant rate $\log |C|/m$ (arguably, the most important complexity measure of codes) have been subsequently devised [Ree60, AGM92], to name a few out a vast body of work on ECC's.

The goal of (always) correctly decoding m in the above scenario is met iff. less than $d(C)/2 < m/2$ bits are modified by the channel.

For many applications, the relaxed notion of error detection suffices, so we can go beyond this bound. That is, the message is either decoded correctly, or an error symbol $\perp$ is returned. In this relaxed setting, up to $d(C) - 1 < m - 1$ errors can be tolerated (there exist codes with large Σ where $d(C)/m$ is arbitrarily close to 1).

[*] The Open University and Ariel University.

[†] Ariel University.

Restating, even for the weaker notion of "usefulness for communication" (of error correction), ECC's only work against the *family* $\mathcal{F}_{THR}$ of adversaries consisting of functions $f : \Sigma^m \rightarrow \Sigma^m$ that flip some $\alpha < 1$ fraction of the bits. It is interesting to ask whether some meaningful notion of "usefulness" can be achieved against families not contained in $\mathcal{F}_{THR}$. The line of work on non malleable codes (NMC), starting with the seminal work of [DPW09] puts forward an interesting notion with this property. Loosely speaking, a code is NMC against a family $\mathcal{F}$, if $f \in \mathcal{F}$ can modify the message to be decoded, but one "unrelated" to the encoded value m. Slightly more precisely, the distribution of the decoded massage when m is encoded and tampered via some $f \in \mathcal{F}$ is $h(m)$, where h is sampled from a distribution D_f over $\{x, 0, \ldots, 2^n - 1, \perp\}$ (that is, the identity function, $\perp$ or the various messages in $\{0,1\}^n$). Furthermore, the code is now specified as a pair of encoding and decoding algorithms $\mathrm{Enc} : \{0,1\}^n \rightarrow \{0,1\}^m, \mathrm{Dec}$ where Enc is randomized (Dec is deterministic), rather then by a set C that uniquely determines the encoding and decoding algorithms (up to the particular embedding of $\{0,1\}^n$ into $\{0,1\}^m$).

NMC's may seem too weak for the 2-party communication problem described above, as a message may be modified into an arbitrary constant message.[1] However, NMC's have found other important applications, such as protecting sensitive data stored in memory against adversarial tampering. For instance, consider data consisting of a secret key of an encryption scheme stored on a tamper-prone hardware (assuming only the memory, rather than logic can be tampered by the adversary).

Later, values encrypted by k are sent in the open over the adversarially corrupted channel. If k could be modified by the adversary into some related k', and it could see values encrypted under k', some information about k could leak. This type of attack is referred as a *related key attack* (see [Bih94], for instance).

Indeed, NMC's can be designed against tampering families not contained in $\mathcal{F}_{THR}$. A simple example is that of bit tampering functions, $\mathcal{F}_{\mathrm{bit}}$, where each $f \in \mathcal{F}_{\mathrm{bit}}$ modifies each bit c_i based only on c_i, rather than the rest of the codeword. It is easy to see that even this simple type of tampering does not allow even error detection (and even if a small error is allowed). For instance, the constant function f_v that modifies each codeword into v, which is some fixed valid codeword is implementable in this family. NMC's against $\mathcal{F}_{bit}$ with constant rate have been devised in [AGM$^+$15]. In fact, already [DPW09] puts forward a (non-constructive) proof of existence of NMC form any $\mathcal{F}$ which is not too large (of size $2^{2^{\alpha n}}$), and several explicit constructions, notably for the family $\mathcal{F}_{2-\mathrm{split}}$, where f modifies each half of the input separately [ADKO15, NMC], achieving constant rate for this challenging setting.

The communication complexity setting Alice and Bob have inputs x, y respectively, and they wish to compute $f(x,y) : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^t$, by sending messages back and forth in rounds over the identity channel (not tampering with the messages). Their goal is to do so while minimizing the communication complexity (CC) between them. Clearly, a solution where one party sends a message and the other party always results in CC of $n + t$ bits, but sometimes one can do a lot better. As a simple example, if we wish to compute the XOR or the parities of x and y, only 2 bits need to be exchanged. As another simple example, if allowing for a small (engligible in n) error, the equality function can be evaluated with $polylog(n)$ communication, where Bob picks a hash function h from a family of quasi-poly size, and Bob sends back $h, h(y)$. Then Alice compares $h(y)$ and $h(x)$, and outputs 1 iff. they are equal. See [KN97] for a great overview of the communication complexity (CC) model.

Generalizing the CC model, it is natural to ask whether protocols with "low" CC exist, when running over a corrupted channel f taken from some family $\mathcal{F}$ (and not apriori known to the protocol's designer).

What do we mean by "low" CC? A good direction is to require that the overhead relatively to π_0 running over the identity channel for the same function is not too large. In other words, we would like to devise a "compiler" transforming protocols π_0 over plain channels into π robust against $\mathcal{F}$.[2] We want to minimize the compiler's *rate* $\sup_{\pi,g} CC(\pi)/CC(\pi')$ as much as possible.

Precisely this question has already been studied in [Sch96], which initiated the field, coining the term CIC (coding for interactive communication) as the name of the task. Schulman considered the family $\mathcal{F}_{1/240}$ of functions modifying up to a 1/240 fraction of the *entire* communication. Schulman

[1] If the parties can afford pre-sharing a random string of length $O(n)$ not known to the adversary, this would reduce the problem to the error detection scenario, while introducing a small error probability.

[2] We stress that f acts on the *entire* protocol transcript. $\mathcal{F}$ is naturally restricted by the setting so that tampering of a particular message sent by the protocol can be based on all previously sent messages in the transcript, but not following ones.

devised a compiler as above with only constant rate (with no increase in the error of computing f).

Note that if rate is not important, the problem is easily solvable by letting Alice send x encoded by an ECC with a good rate, and Bob sending back a reply $f(x, y)$ also encoded and padded, so that both messages are of the same size.[3]. Indeed, as there is an exponential gap even between k-round and $k+1$-round protocols for certain functions [NW93] the above approach would lead to very bad rates.

Schulman's compiler has constant rate. Recently, interest in the setting of CIC (coding of interactive communication) was rekindled in [BR11] which improved Schulman's result to handle $\mathcal{F}_{1/4}$ (with constant rate). Several additional works have followed since (see the survey [Gel15] for details).

However, all work so far on CIC focused on families of channels modifying up to a constant fraction α of the communication. In these works, error correction was the required notion, which is the main source of technical difficulty in these works. In particular, the compiler of a Π_0, attempting to emulate Π_0 with a high round complexity should somehow tackle the issue that entire messages may be arbitrarily modified.

The main question we study in the proposed research is whether we can go beyond threshold families. More precisely:

Q1. Are there useful non-threshold families $\mathcal{F}$ for which there exist compilers from protocols π_0 for evaluating some function $g(x, y)$ in the communication complexity setting, into π' evaluating the same g over $\mathcal{F}$ with a "useful" notion of robustness? The goal is to achieve a meaningful notion of "usefulness", and as good a rate as possible.

As a simple test case, at least the family $\mathcal{F}_{bit}$ should have a compiler according to our definition.

However, it is known that protocols with such a strict correctness requirement can not be correct even against weak non-threshold families. In particular, no protocol π as above (where the party who speaks in round i and the message length are independent of the communication so far) is ϵ-correct against a set of (actually threshold, for $\alpha = 1/4$) functions contained in $\mathcal{F}_{\text{bit}} \cap \mathcal{THR}_{0.25}$ with $\epsilon < 0.5$ (see [Gel15]). Thus, we need to relax the robustness requirement so that $\mathcal{F}_{\text{bit}}$ can be handled, while obtaining a meaningful notion of correctness.

One simple definition that comes to mind is something along the lines of NMC. For simplicity, we only require that Alice gets the output. More specifically, we allow the adversary to modify Alice's output so that $g(x, y)$, $\perp$ is output. The distribution over these two values depends only on the channel f, rather then on the inputs x, y.

A solution to the sub-problem of evaluating the message transmission function ($g(x, y) = y$) can be obtained by NMC. That is, we can view the setting as interactive protocols for $\mathcal{F}$ with error detection, where additionally, the error probability can not be correlated with the inputs.

As an application of the proposed notion, think of a scenario where Bob holds a key k to some encryption algorithm $(\text{Enc}_k, \text{Dec}_k)$ and Alice holds as an input a value x, and they interact to allow Alice to learn $\text{Enc}_k(x)$ or $\text{Dec}_k(x)$ (where x was previously encrypted by Enc_k). The attacker controls the channel in some limited way (can tamper via some $f \in \mathcal{F}$), and can also break into Alice's computer and learn the outputs. To make life easier for the attacker, it is conceivable that the attacker can also influence the choice of the input x (say, the server is an internet store, and the attacker sends emails to clients advertising products to buy).

Assume also that interaction allows to minimize the communication complexity of the encryption (decryption) protocol, which is crucial for the system to work fast enough. As in the original non-interactive scenario, the attacker's goal is to learn something about k, and his strategy is modify k into some related value k'.

If an interactive protocol satisfying our definition is used, the attack does not allow the adversary to learn anything about k, assuming the encryption scheme is secure against standard chosen plaintext and chosen ciphertext attacks.

On the non-triviality of meeting our definition. Standard NMC's do not seem to directly solve the problem for general $g(x, y)$ (against $\mathcal{F}_{\text{bit}}$) since even if we don't care about rate, and have a good NMC for the tampering family, just letting Bob send an encoding of y, allows the

[3] Padding is crucial, since if one of the messages is too short, it may be completely altered, as its length is smaller than the adversary's tampering "budget" 1/240 of the entire communication.

adversary to make Alice output $f(x, y')$, where y' distributed over $\{y\} \cup \{0,1\}^n \cup \{\perp\}$ according to some fixed D_f. This is not exactly what we want (even if we did allow constants as outputs).

At a more high-level look, one reason why standard NMC's for the family $\mathcal{F}$ can not be trivially ported to the interactive setting, is that the encoder is limited to work block-by-block. This is imposed by the nature of the interaction - the message is the entire interaction transcript, and the encoding is done one message at the time, by a "distributed" encoder, implemented by the two parties. The encoding of a block is based only on part of the encoded message so far (the messages the encoding party had sent in the past), and tampered versions of (encoded) blocks the other party had sent. As the parties do not share randomness either, the encoder does not even know the entire randomness string.

Although the adversary is also restricted to tamper each block based on that and previous blocks, this does not restrict its power too dramatically [CGM+15].

On the other hand, interaction allows us to cross-check the sent values, so there is hope to get rid of the constants in D_f's support that are unavoidable without communication. Indeed, this is the definition we pick - we require that for any $f \in \mathcal{F}$, there exists a distribution D_f over $g(x, y) \in \{f(x, y), \perp\}$, so that Alice outputs $g(x, y)$ according chosen from a distribution D'_f which is ϵ-close to D_f (where x, y are the parties original corresponding inputs).

Our results. We devise CIC constructions for several non-threshold families. Observe that the tampering families are always restricted to modify message m_i based on messages m_i or lower, since the "future" part of the communication is not known yet, but the channel needs to send some response replacing m_i without being able to wait for future communication. Implicitly, when we define a family $\mathcal{F}$ in our context, it is assumed to be restricted in this way.

One family which is easy to handle is one modifying each message of Π_0 independently of any other messages, according to some family $\mathcal{F}_{loc}$ for which an NMC code $(\mathrm{Enc}, \mathrm{Dec})$ exists (a code family in fact, for infinitely many message lengths, as we will need to adjust the message length).

As a warm-up, let us consider the simple special case for of $\mathcal{F}_{bit}$. Fix some 2-message Π_0 where Alice sends the first message and outputs $f(x, y)$. Indeed, bitwise tampering of the entire communication can be viewed as applying bitwise tampering to each of the sent messages (independently of the other messages). We compile Π_0 into Π' by encoding each message with a good NMC, and letting Alice add redundancy to her messages, which she can later verify. This allows to catch the "constant" tampering of messages that NMC allows. For simplicity, we describe the construction for a 2-message Π_0, where Alice sends the first message. The general transformation for this case appears in Section 4.2.1.

1. Let (Enc,Dec) denote the NMC that protects individual messages (of length as in Π' below) from being tampered using function from $\mathcal{F}_{bit}$.

2. Alice picks a random value $v_1 \in \{0,1\}^k$ (where k is a security parameter). Let m_1 denote Alice's first message in Π_0 (sampled as in Π_0). She encodes (m_1, v_1) via Enc, obtaining $c_1 = \mathrm{Enc}(m_1, v_1)$. She sends c_1 to Bob.

3. Bob decodes c'_1 it received as (m'_1, v'_1), and produces a response message m_2, as in Π_0 (based on y, r_2, m_1). It computes and sends an encoding $c_2 = Enc(m_2, v_1)$.

4. Alice decodes c'_2, recovering (m'_2, v'_1), and checks whether $v_1 = v'_1$. If so, she outputs what Π_0 would output on m'_2, x, r_1. Otherwise, she outputs $\perp$.

On a high level, error detection is achieved as follows. To modify m_1, the channel needs to tamper with the first message. By robustness of NMC, this leads to modification of v_1 as well. Both into some constant values. Thus, the probability of guessing v_1 is at most $O(2^{-k})$. To modify m_2, as the adversary may not read the first message to tamper with the second, it again needs to guess the right value of v_1. This is regardless of whether it modified v_1 in the first round or not. In any case, the adversary will successfully modify m_1 or m_2 with probability at most 2^{1-k} (of guessing v_1).

A more challenging setting is one where each message can be tampered based on itself, and all previous messages in some restricted way. To handle such families, we define the primitive of multi message NMC (MNMC), and put forward constructions of MNMC for certain non-trivial families. In a nutshell, here we require that the adversary can not change a sequence of some t messages "too much", based on the entire message sequence.

We show a generic construction for CIC against a family $\mathcal{F}$, given MNMC for a related family $\mathcal{G}$. This construction does not work for all $\mathcal{F}$, since the related $\mathcal{G}$ may be too hard even for standard NMC. For instance, for split-state tampering $\mathcal{F}$, $\mathcal{G}$ is the set of all functions, so this approach is not useful for constructing CIC for the interesting case of split-state tampering. However, we are able to devise such CIC given MNMC for an augmented family $\mathcal{G}$ (deviating from our main construction). More concretely, the family $\mathcal{G}$ is simply split-state tampering for messages of suitable size. See 4.1 for a construction.

The MNMC primitive may be interesting of its own right, as a formalization of NMC secure for multiple messages. In particular, additionally to encoding constant messages, it is clear that copying any m_i in place of m_j is not avoidable for multiple messages. We devise MNMC for some non-trivial families where these are the only kinds of tampering allowed – supporting the fact that this is a useful definition.

One interesting family for which we obtain a positive result is that of multi-variate linear functions. That is, the entire communication is split into field elements, and each element m_j is modified into $\sum_i \alpha_{j,i} m_i + b_j$, where $b_j, \alpha_{j,i}$ are constants.[4] It is a major open question to find MNMC for more interesting families, such as split-state as required to obtain CIC for split-state tampering (of the entire communication).

2 Preliminaries

Notation. For a distribution D, we denote by $x \leftarrow D$ the process of sampling a random variable x according to D. For a set S, $x \leftarrow S$ denotes sampling S uniformly at random. For a pair D_1, D_2 of distributions over a domain X, we denote their statistical distance by $SD(D_1, D_2) = \Sigma_{v \in X} |Pr_{x \leftarrow D_1}(x = v) - Pr_{x \leftarrow D_2}(x = v)|$. If $SD(D_1, D_2) \leq \epsilon$, we say that D_1, D_2 are ϵ-close.

Here we spell out standard definitions of coding schemes, and non malleable codes.

Definition 2.1 *(Coding scheme) A coding scheme is a pair of algorithms (Enc, Dec), where Dec is deterministic. $Enc : \{0,1\}^n \to \{0,1\}^m$, and $Dec : \{0,1\}^m \to \{0,1\}^n \cup \{\perp\}$. We have $Pr[Dec(Enc(x)) = x] = 1$, where the probability is taken over the randomness of Enc.*[5]

The following definition of NMC is the standard one, phrased in the flavor of non malleable reductions.

Definition 2.2 *(Non Malleable Code - NMC) Let $\mathcal{F}$ be a family of functions from $\{0,1\}^m \to \{0,1\}^m$. We say that coding scheme (Enc, Dec) with parameters n, m respectively is ϵ-non-malleable for this family, if for every $f \in \mathcal{F}$, there exists a distribution D_f supported on the set of functions $\mathcal{B}_1 = \{g(x) = x, g(x) = \perp\} \cup \{g(x) = v | v \in \{0,1\}^n\}$. such that for every $x \in \{0,1\}^n$, we have $Dec(f(Enc(x))) \approx_\epsilon G(x)$, where $G(x)$ is (freshly!) sampled by picking a function g at random according to D_f, and returning $g(x)$. Here '$\approx_\varepsilon$' means that the statistical distance between two distributions is at most ε.*

We in fact (usually implicitly) consider ensembles of coding schemes $\{(\mathrm{Enc}_n, \mathrm{Dec}_n)\}_{n \in \mathbb{N}}$. The reason to construct an infinite ensemble is that in NMC, encoding a message by breaking it into smaller blocks and encoding each separately does not trivially work. [6].

The coding scheme ensemble has parameters $m(n)$ (length of encoding) and error $\epsilon(n)$ Correspondingly, we consider suitable ensembles of tampering functions $\mathcal{F} = \{\mathcal{F}_{m(n)}\}_n$.[7] We will strive for $\epsilon(n) = 2^{-\theta(n)}$, and rate $m(n)/n$ as small as possible - constant at best, but at most $poly(n)$. As a secondary goal, we strive for Enc, Dec which are polynomial-time in n.[8] We say $\mathcal{G}$ is a sub-ensemble of $\mathcal{F}$ if each $\mathcal{G}_m \in \mathcal{G}$ satisfies $\mathcal{G}_m \subseteq \mathcal{F}_m$. Also, given a pair of ensembles $\mathcal{F}_1, \mathcal{F}_2$, we let $\mathcal{F}^1 \cup \mathcal{F}^2$ denote $\{\mathcal{U}_m | U_m = \mathcal{F}^1_m \cup \mathcal{F}^2_m\}_m$.

[4] As usual, each message can depend only on the preceeding messages, although the MNMC does not rely on this restriction.

[5] Traditionally, the decoding algorithm was deterministic. However, some works, such as [MB16] rely on a more relaxed definition where decoding is allowed to be randomized and a small decoding error is allowed. To keep the notation simple, we stick with perfect correctness here, but it can be readily generalized to capture schemes as in [MB16].

[6] See, e.g. [DPW09] for more discussion.

[7] By sayubg $\mathcal{F}_m$ is empty for some $m \in \mathbb{N}$, this means that m is not a valid length of codewords.

[8] That said, proof-of-concept constructions with super polynomial rate are also interesting as a preliminary result. See Section 4.1, for instance.

3 New notions of NMC

Here we develop several (still non-interactive) extensions of NMC, that will be useful for our purposes, and possibly interesting on their own right.

Definition 3.1 ***(Multi-message Non Malleable Code - MNMC)*** *Consider a coding scheme (ensemble) (Enc, Dec) with rate $m(n)/n$. Let $\mathcal{F}$ denote an ensemble of tampering functions.*[9] *Given $c_1, \ldots, c_t \in \{0,1\}^m$ we denote $\overline{Dec}(c_1, \ldots, c_t) = Dec(c_1), \ldots, Dec(c_t)$.*

We say (Enc, Dec) is t-message ϵ-non-malleable against a tampering family $\mathcal{F}$, if for every $t' \in [1,t], n, f : \{0,1\}^{m(n)t'} \times \{0,1\}^{m(n)t'} \in \mathcal{F}$, there exists a distribution D_f over $(B_{t'})^{t'}$ where $B_{t'} = \{g(x_1, \ldots, x_{t'}) = x_i | i \in [t']\} \cup \{g(x_1, \ldots, x_{t'}) = v | v \in \{\perp\} \cup \{0,1\}^n\}$ such that for every $(x_1, \ldots, x_t) \in (\{0,1\}^n)^t$, we have $\overline{Dec}(f(Enc(x_1)), \ldots, f(Enc(x_t))) \approx_\epsilon G(x)$, where $G(x)$ is (freshly) sampled by picking a function $\overline{g} = (g_1, \ldots, g_{t'})$ according to D_f, and outputting $g_1(x), \ldots, g_{t'}(x)$.

The above generalization of NMC to multiple messages will prove instrumental in our constructions, and may be of independent interest. For the purpose of constructing MNMC, the following definition of closure will be useful.

Definition 3.2 *Let $\mathcal{F}$ denote an ensemble of tampering functions. The t-closure of $\mathcal{F}$, denoted $Cl^t(\mathcal{F})$, is the of ensemble of all functions $\{\mathcal{F}'_m\}_m$ so that each $\mathcal{F}'_m$ is of the form $\{f' : \{0,1\}^m \rightarrow \{0,1\}^m | \exists v \in (\{0,1\}^m)^{t-1}, f \in \mathcal{F}_{mt'}$ such that $f'(y) = f(x_{-i} = v, x_i = y)\}$.* [10]

Definition 3.3 *Let $\mathcal{F}, \mathcal{G}$ denote a pair of function ensembles. Let us denote by $Cl^{\leq t}(\mathcal{F}) = \cup_{t' \leq t} Cl^{t'}(\mathcal{F})$. We say that $\mathcal{F}$ is $(\mathcal{G}, t)$-closed, if $Cl^{\leq t}(\mathcal{F}')$ is a sub-ensemble of $\mathcal{G}$. Furthermore, for each $\mathcal{F}'_m \in Cl^{\leq t}(\mathcal{F})$ then for all $t' \leq t$ $\mathcal{F}_{mt'}$ exists in $\mathcal{F}$, and furthermore, there are no other pairs $(m_1, t_1 \leq t)$ for which $m_1 t_1 = mt'$. If $\mathcal{G}$ is an (infinite) sub-ensemble of $\mathcal{F}$, we say that $\mathcal{F}$ is t-closed. If $\mathcal{F}$ is t-closed for all t, we say that $\mathcal{F}$ is closed.*

Note that all function ensembles $\mathcal{F}$ are trivially $\mathcal{F}_{\text{all}}$-closed (which is not very useful).

A very simple example of a closed function family is the class $\mathcal{F} = \mathcal{F}_{\text{bit}}$ of functions that modify each bit as a function of itself.

Another interesting example is that of the class $\mathcal{F} = \text{Local}_{l_i(m)}^{l_o(m)}$, as considered in [MB16], where $f \in \mathcal{F}_m$ is such that each output bit is influenced by up to $l_o(m)$ input bits, and each input bit influences up to $l_i(m)$ output bits. For a given t, consider the function f' resulting from $f : \{0,1\}^{mt} \rightarrow \{0,1\}^{mt}, v, i$. In f Some of the input (output) bits in i-the block may influence only (be influenced) bits (only by bits) in the same block. Thus, in the worst case f' is only $\text{Local}_{l_i(tm)}^{l_o(tm)}$ local. Thus, $\mathcal{F} = \text{Local}_{l_i(m)}^{l_o(m)}$ is $(\mathcal{G} = \text{Local}_{l_i(mt)}^{l_o(mt)}, t(m))$-local, so it preserves locality with a certain degradation of parameters. For constant t, this degradation is tolerable, for sufficiently large m.

Some (parametrized) function classes do not even tolerate constant t. For instance, for $\mathcal{F}$ which are 4-split-state for instance, choosing $t = 2$ deteriorates $\mathcal{F}$ into $\mathcal{G}$ of 2-split state functions, and $\mathcal{G} = \mathcal{F}_{all}$ for $t \geq 4$. The same holds for the (well studied) threshold families, modifying up to some constant fraction α of the symbols, $\mathcal{THR}_\alpha$. The class $\mathcal{THR}_\alpha$ results in $\mathcal{G} = \mathcal{F}_{\text{all}}$ for $t \geq \alpha^{-1}$.

Clearly, if a coding scheme (ensemble) (Enc, Dec) is t-message ϵ-NMC against a family $\mathcal{F}$ of messages for some $t \geq 1$, then it is ϵ-NMC against the same family. In the other direction, the implication is not clear. We manage to prove the other direction holds for a stronger definition of non-malleability.

Definition 3.4 (Simultaneously non-malleable code - SNMC) *For a coding scheme (Enc, Dec) and function $f \in \mathcal{F}$, for any random input r to Enc, let $g_{f,r} : \{0,1\}^n \rightarrow \{0,1\}^n$ denote the function $Dec(f(Enc(\cdot; r)))$. We say (Enc, Dec) is ϵ simultaneously non-malleable against a tampering family $\mathcal{F}$ (refer as ϵ-SNMC), if for any $f \in \mathcal{F}$, $Pr_r(g_{f,r} \in B) \geq 1 - \epsilon$, where $\mathcal{B} = \{g(x) = x, g(x) = \perp\} \cup \{g(x) = v | v \in \{0,1\}^n\}$.*

It is not hard to see that ϵ-simultaneous non malleability implies ϵ-non-malleability, by picking D_f to be the distribution of $\text{Dec}(f \cdot \text{Enc})(\cdot)$ conditioned on $g_{f,r} \in B$.

However, NMC does not imply SNMC. Not even with a (meaningful) loss in parameters. To see why, consider a simple example with $n = 1$, where the combined effect of $f \cdot \text{Enc}$ is as follows.

[9] Here and in Definition 3.2 the fact that $\mathcal{F}$ is a function ensemble is made explicit, as the definition considers tampering functions for several different code length values.

[10] For our purposes, it suffices to think of t as constant, rather than a function of m.

The output of $f \cdot \mathrm{Enc}$ decodes to $g_0(x) = x$ with probability 0.5, and to $g_1(x) = 1 - x$ with probability 0.5. This requirement can be implemented by the (somewhat contrived) scheme where $\mathrm{Enc}(b) = (b, r)$, where r is a random bit; $\mathrm{Dec}(b, g) = b$, and $\mathcal{F}$ consists, for instance, of a single function $f(b, r) = (g_r(b), r)$.

The scheme clearly does not satisfy the stronger definition for arbitrarily small ϵ, but does satisfy the standard definition with D_f which is uniform over $\{g_0(x) = 0, g_1(x) = 1\}$ with $\epsilon = 0$.

Next, we prove that the stronger notion of simultaneous non-malleability implies the notion of nonmalleablity for multiple messages.

Theorem 3.1 *Let $\mathcal{F}, \mathcal{G}$ denote a pair of function ensembles. Let (Enc, Dec) denote a scheme which is ϵ-SNMC against $\mathcal{G}$, and let $\mathcal{F}$ be a $(\mathcal{G}, t)$-closed family of tampering functions $\mathcal{F}$. Then the same scheme (Enc, Dec) is also t-message $O(t2^{(t-1)n}\epsilon^{1/t})$-MNMC against $\mathcal{F}$ (where $\{0,1\}^n$ is the message domain).*

The requirement in Defintion 3.3 that for each $\mathcal{F}_m$, there is a unique decomposition $m = m't'$ into m', t' is crucial for our construction to work. In a nutshell, we show that encoding each of t' messages with an SNMC tailored to handle attacks by a function in $\mathcal{G}_m$, results in an MNMC against $\mathcal{G}_m$. Now, assume the uniqueness requirement does not hold. Then, for some $\mathcal{F}_m$, there exist two different pairs $(m_1, t_1), (m_2, t_2)$ such that $m_1 t_1 = m_2 t_2 = m$. Then, by definition of $(\mathcal{G}, t)$-closure, there may exist some $f_1, f_2 \in \mathcal{F}_m$ that protect against tampering the message as a sequence of t_1 messages according to $\mathcal{G}_{m_1}$ each or t_2 messages according to $\mathcal{G}_{m_2}$. However, $(\mathrm{Enc}, \mathrm{Dec})$ only encodes messages of length n into messages of a specific length $m(n)$. Thus, if it chooses to encode into length m_1, it may not withstand an attack on t_1 instances by f_2, which treats it as a block of t_2 messages of length m_2 each (as the code not tailored to it). Similarly, a choice m_2 of encoding length will not withstand attacks by f_1.

Corollary 3.2 *Let (Enc, Dec) denote a scheme which is $o(1)$-SNMC against $\mathcal{G}$, and let $\mathcal{F}$ be a $(\mathcal{G}, t)$-closed family of tampering functions $\mathcal{F}$. Then there exists a t-message $o(1)$-MNMC against $\mathcal{F}$.*

The scheme as in the corollary can be achieved by embedding $\{0,1\}^n$ into a larger $\{0,1\}^{n'}$, so that $\mathrm{Enc}_{n'}, \mathrm{Dec}_{n'}$ allow for an error of at most $\epsilon(n')$, such that $\epsilon(n') \leq (\epsilon(n)2^{-nt})^t$.

Proof sketch (of Theorem 3.1) We prove the claim for $t = 2$, which includes the main ideas, and is easy to generalize for other values of t. For $t' = 1$, the claim is trivial by the fact that $(\mathrm{Enc}, \mathrm{Dec})$ is an NMC. Now consider $t' = t$.

We look for pairs r_1, r_2, so that $g^1_{r_1,r_2}(x_1, x_2) = \overline{\mathrm{Dec}}(f(\mathrm{Enc}(x_1; r_1), \mathrm{Enc}(x_2; r_2))[1]$ and $g^2_{r_1,r_2}(x_1, x_2) = \overline{\mathrm{Dec}}(f(\mathrm{Enc}(x_1; r_1), \mathrm{Enc}(x_2; r_2))[2]$ each equal some functions $g_1, g_2 \in \mathcal{B}$ respectively.

We will show that the fraction of such pairs, (r_1, r_2) is very high. To see this, construct an undirected bi-partite inconsistency graph $G = (V = (R_1, R_2), E)$, such that $\{r_1, r_2\} \in E$ iff. there either exists $m_1 \in \{0,1\}^n$, so that $g_{r_1,m_1,r_2}(x_2) = g^2_{r_1,r_2}(m_1, x_2)$ is not in $\mathcal{B}_1$, or $m_2 \in \{0,1\}^n$, so that $g_{r_1,m_2,r_2}(x_1) = g^1_{r_1,r_2}(x_1, m_2)$ is not in $\mathcal{B}_1$.

We will then prove that local consistency implies global consistency. Namely:

Claim 3.3 *If $\{r_1, r_2\} \notin E$, then the (global) condition that $g^1_{r_1,r_2}(m_1, m_2), g^2_{r_1,r_2}(m_1, m_2) \in \mathcal{B}_2$ holds.*

The theorem then immediately follows by combining Claim 3.3, with the observation that the graph above is appropriately sparse. Namely

Claim 3.4 *The graph G above has density $\leq 2^{n+2}\epsilon|R|^2$ edges.*

Proof of Claim 3.4 By closure of $t\text{−}\mathcal{F}$, for all m_1, r_1, the tampering function $f(\mathrm{Enc}(m_1, r_1), c_2)[2]$ applied to $\mathrm{Enc}(m_2, r_2)$ is in $\mathcal{G}$. Thus, as $(\mathrm{Enc}, \mathrm{Dec})$ is an ϵ-SNMC, for each (m_1, r_1), for a $\geq 1 - 2\epsilon$ fraction of r_2's $g^2_{r_1,m_1,r_2}(x_2), g^1_{r_1,m_1,r_2}(x_2)$ is in $\mathcal{B}_1$. The case with $f(c_1, \mathrm{Enc}(m_2, r_2))[1]$ is symmetric. Let us restate E as a union of some E_1, E_2, where E_1 consists of all edges (r_1, r_2) where either $g^1_{r_1,m_1,r_2}(x_2)$ or $g^2_{r_1,m_1,r_2}(x_2)$ is not in $\mathcal{B}_1$ for some m_1, and E_2 is defined analogously. Thus, by union bound, we learn that the degree of each $v \in R_1$ in $G_1 = (R_1, E)$ is bounded by $2 \cdot 2^n\epsilon|R|$ (the situation for E_2 is analogous). Thus, $|E| \leq 2^{n+2}\epsilon|R|^2$. □

Proof of Claim 3.3 We prove a cleaner fact, where only a single function g is involved. The claim follows immediately, by applying the claim to each function simultaneously.

Claim 3.5 *Let $g : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n \cup \{\perp\}$ be a function, such that for every $v \in \{0,1\}^*$, $g(v, \cdot), g(\cdot, v)$ is in B (either constant, or equals the identity function). Then, necessarily $g(x_1, x_2)$ is also in B.*

If $n = 1$, the claim is trivial, so from now we assume that $n > 1$. First assume that there exist some $v_1 \neq v_2$ and $m_{1,1} \neq m_{1,2}, m_2$,for which $g_{|x_1=m_{1,1}} \equiv v_1$ and $g(m_{1,2}, m_2) = v_2$. That is, one of the row functions is constant, and some other row's function does not equal that constant. In this case, $g_{|x_2=m_2}$ is non constant. Thus, it must equal x_1 (by the structure of B). In particular, we must have $v_2 = m_{1,2}$. Now, there are two cases. If $g_{|x_1=m_{1,2}}$ is a constant function, it must equal v_2, In that case, $g_{|x_2=m'_2}$ is x_1 for all m'_2. By similar reasoning to the m_2-column, $g(x_1, x_2) = x_1$, which is a function in B, and we are done. Otherwise, $g_{|x_1=m_{1,2}}$ must equal x_2. In this case, all columns $g_{|x_2=m'_2}$ for $m'_2 \neq m_2$ are non-constant, but also are inconsistent with x_1, which is a contradiction.

It remains to consider the compliment of the above case. One possibility is that all rows equal the same constant function v. Clearly, that means that $g(x_1, x_2) \equiv v$. Otherwise, it must be the case that all rows are the function x_2. This means that $g(x_1, x_2) \equiv x_2$. That proves that $g(x_1, x_2)$ is (globally) x_1 or x_2 or some constant (belongs to $\mathcal{B}_2$).□

Fortunately, it turns out that some known NMC schemes in fact satisfy the stronger SNMC definition, possibly with some loss in parameters. An interesting example for an SNMC family is that of Affine functions. More concretely, for each codeword length m we pick a suitable prime p_m (with $\log p \geq m$) and $\mathcal{F}_m$ consists of functions of the form $f(c) = ac + b$ (operations are over the field $\mathbb{F}_{p_m}$).[11]

More precisely, we have.

Lemma 3.6 *Let $\mathcal{F}_{aff}$ be a class of affine functions as defined above. This class has an $2^{-\theta(n)}$-SNMC with rate $O(\log(n))$.*

Proof Sketch. To see why this holds, consider the construction for $\mathcal{F}_{\text{aff}}$ from [NMC]. Let us denote it by Aff13.

It is not hard to see that Aff13 is also an SNMC with slightly degraded parameters.

Theorem 3.7 (Imported, [NMC]) *Aff13 with $\epsilon(n)$ is an $\epsilon(n)$-NMC for $\mathcal{F}_{aff}$ with rate $O(\log 1/\epsilon + \log n)$.*

The construction above has the following property.

Claim 3.8 *For any choice of $\epsilon(n)$ A modified variant of Aff13, Aff13' (only the mapping $m(n)$ is modified) is a $\epsilon(n)$-SNMC against $\mathcal{F}_{aff}$, with rate $O(\log n + \log 1/\epsilon(n))$.*

Proof of claim. We slightly modify Aff13 into Aff13', which embeds $\{0,1\}^n$ into $\{0,1\}^{n'}$ for $n' = n - \log \epsilon$, and applies Aff13 to the result.

To see why we get an SNMC, we observe (by construction) that for any $f \in \mathcal{F}_{\text{aff}}$, Aff13's construction allows for one of the following situations. (1) If $f = \text{const}$, there exists v, and $a \in \{v, \perp\}$ so that $\forall x Pr(\text{Dec}(f(\text{Enc}(x)) = a)) = 1$. (2) If $f(c) = c$, then $\forall x Pr(\text{Dec}(f(\text{Enc}(x))) = x) = 1$. (3) Otherwise, for all x, $Pr(\text{Dec}(f(\text{Enc}(x))) = \perp) \leq 2^{-n'}$.

We conclude that with probability at least $2^{n-n'} = \epsilon(n)$ $g_{r,f}(x)$ is one of the functions in $\mathcal{B}$, as required. This follows by taking union bound on all messages in $\{0,1\}^n$ for the third case (in the other two cases all pairs (r_1, r_2) lead to $g_{r,f}(x)$ in $\mathcal{B}$). Finally, it is easy to see that the rate of the resulting SNMC is (almost) the same as that of the original scheme, up to linear factors.□

For a given t, and $\mathcal{F}_{\text{aff}}$ as above, assume further that for all $t_1, t_2 \leq t$ and all possible codeword lengths for $\mathcal{F}_{\text{aff}}$ m_1, m_2, we have $m_1 t_1 \neq m_2 t_2$.[12]

Then the corresponding family $\mathcal{F}_{\text{multi-aff},t}$ consists of functions of the form $f(c_1, \ldots, c_{t'}) = (c'_1, \ldots, c'_{t'})$ where $t' \leq t$, and each c'_i is of the form $\Sigma_{j \leq t'} \alpha_j m_j + b$ where the computation is done over $p_{|c_1|}$. (2)

By simple arithmetic (over $\mathbb{F}_{p_{|c_1|}}$), we observe that:

[11] There are, of course many such function families $\mathcal{F}$, corresponding to different sequences of primes

[12] This is a technical requirement that is easy to meet without serious loss in parameters. For instance, use prime values $m > t$ in $\mathcal{F}_{aff}$.

Claim 3.9 *For a given $t \geq 1$, and $\mathcal{F}_{multi\text{-}aff_t}, \mathcal{F}_{aff}$ (related) ensembles of tampering functions as above, $\mathcal{F}_{multi\text{-}aff_t}$ above is $(t, \mathcal{F}_{aff})$-closed.*

As an immediate corollary of Lemma 3.6, Claim 3.9, and Theorem 3.1 we conclude that $\mathcal{F}_{\text{multi-aff}_t}$ has an NMNC.

Corollary 3.10 *Let $t \geq 1$, then there exists a coding scheme (ensemble) $(Enc_n, Dec_n)_n$ of rate $O(t^2 \log(n))$ and error 2^{-nr}, which is t-message MNMC against a family $\mathcal{F}_{multi\text{-}aff,t}$ as above.*

The corollary follows by using the scheme from 3.8, further embedding the message space $\{0,1\}^n$ into $\{0,1\}^{n'}$ where $n' = \Omega(nt^2)$, to compensate for the loss in parameters in the transformation from SNMC into t-message MNMC.

4 Coding for interactive communication over general channel families

Protocol. We consider protocols Π_0 between a pair of parties, Alice and Bob for evaluating functions of the form $f : X \times Y \to Z$, where X, Y, Z are finite domain. Alice holds an input $x \in X$, and Bob holds $y \in Y$, and the goal of the protocol is to interactively compute $f(x, y)$, which should be output at the end of the computation by Alice.[13] The interactive protocol consists of r rounds, where the parties take turns sending a message to the other party (the first party to speak is also fixed by Π). The protocol runs over a channel corrupted according to a function $f \in \mathcal{F}$, where $\mathcal{F}$ is a family of tampering functions. Formally, an interactive protocol Π between two parties consists of a pair of "next message" functions π_A, π_B. The next message function π_A (π_B) takes the input x, round number i and message seqeunce received by Alice (Bob) so far, and outputs the next message to be send by Alice (Bob). For simplicity of notation, we assume π_A, π_B always output binary strings. Furthermore, we assume that each message output by π_A is always of the same length ℓ.[14] Note that the protocol statement does not rely on the channel over which it is executed. We denote the sequence of messages sent or received by Alice throughout the protocol by $\text{trans}_A = m_1^A, \ldots, m_r^A$ and the same sequence from Bob's perspective by $\text{trans}_B = m_1^B, \ldots, m_r^B$ (m_i^A does not necessarily equal m_i^B due to tampering by the channel). We denote the output functions of Alice and Bob respectively by $out_A(x, \text{trans}_A, r_A), out_B(y, \text{trans}_B, r_B)$.

Channels. A channel is a (deterministic) mapping taking the round number i, messages *sent* so far by both parties $(\text{trans}_B, \text{trans}_A)$, and message m sent in the current round, and modifies it into a message m' of the same length (that will be the one received at the other end). As mentioned above, our channels will always be deterministic functions f of the transcript, picked from some family $\mathcal{F}$. In particular, all functions in such a family are limited so that the tampering of a message does not depend on following messages in the transcript.

Protocols running over adversarial channels

Definition 4.1 *For a given protocol π (with parameters r_0, n), we say that it is ϵ-robust protocol for evaluating a function $g(x, y)$ over a family $\mathcal{F}$ of channels if it is:*

1. *$\epsilon(n)$-correct: Over the identity channel, for all x, y Alice outputs $f(x, y)$ with probability $\geq 1 - \epsilon$.*

2. *$\epsilon(n)$-nonmalleably-detectable (NMCD): For each $c \in \mathcal{C}$, there exists a distribution D_c over $\{\perp, f(x, y)\}$ such that for and all x, y, Alice's output distribution D'_c is ϵ-close to a distribution D_c.*

Our main goal in this work, is to study the feasibility and, if possible, the rate of compiling general protocols Π_0 which are correct over the identity channel, into protocols Π for a family $\mathcal{F}$ of channels (evaluating the same function as Π_0) according to Definition 4.1. We refer to such compilers as interactive *coding schemes*.[15]

[13] For simplicity, we require that only one party learns the output.

[14] This setting generalizes the common practice in the CIC literature, where each message is of length exactly 1.

[15] Π_0 is specified in some convenient model. In particular, π has access to π_0's code if needed, rather than just oracle access to its next-message functions.

The main parameter of interest of such a scheme is its rate,
$\text{rate}(m) = \sup_{\pi_0, g:\{0,1\}^n \times \{0,1\}^n \to \{0,1\}} CC(\pi)/m$, where π_0 is a protocol for evaluating the function g in the communication complexity model (over plain channels), m is its communication complexity, and π is the resulting protocol.

4.1 Protocols against 2-split state

In this section, we devise a coding scheme for the family $\mathcal{F}_{\text{2-split}}$ of split state tampering functions, assuming an MNMC for a related family. This is merely a reduction, as it remains open whether an MNMC as required exists. Consider a t-round protocol Π_0 for a function $g(x, y)$. Our compiler assumes the existence of t-MNMC for a family related to $\mathcal{F}_{\text{2-split}}$, we denote by $\mathcal{F}^t_{2-\text{alt-split}}$. In this family, each $f \in \mathcal{F}^t_{mt}$ splits c into t length-m blocks, and each block c_i is divided into two halves (c^L_i, c^R_i). It tampers c by replacing the left-half vector via $f_L(c^L_1, \ldots, c^L_t)$ and the right-half message by $f_R(c^R_1, \ldots, c^R_t)$.[16]

Formally, the reduction is summarized in the following theorem.

Theorem 4.1 *Let Π_0 denote an $\epsilon(n)$-correct, r-round protocol, with length-n messages for evaluating $g(x, y)$ over the identity channel. Assume there exists an r-round ϵ-MNMC (Enc, Dec) against $\mathcal{F}_{2-alt\text{-}split}$. Then there exists a protocol Π for g with error $\epsilon + 2^{-nr}$ against $\mathcal{F}_{2-split}$. For the typical parameter choice of $\epsilon(n) = 2^{-n}$, the rate of the resulting Π is $m(2nr + \log r)/n$.*

Proof sketch. We will need the following simple lemma saying that any NMC against split-state tampering is a (statistical) secret sharing of the encoded message. More precisely

Lemma 4.2 *Let (Enc, Dec) denote an ϵ-NMC against $\mathcal{F}_{m(n)}$. For a given $x \in \{0,1\}^n$, consider the random variable $(L_x, R_x) \leftarrow Enc(x)$. Then for any $x, x' \in \{0,1\}^n$ $L_x, L_{x'}$ are $O(\epsilon)$-close and $R_x, R_{x'}$ are $O(\epsilon)$-close.*

In a nutshell, the lemma holds since otherwise assume $SD(L_x, L_y) > 10\epsilon$ (same holds for R). Then the adversary could tamper with L (R), leading to a decoding of $\perp$ if $m = x'$ and leave it as is for x (R is not tampered in any case). This will result in decoded message distributions with SD higher then ϵ between the output distributions for x and x' – a contradiction to the non-malleability property.

We are now ready to present our coding scheme.

We prove that Π above is an $\epsilon + 2^{-n}$-robust protocol for evaluating $g(x, y)$. We first argue correctness. The key observation here is that the probability that a party can not complement R^2_i into a value that it intends to for a given round i is $O(\epsilon(n'))$, when the protocol runs over the identity channel (for R^1_i, this probability is 0). The reason is that $L_{0^{n'}}$ is $\epsilon(n')$-close to the distribution of L_v for any fixed $v \in \{0,1\}^{n'}$. In particular, when sampling $L_{0^{n'}}$ (in the first half), the probability of getting a value that can not in the support of L_v is $O(\epsilon)$ (bounded by half the statistical distance between L_v and $L_{0^{n'}}$). Taking union bound, we get a bound of $O(r\epsilon(n'))$ on the added bound on the correctness error (if this type of error does not occur for any of the messages, we emulate Π_0 exactly).

We conclude, by construction, that the protocol is $\epsilon(n) + 2^{-nr}$ correct for a suitable choice of $n'(n)$.

The NMCD property follows from the fact that tampering by the channel is according to the $\mathcal{F}_{2-\text{alt-split}}$ family. This is the case, because in the first half of the communication, only L-parts are present, and in the second only R-parts are present. Also, the channel can modify each block based on the other blocks in that half (so we indeed need an MNMC). By properties of the MNMC, the j'th (out of $2r$) message m_j (of the form r_i or (c_i, r'_i)) sent by Alice in the protocol emulation, is replaced by some distribution over $\{m_1, \ldots, m_j\} \cup \{\perp\} \cup \{0,1\}^{n'}$. By properties of MNMC, this distribution is $\epsilon(n')$-close to D_f, which depends only on the channel f (rather then the passing messages). We analyze Alice's output distribution when the tampering of the $2r$-tuples is according to D_f (which is $O(\epsilon(n'))$-close to the protocol's output distribution). Similarly to the correctness argument, sampling a suitable R^2_i or L^2_i succeeds with probability $\geq 1 - O(r\epsilon(n'))$ for all messages, in case each message is mapped to itself by the tampering function. In this case, $f(x, y)$ is output with probability $1 - \epsilon(n) - O(r\epsilon(n'))$.

[16] this is as opposed to modifying the first $t/2$ blocks and last $t/2$ blocks independently.

Algorithm 1: Resulting Π

We compile Π_0 into Π below.

- Let $(\mathrm{Enc}, \mathrm{Dec})$, and Π_0 be as in the theorem.
  We embed $\{0,1\}^n$ into $\{0,1\}^{n'}$ for n' to be chosen later. Denote $m = m(n')$.
- The communication proceeds in two halfs. In the first half, the L-part of the encodings (via Enc_m)of $2r$ messages are sent by the parties. In the second half, the $2r$ corresponding right halves are sent. If Enc_m is such that $|L|, |R|$ differ, then the next party to speak adds an $2r+1$'th message of (say) all-zero's for padding.
- (first half - Left side of the transcript):
  1. For every message c_i in Π_0 to be sent by Bob
     (a) Alice picks a random $r_i \in \{0,1\}^{n'-n}$.
     (b) Alice samples $(L_{r_i}, R_{r_i}) \leftarrow \mathrm{Enc}_{n'}(r_i)$, and sends $L_i^1 = L_{r_i}$ to Bob, and saves $R_i^1 = R_{r_i}$ for future use.
     (c) Bob samples $(L_0, R_0) \leftarrow \mathrm{Enc}_{n'}(0^{n'})$, and sends $L_i^2 = L_0$ to Alice (R_0 is discarded).
  2. For every message c_i in Π_0 to be sent by Alice an analogous procedure is applied. Namely:
     (a) Bob picks a random $r_i \in \{0,1\}^n$.
     (b) Bob samples $(L_i^1, R_i^1) \leftarrow \mathrm{Enc}_{n'}(r_i)$, and sends $L_i^1 = L_{r_i}$ to Alice.
     (c) Alice samples $(L_0, R_0) \leftarrow \mathrm{Enc}_{n'}(0^{n'})$, and sends $L_i^2 = L_0$ to Bob (R_0 is discarded).
- (second half - Right side generation):
  1. Bob (Alice) initializes his emulated Π_0-transcript $\mathrm{trans}_B \leftarrow \phi$ ($\mathrm{trans}_A \leftarrow \phi$) and samples r_B
     (r_A).
  2. For every message c_i in Π_0 to be sent by Bob
     (the protocol for Alice is analogous):
     (a) Alice sends R_i^1 to Bob.
     (b) Bob applies $\mathrm{Dec}_{n'}(L_i^1, R_i^1)$ to decode the received value r_i'.
     (c) Bob checks whether he had decided to abort during the emulation of a previous message.
     If not, he computes the next message $c_i \leftarrow \pi_{0,B}(i, y, \mathrm{trans}_B, r_B)$ to send, and sets $(v_1, v_2) = (c_i, r_i')$. Otherwise, he sets (v_1, v_2) to be a random pair. He complements L_i^2 by sampling R_i^2 at random conditioned on (L_i^2, R_i^2) being in the support of $\mathrm{Enc}_{n'}(v_1, v_2)$.
     Here and elsewhere, if there is no such R_i^2, he samples a value R_i^2 at random as the right part of $\mathrm{Enc}_{n'}(0^{n'})$. He sends Alice R_i^2.
     (d) Alice receives R_i^2 and decodes $(c_i', r_i'') = \mathrm{Dec}(L_i^2, R_i^2)$.
     If $r_i'' = r_i$ which she sent in the first half, she updates the transcript trans_A by appending c_i. Otherwise, she decides to abort.
- Output: If Alice had decided to abort, she outputs $\perp$. Otherwise, she outputs $out_A^{\pi_0}(x, \mathrm{trans}_A, r_A)$
  (in fact, the (emulated) messages sent by her are not required, as they are determined by x, r_A).

Now, consider an event where at least one of the messages is not mapped to itself, and let m_j denote the first such message. m_j is either of the form r_i or (r_i, c_i) for some i. We show that the Alice's output is then $\perp$ with probability at least $1 - O(\epsilon(n'))$. Consider, for instance, the case when $m_j = r_i$ sent to Bob. Then, tampering by any of the options in $\mathcal{F}$ other then the identity, leads to a change in the value of r_i with probability at most $2^{-(n'-n)}$. This is the case since if r_i is replaced by some $r_{i'}$ for $i' < i$ this is the probability of a collision, because each of the $r_{i'}$'s is selected at random. Similarly, for a constant v, this is also the probability that v happens to equal r_i (for messages of the form $(r_{i'}, c_{i'})$, the tampering is detected with probability 1). Thus, Bob will recieve $r'_i \neq r_i$ with overwhelming probability, and send (c_i, r'_i) to Alice. For any kind of tampering of (c_i, r'_i), Alice will decide to abort. The reason is that either the message has the wrong format (when copying a previous message by Alice containing r_i), or will not contain the right r_i value with overwhelming probability $1 - 2^{-(n'-n)}$. As she had set the abort flag, she will output $\perp$ at the end of the protocol.

If Alice was the receiver of the tampered message r_i, then Bob will detect the tampering for the following message sent with overwhelming probability. He will subsequently replace r'_i in the next message (c_i, r'_i) he sends with a random value, and Alice will detect an inequality of r''_i and r_i or wrong syntax with high probability. Namely, the former occurs with probability $1 - 2^{-(n'-n)}$. in case (c_i, r_i) is mapped to itself or some other (previous) message or a constant with this syntax. Otherwise (incorrect syntax of a constant or a previous message), Alice aborts with probability 1.

The case where some $m_j = (c_i, r_i)$ is tampered is similar. [17] By the analysis above, it suffices to pick n' so that $n' - n \geq n$, and $\epsilon(n') \leq 2^{-nr}/r$. For $\epsilon(n) = 2^{-n}$, we can set $n' = 2nr + \log r$. This results in rate $m(2nr + \log r)/n$.

4.2 Families tampering each message "easily"

4.2.1 Each message is tampered independently of others.

Here we show that for any family of tampering functions where each message is tampered separately by a function for which a good NMC exists, a CIC compiling protocols Π_0 over the honest channel exists. As opposed to the previous section, this is an unconditional construction.

Theorem 4.3 *Let Π_0 be a protocol with message length n and r rounds evaluating $g(x, y)$ over the identity channel with ϵ'-robustness.*

Let $\mathcal{F}$ denote a family of channel functions, which splits its message into blocks of size $m(n)$, and tampers each according to $\mathcal{G}$ (independently of other blocks). Let (Enc, Dec) be an $(\epsilon(n))$-NMC (ensemble) against $\mathcal{G}$ where $\epsilon = o(1)$. Then there exists a protocol Π for computing g over $\mathcal{F}$ with robustness $\epsilon' + 2^{-nr}$. For the typical parameter choice of $\epsilon(n) \leq 2^{-n}$, the protocol's rate is $m(2nr + \log r)/n$.

Proof sketch. The high level idea is to encode each message separately via an NMC, authenticated with a random string sent by the party one round earlier. The authentication procedure is similar to the one used in the reduction of CIC for $\mathcal{F}_{2-\text{split}}$ to MNMC. The key difference here is that because each message is tampered separately, there is no need for an MNMC.

Correctness Sketch. As mentioned above, the analysis here is similar to that of Section 4.1's constrction. By the guarantee of ϵ-NMC, we have that the $2r$ messages are tampered according to a distribution D_f over $(f_1, \ldots, f_{2r})$, where the f_j's are independent and each belongs to a set $\{m_j\} \cup \{0,1\}^{n'} \cup \{\perp\}$.

Thus, in case all functions are identity functions, $f(x, y)$ is output with probability $\geq 1 - O(r\epsilon(n'))$. Otherwise, the parties output $\perp$ with probability $O(2^{n-n'})$ by arguments similar to those in Section 4.1. Correctness follows trivially from the fact that an NMC is a coding scheme (with 0-error).

Corollaries of Theorem 4.3. A nice and simple corollary of Theorem 4.3, is the family $\mathcal{F}_{bit}$ of bitwise tampering, serving as a good proof of concept to the feasibility of out CIC definition.

[17] Note that we do not even use Lemma 4.2 to prove NMCD.

Algorithm 2: Resulting Π

- Let (Enc,Dec) denote the NMC that protects individual message from being tampered using function from $\mathcal{G}$ as in Theorem 4.3.
- The parties emulate the protocol, by embedding the messages into $\{0,1\}^{n'}$ for $n'(n)$ to be picked later. To emulate the i'th message of Π_0, if Bob sends it:
  1. Alice picks a random value $r_i \in \{0,1\}^{n'-n}$. She computes and sends Bob $c_0 = (\perp, v_0)$.
  2. Let r'_i denote the value received by Bob (after decoding). If he hadn't decided to abort, he sets c_i to be the next message $\pi^0_B(i, y, r_B, \mathrm{trans}_B)$ according to π_0, where trans_B is the transcript recorded so far, and sets $(v_1, v_2) = (c_i, r'_i)$. Otherwise, if Bob had decided to abort, he picks a random pair as (v_1, v_2) and sends $\mathrm{Enc}_{n'}(v_1, v_2)$.
  3. Alice decodes the message, and gets a pair (c_i, r''_i). If $r''_i = r_i$ which she sent, she updates the transcript $trans_A$ by appending c_i. Otherwise, she decides to abort (sets a flag).

  If Alice sends the message, they perform the symmetric protocol.
- Output: If Alice had set the abort flag, she outputs $\perp$. Otehrwise, she computes and outputs the value $out^0_A(x, \mathrm{trans}_A, r_A)$.

4.2.2 Each message is tampered depending on previous messages.

Here we consider tampering by a $\mathcal{G}$-closed family $\mathcal{F}$. We show that if $\mathcal{G}$ is r-message non-malleable, then there exists a coding scheme for compiling any protocol Π_0 over the identity channel into a protocol Π against $\mathcal{F}$.

More precisely, we have the following theorem.

Theorem 4.4 *Let Π_0 denote an r_0-message protocol for evaluating a function $g(x,y)$ with message length n with ϵ'-correctness.*

Let $\mathcal{F}$ be $(\mathcal{G}, r)$-closed, and assume there exists an r-message $\epsilon(n)$-MNMC for $\mathcal{G}$, where $\epsilon = o(1)$. Then there exists a protocol Π for computing g over $\mathcal{F}$ with a robustness parameter $\epsilon' + 2^{-nr}$.[18]

Proof sketch. The same construction as in the previous section works, with the sole difference that an MNMC instead of an NMC is used.

Concrete corollaries of Theorem 4.4. One interesting example of an MNMC as we need follows from Corollary 3.10, resulting in CIC for the family $\mathcal{F}_{\text{multi-affine}}$ of tampering by affine functions. When working over $\mathbb{F}_2$, $\mathcal{F}_{\text{bit}}$ is a special case of $\mathcal{F}_{\text{multi-affine}}$.

5 Future work

In this work, we have initiated a study of CIC for non-threshold functions. We have obtained several preliminary results, but many questions remain open.

MNMC and SNMC. As demonstrated by our applications to CIC, MNMC is arguably a useful generalization of NMC.

One interesting open question regarding MNMCs is whether SNMC is necessary for MNMC. That is, let $\mathcal{F}, \mathcal{G}$ be a pair of function ensembles, such that $\mathcal{F}$ is $(\mathcal{G}, t)$-closed.

Q1 Does the existence of a (t-message) MNMC for a $(t, \mathcal{G})$-closed family $\mathcal{F}$, imply the existence of a SNMC for $\mathcal{G}$ (or for some sub-ensemble thereof).[19]

[18] The reason we can not apply this theorem to $\mathcal{F} = \mathcal{F}_{2-\text{split}}$, since even $Cl^{\leq 2}(\mathcal{F}_{2-\text{split}})$ is already $\mathcal{F}_{\text{all}}$.

[19] There may also be some loss in ϵ.

As an interesting special case, consider the families $\mathcal{F}_{2-\text{alt-split}}$ as in Section 4.1, which by Theorem 4.1 would yield a CIC against split state tampering.[20]

We were not yet been able to prove any of the known NMC constructions of NMC for $\mathcal{F}_{2-\text{split}}$ to be MNCM. It appears non trivial, and we have not managed construct SNMC (anad thus MNMC) from existing $\mathcal{F}_{2-\text{split}}$ constructions. Perhaps a different technique rather than going through SNMC may be required.

For starters, we ask:

Q1.1 Does there exist a 2-message MNMC against $\mathcal{F}_{2-\text{split}}$?

In turn, SNMC gives rise to MNMC for related function families, and thus appears as a potentially useful notion on its own right.

Q2 Characterize the set of tampering families $\mathcal{F}$ for which $o(1)$-SNMC exists.

Coding for interactive communication There are several open questions here. The main question is understanding the set of channels over which all functions can be evaluated (regardless of rate).

Q3 Characterize the channel families $\mathcal{F}$ for which CIC with error $o(1)$ exists. In particular, does it equal the set of channels for which standard NMC exists?

Q4 For which Channel families compilers with a good rate can be devised?

In particular, our construction for $\mathcal{F}_{\text{multi-affine}}$ has a good rate. More precisely, by Theorem 4.3, and Corollary 3.10, we obtain a rate of $O(nr^3)/nr = O(r^2) = O(m^2)$ (where r is the round complexity of the original protocol Π_0, and n is its message size, and $m = nr$ is its communication complexity).

Q5 Explore other notions of robustness for CIC. In particular, if we relax the NMCD requirement so that D_f's support also includes the set of constant functions. This is arguably a more "natural" extension of NMC to the interactive setting. Can we achieve this notion for families for which our original notion is not achievable?

References

[ADKO15] Divesh Aggarwal, Yevgeniy Dodis, Tomasz Kazana, and Maciej Obremski. Non-malleable reductions and applications. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 459–468. ACM, 2015.

[AGM92] Nguyen Q. A, László Györfi, and James L. Massey. Constructions of binary constant-weight cyclic codes and cyclically permutable codes. *IEEE Trans. Information Theory*, 38(3):940–949, 1992.

[AGM+15] 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 Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 538–557. Springer, 2015.

[Bih94] Eli Biham. New types of cryptanalytic attacks using related keys. *J. Cryptology*, 7(4):229–246, 1994.

[BR11] Mark Braverman and Anup Rao. Towards coding for maximum errors in interactive communication. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 159–166. ACM, 2011.

[CGM+15] Nishanth Chandran, Vipul Goyal, Pratyay Mukherjee, Omkant Pandey, and Jalaj Upadhyay. Block-wise non-malleable codes. *IACR Cryptology ePrint Archive*, 2015:129, 2015.

[20] Clearly, this is equivalent to the same problem for $\mathcal{F}_{2-\text{split}}$.

[DPW09] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. Cryptology ePrint Archive, Report 2009/608, 2009. http://eprint.iacr.org/2009/608.

[Gel15] Ran Gelles. Coding for interactive communication: A survey. 2015.

[Ham50] R. W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160, 1950.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

[MB16] Mukul Kulkarni Tal Malkin Marshall Ball, Dana Dachman-Soled. Non-malleable codes for bounded depth, bounded fan-in circuits. 2016.

[NMC]

[NW93] Noam Nisan and Avi Wigderson. Rounds in communication complexity revisited. *SIAM J. Comput.*, 22(1):211–219, 1993.

[Ree60] Gustave Reed, Irving S.; Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics (SIAM)*, 8(2):300–304, 1960.

[Sch96] Leonard J. Schulman. Coding for interactive communication. *IEEE Trans. Information Theory*, 42(6):1745–1756, 1996.

[Sha48] C. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.