A New Approach to Multi-Party Peer-to-Peer Communication Complexity

Adi Rosén*   Florent Urrutia†

Abstract

We introduce new models and new information theoretic measures for the study of communication complexity in the natural peer-to-peer, multi-party, number-in-hand setting. We prove a number of properties of our new models and measures, and then, in order to exemplify their effectiveness, we use them to prove two lower bounds. The more elaborate one is a tight lower bound of $\Omega(kn)$ on the multi-party peer-to-peer randomized communication complexity of the k-player, n-bit function Disjointness, Disj_k^n. The other one is a tight lower bound of $\Omega(kn)$ on the multi-party peer-to-peer randomized communication complexity of the k-player, n-bit bitwise parity function, Par_k^n. Both lower bounds hold when $n = \Omega(k)$. The lower bound for Disj_k^n improves over the lower bound that can be inferred from the result of Braverman et al. (FOCS 2013), which was proved in the coordinator model and can yield a lower bound of $\Omega(kn/\log k)$ in the peer-to-peer model.

To the best of our knowledge, our lower bounds are the first tight (non-trivial) lower bounds on communication complexity in the natural *peer-to-peer* multi-party setting.

In addition to the above results for communication complexity, we also prove, using the same tools, an $\Omega(n)$ lower bound on the number of random bits necessary for the (information theoretic) private computation of the function Disj_k^n.

*CNRS and Université Paris Diderot, 75205 Paris, France, email: `adiro@irif.fr`. Research supported in part by ANR project RDAM.

†Université Paris Diderot, 75205 Paris, France, email: `urrutia@irif.fr`. Research supported in part by ERC QCC and by ANR project RDAM.

1 Introduction

Communication complexity, first introduced by Yao [48], has become a major topic of research in Theoretical Computer Science, both for its own sake, and as a tool which has yielded important results (mostly lower bounds) in various theoretical computer science fields such as circuit complexity, streaming algorithms, or data structures (e.g., [37, 40, 25, 43, 24]). Communication complexity is a measure for the amount of communication needed in order to solve a problem whose input is distributed among several players. The two-party case, where two players, usually called Alice and Bob, cooperate in order to compute a function of their respective inputs, has been widely studied with many important results; yet major questions in this area are still open today (e.g., the log-rank conjecture, see [37]). The multi-party case, where $k \geq 3$ players cooperate in order to compute a function of their inputs, is much less understood.

A number of variants have been proposed in the literature to extend the two-party setting into the multi-party one. In this paper we consider the more natural *number-in-hand* (NIH) setting, where each player has its own input, as opposed to the so-called *number-on-forehead* (NOF) setting, where each player knows all pieces of the input except one, its own. Moreover, also the communication structure between the players in the multi-party setting was considered in the literature under a number of variants. For example, in the *blackboard* (or *broadcast*) model the communication between the players is achieved by each player writing, in turn, a message on the board, to be read by all other players. In the *coordinator* model, introduced in [21], there is an additional entity, the coordinator, and all players communicate back and forth only with the coordinator. The most natural setting is, however, the *peer-to-peer message-passing* model, where each pair of players is connected by a communication link, and each player can send a separate message to any other player. This latter setting has been studied, in the context of communication complexity, even less than the other multi-party settings, probably due to the difficulty in tracking the distributed communication patterns that occur during a run of a protocol in that setting. This setting is, however, not only the most natural one, and the one that occurs the most in real systems, but is the setting studied widely in the distributed algorithms and distributed computation communities, for complexity measures which are usually other than communication complexity.

In the present paper we attempt to fill this gap in the study of peer-to-peer communication complexity, and, further, to create a more solid bridge between the research field of communication complexity and the research field of distributed computation. We propose a computation model, together with an information theoretic complexity measure, for the analysis of the communication complexity of protocols in the asynchronous multi-party peer-to-peer (number-in-hand) setting. We argue that our model is, on the one hand, only a slight restriction over the asynchronous model usually used in the distributed computation literature, and, on the other hand, stronger than the models that have been previously suggested in order to study communication complexity in the peer-to-peer setting common in the distributed computation literature (e.g., [21, 46]). Furthermore, our model lends itself to the analysis of communication complexity, most notably using information theoretic tools.

Indeed, after defining our model and our information theoretic measure, that we call *Multi-party Information Cost* (MIC), we prove a number of properties of that measure, and then prove a number of fundamental properties of protocols in our model. We then exemplify the effectiveness of our model and information theoretic measure by proving two tight lower bounds. The more elaborate one is a tight lower bound of $\Omega(kn)$, when $n = \Omega(k)$, on the peer-to-peer randomized communication complexity of the function `set-disjointness` (Disj_k^n). This function is a basic, important function, which has been the subject of a large number of studies in communication complexity, and is often seen as a test for our ability to give lower bounds in a given model (cf. [16]). We note that the communication complexity of Disjointness in the two-party case is well understood [30, 42, 3, 7, 9]. From a quantitative point of view, our result for peer-to-peer multi-party Disjointness improves by a $\log k$ factor the lower bound that could be deduced for the peer-to-peer model from the lower bound on the communication complexity of Disjointness in the coordinator model [8]. The second lower bound that we prove is a tight lower bound of $\Omega(kn)$, when $n = \Omega(k)$, on the peer-to-peer randomized communication complexity of the bitwise parity function Par_k^n. Both our lower bounds are obtained by giving a lower bound on the MIC of the function at hand, which yields the lower bound on the communication complexity of that function. We believe that our lower bounds are the first tight (non-trivial) lower bound on communication complexity in a peer-to-peer multi-party setting.[1]

It is important to note that, to the best of our knowledge, there is no known method to obtain tight lower bounds on multi-party communication complexity in a peer-to-peer setting via lower bounds in other known multi-party settings. Lower bounds obtained in the coordinator model can be transferred to the peer-to-peer model at the cost of a $\log k$ factor, where k is the number of players, because any peer-to-peer protocol can be simulated in the coordinator model by having the players attach to every message the identity of the destination of that message [41, 22]. The loss of this factor in the lower bounds is unavoidable when the communication protocols can exploit a flexible communication pattern, since there are examples of functions where this factor in the communication complexity is necessary, while others, e.g., the parity function of single-bit inputs, have the same communication complexity in the coordinator and peer-to-peer settings (see a more detailed discussion on this point in Section 2.2). Therefore, one cannot prove tight lower bounds in the peer-to-peer setting by proving corresponding results in the coordinator model. Note that flexible communication configurations arise naturally for mobile communicating devices, for example, when these devices exchange information with the nearby devices. Constructions based on the pointer jumping problem also seem to be harder in the coordinator model, as solving the problem usually requires exchanging information in a specific order determined by the inputs of the players. It is thus important to develop lower bound techniques which apply directly in the peer-to-peer model, as we do in the present paper. Information theoretic tools seem, as we show, most suitable for this task.

Information theoretic complexity measures. As indicated above, our work makes use of information theoretic tools. Based on information theory, developed by Shannon [44], *Information Complexity* (IC), originally defined in [2, 14], is a powerful tool for the study of two-party communication protocols. Information complexity is a measure of how much *information*, about each other's input, the players must learn during the course of the protocol, if that protocol must compute the function correctly. Since IC can be shown to provide a lower bound on the communication complexity, this measure has proven to be a strong and useful tool for obtaining lower bounds on two-party communication complexity in a sequence of papers (e.g., [3, 4, 11, 7]). However, information complexity cannot be extended in a straightforward manner to the multi-party setting. This is because with three players or more, any function can be computed privately (cf. [5, 19]), i.e., in a way such that the players learn nothing but the value of the function to compute. This implies that the information complexity of any function is too low to provide a meaningful lower bound on the communication complexity in the natural peer-to-peer multi-party setting. Therefore, before the present paper, information complexity and its variants have been used to obtain lower bounds on multi-party communication complexity only in settings which do not allow for private protocols (and most notably not in the natural peer-to-peer setting), with the single exception of [31]. For example, a number of lower bounds have been obtained via information complexity for a promise version of set-disjointness in the broadcast model [3, 13, 27] (also cf. [29]), and external information complexity was used in [10] for a lower bound on the general disjointness function, also in the broadcast model. In the coordinator model, lower bounds

[1]Lower bounds in a seemingly peer-to-peer setting were given in [46]. However, in the model of that paper, the communication pattern is determined by an external view of the transcript, which makes the model equivalent to the coordinator model.

on the communication complexity of set-disjointness were given via variants of information complexity [8]. The latter result was extended in [15] to the function *Tribes*. A notion of external information cost in the coordinator model was introduced in [28] to study maximum matching in a distributed setting. We note that the study of communication complexity in number-in-hand multi-party settings via techniques other than those based on information theory is limited to very few papers. One such example is the technique of *symmetrization* that was introduced for the coordinator model in [41], and was shown to be useful to study functions such as the bitwise AND. That technique was further developed along with other reduction techniques in [45, 46, 47]. Another example is the notion of strong fooling sets, introduced in [12] to study deterministic communication complexity of discreet protocols, also defined in [12].

Private computation. It is well known that in the multi-party number-in-hand peer-to-peer setting, unlike in the two-party case, *any* function can be privately computed [5, 19]. The model that we define in the present paper does allow for (information theoretic) private computation of any function [5, 19, 1]. The minimum amount of private randomness needed in order to compute privately a given function is often referred to in this context as the *randomness complexity* of that function. Randomness complexity (in private computation) is of interest because true randomness is considered a costly resource, and since randomness complexity in private computation has been shown to be related to other complexity measures, such as the circuit size of the function or its sensitivity. For example, it has been shown [38] that a boolean function f has a linear size circuit if and only if f has constant randomness complexity. A small number of works [6, 36, 26, 31] prove lower bounds on the randomness complexity of the parity function. The parity and other modulo-sum functions are, to the best of our knowledge, the only functions for which randomness complexity lower bounds are known. Using the information theoretic results that we obtain in the present paper for the set-disjointness function, we are able to give a lower bound of $\Omega(n)$ on the randomness complexity of Disj_k^n. The significance of this result lies in that it is the first such lower bound that grows with the size of the input (which is kn), while the output remains a single bit, contrary to the sum function (see [6]) or the bitwise parity function (see [31]).

1.1 Our techniques and contributions

Our contribution in the present paper is twofold.

First, on the conceptual, modeling and definitions side we lay the foundations for proving lower bounds on (randomized) communication complexity in the natural peer-to-peer multi-party setting. Specifically, we propose a model that, on the one hand, is a very natural peer-to-peer model, and very close to the model used in the distributed computation literature, and, at the same time, does have properties that allow one to analyze protocols in terms of their information complexity and communication complexity. While at first sight the elaboration of such model does not seem to be a difficult task, many technical, as well as fundamental, issues render this task non-trivial. For example, one would like to define a notion of "transcript" that would guarantee both a relation between the length of the transcript and the communication complexity, and at the same time will contain *all* the information that the players get and use while running the protocol. The difficulty in elaborating such model may be the reason for which, prior to the present paper, hardly any work studied communication complexity directly in a *peer-to-peer*, multi-party setting (cf. [22]), leaving the field with only the results that can be inferred from other models, hence suffering the appropriate loss in the obtained bounds. We propose our model (see Section 2.1) and prove a number of fundamental properties that allow one to analyze protocols in that model (see Section 3.2), as well as prove the accurate relationship between the entropy of the transcript and the communication complexity of the protocol (Proposition 2.4).

We then define our new information theoretic measure, that we call "Multi-party Information Cost"

(MIC), intended to be applied to peer-to-peer multi-party protocols, and prove that it provides, for any (possibly randomized) protocol, a lower bound on the communication complexity of that protocol (Lemma 3.4). We further show that MIC has certain properties such as a certain direct-sum property (Theorem 3.5). We thus introduce a framework as well as tools for proving lower bounds on communication complexity in a peer-to-peer multi-party setting.

Second, we exemplify the effectiveness of our conceptual contributions by proving, using the new tools that we define, two tight lower bounds on the randomized communication complexity of certain functions in the peer-to-peer multi-party setting. Both these lower bounds are proved by giving a lower bound on the Multi-party Information Complexity of the function at hand. The more elaborate lower bound is a tight lower bound of $\Omega(nk)$ on the randomized communication complexity of the function Disj_k^n (under the condition that $n = \Omega(k)$). The function Disjointness is a well studied function in communication complexity and is often seen as a test-case of one's ability to give lower bounds in a given model (cf. [16]). While the general structure of the proof of this lower bound does have similarities to the proof of a lower bound for Disjointness in the coordinator model [8],[2] we do, even in the parts that bear similarities, have to overcome a number of technical difficulties that require new ideas and new proofs. For example, the very basic *rectangularity* property of communication protocols is, in the multi-party (peer-to-peer) setting, very sensitive to the details of the definition of the model and the notion of a transcript. We therefore need first to give a proof of this property in the peer-to-peer model (Lemma 3.6 and Lemma 3.7). We then use a distribution of the input which is a modification over the distributions used in [8, 15] (see Section 5). Our proof proceeds, as in [8], by proving a lower bound for the function AND, on a certain information theoretic measure that, in our proof, is called SMIC (for Switched Multi-party Information Cost), and then, by using a direct-sum-like lemma, to infer a lower bound on SMIC for Disjointness (we note that SMIC is an adaptation to the peer-to-peer model of a similar measure used in [8]). However, the lack of a "coordinator" in a peer-to-peer setting necessitates a definition of a more elaborate reduction protocol, and a more complicated proof for the direct-sum argument, inspired by classic secret-sharing primitives. See Lemma 6.1 for our construction and proof. We then show that SMIC provides a lower bound on MIC, which yields our lower bound on the communication complexity of Disjointness.

We further give a tight lower bound of $\Omega(nk)$ on the randomized communication complexity of the function Par_k^n (bitwise parity) in the peer-to-peer multi-party setting (under the condition that $n = \Omega(k)$). This proof proceeds by first giving a lower bound on MIC for the parity function Par_k^1, and then using a direct-sum property of MIC to get a lower bound on MIC for Par_k^n. The latter yields the lower bound of $\Omega(nk)$ on the communication complexity of Par_k^n.

To the best of our knowledge, our lower bounds are the first tight (non-trivial) lower bound on communication complexity in a peer-to-peer multi-party setting.

In addition to our results on communication complexity, we analyze the number of random bits necessary for private computations [5, 19], making use of the model, tools and techniques we develop in the present paper. It has been shown [31] that the *public information cost* (defined also in [31]) can be used to derive a lower bound on the randomness complexity of private computations. In the present paper we give a lower bound on the public information cost of any synchronous protocol computing the Disjointness function by relating it to its Switched Multi-party Information Cost, which yields the lower bound on the randomness complexity of Disjointness.

Organization. The appendix contains a short review of information theoretic notions that we use in the present paper. We start the paper, in Section 2, by introducing our model and by comparing it to other models. In Section 3 we define our new information theoretic measure, MIC, and prove some of its properties,

[2] The lower bound in [8] would yield an $\Omega(\frac{1}{\log k} \cdot nk)$ lower bound in the peer-to-peer setting.

and then prove a number of fundamental properties of protocols in our peer-to-peer model. In Section 4 we give the lower bound for the bitwise parity function. In Section 5 we prove a lower bound on the switched multi-party information cost of the function AND_k, and in Section 6, we prove, using the results of Section 5, the lower bound on the communication complexity of the disjointness function Disj_k^n. In Section 7, we show how to apply our information theoretic lower bounds in order to give a lower bound on the number of random bits necessary for the private computation of the function Disj_k^n. Last, in Section 8 we discuss some open questions.

2 Multi-party communication protocols

We start with our model, and, to this end, give a number of notations.

Notations. We denote by k the number of players. We often use n to denote the size (in bits) of the input to each player. Calligraphic letters will be used to denote sets. Upper case letters will be used to denote random variables, and given two random variables A and B, we will denote by AB the joint random variable (A, B). Given a string (of bits) s, $|s|$ denotes the length of s. Using parentheses we denote an ordered set (family) of items, e.g., (Y_i). Given a family (Y_i), Y_{-i} denotes the sub-family which is the family (Y_i) *without* the element Y_i. The letter X will usually denote the input to the players, and we thus use the shortened notation X for (X_i), i.e., the input to all players. A protocol will usually be denoted by π.

We now define a natural communication model which is a slight restriction of the general asynchronous peer-to-peer model. The restriction of our model compared to the general asynchronous peer-to-peer model is that for a given player at a given time, the set of players from which that player waits for a message before sending any message of its own is determined by that player's own local view, i.e., from that player's input and the messages it has read so far, as well as its private randomness, and the public randomness. This allows us to define information theoretic tools that pertain to the transcripts of the protocols, and at the same time to use these tools as lower bounds for communication complexity. This restriction however does not exclude the existence of private protocols, as other special cases of the general asynchronous model do. We observe that practically all multi-party protocols in the literature are implicitly defined in our model, and that without such restriction, one bit of communication can bring $\log k$ bits of information, because not only the content of the message, but also the identity of the sender may reveal information. To exemplify why the general asynchronous model is problematic consider the following simple example (that we borrow from our work in [31]).

Example 2.1. *There are 4 players A, B and C, D. The protocol allows A to transmit to B its input bit x. But all messages sent in the protocol are the bit 0, and the protocol generates only a single transcript over all possible inputs. The protocol works as follows:*

A: *If $x = 0$ send 0 to C; after receiving 0 from C, send 0 to D.*
If $x = 1$ send 0 to D; after receiving 0 from D, send 0 to C

B: *After receiving 0 from a player, send 0 back to that player.*

C,D: *After receiving 0 from A send 0 to B. After receiving 0 from B send 0 to A.*

It is easy to see that B learns the value of x from the order of the messages it gets.

In what follows we formally define our model, compare it to the general one and to other restricted ones, and explain the usefulness and logic of our specific model.

2.1 Definition of the model

We work in a *multi-party, number-in-hand, peer-to-peer* setting. Each player $1 \leq i \leq k$ has unbounded local computation power and, in addition to its input X_i, has access to a source of private randomness R_i. We will use the notation R for (R_i), i.e., the private randomness of all players. A source of public randomness R^p is also available to all players. We will call a protocol with no private randomness a public-coins protocol. The system consists of k players and a family of k functions $f = (f_i)_{i \in [\![1,k]\!]}$, with $\forall\, i \in [\![1,k]\!],\ f_i : \Pi_{\ell=1}^{k} \mathcal{X}_\ell \to \mathcal{Y}_i$, where $\mathcal{X}_\ell$ denotes the set of possible inputs of player ℓ, and $\mathcal{Y}_i$ denotes the set of possible outputs of player i. The players are given some input $x = (x_i) \in \Pi_{i=1}^{k} \mathcal{X}_i$, and for every i, player i has to compute $f_i(x)$.

We define the communication model as follows, which is the asynchronous setting, with some restrictions. To make the discussion simpler we assume a global time which is *unknown* to the players. Every pair of players is connected by a bidirectional communication link that allows them to send messages to each other. There is no bound on the delivery time of a message, but every message is delivered in finite time, and the communication link maintains FIFO order in each of the two directions. Given a specific time we define the *view* of player i as the input of this player, X_i, its private randomness, R_i, the public randomness, R^p, and the messages read so far by player i. After the protocol has started, each player runs the protocol in *local rounds*. In each round, player i sends messages to some subset of the other players. The identity of these players, as well as the content of these messages, depend on the current view of player i. The player also decides whether it should stop, and output (or "return") the result of the function f_i. Then (if player i did not stop and return the output), the player waits for messages from a certain subset of the other players, this subset being also determined by the current view of the player. Then the (local) round of player i terminates.[3] To make it possible for the player to identify the arrival of the *complete* message that it waits for, we require that each message sent by a player in the protocol is self-delimiting.

Denote by $\mathcal{D}_i^\ell$ the set of possible views of player i at the end of local round ℓ, $\ell \geq 0$, where the beginning of the protocol is considered round 0.
Formally, a **protocol** π is defined by a set of local programs, one for each player i, where the local program of player i is defined by a sequence of functions, parametrized by the index of the *local* round ℓ, $\ell \geq 1$:

- $S_i^{\ell,s} : \mathcal{D}_i^{\ell-1} \to 2^{\{1,\ldots,k\}\setminus\{i\}}$, defining the set of players to which player i *sends* the messages.
- $m_{i,j}^\ell : \mathcal{D}_i^{\ell-1} \to \{0,1\}^*$, such that for any $D_i^{\ell-1} \in \mathcal{D}_i^{\ell-1}$, if $j \in S_i^{\ell,s}(D_i^{\ell-1})$, then $m_{i,j}^\ell(D_i^{\ell-1})$ is the content of the message player i sends to player j. Each such message is self-delimiting.
- $O_i^\ell : \mathcal{D}_i^{\ell-1} \to \{0,1\}^* \cup \{\perp\}$, defining whether or not the local program of player i stops and the player returns its output, and what is that output. If the value is $\perp$ then no output occurs. If the value is $y \in \{0,1\}^*$, then the local program stops and the player returns the value y.
- $S_i^{\ell,r} : \mathcal{D}_i^{\ell-1} \to 2^{\{1,\ldots,k\}\setminus\{i\}}$, defining the set of players from which player i waits to *receive* a message.

To define the **transcript** of a protocol we proceed as follows. We first define $k(k-1)$ basic transcripts $\Pi_{i,j}^r$, denoting the transcript of the messages read by player i from its link from player j, and another $k(k-1)$ basic transcripts $\Pi_{i,j}^s$, denoting the transcript of the messages sent by player i on its link to player j.
We then define the transcript of player i, Π_i, as the $2(k-1)$-tuple of the $2(k-1)$ basic transcripts $\Pi_{i,j}^r, \Pi_{i,j}^s$, $j \in [\![1,k]\!] \setminus \{i\}$. The transcript of the whole protocol Π is defined as the k-tuple of the k player transcripts

[3] The fact that the receiving of the incoming messages comes as the last step of the (local) round comes only to emphasize that the sending of the messages and the output are a function of only the messages received in previous (local) rounds.

Π_i, $i \in [\![1,k]\!]$. We denote by $\Pi_i(x,r)$ the transcript of player i when protocol π is run on input x and on randomness (public and private of all players) r. By $\Pi_i^\ell(x,r)$ we denote $\Pi_i(x,r)$ modified such that all the messages that player i sends in local rounds $\ell' > \ell$, and all the messages that player i reads in local rounds $\ell' > \ell$ are eliminated from the transcript. Observe that while $\Pi_{i,j}^r$ is always a prefix of $\Pi_{j,i}^s$, the definition of a protocol does not imply that they are equal. Further observe that each bit sent in π appears in Π at most twice.

We note that while seemingly the model that we introduce here is the same as the one used in [31], there are important differences between the models, and that these differences are crucial for the properties that we prove in the present paper to hold. See Section 2.2 for a comparison.

For a k-party protocol π we denote the set of possible inputs as $\mathcal{X}$, and denote the projection of this set on the i'th coordinate (i.e., the set of possible inputs for player i) by $\mathcal{X}_i$. Thus $\mathcal{X} \subseteq \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$. The set of possible transcripts for a protocol is denoted $\mathcal{T}$, and the projection of this set on the i'th coordinate (i.e., the set of possible transcripts of player i) is denoted $\mathcal{T}_i$. Observe that $\mathcal{T} \subseteq \mathcal{T}_1 \times \cdots \times \mathcal{T}_k$.

Furthermore, in the course of the proofs, we sometimes consider a protocol that does *not* have access to public randomness (but may have private randomness). We call such protocol a *private-coins protocol*.

We now formally define the notion of a protocol computing a given function with certain bounded error. We will give most of the following definitions for the case where all functions f_i are the same function, that we denote by f. The definitions in the case of family of functions are similar.

Definition 2.2. *For a given $0 \le \epsilon < 1$, a protocol π ϵ-computes a function f if for all $x \in \Pi_{i=1}^k \mathcal{X}_i$:*

- *For all possible assignments for the random sources R_i, $1 \le i \le k$, and R^p, every player eventually stops and returns an output.*
- *With probability at least $1-\epsilon$ (over all random sources) the following event occurs: each player i outputs the value $f(x)$, i.e., the correct value of the function.*

We also consider the notion of *external computation*.

Definition 2.3. *For a given $0 \le \epsilon < 1$, a protocol π* externally *ϵ-computes f if there exists a deterministic function θ taking as input the possible transcripts of π and verifying $\forall\, x \in \mathcal{X},\ \Pr[\theta(\Pi(x)) = f(x)] \ge 1-\epsilon$.*

The **communication complexity** of a protocol is defined as the worst case, over the possible inputs and the possible randomness, of the number of bits sent by all players. For a protocol π we denote its communication complexity by $\mathsf{CC}(\pi)$. For a given function f and a given $0 \le \epsilon < 1$, we denote by $\mathsf{CC}^\epsilon(f)$ the *ϵ-error communication complexity of f*, i.e., $\mathsf{CC}^\epsilon(f) = \inf_{\pi\ \epsilon\text{-computing}\ f} \mathsf{CC}(\pi)$.

Finally, we give a proposition that relates the communication complexity of a k-party protocol π to the entropy of the transcripts of the protocol π.

Proposition 2.4. *Let the input to a k-party protocol π be distributed according to an arbitrary distribution. Then, $\sum_{i=1}^k H(\Pi_i) \le 4 \cdot \mathsf{CC}(\pi) + 4k^2$, where the entropy is according to the input distribution and the randomization of protocol π.*

Proof. We first encode Π_i, for any i, into a variable Π_i' such that the set of possible values of Π_i' is a prefix-free set of strings. Observe that the transcript Π_i is composed of a number of basic transcripts: for every $j \in [\![1,k]\!] \setminus \{i\}$, a pair of transcripts of messages, $\Pi_{i,j}^s$, $\Pi_{i,j}^r$ containing the messages sent by player i to player j, and the messages read by player i from player j, respectively. We convert Π_i into Π_i' as follows: In each one of the above $2(k-1)$ components we replace every bit $b \in \{0,1\}$ by $b.b$, and then add at the end of the component the two bits 01. We then concatenate all components in order. Clearly this a one-to-one encoding, and the set of possible values of Π'_i is a prefix-free set of strings.

Defining $|\Pi_i| = \sum_{j \neq i} |\Pi^s_{i,j}| + |\Pi^r_{i,j}|$ and $|\Pi| = \sum_{i=1}^{k} |\Pi_i|$, we have $H(\Pi'_i) = H(\Pi_i)$, and $\mathbb{E}[|\Pi'_i|] = 2\,\mathbb{E}[|\Pi_i|] + 4(k-1)$.

We get

$$
\begin{aligned}
\sum_{i=1}^{k} H(\Pi_i) &= \sum_{i=1}^{k} H(\Pi'_i) \\
&\leq \sum_{i=1}^{k} \mathbb{E}[|\Pi'_i|] \quad \text{(by Theorem A.3)} \\
&\leq \sum_{i=1}^{k} \left(2\,\mathbb{E}[|\Pi_i|] + 4(k-1)\right) \\
&\leq 2 \cdot \mathbb{E}[|\Pi|] + 4k^2 \\
&\leq 4 \cdot \mathsf{CC}(\pi) + 4k^2 \ ,
\end{aligned}
$$

where the last factor of 2 is due to the fact that each message sent from, say, player i to player j, may appear in at most 2 basic transcripts $\Pi^s_{i,j}$ and $\Pi^r_{j,i}$. ⌟

2.2 Comparison to other models

The somewhat restricted model (compared to the general asynchronous model) that we work with allows us to use information theoretic tools for the study of protocols in this model, and in particular to give lower bounds on the multi-party communication complexity. Notice that the general asynchronous model is problematic in this respect since one bit of communication can bring $\log k$ bits of information, because not only the content of the message, but also the identity of the sender may reveal information. Thus, information cannot be used as a lower bound on communication. In our case, the sets $S_i^{l,r}$ and $S_i^{l,s}$ are determined by the current view of the player, Π contains only the content of the messages, and thus the desirable relation between the communication and the information is maintained. On the other hand, our restriction is natural, does not seem to be very restrictive (practically all protocols in the literature adhere to our model), and does not exclude the existence of private protocols. To exemplify why the general asynchronous model is problematic see Example 2.1.

While the model that we introduce in the preset paper bears some similarities to the model used in [31], there are a number of important differences between them. First, the definition of the transcript is different, resulting in a different relation between the entropy of the transcript and the communication complexity. More important is the natural property of the model in the present paper that the local program of a protocol in a given node ends its execution when it locally gives its output. It turns out that the very basic rectangularity property of protocols, used in many papers, holds in this case (and when the transcript is defied as we define in the present paper), while if the local protocol may continue to operate after output, there are examples where this property does not hold. Thus, we view the introduction of the present model also as a contribution towards identifying the necessary features of a peer-to-peer model so that basic and useful properties of protocols hold in the peer-to-peer setting.

There has been a long series of works about multi-party communication protocols in different variants of models, for example [21, 13, 27, 29, 41, 17, 18] (see [22] for a comparison of a few of these models). In the *coordinator model* (cf. [21, 41, 8]), an additional player (the coordinator) with no input can communicate privately with each player, and the players can only communicate with the coordinator. We first note that the coordinator model does not yield exact bounds for the multi-party communication complexity in the peer-to-peer setting (neither in our model nor in the most general one). Namely, any protocol in the peer-to-peer model can be transformed into a protocol in the coordinator model with an $O(\log k)$ multiplicative factor in the communication complexity, by sending each message to the coordinator with an $O(\log k)$-bit label indicating its destination. This factor is sometimes necessary, e.g., for the `permutation` functional defined as follows: Given a permutation $\sigma : [\![1, k]\!] \to [\![1, k]\!]$, each player i has as input a bit b_i and $\sigma^{-1}(\sigma(i) - 1)$ and $\sigma^{-1}(\sigma(i) + 1)$ (i.e., each player has as input the indexes of the players before and after itself in the permutation).[4] For player i the function f_i is defined as $f_i = b_{\sigma^{-1}(\sigma(i)+1)}$ (i.e., the value of the input bit of the next player in the permutation σ). Clearly in our model the communication complexity of this function is k (each player sends its input bit to the correct player), and the natural protocol is valid in our model. On the other hand, in the coordinator model $\Omega(k \log k)$ bits of communication are necessary. But this multiplicative factor between the complexities in the two models is not always necessary: the communication complexity of the parity function Par is $\Theta(k)$ both in the peer-to-peer model and in the coordinator model.

Moreover, when studying private protocols in the multi-party setting, the coordinator model does not offer any insight. In the coordinator model, described in [21] and used for instance in [8], if one does not impose any privacy requirement with respect to the coordinator, it is trivial to have a private protocol by all players sending their input to the coordinator, and the coordinator returning the results to the players. If there is a privacy requirement with respect to the coordinator, then if there is a random source shared by all the players (but not the coordinator), privacy is always possible using the protocol of [23]. If no such source exists, privacy is impossible in general. This follows from the results of Braverman et al. [8] who show a non-zero lower bound on the total internal information complexity of all parties (including the coordinator) for the function *Disjointness* in that model. Our model, on the other hand, does allow for the private computation of any function [5, 19, 1].

It is worthwhile to contrast our model, and the communication complexity measure that we are concerned with, with work in the so-call congested-clique model that has gained increasing attention in the distributed computation literature (cf. [34, 35]). While both models are based on a communication network in the form of a complete graph (i.e., every player can send messages to any other player, and these messages can be different) there are two significant differences between them. Most of the works in the congested clique model deal with graph-theoretic problems and the input to each player is related to the adjacency list of a node (identified with that player) in the input graph, while in our model the input is not associated in any way with the communication graph. More importantly, the congested clique model is a *synchronous* model while ours is an asynchronous one. This brings about a major difference between the complexity measures studied in each of the models. Work in the congested clique model is concerned with giving bounds on the number of rounds necessary to fulfill a certain task under the condition that in each round each player can send to any other player a limited number of bits (usually $O(\log k)$ bits). The measure of communication complexity, that is of interest to us in the present paper, deals with the total number of communication bits necessary to fulfill a certain task in an asynchronous setting without any notion of global rounds.[5]

[4] All additions are modulo k. This is a promise problem.

[5] Any function can be computed in the congested clique model with $O(k)$ communication complexity (at a cost of having many rounds) by each player, having input x, sending a single bit to player 1 only at round number x. On the other hand, in the asynchronous model any function can be computed in a single "round" (at a cost of high communication complexity) by each player sending its whole input to player 1.

3 Tools for the study of multi-party communication protocols

In this section we consider two important tools for the study of peer-to-peer multi-party communication protocols. First, we define and introduce an information theoretic measure that we call *Multi-party Information Cost* (MIC); we later use it to prove our lower bounds. Then, we prove, in the peer-to-peer multi-party model that we define, the so-called rectangularity property of communication protocols, that we also use in our proofs.

3.1 Multi-party Information Cost

We now introduce an information theoretic measure for multi-party peer-to-peer protocols that we later show to be useful for proving lower bounds on the communication complexity of multi-party peer-to-peer protocols. We note that a somewhat similar measure was proposed in [8] for the coordinator model, but, to the best of our knowledge, never found an application as a tool in a proof of a lower bound.

Definition 3.1. *For any k-player protocol π and any input distribution μ, we define the multi-party information cost of π:*

$$\mathsf{MIC}_\mu(\pi) = \sum_{i=1}^{k} \left(I(X_{-i}; \Pi_i \mid X_i R_i) + I(X_i; \Pi_i \mid X_{-i} R_{-i}) \right) \ .$$

Observe that the second part of each of the k summands can be interpreted as the information that player i "leaks" to the other players on its input. While the "usual" intuitive interpretation of two-party IC is "what Alice learns on Bob's input plus what Bob learns on Alice's input", one can also interpret two-party IC as "what Alice learns on Bob's input plus what Alice leaks on her input". Thus, MIC can be interpreted as summing over all players i of "what player i learns on the other players' inputs, plus what player i leaks on its input." Indeed, the expression defining MIC is equal to the sum, over all players i, of the two-party IC for the two-party protocol that results from collapsing all players, except i, into one virtual player. Thus, for number of players $k = 2$, $\mathsf{MIC} = 2 \cdot \mathsf{IC}$. We note that defining our measure without the private randomness in the condition of the mutual information expressions would yield the exact same measure (as is the case for 2-party IC); we prefer however to define MIC with the randomness in the conditions, as we believe that it allows one to give shorter, but still clear and accurate, proofs.

On the other hand observe that the second of the two mutual information expressions has X_{-i} in the condition, contrary to a seemingly similar measure used in [8] (Definition 3 in [8]). Our measure is thus "internal" in nature, while the one of [8] has an "external" component. The fact that MIC is "internal" allows us to give lower bounds on MIC, and thus to use it for lower bounds on the communication complexity, contrary to the measure of [8].

Further observe that the summation, over all players, of each one of the two mutual information expressions alone would not yield a measure useful for proving lower bounds on the communication complexity of functions. The first mutual information expression would yield a measure for functions that would never be higher than the entropy of the function at hand, due to the existence of private protocols for all functions [5, 19]. For the second mutual information expression there are functions for which that measure would be far too low compared to the communication complexity: e.g., the function $f = x_1$, $x \in \{0,1\}^n$ (i.e., the value of the function is the input of player 1); in that case the measure would equal only n, while the communication complexity of that function is $\Omega(kn)$.

We now define the multi-party information complexity of a function.

Definition 3.2. *For any function f, any input distribution μ, and any $0 \le \epsilon \le 1$, we define the quantity*

$$\mathsf{MIC}^{\epsilon}_{\mu}(f) = \inf_{\pi\ \epsilon\text{-computing } f} \mathsf{MIC}_{\mu}(\pi)\ .$$

Definition 3.3. *For any f, and any $0 \le \epsilon \le 1$, we define the quantity*

$$\mathsf{MIC}^{\epsilon}(f) = \inf_{\pi\ \epsilon\text{-computing } f} \sup_{\mu}\ \mathsf{MIC}_{\mu}(\pi)\ .$$

We now claim that the multi-party information cost and the communication complexity of a protocol are related, as formalized by the following lemma.

Lemma 3.4. *For any k-player protocol π, and for any input distribution μ,*

$$\mathsf{CC}(\pi) \ge \frac{1}{8}\mathsf{MIC}_{\mu}(\pi) - k^2\ .$$

Proof.

$$\begin{aligned}
\mathsf{MIC}_{\mu}(\pi) &= \sum_{i=1}^{k} \left(I(X_{-i};\Pi_i \mid X_i R_i) + I(X_i;\Pi_i \mid X_{-i} R_{-i}) \right) \\
&\le 2\sum_{i=1}^{k} H(\Pi_i) \\
&\le 8\cdot \mathsf{CC}(\pi) + 8k^2\ ,
\end{aligned}$$

where the first inequality follows from Proposition A.2, and the last one from Proposition 2.4. ⌟

We now show that the multi-party information cost satisfies a direct sum property for product distributions. In what follows, the notation $f^{\otimes n}$ denotes the task of computing n instances of f, where the requirement from an ϵ-computing protocol is that each instance is computed correctly with probability at least $1-\epsilon$ (as opposed to the stronger requirement that the whole vector of instances is computed correctly with probability at least $1-\epsilon$).

Theorem 3.5. *For any protocol π (externally) ϵ-computing a function $f^{\otimes n}$, there exists a protocol π' (externally) ϵ-computing f such that, for any product distribution μ for the input, it holds that*

$$\mathsf{MIC}_{\mu^n}(\pi) \ge n\cdot \mathsf{MIC}_{\mu}(\pi')\ .$$

Proof. We define π' on input $(Y_i)_{i\in[\![1,k]\!]}$ as follows. We denote by R'^p the public randomness available to the players, and by R'_i the private randomness available to the players.

We consider the public and private randomness R'_i, $1 \le i \le k$, and R'^p as strings of random bits. The players first use the first bits of the public randomness to publicly sample a random index L uniformly in $[\![1,n]\!]$, and define $X_i^L = Y_i$. The players then, using the next random bits of the public randomness, publicly sample, for every $d < L$, X^d according to μ. Each player i then, using the first bits of its private randomness, samples privately, for every $d > L$, X_i^d according to μ. The player then run π on input X. They output as the output of π' the L'th coordinate of the output of π. Observe that π' has error at most ϵ, and that if the input to π' is distributed according to μ, then the input of π is distributed according to μ^n.

Note that there is no extra communication in π' compared to π, only some (private and public) sampling. Therefore we have $\Pi'_i = \Pi_i$ for every $1 \le i \le k$. We further denote by R^p the random bits of R'^p beyond those used by the public sampling at the start of π'. Similarly, we denote by R_i, $1 \le i \le k$, the random bits of R'_i beyond those used by the private sampling at the start of π'.

We now show that $\mathsf{MIC}_\mu(\pi') = \frac{1}{n}\mathsf{MIC}_{\mu^n}(\pi)$. In what follows we explicitly state the public randomness next to the transcript. Thus,

$$\mathsf{MIC}_\mu(\pi') = \sum_{i=1}^{k} \left(I(Y_{-i}; R'^p\Pi'_i \mid Y_iR'_i) + I(Y_i; R'^p\Pi'_i \mid Y_{-i}R'_{-i}) \right).$$

We have, for every player i,

$$\begin{aligned}
I(Y_{-i}; R'^p\Pi'_i \mid Y_iR'_i) &= I(Y_{-i}; LX^{<L}R^p\Pi_i \mid Y_iX_i^{>L}R_i) \quad \text{(making explicit the sampling from } R'_i, R'^p\text{)} \\
&= I(Y_{-i}; X^{<L}R^p\Pi_i \mid Y_iX_i^{>L}R_i) \quad \text{(because } I(Y_{-i}; L \mid Y_iX_i^{>L}R_iX^{<L}R^p\Pi_i) = 0\text{)} \\
&= I(Y_{-i}; R^p\Pi_i \mid Y_iX_i^{>L}R_iX^{<L}) \quad \text{(because } Y_{-i} \text{ and } X^{<L} \text{ are independent)} \\
&= \mathop{\mathbb{E}}_{\ell}[I(X^{\ell}_{-i}; R^p\Pi_i \mid X_i^{\ell}X_i^{>\ell}R_iX^{<\ell})] \\
&= \mathop{\mathbb{E}}_{\ell}[I(X^{\ell}_{-i}; R^p\Pi_i \mid X_i^{\ell}X_i^{>\ell}R_iX_i^{<\ell}X_{-i}^{<\ell})] \\
&= \mathop{\mathbb{E}}_{\ell}[I(X^{\ell}_{-i}; R^p\Pi_i \mid X_iR_iX_{-i}^{<\ell})] \\
&= \frac{1}{n}\sum_{\ell}[I(X^{\ell}_{-i}; R^p\Pi_i \mid X_iR_iX_{-i}^{<\ell})] \\
&= \frac{1}{n}I(X_{-i}; R^p\Pi_i \mid X_iR_i) \quad \text{(chain rule)},
\end{aligned}$$

and

$$\begin{aligned}
I(Y_i; R'^p\Pi'_i \mid Y_{-i}R'_{-i}) &= I(Y_i; LX^{<L}R^p\Pi_i \mid Y_{-i}X_{-i}^{\geq L}R_{-i}) \quad \text{(making explicit the sampling from } R'_i, R'^p\text{)} \\
&= I(Y_i; X^{<L}R^p\Pi_i \mid Y_{-i}X_{-i}^{\geq L}R_{-i}) \quad \text{(because } I(Y_i; L \mid Y_{-i}X_{-i}^{\geq L}R_{-i}X^{<L}R^p\Pi_i) = 0\text{)} \\
&= I(Y_i; R^p\Pi_i \mid Y_{-i}X_{-i}^{\geq L}R_{-i}X^{<L}) \quad \text{(because } Y_i \text{ and } X^{<L} \text{ are independent)} \\
&= \mathop{\mathbb{E}}_{\ell}[I(X_i^{\ell}; R^p\Pi_i \mid X^{\ell}_{-i}X_{-i}^{\geq \ell}R_{-i}X_{-i}^{\leq \ell}X_i^{<\ell})] \\
&= \mathop{\mathbb{E}}_{\ell}[I(X_i^{\ell}; R^p\Pi_i \mid X_{-i}R_{-i}X_i^{<\ell})] \\
&= \frac{1}{n}\sum_{\ell}[I(X_i^{\ell}; R^p\Pi_i \mid X_{-i}R_{-i}X_i^{<\ell})] \\
&= \frac{1}{n}I(X_i; R^p\Pi_i \mid X_{-i}R_{-i}) \quad \text{(chain rule)}.
\end{aligned}$$

Summing over $i \in [\![1, k]\!]$ concludes the proof. ⌟

3.2 The rectangularity property

Rectangularity. The *rectangularity property* (or *Markov property*) is one of the key properties that follow from the structure and definition of (some) protocols. For randomized protocols it was introduced in the two-party setting and in the multi-party blackboard model in [3], and in the coordinator model in [8]. We prove a similar rectangularity property in the peer-to-peer model that we consider in the present paper.

We note that the proof of this property in the peer-to-peer model makes explicit use of the specific properties of the model we defined: the proof that follows explicitly uses the definition of the transcript on an edge by edge basis as in our model, as well as the fact that a player returns and stops as one operation. One can build examples where if any of these two properties does not hold, then the rectangularity property of protools does not hold. Thus we view the following proof of rectangularity in our model also as an identification of model properties needed for the useful rectangularity property of multiparty peer-to-peer protocols to hold.

To define this property, for any transcript $\overline{\tau} \in \mathcal{T}_i$, let $\mathcal{A}_i(\overline{\tau}) = \{(x, r) \mid \Pi_i(x, r) = \overline{\tau}\}$ (i.e., the set of input, randomness pairs that lead to transcript $\overline{\tau}$), and define the projection of $\mathcal{A}_i(\overline{\tau})$ on coordinate i as

$$\mathcal{I}_i(\overline{\tau}) = \{(x', r'), \exists\, (x, r) \in \mathcal{A}_i(\overline{\tau}), x' = x_i \,\&\, r' = r_i\} \ ,$$

and the projection of $\mathcal{A}_i(\overline{\tau})$ on the complement of coordinate i as

$$\mathcal{J}_i(\overline{\tau}) = \{(x', r'), \exists\, (x, r) \in \mathcal{A}_i(\overline{\tau}), x' = x_{-i} \,\&\, r' = r_{-i}\} \ .$$

Similarly, for any transcript $\tau \in \mathcal{T}$, let $\mathcal{B}(\tau) = \{(x, r) \mid \Pi(x, r) = \tau)\}$, and for any player i, let $\mathcal{H}_i(\tau) = \{(x', r'), \exists\, (x, r) \in \mathcal{B}(\tau), x' = x_{-i} \,\&\, r' = r_{-i}\}$.

We start by proving a combinatorial property of transcripts of communication protocols, which intuitively follows from the fact that each player has access to only its own input and private randomness. The proof of this property is technically more involved compared to the analogous property in other settings, since the structure of protocols and the manifestation of the transcripts in the peer-to-peer setting are more flexible than in the other settings.

Lemma 3.6. *Let π be a k-player private-coins protocol [6] with inputs from $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$. Let $\mathcal{T}$ denote the set of possible transcripts of π, and for $i \in [\![1, k]\!]$ let $\mathcal{T}_i$ denote the set of possible transcript observed by player i, so that $\mathcal{T} \subseteq \mathcal{T}_1 \times \cdots \times \mathcal{T}_k$. Then, $\forall\, i \in [\![1, k]\!]$:*

- $\forall\, \overline{\tau} \in \mathcal{T}_i,\ \ \mathcal{A}_i(\overline{\tau}) = \mathcal{I}_i(\overline{\tau}) \times \mathcal{J}_i(\overline{\tau})$.
- $\forall\, \tau \in \mathcal{T},\ \ \mathcal{B}(\tau) = \mathcal{I}_i(\tau_i) \times \mathcal{H}_i(\tau)$.

Proof. We start by proving the first claim. Since the other inclusion is immediate from the definition, we only need to show that

$$\forall\, \overline{\tau} \in \mathcal{T}_i,\ \mathcal{I}_i(\overline{\tau}) \times \mathcal{J}_i(\overline{\tau}) \subseteq \mathcal{A}_i(\overline{\tau}) \ .$$

To this end take an arbitrary $(x_i, r_i) \in \mathcal{I}_i(\overline{\tau})$ and an arbitrary $(x_{-i}, r_{-i}) \in \mathcal{J}_i(\overline{\tau})$. Since $(x_i, r_i) \in \mathcal{I}_i(\overline{\tau})$, we have that $\exists\, (\tilde{x}, \tilde{r}) \in \mathcal{A}_i(\overline{\tau})$ s.t. $x_i = \tilde{x}_i \,\&\, r_i = \tilde{r}_i$. Similarly, since $(x_{-i}, r_{-i}) \in \mathcal{J}_i(\overline{\tau})$,

$$\exists\, (\hat{x}, \hat{r}) \in \mathcal{A}_i(\overline{\tau}) \mid x_{-i} = \hat{x}_{-i} \,\&\, r_{-i} = \hat{r}_{-i} \ .$$

Let (x, r) be $((x_i, r_i), (x_{-i}, r_{-i})) \in \mathcal{I}_i(\overline{\tau}) \times \mathcal{J}_i(\overline{\tau})$. We will now show that

$$(x, r) \in \mathcal{A}_i(\overline{\tau}) \ .$$

Let L be the number of local rounds of player i in the run of π on input (x, r). We will show by induction on the index of the local round of player i that for any $\ell \leq L$, $\Pi_i^{\ell}(x, r) = \Pi_i^{\ell}(\tilde{x}, \tilde{r})$. Observe that whether or not the protocol of a player stops and returns its output at a given round is a function of its input and its

[6] Recall that a private-coins protocol does not have access to public randomness, but may have private randomness.

transcript until that round, as well as its private randomness. Therefore, since the protocol of player i stops and returns its value at local round L if the input is (x, r), it will follow from $\Pi_i^L(x, r) = \Pi_i^L(\tilde{x}, \tilde{r})$ that player i stops and returns its output at local round L also when the input is $(\tilde{x}, \tilde{r})$. We will thus get that $\Pi_i(x, r) = \overline{\tau}$, and hence $(x, r) \in \mathcal{A}_i(\overline{\tau})$.

The base of the induction, for $\ell = 0$, follows since the transcript is empty. We now prove the claim for $\ell + 1 \leq L$, based on the induction hypothesis that the claim holds for ℓ.[7]

The messages that player i sends at local round $\ell + 1$ are a function of x_i, r_i and $\Pi_i^\ell(x, r)$. As $x_i = \tilde{x}_i$ and $\tilde{r}_i = r_i$, and using the induction hypothesis, we get that the messages sent by player i at local round $\ell + 1$ are the same in $\pi_i(x, r)$ and in $\pi_i(\tilde{x}, \tilde{r})$.

For the same reason we also get that the set of players from which player i waits for a message at round $\ell + 1$ is the same when π is run on input in (x, r) and on input $(\tilde{x}, \tilde{r})$.

We now claim that the messages read by player i at round $\ell + 1$ are the same when π is run on input (x, r) and on input $(\tilde{x}, \tilde{r})$. To this end we define an imaginary "protocol" ψ where player i sends in its first local round all the messages that it sends in $\overline{\tau}$, and the players in $Q_i = [\![1, k]\!] \setminus \{i\}$ run π.[8] Player i sends the messages on each link according to the order in $\overline{\tau}$.[9] The messages that the players in Q_i send in each of their local rounds are a function of their inputs (and their local randomness) and the messages they read from the links that connect to player i. Since $\Pi(\hat{x}, \hat{r}) = \overline{\tau}$, we can conclude that in ψ (when the input is $(\hat{x}, \hat{r})$) the messages sent by the players in Q_i (in particular, to player i) are the same as those sent in π on input $(\hat{x}, \hat{r})$.

Recall that we have proved above that when π is run on (x, r), the messages player i sends up to round $\ell + 1$ are consistent with $\overline{\tau}$. We therefore can consider now a "protocol" ψ' which is the same as ψ with the only difference that player i sends (in its first local round) only the messages of $\overline{\tau}$ it would have sent in $\pi(x, r)$ until (and including) round $\ell + 1$ (and not all the message it sends in $\overline{\tau}$). It follows that in ψ', when run on input $(\hat{x}, \hat{r})$, the sequences of messages sent from the players in Q_i to i are a prefix of the sequences they send in ψ. Since $x_{-i} = \hat{x}$ and $r_{-i} = \hat{r}$, the same claim holds when ψ' is run on (x, r). Observe now that when π is run on (x, r), at the time where player i is waiting at local round $\ell + 1$ for incoming messages it, has sent exactly the messages that player i sends in ψ'.

Using the induction hypothesis $\Pi_i^\ell(x, r) = \Pi_i^\ell(\tilde{x}, \tilde{r})$, the fact hat $x_i = \tilde{x}_i$ and $r_i = \tilde{r}_i$, and the fact that the set of players from which player i waits for a message at local round $\ell + 1$ is the same for input (x, r) and $(\tilde{x}, \tilde{r})$, we can conclude that the messages that player i reads while waiting for messages at local round $\ell + 1$ when π is run on (x, r) are consistent with the messages it would read when π is run on $(\tilde{x}, \tilde{r})$. Since player i running π must, by the definition of a protocol, reach its "return" statement, it must receive messages from all the players it is waiting for. We therefore conclude that the messages read by player i in local round $\ell + 1$ when π is run on (x, r) are the same as those it read when run on $(\tilde{x}, \tilde{r})$.

Together with the induction hypothesis, and the fact (proved above) that the messages sent by player i at local round $\ell+1$ are the same when π is run on in (x, r) and on $(\tilde{x}, \tilde{r})$, we have that $\Pi_i^{\ell+1}(x, r) = \Pi_i^{\ell+1}(\tilde{x}, \tilde{r})$.

We now prove the second claim. We only need to show that

$$\forall\, \tau \in \mathcal{T},\ \mathcal{I}_i(\tau_i) \times \mathcal{H}_i(\tau) \subseteq \mathcal{B}(\tau)\ ,$$

the other inclusion being immediate from the definitions, since $\mathcal{B}(\tau) \subseteq \mathcal{A}_i(\tau_i)$.

Take an arbitrary $(x_i, r_i) \in \mathcal{I}_i(\tau_i)$ and an arbitrary $(x_{-i}, r_{-i}) \in \mathcal{H}_i(\tau)$. Since $(x_{-i}, r_{-i}) \in \mathcal{H}_i(\tau)$, $\exists\ (\hat{x}, \hat{r})$ s.t. $\pi(\hat{x}, \hat{r}) = \tau$ and $x_{-i} = \hat{x}_{-i}$ and $r_{-i} = \hat{r}_{-i}$. Let $(x, r) = ((x_i, r_i), (x_{-i}, r_{-i}))$. Since

[7] Note that $\Pi_i(x, r)$ by itself does not define which messages are sent/read in which local round.

[8] Technically speaking, this is not a protocol according to our definition as more than one message may be sent in a single round on a single link.

[9] Recall that a transcript of a players is a $2(k-1)$-tuple of transcripts, one for each of its $2(k-1)$ directed links.

$\mathcal{B}(\tau) \subseteq \mathcal{A}(\tau_i)$, we have $\mathcal{H}_i(\tau) \subseteq \mathcal{J}_i(\tau_i)$. Thus, using the first claim,

$$\mathcal{I}_i(\tau_i) \times \mathcal{H}_i(\tau) \subseteq \mathcal{I}_i(\tau_i) \times \mathcal{J}_i(\tau_i) \subseteq \mathcal{A}_i(\tau_i) \ ,$$

and $\Pi_i(x, r) = \tau_i$. It remains to show that $\forall\, j \neq i,\ \Pi_j(x, r) = \tau_j$.

Consider the two runs of protocol π on the input (x, r) and on the input $(\hat{x}, \hat{r})$. We have that $\Pi(\hat{x}, \hat{r}) = \tau$, and that $\Pi_i(x, r) = \tau_i$. Since $x_{-i} = \hat{x}_{-i}$ and $r_{-i} = \hat{r}_{-i}$, we have that also for all $j \neq i$ $\Pi_j(x, r) = \Pi_j(\hat{x}, \hat{r}) = \tau_j$. It follows that $(x, r) \in \mathcal{B}(\tau)$ as needed.

⌟

We now prove the *rectangularity property of randomized protocols* in the peer-to-peer setting. It follows from Lemma 3.6 and straightforward calculations. The full proof is given in the appendix.

Lemma 3.7. *Let π be a k-player private-coins protocol with inputs from $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$. Let $\mathcal{T}$ denote the set of possible transcripts of π, and for $i \in [\![1, k]\!]$ let $\mathcal{T}_i$ denote the set of possible transcript observed by player i, so that $\mathcal{T} \subseteq \mathcal{T}_1 \times \cdots \times \mathcal{T}_k$. Then for every $i \in [\![1, k]\!]$, there exist functions $q_i : \mathcal{X}_i \times \mathcal{T}_i \to [0, 1]$, $q_{-i} : \mathcal{X}_{-i} \times \mathcal{T}_i \to [0, 1]$ and $p_{-i} : \mathcal{X}_{-i} \times \mathcal{T} \to [0, 1]$ such that*

$$\forall\, x \in \mathcal{X}, \forall\, \tau = (\tau_1, \ldots, \tau_k) \in \mathcal{T}, \Pr[\Pi_i(x) = \tau_i] = q_i(x_i, \tau_i) q_{-i}(x_{-i}, \tau_i) \ ,$$

and

$$\forall\, x \in \mathcal{X}, \forall\, \tau = (\tau_1, \ldots, \tau_k) \in \mathcal{T}, \Pr[\Pi(x) = \tau] = q_i(x_i, \tau_i) p_{-i}(x_{-i}, \tau) \ .$$

The following lemma formalizes the fact that the distribution of the transcript of a protocol that externally-computes a function f must differ on two inputs with different values of f (see also [3]). The proof is deferred to the appendix.

Lemma 3.8. *Let f be a k-party function, and let π be a protocol externally ϵ-computing f. If x and y are two inputs such that $f(x) \neq f(y)$, then $h(\Pi(x), \Pi(y)) \geq \frac{1-2\epsilon}{\sqrt{2}}$.*

The Diagonal Lemma. The following lemma is often called the *diagonal lemma*. It was proved in [3] for the two-party setting under the name of the *Pythagorean lemma*, and in [8] for the coordinator model. We show here that is also holds in the peer-to-peer model. This lemma follows from Lemma 3.7 and Proposition A.12 in the same way that its two-party analogue follows from the analogous lemma and proposition. For completeness we give the proof in the appendix. For $x \in \{0, 1\}^k$ and $b \in \{0, 1\}$, let $x_{[i \leftarrow b]}$ represent the input obtained from x by replacing the i^{th} bit of x by b.

Lemma 3.9. *Let π be a k-party private-coins protocol taking input in $\{0, 1\}^k$. Then $\forall\, x \in \{0, 1\}^k$, $\forall\, y \in \{0, 1\}^k$, $\forall\, i \in [\![1, k]\!]$, $h^2(\Pi(x), \Pi(y)) \geq \frac{1}{2}\left[h^2(\Pi(x), \Pi(y_{[i \leftarrow x_i]})) + h^2(\Pi(x_{[i \leftarrow y_i]}), \Pi(y))\right]$.*

4 The function parity

We now prove a lower bound on the multi-party peer-to-peer randomized communication complexity of the k-party n-bit parity function Par_k^n, defined as follows: each player i receives n bits $(x_i^p)_{p \in [\![1, n]\!]}$ and player 1 has to output the bitwise sum modulo 2 of the inputs, i.e.,

$$\mathsf{Par}_k^n(x) = \left(\oplus_{i=1}^k x_i^1, \oplus_{i=1}^k x_i^2, \ldots, \oplus_{i=1}^k x_i^n\right)$$

(the case where all k players compute the function is trivial). To start, we prove a lower bound on the multi-party information complexity of the parity function, where each player has a single input bit. For simplicity we denote this function Par_k, rather than Par_k^1.

Theorem 4.1. *Let μ be the uniform distribution on $\{0,1\}^k$. Given any fixed $0 \leq \epsilon < \frac{1}{2}$, for any protocol π ϵ-computing Par_k, it holds that $\mathsf{MIC}_\mu(\pi) = \Omega(k)$.*

Proof.

$$
\begin{aligned}
\mathsf{MIC}_\mu(\pi) &= \sum_{i=1}^{k} \left(I(X_{-i};\Pi_i \mid X_i R_i) + I(X_i;\Pi_i \mid X_{-i}R_{-i})\right) \\
&\geq \sum_{i=2}^{k} I(X_i;\Pi_i \mid X_{-i}R_{-i}) \\
&= \sum_{i=2}^{k} (I(X_i;\Pi_i \mid X_{-i}R_{-i}) + I(X_i;\Pi_1 \mid X_{-i}R_{-i}\Pi_i)) \quad (\text{as } H(\Pi_1 \mid X_{-i}R_{-i}\Pi_i) = 0) \\
&= \sum_{i=2}^{k} I(X_i;\Pi_1\Pi_i \mid X_{-i}R_{-i}) \quad (\text{chain rule}) \\
&\geq \sum_{i=2}^{k} I(X_i;\Pi_1 \mid X_{-i}R_{-i}) \\
&= \sum_{i=2}^{k} (H(X_i \mid X_{-i}R_{-i}) - H(X_i \mid X_{-i}R_{-i}\Pi_1)) \\
&= \sum_{i=2}^{k} (1 - H(X_i \mid X_{-i}R_{-i}\Pi_1)) \quad (\text{because } X_i \text{ is uniform and independent of } X_{-i} \text{ and of } R_{-i}) \\
&\geq \sum_{i=2}^{k} (1 - H(\mathsf{Par}_k(X) \mid X_{-i}R_{-i}\Pi_1)) \quad (\text{data processing inequality, as } \exists\, \Phi \mid X_i = \Phi(\mathsf{Par}_k(X), X_{-i})) \\
&\geq \sum_{i=2}^{k} (1 - H(\mathsf{Par}_k(X) \mid X_1 R_1 \Pi_1)) \\
&\geq (k-1)(1 - H(\mathsf{Par}_k(X) \mid X_1 R_1 \Pi_1)) \\
&\geq (k-1)(1 - h(\epsilon)) \quad (\text{since player 1 outputs } \mathsf{Par}_k(X) \text{ with error } \epsilon; \text{ see Claim B.1})\ .
\end{aligned}
$$

┘

The next theorem follows immediately from Theorem 4.1 and Theorem 3.5.

Theorem 4.2. *Let μ be the uniform distribution on $\{0,1\}^k$. Given any fixed $0 \leq \epsilon < \frac{1}{2}$, for any protocol π ϵ-computing Par_k^n, it holds that $\mathsf{MIC}_{\mu^n}(\pi) = \Omega(kn)$.*

We can now prove a lower bound on the communication complexity of Par_k^n. Note that the lower bound for Par_k^n given in [31] is valid only for a restricted class of protocols, called "oblivious" in [31].

Theorem 4.3. *Given any fixed $0 \leq \epsilon < \frac{1}{2}$, there is a constant α such that for $n \geq \frac{1}{\alpha}k$,*

$$\mathsf{CC}^\epsilon(\mathsf{Par}_k^n) = \Omega(kn)\ .$$

Proof. Let π be a protocol ϵ-computing Par_k^n. By Lemma 3.4 and Theorem 4.2, there exists a constant β such that $\mathsf{CC}(\pi) \geq \beta kn - k^2$. Let $\alpha < \beta$ be a constant. For $n \geq \frac{1}{\alpha}k$, we have $k^2 \leq \alpha kn$ and we get $\mathsf{CC}(\pi) \geq (\beta - \alpha)kn = \Omega(kn)$.

┘

5 The function AND

In this section we consider an arbitrary k-party protocol, π, where each player has an input bit x_i, and where π has to compute the AND of all the input bits. We prove a lower bound on a certain information theoretic measure (that we define below) for π. The proof makes use of a certain input distribution that we will define below. In the proof we use the following notations. Denote by $\overline{1}^t$ the all-1 bit-vector of length t. Denote by $\overline{e}^t_{a_1,\ldots,a_d}$ the vector obtained from $\overline{1}^t$ by changing the bit 1 into the bit 0 at indexes $a_1, \ldots, a_d$. To simplify notations, we sometimes omit the superscript t when $t = k$, and write $\overline{e}_{a_1,\ldots,a_d}$ or $\overline{1}$. We further use in the sequel the notation $\delta_{a,b}$ for the Kronecker delta, i.e., $\delta_{a,b} = 1$ if $a = b$ and 0 otherwise.

Input distribution. Consider the distribution μ defined as follows. Draw a bit $M \sim \mathbf{Ber}(\frac{2}{3}, \frac{1}{3})$, and a uniformly random index $Z \in [\![1, k]\!]$. Assign 0 to X_Z. If $M = 0$, sample X_{-Z} uniformly in $\{0, 1\}^{k-1}$; if $M = 1$, assign 1^{k-1} to X_{-Z}. We will also work with the product distribution μ^n. Our distribution is similar to the ones of [8, 15] in that it leads to a high information cost (or similar measures) for the function AND_k. The distribution that we use has the property that the AND of any input in the support of μ is 0. This allows us to prove lower bounds for the Disjointness function without the constraint that $k = \Omega(\log n)$ which was necessary in [8] (but not in [15]).

Given a protocol π, let $\Pi_i[x_i, m, z]$ denote the distribution of Π_i, when the input X is sampled as follows: $X \sim \mu$, conditioned on the fact that $X_i = x_i$, $M = m$ and $Z = z$.

5.1 Basic properties

We first prove a number of basic properties of π, under the input distribution μ. The proofs make use of the general properties of protocols, proved in Section 3.2.

Rectangularity. We first prove the following lemma, which is an application of Lemma 3.7 to the specific case of the distribution μ that we defined above. Its proof is given in the appendix.

Lemma 5.1. *Let π be a private-coins protocol. Let $\mathcal{T}$ denote the set of possible transcripts of π, and for $i \in [\![1, k]\!]$ let $\mathcal{T}_i$ denote the set of possible transcript of by player i so that $\mathcal{T} \subseteq \mathcal{T}_1 \times \cdots \times \mathcal{T}_k$. Then there exists a function $c : \{0, 1\} \times [\![1, k]\!] \times \mathcal{T} \to [0, 1]$, and for every $i \in [\![1, k]\!]$ there is a function $c_i : \{0, 1\} \times [\![1, k]\!] \times \mathcal{T}_i \to [0, 1]$, such that $\forall\, i \in [\![1, k]\!]$, $\forall\, x' \in \{0, 1\}$, $\forall\, m \in \{0, 1\}$, $\forall\, z \in [\![1, k]\!] \setminus \{i\}$, $\forall\, \tau = (\tau_1, \ldots, \tau_k) \in \mathcal{T}$,*

$$\Pr[\Pi_i = \tau_i \mid X_i = x', M = m, Z = z] = q_i(x', \tau_i)c_i(m, z, \tau_i) ,$$

and

$$\Pr[\Pi = \tau \mid X_i = x', M = m, Z = z] = q_i(x', \tau_i)c(m, z, \tau) .$$

Diagonal lemma. The following lemma is a version of Lemma 3.9 adapted to our distribution. Its proof is given in the appendix.

Lemma 5.2. *Let π be a private-coins protocol. For any $i, j \in [\![1, k]\!]$ with $i \neq j$, we have $h^2(\Pi_i[0, 0, j], \Pi_i[1, 1, j]) \geq \frac{1}{2}h^2(\Pi_i(\overline{e}_{i,j}), \Pi_i(\overline{e}_j))$.*

Localization. The following lemma formalizes the fact that if changing the input of a player changes the transcript of the protocol, then this change necessarily appears in the partial transcript of that player. For randomized protocols this change is observed and quantified by the Hellinger distance between the distributions of the transcripts. The proof is given in the appendix.

Lemma 5.3. *Let π be a private-coins protocol. $\forall i \in [\![1,k]\!],\ \forall j \in [\![1,k]\!] \setminus \{i\}$,*

$$h(\Pi_i(\overline{e}_{i,j}), \Pi_i(\overline{e}_j)) = h(\Pi(\overline{e}_{i,j}), \Pi(\overline{e}_j)) \ .$$

5.2 Switched multi-party information cost of AND_k

We propose the following definition, which is an adaptation of the switched information cost of [8]. We call it Switched Multi-party Information Cost (SMIC).

Definition 5.4. *For a k-player protocol π with inputs drawn from μ^n let*

$$\mathsf{SMIC}_{\mu^n}(\pi) = \sum_{i=1}^{k} \left(I(X_i; \Pi_i \mid MZ) + I(M; \Pi_i \mid X_i Z) \right) \ .$$

Note that the notion of SMIC is only defined with respect to the distribution μ^n that we defined, and we may thus omit the distribution from the notation. We note that in order to simplify the expressions we often consider the public randomness as implicit in the information theoretic expressions we use below. It can be materialized either as part of the transcript or in the conditioning of the information theoretic expressions.

We can now prove the main result of this section.

Theorem 5.5. *For any fixed $0 \le \epsilon < \frac{1}{2}$, for any protocol π externally ϵ-computing AND_k,*

$$\mathsf{SMIC}_{\mu}(\pi) = \Omega(k) \ .$$

Proof. We prove below the claim for an arbitrary private-coins protocol π. The claim for general protocols (i.e., with public randomness) then follows from averaging over all possible assignments to the public randomness.

Observe that by the definition of μ, for any $i \in [\![1,k]\!]$, if $M = 0$ and $Z = z \neq i$, then $X_i \sim \mathbf{Ber}(\frac{1}{2}, \frac{1}{2})$. We therefore get by Lemma A.13 that

$$\forall i \in [\![1,k]\!],\ \forall z \in [\![1,k]\!] \setminus \{i\},\ \ I(X_i; \Pi_i \mid M = 0, Z = z) \ge h^2(\Pi_i[0,0,z], \Pi_i[1,0,z]) \ . \tag{1}$$

Similarly, by the definition of μ we have that for any $i \in [\![1,k]\!]$, if $X_i = 1$ and $Z = z \neq i$, then $M \sim \mathbf{Ber}(\frac{1}{2}, \frac{1}{2})$, and we get by Lemma A.13 that

$$\forall i \in [\![1,k]\!],\ \forall z \in [\![1,k]\!] \setminus \{i\},\ \ I(M; \Pi_i \mid X_i = 1, Z = z) \ge h^2(\Pi_i[1,0,z], \Pi_i[1,1,z]) \ . \tag{2}$$

Let us now define $\mathsf{SMIC}_i(\pi) = I(X_i; \Pi_i \mid MZ) + I(M; \Pi_i \mid X_i Z)$, so that $\mathsf{SMIC}(\pi) = \sum_{i=1}^{k} \mathsf{SMIC}_i(\pi)$.

We get

$$\begin{aligned}
\mathsf{SMIC}_i(\pi) &= I(X_i; \Pi_i \mid MZ) + I(M; \Pi_i \mid X_i Z) \\
&= \mathop{\mathbb{E}}_{z}\left[I(X_i; \Pi_i \mid M, Z = z) + I(M; \Pi_i \mid X_i, Z = z)\right] \\
&\geq \frac{1}{k} \sum_{z \neq i} \left[I(X_i; \Pi_i \mid M, Z = z) + I(M; \Pi_i \mid X_i, Z = z)\right] \\
&\geq \frac{1}{k} \sum_{z \neq i} \Big[\Pr[M = 0 \mid Z = z] I(X_i; \Pi_i \mid M = 0, Z = z) + \\
&\qquad\qquad \Pr[X_i = 1 \mid Z = z] I(M; \Pi_i \mid X_i = 1, Z = z)\Big] \; .
\end{aligned}$$

By the definition of μ, $\Pr[M = 0 \mid Z = z] = \frac{2}{3}$ for any z. Also, for any $i \neq z$,

$$\begin{aligned}
\Pr[X_i = 1 \mid Z = z] &= \Pr[M = 0 \mid Z = z] \Pr[X_i = 1 \mid M = 0, Z = z] \\
&\quad + \Pr[M = 1 \mid Z = z] \Pr[X_i = 1 \mid M = 1, Z = z] \\
&= \frac{2}{3} \cdot \frac{1}{2} + \frac{1}{3} \cdot 1 = \frac{2}{3} \, .
\end{aligned}$$

Thus, using Inequalities (1) and (2), we have

$$\begin{aligned}
\mathsf{SMIC}_i(\pi) &\geq \frac{1}{k} \sum_{z \neq i} \left[\frac{2}{3} h^2(\Pi_i[0,0,z], \Pi_i[1,0,z]) + \frac{2}{3} h^2(\Pi_i[1,0,z], \Pi_i[1,1,z])\right] \\
&\geq \frac{1}{3k} \sum_{z \neq i} \left[h(\Pi_i[0,0,z], \Pi_i[1,0,z]) + h(\Pi_i[1,0,z], \Pi_i[1,1,z])\right]^2 \\
&\geq \frac{1}{3k} \sum_{z \neq i} h^2(\Pi_i[0,0,z], \Pi_i[1,1,z]) \quad \text{(by the triangular inequality).}
\end{aligned}$$

We have

$$\begin{aligned}
\mathsf{SMIC}(\pi) &= \sum_{i=1}^{k} \mathsf{SMIC}_i(\pi) \\
&\geq \frac{1}{3k} \sum_{i,z \mid i \neq z} h^2(\Pi_i[0,0,z], \Pi_i[1,1,z]) \\
&\geq \frac{1}{3k} \sum_{\{i,z\}} [h^2(\Pi_i[0,0,z], \Pi_i[1,1,z]) + h^2(\Pi_z[0,0,i], \Pi_z[1,1,i])] \\
&\geq \frac{1}{6k} \sum_{\{i,z\}} [h^2(\Pi_i(\overline{e}_{i,z}), \Pi_i(\overline{e}_z)) + h^2(\Pi_z(\overline{e}_{i,z}), \Pi_z(\overline{e}_i))] \quad \text{(by Lemma 5.2)} \\
&\geq \frac{1}{6k} \sum_{\{i,z\}} [h^2(\Pi(\overline{e}_{i,z}), \Pi(\overline{e}_z)) + h^2(\Pi(\overline{e}_{i,z}), \Pi(\overline{e}_i))] \quad \text{(by Lemma 5.3)} \\
&\geq \frac{1}{12k} \sum_{\{i,z\}} [h(\Pi(\overline{e}_{i,z}), \Pi(\overline{e}_z)) + h(\Pi(\overline{e}_{i,z}), \Pi(\overline{e}_i))]^2 \\
&\geq \frac{1}{12k} \sum_{\{i,z\}} h^2(\Pi(\overline{e}_i), \Pi(\overline{e}_z)) \quad \text{(by the triangular inequality)} \\
&\geq \frac{1}{24k} \sum_{\{i,z\}} h^2(\Pi(\overline{e}_i), \Pi(\overline{1})) \quad \text{(by Lemma 3.9, omitting part of the right hand side term)} \\
&\geq \frac{1}{24k} \sum_{\{i,z\}} \frac{(1-2\epsilon)^2}{2} \quad \text{(by Lemma 3.8)} \\
&\geq \frac{(k-1)(1-2\epsilon)^2}{96} = \Omega(k) \, .
\end{aligned}$$

┘

6 The function Disjointness

In the k players n-bit disjointness function Disj_k^n, every player $i \in [\![1,k]\!]$ has an n-bit string $(x_i^\ell)_{\ell \in [\![1,n]\!]}$, and the players have to output 1 if and only if there exists a coordinate ℓ where all players have the bit 1. Formally, $\mathsf{Disj}_k^n(x) = \bigvee_{\ell=1}^n \bigwedge_{i=1}^k x_i^\ell$.

6.1 Switched multi-party information cost of Disj_k^n

We first prove a direct-sum-type property which allows us to make the link between the functions AND_k and Disj_k^n. A similar property was proved in [8] in the coordinator model; our peer-to-peer model requires a different, more involved, construction, since we do not have the coordinator, and moreover no player can act as the coordinator since it would get too much information. Since Disj_k^n is the disjunction of n AND_k functions, we analyze the switched multi-party information cost of Disj_k^n using the distribution μ^n.

Lemma 6.1. *Let $k > 3$. For any protocol π externally ϵ-computing Disj_k^n, there exists a protocol π' externally ϵ-computing AND_k such that*

$$\mathsf{SMIC}_{\mu^n}(\pi) \geq n \cdot \mathsf{SMIC}_\mu(\pi') \, .$$

Proof. Based on an arbitrary protocol π for Disj_k^n, we define a protocol π' for AND_k, and then analyze $\mathsf{SMIC}_{\mu^n}(\pi)$ and $\mathsf{SMIC}_\mu(\pi')$. Let $u \in \{0,1\}^k$ be the input to π' such that u_i is given to player i. We note that we cannot use a protocol similar to the one used in [8] since in the peer-to-peer setting one does not have a coordinator that can sample the inputs for the player. We thus need to sample the inputs in a distributed way, while keeping the information complexity under control using classic secret sharing techniques. The protocol π' is defined as follows.

1. The players first sample publicly an index L uniformly in $[\![1,n]\!]$, and then sample publicly Z^t, for $t \in [\![1,n]\!] \setminus \{L\}$, independently and uniformly in $[\![1,k]\!]$.
2. They then proceed to sample M^t, for $t \in [\![1,n]\!] \setminus \{L\}$, as follows. The set of players is partitioned into two subsets, $\{1,2\}$ and $\{3,\dots,k\}$. Player 1 samples $M^1 \dots M^{L-1}$ and sends the sampled values to player 2 (player 3 samples $M^{L+1} \dots M^n$, see below).
3. Then Player 1 samples $X_1^1 \dots X_1^{L-1}$ according to the distribution μ, and player 2 samples $X_2^1 \dots X_2^{L-1}$, according to the distribution μ. Observe that they can do this as they know $M^1, \dots, M^{L-1}, Z^1, \dots, Z^{L-1}$.
4. Players 1 and 2 then apply the following procedure to communicate X_j^t to player j, for $j > 2$ and $t < L$: Player 1 sends a bit p_j^t to player j, and sends a bit v_j^t to player 2. Player 2 then sends a bit q_j^t to player j. Player j then defines $X_j^t = p_j^t \oplus q_j^t$. The bits p_j^t, q_j^t and v_j^t are generated in the following way.
   - If $Z^t = j$ player 1 privately samples a random bit v_j^t. It then sets $p_j^t = v_j^t$. Player 2 sets $q_j^t = v_j^t$. Player j thus defines $X_j^t = 0$.
   - If $Z^t \neq j$ and $M^t = 0$, player 1 privately samples two independent random bits p_j^t and v_j^t. Player 2 privately samples a random bit q_j^t. The bit X_j^t defined by player j is in this case a uniform random bit. Note that it is not necessary for the correctness of the protocol that bit v_j^t is sent to Player 2 in this case; it is sent here only to make our notations simpler.
   - If $Z^t \neq j$ and $M^t = 1$, player 1 privately samples a random bit v_j^t. It then sets $p_j^t = v_j^t$. Player 2 defines $q_j^t = v_j^t \oplus 1$. Player j thus defines $X_j^t = 1$.
5. Player 3 samples $M^{L+1} \dots M^n$ and sends the sampled values to players 4 to k. Every player $i \geq 3$ privately samples $X_i^{L+1} \dots X_i^n$.
6. Players 3 and 4 then apply the same procedure as players 1 and 2, in order to communicate X_j^t to player j, for $j \leq 2$ and $t > L$. We denote by p_1^t and by p_2^t the bits sent by player 3 to player 1 and to player 2, respectively; by q_1^t and by q_2^t the bits sent by player 4 to player 1 and to player 2, respectively; and by v_1^t and by v_2^t the bits sent by player 3 to player 4.
7. Now all the players run protocol π, on the input composed of (1) the values defined above for x_i^t, $i \in [\![1,k]\!]$, $t \in [\![1,n]\!] \setminus \{L\}$, and (2) $x_i^L = u_i$, for $i \in [\![1,k]\!]$.
8. The output of the protocol π' is the output of the protocol π.

First observe that if π computes Disj_k^n with error ϵ, then π' computes AND_k with error ϵ, and this is regardless of the values of the random bits used in the construction of the input to π (this property of the distribution of the input to π is called *collapsing on coordinate* L in, e.g., [8]).

Now observe that if the input to protocol π', denote it U, is distributed according to μ (as defined above) then the definition of π' guarantees that the input to protocol π, X, is distributed according to μ^n. Using the notation we use for μ we can write that if $(U, N, S) \sim \mu$ then $(X, M, Z) \sim \mu^n$.

We now give an upper bound on $\mathsf{SMIC}_\mu(\pi')$ in terms of $\mathsf{SMIC}_{\mu^n}(\pi)$. To this end we first express the transcripts of protocol π', Π'_i, $1 \le i \le k$, in terms of the transcripts (Π_i) of the protocol π, run in Step 7.

Let us take player 2 and express Π'_2 as a function of Π_2. Taking into account the preliminary sampling procedure of protocol π', we can write Π'_2 in four parts.

1. The values which are a function of the public randomness used by π': L, Z^{-L} (for simplicity we include the sampled values and not the random bits).
2. - Read by player 2 (and sent by player 1), $M^{<L}$.
   - Read by player 2 (and sent by player 1), all the v_j^t for $j > 2, t < L$ (denoted below as $v_{>2}^{<L}$).
   - Sent by player 2, all the q_j^t, for $j > 2, t < L$ (denoted below as $q_{>2}^{<L}$).
3. Player 2 also receives $p_2^{L+1} \dots p_2^n, q_2^{L+1} \dots q_2^n$ from players 3 and 4 (denoted below as $p_2^{>L}$ and $q_2^{>L}$).
4. The last part is the transcript of player 2 when running π.

Thus, the transcript Π'_2 can be written as $LZ^{-L}M^{<L}v_{>2}^{<L}q_{>2}^{<L}p_2^{>L}q_2^{>L}\Pi_2$. However, in the manipulations of SMIC we can write Π'_2 also as $Z^{-L}M^{<L}X_2^{>L}\Pi_2$. This is because

$$\begin{aligned}
I(U_2; \Pi'_2 \mid NS) + I(N; \Pi'_2 \mid U_2 S) &= I(U_2; LZ^{-L}M^{<L}v_{>2}^{<L}q_{>2}^{<L}p_2^{>L}q_2^{>L}\Pi_2 \mid NS) + \\
&\quad I(N; LZ^{-L}M^{<L}v_{>2}^{<L}q_{>2}^{<L}p_2^{>L}q_2^{>L}\Pi_2 \mid U_2 S) \\
&= I(U_2; LZ^{-L}M^{<L}v_{>2}^{<L}q_{>2}^{<L}X_2^{>L}\Pi_2 \mid NS) + \\
&\quad I(N; LZ^{-L}M^{<L}v_{>2}^{<L}q_{>2}^{<L}X_2^{>L}\Pi_2 \mid U_2 S) \\
&= I(U_2; Z^{-L}M^{<L}X_2^{>L}\Pi_2 \mid NS) + I(U_2; Lv_{>2}^{<L}q_{>2}^{<L} \mid NSZ^{-L}M^{<L}X_2^{>L}\Pi_2) + \\
&\quad I(N; Z^{-L}M^{<L}X_2^{>L}\Pi_2 \mid U_2 S) + I(N; Lv_{>2}^{<L}q_{>2}^{<L} \mid U_2 SZ^{-L}M^{<L}X_2^{>L}\Pi_2) \\
&= I(U_2; Z^{-L}M^{<L}X_2^{>L}\Pi_2 \mid NS) + I(N; Z^{-L}M^{<L}X_2^{>L}\Pi_2 \mid U_2 S) \, ,
\end{aligned}$$

where the second equality follows from the fact that the distribution of $p_2^t q_2^t$, for all $t > L$, is uniform for $p_2^t \oplus q_2^t = x_2^t$ and independent of U_2 (resp., of N), conditioned on $X_2^{>L}$, the rest of the transcript Π'_2, and S (resp., U_2); the third equality follows from the chain rule; and the last equality follows from the fact that $I(U_2; Lv_{>2}^{<L}q_{>2}^{<L} \mid NSZ^{-L}M^{<L}X_2^{>L}\Pi_2) = 0$ and $I(N; Lv_{>2}^{<L}q_{>2}^{<L} \mid U_2 SZ^{-L}M^{<L}X_2^{>L}\Pi_2) = 0$. These last two equations follow from the fact that $Lv_{>2}^{<L}q_{>2}^{<L}$ is independent of U_2 (resp., of N), even conditioned on $SZ^{-L}M^{<L}X_2^{>L}\Pi_2$ and on N (resp., on U_2).

By similar argument we can write, in the manipulations of SMIC, Π'_1 as $Z^{-L}M^{<L}X_1^{>L}\Pi_1$, and for $i \ge 3$, Π'_i as $Z^{-L}M^{>L}X_i^{<L}\Pi_i$.

We have

$$\begin{aligned}
\mathsf{SMIC}_\mu(\pi') &= \sum_{i=1}^{k} \big(I(U_i; \Pi'_i \mid NS) + I(N; \Pi'_i \mid U_i S)\big) \\
&= \mathop{\mathbb{E}}_{\ell}\Bigg[\sum_{i=1}^{2} \Big(I(X_i^\ell; Z^{-\ell} M^{<\ell} X_i^{>\ell} \Pi_i \mid M^\ell Z^\ell) + I(M^\ell; Z^{-\ell} M^{<\ell} X_i^{>\ell} \Pi_i \mid X_i^\ell Z^\ell)\Big) \\
&\quad + \sum_{i=3}^{k} \Big(I(X_i^\ell; Z^{-\ell} M^{>\ell} X_i^{<\ell} \Pi_i \mid M^\ell Z^\ell) + I(M^\ell; Z^{-\ell} M^{>\ell} X_i^{<\ell} \Pi_i \mid X_i^\ell Z^\ell)\Big)\Bigg] \\
&= \mathop{\mathbb{E}}_{\ell}\Bigg[\sum_{i=1}^{2} \Big(I(X_i^\ell; \Pi_i \mid X_i^{>\ell} M^{\leq\ell} Z) + I(M^\ell; \Pi_i \mid X_i^{\geq\ell} M^{<\ell} Z)\Big) \\
&\quad + \sum_{i=3}^{k} \Big(I(X_i^\ell; \Pi_i \mid X_i^{<\ell} M^{\geq\ell} Z) + I(M^\ell; \Pi_i \mid X_i^{\leq\ell} M^{>\ell} Z)\Big)\Bigg] .
\end{aligned}$$

Now, applying Lemma A.9, we have that for any ℓ

$$\begin{aligned}
I(X_i^\ell; \Pi_i \mid X_i^{>\ell} M^{\leq\ell} Z) &\leq I(X_i^\ell; \Pi_i \mid X_i^{>\ell} M Z) \quad (\text{since } I(X_i^\ell; M^{>\ell} \mid X_i^{>\ell} M^{\leq\ell} Z) = 0), \\
I(M^\ell; \Pi_i \mid X_i^{\geq\ell} M^{<\ell} Z) &\leq I(M^\ell; \Pi_i \mid X_i M^{<\ell} Z) \quad (\text{since } I(M^\ell; X_i^{<\ell} \mid X_i^{\geq\ell} M^{<\ell} Z) = 0), \\
I(X_i^\ell; \Pi_i \mid X_i^{<\ell} M^{\geq\ell} Z) &\leq I(X_i^\ell; \Pi_i \mid X_i^{<\ell} M Z) \quad (\text{since } I(X_i^\ell; M^{<\ell} \mid X_i^{<\ell} M^{\geq\ell} Z) = 0), \\
I(M^\ell; \Pi_i \mid X_i^{\leq\ell} M^{>\ell} Z) &\leq I(M^\ell; \Pi_i \mid X_i M^{>\ell} Z) \quad (\text{since } I(M^\ell; X_i^{>\ell} \mid X_i^{\leq\ell} M^{>\ell} Z) = 0).
\end{aligned}$$

Thus

$$\begin{aligned}
\mathsf{SMIC}_\mu(\pi') &\le \mathop{\mathbb{E}}_{\ell}\left[\sum_{i=1}^{2}\Big(I(X_i^\ell;\Pi_i \mid X_i^{>\ell}MZ) + I(M^\ell;\Pi_i \mid X_iM^{<\ell}Z)\Big)\right.\\
&\quad\left. + \sum_{i=3}^{k}\Big(I(X_i^\ell;\Pi_i \mid X_i^{<\ell}MZ) + I(M^\ell;\Pi_i \mid X_iM^{>\ell}Z)\Big)\right]\\
&\le \frac{1}{n}\sum_{\ell=1}^{n}\left[\sum_{i=1}^{2}\Big(I(X_i^\ell;\Pi_i \mid X_i^{>\ell}MZ) + I(M^\ell;\Pi_i \mid X_iM^{<\ell}Z)\Big)\right.\\
&\quad\left. + \sum_{i=3}^{k}\Big(I(X_i^\ell;\Pi_i \mid X_i^{<\ell}MZ) + I(M^\ell;\Pi_i \mid X_iM^{>\ell}Z)\Big)\right]\\
&\le \frac{1}{n}\left[\sum_{i=1}^{2}\left(\sum_{\ell=n}^{1} I(X_i^\ell;\Pi_i \mid X_i^{>\ell}MZ) + \sum_{\ell=1}^{n} I(M^\ell;\Pi_i \mid X_iM^{<\ell}Z)\right)\right.\\
&\quad\left. + \sum_{i=3}^{k}\left(\sum_{\ell=1}^{n} I(X_i^\ell;\Pi_i \mid X_i^{<\ell}MZ) + \sum_{\ell=n}^{1} I(M^\ell;\Pi_i \mid X_iM^{>\ell}Z)\right)\right]\\
&\le \frac{1}{n}\left[\sum_{i=1}^{2}\left(I(X_i;\Pi_i \mid MZ) + I(M;\Pi_i \mid X_iZ)\right)\right.\\
&\quad\left. + \sum_{i=3}^{k}\left(I(X_i;\Pi_i \mid MZ) + I(M;\Pi_i \mid X_iZ)\right)\right]\\
&\le \frac{1}{n}\sum_{i=1}^{k}\left(I(X_i;\Pi_i \mid MZ) + I(M;\Pi_i \mid X_iZ)\right)\\
&\le \frac{1}{n}\mathsf{SMIC}_{\mu^n}(\pi)\ .
\end{aligned}$$

⌟

Coupled with the lower bound on $\mathsf{SMIC}(\pi')$ for any protocol π' that computes AND_k (Section 5), the above lemma gives us a lower bound on $\mathsf{SMIC}(\pi)$ for any protocol that computes the function Disj_k^n :

Theorem 6.2. *Let $k > 3$. Given any fixed $0 \le \epsilon < \frac{1}{2}$, for any protocol π externally ϵ-computing Disj_k^n it holds that*

$$\mathsf{SMIC}_{\mu^n}(\pi) = \Omega(kn)\ .$$

6.2 Multi-party information complexity and communication complexity of Disj_k^n

We now prove a lemma that will allow us to obtain a lower bound on the multi-party peer-to-peer communication complexity of the disjointness function.

Lemma 6.3. *For any k-player protocol π, $\mathsf{SMIC}_{\mu^n}(\pi) \le \mathsf{MIC}_{\mu^n}(\pi)$.*

Proof. We first prove that

$$\forall\, i \in [\![1,k]\!],\ \ I(M;\Pi_i \mid X_iR_iZ) \le I(X_{-i};\Pi_i \mid X_iR_i)\ .$$

$$
\begin{aligned}
I(M;\Pi_i \mid X_iR_iZ) &\le I(MX_{-i};\Pi_i \mid X_iR_iZ)\\
&= I(X_{-i};\Pi_i \mid X_iR_iZ) + I(M;\Pi_i \mid XR_iZ) \quad \text{(chain rule)}\\
&\le I(X_{-i};\Pi_i \mid X_iR_iZ) + I(M;\Pi_iR_{-i} \mid XR_iZ)\\
&= I(X_{-i};\Pi_i \mid X_iR_iZ) + I(M;R_{-i} \mid XR_iZ) + I(M;\Pi_i \mid XRZ) \quad \text{(chain rule)}\\
&= I(X_{-i};\Pi_i \mid X_iR_iZ) + I(M;\Pi_i \mid XRZ)\\
&\le I(X_{-i};\Pi_i \mid X_iR_iZ) + H(\Pi_i \mid XRZ)\\
&= I(X_{-i};\Pi_i \mid X_iR_iZ) \quad \text{(because } XR \text{ determines } \Pi_i)\\
&= H(\Pi_i \mid X_iR_iZ) - H(\Pi_i \mid XR_iZ)\\
&= H(\Pi_i \mid X_iR_iZ) - H(\Pi_i \mid XR_i)\\
&\le H(\Pi_i \mid X_iR_i) - H(\Pi_i \mid XR_i)\\
&\le I(X_{-i};\Pi_i \mid X_iR_i)\ .
\end{aligned}
$$

We now prove that

$$\forall\, i \in [\![1,k]\!],\ \ I(X_i;\Pi_i \mid MZ) \le I(X_i;\Pi_i \mid X_{-i}R_{-i}).$$

Since by the definition of μ $I(X_i;X_{-i}R_{-i} \mid MZ) = 0$, we get by Lemma A.9 that

$$I(X_i;\Pi_i \mid MZ) \le I(X_i;\Pi_i \mid X_{-i}R_{-i}MZ)\ ,$$

and

$$
\begin{aligned}
I(X_i;\Pi_i \mid X_{-i}R_{-i}MZ) &= H(\Pi_i \mid X_{-i}R_{-i}MZ) - H(\Pi_i \mid XR_{-i}MZ)\\
&= H(\Pi_i \mid X_{-i}R_{-i}MZ) - H(\Pi_i \mid XR_{-i})\\
&\le H(\Pi_i \mid X_{-i}R_{-i}) - H(\Pi_i \mid XR_{-i})\\
&= I(X_i;\Pi_i \mid X_{-i}R_{-i})\ .
\end{aligned}
$$

Thus we have

$$I(X_i;\Pi_i \mid MZ) \le I(X_i;\Pi_i \mid X_{-i}R_{-i})\ .$$

Summing over $i \in [\![1,k]\!]$ concludes the proof. ⌟

The next theorem follows immediately from Theorem 6.2 and Lemma 6.3.

Theorem 6.4. *Let $k > 3$. Given any fixed $0 \le \epsilon < \frac{1}{2}$, for any protocol π externally ϵ-computing Disj_k^n, it holds that*

$$\mathsf{MIC}_{\mu^n}(\pi) = \Omega(kn)\ .$$

We now conclude with a lower bound on the randomized communication complexity of the disjointness function.

Theorem 6.5. *Given any fixed $0 \le \epsilon < \frac{1}{2}$, there is a constant α such that for $n \ge \frac{1}{\alpha}k$,*

$$\mathsf{CC}^{\epsilon}(\mathsf{Disj}_k^n) = \Omega(kn)\ .$$

Proof. For $k = 3$ the theorem follows from the fact that $\mathsf{CC}^\epsilon(\mathsf{Disj}_3^n) \geq \mathsf{CC}^\epsilon(\mathsf{Disj}_2^n)$ (simply by letting Alice simulate internally a third player with an all-1 input), and from $\mathsf{CC}^\epsilon(\mathsf{Disj}_2^n) = \Omega(n)$ (cf. [16]).

Assume now that $k > 3$. Let π be a protocol ϵ-computing Disj_k^n. We first convert π into a protocol π' which *externally* ϵ-computes Disj_k^n. The protocol π' is defined as follows. For every bit b sent by a player in π, the same player sends in π' two bits $b.b$. In addition, in π', when player 1 stops and returns its output, it sends to player 2 the message $b.(1-b)$, where b is the output it computed.

Since in π player 1 ϵ-computes the function Disj_k^n, π' externally ϵ-computes Disj_k^n. Observe that $\mathsf{CC}(\pi') = 2 \cdot \mathsf{CC}(\pi) + 2$.

By Lemma 3.4 and Theorem 6.4, there exists a constant β such that $\mathsf{CC}(\pi') \geq \beta kn - k^2$. Let $\alpha < \beta$ be a constant. For $n \geq \frac{1}{\alpha}k$, we have $k^2 \leq \alpha kn$ and we get $\mathsf{CC}(\pi') \geq (\beta - \alpha)kn = \Omega(kn)$, and $\mathsf{CC}(\pi) = \Omega(kn)$. ⌟

We note that our tight lower bound holds also for protocols where only one player is required to output the value of the function.

7 Randomness complexity of private protocols

In this section we give a lower bound of $\Omega(n)$ on the (information theoretic private computation) randomness complexity of the function Disj_k^n, i.e., we prove that in order to privately compute Disj_k^n one needs $\Omega(n)$ random bits. The significance of this result lies in that it is the first such lower bound that grows with the size of the input, which is kn, while the output remains a single bit.

7.1 Private protocols and randomness

A protocol π is said to *privately* compute a given function if, at the end of the execution of the protocol, the players have learned nothing but the value of that function. We note that the literature devoted to private computation usually focuses on 0-error protocols, and therefore, in the rest of this section, we will restrict ourselves to the case of 0-error protocols. The definitions for the case of ϵ-error privacy are similar, and the propositions and their proofs presented in this section can be easily translated to the setting of ϵ-error randomness complexity.

Furthermore, the literature on private computation is focused on *synchronous* protocols. In what follows we therefore only consider protocols in that setting. In the synchronous setting, protocols advance according to a global round structure. At every round, each player sends a message to every other player. In addition, each player has an output tape. In order to ensure that no player is ever engaged in an infinite computation process, it is required that on any input and randomness assignment, every player eventually stops sending messages. That is, for a synchronous protocol π let $t_i(x, r)$ be the smallest integer such that if π is run on (x, r) then player i does not send any message and does not write on its output tape after round $t_i(x, r)$. If no such integer exists then $t_i(x, r) = \infty$. The requirement is that for every player i, input x, and randomness assignment r $t_i(x, r) < \infty$.

The following lemma, which is a consequence of König's lemma (cf. [32]), applies to any synchronous protocol.

Lemma 7.1. *Let π be a synchronous protocol. If for any i, x, and r, $t_i(x, r) < \infty$, then there exists an integer t_f such that for any i, x, and r, $t_i(x, r) < t_f$.*

Based on the above lemma one can transform any synchronous protocol into a protocol that always runs in a fixed number of rounds, and where all players output at the protocol's end. This is done by simply delaying the output until round t_f. Observe that such transformation does not change the transcript of the protocol or any other measure such as the number of random bits used.

We can now formally define privacy:

Definition 7.2. *A k-player protocol π computing a function f is private*[10] *if for every player $i \in [\![1, k]\!]$, for all pairs of inputs $x = (x_1, \ldots, x_k)$ and $x' = (x'_1, \ldots, x'_k)$, such that $f(x) = f(x')$ and $x_i = x'_i$, for all possible private random assignments r_i of player i, and all possible public random assignments r^p, it holds that for any transcript T*

$$\Pr[\Pi_i = T \mid R_i = r_i \; ; \; X = x \; ; \; R^p = r^p] = \Pr[\Pi_i = T \mid R_i = r_i \; ; \; X = x' \; ; \; R^p = r^p]$$

where the probability is over the randomness R_{-i}, and where Π_i is the sequence of all messages sent to player i.

It is well known that in the multi-party case, i.e., when we have $k \geq 3$ players, any function can be computed privately in the peer-to-peer model [5, 19]. Private protocols require the players to make use of their private randomness. The minimal amount of private randomness needed to design a private protocol for a given function is referred to as the *randomness complexity* of that function. While in the present paper we make use of the notion of entropy, many papers on randomness in private protocols make use of the notion of the number of random bits in order to measure "the amount of randomness used". We repeat here the definitions used in those papers.

Definition 7.3. *A communication protocol is said to be d-random if, on any run, the total number of private random bits used by all the players is at most d.*

Definition 7.4. *The randomness complexity $\mathcal{R}(f)$ of a function f is the minimal integer d such that there exists a d-random private protocol computing f.*

We will also use the following two (finer) notions which in fact make use of the notion of entropy.

Definition 7.5. *The randomness complexity of a protocol π on input distribution η is defined as*

$$\mathcal{R}_\eta(\pi) = H(\Pi \mid X R^p) \; .$$

Definition 7.6. *The randomness complexity of a function f on input distribution η is defined as*

$$\mathcal{R}_\eta(f) = \inf_{\pi \text{ private protocol computing } f} \mathcal{R}_\eta(\pi) \; .$$

Once the input and the public coins are fixed, the entropy of the transcript of a protocol comes solely from the private randomness. Thus, for any input distribution η, $\mathcal{R}_\eta(\pi)$ provides a lower bound on the entropy of the private randomness used by all the players in the protocol π. In order to relate our results (which are stated in terms of entropy) to the notions previously used in the literature on the analysis of randomness in private protocols, we use the fact that, up to constant factors, the number of (uniform) random bits necessary for the generation of a random variable with a given entropy is equal to that entropy (cf. [33]). The following lemma is then immediate.

[10] In this paper we consider only the setting of 1-privacy, which we call here for simplicity, privacy.

Lemma 7.7. *Let d be an integer. If there exists an input distribution η such that $\mathcal{R}_\eta(f) > d$, then $\mathcal{R}(f) > d$.*

This means that in order to give a lower bound on the randomness complexity of a function f, we can find an input distribution η such that the randomness complexity of a function f on η is high. Since we are interested here in characterizing the randomness used in private protocols, in the rest of this section, when we use information terms such as SMIC, we will make the randomness appear explicitly in the conditioning.

To make private protocols formally fit into our model (Section 2.1), we further technically modify them such that whenever a player does not send a message, it sends instead a special message indicating "empty message". Such protocols formally fit in our model and satisfy several additional properties. We call such protocols *proper synchronous* protocols as defined below.

Definition 7.8. *We say that a protocol as defined in Section 2.1 is proper synchronous if there is an integer t_f such that for every player i, every input x, and every random assignment r it holds that*

- *In every (local) round $t < t_f$ player i sends messages to all other players, and reads messages from all other players.*
- *Player i stops at (local) round t_f.*

Observe that the above transformation from a synchronous protocol to a proper synchronous protocol preserves privacy (if the original protocol was private), and the number of random bits used by the protocol does not change. We therefore prove below our lower bound for (private) proper synchronous protocols.

7.2 Public information cost

The notion of public information cost was introduced in [31].

Definition 7.9. *For any k-player protocol π and any input distribution η, we define the public information cost of π:*

$$\mathsf{PIC}_\eta(\pi) = \sum_{i=1}^{k} I(X_{-i}; \Pi_i R_{-i} \mid X_i R_i R^p) \; .$$

Note that this definition of $\mathsf{PIC}_\eta(\pi)$ slightly differs from the one given in [31], as the "transcript" Π_i is defined in the present paper in a different way compared to the way it is defined in [31]. However, since we work in this section in the setting of proper synchronous protocols, the two definitions of a "transcript" are completely equivalent in terms of information, and thus the definition of $\mathsf{PIC}_\eta(\pi)$ in the present paper is equivalent to the one of [31].[11]

Definition 7.10. *For any function f and any input distribution η, the zero-error public information cost of f is*

$$\mathsf{PIC}_\eta(f) = \inf_\pi \mathsf{PIC}_\eta(\pi)$$

where the infimum is taken over all protocols π which compute f with 0 error.

It was shown [31] that the public information cost can be used to prove randomness complexity lower bounds via the following theorem.

[11] In fact, they would be equivalent even if we were not restricting ourselves to proper synchronous protocols. This is because of the appearance of X_i, R_i in the conditioning.

Theorem 7.11 ([31]). *For any function of k variables f, for any input distribution η,*

$$\mathcal{R}_\eta(f) \geq \frac{\mathsf{PIC}_\eta(f) - k \cdot H_\eta(f(X))}{k} .$$

We will need the following property of the public information cost.

Theorem 7.12 ([31]). *For any function f and input distribution η,*

$$\mathsf{PIC}_\eta(f) = \inf_{\pi \text{ computing } f, \text{ using only public coins}} \mathsf{PIC}_\eta(\pi) .$$

7.3 Randomness complexity of Disjointness

We will prove that the switched multi-party information cost gives a lower bound on the public information cost. Let μ be the input distribution for the function AND_k defined in Section 5.

Theorem 7.13. *For any public-coins proper synchronous k-player protocol π, where the players have n-bits inputs X from $(X, M, Z) \sim \mu^n$, it holds that*

$$\mathsf{PIC}_{\mu^n}(\pi) \geq \frac{1}{2}\mathsf{SMIC}_{\mu^n}(\pi) .$$

We start with a number of notations. Recall that we consider a *proper synchronous* protocol π. We denote by $(\overrightarrow{T_i^{\ell}})_{\ell \geq 0}$ the sequence of messages sent by player i in the protocol π, ordered by local round of player i, and within each round ordered by the index of the recipient player. Further denote by $j(i, \ell)$ the player to which message $\overrightarrow{T_i^{l}}$ is sent. Similarly, we denote by $(\overleftarrow{T_i^{\ell}})_{\ell \geq 0}$ the sequence of messages received by player i, ordered in the same way. Observe that since π is proper synchronous there exists a function $\ell'(i, \ell)$, for $1 \leq i \leq k$, $\ell \geq 0$, such that $\overrightarrow{T_i^{\ell}}$ and $\overleftarrow{T_{j(i,\ell)}^{\ell'(i,\ell)}}$ denote the same message. Further, for any $\ell_0 \geq 0$, let $\overrightarrow{T_i^{<\ell_0}}$ be the random variable representing the so-far history, i.e., all the messages sent by player i and all the messages received by player i until player i sends message $\overrightarrow{T_i^{\ell_0}}$ (for the same local round we define the order by the identity of the player sending or receiving the message). In a similar way, define $\overleftarrow{T_i^{<\ell_0}}$ to be the random variable representing the messages sent by player i and the messages received by player i until it receives message $\overleftarrow{T_i^{\ell_0}}$. Further, we denote by $\overleftarrow{\Pi_i}$ the partial transcript of player i composed only of the incoming messages. I.e., $\overleftarrow{\Pi_i}$ is the $(k-1)$-tuple $(\Pi_{i,j}^r)_{j \neq i}$.

Before giving the actual proof of Theorem 7.13 we define two information theoretic measures, which we will use as intermediate quantities in that proof. These measures are defined only with respect to the input distribution μ^n, and thus we do not indicate the distribution in the notation of these measures.

Definition 7.14.

$$\widehat{\mathsf{IC}}(\pi) = \sum_{j=1}^{k} I(X_{-j}; \overleftarrow{\Pi_j} \mid X_j R^p M Z) .$$

Definition 7.15.

$$\widetilde{\mathsf{IC}}(\pi) = \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R^p Z) .$$

We now start the proof with two lemmas that relate the intermediate measures that we just defined to the measure PIC.

Lemma 7.16. *For any public-coins protocol* π, $\widehat{\mathsf{IC}}(\pi) \le \mathsf{PIC}_{\mu^n}(\pi)$.

Proof. For any $i \in [\![1,k]\!]$,

$$\begin{aligned}
I(X_{-i};\overleftarrow{\Pi_i} \mid X_iR^pMZ) &\le I(X_{-i};\Pi_i \mid X_iR^pMZ)\\
&\le H(\Pi_i \mid X_iR^pMZ)\\
&\le H(\Pi_i \mid X_iR^p) \quad \text{(by Proposition A.2)}\\
&= H(\Pi_i \mid X_iR^p) - H(\Pi_i \mid X_iX_{-i}R^p) \quad \text{(because } X_iX_{-i}R^p \text{ determines } \Pi_i\text{)}\\
&= I(X_{-i};\Pi_i \mid X_iR^p)\ .
\end{aligned}$$

Summing over $i \in [\![1,k]\!]$ concludes the proof. ⌟

Lemma 7.17. *For any public-coins protocol* π, $\widetilde{\mathsf{IC}}(\pi) \le \mathsf{PIC}_{\mu^n}(\pi)$.

Proof. The proof is similar to the one of Lemma 7.16. For any $i \in [\![1,k]\!]$,

$$\begin{aligned}
I(X_{-i};\Pi_i \mid X_iR^pZ) &\le H(\Pi_i \mid X_iR^pZ)\\
&\le H(\Pi_i \mid X_iR^p) \quad \text{(by Proposition A.2)}\\
&= H(\Pi_i \mid X_iR^p) - H(\Pi_i \mid X_iX_{-i}R^p) \quad \text{(because } X_iX_{-i}R^p \text{ determines } \Pi_i\text{)}\\
&= I(X_{-i};\Pi_i \mid X_iR^p)\ .
\end{aligned}$$

Summing over $i \in [\![1,k]\!]$ concludes the proof. ⌟

The next two lemmas together relate SMIC to the intermediate measures that we defined.

Lemma 7.18. *For any public-coins protocol* π, $\sum_{i=1}^{k} I(M;\Pi_i \mid X_iR^pZ) \le \widetilde{\mathsf{IC}}(\pi)$.

Proof. We prove that $\forall\, i \in [\![1,k]\!],\ \ I(M;\Pi_i \mid X_iR^pZ) \le I(X_{-i};\Pi_i \mid X_iR^pZ)$.

$$\begin{aligned}
I(M;\Pi_i \mid X_iR^pZ) &\le I(MX_{-i};\Pi_i \mid X_iR^pZ)\\
&\le I(X_{-i};\Pi_i \mid X_iR^pZ) + I(M;\Pi_i \mid XR^pZ) \quad \text{(chain rule)}\\
&\le I(X_{-i};\Pi_i \mid X_iR^pZ) + H(\Pi_i \mid XR^pZ)\\
&= I(X_{-i};\Pi_i \mid X_iR^pZ) \quad \text{(because } XR^p \text{ determines } \Pi_i\text{)}\ .
\end{aligned}$$

Summing over i concludes the proof. ⌟

The ideas behind the proof of the next lemma are similar to the ones developed in the proof of the lower bound on the randomness complexity of the Parity function in [31]. However, the distribution and the quantities involved being different, a different analysis is required here. We differ the proof of the next lemma to the appendix.

Lemma 7.19. *For any public-coins proper synchronous protocol π, $\sum_{i=1}^{k} I(X_i; \Pi_i \mid R^p M Z) \leq \widehat{\mathsf{IC}}(\pi)$.*

We can now give the actual proof of Theorem 7.13.

Proof of Theorem 7.13. By Lemma 7.18 and Lemma 7.19 we have that

$$\mathsf{SMIC}_{\mu^n}(\pi) \leq \widetilde{\mathsf{IC}}(\pi) + \widehat{\mathsf{IC}}(\pi) ,$$

and using Lemma 7.17 and Lemma 7.16 we get

$$\mathsf{SMIC}_{\mu^n}(\pi) \leq 2 \cdot \mathsf{PIC}_{\mu^n}(\pi) .$$

┘

We can now give a lower bound on the public information cost of the disjointness function.

Theorem 7.20. *Let $k > 3$. For any proper synchronous protocol π computing Disj_k^n it holds that*

$$\mathsf{PIC}_{\mu^n}(\pi) = \Omega(kn) .$$

Proof. By Theorem 7.12, we only have to consider public-coins protocols. Observe that by adding an additional round to all players, such that, say, player 1 sends to player 2 his output, and all other $k(k-1)-1$ messages are constant, we can convert π into a protocol π' *externally* computing Disj_k^n. By Theorem 6.2 and Theorem 7.13, it holds that $\mathsf{PIC}_{\mu^n}(\pi') = \Omega(kn)$. Since $\mathsf{PIC}_{\mu^n}(\pi') \leq \mathsf{PIC}_{\mu^n}(\pi) + 1$, we get that $\mathsf{PIC}_{\mu^n}(\pi) = \Omega(kn)$.

┘

Our lower bound on the randomness complexity of the disjointness function then follows.

Theorem 7.21. *Let $k > 3$. $\mathcal{R}(\mathsf{Disj}_k^n) = \Omega(n)$.*

Proof. By Theorem 7.20, $\mathsf{PIC}_{\mu^n}(\mathsf{Disj}_k^n) = \Omega(kn)$. Moreover, $H_{\mu^n}(\mathsf{Disj}_k^n) = 0$. Applying Theorem 7.11 (and Lemma 7.7), we get

$$\mathcal{R}(\mathsf{Disj}_k^n) \geq \mathcal{R}_{\mu^n}(\mathsf{Disj}_k^n) \geq \frac{\Omega(kn)}{k} = \Omega(n) .$$

┘

8 Conclusions and open problems

We introduce new models and new information theoretic tools for the study of communication complexity, and other complexity measures, in the natural peer-to-peer, multi-party, number-in-hand setting. We prove a number of properties of our new models and measures, and exemplify their effectiveness by proving two lower bounds on communication complexity, as well as a lower bound on the amount of randomness necessary for certain private computations.

To the best of our knowledge, our lower bounds on communication complexity are the first tight (non-trivial) lower bounds on communication complexity in the natural *peer-to-peer* multi-party setting, and our lower bound on the randomness complexity of private computations is the first that grows with the size of the input, while the computed function is a boolean one (i.e., the size of the output does not grow with the size of the input).

We believe that our models and tools may find additional applications and may open the way to further study of the natural peer-to-peer setting and to the building of a more solid bridge between the the fields of communication complexity and of distributed computation.

Our work raises a number of questions. First, how can one relax the restrictions that we impose on the general asynchronous model and still prove communication complexity lower bounds in a peer-to-peer setting? Our work seems to suggest that novel techniques and ideas, possibly not based on information theory, are necessary for this task, and it would be most interesting to find those. Second, it would be interesting to identify the necessary and sufficient conditions that guarantee the "rectangularity" property of communication protocols in a peer-to-peer setting. While this property is fundamental to the analysis of two-party protocols, it turns out that once one turns to the multi-party peer-to-peer setting, not only does this property become subtle to prove, but also this property does not always hold. Given the central (and sometimes implicit) role of the rectangularity property in the literature, it would be interesting to identify when it holds in the multi-party peer-to-peer number-in-hand setting.

Acknowledgments. We thank Iordanis Kerenidis and Rotem Oshman for very useful discussions.

References

[1] Gilad Asharov and Yehuda Lindell. A full proof of the BGW protocol for perfectly secure multiparty computation. *J. Cryptology*, 30(1):58–151, 2017.

[2] Reuven Bar-Yehuda, Benny Chor, Eyal Kushilevitz, and Alon Orlitsky. Privacy, additional information and communication. *IEEE Transactions on Information Theory*, 39(6):1930–1943, 1993.

[3] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *J. Comput. Syst. Sci.*, 68(4):702–732, 2004.

[4] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 67–76, New York, NY, USA, 2010. ACM.

[5] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, STOC '88, pages 1–10, New York, NY, USA, 1988. ACM.

[6] C. Blundo, A. De Santis, G. Persiano, and U. Vaccaro. Randomness complexity of private computation. *computational complexity*, 8(2):145–168, 1999.

[7] Mark Braverman. Interactive information complexity. *SIAM J. Comput.*, 44(6):1698–1739, 2015.

[8] Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A tight bound for set disjointness in the message-passing model. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 668–677. IEEE Computer Society, 2013.

[9] Mark Braverman, Ankit Garg, Denis Pankratov, and Omri Weinstein. From information to exact communication. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, STOC '13, pages 151–160, New York, NY, USA, 2013. ACM.

[10] Mark Braverman and Rotem Oshman. On information complexity in the broadcast model. In Chryssis Georgiou and Paul G. Spirakis, editors, *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, pages 355–364. ACM, 2015.

[11] Mark Braverman and Anup Rao. Information equals amortized communication. *IEEE Trans. Information Theory*, 60(10):6058–6069, 2014.

[12] Amit Chakrabarti and Sagar Kale. Strong fooling sets for multi-player communication with applications to deterministic estimation of stream statistics. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, New Brunswick, New Jersey, USA*, pages 41–50. IEEE Computer Society, 2016.

[13] Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *In IEEE Conference on Computational Complexity*, pages 107–117, 2003.

[14] Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Chi-Chih Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *FOCS*, pages 270–278, 2001.

[15] Arkadev Chattopadhyay and Sagnik Mukhopadhyay. Tribes is hard in the message passing model. In Mayr and Ollinger [39], pages 224–237.

[16] Arkadev Chattopadhyay and Toniann Pitassi. The story of set disjointness. *SIGACT News*, 41(3):59–85, September 2010.

[17] Arkadev Chattopadhyay, Jaikumar Radhakrishnan, and Atri Rudra. Topology matters in communication. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 631–640, 2014.

[18] Arkadev Chattopadhyay and Atri Rudra. The range of topological effects on communication. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 540–551. Springer, 2015.

[19] David Chaum, Claude Crépeau, and Ivan Damgard. Multiparty unconditionally secure protocols. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, STOC '88, pages 11–19, New York, NY, USA, 1988. ACM.

[20] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[21] Danny Dolev and Tomás Feder. Multiparty communication complexity. In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, pages 428–433. IEEE Computer Society, 1989.

[22] Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. Brief Announcement: Private Channel Models in Multi-party Communication Complexity. In *27th International Symposium on Distributed Computing (DISC), Jerusalem, Israel*, pages 575–576, 2013.

[23] Uri Feige, Joe Killian, and Moni Naor. A minimal model for secure computation (extended abstract). In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 554–563, New York, NY, USA, 1994. ACM.

[24] Silvio Frischknecht, Stephan Holzer, and Roger Wattenhofer. Networks cannot compute their diameter in sublinear time. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1150–1162. SIAM, 2012.

[25] Anna Gál and Parikshit Gopalan. Lower bounds on streaming algorithms for approximating the length of the longest increasing subsequence. *SIAM J. Comput.*, 39(8):3463–3479, 2010.

[26] Anna Gál and Adi Rosén. Omega(log n) lower bounds on the amount of randomness in 2-private computation. *SIAM J. Comput.*, 34(4):946–959, 2005.

[27] Andre Gronemeier. Asymptotically optimal lower bounds on the nih-multi-party information complexity of the and-function and disjointness. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*, volume 3 of *LIPIcs*, pages 505–516. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.

[28] Zengfeng Huang, Bozidar Radunovic, Milan Vojnovic, and Qin Zhang. Communication complexity of approximate matching in distributed graphs. In Mayr and Ollinger [39], pages 460–473.

[29] T. S. Jayram. Hellinger strikes back: A note on the multi-party information complexity of and. In *Proceedings of the 12th International Workshop and 13th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, APPROX '09 / RANDOM '09, pages 562–573, Berlin, Heidelberg, 2009. Springer-Verlag.

[30] Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM J. Discret. Math.*, 5(4):545–557, November 1992.

[31] Iordanis Kerenidis, Adi Rosén, and Florent Urrutia. Multi-party protocols, information complexity and privacy. In Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier, editors, *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland*, volume 58 of *LIPIcs*, pages 57:1–57:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[32] Stephen Cole Kleene. *Mathematical Logic*. Dover, 2002. Reprint of the John Wiley & Sons, Inc., New York, 1967 edition.

[33] Donald E. Knuth and Andrew C. Yao. The complexity of nonuniform random number generation. pages 375–428, 1976.

[34] Janne H. Korhonen and Jukka Suomela. Brief announcement: Towards a complexity theory for the congested clique. In Andréa W. Richa, editor, *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria*, volume 91 of *LIPIcs*, pages 55:1–55:3. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[35] Janne H. Korhonen and Jukka Suomela. Towards a complexity theory for the congested clique. *CoRR*, abs/1705.03284, 2017.

[36] Eyal Kushilevitz and Yishay Mansour. Randomness in private computations. *SIAM J. Discrete Math.*, 10(4):647–661, 1997.

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

[38] Eyal Kushilevitz, Rafail Ostrovsky, and Adi Rosén. Characterizing linear size circuits in terms of pricacy. *J. Comput. Syst. Sci.*, 58(1):129–136, 1999.

[39] Ernst W. Mayr and Nicolas Ollinger, editors. *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, volume 30 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[40] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *J. Comput. Syst. Sci.*, 57(1):37–49, 1998.

[41] Jeff M. Phillips, Elad Verbin, and Qin Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. *SIAM J. Comput.*, 45(1):174–196, 2016.

[42] A. A. Razborov. On the distributional complexity of disjointness. *Theor. Comput. Sci.*, 106(2):385–390, December 1992.

[43] Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. Distributed verification and hardness of distributed approximation. *CoRR*, abs/1011.3049, 2010.

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

[45] David P. Woodruff and Qin Zhang. Tight bounds for distributed functional monitoring. *CoRR*, abs/1112.5153, 2011.

[46] David P. Woodruff and Qin Zhang. When distributed computation does not help. *CoRR*, abs/1304.4636, 2013.

[47] David P. Woodruff and Qin Zhang. An optimal lower bound for distinct elements in the message passing model. In *Proceedings of the Twenty-fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 718–733, Philadelphia, PA, USA, 2014. Society for Industrial and Applied Mathematics.

[48] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 160–164. IEEE Computer Society, 1982.

A Background in Information Theory

We give a reminder on basic information theory tools that are of use in the present paper. A good reference is the book of Cover and Thomas [20]. We always consider a probability space over a discrete domain.

A.1 Entropy and mutual information

Definition A.1. *The entropy*[12] *of a random variable X is*

$$H(X) = \sum_x \Pr[X = x] \log\left(\frac{1}{\Pr[X = x]}\right) .$$

We further use the notation

$$H(X \mid Y = y) = \sum_x \Pr[X = x \mid Y = y] \log\left(\frac{1}{\Pr[X = x \mid Y = y]}\right) .$$

The conditional entropy $H(X \mid Y)$ is defined as $\mathop{\mathbb{E}}_y[H(X \mid Y = y)]$.

Proposition A.2. *For any random variables X and Y, $H(X \mid Y) \le H(X)$.*

The entropy of a random variable is always non-negative.

Theorem A.3 (Shannon)**.** *For all prefix-free finite set $\mathcal{X} \subseteq \{0,1\}^*$ and all random variable X with support supp$(X) \subseteq \mathcal{X}$, it holds*

$$H(X) \le \mathbb{E}[|X|] .$$

Definition A.4. *The mutual information between two random variables X, Y is*

$$I(X;Y) = H(X) - H(X \mid Y) .$$

The mutual information of X and Y conditioned on Z is

$$I(X;Y \mid Z) = H(X \mid Z) - H(X \mid YZ) .$$

The mutual information measures the change in the entropy of X when one learns the value of Y. It is symmetric, and non-negative.

Proposition A.5. *For any random variables X, Y and Z, $I(X;Y \mid Z) = 0$ if and only if X and Y are independent conditioned on every possible value of Z.*

[12] In this paper we refer to the binary entropy by simply saying "entropy".

We will use extensively the following proposition, known under the name of *chain rule*.

Proposition A.6. *For any random variables A, B, C, D,*

$$I(AB; C \mid D) = I(A; C \mid D) + I(B; C \mid DA) .$$

The *data processing inequality* expresses the fact that information can only be lost when applying a function to a random variable.

Proposition A.7. *For any random variables X, Y, Z, and any function f*

$$I(X; f(Y) \mid Z) \leq I(X; Y \mid Z) .$$

We will occasionally make use of the two following lemmas, which allow to add or remove a random variable from the conditioning.

Lemma A.8 ([7])**.** *For any random variables A, B, C, D such that $I(B; D \mid AC) = 0$,*

$$I(A; B \mid C) \geq I(A; B \mid CD) .$$

Lemma A.9 ([7])**.** *For any random variables A, B, C, D such that $I(B; D \mid C) = 0$,*

$$I(A; B \mid C) \leq I(A; B \mid CD) .$$

We further give a lemma which is an certain extension of the data processing inequality, allowing the processing to depend also on part of the conditioning.

Lemma A.10. *Let A, B, C, D, $\phi = \varphi(C, B)$ be random variables. Then,*

$$I(A; \phi \mid CD) \leq I(A; B \mid CD) .$$

Proof.

$$\begin{aligned}
I(A; \phi \mid CD) &= I(A; \varphi(C, B) \mid CD) \\
&= \mathop{\mathbb{E}}_{c}[I(A; \varphi(c, B) \mid C = c, D)] \\
&\leq \mathop{\mathbb{E}}_{c}[I(A; B \mid C = c, D)] \quad \text{(by the data processing inequality, Proposition A.7)} \\
&\leq I(A; B \mid CD).
\end{aligned}$$

⌟

A.2 Hellinger distance

We will make an extensive use of the Hellinger distance.

Definition A.11. *Let P and Q be two distributions over a domain Ω. The Hellinger distance between P and Q is $h(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{\omega \in \Omega} \mid \sqrt{P(\omega)} - \sqrt{Q(\omega)} \mid^2}$.*

It can be easily checked that the Hellinger distance is indeed a "distance". When using the square of the Hellinger distance, we often use the following identity.

Proposition A.12. *Let P and Q be two distributions over a domain Ω.*

$$h^2(P,Q) = 1 - \sum_{\omega\in\Omega} \sqrt{P(\omega)Q(\omega)}\ .$$

Hellinger distance can be related to mutual information by the following relation.

Lemma A.13 ([3]). *Let η_0, η_1 be two distributions over the same domain, and suppose that Y is generated as follows: first select S uniformly in $\{0,1\}$, and then sample Y according to η_S. Then $I(S;Y) \geq [h(\eta_0,\eta_1)]^2$.*

Another useful measure is the statistical distance.

Definition A.14. *Let P and Q be two distributions over a domain Ω. The statistical distance between P and Q is*

$$\Delta(P,Q) = \max_{\Omega'\subseteq\Omega} \mid P(\Omega') - Q(\Omega') \mid\ .$$

Hellinger distance and statistical distance are related by the following relation.

Lemma A.15. *Let P and Q be two distributions over the same domain. $h(P,Q) \geq \frac{1}{\sqrt{2}}\Delta(P,Q)$.*

B A technical lemma

Claim B.1. *Let π be a protocol, let i be a given player, and let $0 \leq \epsilon \leq \frac{1}{2}$ be fixed. If, when running π, player i ϵ-computes a boolean function f, then $H(f(X) \mid X_i R_i R^p \Pi_i) \leq h(\epsilon)$, where h is the binary entropy function.*

Proof. Let θ be the (deterministic) function that takes as parameter (x_i, r_i, r^p, π_i) and returns the output of player i. Define the random variable $P = \theta(X_i, R^p, R_i, \Pi_i)$, and the random variable $M = 1 - \delta_{f(X),P}$, i.e., the indicator variable of the event $f(X) \neq P$. Observe that

$$\begin{aligned}
\Pr(M=1) &= \mathbb{E}[M] \\
&= \sum_x \Pr(X=x)\,\mathbb{E}[M \mid X=x] \\
&= \sum_x \Pr(X=x)\Pr(M=1 \mid X=x) \\
&\leq \sum_x \Pr(X=x)\cdot\epsilon \quad \text{(since player } i \ \epsilon\text{-computes } f) \\
&\leq \epsilon\ .
\end{aligned}$$

Thus we have

$$\begin{aligned}
H(f(X) \mid X_i R_i R^p \Pi_i) &\leq H(f(X) \mid P) \quad \text{(data processing inequality)} \\
&= H(M \mid P) \quad \text{(since, given } P\text{, there is a bijection between } f(X) \text{ and } M) \\
&\leq H(M) \\
&= h(\Pr(M=1)) \quad \text{(since M is binary)} \\
&\leq h(\epsilon) \quad \text{(since } h \text{ is increasing in } [0,1/2]).
\end{aligned}$$

⌟

C Some of the proofs

This section contains the proofs that were deferred to the appendix.

Proof of Lemma 3.7. We prove the claim for an arbitrary player $i \in [\![1,k]\!]$. To prove the statement of the lemma define, for $x'_i \in \mathcal{X}_i$,

$$q_i(x'_i, \overline{\tau}) = \Pr[(x'_i, R_i) \in \mathcal{I}_i(\overline{\tau})] \ ,$$

and for $x'_{-i} \in \mathcal{X}_{-i}$,

$$q_{-i}(x'_{-i}, \overline{\tau}) = \Pr[(x'_{-i}, R_{-i}) \in \mathcal{J}_i(\overline{\tau})] \ .$$

We have, for $x \in \mathcal{X}$,

$$\begin{aligned}
\Pr[\Pi_i(x) = \overline{\tau}] &= \Pr[(x, R) \in \mathcal{A}_i(\overline{\tau})] \\
&= \Pr[(x_i, R_i) \in \mathcal{I}_i(\overline{\tau}) \ \&\ (x_{-i}, R_{-i}) \in \mathcal{J}_i(\overline{\tau})] \quad \text{by Lemma 3.6} \\
&= \Pr[(x_i, R_i) \in \mathcal{I}_i(\overline{\tau})] \times \Pr[(x_{-i}, R_{-i}) \in \mathcal{J}_i(\overline{\tau})] \\
&= q_i(x_i, \overline{\tau}) q_{-i}(x_{-i}, \overline{\tau}) \ .
\end{aligned}$$

We now prove the second claim. Define, for $x'_{-i}, \in \mathcal{X}_{-i}$,

$$p_{-i}(x'_{-i}, \tau) = \Pr[(x'_{-i}, R_{-i}) \in \mathcal{H}_i(\tau)] \ .$$

We have

$$\begin{aligned}
\Pr[\Pi(x) = \tau] &= \Pr[(x, R) \in \mathcal{B}(\tau))] \\
&= \Pr[(x_i, R_i) \in \mathcal{I}_i(\tau_i) \ \&\ (x_{-i}, R_{-i}) \in \mathcal{H}_i(\tau))] \quad \text{by Lemma 3.6} \\
&= \Pr[(x_i, R_i) \in \mathcal{I}_i(\tau_i)] \times \Pr[(x_{-i}, R_{-i}) \in \mathcal{H}_i(\tau))] \\
&= q_i(x_i, \tau_i) p_{-i}(x_{-i}, \tau) \ .
\end{aligned}$$

┘

Proof of Lemma 3.8. By Lemma A.15, we only need to show that $\Delta(\Pi(x), \Pi(y)) \geq 1 - 2\epsilon$. By definition, there exists a function θ taking as input the possible transcripts of π and verifying $\forall\, x \in \mathcal{X}, \Pr[\theta(\Pi(x)) = f(x)] \geq 1 - \epsilon$.
Let $\Omega' = \theta^{-1}(f(x))$. We have $\Pr[\Pi(x) \in \Omega'] = \Pr[\theta(\Pi(x)) = f(x)] \geq 1 - \epsilon$ and

$$\begin{aligned}
\Pr[\Pi(y) \in \Omega'] &= \Pr[\Pi(y) \in \theta^{-1}(f(x))] \\
&\leq \Pr[\Pi(y) \notin \theta^{-1}(f(y))] \quad \text{since } \theta^{-1}(f(x)) \cap \theta^{-1}(f(y)) = \varnothing \\
&\leq 1 - \Pr[\Pi(y) \in \theta^{-1}(f(y))] \\
&\leq 1 - (1 - \epsilon) = \epsilon.
\end{aligned}$$

Thus

$$\begin{aligned}
\Delta(\Pi(x), \Pi(y)) &\geq \Pr[\Pi(x) \in \Omega'] - \Pr[\Pi(y) \in \Omega'] \\
&\geq (1 - \epsilon) - \epsilon = 1 - 2\epsilon.
\end{aligned}$$

┘

Proof of Lemma 3.9. Let $\mathcal{T}$ be the set of all possible transcripts of π. In what follows we simplify notation and write $\sum_{\tau}$ instead of $\sum_{\tau \in \mathcal{T}}$. Using Proposition A.12,

$$\begin{aligned}
1 - h^2(\Pi(x), \Pi(y)) &= \sum_{\tau} \sqrt{\Pr[\Pi(x) = \tau] \Pr[\Pi(y) = \tau]} \\
&= \sum_{\tau} \sqrt{q_i(x_i, \tau_i) p_{-i}(x_{-i}, \tau) q_i(y_i, \tau_i) p_{-i}(y_{-i}, \tau)} \quad \text{By Lemma 3.7} \\
&= \sum_{\tau} \sqrt{q_i(x_i, \tau_i) q_i(y_i, \tau_i)} \sqrt{p_{-i}(x_{-i}, \tau) p_{-i}(y_{-i}, \tau)} \\
&\leq \sum_{\tau} \frac{q_i(x_i, \tau_i) + q_i(y_i, \tau_i)}{2} \sqrt{p_{-i}(x_{-i}, \tau) p_{-i}(y_{-i}, \tau)} \\
&\leq \frac{1}{2} \Bigg(\sum_{\tau} \sqrt{q_i(x_i, \tau_i) p_{-i}(x_{-i}, \tau) q_i(x_i, \tau_i) p_{-i}(y_{-i}, \tau)} \\
&\quad + \sum_{\tau} \sqrt{q_i(y_i, \tau_i) p_{-i}(x_{-i}, \tau) q_i(y_i, \tau_i) p_{-i}(y_{-i}, \tau)} \Bigg) \\
&\leq \frac{1}{2} \Bigg(\sum_{\tau} \sqrt{\Pr[\Pi(x) = \tau] \Pr[\Pi(y_{[i \leftarrow x_i]}) = \tau]} \\
&\qquad + \sum_{\tau} \sqrt{\Pr[\Pi(x_{[i \leftarrow y_i]}) = \tau] \Pr[\Pi(y) = \tau]} \Bigg) \\
&\leq \frac{1}{2} \left[1 - h^2(\Pi(x), \Pi(y_{[i \leftarrow x_i]})) + 1 - h^2(\Pi(x_{[i \leftarrow y_i]}), \Pi(y)) \right] \\
&\leq 1 - \frac{1}{2} \left[h^2(\Pi(x), \Pi(y_{[i \leftarrow x_i]})) + h^2(\Pi(x_{[i \leftarrow y_i]}), \Pi(y)) \right].
\end{aligned}$$

⌟

Proof of Lemma 5.1. We write $\Pr[\Pi_i = \tau_i \mid X_i = x', M = m, Z = z]$ as

$$\sum_{x \in \{0,1\}^k} \left(\Pr[X = x \mid X_i = x', M = m, Z = z] \times \Pr[\Pi_i = \tau_i \mid X = x, X_i = x', M = m, Z = z] \right).$$

Note that

$$\Pr[X = x \mid X_i = x', M = m, Z = z] = \delta_{x_i, x'} \Pr[X_{-i} = x_{-i} \mid M = m, Z = z],$$

since, conditioned on $M = m, Z = z$, X_i and X_{-i} are independent. Further note that for x such that $x_i = x'$,

$$\Pr[\Pi_i = \tau_i \mid X = x, X_i = x', M = m, Z = z] = \Pr[\Pi_i(x) = \tau_i].$$

By Lemma 3.7, there exist functions q_i and q_{-i} such that

$$\forall\, x \in \{0,1\}^k, \ \Pr[\Pi_i(x) = \tau_i] = q_i(x_i, \tau_i) q_{-i}(x_{-i}, \tau_i).$$

Therefore we can write

$$\begin{aligned}\Pr[\Pi_i = \tau_i \mid X_i = x, M = m, Z = z] = &\sum_{x \in \{0,1\}^k} (\delta_{x_i,x'} q_i(x_i,\tau_i) q_{-i}(x_{-i},\tau_i)\times \\ &\qquad \Pr[X_{-i} = x_{-i} \mid M = m, Z = z]) \\ \\ &= q_i(x',\tau_i) \sum_{\hat{x} \in \{0,1\}^{k-1}} (q_{-i}(\hat{x},\tau_i)\times \\ &\qquad \Pr[X_{-i} = \hat{x} \mid M = m, Z = z]) \\ \\ &= q_i(x',\tau_i) c_i(m,z,\tau_i)\end{aligned}$$

where $c_i(m,z,\tau_i) = \sum_{\hat{x} \in \{0,1\}^{k-1}} q_{-i}(\hat{x},\tau_i) \Pr[X_{-i} = \hat{x} \mid M = m, Z = z]$.

The proof of the second statement is similar:

$$\begin{aligned}\Pr[\Pi = \tau \mid X_i = x, M = m, Z = z] = &\sum_{x \in \{0,1\}^k} (\Pr[X = x \mid X_i = x', M = m, Z = z]\times \\ &\Pr[\Pi = \tau \mid X = x, X_i = x', M = m, Z = z]).\end{aligned}$$

Note that

$$\Pr[X = x \mid X_i = x', M = m, Z = z] = \delta_{x_i,x'} \Pr[X_{-i} = x_{-i} \mid M = m, Z = z] \ ,$$

since, conditioned on $M = m, Z = z,$, X_i and X_{-i} are independent. Further note that for x such that $x_i = x'$,

$$\Pr[\Pi = \tau \mid X = x, X_i = x', M = m, Z = z] = \Pr[\Pi(x) = \tau] \ .$$

By Lemma 3.7, there exist functions q_i and p_{-i} such that

$$\forall\, x \in \{0,1\}^k,\ \ \Pr[\Pi(x) = \tau] = q_i(x_i,\tau_i) p_{-i}(x_{-i},\tau) \ .$$

Therefore we can write

$$\begin{aligned}\Pr[\Pi = \tau \mid X_i = x', M = m, Z = z] = &\sum_{x \in \{0,1\}^k} \big(\delta_{x_i,x'} q_i(x_i,\tau_i) p_{-i}(x_{-i},\tau)\times \\ &\qquad \Pr[X_{-i} = x_{-i} \mid M = m, Z = z]) \\ \\ &= q_i(x',\tau_i) \sum_{\hat{x} \in \{0,1\}^{k-1}} (p_{-i}(\hat{x},\tau)\times \\ &\qquad \Pr[X_{-i} = \hat{x} \mid M = m, Z = z]) \\ \\ &= q_i(x',\tau_i) c(m,z,\tau)\end{aligned}$$

where $c(m,z,\tau) = \sum_{\hat{x} \in \{0,1\}^{k-1}} p_{-i}(\hat{x},\tau) \Pr[X_{-i} = \hat{x} \mid M = m, Z = z]$.

┘

Proof of Lemma 5.2. Using Lemma 5.1, we write
$\Pr[\Pi_i[0,0,j] = \overline{\tau}] = q_i(0,\overline{\tau})c_i(0,j,\overline{\tau})$ and $\Pr[\Pi_i[1,1,j] = \overline{\tau}] = q_i(1,\overline{\tau})c_i(1,j,\overline{\tau})$.

Using Lemma 3.7, we write
$\Pr[\Pi_i(\overline{e}^k_{i,j}) = \overline{\tau}] = q_i(0,\overline{\tau})q_{-i}(\overline{e}^{k-1}_j,\overline{\tau})$ and $\Pr[\Pi_i(\overline{e}^k_j) = \overline{\tau}] = q_i(1,\overline{\tau})q_{-i}(\overline{e}^{k-1}_j,\overline{\tau})$.

Note that $\Pi_i[1,1,j] = \Pi_i(\overline{e}^k_j)$, and thus

$$q_i(1,\overline{\tau}) \neq 0 \Rightarrow c_i(1,j,\overline{\tau}) = q_{-i}(\overline{e}^{k-1}_j,\overline{\tau}) \; .$$

By Proposition A.12,

$$
\begin{aligned}
1 - h^2(\Pi_i[0,0,j], \Pi_i[1,1,j]) &= \sum_{\overline{\tau}} \sqrt{\Pr[\Pi_i[0,0,j] = \overline{\tau}] \Pr[\Pi_i[1,1,j] = \overline{\tau}]} \\
&= \sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau}) c_i(0,j,\overline{\tau}) q_i(1,\overline{\tau}) c_i(1,j,\overline{\tau})} \\
&\leq \sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau}) q_i(1,\overline{\tau})} \left(\frac{c_i(0,j,\overline{\tau}) + c_i(1,j,\overline{\tau})}{2} \right) \\
&\leq \frac{1}{2} \left(\sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau}) c_i(0,j,\overline{\tau}) q_i(1,\overline{\tau}) c_i(0,j,\overline{\tau})} \right. + \\
&\qquad \left. \sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau}) c_i(1,j,\overline{\tau}) q_i(1,\overline{\tau}) c_i(1,j,\overline{\tau})} \right) \\
&\leq \frac{1}{2} \left(\sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau}) c_i(0,j,\overline{\tau}) q_i(1,\overline{\tau}) c_i(0,j,\overline{\tau})} \right. + \\
&\qquad \left. \sum_{\overline{\tau} \mid q_i(1,\overline{\tau}) \neq 0} \sqrt{q_i(0,\overline{\tau}) c_i(1,j,\overline{\tau}) q_i(1,\overline{\tau}) c_i(1,j,\overline{\tau})} \right) \\
&\leq \frac{1}{2} \left(\sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau}) c_i(0,j,\overline{\tau}) q_i(1,\overline{\tau}) c_i(0,j,\overline{\tau})} \right. + \\
&\qquad \left. \sum_{\overline{\tau} \mid q_i(1,\overline{\tau}) \neq 0} \sqrt{q_i(0,\overline{\tau}) q_{-i}(\overline{e}_j^{k-1},\overline{\tau}) q_i(1,\overline{\tau}) q_{-i}(\overline{e}_j^{k-1},\overline{\tau})} \right) \\
&\leq \frac{1}{2} \left(\sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau}) c_i(0,j,\overline{\tau}) q_i(1,\overline{\tau}) c_i(0,j,\overline{\tau})} \right. + \\
&\qquad \left. \sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau}) q_{-i}(\overline{e}_j^{k-1},\overline{\tau}) q_i(1,\overline{\tau}) q_{-i}(\overline{e}_j^{k-1},\overline{\tau})} \right) \\
&\leq \frac{1}{2} \left(\sum_{\overline{\tau}} \sqrt{\Pr[\Pi_i[0,0,j] = \overline{\tau}] \Pr[\Pi_i[1,0,j] = \overline{\tau}]} \right. + \\
&\qquad \left. \sum_{\overline{\tau}} \sqrt{\Pr[\Pi_i(\overline{e}_{i,j}^k) = \overline{\tau}] \Pr[\Pi_i(\overline{e}_j^k) = \overline{\tau}]} \right) \\
&\leq \frac{1}{2}(1 - h^2(\Pi_i[0,0,j], \Pi_i[1,0,j]) + 1 - h^2(\Pi_i(\overline{e}_{i,j}^k), \Pi_i(\overline{e}_j^k))) \\
&\leq 1 - h^2(\Pi_i(\overline{e}_{i,j}^k), (\Pi_i(\overline{e}_j^k)).
\end{aligned}
$$

⌟

Proof of Lemma 5.3. Using Lemma 3.7, we write

$$\Pr[\Pi_i(\overline{e}_{i,j}^k) = \overline{\tau}] = q_i(0,\overline{\tau}) q_{-i}(\overline{e}_j^{k-1},\overline{\tau})$$

and

$$\Pr[\Pi(\overline{e}^k_{i,j}) = \tau] = q_i(0,\tau_i)p_{-i}(\overline{e}^{k-1}_j,\tau)\ .$$

As $\Pr[\Pi_i(\overline{e}^k_{i,j}) = \overline{\tau}] = \sum_{\tau|\tau_i=\overline{\tau}} \Pr[\Pi(\overline{e}^k_{i,j}) = \tau]$ we have

$$q_i(0,\overline{\tau})q_{-i}(\overline{e}^{k-1}_j,\overline{\tau}) = \sum_{\tau|\tau_i=\overline{\tau}} q_i(0,\tau_i)p_{-i}(\overline{e}^{k-1}_j,\tau) = q_i(0,\overline{\tau}) \sum_{\tau|\tau_i=\overline{\tau}} p_{-i}(\overline{e}^{k-1}_j,\tau)\ ,$$

and thus

$$q_i(0,\overline{\tau}) \neq 0 \Rightarrow q_{-i}(\overline{e}^{k-1}_j,\overline{\tau}) = \sum_{\tau|\tau_i=\overline{\tau}} p_{-i}(\overline{e}^{k-1}_j,\tau)\ .$$

Using Proposition A.12, we can write

$$\begin{aligned}
1 - h^2(\Pi_i(\overline{e}^k_{i,j}), \Pi_i(\overline{e}^k_j)) &= \sum_{\overline{\tau}} \sqrt{\Pr[\Pi_i(\overline{e}^k_{i,j}) = \overline{\tau}]\Pr[\Pi_i(\overline{e}^k_j) = \overline{\tau}]} \\
&= \sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau})q_{-i}(\overline{e}^{k-1}_j,\overline{\tau})q_i(1,\overline{\tau})q_{-i}(\overline{e}^{k-1}_j,\overline{\tau})} \\
&= \sum_{\overline{\tau}} \sqrt{q_i(0,\overline{\tau})q_i(1,\overline{\tau})}q_{-i}(\overline{e}^{k-1}_j,\overline{\tau}) \\
&= \sum_{\overline{\tau}|q_i(0,\overline{\tau})\neq 0} \sqrt{q_i(0,\overline{\tau})q_i(1,\overline{\tau})}q_{-i}(\overline{e}^{k-1}_j,\overline{\tau}) \\
&= \sum_{\overline{\tau}|q_i(0,\overline{\tau})\neq 0} \left(\sqrt{q_i(0,\overline{\tau})q_i(1,\overline{\tau})} \sum_{\tau|\tau_i=\overline{\tau}} p_{-i}(\overline{e}^{k-1}_j,\tau)\right) \\
&= \sum_{\overline{\tau}} \left(\sqrt{q_i(0,\overline{\tau})q_i(1,\overline{\tau})} \sum_{\tau|\tau_i=\overline{\tau}} p_{-i}(\overline{e}^{k-1}_j,\tau)\right) \\
&= \sum_{\tau} \left(\sqrt{q_i(0,\tau_i)q_i(1,\tau_i)}p_{-i}(\overline{e}^{k-1}_j,\tau)\right) \\
&= \sum_{\tau} \sqrt{\Pr[\Pi(\overline{e}^k_{i,j}) = \tau]\Pr[\Pi(\overline{e}^k_j) = \tau]} \\
&= 1 - h^2(\Pi(\overline{e}^k_{i,j}), \Pi(\overline{e}^k_j))\ .
\end{aligned}$$

⌟

Proof of Lemma 7.19. For the purpose of the proof we define a certain order on the messages in Π_i, i.e., on all messages sent and received by player i as follows. We order the messages of Π_i by (local) rounds of player i, and inside each round have first the messages sent by player i, ordered by the index of the recipient, and have then the messages received by player i, ordered by the index of the sender. For a given player i, we denote the sequence thus defined as $(B^d)_{d\geq 0}$.

Now, by the chain rule, applied on the messages of Π_i by the order we just defined, and after rearranging the summands, we have

$$I(X_i;\Pi_i \mid R^pMZ) = \sum_{\ell} I(X_i;\overrightarrow{T_i^{\ell}} \mid \overrightarrow{T_i^{<\ell}}R^pMZ) + \sum_{\ell} I(X_i;\overleftarrow{T_i^{\ell}} \mid \overleftarrow{T_i^{<\ell}}R^pMZ)\ .$$

We now show that every summand of the second sum equals 0.
To this end, we now show by induction on the index d that $\forall d,\ I(X_i; X_{-i} \mid MZR^pB^0 \dots B^d) = 0$. We have $I(X_i; X_{-i} \mid MZR^p) = 0$, because according to μ^n, conditioned on MZ, X_i and X_{-i} are independent. Assume now the induction hypothesis that for some d, $I(X_i; X_{-i} \mid MZR^pB^0 \dots B^d) = 0$. If the message B^{d+1} is *sent* by player i, then B^{d+1} is a function of X_i, R^p and $B^0 \dots B^d$ and thus

$$\begin{aligned} I(X_i; X_{-i} \mid MZR^pB^0 \dots B^{d+1}) &= H(X_{-i} \mid MZR^pB^0 \dots B^{d+1}) - \\ &\quad H(X_{-i} \mid MZR^pB^0 \dots B^{d+1}X_i) \\ &\le H(X_{-i} \mid MZR^pB^0 \dots B^d) - \\ &\quad H(X_{-i} \mid MZR^pB^0 \dots B^dX_i) \\ &= I(X_i; X_{-i} \mid MZR^pB^0 \dots B^d) \\ &= 0 \ . \end{aligned}$$

Similarly, if the message B^{d+1} is *received* by player i, then B^{d+1} is a function of X_{-i}, R^p and $B^0 \dots B^d$ and thus

$$\begin{aligned} I(X_i; X_{-i} \mid MZR^pB^0 \dots B^{d+1}) &= H(X_i \mid MZR^pB^0 \dots B^{d+1}) - \\ &\quad H(X_i \mid MZR^pB^0 \dots B^{d+1}X_{-i}) \\ &\le H(X_i \mid MZR^pB^0 \dots B^d) - \\ &\quad H(X_i \mid MZR^pB^0 \dots B^dX_{-i}) \\ &= I(X_i; X_{-i} \mid MZR^pB^0 \dots B^d) \\ &= 0 \ . \end{aligned}$$

Thus we have that

$$\forall d,\ \ I(X_i; X_{-i} \mid MZR^pB^0 \dots B^d) = 0 \ . \tag{3}$$

From Eq. (3), by choosing the relevant d for any given ℓ, we can also write for all ℓ $I(X_i; X_{-i} \mid T_i^{\overleftarrow{<\ell}} R^pMZ) = 0$. Applying Lemma A.10 with $A = X_i$, $B = X_{-i}$, $C = (T_i^{\overleftarrow{<\ell}}, R^p)$, $D = (M, Z)$ and $\phi = T_i^{\overleftarrow{\ell}} = \varphi(T_i^{\overleftarrow{<\ell}}, R^p, X_{-i})$ yields $I(X_i; T_i^{\overleftarrow{\ell}} \mid T_i^{\overleftarrow{<\ell}} R^pMZ) = 0$. We have thus shown that

$$I(X_i; \Pi_i \mid R^pMZ) = \sum_{\ell} I(X_i; T_i^{\overrightarrow{\ell}} \mid T_i^{\overrightarrow{<\ell}} R^pMZ) \ ,$$

and thus

$$\sum_{i=1}^{k} I(X_i; \Pi_i \mid R^pMZ) = \sum_i \sum_{\ell} I(X_i; T_i^{\overrightarrow{\ell}} \mid T_i^{\overrightarrow{<\ell}} R^pMZ) \ . \tag{4}$$

We note that the equation $I(X_i; \Pi_i \mid R^pMZ) = \sum_{\ell} I(X_i; T_i^{\overrightarrow{\ell}} \mid T_i^{\overrightarrow{<\ell}} R^pMZ)$ that we proved above formalizes the intuitive assertion that if we consider the messages of Π_i in their order of appearance, then additional information on X_i is obtained only from messages *sent by* player i, but not from messages *received by* player i.

We now relate $\widehat{\mathsf{IC}}$ to the right-hand-side of Eq. (4). Starting from the definition of $\widehat{\mathsf{IC}}$ and using the chain rule, we decompose $\widehat{\mathsf{IC}}$ into a sum over all messages received in the protocol:

$$\begin{aligned}\widehat{\mathsf{IC}}(\pi) &= \sum_{j=1}^{k}\sum_{\ell'\geq 0} I(X_{-j}; T_j^{\overleftarrow{\ell'}} \mid T_j^{\overleftarrow{0}} \dots T_j^{\overleftarrow{\ell'-1}} X_j R^p M Z) \\ &= \sum_{j=1}^{k}\sum_{\ell'\geq 0} I(X_{-j}; T_j^{\overleftarrow{\ell'}} \mid T_j^{<\overleftarrow{\ell'}} X_j R^p M Z)\ ,\end{aligned}$$

where the second equality follows from the fact that the messages in $T_j^{<\overleftarrow{\ell'}}$ which are *sent* by player j are a function of X_j, R^p and of the messages in $T_j^{<\overleftarrow{\ell'}}$ which are *received* by player j.

We now rearrange this sum by considering the messages from the point of view of the sender rather than the receiver. In what follows we use j as a shorthand of $j(i,\ell)$ and ℓ' as a shorthand of $\ell'(i,\ell)$.[13] We have

$$\widehat{\mathsf{IC}}(\pi) = \sum_{i=1}^{k}\sum_{\ell\geq 0} I(X_{-j}; T_i^{\overrightarrow{\ell}} \mid T_j^{<\overleftarrow{\ell'}} X_j R^p M Z)\ .$$

To conclude the proof our objective now is to show that for any message $T_i^{\overrightarrow{\ell}}$,

$$I(X_i; T_i^{\overrightarrow{\ell}} \mid T_i^{<\overrightarrow{\ell}} R^p M Z) \leq I(X_{-j}; T_i^{\overrightarrow{\ell}} \mid T_j^{<\overleftarrow{\ell'}} X_j R^p M Z)\ . \tag{5}$$

Observe that since $T_i^{\overrightarrow{\ell}}$ is determined by $X_i R^p T_i^{<\overrightarrow{\ell}}$, we have $H(T_i^{\overrightarrow{\ell}} \mid X_i T_i^{<\overrightarrow{\ell}} R^p M Z) = 0$, and thus $I(X_i; T_i^{\overrightarrow{\ell}} \mid T_i^{<\overrightarrow{\ell}} R^p M Z) = H(T_i^{\overrightarrow{\ell}} \mid T_i^{<\overrightarrow{\ell}} R^p M Z)$. Similarly, we have that $I(X_{-j}; T_i^{\overrightarrow{\ell}} \mid T_j^{<\overleftarrow{\ell'}} X_j R^p M Z) = H(T_i^{\overrightarrow{\ell}} \mid T_j^{<\overleftarrow{\ell'}} X_j R^p M Z)$. Thus,

$$\begin{gathered} I(X_i; T_i^{\overrightarrow{\ell}} \mid T_i^{<\overrightarrow{\ell}} R^p M Z) \leq I(X_{-j}; T_i^{\overrightarrow{\ell}} \mid T_j^{<\overleftarrow{\ell'}} X_j R^p M Z) \\ \Updownarrow \\ H(T_i^{\overrightarrow{\ell}} \mid T_i^{<\overrightarrow{\ell}} R^p M Z) \leq H(T_i^{\overrightarrow{\ell}} \mid T_j^{<\overleftarrow{\ell'}} X_j R^p M Z) \\ \Updownarrow \\ I(T_i^{\overrightarrow{\ell}}; T_i^{<\overrightarrow{\ell}} R^p M Z) \geq I(T_i^{\overrightarrow{\ell}}; T_j^{<\overleftarrow{\ell'}} X_j R^p M Z)\ . \end{gathered}$$

The last inequality clearly holds if $I(T_i^{\overrightarrow{l}}; T_i^{<\overrightarrow{\ell}} R^p M Z) = I(T_i^{\overrightarrow{\ell}}; T_i^{<\overrightarrow{\ell}} T_j^{<\overleftarrow{\ell'}} X_j R^p M Z)$, which itself holds if

$$I(T_i^{\overrightarrow{\ell}}; T_j^{<\overleftarrow{\ell'}} X_j \mid T_i^{<\overrightarrow{\ell}} R^p M Z) = 0\ . \tag{6}$$

Notice that given the value of $T_i^{<\overrightarrow{l}} R^p M Z$, $T_i^{\overrightarrow{\ell}}$ is determined by X_i, and thus by the data processing inequality (Proposition A.7) we have

$$I(X_i; T_j^{<\overleftarrow{\ell'}} X_j \mid T_i^{<\overrightarrow{\ell}} R^p M Z) \geq I(T_i^{\overrightarrow{\ell}}; T_j^{<\overleftarrow{\ell'}} X_j \mid T_i^{<\overrightarrow{\ell}} R^p M Z)\ ,$$

[13] Recall that $j(i,\ell)$ and $\ell'(i,\ell)$ are defined such that message $T_i^{\overrightarrow{\ell}}$ is identified with the message $T_{j(i,\ell)}^{\overleftarrow{\ell'}(i,\ell)}$.

and Eq. 6 holds if $I(X_i; T_j^{<\overleftarrow{\ell'}} X_j \mid T_i^{<\overrightarrow{\ell}} R^p M Z) = 0$.

Let t be the (global) round in which player j receives $T_j^{\overleftarrow{\ell'}}$, which is also the (global) round in which player i sends $T_i^{\overrightarrow{\ell}}$ (recall that in fact $T_j^{\overleftarrow{\ell'}}$ is the same message as $T_i^{\overrightarrow{\ell}}$). Now, all the messages in $T_j^{<\overleftarrow{\ell'}}$ are received or sent by player j no later than (global) round t, and all the messages sent by player i which are not in $T_i^{<\overrightarrow{\ell}}$ are sent by player i no earlier than (global) round t. Hence, $T_j^{<\overleftarrow{\ell'}}$ is a function of $(X_{-i}, T_i^{<\overrightarrow{\ell}})$. We also have trivially that X_j is a function of X_{-i}. Thus, the data processing inequality implies that

$$I(X_i; T_j^{<\overleftarrow{\ell'}} X_j \mid T_i^{<\overrightarrow{\ell}} R^p M Z) \leq I(X_i; X_{-i} T_i^{<\overrightarrow{\ell}} \mid T_i^{<\overrightarrow{\ell}} R^p M Z) = I(X_i; X_{-i} \mid T_i^{<\overrightarrow{\ell}} R^p M Z)\,.$$

By Eq. (3), $I(X_i; X_{-i} \mid T_i^{<\overrightarrow{\ell}} R^p M Z) = 0$, which concludes the proof of Eq. (6) and therefore of Eq. (5) and of the lemma. ⌟