# Multi-Party Protocols, Information Complexity and Privacy

Iordanis Kerenidis[*] Adi Rosén[†] Florent Urrutia[‡]

**Abstract**

We introduce a new information theoretic measure that we call *Public Information Complexity* (PIC), as a tool for the study of multi-party computation protocols, and of quantities such as their communication complexity, or the amount of randomness they require in the context of information-theoretic private computations. We are able to use this measure directly in the natural asynchronous message-passing *peer-to-peer* model and show a number of interesting properties and applications of our new notion: the Public Information Complexity is a lower bound on the Communication Complexity and an upper bound on the Information Complexity; the difference between the Public Information Complexity and the Information Complexity provides a lower bound on the amount of randomness used in a protocol; any communication protocol can be compressed to its Public Information Cost; an explicit calculation of the zero-error Public Information Complexity of the $k$-party, $n$-bit Parity function, where a player outputs the bit-wise parity of the inputs. The latter result also establishes that the amount of randomness needed by a private protocol that computes this function is $\Omega(n)$.

## 1 Introduction

Communication complexity, originally introduced by Yao [Yao79], is a prolific field of research in theoretical computer science that yielded many important results in various fields. Informally, it attempts to answer the question “How many bits must distributed players transmit to solve a given distributed problem ?” The study of the two-party case has produced a large number of interesting and important results, both upper and lower bounds, with many applications in other areas in theoretical computer science such as circuit complexity, data structures, streaming algorithms and distributed computation (see, e.g., [KN97, MNSW98, GG10, SHK$^+$10, FHW12]).

A powerful tool recently introduced for the study of two-party communication protocols is the measure of *Information Complexity* (or *cost*). This measure, originally defined in [BCKO93] and [CSWY01], extends the notions of information theory, originally introduced by Shannon [Sha48], to interactive settings. It quantifies, roughly speaking, the amount of information about their respective inputs that Alice and Bob must leak to each other in order to compute a given function $f$ of their inputs. Information complexity (IC) has been used in a long series of papers to prove lower bounds on communication complexity and other properties of (two-party) communication protocols (e.g., [BJKS04, BBCR13, BR14, Bra15]). An interesting property of information complexity is that it satisfies a direct sum property. The *direct sum* question, one of the most interesting questions in complexity theory, asks whether solving $n$ independent copies of the same problem must cost (in a given measure) $n$ times the cost of solving a single instance. In the case of communication complexity, this

[*] CNRS and Université Paris Diderot, email: jkeren@irif.fr.
[†] CNRS and Université Paris Diderot, email: adiro@irif.fr.
[‡] Université Paris Diderot, email: urrutia@irif.fr.

question has been studied in, e.g., [FKNN95, CSWY01, Sha03, JRS03, HJMR10, BBCR13, Kla10, Jai15] and in many cases it remains open whether a direct sum property holds.

Another important question in communication complexity is the relation between the information complexity of a function and its communication complexity. One would like to know if it is possible to compute a function by sending a number of bits which is not (too much) more than the information the protocol actually has to reveal. Put differently, is it always possible to *compress* the communication cost of a protocol to its information cost? For the two-party case it is known that perfect compression is not possible for single shot protocols [GKR15a, GKR15b] (unless they are restricted to a constant number of rounds [JRS03]). Still, several interesting compression results are known. The equality between information cost and *amortized* communication cost is shown in [BR14, Bra15], and other compression techniques are given in [BBCR13, BMY15, BBK+16, Pan15]. It remains open if one can compress interactive communication up to some small loss (for example logarithmic in the size of the input). The specific case of compression under product distributions was studied in [Kol16, She18], leading to a compression to $\mathcal{O}(I \text{ polylog}(I))$.

When trying to study the *multi-party* (i.e., where at least 3 players are involved) communication settings using similar information-theoretic methods, such as IC, one encounters a serious problem. The celebrated results on information-theoretic private computation [BOGW88, CCD88] state that if the number of players is at least 3, then *any function* can be computed by a randomized protocol such that *no information* about the inputs is revealed to the other players (other than what is implied by the value of the function and their own input). Thus, in the multi-party case, the $\mathsf{IC}$ of any function $f$ is 0 (or only the entropy of $f$, depending on the definition of $\mathsf{IC}$), and cannot serve to study multi-party protocols.

For this reason, information theory has rarely been used in the multi-party setting, where most results related to communication complexity have been obtained via combinatorial techniques. Among the interesting works on multi-party settings are [PVZ16, WZ14] which introduce the techniques of *symmetrization* and *composition*, and [CRR14, CR15] which study the influence of the topology of the network. One notable exception is the work of Braverman et al. [BEO+13] which studies the *set-disjointness* problem using information theoretic tools. Braverman et al. provide almost tight bounds in the so-called coordinator model (that differs from the more natural peer-to-peer model) by analyzing the information leaked between the players but also the information obtained by the coordinator itself. The set disjointness problem is arguably one of the most extensively studied problem in communication complexity (cf. [Bra15, BJKS04, CKS03, Gro09, Jay09, BGPW13]). This line of research was followed by [CM15] which also uses information theory to obtain tight bounds on the communication complexity of the function *Tribes* in the coordinator model. Information theory is also used in [BO15] to study set-disjointness in the broadcast model. A compression procedure for the broadcast model is described in [KOS17].

A number of sub-models have been considered in the literature for the multi-party computation protocols setting: the *number in hand model* (NIH), where each player has a private input, is arguably the most natural one, while in the *number on the forehead model* (NOF), each player $i$ knows all inputs $x_j$, $j \neq i$, i.e., the "inputs" of all players except its own. As to the communication pattern, a number of variants have been considered as well: in the *blackboard* model, the players communicate by broadcasting messages (or writing them on a "blackboard"); in the *message passing* model, each pair of players is given a private channel to mutually communicate (for more details on the different variants see [KN97]). Most of the results obtained in multi-party communication complexity were obtained for the NOF model or the blackboard model. The present paper studies, however, the NIH, message passing (peer to peer) model, which is also the most closely related to the work done on message passing protocols in the distributed computing and networking communities.

### 1.1 Our contributions

Our main goal is to introduce novel information-theoretical measures for the study of number-in-hand, message-passing multi-party protocols, coupled with a natural model that, among other things, allows private protocols (which is not the case for, e.g., the coordinator model).

We define the new measure of *Public Information Complexity* (PIC), as a tool for the study of multi-party computation protocols, and of quantities such as their communication complexity, or the amount of randomness they require in the context of information-theoretic private computations. Intuitively, our new measure captures a combination of the amount of information about the inputs that the players leak to other players, and the amount of randomness that the protocol uses. By proving lower bounds on PIC for a given multi-party function $f$, we are able to give lower bounds on the multi-party communication complexity of $f$ and on the amount of randomness needed to privately compute $f$. The crucial point is that the PIC of functions, in our multi-party model, is not always $0$, unlike their IC.

Our new measure works in a model which is a slight restriction of the most general asynchronous model, where, for a given player at a given time, the set of players from which that player waits for a message can be determined by that player's own local view. This allows us to have the property that for any protocol, the information which is leaked during the execution of the protocol is at most the communication cost of the protocol. Note that in the multi-party case, the information cost of a protocol may be higher than its communication cost, because the identity of the player from which one receives a message might carry some information. We are able to define our measure and use it directly in a natural asynchronous *peer-to-peer* model (and not, e.g., in the coordinator model used in most works studying the multi-party case, c.f. [DF89]). The latter point is particularly important when one is interested in private computation, since our model allows for private protocols, while this is not necessarily the case for other models. Furthermore, if one seeks to accurately understand communication complexity in the natural peer-to-peer model, suppressing polylog-factor inaccuracies, one has to study directly the peer-to-peer model, because lower bounds in the coordintor model translate to the peer-to-peer model only up to logarithmic factors (see the comparison of models in subsection 3.1).

We then continue and show a number of interesting properties and applications of our new notion:

- The Public Information Complexity is a lower bound on the Communication Complexity and an upper bound on the Information Complexity. In fact, it can be strictly larger than the Information Complexity.
- The difference between the Public Information Complexity and the Information Complexity provides a lower bound on the amount of randomness used in a protocol.
- We compress any communication protocol to their PIC (up to logarithmic factors), by extending to the multi-party setting the work of Brody et al. [BBK$^+$16] and Pankratov [Pan15].
- We show that one can approach the central question of direct sum in communication complexity by trying to prove a direct sum result for PIC. Indeed, we show that a direct sum property for PIC implies a certain direct sum property for communication complexity.
- We precisely calculate the zero-error *Public Information Complexity* of the $k$-party, $n$-bit Parity function (Par), where a player outputs the bit-wise parity of the inputs. We show that the PIC of this function is $n(k-1)$. This result is tight and it also establishes that the amount of randomness needed for a private protocol that computes this function is $\Omega(n)$. While this sounds a reasonable assertion no previous proof for such claim existed.

### 1.2 Organization

The paper is organized as follows. In section 2 we review several notations and information theory basics. In Section 3 we define the communication model that we work with and a number of traditional complexity measures. In Section 4 we define the new measure PIC that we introduce in the present paper, and in Section 5 we discuss its relation to randomness and multi-party private computation. In Section 6 we give tight bounds for the parity function Par, using PIC. In section 7, we discuss the existence of a direct sum property for PIC.

## 2 Preliminaries

We start by defining a number of 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. $\pi$ will be used to denote a protocol. We use the term *entropy* to talk about binary entropy.

We give a reminder on basic information theory, as introduced in [Sha48].

**Definition 2.1.** *The entropy of a (discrete) random variable $X$ is*

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

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

**Proposition 2.2.** *For any finite set $\mathcal{X} \subseteq \{0,1\}^*$ and any random variable $X$ with support $supp(X) \subseteq \mathcal{X}$, it holds*

$$H(X) \leq \log(|\mathcal{X}|).$$

*Moreover, if the set $\mathcal{X}$ is prefix-free, it holds $H(X) \leq \mathbb{E}[|X|]$.*

**Definition 2.3.** *The mutual information between two random variables $X, Y$ is*

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

*The conditional mutual information $I(X;Y \mid Z)$ is $H(X \mid Z) - H(X \mid YZ)$.*

The mutual information measures the change in the entropy of $X$ when one learns the value of $Y$. It is non negative, and is symmetric: $I(X;Y) = I(Y;X)$.

**Proposition 2.4.** *For any random variables $X$, $Y$ and $Z$, $I(X;Y \mid Z) = 0$ if and only if $X$ and $Y$ are independent for each possible value of $Z$.*

**Proposition 2.5.** *For any random variables $X$ and $Y$, $H(X \mid Y) \leq H(X)$.*

**Proposition 2.6** (Chain Rule)**.** *Let $A$, $B$, $C$, $D$ be four random variables. Then*

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

**Lemma 2.7** (Data processing inequality)**.** *For any $X$, $Y$, $Z$, and any function $f$, it holds: $I(X; f(Y) \mid Z) \leq I(X; Y \mid Z)$*

*Proof.*

$$\begin{aligned} I(X; f(Y) \mid Z) &\leq I(X; f(Y) \mid Z) + I(X; Y \mid f(Y)Z) \\ &= I(X; Y f(Y) \mid Z) \\ &= I(X; Y \mid Z) + I(X; f(Y) \mid YZ) \\ &= I(X; Y \mid Z). \end{aligned}$$

□

**Proposition 2.8** ([Bra15])**.** *Let $A$, $B$, $C$, $D$ be four random variables such that $I(B; D \mid AC) = 0$. Then*

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

**Proposition 2.9** ([Bra15])**.** *Let $A$, $B$, $C$, $D$ be four random variables such that $I(B; D \mid C) = 0$. Then*

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

# 3 The model

We now define a natural communication model which is a slight restriction of the most general asynchronous peer-to-peer model. Its restriction is that for a given player at a given time, the set of players from which that player waits for a message can be determined by that player's own local view. The player continues its computation only after messages are received from this set. This allows us to define information theoretical 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 without such restriction the information revealed by the execution of a protocol might be higher than the number of bits transmitted and that, on the other hand, practically all multi-party protocols in the literature are implicitly defined in our model. We also compare our model to the general one and to other restricted ones and explain the usefulness and logic of our specific model.

## 3.1 Definition of the model

We work in the *multi-party number in hand peer-to-peer* model. Each player 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. The system consists of $k$ players and a family of $k$ functions $f = (f_i)_{i \in \llbracket 1,k \rrbracket}$, with $\forall\, i \in \llbracket 1, k \rrbracket, f_i : \prod_{l=1}^{k} \mathcal{X}_l \to \mathcal{Y}_i$, where $\mathcal{X}_l$ 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 $x = (x_i) \in \prod_{i=1}^{k} \mathcal{X}_i$, and for every $i$, player $i$ has to compute $f_i(x)$. Each player has a special write-only output tape.

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 in both directions. There is no bound on the delivery time (i.e., when the message arrives to its destination node) of a message, but every message is delivered in finite time, and the communication link maintains FIFO order in each of the two directions. Messages that arrive to the head of the link at the destination node of that link are buffered until they are read by that node. Given a specific time we define the *view* of player $i$, denoted $D_i$, as the input of that player, $X_i$, its private randomness, $R_i$, and the messages received so far by player $i$. The protocol of each player $i$ runs 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 to write a (nonempty) string on its output tape. Then, the player waits for messages from a certain subset of the other players, where this subset is also determined by the current view of the player. That is, the player reads a single message from each of the incoming links that connect it to that subset of other players. If for a certain such link no message is available, then the player waits until such message is available (i.e., arrives). Then the (local) round of player $i$ terminates[1]. 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 be self-delimiting.

Denote by $\mathcal{D}_i^j$ the set of possible views of player $i$ at the end of local round $j$, $j \geq 0$, where the beginning of the protocol is considered round $0$. Formally, a protocol $\pi$ 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 $j$, $j \geq 1$:

- $\overline{S}_i^j : \mathcal{D}_i^{j-1} \to 2^{\{1,\ldots,k\}\setminus\{i\}}$, defining the set of players to which player $i$ *sends* the messages.
- $m_{i,q}^j : \mathcal{D}_i^{j-1} \to \{0,1\}^*$, such that for any $D_i^{j-1} \in \mathcal{D}_i^{j-1}$, if $q \in \overline{S}_i^j(D_i^{j-1})$, then $m_{i,q}^j(D_i^{j-1})$ is the content of the message player $i$ sends to player $q$. Each such message is self-delimiting.
- $O_i^j : \mathcal{D}_i^{j-1} \to \{0,1\}^*$, defining what the player writes on the output tape. Each player can write on its output tape a non-empty string only once.[2]
- $S_i^j : \mathcal{D}_i^{j-1} \to 2^{\{1,\ldots,k\}\setminus\{i\}}$, defining the set of players from which player $i$ waits to *receive* a message.

We define the transcript of the protocol of player $i$, denoted $\Pi_i$, as the concatenation of the messages read by player $i$ from the links of the sets $S_i^1, S_i^2, \ldots$, ordered by local round number, and within each round by the index of the player. We denote by $\overleftrightarrow{\Pi_i}$ the concatenation of $\Pi_i$ together with a similar concatenation $\overrightarrow{\Pi_i}$ of the messages sent by player $i$ to the sets $\overline{S}_i^0, \overline{S}_i^1, \ldots$ We denote by $\Pi_{i \to j}$ the concatenation of the messages sent by player $i$ to player $j$ during the course of the protocol. The transcript of the (whole) protocol, denoted $\Pi$, is obtained by concatenating all the $\Pi_i$ ordered by, say, player index.

We will give most of the 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 3.1.** *For $\epsilon \geq 0$, a protocol $\pi$ $\epsilon$-computes a function $f$ if for all $(x_1, \ldots, x_k) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$:*

1. *For all possible assignments for the random sources $R_i$, $1 \leq i \leq k$, and $R^p$, every player eventually (i.e., in finite time) writes on its output tape (a non-empty string).*

[1] 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 writing on the output tape are a function of only the messages received in previous (local) rounds.

[2] We require that each player writes only once on its output tape so that the local view of the player determines the local output of the protocol (i.e., so that the player itself "knows" the output). This requirement is needed since a player may not know locally that the protocol ended.

2. *With probability at least $1 - \epsilon$ (over all random sources) the following event occurs: each player $i$ writes on its output tape the value $f(x)$, i.e., the correct value of the function.*

For simplicity we also assume that a protocol must eventually stop. That is, for all possible inputs and all possible assignments for the random sources, eventually (i.e., in finite time) there is no message in transit.

### 3.2 Comparison to other models

The somewhat restricted model (compared to the general asynchronous model) that we work with allows us to define a measure similar to information cost that we will later show to have desirable properties and to be of use. Notice that the general asynchronous model is problematic in this respect since one bit of communication can bring $\log(k)$ bits of information, as 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 $S_i^l$ and $\overline{S}_i^l$ are determined by the current view of the player, $(\Pi_i)$ 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 the above mentioned issue in the general asynchronous model consider the following simple example of a deterministic protocol, for 4 players $A$, $B$ and $C$, $D$, which allows $A$ to transmit to $B$ its input bit $x$, but where all messages sent in the protocol are the bit 0, and the protocol generates only a single transcript over all possible inputs.

**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.

There has been a long series of works about multi-party communication protocols in different variants of models, for example [CKS03, Gro09, Jay09, PVZ16, CRR14, CR15]. In [BEO+13], Braverman et al. consider a restricted class of protocols working in the *coordinator model*: an additional player 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 model (neither in our model nor in the most general one). Namely, a 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 any message to the coordinator with a $O(\log k)$-bit label indicating its destination. This factor is sometimes necessary, e.g., for the $q$-index function, where players $P_i$, $0 \leq i \leq k-1$, each holds an input bit $x_i$, player $P_k$ holds $q$ indices $0 \leq j_\ell \leq k-1$, $1 \leq \ell \leq q$, and $P_k$ should learn the vector $(x_{j_1}, x_{j_1}, \ldots, x_{j_q})$: in the coordinator model the communication complexity of this function is $\Theta(\min\{k, q \log k\})$, while in both peer-to-peer models there is a protocol for this function that sends only (at most) $\min\{k, 2q\}$ bits, where $P_k$ just queries the appropriate other players. 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 models and in the coordinator model.

Moreover, when studying private protocols in the peer-to-peer model, the coordinator model does not offer any insight. In the (asynchronous) coordinator model, described in [DF89] and used for instance in [BEO+13], if there is no 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 [FKN94]. If no such source exists, privacy is impossible in general. This follows from the results of Braverman et al. [BEO+13] 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.

Note also that the private protocols described in [BOGW88, CCD88] (and further work) are defined in the synchronous setting, and thus can be adapted to our communication model (the sets $\overline{S}_i^j$ and $S_i^j$ are always all the players and hence even independent of the current views).

In the sequel we also use a special case of our model, where the sets $\overline{S}_i^j$ and $S_i^j$ are a function only of $i$ and $j$, and not of the entire current view of the player. This is a natural special case for protocols which we call *oblivious protocols*, where the communication pattern is fixed and is not a function of the input or randomness. Clearly, the messages themselves remain a function of the view of the players. We observe that synchronous protocols are a special case of oblivious protocols.

### 3.3 Communication complexity and information complexity

Communication complexity, introduced in [Yao79], measures how many bits of communication are needed in order for a set of players to compute with error $\epsilon$ a given function of their inputs. The allowed error $\epsilon$, implicit in many of the contexts, will be written explicitly as a superscript when necessary.

**Definition 3.2.** *The communication cost of a protocol $\pi$, $\mathsf{CC}(\pi)$, is the maximal length of the transcript of $\pi$ over all possible inputs, private randomness and public randomness.*

**Definition 3.3.** $\mathsf{CC}(f)$ *denotes the communication cost of the best protocol computing $f$.*

Information complexity measures the amount of information that must be transmitted so that the players can compute a given function of their joint inputs. One of its main uses is to provide a lower bound on the communication complexity of the function. In the two-party setting, this measure led to interesting results on the communication complexity of various functions, such as *AND* and *Disjointness*. We now focus on designing an analogue to the information cost, for the multi-party setting. The notion of internal information cost for two-party protocols (c.f. [CSWY01, BJKS04, Bra15]) can be easily generalized to any number of players:

**Definition 3.4.** *The internal information cost of a protocol $\pi$ for $k$ players, with respect to input distribution $\mu$, is the sum of the information revealed to each player about the inputs of the other players:*

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

Intuitively, the information cost of a protocol is the amount of information each player learns about the inputs of the other players during the protocol. The definition we give above, when restricted to two players is the same as in [Bra15], even though they look slightly different. This is because we make explicit the role of the randomness, which will allow us to later give bounds on the amount of randomness needed for private protocols in the multi-party setting.

The *internal information complexity* of a function $f$ with respect to input distribution $\mu$, as well as the *internal information complexity* of a function $f$, can be defined for the multi-party case based on the information cost of a protocol, just as in the 2-party case.

**Definition 3.5.** *The internal information complexity of a function $f$, with respect to input distribution $\mu$ is the infimum of the internal information cost over all protocols computing $f$ on input distribution $\mu$:*

$$\mathsf{IC}_\mu(f) = \inf_{\pi \text{ computing } f} \mathsf{IC}_\mu(\pi).$$

The information revealed to a given player by a protocol can be written in several ways:

**Proposition 3.6.** *For any protocol $\pi$, for any player $i$:*

$$I(X_{-i};\ \overleftrightarrow{\Pi_i} \mid X_i R_i R^p) = I(X_{-i}; \Pi_i \mid X_i R_i R^p).$$

*Proof.* For any protocol $\pi$, for any player $i$:

$$\begin{aligned} I(X_{-i};\ \overleftrightarrow{\Pi_i} \mid X_i R_i R^p) &= I(X_{-i};\ \overrightarrow{\Pi_i}\Pi_i \mid X_i R_i R^p) \\ &= I(X_{-i}; \Pi_i \mid X_i R_i R^p) + I(X_{-i};\ \overrightarrow{\Pi_i} \mid X_i R_i R^p \Pi_i) \quad \text{(chain rule)} \\ &= I(X_{-i}; \Pi_i \mid X_i R_i R^p) \quad (\text{since } H(\overrightarrow{\Pi_i} \mid X_i R_i R^p \Pi_i) = 0)\ . \end{aligned}$$

□

## 3.4 Information complexity and privacy

The definition of a *private protocol* as defined in [BOGW88, CCD88] is the following.

**Definition 3.7.** *A $k$-player protocol $\pi$ for computing a family of functions $(f_i)$ is private*[3] *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 $f_i(x) = f_i(x')$ and $x_i = x'_i$, for all possible private random tapes $r_i$ of player $i$, and all possible public random tapes $r^p$, 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 $R_{-i}$.*

The notion of privacy has an equivalent formulation in terms of information.

**Proposition 3.8.** *A protocol $\pi$ is private if and only if for all input distributions $\mu$,*

$$\sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R_i R^p f_i(X)) = 0.$$

*Proof.* By proposition 2.4, definition 3.7 is equivalent to the following:

$$\forall\, i, I(X_{-i}; \Pi_i \mid X_i R_i R^p f_i(X)) = 0\ .$$

Since $I$ is non-negative, this is equivalent to

$$\sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R_i R^p f_i(X)) = 0\ .$$

□

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

It is well known that in the multi-party number-in-hand peer-to-peer setting (for $k \geq 3$), unlike in the two-party case, *any* function can be privately computed.

**Theorem 3.9** ([BOGW88],[CCD88]). *Any family of functions of more than two variables can be computed by a private protocol.*

Using the above theorem, we can give the following lemma.

**Lemma 3.10.** *For any family of functions $(f_i)$ of more than two variables and any $\mu$,*
$\mathsf{IC}_\mu(f) \leq \sum_{i=1}^{k} H(f_i(X))$, *where $X$ is distributed according to $\mu$.*

*Proof.* Let $\pi$ be a $k$-player private protocol computing $(f_i)$. Fix a distribution $\mu$ on the inputs.

$$\begin{aligned}
\mathsf{IC}_\mu(\pi) &= \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R_i R^p) \\
&\leq \sum_{i=1}^{k} I(X_{-i}; \Pi_i f_i(X) \mid X_i R_i R^p) \\
&= \sum_{i=1}^{k} [I(X_{-i}; f_i(X) \mid X_i R_i R^p) + I(X_{-i}; \Pi_i \mid X_i R_i R^p f_i(X))] \\
&= \sum_{i=1}^{k} I(X_{-i}; f_i(X) \mid X_i R_i R^p) \\
&\leq \sum_{i=1}^{k} H(f_i(X)).
\end{aligned}$$

Now, $\mathsf{IC}_\mu(f) \leq \mathsf{IC}_\mu(\pi) \leq \sum_{i=1}^{k} H(f_i(X))$.

□

This lemma shows that $\mathsf{IC}$ cannot be used in the multi-party setting for any meaningful lower bounds on the communication complexity, since its value is always upper bounded by the entropies of the functions. Our goal is to get lower bounds tight in both $k$ and $n$. For this reason, we introduce a new information-theoretic quantity for the multi-party setting.

# 4 The new measure: Public Information Cost

We now introduce a new information theoretic quantity which can be used instead of $\mathsf{IC}$ in the multi-party setting. The notion we define will be suitable for studying multi-party communication in a model which is only a slight restriction on the general asynchronous model, and which allows for private protocols. This means that while $\mathsf{IC}$ will be at most the entropies of the functions, our new notion remains a strong lower bound for communication.

**Definition 4.1.** *For any $k$-player protocol $\pi$ and any input distribution $\mu$, we define the public information cost of $\pi$:*

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

The difference between the definition of $\mathsf{PIC}$ and that of $\mathsf{IC}$ is the presence of the other parties' private randomness, $R_{-i}$, in the formula. Thus, if $\pi$ is a protocol using only public randomness, then for any input distribution $\mu$, $\mathsf{PIC}_\mu(\pi) = \mathsf{IC}_\mu(\pi)$, and hence the name "public information cost".

Informally speaking, the public information cost measures both the information about the inputs learned by the players and the information that is hidden by the use of private coins. $\mathsf{PIC}$ can be decomposed, using the chain rule, into two terms, making explicit the contribution of the internal information cost and that of the private randomness of the players.

**Proposition 4.2.** *For any $k$-player protocol $\pi$ and any input distribution $\mu$,*

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

A possible intuitive meaning of the second term could be the following. At the end of the protocol, player $i$ knows its input $X_i$, its private coins $R_i$, the public coins $R^p$ and its transcript $\Pi_i$. Suppose that the private randomness $R_{-i}$ of the other players is now revealed to player $i$. This brings to that player some new information, quantified by $I(R_{-i}; X_{-i}|X_i\Pi_i R_i R^p)$, about the inputs $X_{-i}$ of the other players.

We also define the public information complexity of a function, given error probability $\epsilon$. In the sequel, when clear from the context, we sometimes omit $\epsilon$.

**Definition 4.3.** *For any function $f$, $\epsilon \geq 0$, and any input distribution $\mu$, we define the quantity*

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

**Definition 4.4.** *For any $f$, we define the quantity*

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

The public information cost is a lower bound on the communication complexity.

**Proposition 4.5.** *For any protocol $\pi$ and input distribution $\mu$, $\mathsf{CC}(\pi) \geq \mathsf{PIC}_\mu(\pi)$. Thus, for any function $f$, $\mathsf{CC}(f) \geq \mathsf{PIC}(f)$.*

*Proof.*

$$\begin{aligned}
\mathsf{PIC}_\mu(\pi) &= \sum_{i=1}^{k} I(X_{-i}; R_{-i} \mid X_i R_i R^p) + I(X_{-i}; \Pi_i \mid X_i R R^p) \quad \text{(by the chain rule)} \\
&= \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R R^p) \quad \text{(since the first term is 0 by Proposition 2.4)} \\
&= \sum_{i=1}^{k} H(\Pi_i \mid X_i R R^p) \quad \text{(since } H(\Pi_i \mid X R R^p) = 0 \text{ )} \\
&\leq \sum_{i=1}^{k} H(\Pi_i) \quad \text{(by Proposition 2.5)} \ .
\end{aligned}$$

Using Proposition 2.2, for each $i$, $H(\Pi_i)$ is upper bounded by the expected size of $\Pi_i$. As the expected size of $\Pi$ is upper bounded by the sum over $i$ of the expected size of $\Pi_i$, we get $\mathsf{CC}(\pi) \geq \mathsf{PIC}_\mu(\pi)$.

□

In fact, as we show below, the public information cost of a function is equal to its information cost (IC) in a setting where only public randomness is allowed. The role of private coins in communication protocols has been studied for example in [BG14, BBK+16, Koz15]. In the next section we will see that the difference between the public information cost and the information cost is related to the private coins used during the protocol.

**Theorem 4.6.** *For any protocol $\pi$ there exists a public coin protocol $\pi'$ such that for all input distributions $\mu$, $\mathsf{PIC}_\mu(\pi') = \mathsf{PIC}_\mu(\pi)$. If $\pi$ is oblivious then so is $\pi'$.*

*Proof.* Given an an arbitrary protocol $\pi$, we build a public coin protocol $\pi'$ as follows.
Let $R'^p$ denote the public random tape of $\pi'$. We consider $R'^p$ as being composed of $k+1$ parts, one to be used as the public random tape of $\pi$, and the $k$ other parts as the $k$ private random tapes, in $\pi$, of the $k$ players. This can be done by interleaving the $k+1$ tapes bit by bit on $R'^p$. We denote the $k+1$ resulting tapes as $R^p$ and $R_i$, $1 \leq i \leq k$. Protocol $\pi'$ is then defined as running protocol $\pi$, when the players use the corresponding "public random tape" $R^p$ and "private random tapes" $R_i$, as defined for $\pi$. Observe that the transcripts of $\pi'$ and $\pi'$ are therefore identical. Observe also that if $\pi$ is oblivious, so is $\pi'$. Let $R$ denote $(R_i)$. We have,

$$\begin{aligned}
\mathsf{PIC}_\mu(\pi') &= \sum_{i=1}^{k} I(X_{-i};\Pi'_i \mid X_i R'^p) \\
&= \sum_{i=1}^{k} I(X_{-i};\Pi'_i \mid X_i R R^p) \\
&= \sum_{i=1}^{k} [I(X_{-i};\Pi'_i R_{-i} \mid X_i R_i R^p) - I(X_{-i};R_{-i} \mid X_i R_i R^p)] \text{ (chain rule)} \\
&= \sum_{i=1}^{k} I(X_{-i};\Pi'_i R_{-i} \mid X_i R_i R^p) \text{ (since the second term equals 0)} \\
&= \sum_{i=1}^{k} I(X_{-i};\Pi_i R_{-i} \mid X_i R_i R^p) \text{ (since the transcripts of } \pi' \text{ and of } \pi \text{ are identical)} \\
&= \mathsf{PIC}_\mu(\pi)\,.
\end{aligned}$$

□

The next theorem is a direct consequence of Theorem 4.6.

**Theorem 4.7.** *For any function $f$ and input distribution $\mu$,*

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

*and*

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

The following property of the public information cost will be useful for zero-error protocols.

**Proposition 4.8.** *For any function $f$, for any input distribution $\mu$, $\mathsf{PIC}^0_\mu(f) = \mathsf{IC}^{det}_\mu(f)$ where*

$$\mathsf{IC}^{det}_\mu(f) = \inf_{\pi \text{ deterministic protocol computing } f} \mathsf{IC}_\mu(\pi)\,.$$

*Proof.* Let $\delta > 0$ be arbitrary. To prove the claim we show that there exists a deterministic protocol computing $f$, $\pi^0$, such that $\mathsf{IC}_\mu(\pi^0) \leq \mathsf{PIC}^0_\mu(f) + \delta$.

Let $\pi$ be a zero-error protocol for $f$ such that $\mathsf{PIC}_\mu(\pi) \leq \mathsf{PIC}^0_\mu(f) + \frac{\delta}{2}$. By Theorem 4.7, one can assume that $\pi$ has no private randomness.

$$\begin{aligned}
\mathsf{IC}_\mu(\pi) &= \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R^p) \\
&= \sum_{i=1}^{k} \mathop{\mathbb{E}}_{r} \left[ I(X_{-i}; \Pi_i \mid X_i, R^p = r) \right] \\
&= \mathop{\mathbb{E}}_{r} \left[ \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i, R^p = r) \right] .
\end{aligned}$$

Letting $t(r) = \sum\limits_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i, R^p = r)$, it holds that $\mathsf{IC}_\mu(\pi) = \mathop{\mathbb{E}}\limits_{r}[t(r)]$. Let $r_0$ be a value of the public random tape such that $t(r_0) \leq \mathsf{IC}_\mu(\pi) + \frac{\delta}{2}$ and define $\pi^0$ to be the protocol operating like $\pi$ when the random tape is $r_0$. Note that $\pi^0$ is a deterministic (zero-error) protocol computing $f$.

$$\begin{aligned}
\mathsf{PIC}_\mu(\pi^0) &= \mathsf{IC}_\mu(\pi^0) \\
&= \sum_{i=1}^{k} I(X_{-i}; \Pi^0_i \mid X_i) \\
&= \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R = r_0) \\
&= t(r_0) \\
&\leq \mathsf{IC}_\mu(\pi) + \frac{\delta}{2} \\
&\leq \mathsf{PIC}_\mu(\pi) + \frac{\delta}{2} \\
&\leq \mathsf{PIC}^0_\mu(f) + \delta .
\end{aligned}$$

$\delta$ being arbitrary, this concludes the proof. □

We now observe that $\mathsf{PIC}$ and $\mathsf{IC}$ are strictly different even in the two-party case. We prove below that for the $\mathsf{AND}$ function, the public information cost is $\log(3) \simeq 1.58$, while, as shown in [BGPW13], $\mathsf{IC}^0(\mathsf{AND}) \simeq 1.49$. This implies that the protocol that achieves the optimal information cost for $\mathsf{AND}$ must use private coins. We remark that in [BGPW13] it is shown that the external information cost of $\mathsf{AND}$, that we do not consider here, is $\log(3)$.

**Proposition 4.9.** *For two players,* $\mathsf{PIC}^0(\mathsf{AND}) = \log_2 3 \simeq 1.58$.

*Proof.* In this proof we denote $\mathsf{PIC}^0(\cdot)$ by simply $\mathsf{PIC}(\cdot)$. We call a protocol $\pi'$ symmetric to $\pi$ (and an input distribution $\mu'$ symmetric to $\mu$) when the roles of Alice and Bob (or of the inputs $X$ and $Y$) are flipped.

We first prove that there exists a protocol $\pi^*$ for $\mathsf{AND}$ such that $\sup_\mu \mathsf{PIC}_\mu(\pi^*) = \inf_\pi \sup_\mu \mathsf{PIC}_\mu(\pi)$, where the infimum is over all protocols $\pi$ computing $\mathsf{AND}$. To this end we now prove that for any protocol

$\pi$ for AND it holds that $\sup_\mu \mathsf{PIC}_\mu(\pi^*) \le \sup_\mu \mathsf{PIC}_\mu(\pi)$, where $\pi^*$ is a protocol for AND that we define below. Consider an arbitrary protocol $\pi$. By Proposition 4.8 we can assume w.l.o.g. that $\pi$ is deterministic. Since $\pi$ computes AND there must be a non-constant bit sent in $\pi$. Assume w.l.o.g. that the first player to send a non-constant bit in $\pi$ is Alice (having input $X$).[4] Since $\pi$ is deterministic, this first non-constant bit is either Alice's input bit, $x$, or, $1-x$. Since Alice must compute the value of AND, we also have that $H(\mathsf{AND}(X,Y)|X\Pi_A) = 0$, where $\Pi_A$ is the transcript of the messages received by Alice. Consider now the protocol $\pi^*$ defined as follows: Alice sends her input bit $x$ to Bob; Bob, who can now compute $\mathsf{AND}(X,Y)$, sends to Alice that value.

For any input distribution $\mu$ we have

$$\begin{aligned}
\mathsf{PIC}(\pi^*) &= \mathsf{IC}_\mu(\pi^*) \\
&= I(X;\Pi^*_B \mid Y) + I(Y;\Pi^*_A \mid X) \\
&= H(X \mid Y) + (H(Y \mid X) - H(Y \mid X\ \mathsf{AND}(X,Y))) \\
&= I(X;\Pi_B \mid Y) + (H(Y \mid X) - H(Y \mid X\ \mathsf{AND}(X,Y))) \\
&\le I(X;\Pi_B \mid Y) + (H(Y \mid X) - H(Y \mid X\Pi_A)) \quad \text{(because } H(\mathsf{AND}(X,Y)|X\Pi_A) = 0\text{)} \\
&= I(X;\Pi_B \mid Y) + I(Y;\Pi_A \mid X) \\
&= \mathsf{PIC}(\pi)\ .
\end{aligned}$$

It follows that for any $\pi$ computing AND, it holds that $\sup_\mu \mathsf{PIC}_\mu(\pi^*) \le \sup_\mu \mathsf{PIC}_\mu(\pi)$, and hence $\sup_\mu \mathsf{PIC}_\mu(\pi^*) = \inf_\pi \sup_\mu \mathsf{PIC}_\mu(\pi)$.

To finalize the proof we now show that $\sup_\mu \mathsf{PIC}_\mu(\pi^*) = \mathsf{PIC}_{\mu^*}(\pi^*)$ for $\mu^*$ defined as follows: $X$ and $Y$ are independent; $X \sim \mathbf{Ber}(\frac{1}{3}, \frac{2}{3})$; $Y \sim \mathbf{Ber}(\frac{1}{2}, \frac{1}{2})$.

Consider an arbitrary input distribution $\mu$. Let $\alpha$ and $\beta$ be such that $\Pr_\mu[X=0] = \alpha$ and $\Pr_\mu[Y=0] = \beta$. Observe that $X$ and $Y$ are not necessarily independent. We have

$$\begin{aligned}
\mathsf{PIC}_\mu(\pi^*) &= I_\mu(X;\Pi^* \mid Y) + I_\mu(Y;\Pi^* \mid X) \\
&= H_\mu(X \mid Y) + [\alpha \cdot I_\mu(Y;\Pi^* \mid X=0) + (1-\alpha)\cdot I_\mu(Y;\Pi^* \mid X=1)]\ ,
\end{aligned}$$

where the second equality follows from $H_\mu(X \mid Y\Pi^*)$=0, as the transcript of $\pi^*$ fully determines $X$. Observe that when $X=0$, Alice doesn't learn from $\Pi^*$ anything about $Y$, while when $X=1$, Alice learns from $\Pi^*$ the value of $Y$. Thus

$$\mathsf{PIC}_\mu(\pi^*) = H_\mu(X \mid Y) + (1-\alpha)\cdot H_\mu(Y \mid X=1)\ .$$

Now define another input distribution $\mu'$ such that: $X$ and $Y$ are independent; $X \sim \mathbf{Ber}(\alpha, 1-\alpha)$; $\Pr[Y=1] = \Pr_\mu[Y=1 \mid X=1]$. Note that $H_{\mu'}(X \mid Y) = H_{\mu'}(X) = H_\mu(X)$ and that $H_{\mu'}(Y \mid X = 1) = H_{\mu'}(Y) = H_\mu(Y \mid X=1)$. We thus have that

$$\begin{aligned}
\mathsf{PIC}_{\mu'}(\pi^*) &= I_{\mu'}(X;\Pi^* \mid Y) + I_{\mu'}(Y;\Pi^* \mid X) \\
&= H_{\mu'}(X \mid Y) + (1-\alpha)\cdot H_{\mu'}(Y \mid X=1) \\
&= H_\mu(X) + (1-\alpha)\cdot H_\mu(Y \mid X=1) \\
&\ge H_\mu(X \mid Y) + (1-\alpha)\cdot H_\mu(Y \mid X=1) \\
&= \mathsf{PIC}_\mu(\pi^*)\ .
\end{aligned}$$

[4] We can assume this w.l.o.g. because for any protocol $\pi$, $\sup_\mu \mathsf{PIC}_\mu(\pi) = \sup_\mu \mathsf{PIC}_\mu(\pi')$, where $\pi'$ is the protocol symmetric to $\pi$. This is because any input distribution $\mu$ has a symmetric one, $\mu'$.

Therefore, to find $\sup_\mu \mathsf{PIC}_\mu(\pi^*)$ we can consider only input distributions $\mu'$ such that $X$ and $Y$ are independent. For such $\mu'$ we define $\alpha'$ and $\beta'$ such that $X \sim \mathbf{Ber}(\alpha', 1-\alpha')$ and $Y \sim \mathbf{Ber}(\beta', 1-\beta')$. We have

$$\mathsf{PIC}_{\mu'}(\pi^*) = H_{\mu'}(X) + (1-\alpha')H_{\mu'}(Y) \ .$$

Thus, for any $\alpha'$, $\mathsf{PIC}_{\mu'}(\pi^*)$ is maximized when $H_{\mu'}(Y) = 1$, i.e., when $\beta' = \frac{1}{2}$. In that case we have $\mathsf{PIC}_{\mu'}(\pi^*) = H_{\mu'}(X) + (1-\alpha')$. Thus, to find $\sup_\mu \mathsf{PIC}_\mu(\pi^*)$, we study the function $f : [0,1] \to \mathbb{R}$, defined as $f(\alpha') = -\alpha' \log(\alpha') + (\alpha'-1)\log(1-\alpha') + 1 - \alpha'$. [5]

Now, $f$ is continuous on $[0,1]$ and differentiable on $(0,1)$. For $0 < \alpha < 1$, we have: (1) $f'(\alpha') = -\log(\alpha') - 1 + \log(1-\alpha') + 1 - 1 = \log(\frac{1}{\alpha'} - 1) - 1$; (2) $f'$ is continuous and decreasing on $(0,1)$; and (3) $f'$ admits the unique root $\frac{1}{3}$. Thus, $f$ is maximized for $\alpha' = \frac{1}{3}$, its maximum value being $f(\frac{1}{3}) = \log(3)$.

We thus have that $\sup_\mu \mathsf{PIC}_\mu(\pi^*) = \mathsf{PIC}_{\mu^*}(\pi^*)$ for $\mu^*$ defined as follows: $X$ and $Y$ are independent; $X \sim \mathbf{Ber}(\frac{1}{3}, \frac{2}{3})$; $Y \sim \mathbf{Ber}(\frac{1}{2}, \frac{1}{2})$, that $\mathsf{PIC}_{\mu^*}(\pi^*) = \log(3)$, and that $\mathsf{PIC}(\mathsf{AND}) = \log(3) \simeq 1.58$. □

# 5 Private computation, randomness, and PIC

We have seen that the public information cost of a function is equal to the information cost of the function when we only consider public coin protocols, and that in order to decrease the information cost even further, the players must use private randomness. We will see now that the difference between the public information cost of a protocol and its information cost can provide a lower bound on the amount of private randomness the players use during the protocol. The entropy of the transcript of the protocol, conditioned on the inputs and the public coins, is defined as $H(\Pi \mid XR^p)$. Once the input and the public coins are fixed, the entropy of the transcript of the protocol comes solely from the private randomness. Thus the entropy of the transcript of the protocol provides a lower bound on the entropy of the private randomness used by the players.

**Theorem 5.1.** *Let $f = (f_i)$ be a family of functions of $k$ variables. Let $\pi$ be a protocol for $f$. For any input distribution $\mu$, it holds:*

$$H_\mu(\Pi \mid XR^p) \geq \frac{\mathsf{PIC}_\mu(\pi) - \mathsf{IC}_\mu(\pi)}{k} \ .$$

*Thus, running a protocol for $f$ with information cost $I_\mu$ requires entropy*

$$H_\mu(\Pi \mid XR^p) \geq \frac{\mathsf{PIC}_\mu(f) - I_\mu}{k} \ .$$

*Proof.* We assume in what follows the input distribution $\mu$ without explicitly denoting it.

Define $Q_i$ as

$$Q_i = I(X_{-i}; R_{-i} \mid X_i R_i \Pi_i R^p) \ .$$

By Proposition 4.2 we have,

$$\begin{aligned} \mathsf{PIC}(\pi) &= \mathsf{IC}(\pi) + \sum_{i=1}^{k} I(X_{-i}; R_{-i} \mid X_i R_i R^p \Pi_i) \\ &= \mathsf{IC}(\pi) + \sum_{i=1}^{k} Q_i \ . \end{aligned}$$

[5] We denote here by log the logarithm base 2.

Now,

$$\begin{aligned} Q_i &= I(X_{-i}; R_{-i} \mid X_i R_i \Pi_i R^p) \\ &= I(X_{-i}\Pi_i; R_{-i} \mid X_i R_i R^p) - I(\Pi_i; R_{-i} \mid X_i R_i R^p) \quad \text{(chain rule)} \\ &\leq I(X_{-i}\Pi_i; R_{-i} \mid X_i R_i R^p) \\ &= I(X_{-i}; R_{-i} \mid X_i R_i R^p) + I(\Pi_i; R_{-i} \mid X_i R_i X_{-i} R^p) \quad \text{(chain rule)} \\ &= I(\Pi_i; R_{-i} \mid X R_i R^p) \\ &= H(\Pi_i \mid X R_i R^p) \\ &\leq H(\Pi \mid X R^p) \;. \end{aligned}$$

Thus,

$$\mathsf{PIC}(\pi) \leq \mathsf{IC}(\pi) + k \cdot H(\Pi \mid X R^p) \;.$$

□

Using Lemma 3.10, we can give a lower bound on the randomness required to run a private protocol.

**Corollary 5.2.** *Let $f = (f_i)$ be a family of functions of $k$ variables. Let $\pi$ be a $k$-party private protocol for $f$. For any distribution $\mu$ on inputs,*

$$H_\mu(\Pi \mid X R^p) \geq \frac{1}{k} \cdot \left( \mathsf{PIC}_\mu(f) - \sum_{i=1}^{k} H_\mu(f_i) \right) \;.$$

# 6 Tight lower bounds for the parity function Par

We now show how one can indeed use PIC to study multi-party communication protocols and to prove tight bounds. We study one of the canonical problems for zero-error multi-party computation, the parity function. The $k$-party parity problem with $n$-bit inputs $\mathsf{Par}_k^n$ is defined as follows. Each player $i$ receives $n$ bits $(x_i^p)_{p \in [\![1,n]\!]}$ and Player 1 has to output the bitwise XOR of the inputs $\left( \bigoplus_{i=1}^{k} x_i^1, \bigoplus_{i=1}^{k} x_i^2, \ldots, \bigoplus_{i=1}^{k} x_i^n \right)$. We give a lower bound on $\mathsf{Par}_k^n$ and then use it to prove tight lower bounds on the randomness complexity of private computations of $\mathsf{Par}_k^n$.

There is a simple private protocol for $\mathsf{Par}_k^n$ that uses $n$ bits of private randomness. Player 1 uses a private random $n$-bit string $r$ and sends to Player 2 the string $x_1 \oplus r$. Then, Player 2 computes the bit-wise parity of its input with that message and sends $x_2 \oplus x_1 \oplus r$ to Player 3. The players continue until Player 1 receives back the message $x_k \oplus \ldots \oplus x_1 \oplus r$. Player 1 then takes the bit-wise parity of this message with the private string $r$ to compute the value of the parity function. It is easy to see that this protocol has information cost equal to $n$, since Player 1 just learns the value of the function and all other players learn nothing. We thus see that information cost (IC) cannot provide here lower bounds that scale with $k$.

We note that we prove our lower bound for $\mathsf{Par}_k^n$ for a wider class of protocols, where we allow the player outputting $\oplus_{i=1}^{k} x_i^p$ to be different for each coordinate $p$ and where the identity of that player may depend on the input. On the other hand, we prove our lower bound for the restricted class of 0-error oblivious protocols. We now prove a tight lower bound of $\Omega(nk)$ on the PIC of $\mathsf{Par}_k^n$ (for 0-error oblivious protocols) which can then be used to derive other lower bounds for protocols for $\mathsf{Par}_k^n$.

For the purpose of the proof we define a (natural) full order on the messages of an oblivious protocol. The order is defined as follows. We define an ordered series of *lots of messages*. In each lot there is at most one message on any of the $k(k-1)$ directional links. The order of the messages is defined by the order of the lots, and within each lot, the messages are ordered by, say, the lexicographical order of the links on which they are sent. The messages are assigned to lots as follows: The first lot consists of all messages sent by all the players in their first respective local round. The messages assigned to lot $s \geq 1$ are defined inductively after lots $s' < s$ have been defined. To define the messages of lot $s > 1$, we proceed as follows for each player $i$: run the protocol $\pi$, and whenever player $i$ is waiting for a message, extract a message from the already defined lots (lots $s' < s$), if such message is assigned to one of them. Continue until a needed message is not available (i.e., the protocol "gets stuck"), or after player $i$ sends, according to the protocol, a message not already assigned to a lot $s' < s$. In the latter case, assign to lot $s$ all the messages sent by player $i$ in the same local round (i.e., for any player $i$ and local round $r$, all messages sent by player $i$ in local round $r$ are in the same lot).

To see that all the messages of the protocol are assigned to lots, build the following graph where each node is identified by a pair $(i, r)$, for a player $i$ and local round $r$ of player $i$. There is a directed edge from any node $(i, r')$ to node $(i, r)$, if $r' < r$ and there is at least one message sent by player $i$ in round $r'$. Further, there is a directed edge from node $(j, r')$ to node $(i, r)$ if there is an integer $\ell$ such that the $\ell$'th message from player $j$ to player $i$ is sent by player $j$ in its local round $r'$ and read by player $i$ in its local round $r$. Observe that a node $(i, r)$ is not on a directed cycle if and only if, when the protocol is run, player $i$ reaches the sending-of-messages phase of its local round $r$. Define a partial order on the nodes which are not on a directed cycle, according to the orientation of the edges. We define the "level" of a node to be the length of the longest directed path leading to it. Observe that by induction on this level, the messages sent by player $i$ in local round $r$, where node $(i, r)$ is of level $s$, are assigned to lot number $s$.

Observe that the enumeration of the messages as defined above respects the intuitive "temporal causality" of the messages of the protocol. More formally, the following two properties hold for the above defined order: (1) the relative order of the local rounds of two messages that are both sent from player, say, $i$, to player, say, $j$, is the same as the relative order of these messages according to the global order, and (2) the value of a message number $\ell$ (in the global order) sent from player $i$ is fully determined by the input to player $i$ and the values of the messages with indices less than $\ell$ that are received by player $i$.

Denote by $(M_i^{\overrightarrow{l}})_{l \geq 0}$ the ordered sequence of all messages sent by player $i$ in the protocol $\pi$, ordered according to the enumeration defined above. Similarly, denote by $(M_i^{\overleftarrow{l}})_{l \geq 0}$ the ordered sequence of messages received by player $i$. Denote by $j(i,l)$ the player receiving message $M_i^{\overrightarrow{l}}$, and by $l'(i,l)$ the integer such that the random variable $M_{j(i,l)}^{\overleftarrow{l'(i,l)}}$ and the random variable $M_i^{\overrightarrow{l}}$ represent the same message. Observe that since we consider here an oblivious protocol, the functions $j(i,l)$ and $l'(i,l)$ are well defined. For any $l_0$, let $M_i^{\overrightarrow{<l_0}}$ be the random variable representing the so-far history of player $i$, i.e., all the messages to and from player $i$ which appear before message $M_i^{\overrightarrow{l_0}}$ in the enumeration of messages defined above. In a similar way, define $M_i^{\overleftarrow{<l_0}}$ to be the random variable representing the so-far history of the messages to and from player $i$ which appear before message $M_i^{\overleftarrow{l_0}}$.

**Theorem 6.1.** *For oblivious protocols,* $\mathsf{PIC}_\mu^0(\mathsf{Par}_k^n) \geq n(k-1)$ *where* $\mu$ *is the uniform input distribution.*

*Proof.* Throughout the proof we consider the uniform input distribution $\mu$ without explicitly stating it. Since we are looking at 0-error protocols, the public information cost is equal to the information cost of deterministic protocols. Let $\pi$ be a 0-error deterministic protocol for $\mathsf{Par}_k^n$ for $k$ players and $n$-bit input per player.

We first prove that

$$\mathsf{PIC}^0(\pi) \geq \sum_{i=1}^{k} I(X_i; \overleftrightarrow{\Pi}_i) \; . \tag{1}$$

Intuitively, this means that $\mathsf{PIC}$ is at least the sum over the players $i$ of the amount of information that player $i$ leaks about its input to some entity that has access to all messages to and from player $i$.

Since $\pi$ is a deterministic protocol we have $\mathsf{PIC}^0(\pi) = \sum_{j=1}^{k} I(X_{-j}; \Pi_j \mid X_j)$. We will therefore show that $\sum_{j=1}^{k} I(X_{-j}; \Pi_j \mid X_j) \geq \sum_{i=1}^{k} I(X_i; \overleftrightarrow{\Pi}_i)$.

Using the chain rule, we decompose $\sum_{j=1}^{k} I(X_{-j}; \Pi_j \mid X_j)$ into a sum over all messages received in the protocol:

$$\begin{aligned} \sum_{j=1}^{k} I(X_{-j}; \Pi_j \mid X_j) &= \sum_{j=1}^{k} \sum_{\ell' \geq 0} I(X_{-j}; M_j^{\overleftarrow{\ell'}} \mid M_j^{\overleftarrow{0}} \dots M_j^{\overleftarrow{\ell'}-1} X_j) \\ &= \sum_{j=1}^{k} \sum_{\ell' \geq 0} I(X_{-j}; M_j^{\overleftarrow{\ell'}} \mid M_j^{<\overleftarrow{\ell'}} X_j) \; . \end{aligned}$$

We now consider each message from the point of view of the receiver rather than that of the sender. Recall that each message in the protocol is represented by two random variables: for any $i$ and $l$ the two random variables $M_i^{\overrightarrow{l}}$ and $M_{j(i,l)}^{\overleftarrow{l'}(i,l)}$ represent the same message. Thus, we can rearrange the last summation, using $j$ as a shorthand for $j(i,l)$ and $l'$ as a shorthand of $l'(i,l)$, and get

$$\sum_{j=1}^{k} I(X_{-j}; \Pi_j \mid X_j) = \sum_{i=1}^{k} \sum_{\ell \geq 0} I(X_{-j}; M_i^{\overrightarrow{\ell}} \mid M_j^{<\overleftarrow{\ell'}} X_j) \; .$$

Note that using the chain rule, we have for all $i \in [\![1,k]\!]$,

$$\begin{aligned} I(X_i; \overleftrightarrow{\Pi}_i) &= \sum_{\ell} I(X_i; M_i^{\overrightarrow{\ell}} \mid M_i^{<\overrightarrow{\ell}}) + \sum_{\ell} I(X_i; M_i^{\overleftarrow{\ell}} \mid M_i^{<\overleftarrow{\ell}}) \\ &= \sum_{\ell} I(X_i; M_i^{\overrightarrow{\ell}} \mid M_i^{<\overrightarrow{\ell}}) \; , \end{aligned}$$

where we used the fact that every term of the second sum is $0$. This is true using Proposition 2.4, which can we used since, for any $\ell$, conditioned on $M_i^{<\overleftarrow{\ell}}$, $X_i$ is independent of the variable $M_i^{\overleftarrow{\ell}}$ (intuitively: when the input distribution is a product distribution, the incoming messages to a player do not carry any information on the input of that player).

Therefore, our objective now is to show that for any message $M_i^{\overrightarrow{\ell}}$,

$$I(X_{-j}; M_i^{\overrightarrow{\ell}} \mid M_j^{<\overleftarrow{\ell'}} X_j) \geq I(X_i; M_i^{\overrightarrow{\ell}} \mid M_i^{<\overrightarrow{\ell}}) \; . \tag{2}$$

Since $M_i^{\overrightarrow{l}}$ is determined by $(X_i, M_i^{<\overrightarrow{l}})$, we have that $H(M_i^{\overrightarrow{l}} \mid X_i M_i^{<\overrightarrow{l}}) = 0$, and $I(X_i; M_i^{\overrightarrow{l}} \mid M_i^{<\overrightarrow{l}}) = H(M_i^{\overrightarrow{l}} \mid M_i^{<\overrightarrow{l}})$. Similarly (as $X_i$ is trivially a function of $X_{-j}$), $I(X_{-j}; M_i^{\overrightarrow{l}} \mid M_j^{<\overleftarrow{l'}} X_j) = H(M_i^{\overrightarrow{l}} \mid M_j^{<\overleftarrow{l'}} X_j)$.

Thus,

$$I(X_i; M_i^{\overrightarrow{\ell}} \mid M_i^{\overrightarrow{<\ell}}) \le I(X_{-j}; M_i^{\overrightarrow{\ell}} \mid M_j^{\overleftarrow{<\ell'}} X_j)$$
$$\Updownarrow$$
$$H(M_i^{\overrightarrow{\ell}} \mid M_i^{\overrightarrow{<\ell}}) \le H(M_i^{\overrightarrow{\ell}} \mid M_j^{\overleftarrow{<\ell'}} X_j)$$
$$\Updownarrow$$
$$I(M_i^{\overrightarrow{\ell}}; M_i^{\overrightarrow{<\ell}}) \ge I(M_i^{\overrightarrow{\ell}}; M_j^{\overleftarrow{<\ell'}} X_j) \ .$$

The last inequality holds if $I(M_i^{\overrightarrow{\ell}}; M_i^{\overrightarrow{<\ell}}) = I(M_i^{\overrightarrow{\ell}}; M_i^{\overrightarrow{<\ell}} M_j^{\overleftarrow{<\ell'}} X_j)$, which itself holds if

$$I(M_i^{\overrightarrow{\ell}}; M_j^{\overleftarrow{<\ell'}} X_j \mid M_i^{\overrightarrow{<\ell}}) = 0 \ . \tag{3}$$

Observe that given $M_i^{\overrightarrow{<\ell}}$, $M_i^{\overrightarrow{\ell}}$ is fixed by $X_i$, and therefore by the data processing inequality

$$I(X_i; M_j^{\overleftarrow{<\ell'}} X_j \mid M_i^{\overrightarrow{<\ell}}) \ge I(M_i^{\overrightarrow{\ell}}; M_j^{\overleftarrow{<\ell'}} X_j \mid M_i^{\overrightarrow{<\ell}}) \ ,$$

and thus Equality (3) holds if

$$I(X_i; M_j^{\overleftarrow{<\ell'}} X_j \mid M_i^{\overrightarrow{<\ell}}) = 0 \ . \tag{4}$$

Observe now that the ordering of the messages that we defined implies that $M_j^{\overleftarrow{<\ell'}}$ (which is the same message as $M_i^{\overrightarrow{\ell}}$) is determined by $(X_{-i}, M_i^{\overrightarrow{<\ell}})$. Furthermore, $X_j$ is trivially determined by $X_{-i}$. Using the data processing inequality we thus have

$$I(X_i; M_j^{\overleftarrow{<\ell'}} X_j \mid M_i^{\overrightarrow{<\ell}}) \le I(X_i; X_{-i} M_i^{\overrightarrow{<\ell}} \mid M_i^{\overrightarrow{<\ell}}) = I(X_i; X_{-i} \mid M_i^{\overrightarrow{<\ell}}) \ .$$

Thus Equality (4) holds if $I(X_i; X_{-i} \mid M_i^{\overrightarrow{<\ell}}) = 0$. To prove the latter, denote by $(B^d)_{d>0}$ all the messages in $M_i^{\overrightarrow{<\ell}}$, ordered by local rounds of player $i$, and inside each round having first the messages sent by player $i$, ordered by the index of the recipient, and then the messages received by player $i$, ordered by the index of the sender. For convenience of notation we also define the message $B^0$, which is the "empty message" at the beginning of $M_i^{\overrightarrow{<\ell}}$.

We prove, by induction on $d$, that for any $d \ge 0$, $I(X_i; X_{-i} \mid B^0 B^1 \dots B^d) = 0$. For the base of the induction ($d = 0$) we have $I(X_i; X_{-i} \mid B^0) = I(X_i; X_{-i}) = 0$, since $X$ is distributed according to $\mu$.

By the induction hypothesis, for some $d \ge 0$, $I(X_i; X_{-i} \mid B^0 \dots B^d) = 0$. If the message $B^{d+1}$ is *sent* by player $i$, then $B^{d+1}$ is a function of $X_i$ and $B^0 \dots B^d$, and thus

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

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

$$\begin{aligned} I(X_i; X_{-i} \mid B^0 \dots B^{d+1}) &= H(X_i \mid B^0 \dots B^{d+1}) \; - H(X_i \mid B^0 \dots B^{d+1} X_{-i}) \\ &\leq H(X_i \mid B^0 \dots B^d) \; - H(X_i \mid B^0 \dots B^d X_{-i}) \\ &= I(X_i; X_{-i} \mid B^0 \dots B^d) \\ &= 0 \, . \end{aligned}$$

We thus have that $I(X_i; X_{-i} \mid M_i^{<\overrightarrow{\ell}}) = 0$ and the proofs of Inequality (4) and of Equality (3) are concluded. Inequality (2) and Inequality (1) then follow.

To conclude the proof of the theorem we now show that

$$\sum_{i=1}^{k} I(X_i; \overleftrightarrow{\Pi_i}) \geq n(k-1) \, . \tag{5}$$

Let $x = (x_i^p) \in \{0,1\}^{nk}$ be an arbitrary input, where $x_i$ is the $n$-bit input of player $i$. For any index $1 \leq p \leq n$, any player $1 \leq q \leq k$, we consider the question whether $H(\bigoplus_{i=1}^{k} x_i^p \mid X_q = x_q, \Pi_q = \pi_q(x)) = 0$. Intuitively, if this is the case then player $q$ can output $\bigoplus_{i=1}^{k} x_i^p$. Observe that since $\pi$ is a $0$-error protocol for $\mathsf{Par}_k^n$, then for each index $p$ there is at least one player $q$ such that $H(\bigoplus_{i=1}^{k} x_i^p \mid X_q = x_q, \Pi_q = \pi_q(x)) = 0$. Denote by $q^p(x)$ an arbitrary such player. For any player $i$, define $C_i(x) = \{p \mid q^p(x) \neq i\}$. Intuitively, when the input is $x$, then for each such coordinate, player $i$ has to leak its own input on that coordinate. Let $c_i(x) = |C_i(x)|$.

We now show that $\forall\, i,\; H(X_i \mid \overleftrightarrow{\Pi_i} = \overleftrightarrow{\pi_i}(x)) \leq n - c_i(x)$. Assume towards a contradiction that for some $i$, $H(X_i \mid \overleftrightarrow{\Pi_i} = \overleftrightarrow{\pi_i}(x)) > n - c_i(x)$. This implies that the number of possible values for $X_i$ consistent with $\overleftrightarrow{\Pi_i} = \overleftrightarrow{\pi_i}(x)$ is more than $2^{n-c_i(x)}$, and thus the number of coordinates of the input of the $i$-th player that are fixed by $\overleftrightarrow{\Pi_i} = \overleftrightarrow{\pi_i}(x)$ is strictly less than $c_i(x)$. In particular there exists an input $x'$ such that

- $\overleftrightarrow{\pi_i}(x') = \overleftrightarrow{\pi_i}(x)$, and
- $\exists\, p \in C_i(x)$ such that $x_i'^p \neq x_i^p$.

Observe that we consider here *oblivious* multi-party protocols. Therefore, $\overleftrightarrow{\pi_i}(x') = \overleftrightarrow{\pi_i}(x)$ implies that $\overleftrightarrow{\pi_i}(x) = \overleftrightarrow{\pi_i}(x_i', x_{-i})$ (by considering player $i$ as Alice, and all other players together as Bob, and using arguments as those used for a similar property for 2-party protocols). As $q^p(x) \neq i$, this is a contradiction, since if $\overleftrightarrow{\pi_i}(x) = \overleftrightarrow{\pi_i}(x_i', x_{-i})$ then (the output) $\bigoplus_{i=1}^{k} x_i^p$ is not fixed by $X_q = x_q$ and $\Pi_{q^p(x)} = \pi_{q^p(x)}(x)$, contradicting the definition of $q^p(x)$.

We now consider, for a given player $i$, the quantity $\mathop{\mathbb{E}}_x[c_i(x)]$. For any given $x$ and any given player $i$ we proved above that $H(X_i \mid \overleftrightarrow{\Pi_i} = \overleftrightarrow{\pi_i}(x)) \leq n - c_i(x)$. Thus, for any player $i$ we have $H(X_i \mid \overleftrightarrow{\Pi_i}) \leq \mathop{\mathbb{E}}_x[n - c_i(x)] = n - \mathop{\mathbb{E}}_x[c_i(x)]$. We get $I(X_i; \overleftrightarrow{\Pi_i}) \geq \mathop{\mathbb{E}}_x[c_i(x)]$.

Summing over all $i$, we get $\sum_{i=1}^{k} I(X_i; \overleftrightarrow{\Pi_i}) \geq \sum_{i=1}^{k} \mathbb{E}_x[c_i(x)] = \mathbb{E}_x[\sum_{i=1}^{k} c_i(x)]$ and since by simple counting, for any $x$, it holds that $\sum_{i=1}^{k} c_i(x) = n(k-1)$, we get $\sum_{i=1}^{k} I(X_i; \overleftrightarrow{\Pi_i}) \geq \mathbb{E}_x[n(k-1)] = n(k-1)$. This concludes the proof of Inequality (5).

Inequality (1) together with Inequality (5) conclude the proof of the theorem. □

**Theorem 6.2.** *The entropy in the private randomness of an oblivious private protocol for* $\mathsf{Par}_k^n$ *is at least* $\frac{k-2}{k} \cdot n$.

*Proof.* For $\mathsf{Par}_k^n$, where one player outputs the parity for each coordinate, we have $\sum_{i=1}^{k} H(f_i) = n$. Applying Corollary 5.2, we get: $H(\Pi \mid XR^p) \geq \frac{k-2}{k} \cdot n$.

□

We note that all private protocols considered in the literature are oblivious protocols. Observe also that using Theorem 5.1 one can also give a lower bound on the randomness needed by protocols that are allowed to leak a given limited amount of information about the inputs of the players.

# 7 A direct sum for PIC ?

The *direct sum* property is a fundamental question in complexity theory, and has been studied for many computation models. A direct sum theorem affirms that the amount of resources needed to perform $t$ independent tasks is at least the sum of the resources needed to perform each of the $t$ tasks. In this section we show that a certain direct sum property for PIC implies a certain direct sum property for CC. To this end, we prove a compression result by extending previous results [BBK$^+$16, Pan15] to the multi-party case. Note that information complexity (IC) has a direct sum property in the multi-party case. For PIC, it is easy to prove the following inequality.

**Theorem 7.1.** *For any $k$-variable functions $f$ and $g$, for any distribution $\mu$ on the inputs of $f$, for any distribution $\eta$ on the inputs of $g$, it holds that*

$$\mathsf{PIC}_{\mu\times\eta}(f \times g) \leq \mathsf{PIC}_{\mu}(f) + \mathsf{PIC}_{\eta}(g) .$$

We use here the notation $f \times g$ to indicate the task of computing $f$ with error $\epsilon$ and computing $g$ with error $\epsilon$ (as opposed to computing the couple function $(f, g)$ with error $\epsilon$). In order to understand whether the opposite inequality holds, i.e., whether a direct sum property holds for PIC, we first need to study the problem of compressing communication.

## 7.1 Relation between PIC and CC: A compression result

An important open question is how well can we compress the communication cost of an interactive protocol. Compression results have appeared in [BBCR13, BR14, BBK$^+$16, Pan15, BMY15], while, on the other hand, [GKR14, GKR15b, RS15, FJK$^+$16, GKR15a] focus on the hardness of compressing communication protocols. Here, we present a compression result with regards to the average-case communication complexity, distributional error, and the public information cost.

**Definition 7.2.** *Given an input distribution $\mu$, a protocol is said to compute a function with distributional error $\epsilon$ if the probability, over the input and the randomness of the protocol, that the protocol fails is at most $\epsilon$.*

**Definition 7.3.** *The average-case communication complexity of a protocol $\pi$ with respect to the input distribution $\mu$, denoted $\mathsf{ACC}_\mu(\pi)$, is the expected number of bits that are transmitted in an execution of $\pi$ for inputs distributed according to $\mu$ and for uniform randomness.*

**Theorem 7.4.** *Suppose there exists an oblivious protocol $\pi$ to compute a $k$-variable function $f$ over the distribution $\mu$ with distributional error probability $\epsilon$. Then for any fixed $\delta > 0$ there exists a public-coin protocol $\rho$ that computes $f$ over $\mu$ with distributional error $\epsilon + \delta$, and with average communication complexity*

$$\mathsf{ACC}_\mu(\rho) = \mathcal{O}\left(k^2 \cdot \mathsf{IC}_\mu(\pi) \cdot \log \frac{k^2 \cdot \mathsf{IC}_\mu(\pi) \cdot \mathsf{CC}(\pi)}{\delta}\right).$$

The proof of the above theorem will follow from extending, to the case of $k > 2$ players, the compression result presented in [BBK+16, Pan15], as stated below. Thus, the proof of Theorem 7.4 follows from Theorem 4.6 and from Theorem 7.5. We remark that it is an interesting question whether the $k^2$ factor is necessary or whether it can be replaced by smaller function of $k$.

**Theorem 7.5.** *Suppose there exists an oblivious public-coin protocol $\pi$ to compute a $k$-variable function $f$ over the distribution $\mu$ with distributional error probability $\epsilon$. Then for any fixed $\delta > 0$ there exists a public-coin protocol $\rho$ that computes $f$ over $\mu$ with distributional error $\epsilon + \delta$, and with average communication complexity*

$$\mathsf{ACC}_\mu(\rho) = \mathcal{O}\left(k^2 \cdot \mathsf{IC}_\mu(\pi) \cdot \log \frac{k^2 \cdot \mathsf{IC}_\mu(\pi) \cdot \mathsf{CC}(\pi)}{\delta}\right).$$

In the two-party compression scheme of [BBK+16, Pan15], the two players, given their corresponding inputs, try to guess the transcript $\pi(x_1, x_2)$ of the protocol $\pi$. For this, player 1 picks a candidate $t_1$ from the set $\mathrm{Im}(\pi(x_1, \cdot))$ of possible transcripts consistent with input $x_1$, while player 2 picks a candidate $t_2$ from the set $\mathrm{Im}(\pi(\cdot, x_2))$. The two players then communicate in order to find the first bit on which $t_1$ and $t_2$ disagree. The general structure of protocols ensures that the common prefix of $t_1$ and $t_2$ (until the first bit of disagreement) is identical to the beginning of the correct transcript on inputs $x_1$ and $x_2$, i.e., identical to $\pi(x_1, x_2)$. Starting from this correct prefix, the players then pick new candidates for the transcript of the protocol $\pi(x_1, x_2)$, and so on, until they agree on the full transcript $\pi(x_1, x_2)$. Clever choices of the candidates, along with an efficient technique to find the first bit which differs between the candidates, lead to a protocol with a small amount of communication.

In extending the proof in [BBK+16, Pan15] to the multi-party case new difficulties are encountered. The players can no longer try to guess the full transcript, as they have little information about the communication between the other players, and can only try to guess their partial transcripts, according to their own input. Then, in order to find the first disagreement in the global transcript, pairs of players need to find and communicate the place of the first disagreement between their respective partial transcripts.

For technical reasons, in this section we use the notation $\overleftrightarrow{\Pi}_i$ not as defined in Section 3 to denote the concatenation of $\Pi_i$ together with a similar string $\overrightarrow{\Pi_i}$ of the messages sent by player $i$. Rather, we define $\overleftrightarrow{\Pi}_i$ as a concatenation, local round of player $i$ by local round of player $i$, of, first, the messages sent by player $i$ and, then, the messages received by player $i$. Observe that since in this section we consider oblivious protocols there is a one-to-one correspondence between the transcripts of player $i$, $\overleftrightarrow{\Pi}_i$, according to the two definitions.

Following [BBK$^+$16], in the definition of our protocol we will use a two-party "device" as a black box, call it the *lcp box* (for *longest common prefix*), which can be used by two players $A$ and $B$ in the following way: $A$ inputs a string $x$, $B$ inputs a string $y$, and the box returns the first index $j$ such that $x_j \neq y_j$, if $x \neq y$, or returns that $x = y$, otherwise. The conceptual device is assumed to operate with 0 communication complexity.

This black box device can be efficiently simulated if we allow error:

**Lemma 7.6** ([FRPU94])**.** *For any $\epsilon > 0$, there exists a randomized public coin protocol, such that on input two $n$-bits strings $x$ and $y$, it outputs the first index $j$ such that $x_j \neq y_j$ with probability at least $1 - \epsilon$, if such $j$ exists, and otherwise outputs that the two strings are equal. The communication complexity of this protocol is $\mathcal{O}(\log(n/\epsilon))$.*

We note that this simulation can easily be extended to the case when the two input strings are not of the same length, by first communicating the two lengths, and continuing only if they are equal. This leaves the communication complexity of the simulation protocol $O(\log(n/\epsilon))$ where $n = \max(|x|, |y|)$.

We will use the following lemma. This lemma, and its proof, are implicit in [BBK$^+$16]. We give here the proof for completeness.

**Lemma 7.7** ([BBK$^+$16])**.** *For every input distribution $\mu$, and every positive error probability $\delta$, any protocol $\tilde{\rho}$ that uses the lcp box $\ell$ times on average (on the input distribution $\mu$ and the internal randomness of $\tilde{\rho}$) on strings of length at most $C$, can be simulated with error $\delta$ by a protocol $\rho$ that does not use an lcp box and communicates on average $O(\ell \log(\frac{\ell C}{\delta}))$ bits more than $\tilde{\rho}$.*

*Proof.* The protocol $\rho$ simulates $\tilde{\rho}$ by replacing each use of the lcp box with the protocol given by Lemma 7.6, with error $\epsilon$, $\epsilon$ to be defined later.

Since each call to that protocol fails with probability at most $\epsilon$, the (distributional) error introduced by the use of the simulation protocol instead of the lcp box is at most $\epsilon\ell$. We thus take $\epsilon = \delta/\ell$ and get that the simulation fails with (distributional) probability $\delta$.

By Lemma 7.6 each call to the protocol simulatimg the lcp box has communication complexity $O(\log(C/\epsilon))$. We get that on average $\rho$ sends $O(\ell \log(C/\epsilon)) = O(\ell \log(\frac{\ell C}{\delta}))$ bits more than $\tilde{\rho}$. □

We use the lcp box in the definition of the protocols in our proof, and then use Lemma 7.7 to obtain our final result at the end.

*Proof of theorem 7.5.* Fix the public randomness to be $r$. For each $i$, define the set $\mathcal{X}_i$ to be the set of possible inputs of player $i$, and the set $\Pi_{(i)}(x_i)$ to be the set of possible transcripts of player $i$, given that player $i$ has input $x_i$ (and the public randomness is $r$):

$$\Pi_{(i)}(x_i) = \overleftrightarrow{\pi_i}(\mathcal{X}_1, \ldots, \mathcal{X}_{i-1}, x_i, \mathcal{X}_{i+1}, \ldots, \mathcal{X}_k, r) .$$

The messages being self-delimiting and the protocol $\pi$ being oblivious, $\Pi_{(i)}(x_i)$ is naturally defined as a set of binary strings.

Each player $i$ can now represent $\Pi_{(i)}(x_i)$ by a binary tree $T_i$ as follows. We note that actually computing $T_i$ takes exponential time. However, we are concerned with the communication complexity of the protocol and not by its computational complexity.

1. The root is the largest common prefix (lcp) of the transcripts in $\Pi_{(i)}(x_i)$, and the remaining nodes are defined inductively.

2. For node $\tau$, we have

   - the first child of $\tau$ is the lcp of the transcripts in $\Pi_{(i)}(x_i)$ beginning with $\tau \circ 0$, i.e., $\tau$ concatenated with the bit $0$.
   - the second child of $\tau$ is the lcp of the transcripts in $\Pi_{(i)}(x_i)$ beginning with $\tau \circ 1$.

3. The leaves are labelled by the possible transcripts of player $i$, i.e., the elements of $\Pi_{(i)}(x_i)$.

We define the *weight* of a leaf $f$ with label $t_i$ to be

$$w(t_i) = \Pr_{(X_j)_{j \neq i} | X_i = x_i} \left[\overleftrightarrow{\pi}_i(X_1, \ldots, X_{i-1}, x_i, X_{i+1}, \ldots, X_k, r) = t_i\right].$$

The weight of a non-leaf node is defined by induction as the sum of the weights of its children. By construction, the weight of the root is $1$.

We say that $(t_1, \ldots, t_k) \in \Pi_{(1)}(x_1) \times \ldots \times \Pi_{(k)}(x_k)$ is a *coherent profile* if every message from $i$ to $j$ appears with the same content in $t_i$ and $t_j$. In fact, given $(x_1, \ldots, x_k)$, the profile $(\overleftrightarrow{\pi}_1(x_1, \ldots, x_k, r), \ldots, \overleftrightarrow{\pi}_k(x_1, \ldots, x_k, r))$ is the only coherent profile. Assume towards a contradiction that there are two distinct coherent profiles, given $(x_1, \ldots, x_k)$. Each coherent profile gives rise to a transcript of the protocol. Let $m$ be the first message, according to the global order of messages of an oblivious protocol as defined in Section 6, which is different in these two transcripts. But, each message sent from player $i$ to player $j$ is fully determined by the input $x_i$ and the previous messages according to that order (and the shared randomness), and thus $m$ cannot differ in the two transcripts.

We now define the protocol $\tilde{\rho}$ which allows the players to collaborate and efficiently find this coherent profile, i.e., protocol $\tilde{\rho}$ allows each player $i$ to find $\overleftrightarrow{\pi}_i(x_1, \ldots, x_k, r)$.
The players proceed in stages $s = 1, 2 \ldots$. We will have the invariant that at the beginning of any stage $s$, each player $i$ is at a node $\tau_i(s)$ of its transcript tree $T_i$, such that $(\tau_1(s), \ldots, \tau_k(s))$ is a (term-wise) prefix of $(\overleftrightarrow{\pi}_1(x_1, \ldots, x_k, r), \ldots, \overleftrightarrow{\pi}_k(x_1, \ldots, x_k, r))$. At any time, given $\tau_i(s)$, for any $i$ and $s$, player $i$ furthermore has a candidate leaf $t_i(s)$ in the tree $T_i$ (representing a candidate for its transcript), defined as follows: player $i$ defines $\tau^1 = \tau_i(s)$, and then defines inductively $\tau^{j+1}$ to be the child of $\tau^j$ which has higher weight (breaking ties arbitrarily), until it reaches a leaf: this is the candidate $t_i(s)$. Observe that $t_i(s)$ is a descendent of $\tau_i(s)$ in $T_i$ [6] and that $t_i(s)$ corresponds to the transcript with highest probability conditioned on that the prefix of the transcript is the string corresponding to $\tau_i(s)$.

At the beginning, each player $i$ starts the protocol being at the node $\tau_i(1)$, which is the root of the tree $T_i$, and the invariant above clearly holds. For each stage $s$ the players proceed as follows:

1. Each pair of players $(i, j)$ uses an lcp box to find the first occurrence where the transcript between $i$ and $j$ in $t_i(s)$ is not coherent with the transcript between $i$ and $j$ in $t_j(s)$. Let $q_{i,j}$ be the index of the message that includes this first occurrence, where the messages are numbered according to the global order of all messages of an oblivious protocol as defined in Section 6, and $\infty$ if no such occurrence was found. Let $Q_i = \min_j\{q_{i,j}\}$. Observe that if for all pairs of players there is no such occurrence (i.e., $Q_i = \infty$ for all $i$), it means that $(t_1(s), \ldots, t_k(s))$ is a coherent profile, each player $i$ has found $\overleftrightarrow{\pi}_i(x_1, \ldots, x_k, r)$.

2. Each player $i$ now broadcasts $Q_i$. Each player can then find $Q = \min_i\{Q_i\}$. If $Q = \infty$, i.e., no pairwise inconsistency has been found between any two nodes, the protocol terminates and $(t_1(s), \ldots, t_k(s))$ is found as the coherent profile.

[6] We define here a node to be a descendent of itself.

3. Let $(i, j)$ be the pair of players such that $Q = q_{i,j}$. The player who has the sender role of message number $Q$ is considered "correct". Let this player be player $j$ and the player receiving the message, player $i$. Player $i$ sets its $\tau_i(s+1)$: in $T_i$, starting from $t_i(s)$, it goes up the tree toward $\tau_i(s)$, until it reaches a node $\hat{\tau}_i$ which is correct (according to the result of the lcp box). Then, it defines $\tau_i(s+1)$ as the child of $\hat{\tau}_i$ which is not on the path from $\hat{\tau}_i$ to $t_i(s)$.

4. Any other player $j \neq i$ defines $\tau_j(s+1) = \tau_j(s)$.

We now claim by induction on the stages that the invariant stated above is preserved for all players at all times. It clearly holds at the beginning. We claim that if it holds after stage $s$ then it also holds after stage $s+1$. For the $k-1$ players which define $\tau_j(s+1) = \tau_j(s)$ it clearly continues to hold. For the single player, say player $i$, which defines a new node as $\tau_i(s+1)$ in Step (3) we proceed as follows.

We first claim, by induction on the index of the messages in the global order, that for all messages with index $\ell < Q$, where message $\ell$ is sent from player $j$ to player $i$, it holds that the value of message number $\ell$ is the same in the coherent profile $(\overleftrightarrow{\pi_1}(x_1, \ldots, x_k, r), \ldots, \overleftrightarrow{\pi_k}(x_1, \ldots, x_k, r))$ and in both $t_i(s+1)$ and $t_j(s+1)$. The basis of the induction ($\ell = 0$) clearly holds. The inductive step follows from observing that message $\ell$ is fully determined by the input to player $j$ and the messages that appear before message $\ell$ in $\overleftrightarrow{\pi_j}$. Thus, by the induction hypothesis the value of message $\ell$ in $t_j(s+1)$ is as it appears in the coherent profile $(\overleftrightarrow{\pi_1}(x_1, \ldots, x_k, r), \ldots, \overleftrightarrow{\pi_k}(x_1, \ldots, x_k, r))$. It follows from the definition of $Q$ that the value of message $\ell$ is the same in $t_i(s+1)$ and $t_j(s+1)$.

For message $Q$, we have by similar arguments that its value according to $t_j(s+1)$ is consistent with $\overleftrightarrow{\pi_i}(x_1, \ldots, x_k, r)$. The prefix of message $Q$ as appears in the path from the root of $T_i$ and delimited by $\tau_i(s+1)$ is consistent with $t_j(s+1)$ by the choice of $\tau_i(s+1)$ in Step (3).

Now, since the relative order of messages in a transcript $\overleftrightarrow{\Pi}_i$ and in the global order is the same, it follows that $\tau_i(s+1)$ represents a prefix of $\overleftrightarrow{\pi_i}(x_1, \ldots, x_k, r)$, as required.

We now show that for player $i$ which is the (single) player that sets its $\tau_i(s+1)$ in Step (3) (i.e., the single player that changes its $\tau$ node and its guess of the transcript), $w(\tau_i(s+1)) \leq \frac{1}{2}w(\tau_i(s))$. We look at the sequence $(\tau^j)$ defined by player $i$ when defining its candidate leaf $t_i(s)$ as a function of $\tau_i(s)$. Let $\tau^j$ be the first common ancestor of $t_i(s)$ and $\tau_i(s+1)$. By construction, $\tau_i(s+1)$ is a child of $\tau^j$, and $t_i(s)$ is a descendant of the other child of $\tau^j$. By the candidate leaf's construction process, $w(\tau_i(s+1)) \leq \frac{1}{2}w(\tau^j) \leq \frac{1}{2}w(\tau_i(s))$.

We conclude the analysis. On inputs $(x_1, \ldots, x_k)$, let $(t_1, \ldots, t_k)$ denote the coherent profile. First note that with each stage the depth of one of the nodes $\tau_i$ increases. We proved that at any time $(\tau_1, \ldots, \tau_k)$ is a term-wise prefix of $(t_1, \ldots, t_k)$. Thus (unless $\mathsf{CC}(\pi)$ is not finite, in which case the theorem trivially holds), the protocol terminates in finite time, with the "candidate" profile $(t_1, \ldots, t_k)$. To give an upper bound on the number of stages until this happens, observe that each player will set its $\tau_i$ in Step (3) (i.e., will change its $\tau_i$) at most $\log \frac{1}{w(t_i)}$ times, because the weight of the node $\tau_i$ at least halves with each such change (recall that the root has weight 1). Since in each stage there is exactly one player that changes its $\tau_i$, the total number of stages, $S$, is bounded from above by $\sum_{i=1}^{k} \log \frac{1}{w(t_i)}$. We now take the average over inputs and over the shared randomness:

$$
\begin{aligned}
\mathop{\mathbb{E}}_{r,x}[S] &\leq \mathop{\mathbb{E}}_{r,x} \sum_{i=1}^{k} \log \frac{1}{w(t_i)} \\
&= \sum_{i=1}^{k} \mathop{\mathbb{E}}_{r,x_i} \left[ \mathop{\mathbb{E}}_{(x_j)_{j\neq i} | X_i = x_i} \left[ \log \frac{1}{w(t_i)} \right] \right] \\
&= \sum_{i=1}^{k} \mathop{\mathbb{E}}_{r,x_i} \left[ \mathop{\mathbb{E}}_{(x_j)_{j\neq i} | X_i = x_i} \left[ \log \frac{1}{\Pr\limits_{(X_j)_{j\neq i} | X_i = x_i} [\overleftrightarrow{\pi_i}(X_1, \ldots, X_{i-1}, x_i, X_{i+1}, \ldots, X_k, r) = \overleftrightarrow{\pi_i}(x_1, \ldots, x_k, r)]} \right] \right] \\
&= \sum_{i=1}^{k} \mathop{\mathbb{E}}_{r,x_i} \left[ \mathop{\mathbb{E}}_{t_i | X_i = x_i, R = r} \left[ \log \frac{1}{\Pr\limits_{(X_j)_{j\neq i} | X_i = x_i} [\overleftrightarrow{\pi_i}(X_1, \ldots, X_{i-1}, x_i, X_{i+1}, \ldots, X_k, r) = t_i]} \right] \right] \\
&= \sum_{i=1}^{k} \mathop{\mathbb{E}}_{r,x_i} \left[ H(\overleftrightarrow{\Pi_i} \mid x_i r) \right] \\
&= \sum_{i=1}^{k} H(\overleftrightarrow{\Pi_i} | X_i R^p) \\
&= \sum_{i=1}^{k} I(X_{-i}; \overleftrightarrow{\Pi_i} | X_i R^p) \\
&= \sum_{i=1}^{k} I(X_{-i}; \Pi_i | X_i R^p) \\
&= \mathsf{IC}_\mu(\pi) \ ,
\end{aligned}
$$

where the one before last equality follows from Proposition 3.6 (and the one-to-one correspondence between the definition of $\overleftrightarrow{\Pi_i}$ used here and the definition of Section 3).

We have shown that the average number of stages is bounded by $\mathsf{IC}_\mu(\pi)$. At each stage, the communication consists of $\frac{k(k-1)}{2}$ calls to the lcp box on strings of length at most $\mathcal{O}(\mathsf{CC}(\pi))$ (one call for each pair of players), plus $k(k-1)$ messages of broadcasts of indices at Step (2), each message of size $\mathcal{O}(\log \mathsf{CC}(\pi))$. Hence we have a protocol with, on average, $O(k^2 \cdot \mathsf{IC}_\mu(\pi))$ calls to the lcp box on strings of length at most $\mathcal{O}(\mathsf{CC}(\pi))$ and with

$$
\mathsf{ACC}_\mu(\tilde{\rho}) = \mathcal{O}\left(k^2 \cdot \mathsf{IC}_\mu(\pi) \cdot \log(\mathsf{CC}(\pi))\right) \ .
$$

Using Lemma 7.7 we can replace each use of the lcp box with a simulation protocol, to get the protocol $\rho$ which simulates $\pi$ with distributional error $\epsilon + \delta$ and average communication:

$$
\begin{aligned}
\mathsf{ACC}_\mu(\rho) &= \mathsf{ACC}_\mu(\tilde{\rho}) + \mathcal{O}\left(k^2 \cdot \mathsf{IC}_\mu(\pi) \cdot \log \frac{k^2 \cdot \mathsf{IC}_\mu(\pi) \cdot \mathsf{CC}(\pi)}{\delta}\right) \\
&= \mathcal{O}\left(k^2 \cdot \mathsf{IC}_\mu(\pi) \cdot \log \frac{k^2 \cdot \mathsf{IC}_\mu(\pi) \cdot \mathsf{CC}(\pi)}{\delta}\right) .
\end{aligned}
$$

□

## 7.2 A direct sum for $\mathsf{PIC}$ implies a direct sum for $\mathsf{CC}$

The next theorem states that if $\mathsf{PIC}$ has a certain direct sum property then one can compress the communication of certain multi-party protocols. Note that the result of this theorem is meaningful when $t$ is large with respect to $k$.

**Theorem 7.8.** *In the oblivious setting, given a $k$-variable function $f$, if for any $t$ and any distribution $\mu$ on inputs of $f$ the existence of a protocol $\pi$ computing $f^{\otimes t}$ with error $\epsilon \geq 0$ implies that there exists a protocol $\pi'$ computing $f$ with error $\epsilon$ and satisfying $\mathsf{PIC}_\mu(\pi') \leq \frac{1}{t}\mathsf{PIC}_{\mu^{\otimes t}}(\pi)$, $\mathsf{CC}(\pi') \leq \mathsf{CC}(\pi)$, then for any fixed $\delta > 0$, for any $t$*

$$\mathsf{CC}^{2(\epsilon+\delta)}(f) = \mathcal{O}\left(\frac{1}{t(\epsilon+\delta)} \cdot k^3 \cdot \log(k) \cdot \mathsf{CC}^{\epsilon}(f^{\otimes t}) \cdot \log \frac{k^2 \cdot (\mathsf{CC}^{\epsilon}(f^{\otimes t})^2}{\delta}\right) .$$

To prove this theorem, we first need the following lemma.

**Lemma 7.9.** *Given an input distribution $\mu$, any $k$-party protocol with distributional error $\frac{\epsilon}{2}$ and average communication complexity $C$ can be turned into an oblivious protocol with distributional error $\epsilon$ and worst case communication complexity $\frac{C\cdot k\cdot\log(k)}{\epsilon}$.*

*Proof.* Let $\pi$ be a protocol with error $\frac{\epsilon}{2}$ and average communication complexity $C$. We now define a protocol $\pi'$, which is similar to $\pi$ but where player 1 acts as a “coordinator”, in addition to his original role in $\pi$, and the other players can only communicate with player 1.

In $\pi'$ the players will receive the messages from their peers via the coordinator in a way to be described below. When they wish to send a message to a peer, they will add this message, as a string of bits, to a local queue, together with the destination of the message. They will send the messages to their peers via the coordinator in a way to be described below.

So that the players can send and receive the messages the coordinator (player 1) imposes *phases* on the players. In every phase, player 1 sends a message to all players indicating the beginning of the phase. Each player then takes the next *bit*, denote it $b$, from its local queue and sends to the coordinator the message $(b, i)$, where $i$ is the destination of the the message $b$ is part of. If the player has no bits in its queue it sends the message “no” to player 1. Player 1, after having received all $k-1$ messages, forwards the bits it received to the various players. Every player $i$, where at least one bit destined to $i$ has been received, receives a message of the form $(b_1, j_1), \dots (b_q, j_q)$ (encoded in a self-delimiting manner) where $j_\ell$, $1 \leq \ell \leq q$ denote the origins of the bit, and all other players receive the message “no”. Observe that the players receiving the bits in this way can reconstruct the messages of the protocol $\pi$ since all messages (of $\pi$) are self delimiting, and can thus locally run the original protocol $\pi$. The protocol $\pi'$ consists of exactly $T = \left\lceil \frac{2C}{\epsilon} \right\rceil$ such phases. If at the end of $\pi'$ a certain player did not output according to $\pi$, then in $\pi'$ that player outputs an arbitrary output.

Note that $\pi'$ is oblivious. Moreover, $\pi'$ fails to simulate $\pi$ (i.e., there is at least one player which outputs differently in $\pi$ and in $\pi'$) only if $\pi'$ interrupts the simulation of $\pi$ at the end of the $T$’th phase. Since every phase in $\pi'$ transmits at least one additional bit of the communication of $\pi$, the probability that $\pi'$ interrupts the simulation of $\pi$ is the probability that the communication cost of $\pi$ is more than $T$. We have $\Pr_{x,r}(|\Pi(x)| \geq T) \leq \frac{C}{T} \leq \frac{\epsilon}{2}$ by Markov inequality. Adding that to the original error probability of $\pi$, we have that the protocol $\pi'$ has error $\epsilon$.

Last, every phase in protocol $\pi'$ consists of communication $\mathcal{O}(k\log(k))$, and protocol $\pi'$ thus has worst case communication $\mathcal{O}(T \cdot k \cdot \log(k)) = \mathcal{O}\left(\frac{C\cdot k\cdot\log(k)}{\epsilon}\right)$. □

*Proof of Theorem 7.8.* Consider a protocol $\pi$ computing $f^{\otimes t}$ with error $\epsilon$. Let $\mu$ be a distribution on inputs of $f$. By hypothesis, there exist a protocol $\pi'$ computing $f$ with error $\epsilon$ and satisfying $\mathsf{PIC}_\mu(\pi') \leq \frac{1}{t} \cdot \mathsf{PIC}_{\mu^{\otimes t}}(\pi)$, $\mathsf{CC}(\pi') \leq \mathsf{CC}(\pi)$. By Theorem 4.6, there exists such $\pi'$ that uses only public randomness.

Applying successively Theorem 7.4 and Lemma 7.9, we get a protocol $\rho_\mu$ with distributional error $2(\epsilon+\delta)$ such that

$$\begin{aligned}\mathsf{CC}(\rho_\mu) &= \mathcal{O}\left(\frac{1}{(\epsilon+\delta)} \cdot k^3 \cdot \log(k) \cdot \mathsf{PIC}_\mu(\pi') \cdot \log \frac{k^2 \cdot \mathsf{PIC}_\mu(\pi') \cdot \mathsf{CC}(\pi')}{\delta}\right) \\ &= \mathcal{O}\left(\frac{1}{(\epsilon+\delta)} \cdot k^3 \cdot \log(k) \cdot \mathsf{PIC}_\mu(\pi') \cdot \log \frac{k^2 \cdot (\mathsf{CC}(\pi'))^2}{\delta}\right) .\end{aligned}$$

Thus,

$$\begin{aligned}\mathsf{CC}(\rho_\mu) &= \mathcal{O}\left(\frac{1}{t(\epsilon+\delta)} \cdot k^3 \cdot \log(k) \cdot \mathsf{PIC}_{\mu^{\otimes t}}(\pi) \cdot \log \frac{k^2 \cdot (\mathsf{CC}(\pi))^2}{\delta}\right) \\ &= \mathcal{O}\left(\frac{1}{t(\epsilon+\delta)} \cdot k^3 \cdot \log(k) \cdot \mathsf{CC}(\pi) \cdot \log \frac{k^2 \cdot (\mathsf{CC}(\pi))^2}{\delta}\right) .\end{aligned}$$

Since the above holds for any distribution $\mu$, the minimax theorem implies that

$$\mathsf{CC}^{2(\epsilon+\delta)}(f) = \mathcal{O}\left(\frac{1}{t(\epsilon+\delta)} \cdot k^3 \cdot \log(k) \cdot \mathsf{CC}^{\epsilon}(f^{\otimes t}) \cdot \log \frac{k^2 \cdot (\mathsf{CC}^{\epsilon}(f^{\otimes t}))^2}{\delta}\right) .$$

□

# References

[BBCR13] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. *SIAM J. Comput.*, 42(3):1327–1363, 2013.

[BBK$^+$16] Joshua Brody, Harry Buhrman, Michal Koucký, Bruno Loff, Florian Speelman, and Nikolai K. Vereshchagin. Towards a reverse newman's theorem in interactive information complexity. *Algorithmica*, 76(3):749–781, 2016.

[BCKO93] 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.

[BEO$^+$13] 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.

[BG14] Mark Braverman and Ankit Garg. Public vs private coin in bounded-round information. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 502–513. Springer, 2014.

[BGPW13] 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.

[BJKS04] 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.

[BMY15] Balthazar Bauer, Shay Moran, and Amir Yehudayoff. Internal compression of protocols to entropy. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, volume 40 of *LIPIcs*, pages 481–496. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[BO15] 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.

[BOGW88] 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.

[BR14] Mark Braverman and Anup Rao. Information equals amortized communication. *IEEE Trans. Information Theory*, 60(10):6058–6069, 2014.

[Bra15] Mark Braverman. Interactive information complexity. *SIAM J. Comput.*, 44(6):1698–1739, 2015.

[CCD88] 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.

[CKS03] 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.

[CM15] Arkadev Chattopadhyay and Sagnik Mukhopadhyay. Tribes is hard in the message passing model. In 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*, pages 224–237. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[CR15] 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.

[CRR14] 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.

[CSWY01] 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.

[DF89] 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.

[FHW12] 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.

[FJK$^+$16] Lila Fontes, Rahul Jain, Iordanis Kerenidis, Sophie Laplante, Mathieu Laurière, and Jérémie Roland. Relative discrepancy does not separate information and communication complexity. *TOCT*, 9(1):4:1–4:15, 2016.

[FKN94] 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.

[FKNN95] Tomás Feder, Eyal Kushilevitz, Moni Naor, and Noam Nisan. Amortized communication complexity. *SIAM J. Comput.*, 24(4):736–750, 1995.

[FRPU94] Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM J. Comput.*, 23(5):1001–1018, October 1994.

[GG10] 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.

[GKR14] Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of information and communication. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 176–185, 2014.

[GKR15a] Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of communication and external information. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:88, 2015.

[GKR15b] Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of information and communication for boolean functions. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 557–566. ACM, 2015.

[Gro09] 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.

[HJMR10] Prahladh Harsha, Rahul Jain, David A. McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. *IEEE Transactions on Information Theory*, 56(1):438–449, 2010.

[Jai15] Rahul Jain. New strong direct product results in communication complexity. *J. ACM*, 62(3):20, 2015.

[Jay09] 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.

[JRS03] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A direct sum theorem in communication complexity via message compression. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, Eindhoven, The Netherlands, June 30 - July 4, 2003. Proceedings*, volume 2719 of *Lecture Notes in Computer Science*, pages 300–315. Springer, 2003.

[Kla10] Hartmut Klauck. A strong direct product theorem for disjointness. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 77–86. ACM, 2010.

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

[Kol16] Gillat Kol. Interactive compression for product distributions. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, pages 987–998, New York, NY, USA, 2016. ACM.

[KOS17] Gillat Kol, Rotem Oshman, and Dafna Sadeh. Interactive Compression for Multi-Party Protocol. In Andréa W. Richa, editor, *31st International Symposium on Distributed Computing (DISC 2017)*, volume 91 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:15, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Koz15] Alexander Kozachinskiy. *Computer Science – Theory and Applications: 10th International Computer Science Symposium in Russia, CSR 2015, Listvyanka, Russia, July 13-17, 2015, Proceedings*, chapter Making Randomness Public in Unbounded-Round Information Complexity, pages 296–309. Springer International Publishing, Cham, 2015.

[MNSW98] 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.

[Pan15] Denis Pankratov. *Communication complexity and information complexity*. PhD thesis, The university of Chicago, 2015.

[PVZ16] 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.

[RS15] Anup Rao and Makrand Sinha. Simplified separation of information and communication. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:57, 2015.

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

[Sha03] Ronen Shaltiel. Towards proving strong direct product theorems. *Computational Complexity*, 12(1-2):1–22, 2003.

[She18] Alexander A. Sherstov. Compressing interactive communication under product distributions. *SIAM J. Comput.*, 47(2):367–419, 2018.

[SHK$^+$10] 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.

[WZ14] David P. Woodruff and Qin Zhang. An optimal lower bound for distinct elements in the message passing model. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 718–733. SIAM, 2014.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing(preliminary report). In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, STOC ’79, pages 209–213, New York, NY, USA, 1979. ACM.