Abstract

We introduce 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. 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 establishes that the amount of randomness needed for a private protocol that computes this function is $\Omega(n)$.

1998 ACM Subject Classification F.0 - theory of computation, general

Keywords and phrases multi-party protocols, information theory, communication complexity, privacy, randomness

1 Introduction

Communication complexity, originally introduced by Yao [44], is a prolific field of research in theoretical computer science that yielded many important results in various fields. Informally, it answers the question “How many bits must the players transmit to solve a distributed problem?” The study of the two-party case has produced a large number of interesting and important results, 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., [36, 37, 25, 40, 24]).

A recent powerful tool for the study of two-party communication protocols is the measure of Information Complexity (or cost). This measure, originally defined in [1] and [14], extends the notions of information theory, originally introduced by Shannon [42], 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., [2, 3, 11, 6]). An interesting property of information complexity is the fact that it satisfies a direct sum. 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, it has been studied in [20, 14, 41, 32, 30, 3, 34, 31] 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. We 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 the perfect compression is not possible [27, 28]. Still, several interesting compression results are known. The equality between information cost and amortized communication cost is shown in [11, 6], and other compression techniques are given in [3, 4, 12]. It remains open if one can compress interactive communication up to some small loss (for example logarithmic in the size of the input).

When trying to study the multi-party (where at least 3 players are involved) communication setting using similar information-theoretic methods, such as IC, one encounters a serious problem. The celebrated results on information-theoretic private computation [5, 18] 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 IC of any function $f$ is 0 (or only the entropy of $f$, depending on the definition of IC), and cannot serve to study multi-party protocols.

For this reason, information complexity has rarely been used in the multi-party setting, where most results have been obtained via combinatorial techniques. Among the interesting works on multi-party setting are [38] and its follow-up [43] which introduce the techniques of symmetrization and composition, and [16, 17] which study the influence of the topology of the network. One notable exception is the work of Braverman et al. [7] which studies the set-disjointness problem using information theoretic tools. They provide almost tight bounds in the so-called coordinator model (that differs from the more natural peer-to-peer model) by using the information leaked between the players but also the information obtained by the coordinator itself. The set disjointness problem is maybe one of the most extensively studied problem in communication complexity (cf. [6, 2, 13, 29, 33, 9]). This line of research was followed by [15] which also uses information complexity to obtain tight bounds on the communication complexity of the function Tribes in the coordinator model. Information complexity is also used in [10] to study set-disjointness in the broadcast model.

A number of sub-models have been considered in the literature for the multi-party setting: the number in hand model (NIH), where each player has a private input, is maybe 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 itself. As to the communication pattern a number of variants exist 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 different variants see [36]). Most of the results obtained in multi-party communication complexity were obtained for the NOF model and/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 and tools 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 for 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. The goal is to prove lower bounds on PIC for a given
function \( f \), which will enable us to give lower bounds on the communication complexity
of the (multi-party) \( f \) and on the amount of randomness needed to privately compute \( f \).
The important point is that the PIC of functions, in our multi-party model, is not always 0,
unlike the value of 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. This is another issue when trying to use IC in the peer-to-
peer multi-party setting (a problem which does not occur in the coordinator model). 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. [19]). The latter point is particularly important when one is interested in privacy, since
our model allows for private protocols, while this is not necessarily the case for other models,
including the coordinator model. Furthermore, if one seeks to understand the natural peer-
to-peer model, suppressing polylog-factor-wide inaccuracies, one has to study directly the
peer-to-peer model (see the comparison of models of computation in subsection 3.1).

We go on to 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 Com-
  plexity provides a lower bound on the amount of randomness used in a protocol.
- We compress any communication protocol to their Public Information Cost (up to loga-
  rithmic factors), by extending to the multi-party setting the work of Brody et al. [12].
- We explicitly calculate 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. We show
  that PIC of that 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

In section 2 we review several notations and information theory basics. In section 3 we define our model, compare it to other models, and review the concept of privacy in multi-party computation. In section 4 we define the measure of public information cost (PIC) and prove a number of properties. Section 5 deals with the relation between PIC and the amount of randomness needed for private protocols. In section 6, we study the problem of compressing communication in multi-party protocols. In section 7, we prove tight bounds for the function Parity. We conclude in section 8 with some conclusions and directions for future work.

2 Preliminaries

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. For simplicity, we use the term entropy to talk about binary entropy.

▶ **Definition 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 \( \mathbb{E}_y[H(X \mid Y = y)] \).

▶ **Proposition 2.** For all finite set \( \mathcal{X} \subseteq \{0, 1\}^* \) and all random variable \( X \) with support \( \text{supp}(X) \subseteq \mathcal{X} \), 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 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 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 values of \( Z \).

▶ **Proposition 5.** For any random variables \( X \) and \( Y \), \( H(X \mid Y) \leq H(X) \).
Proposition 6 (Chain Rule). Let $A, B, C, D$ be four random variables. Then

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

Lemma 7 (Data processing inequality). For any $X, Y, Z$, and any function $f$, it holds:

$$I(X; f(Y) | Z) \leq I(X; Y | Z)$$

Proof.

$$I(X; f(Y) | Z) \leq I(X; Y | Z) + I(X; f(Y) | Y Z) \leq I(X; Y | Z) + I(X; f(Y) | Y Z)$$

Proposition 8 ([6]). Let $A, B, C, D$ be four random variables such that $I(B; D | AC) = 0$. Then

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

Proposition 9 ([6]). Let $A, B, C, D$ be four random variables such that $I(B; D | C) = 0$. Then

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

3 The model

In this section we 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. 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 multiparty 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 $f = (f_i)_{i \in [1, k]}$, with $\forall i \in [1, k], f_i : \prod_{l=1}^{k} X_l \to Y_i$, where $X_i$ denotes the set of possible inputs of player $i$, and $Y_i$ denotes the set of possible outputs of players $i$. The players are given some input $x = (x_i) \in \prod_{i=1}^{k} X_i$, and for every $i$, player $i$ has to compute $f_i(x)$. Each player has a special write-only output tape on which it writes the output.
We define the communication model as follows, which is the asynchronous setting, with some restrictions. To make the discussion and the proofs 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 of a message, but every message is delivered in finite time, and the communication link maintains FIFO order in each of the two directions. Given a specific time we define the view of player $i$, 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. Then the (local) round of player $i$ terminates. To make it possible for the player to identify the arrival of the complete message that it waits for, we require that each message sent by a player in the protocol is self-delimiting.

Denote by $D_i^j$ the view 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 sequence of functions for each player $i$, parametrized by the local round $j$, $j \geq 1$:

1. $S_i^j : D_i^{j-1} \rightarrow 2^{\{1,\ldots,k\}\setminus\{i\}}$, defining the set of players to which player $i$ sends the messages.
2. $m_i^j : D_i^{j-1} \rightarrow \{0,1\}^*$, for all $q \in S_i^j(D_i^{j-1})$, defining the content of the messages player $i$ sends. Each such message has to be self-delimiting.
3. $O_i^j : D_i^{j-1} \rightarrow \{0,1\}^*$, defining what the player writes on the output tape. Each player can write on its output tape a non-empty string only once.
4. $S_i^j : D_i^{j-1} \rightarrow 2^{\{1,\ldots,k\}\setminus\{i\}}$, defining the set of players from which player $i$ waits for 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 $S_i^0, S_i^1, \ldots$, ordered by local round number, and within each round by, say, the index of the player. We denote by $\Pi_i$ the concatenation of $\Pi_i$ together with a similar concatenation of the messages sent by player $i$ to the sets $S_i^0, S_i^1, \ldots$. We denote by $\Pi_{i\rightarrow j}$ 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 in the sequel for the case where all functions $f_i$ are the same function, that we denote simply $f$. The definitions in the case of family of functions are similar.

**Definition 10.** For $\epsilon \geq 0$, a protocol $\pi$ $\epsilon$-computes a function $f$ if for all $(x_1,\ldots,x_k) \in X_1 \times \cdots \times X_k$:

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

---

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.
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. In our case, the sets \( S^i \) and \( S^j \) 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 the restriction imposed in our model is natural and does not seem to be very restrictive, and furthermore does not exclude the existence of private protocols (as those are defined in the synchronous setting).

To exemplify the above mentioned issue in the general asynchronous model consider the following simple example of a 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 runs (when defining the transcript to be the concatenation of the transcripts on each link).

**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 Bob 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 [13, 29, 33, 38, 16, 17]. In [7], 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 multiplicative factor is sometimes necessary, e.g., for the \( q \)-index function, where players \( P_i \), \( 0 \leq i \leq k - 1 \), each hold an input bit \( x_i \), player \( P_k \) holds \( q \) indices \( 0 \leq j \leq k - 1 \), \( 1 \leq \ell \leq q \), and \( P_k \) should learn the vector \( (x_{j_1}, x_{j_2}, \ldots, x_{j_q}) \): 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 \( (\text{at most}) \min\{k, 2q\} \) bits, where \( P_k \) just queries the appropriate other players. But this factor is not always necessary: the communication complexity of the parity function \( \text{Par} \) is \( \Theta(k) \) both in the peer-to-peer models and in the coordinator model.

Moreover, when one wants to study private protocols in the peer-to-peer model, the coordinator model does not offer much insight. Observe that in the (asynchronous) coordinator model, described in [19] and used for instance in [7], 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 [21]. If no such source exists, privacy is impossible (if we require privacy also with respect to the coordinator). This follows from the results of Braverman et al. [7] who essentially show a non-zero lower bound on the total internal information complexity of all parties (including the coordinator) for the function \textbf{Disjointness} in that model.

Note also that the private protocols described in [5, 18] (and further work) are defined in the synchronous setting, and thus can be adapted to our communication model (basically, the sets $S_j^i$ and $S_j^j$ are always all the players and hence even independent of the inputs).

In the sequel we also use a special case of our model, where the sets $S_j^i$ and $S_j^j$ (i.e., the sets that define to which players player $i$ sends messages in (local) round $j$, and from which players he waits for messages in that round) 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 \textit{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. As mentioned above, this model also allows for private protocols.

3.3 Communication complexity and information complexity

Communication complexity, introduced in [44], measures how many bits of communication are needed so that the players can compute with error $\epsilon$ a given function of their joint inputs. The allowed error $\epsilon$, implicit in the notations, will be written explicitly as a subscript whenever necessary.

▶ Definition 11. The \textit{communication cost} of a protocol $\pi$, $CC(\pi)$, is the maximal length of the transcript of $\pi$ over all possible inputs, private randomness and public randomness.

▶ Definition 12. $CC(f)$ 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, it has led to interesting results on the communication complexity of various functions, such as \textbf{AND} and \textbf{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 (see [14, 2, 6] for more details) can be easily generalized to any number of players:

▶ Definition 13. The \textit{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: $IC_{\mu}(\pi) = \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i, R, R^p)$.

Intuitively, the information cost of a protocol is the amount of information that each player learns about the inputs of the other players when executing the protocol. The definition we gave above, when restricted to two players is the same as the one defined in [6], even though they look slightly different. This is due to the fact that we explicit the role of the randomness, which will allow us to later bound 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, in the same way this is done for the 2-party case.

**Definition 14.** 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 $\mu$:

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

**Definition 15.** The internal information complexity of a function $f$ is the infimum, over all protocols $\pi$ computing $f$, of the information cost of $\pi$ when run on the worst input distribution for that protocol:

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

**Proposition 16.** [11] For any protocol $\pi$ and input distribution $\mu$, $\text{CC}(\pi) \geq \text{IC}_\mu(\pi)$. Thus, for any function $f$, $\text{CC}(f) \geq \text{IC}(f)$.

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

**Proposition 17.** For any protocol $\pi$, for any player $i$:

$$I(X_{-i}; \Pi^i_e | X_iR^p) = I(X_{-i}; \Pi^i_e | X_iR^p),$$

$$I(X_{-i}; \Pi^i_e | X_iR^p f(X)) = I(X_{-i}; \Pi^i_e | X_iR^p f(X)).$$

*Proof.* We only prove the first statement, since the two statements have similar proofs.

We first prove the equality $I(X_{-i}; \Pi^i_e | X_iR^p) = I(X_{-i}; \Pi^i_e | X_iR^p)$. We write $\Pi^i_e$ as a sequence $b_0 \ldots b_q$ of bits, and define $\Pi^i_{< j}$ to be the set of bits with index less than $j$ and received by player $i$. Similarly, $\Pi^i_{< j}$ is defined to be the set of bits with index less than $j$ and sent by player $i$.

$$I(X_{-i}; \Pi^i_e | X_iR^p) = I(X_{-i}; b_0 \ldots b_q | X_iR^p)$$

$$= \sum_{j \geq 0} I(X_{-i}; b_j | X_iR^p b_0 \ldots b_{j-1})$$

(assuming chain rule (proposition 6))

For any $j$ and any bit $b_j \in \Pi^i_e$ that is sent in local round $\ell$, bit $b_j$ is determined by $D^\ell_{i-1}$. Therefore, for such $b_j$, $H(b_j | X_iR^p b_0 \ldots b_{j-1}) = 0$ and thus $I(X_{-i}; b_j | X_iR^p b_0 \ldots b_{j-1}) = 0$. Hence

$$I(X_{-i}; \Pi^i_e | X_iR^p) = \sum_{b_j \in \Pi^i_e} I(X_{-i}; b_j | X_iR^p b_0 \ldots b_{j-1}).$$

Fix $j$ such that $b_j$ is received by player $i$. We now show that

$$I(X_{-i}; b_j | X_iR^p \Pi^i_{< j} \Pi^i_{> j}) = I(X_{-i}; b_j | X_iR^p \Pi^i_{> j}).$$

Since $H(\Pi^i_{< j} | X_iR^p \Pi^i_{< j}) = 0$, we have $I(X_{-i}; \Pi^i_{< j} | X_iR^p \Pi^i_{< j}) = 0$ and applying lemma 9,

$$I(X_{-i}; b_j | X_iR^p \Pi^i_{< j} \Pi^i_{> j}) \geq I(X_{-i}; b_j | X_iR^p \Pi^i_{> j}).$$
Since $H(\overrightarrow{\Pi_i^{\leq j}} | X_i R_i R^p b_j \Pi_i^{\leq j}) = 0$, we have $I(X_{-i}; \overrightarrow{\Pi_i^{\leq j}} | X_i R_i R^p b_j \Pi_i^{\leq j}) = 0$ and applying lemma 8,

$$I(X_{-i}; b_j | X_i R_i R^p \Pi_i^{\leq j} \overrightarrow{\Pi_i^{\leq j}}) \leq I(X_{-i}; b_j | X_i R_i R^p \Pi_i^{\leq j}).$$

We get:

$$I(X_{-i}; \overrightarrow{\Pi_i} | X_i R_i) = \sum_{j|b_j \in \Pi_i} I(X_{-i}; b_j | X_i R_i R^p b_0 \ldots b_{j-1})$$

$$= \sum_{j|b_j \in \Pi_i} I(X_{-i}; b_j | X_i R_i R^p \Pi_i^{\leq j} \overrightarrow{\Pi_i^{\leq j}})$$

$$= \sum_{j|b_j \in \Pi_i} I(X_{-i}; b_j | X_i R_i R^p \Pi_i^{\leq j})$$

$$= I(X_{-i}; \overrightarrow{\Pi_i} | X_i R_i R^p) \text{ (using chain rule (proposition 6))}$$

We now prove the equality $I(X_{-i}; \overrightarrow{\Pi_i} | X_i R_i R^p) = I(X_{-i}; \overrightarrow{\Pi_i} | X_i R_i R^p)$. We denote by $\overrightarrow{\Pi_i^{\leq j}} = b_0 \ldots b_{j-1}$. We have:

$$I(X_{-i}; \overrightarrow{\Pi_i} | X_i R_i R^p) = I(X_{-i}; b_0 \ldots b_j | X_i R_i R^p)$$

$$= \sum_j I(X_{-i}; b_j | X_i R_i R^p \overrightarrow{\Pi_i^{\leq j}}) \text{ (using chain rule (proposition 6))}$$

We show that $I(X_{-i}; R_i | X_i R^p \overrightarrow{\Pi_i^{\leq j}}) = 0$. By proposition 4, it is equivalent to show that for each possible value of $(X_i, R_i, \overrightarrow{\Pi_i^{\leq j}})$, the random variables $X_{-i}$ and $R_i$ are independent. To do this, we fix a value $(x_i, r, \overrightarrow{\Pi_i^{\leq j}})$ of $(X_i, R_i, \overrightarrow{\Pi_i^{\leq j}})$, take a value $x_{-i}$ of $X_{-i}$ such that the event $X_{-i} = x_{-i}$ is compatible with the event $(X_i, R_i, \overrightarrow{\Pi_i^{\leq j}}) = (x_i, r, \overrightarrow{\Pi_i^{\leq j}})$, and prove that for any possible value $r_i$ of $R_i$, the quantity $Pr[r_i | x_i r \overrightarrow{\Pi_i^{\leq j}} x_{-i}]$ does not depend on $x_{-i}$.

We have $Pr[r_i | x_i r \overrightarrow{\Pi_i^{\leq j}} x_{-i}] = \sum_{r \in \Pi_i} Pr(r_i | x_i r \overrightarrow{\Pi_i^{\leq j}} x_{-i} r_{-i}) Pr[r_{-i} | x_i r \overrightarrow{\Pi_i^{\leq j}} x_{-i}]$.

We define the set function $\rho: (x_i r \overrightarrow{\Pi_i^{\leq j}} x_{-i} r_{-i}) \mapsto \{r_i | \overrightarrow{\Pi_i^{\leq j}}(x_i, x_{-i}, r, r_i, r_{-i}) = \overrightarrow{\Pi_i^{\leq j}}\}$, i.e., the set of possible values for $r_i$, such that inputs $x_i, x_{-i}$, public randomness $r$ and private randomness $r_i, r_{-i}$ will lead to a transcript of player $i$ beginning by $\overrightarrow{\Pi_i^{\leq j}}$.

Note that $Pr[r_{-i} | x_i r \overrightarrow{\Pi_i^{\leq j}} x_{-i}] \neq 0 \iff \rho(x_i r \overrightarrow{\Pi_i^{\leq j}} x_{-i}) \neq \emptyset$.

Also note that $r_i \notin \rho(x_i r \overrightarrow{\Pi_i^{\leq j}} x_{-i}) \implies Pr[r_i | x_i r \overrightarrow{\Pi_i^{\leq j}} x_{-i} r_{-i}] = 0$ and that

$$\exists r_i \in \rho(x_i r \overrightarrow{\Pi_i^{\leq j}} x_{-i} r_{-i}) \implies Pr[r_i | x_i r \overrightarrow{\Pi_i^{\leq j}} x_{-i} r_{-i}] \leq \frac{1}{|\rho(x_i r \overrightarrow{\Pi_i^{\leq j}} x_{-i})|}.$$
\[
\Pr[r_i \mid x_i \ x \leftarrow \pi_i \leftarrow j x_{-i}] = \sum_{r_{-i}} \Pr[r_i \mid x_i \ x \leftarrow \pi_i \leftarrow j x_{-i} \ r_{-i}] \Pr[r_{-i} \mid x_i \ x \leftarrow \pi_i \leftarrow j x_{-i}]
\]
\[
= \sum_{r_{-i}} \frac{1}{|\rho(x_i, r, \pi_i \leftarrow j, x_{-i}, r_{-i})|} \Pr[r_{-i} \mid x_i \ x \leftarrow \pi_i \leftarrow j x_{-i}]
\]
\[
= \sum_{r_{-i}} \frac{1}{\alpha(x_i, r, \pi_i \leftarrow j)} \Pr[r_{-i} \mid x_i \ x \leftarrow \pi_i \leftarrow j x_{-i}]
\]
\[
= \frac{1}{\alpha(x_i, r, \pi_i \leftarrow j)}
\]

which is independent of \(x_{-i}\). Thus \(\Pr[r_i \mid x_i \ x \leftarrow \pi_i \leftarrow j x_{-i}] = \Pr[r_i \mid x_i \ x \leftarrow \pi_i \leftarrow j] \). We have shown that the random variables \(X_{-i} \) and \(R_i\) are independent conditioned on \((X_i, R^p, \pi_i \leftarrow j)\), and thus
\[
I(X_{-i}; R_i \mid X_i R^p \pi_i \leftarrow j) = 0.
\]
By lemma 9, this implies
\[
I(X_{-i}; b_j \mid X_i R^p R_i \pi_i \leftarrow j) \geq I(X_{-i}; b_j \mid X_i R^p \pi_i \leftarrow j). \]
A similar reasoning, along with lemma 8, leads to
\[
I(X_{-i}; b_j \mid X_i R^p R_i \pi_i \leftarrow j) \leq I(X_{-i}; b_j \mid X_i R^p \pi_i \leftarrow j).
\]

We have:
\[
I(X_{-i}; \pi_i \leftarrow j \mid X_i R^p R_i) = \sum_j I(X_{-i}; b_j \mid X_i R_i R^p \pi_i \leftarrow j)
\]
\[
= \sum_j I(X_{-i}; b_j \mid X_i R^p \pi_i \leftarrow j)
\]
\[
= I(X_{-i}; \pi_i \leftarrow j \mid X_i R^p) \text{ (using chain rule (proposition 6))}
\]

### 3.4 Information complexity and privacy

The definition of a private protocol as defined in [5, 18] is the following.

**Definition 18.** A \(k\)-player protocol \(\pi\) for computing a family of functions \((f_i)\) is private\(^2\) if for every player \(i \in [1, k]\), for all pair 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 19.** A protocol \(\pi\) is private if and only if for all input distributions \(\mu\),
\[
\sum_{i=1}^{k} I(X-; \Pi_i \mid X_i R_i R^p f_i(X)) = 0.
\]

**Proof.** By proposition 4, definition 18 is equivalent to the following:
\[
\forall i, I(X-; \Pi_i \mid X_i R_i R^p f(X)) = 0 .
\]

\(^2\) In this paper we consider only the setting of 1-privacy, which we call here for simplicity, privacy.
Since $I$ is non-negative, this is equivalent to
\[ \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R_i R^p f_i(X)) = 0. \]

It is well known that in the multi-party NIH peer-to-peer setting (for $k \geq 3$), unlike in the two-party case, any function can be privately computed. This is also true in our model.

**Theorem 20 ([5],[18]).** 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 21.** For any family of functions $(f_i)$ of more than two variables and any $\mu$,
\[ IC_\mu(f) = \sum_{i=1}^{k} H(f_i(X)), \text{ where } X \text{ is distributed according to } \mu. \]

**Proof.** Let $\pi$ be a $k$-player private protocol computing $(f_i)$. Fix a distribution $\mu$ on the inputs.
\[
IC_\mu(\pi) = \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i R_i R^p) \\
\leq \sum_{i=1}^{k} I(X_{-i}; f_i(X) \mid X_i R_i R^p) \\
\leq \sum_{i=1}^{k} [I(X_{-i}; f_i(X) \mid X_i R_i R^p) + I(X_{-i}; \Pi_i \mid X_i R_i R^p f_i(X))] \\
\leq \sum_{i=1}^{k} I(X_{-i}; f_i(X) \mid X_i R_i R^p) \\
\leq \sum_{i=1}^{k} H(f_i(X))
\]

Now, $IC_\mu(f) \leq IC_\mu(\pi) \leq \sum_{i=1}^{k} H(f_i(X)).$
Definition 22. For any \( k \)-player protocol \( \pi \) and any input distribution \( \mu \), we define the

\[ \text{PIC}_\mu(\pi) = \sum_{i=1}^{k} I(X_{-i}; \Pi_i, R_{-i} \mid X_i, R_i, R_p) \]

The difference between PIC and IC is the presence of the other parties private coins, \( R_{-i} \), in

the formula. If \( \pi \) is a protocol using only public randomness, then for any input distribution

\( \mu \), \( \text{PIC}_\mu(\pi) = \text{IC}_\mu(\pi) \), and hence the name public information cost.

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. It can be

decomposed, using the chain rule, into two terms, making explicit the contribution of the

internal information cost and of the private randomness of the players.

Proposition 23. For any \( k \)-player protocol \( \pi \) and any input distribution \( \mu \),

\[ \text{PIC}_\mu(\pi) = \text{IC}_\mu(\pi) + \sum_{i=1}^{k} I(R_{-i}; X_{-i} \mid X_i, \Pi_i, R_i, R_p) \]

The meaning of the second term is 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

it some new information \( I(R_{-i}; X_{-i} \mid X_i, \Pi_i, R_i, R_p) \) about the inputs \( X_{-i} \) of the other players.

We also define the public information complexity of a function.

Definition 24. For any function \( f \) and any input distribution \( \mu \), we define the quantity

\[ \text{PIC}^\mu(f) = \inf_{\pi \text{ computing } f} \text{PIC}_\mu(\pi) \]

Definition 25. For any function \( f \), we define the quantity \( \text{PIC}(f) = \inf_{\pi \text{ computing } f} \sup_{\mu} \text{PIC}_\mu(\pi) \).

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

Proposition 26. For any protocol \( \pi \) and input distribution \( \mu \), \( \text{CC}(\pi) \geq \text{PIC}_\mu(\pi) \). Thus, for any function \( f \), \( \text{CC}(f) \geq \text{PIC}(f) \).

Proof.

\[ \text{PIC}_\mu(\pi) = \sum_{i=1}^{k} I(X_{-i}; R_{-i} \mid X_i, R_i, R_p) + I(X_{-i}; \Pi_i \mid X_i, R_i, R_p) \]

(by the chain rule)

\[ = \sum_{i=1}^{k} I(X_{-i}; \Pi_i \mid X_i, R_i, R_p) \]

(since the first term was 0 by proposition 4)

\[ = \sum_{i=1}^{k} H(\Pi_i \mid X_i, R_i, R_p) \]

(since \( H(\Pi_i \mid X_i, R_i, R_p) = 0 \))

\[ \leq \sum_{i=1}^{k} H(\Pi_i) \]

(by proposition 5)

Using proposition 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

g et \( \text{CC}(\pi) \geq \text{PIC}_\mu(\pi) \).
In fact, the public information cost of a function is equal to its internal information cost in a setting where only public randomness is allowed. The role of private coins in communication protocol has been studied for example in [8, 12, 35]. 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.

\begin{theorem}
For any function \( f \) and input distribution \( \mu \),

\[
\text{PIC}_\mu(f) = \inf \limits_{\pi \text{ computing } f, \text{ using only public coins}} \text{IC}_\mu(\pi)
\]

and

\[
\text{PIC}(f) = \inf \limits_{\pi \text{ computing } f, \text{ using only public coins}} \sup \limits_{\mu} \text{IC}_\mu(\pi).
\]
\end{theorem}

\begin{proof}
It suffices to show than one can turn any protocol \( \pi \) into a public coin protocol \( \pi' \) such that for all input distributions \( \mu \), \( \text{PIC}_\mu(\pi') = \text{PIC}_\mu(\pi) \).

Fix a protocol \( \pi \) and an input distribution \( \mu \). Let \( R_i \) denote the private randomness of player \( i \) in \( \pi \), and \( R = (R_i) \). Define \( \pi' \), where the players act as they do in \( \pi \), but use public randomness instead of their private randomness whenever they need to use it.

For this, split the random tape into two sub-tapes \( R = (R_i) \), to be used instead of the private randomness of each player, and \( R^p \), to be used as public randomness. We have

\[
\text{PIC}_\mu(\pi') \leq \sum_{i=1}^{k} I(X_{-i}; \Pi_i | X_i R R^p)
\]

\[
\leq \sum_{i=1}^{k} \left[ I(X_{-i}; \Pi_i R_{-i} | X_i R_i R^p) - I(X_{-i}; R_{-i} | X_i R_i R^p) \right] \quad \text{(chain rule)}
\]

\[
= \sum_{i=1}^{k} I(X_{-i}; \Pi_i R_{-i} | X_i R_i R^p) \quad \text{(since the second term was 0)}
\]

\[
= \text{PIC}_\mu(\pi)
\]

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

\begin{proposition}
For any function \( f \), for any input distribution \( \mu \), \( \text{PIC}_\mu^0(f) = \text{IC}_\mu^\text{det}(f) \) where

\[
\text{IC}_\mu^\text{det}(f) = \inf \limits_{\pi \text{ deterministic protocol computing } f} \text{IC}_\mu(\pi).
\]
\end{proposition}

\begin{proof}
Let \( \pi \) be a zero-error protocol for \( f \). By theorem 27, one can assume that \( \pi \) has no private randomness.

\[
\text{IC}_\mu(\pi) = \sum_{i=1}^{k} I(X_{-i}; \Pi_i | X_i R^p) = \sum_{i=1}^{k} \mathbb{E} [I(X_{-i}; \Pi_i | X_i, R^p = r) | r] = \mathbb{E} \left[ \sum_{i=1}^{k} I(X_{-i}; \Pi_i | X_i, R^p = r) \right]
\]

Letting \( t(r) = \sum_{i=1}^{k} I(X_{-i}; \Pi_i | X_i, R^p = r) \), it holds \( \text{IC}_\mu(\pi) = \mathbb{E} [t(r)] \).

Let \( r_0 \) be a value of the public random tape minimizing the function \( t \), and define \( \pi^0 \) as the protocol behaving like \( \pi \) on the random tape \( r_0 \). Note that \( \pi^0 \) is a deterministic zero-error protocol computing \( f \). We now prove that \( \text{IC}_\mu(\pi^0) \leq \text{IC}_\mu(\pi) \), which concludes the proof, by using the definition of \( \pi^0 \).

\[
\text{IC}_\mu(\pi^0) = \sum_{i=1}^{k} I(X_{-i}; \Pi_i | X_i) = \sum_{i=1}^{k} I(X_{-i}; \Pi_i | X_i R = r_0) \leq \text{IC}_\mu(\pi).
\]
PIC and IC are strictly different even in the two party case. We prove below that for the AND function, the public information cost is log 3, while, as shown in [9], IC(AND) \( \simeq 1.49 \). This implies that the protocol that achieves the optimal information cost must use private coins. We remark also that in [9], it is shown that the external information cost of AND, that we do not consider here, is log(3).

\[ \text{Proposition 29.} \] For two players, PIC\(^0\)(AND) = log(3).

**Proof.** Let \( \pi \) be a zero-error protocol for AND. By proposition 28, we can assume that \( \pi \) is deterministic. Suppose player 1 is speaking first. \( \pi \) being deterministic, player 1 is either sending his bit \( x \) or sending \( 1-x \). Player 2 is then able to compute the value of AND, and as player 1 must be able to compute AND at the end of the protocol, the optimal protocol consists in player 2 sending back the value of AND to player 1. We compute the value of PIC\( _\mu \)(\( \pi \)), for \( \mu \) defined as follows: \( X \sim \text{Ber}(\alpha, 1-\alpha) \) and \( Y \sim \text{Ber}(\beta, 1-\beta) \).

\[
\text{PIC}_\mu(\pi) = I(X; \Pi|Y) + I(Y; \Pi|X) = H(X) + [\alpha I(Y; \Pi|X = 0) + (1-\alpha) I(Y; \Pi|X = 1)].
\]

When \( X = 1 \), player 1 will learn the value of \( Y \), as \( \text{AND}(X,Y) = Y \). Thus PIC\( _\mu \)(\( \pi \)) = \( H(X) + (1-\alpha)H(Y) \). For any \( \alpha \), this quantity is maximal for \( \beta = \frac{1}{2} \). Thus we study the function \( f(\alpha) = \log(\alpha) + (\alpha - 1)\log(1-\alpha) + 1 - \alpha \) for \( \alpha \in [0,1] \). \( f \) is continuous on \([0,1]\) and differentiable on \([0,1]\). We have:

\[
f'(\alpha) = -\log(\alpha) - 1 + \log(1-\alpha) + 1 = \log\left(\frac{1}{\alpha} - 1\right) - 1.
\]

\( f' \) is continuous and decreasing on \([0,1]\), and admits the unique root \( \frac{1}{3} \). PIC\( _\mu \)(\( \pi \)) is thus maximized for \( \alpha = \frac{1}{3} \), its value being \( f(\alpha) = \log(3) \).

\[ \text{Proposition 30.} \] 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: \( H_\mu(\Pi | X R^p) \geq \frac{\text{PIC}_\mu(\pi) - \text{IC}_\mu(\pi)}{k-1} \). Thus running a protocol for \( f \) with information cost \( I_\mu \) requires entropy \( H_\mu(\Pi | X R^p) \geq \frac{\text{PIC}_\mu(f) - I_\mu}{k-1} \).

**Proof.** Fix \( \mu \) a distribution on inputs.

\[
\text{PIC}_\mu(\pi) = \text{IC}_\mu(\pi) + \sum_{i=1}^{k} I(R_{-i}; X_{-i} | X_i R_i R^p \Pi_k) \quad \text{(proposition 23)}
\]

\[
\leq \text{IC}_\mu(\pi) + \sum_{i=1}^{k} I(R_{-i}; XR_i \Pi_i | \ R^p).
\]
For any $i$,
\[
I(R_{-i}; XR_{i} \Pi | R^p) \leq I(R_{-i}; XR_{i} \Pi | R^p) \leq I(R_{-i}; \Pi | R^p) + I(R_{-i}; XR_{i} | \Pi R^p) \leq I(R_{-i}; \Pi | R^p) + I(R_{-i}; X | \Pi R^p) \quad (\text{as } I(R_{-i}; R_{i} | XR_{i} \Pi R^p) = 0, \text{ cf. lemma 42})
\]
\[
\leq I(R_{-i}; \Pi X | R^p) = \sum_{j \neq i} I(R_j; \Pi X | R^p) \quad (\text{by lemma 8 with } I(R_j; R_{<j} \neq i | XR_{i} \Pi R^p) = 0).
\]

Thus
\[
\text{PIC}_{\mu}(\pi) \leq I_{\mu}(\pi) + (k - 1) \sum_{i=1}^{k} I(R_i; \Pi X | R^p)
\]
\[
\leq I_{\mu}(\pi) + (k - 1) I(R; \Pi X | R^p) \quad (\text{by lemma 9})
\]
\[
\leq I_{\mu}(\pi) + (k - 1) I(R; X | R^p) \quad (\text{as } I(R; X | R^p) = 0)
\]
\[
\leq I_{\mu}(\pi) + (k - 1) H_{\mu}(\Pi | X R^p)
\]

Using lemma 21, we can bound the randomness required to run a private protocol.

\[\blacktriangleleft\]

\textbf{Corollary 31.} 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,
\[
H_{\mu}(\Pi | X R^p) \geq \frac{\text{PIC}_{\mu}(f) - \sum_{i=1}^{k} H_{\mu}(f_i)}{k - 1}
\]

\section{Relation between PIC and CC : Compression result}

An important open question is how well we can compress the communication cost of an interactive protocol. Compression results have appeared in [3, 11, 12, 4], while, on the other hand, [26, 28, 39, 23, 27] focus on the hardness of compressing communication protocols. Here, we present a compression result with regards to the average-case communication complexity and the public information cost.

\textbf{Definition 32.} The \textit{average-case communication complexity} of a protocol $\pi$ with respect to the input distribution $\mu$, denoted $\text{ACC}_{\mu}(\pi)$, is the expected number of bits that are transmitted in an execution of $\pi$ for inputs distributed according to $\mu$ and uniform randomness.

\textbf{Theorem 33.} Suppose there exists a protocol $\pi$ to compute a $k$-variable function $f$ over the distribution $\mu$ with error probability $\epsilon$. Then there exists a public-coin protocol $\rho$ that computes $f$ over $\mu$ with error $\epsilon + \delta$, and with average communication complexity
\[
\text{ACC}_{\mu}(\rho) = O \left( k^2 \text{PIC}_{\mu}(\pi) \log(\text{CC}(\pi)) \left( \log(k \text{CC}(\pi)) + \log \frac{k^2 \text{PIC}_{\mu}(\pi) \log(\text{CC}(\pi))}{\delta} \right) \right).
\]

The proof of the above theorem will follow from extending the compression result presented in [12] to the case of $k > 2$ players.
Theorem 34. Suppose there exists a public coin protocol $\pi$ to compute a $k$-variable function $f$ over the distribution $\mu$ with error probability $\epsilon$. Then there exists a public-coin protocol $\rho$ that computes $f$ over $\mu$ with error $\epsilon + \delta$, and with average communication complexity
\[
\text{ACC}_\mu(\rho) = \mathcal{O} \left( k^2 I_C(\mu) \log(\text{CC}(\pi)) \left( \log(k \text{CC}(\pi)) + \log \frac{k^2 I_C(\mu) \log(\text{CC}(\pi))}{\delta} \right) \right).
\]

Theorems 34 and 27, which makes the link between the public information cost of general protocols and the information cost of public coins protocol, implies theorem 33.

The compression scheme of [12] for the two party case works as follows. In the compressed protocol $\rho$, the two players, given their own input, 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 $\text{Im}(\pi(x_1, \cdot))$ of possible transcripts knowing that he has input $x_1$, while player 2 picks a candidate $t_2$ from the set $\text{Im}(\pi(\cdot, x_2))$. The two players then communicate to find the first bit on which $t_1$ and $t_2$ disagree. The general structure of communication protocols ensures that the common prefix of $t_1$ and $t_2$ is identical to the beginning of $\pi(x_1, x_2)$. 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 little communication.

For the multi-party case, we extend the proof from [12]. New difficulties occur. The players can no longer try to guess the full transcript, as they have little information about the conversation between the other players, but can only try to guess their partial transcript of the protocol, according to their own input. Then, in order to find the first mistake in the global transcript, every pair of parties needs to find and communicate the place of the first mistake in their partial transcript. This induces the $k^2$ factor in our compression scheme. While we do not know if this is necessary, it seems natural for compression schemes in the peer-to-peer models. Last, the players lack a common reference time. To solve this problem, we will introduce, as a technical tool in the proof, a coordinator, whose role is to introduce a round structure in the protocol $\pi$. Note that this is only in the proof, and that the stated results hold in our model.

We will use a black box, call it lcp box (for largest common prefix), which can be used by two players $A$ and $B$ in the following way: $A$ puts a string $x$ in the box, $B$ puts a string $y$ in the box, and the box gives them back the first index $j$ such that $x_j \neq y_j$ if $x \neq y$, or tells them that $x = y$ otherwise. The price to pay for using this black box is $\log n$ bits of communication, where $n = \max(|x|, |y|)$.

This box can be efficiently simulated if we allow error:

Lemma 35 ([22]). There is a randomized public coin protocol such that on input two $n$-bits strings $x$ and $y$, it outputs the first index $j$ such that $x_j = y_j$ with probability at least $1 - \epsilon$ if such a $j$ exists, and otherwise outputs that the two strings are equal, with worst case communication complexity $\mathcal{O}(\log(n/\epsilon))$.

Corollary 36 ([12]). Any protocol $\tilde{\rho}$ that uses an lcp box $l$ times on average can be simulated with error $\delta$ by a protocol $\rho$ that does not use an lcp box, and communicates an average of $\mathcal{O}(l \log(\frac{1}{\delta}))$ extra bits.

We will use the lcp box in the proof, and use corollary 36 to perform the analysis at the end.

Proof of theorem 34. Starting from protocol $\pi$, we build a protocol $\pi'$ which introduces a new player, called coordinator. In the beginning, the coordinator sends a signal message to
each player. When a player $i_1$ receives the signal from the coordinator, if he wants to send a bit $m$ to some player $i_2$ in $\pi$, in $\pi'$ player $i_1$ sends instead $(m,i_2)$ to the coordinator (he can send messages to different players). If a player does not want to send any bit to the other players, he sends a “no” signal to the coordinator. When the coordinator has received an answer from all the players, he forwards the messages: if the coordinator has received some message $(m,i_2)$ from $i_1$, he sends $(m,i_1)$ to player $i_2$. If the coordinator has no message to send to a player, he sends him a “no” signal. This was the first round. Each player can now either send messages to be forwarded by the coordinator, either a “no” signal, and so on. Thus, in $\pi'$ each player communicates only with the coordinator. $\pi'$ computes the same function as $\pi$ with the same error ratio, and $\text{CC}(\pi') = O(k\text{CC}(\pi))$. Just as $\pi$, $\pi'$ uses only public randomness.

We can now define the protocol $\tilde{\rho}$ (in which there is no coordinator). The players start by picking the shared randomness $r$ as in protocol $\pi'$. For each $i$, define the set $X_i$ to be the set of possible inputs of player $i$, and the set $\Pi(x_i)$ the set of possible transcripts of the communication between player $i$ and the coordinator, knowing that player $i$ has input $x_i$:

$$\Pi(x_i) = \bigotimes_{i=1}^{\#(X)} (X_1, \ldots, X_{i-1}, x_i, X_{i+1}, \ldots, X_k, r).$$

We define a canonical way to write down the transcripts. A transcript $t_i$ has the form $t_1^i \# t_2^i \# \ldots$ where $t_j^i$ is a description of the partial transcript of player $i$ of the $m$th round of the protocol. We now define the partial transcripts. Formally, for $i < j$, denoting by $t_j^m$ the partial transcript of player $i$ of the transcript and by $t_j^m$ the partial transcript of player $j$ of the transcript within the $m$th round, the messages sent by player $i$ come before those sent by player $j$ in $t_j^m$ and $t_j^m$, and if $j_1 < j_2$, the messages sent to or received from player $j_1$ come before the messages sent to or received from to $j_2$ in $t_j^m$ for $i \neq j_1, i \neq j_2$. $t_j^m$ is a sequence of messages of the form $(b, i)$ where $b$ is a bit and $i$ is the index of the other player involved, and of $\#$ which mark the separation between the messages exchanged with player $i$ and the messages exchanged with player $i + 1$ and the separation between messages sent to player $i$ and messages received from player $i$.

For example, the transcript of the $m$th round by player 2 in a protocol with three players is of the form

$$t_2^m = (\text{messages by 1 to 2}) \# (\text{messages by 2 to 1}) \# (\text{messages by 2 to 3}) \# (\text{messages by 3 to 2}).$$

Note that for any $t \in \Pi(x_i)$, thanks to the $\#$ messages, it is easy to associate to each message the round it has been sent or received.

Each player $i$ represents $\Pi(x_i)$ by a ternary tree $T_i$ as follows.

1. The root is the largest common prefix (lcp) of the transcripts in $\Pi(x_i)$, and the remaining nodes are defined inductively.
2. If we have node $\tau$, then
   - the first child of $\tau$ is the lcp of the transcripts in $\Pi(x_i)$ beginning with $\tau \circ (0,j)$, i.e. $\tau$ concatenated with the message $(0,j)$, where $j$ is the next player to communicate with according to $\tau$ (this $j$ is determined by the number of $\#$ in $\tau$).
   - the second child of $\tau$ is the lcp of the transcripts in $\Pi(x_i)$ beginning with $\tau \circ (1,j)$, where $j$ is the next player to communicate with according to $\tau$.
   - the third child is $\tau \circ \#$. 

The leaves are labelled by the possible transcripts of player \( i \), i.e., the elements of \( \Pi_i(x_i) \). We impose that each transcript ends with a “end” signal, thus a transcript cannot be the prefix of another transcript. The leaf \( f \) with label \( t_i \) has weight \( w(t_i) = \Pr_{(X_i)_1, x_i = t_i} [\tau_1^\pi(x_1, \ldots, X_i = t_i, x_i, X_{i+1}, \ldots, X_k, r) = t_i] \).

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, for each round \( i \), any message which appears in \( t_i \) as sent to player \( j \) also appears in \( t_j \) as coming from \( i \).

In fact, \((\tau_1^\pi(x_1, \ldots, x_k, r), \ldots, \tau_k^\pi(x_1, \ldots, x_k, r))\) is the only coherent profile. Indeed, take a coherent profile \((t_1, \ldots, t_k)\). For each \( i \), \( t_i \) is of the form \( \tau_i^\pi(x_{i1}, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{ik}) \) where \( \forall j \neq i \), \( x_j \in X_j \). Since whenever a player sends a message, his choice is based only on his own input and on the messages he has received before, it implies (by induction on the rounds of the protocol) that \((t_1, \ldots, t_k) = (\tau_1^\pi(x_1, \ldots, x_k, r), \ldots, \tau_k^\pi(x_1, \ldots, x_k, r))\).

We now show how the players can collaborate to find efficiently this coherent profile, that is how each player \( i \) can find \( \tau_i^\pi(x_1, \ldots, x_k, r) \). The players will proceed in stages \( s = 1, 2 \ldots \)

1. At the beginning of stage \( s \), each player \( i \) is at a node \( \tau_i(s) \) such that \((\tau_1(s), \ldots, \tau_k(s))\) is a (term-wise) prefix of \((\tau_1^\pi(x_1, \ldots, x_k, r), \ldots, \tau_k^\pi(x_1, \ldots, x_k, r))\). Player \( i \) starts the protocol with the node \( \tau_i(1) \), which is the root of the tree \( T_i \).
2. Each player \( i \) picks a candidate leaf \( t_i(s) \) in the tree \( T_i \), such that \( t_i(s) \) is a successor of \( \tau_i(s) \). We describe later how each player chooses his candidate leaf.
3. Each pair of players \((i, j)\) will use an lcp box to find the first occurrence where \( t_i(s) \) is not coherent with \( t_j(s) \), that is one of the players is supposed to send some message \( m \) in some round but the other is not supposed to receive it during this round, or one of the players is not supposed to send some message \( m \) in some round but the other is supposed to receive it during this round. If for all pairs of players there is no such occurrence, it means that \((t_1(s), \ldots, t_k(s))\) is a coherent profile, each player \( i \) has found \( \tau_i^\pi(x_1, \ldots, x_k, r) \), and the protocol terminates.
4. Denote by \( z_i \) the earliest round where player \( i \) disagrees with some other player. Then player \( i \) sends a signal to the first player (i.e. with the lowest index) he disagrees with during round \( z_i \).
5. For any \((i, j)\) such that each one has sent a signal to the other (note that there is at least one such pair \((i, j)\): each player receiving a signal also sends a signal, and the signals cannot form a cycle of length more than 2), players \( i \) and \( j \) exchange information about the first occurrence they found using the lcp box: each player tells the other what he wanted to send, or what he expected to receive. The player who has the sender role is always correct: for instance, if player \( i \) wanted to send 0 but player \( j \) expected to receive 1, or nothing (as encoded by \( \xi \)), then \( i \) is right, because what player \( i \) sent in the protocol \( \pi \) is based on the previous rounds (which are correct by hypothesis). Thus the player \( j \) having the receiver role is wrong. Player \( j \) updates his \( \tau_j \) in \( T_j \), starting from \( t_j(s) \), he goes up toward \( \tau_j(s) \), until he reaches a node \( \tilde{\tau}_j \) which is correct (according to the result of the lcp box). Then, he defines \( \tau_j(s+1) \) as the correct child of \( \tilde{\tau}_j \), which he infers from the information exchanged with player \( i \).
6. The players who have not updated their \( \tau_i \) yet define \( \tau_i(s + 1) = \tau_i(s) \).
First, note that the property of item (1) has been preserved for all players. This property ensures that the players will eventually agree on $\hat{\psi}_i'(x_1, \ldots, x_k, r)$. We now specify how each player $i$ chooses his candidate leaf at each stage. Player $i$ picks the leaf $t_i(s)$ which corresponds to the transcript with highest probability conditioned on the prefix specified by the node the player is at. Formally, at stage $s$, player $i$ defines $\tau^i = \tau_i(s)$, and then defines inductively $\tau_j^i = \tau^i$ to be the child of $\tau_j^i$ which has higher weight (breaking ties arbitrarily), until he reaches a leaf: this is the candidate $t_i(s)$.

Suppose that during round $s$, player $i$ has participated in step (5) and has made the wrong choice (there must be at least one such player per round). We show that $w(\tau_i(s + 1)) \leq \frac{1}{2} w(\tau_i(s))$. We look at the sequence $(\tau^i)$ defined by player $i$ when he chose his candidate leaf $t_i(s + 1)$ at step $s$. Let $\tau^i$ be the first common ancestor of $t_i$ and $\tau_i(s + 1)$. By construction, $\tau_i(s + 1)$ is a direct child of $\tau^i$, and $t_i$ is a descendant of another child of $\tau^i$. By the candidate leaf's construction process, $w(\tau_i(s + 1)) \leq \frac{1}{2} w(\tau^i) \leq \frac{1}{2} w(\tau_i(s))$.

We conclude the analysis. On inputs $(x_1, \ldots, x_k)$, let $\{t_1, \ldots, t_k\}$ denote the coherent profile. Each player will correct his $\tau_i$ no more than $\log \frac{1}{w(t_i)}$ times, because the weight of the node $\tau_i$ halves with each correction, as noticed before, and given that the root has weight 1. Hence, the total number of corrections, and thus the number of stages $S$, is bounded by $\sum_{i=1}^k \log \frac{1}{w(t_i)}$. We take the average over inputs and shared randomness:

$$
E[S] = E \sum_{r, x} \sum_{i=1}^k \log \frac{1}{w(t_i)}
$$

$$
= \sum_{i=1}^k E_{r, x} \left[ \sum_{(x_j)_{j \neq i}, X_i = x_i} \left[ \log \frac{1}{w(t_i)} \right] \right]
$$

$$
= \sum_{i=1}^k E_{r, x} \left[ \sum_{(x_j)_{j \neq i}, X_i = x_i} \left[ \log \frac{\Pr_{(X_j)_{j \neq i}, X_i = x_i} [\hat{\psi}_i'(X_1, \ldots, X_{i-1}, X_i, X_{i+1}, \ldots, X_k, r)]}{\Pr_{(X_j)_{j \neq i}, X_i = x_i} [\hat{\psi}_i'(X_1, \ldots, x_k, r)]} \right] \right]
$$

$$
= \sum_{i=1}^k E_{r, x} \left[ \sum_{t_i, X_i = x_i, R = r} \left[ \log \frac{1}{\Pr_{(X_j)_{j \neq i}, X_i = x_i} [\hat{\psi}_i'(X_1, \ldots, X_{i-1}, x_i, X_{i+1}, \ldots, X_k, r)]} \right] \right]
$$

$$
= \sum_{i=1}^k E_{r, x} \left[ H(\hat{\psi} R | x_i r) = \sum_{i=1}^k H(\hat{\psi} | X_i R) = \sum_{i=1}^k I(X_i; \hat{\psi} | X_i R) = IC_{\mu}(\pi') \right]
$$

We have shown that the average number of stages is bounded by $IC_{\mu}(\pi') \leq IC_{\mu}(\pi) \log(CC(\pi))$ as the only additional information players get in $\pi'$ is the round of each message. At each stage, the communication consists of $\frac{k(k-1)}{2}$ calls to the lcp box on strings of length at most $O(kCC(\pi))$ (one call for each pair of players) plus the $k$ signals of step (4) and the exchange of information of step (5) (there are at most $\frac{k}{2}$ pairs getting to step (5)). Hence

$$
ACC_{\mu}(\rho) = O \left( IC_{\mu}(\pi) \log(CC(\pi)) (k(k-1) \log(kCC(\pi)) + k + \frac{k}{2}) \right)
$$

$$
= O \left( k^2 IC_{\mu}(\pi) \log(kCC(\pi)) \log(CC(\pi)) \right).
$$

Using corollary 36 we can replace each use of the lcp box with the protocol given by Lemma 36, to get the protocol $\rho$ which simulates $\pi$ with error $\delta$ and average communication:
\[ \text{ACC}_\mu(\rho) = \text{ACC}_\mu(\bar{\rho}) + O \left( k^2 \text{PIC}_\mu(\pi) \log(k \text{CC}(\pi)) + \log \frac{k^2 \text{PIC}_\mu(\pi) \log(k \text{CC}(\pi))}{\delta} \right) \]

As a corollary, we have:

\textbf{Corollary 37.} Suppose there exists a protocol \( \pi \) to compute a \( k \)-variable \( f \) over the distribution \( \mu \) with error probability \( \epsilon \). Then there exists a public-coin protocol \( \rho \) that computes \( f \) over \( \mu \) with error \( 2(\epsilon + \delta) \), and with

\[ \text{CC}(\rho) = O \left( \frac{1}{\epsilon} k^2 \text{PIC}_\mu(\pi) \log(k \text{CC}(\pi)) \log(k) \left( \log(k \text{CC}(\pi)) + \log \frac{k^2 \text{PIC}_\mu(\pi) \log(k \text{CC}(\pi))}{\delta} \right) \right). \]

This corollary is a consequence of the following lemma.

\textbf{Lemma 38.} Any \( k \)-party protocol with error \( \frac{\epsilon}{2} \) and average communication complexity \( C \) can be turned into a protocol with error \( \epsilon \) and worst case communication complexity \( 4C \log(k) / \epsilon \).

\textbf{Proof.} Use one of the players as a relay. This increase the communication from \( C \) to \( 2C \log(k) \). We can then control the communication of the protocol just using the well-known following fact.

\textbf{Fact 39.} Any 2-party protocol with error \( \frac{\epsilon}{2} \) and average communication complexity \( C \) can be turned into a protocol with error \( \epsilon \) and worst case communication complexity \( \frac{2C}{\epsilon} \).

\section{Tight lower bounds for the parity function \( \text{Par} \)}

We now show how one can indeed use PIC to study multi-party communication and 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 \( \text{Par}_n^k \) is defined as follows. Each player \( i \) receives \( n \) bits \( (x^p_i)_{p \in [1,n]} \) and Player 1 has to output the bitwise XOR of the inputs \( \left( \bigoplus_{i=1}^k x^1_i, \bigoplus_{i=1}^k x^2_i, \ldots, \bigoplus_{i=1}^k x^n_i \right) \).

There is a simple private protocol for \( \text{Par}_n^k \) 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 his input with the message and sends \( x_2 \oplus x_1 \oplus r \) to Player 3. The players continue until Player 1 receives back the message \( x_k \oplus \ldots \oplus x_1 \oplus r \). He 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 see that information cost will never be able to provide lower bounds that scale with \( k \).

We now prove tight lower bounds for this problem using the measure of PIC. For this, we study a restricted class of protocols: we only consider protocols such that for any player \( i \), the sets \( \bar{S}_{\pi_I}^i \) and \( \bar{S}_{\pi_I}^i \) do not depend on the input \( x \) or the randomness. In other words, the structure of the protocol is fixed and independent of the input and randomness. Thus
we can assume that the round structure is synchronized among the players. Note that the private protocol we described above fits in this model.

Our lower bound for PIC of \( \text{Par}_k^0 \) is in fact proved for a wider class of protocols, where we allow that the player who outputs \( \oplus_{i=1}^k x_i^p \) be different for each coordinate \( p \) and also depend on the input.

**Theorem 40.** \( \text{PIC}_\mu^0(\text{Par}_k^0) \geq n(k - 1) \) where \( \mu \) is the uniform input distribution.

**Proof.** We use the uniform distribution \( \mu \) throughout the proof. Since we are looking at zero-error protocols, the public information cost is equal to the information cost of deterministic protocols. Let \( \pi \) be a protocol solving \( \text{Par}_k^0 \) for \( k \) players and \( n \) bits per player. For any \( i \), we define \( \Pi_i^0 \) to be the transcript of player \( i \), i.e. all messages he sends and receives. We split this transcript into the messages he sends and the messages he receives. Denote by \((M_i^l)_{l \geq 0}\) the set of messages sent by player \( i \) in the protocol \( \pi \), ordered by round and inside each round ordered by index of the recipient. Similarly, denote by \((M_j^l)_{l \geq 0}\) the set of messages received by player \( i \). Denote by \( j(i, l) \) the player receiving \( M_j^l \).

For any \( l_0 \), let \( M_i^{< l_0} \) be the random variable representing the so-far history, i.e. all messages to and from player \( i \) until the time of message \( M_i^{l_0} \). In a similar way, define \( M_i^{< l_0} \) to be the random variable representing the history of the messages to and from player \( i \) until the time of message \( M_i^l \). We also define a function \( l'(i, l) \) such that every message \( M_i^l \) sent by player \( i \) is received by player \( j(i, l) \) as \( M_j^{l'} \) where \( l' \) is a shortcut for \( l'(i, l) \) and \( j \) for \( j(i, l) \).

Our first goal is to prove that

\[
\text{PIC}_\mu^0(\pi) \geq \sum_i I(X_i; \Pi_i^0)
\]

Intuitively, this means that PIC at least takes into account the information each player leaks about his input to someone who has access to all messages that involve this player.

Note that we have for all \( i \)

\[
I(X_i; \Pi_i^0) = \sum_l I(X_i; M_i^l | M_i^{< l}) + \sum_l I(X_i; M_i^l | M_i^{< l}) = \sum_l I(X_i; M_i^l | M_i^{< l}),
\]

where we used the fact that every term of the second sum is 0, as for any \( l \), conditioned on \( M_i^{< l} \), \( X_i \) is independent of the variable \( M_i^l \).

Starting from the definition of PIC and using the chain rule, we can decompose it as a sum over all messages received in the protocol:

\[
\text{PIC}_\mu^0(\pi) = \sum_i \sum_l I(X_{-i}; M_i^l | M_i^{< l} X_i).
\]

We rearrange the sum by considering the messages from the point of view of the sender rather than the receiver.

\[
\text{PIC}_\mu^0(\pi) = \sum_i \sum_l I(X_{-j}; M_i^l | M_j^{< l} X_j).
\]

We will show that for any message \( M_i^l \), \( I(X_{-j}; M_i^l | M_j^{< l} X_j) \geq I(X_i; M_i^l | M_i^{< l}) \).

As \( M_i^l \) is determined by \( X_i \) and \( M_i^{< l} \), \( H(M_i^l | X_i M_i^{< l}) = 0 \), and we have

\[
I(X_i; M_i^l | M_i^{< l}) = H(M_i^l | M_i^{< l}),
\]

and similarly

\[
I(X_{-j}; M_i^l | M_j^{< l} X_j) = H(M_i^l | M_j^{< l} X_j).
\]
Thus \( I(X_i; M^{c\ominus}_j | M^{c\ominus}_j) \leq I(X_j; M^{c\ominus}_j | M^{c\ominus}_j X_j) \iff H(M^{c\ominus}_j | M^{c\ominus}_j X_j) \leq H(M^{c\ominus}_j | M^{c\ominus}_j X_j) \)

\[ \iff I(M^{c\ominus}_j; M^{c\ominus}_j) \geq I(M^{c\ominus}_j; M^{c\ominus}_j X_j). \]

We show that \( I(M^{c\ominus}_j; M^{c\ominus}_j) = I(M^{c\ominus}_j; M^{c\ominus}_j M^{c\ominus}_j X_j), \) which implies that the last inequality is true. For this we just need to check that \( I(M^{c\ominus}_j; M^{c\ominus}_j X_j | M^{c\ominus}_j) = 0. \) To see this, notice that given the value of \( M^{c\ominus}_j, M^{c\ominus}_j \) is determined by \( X_i \) and thus \( I(X_i; M^{c\ominus}_j X_j | M^{c\ominus}_j) = I(M^{c\ominus}_j; M^{c\ominus}_j X_j | M^{c\ominus}_j) \) and notice that \( I(X_i; M^{c\ominus}_j X_j | M^{c\ominus}_j) = 0 \) (cf. lemma 43 in the appendix).

We showed so far that \( \text{PIC}^0_p(\pi) \geq \sum_i I(X_i; \bar{\Pi}^1_i). \) Showing that \( I(X_i; \bar{\Pi}^1_i) \geq n(k-1) \) will conclude the proof.

Let \( x = (x^p_i) \in \{0, 1\}^{nk}, \) where \( x_i \) is the \( n \)-bit input of player \( i. \) For any \( p, \) let \( q^p(x) \) be the index of the first player able to compute \( \bigoplus_{i=1}^k x^p_i \) (formally, we say that a player is able to compute \( \bigoplus_{i=1}^k x^p_i \) at some time if the value of \( \bigoplus_{i=1}^k x^p_i \) is fixed given his input and his transcript until that time). For any \( i, \) define \( C_i(x) = \{ p, q^p(x) \neq i \}, \) which represents the coordinates of his input that player \( i \) is intuitively going to leak when the players are given input \( x, \) and let \( c_i(x) = |C_i(x)|. \)

We show that \( \forall i, H(X_i | \bar{\Pi}^1_i) = \mathbb{H}^{c^\ominus}_i(x) \leq n - c_i(x). \) Assume towards a contradiction that for some \( i, H(X_i | \bar{\Pi}^1_i) = \mathbb{H}^{c^\ominus}_i(x) > n - c_i(x). \) This implies that the number of possible values for \( X_i \) consistent with \( \bar{\Pi}^1_i = \mathbb{H}^{c^\ominus}_i(x) \) 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 the transcript is strictly less than \( c_i(x). \) Then there must exists \( x' \) such that

\[ \mathbb{H}^{c^\ominus}_i(x') = \mathbb{H}^{c^\ominus}_i(x), \]

\[ \exists p \in C_i(x) \text{ such that } x^p_i \neq x'^p_i. \]

Note that \( \mathbb{H}^{c^\ominus}_i(x') = \mathbb{H}^{c^\ominus}_i(x) \) implies (as a general property of communication protocols) that \( \mathbb{H}^{c^\ominus}_i(x) = \mathbb{H}^{c^\ominus}_i(x'_i, x_{-i}). \) As \( q^p(x) \neq i, \) this is a contradiction, since then player \( q^p(x) \) would output the same answer on \( x \) and \( (x'_i, x_{-i}), \) while \( \bigoplus_{j=1}^k x^p_j \neq x'^p_j \oplus \bigoplus_{j \neq i} x^p_j. \)

Thus \( \forall i, H(X_i | \bar{\Pi}^1_i) = \mathbb{H}^{c^\ominus}_i(x) \leq n - c_i(x) \) and \( H(X_i | \bar{\Pi}^1_i) \leq \mathbb{E}[n - c_i(x)] = n - \mathbb{E}[c_i(x)]. \) Thus \( I(X_i; \bar{\Pi}^1_i) \geq \mathbb{E}[c_i(x)]. \)

Summing over all \( i, \) we get \( \sum_i I(X_i; \bar{\Pi}^1_i) \geq \sum_i \mathbb{E}[c_i(x)] = \mathbb{E}[\sum_i c_i(x)] \) and since \( \sum_i c_i(x) = n(k-1) \) for any \( x, \) we get \( \sum_i I(X_i; \bar{\Pi}^1_i) \geq \mathbb{E}[n(k-1)] = n(k-1). \)

As \( \text{PIC}^0_p(\pi) \geq \sum_i I(X_i; \bar{\Pi}^1_i), \) we have shown that \( \text{PIC}^0_p(\pi) \geq n(k-1). \)

This result enables us to give a lower bound on the amount of randomness needed for the private computation of \( \text{Par}^n_k. \)

\textbf{Theorem 41.} The entropy in the private randomness of a private protocol for \( \text{Par}^n_k \) is at least \( \frac{k-2}{k-1} n. \)
Proof. For $\text{Par}_k^n$, where one player outputs the parity for each coordinate, we have $\sum_i H(f_i) = n$. Applying corollary 31, we get: $H(\Pi | XR^p) \geq \frac{k - 2}{k - 1}$.

Using Theorem 30, we can also lower bound the randomness one needs in order to have partially private protocols, for instance when the protocols is allowed to leak some limited amount of information about the inputs of the players.

8 Conclusions

In this paper we introduce a new information-theoretic measure, that we call PIC, for the study of multi-party computation protocols in the number-in-hand, peer-to-peer model. This is probably the most natural (distributed) computation model, and also closely related to the models used in the distributed algorithms community. Previous information-theoretic measures that were used successfully for the study of two-party computation protocols do not extend immediately to the multi-party case due to the fact that private protocols exist for any function in the multi-party setting [5, 18]. Our notion of PIC provides an alternative way of studying multi-party protocols, especially when one is interested in notions of privacy. Furthermore, PIC may yield tight results for certain functions, for which using other models, such as the coordinator model, would imply a loss of a logarithmic factor.

We define this measure in a computation model which, albeit being slightly restricted compared to the general asynchronous model, still allows private protocols, and is natural enough to apply to almost any distributed computation peer-to-peer protocol in the literature. We prove a number of properties of our new measure, PIC, and a number of connections between PIC and other complexity measures, such as the amount of randomness needed for private computation or the central notion of compression of communication.

Our work opens the way to a number of interesting directions for further work. A challenging direction would be to prove a tight lower bound for Disjointness in the message passing peer-to-peer model (without the loss of a logarithmic factor). An ambitious goal would be to prove a direct sum property for the Public Information Cost.

References


9 Technical statements

Lemma 42. In the conditions of the proof of theorem 30, \( I(R_{-i}; R_i \mid X \Pi R^p) = 0 \).

Proof. \( I(R_i; R_{-i} \mid X \Pi R^p) = H(R_i \mid X \Pi R^p) - H(R_i \mid R_{-i}X \Pi R^p) \)

\[ = -H(R_i \mid X R^p) + H(R_i \mid X \Pi R^p) + H(R_i \mid X R^p R_{-i}) - H(R_i \mid R_{-i}X \Pi R^p) \]

(As \( R_i, R_{-i}, R^p \) and \( X \) are independent)

\[ = -I(R_i; \Pi \mid X R^p) + I(R_i; \Pi \mid X R^p R_{-i}) \]

Thus we just need to prove that \( I(R_i; \Pi \mid X R^p R_{-i}) \leq I(R_i; \Pi \mid X R^p) \). From now on the proof is similar to the proof that the internal IC of a protocol is lower than its external IC (cf. [6]).

Let us write \( \Pi = \Pi^1 \ldots \Pi^t \) where the \( \pi^p \) represent the messages ordered by round, and let \( \Pi^{<p} = \Pi^1 \ldots \Pi^{p-1} \). Using the chain rule,

\[ I(R_i; \Pi \mid X R^p R_{-i}) = \sum_{p=1}^{t} I(R_i; \Pi^p \mid X R^p R_{-i} \Pi^{<p}) \]

\[ I(R_i; \Pi \mid X R^p) = \sum_{p=1}^{t} I(R_i; \Pi^p \mid X R^p \Pi^{<p}). \]

We prove the inequality \( \sum_{p=1}^{t} I(R_i; \Pi^p \mid X R^p R_{-i} \Pi^{<p}) \leq \sum_{p=1}^{t} I(R_i; \Pi^p \mid X R^p \Pi^{<p}) \) term-wise, for any \( p \). If the sender of \( \Pi^p \) is not player \( i \), as \( \Pi^p \) is a function of \( (X, R^p, R_{-i}, \Pi^{<p}) \), \( I(R_i; \Pi^p \mid X R^p R_{-i} \Pi^{<p}) = 0 \) and the inequality holds. Similarly, if \( \Pi^p \) is sent by player \( i \), \( I(R_{-i}; \Pi^p \mid X R^p R_{-i} \Pi^{<p}) = 0 \) and applying lemma 8, we get that \( I(R_i; \Pi^p \mid X R^p R_{-i} \Pi^{<p}) \leq I(R_i; \Pi^p \mid X R^p \Pi^{<p}). \) ◀

Lemma 43. In the conditions of the proof of theorem 40, \( I(X_i; M_i^{<\Pi^p} X_j \mid M_i^{<\Pi^p}) = 0 \).

Proof. Note that \( (M_i^{<\Pi^p}, X_j) \) is a function of \( (X_{-i}, M_i^{<\Pi^p}) \). The data processing inequality 7 implies that \( I(X_i; M_i^{<\Pi^p} X_j \mid M_i^{<\Pi^p}) \leq I(X_i; X_{-i} M_i^{<\Pi^p} \mid M_i^{<\Pi^p}) \) and thus \( I(X_i; M_i^{<\Pi^p} X_j \mid M_i^{<\Pi^p}) \leq I(X_i; X_{-i} \mid M_i^{<\Pi^p}). \)

We show that \( I(X_i; X_{-i} \mid M_i^{<\Pi^p}) = 0 \), which will conclude the proof. The situation is very similar to the one of lemma 42.

\[ I(X_i; X_{-i} \mid M_i^{<\Pi^p}) = H(X_i \mid M_i^{<\Pi^p}) - H(X_i \mid X_{-i} M_i^{<\Pi^p}) \]

\[ = -H(X_i) + H(X_i \mid M_i^{<\Pi^p}) + H(X_i \mid X_{-i}) - H(X_i \mid X_{-i} M_i^{<\Pi^p}) \] (\( X_i \), \( X_{-i} \) independent)

\[ = -I(X_i; M_i^{<\Pi^p}) + I(X_i; M_i^{<\Pi^p} \mid X_{-i}) \]

Thus we just need to prove that \( I(X_i; M_i^{<\Pi^p} \mid X_{-i}) \leq I(X_i; M_i^{<\Pi^p}) \). From now on the proof is similar to the proof that the internal IC of a protocol is lower than its external IC (cf. [6]).

Let us write \( M_i^{<\Pi^p} = M_1^{<\Pi^p} \ldots M_t^{<\Pi^p} \), and let \( M_i^{<\Pi^p} = M_1^{<\Pi^p} \ldots M_{p-1}^{<\Pi^p} \). Using the chain rule,

\[ I(X_i; M_i^{<\Pi^p} \mid X_{-i}) \leq I(X_i; M_i^{<\Pi^p}) \iff \sum_{p=1}^{t} I(X_i; M_p^{<\Pi^p} \mid X_{-i} M_i^{<\Pi^p}) \leq \sum_{p=1}^{t} I(X_i; M_p^{<\Pi^p} \mid M_i^{<\Pi^p}). \]

We prove the inequality term-wise, for any \( p \). If \( M_p^{<\Pi^p} \) is received by player \( i \), as \( M_p^{<\Pi^p} \) is a function of \( (X_{-i}, M_i^{<\Pi^p}) \), \( I(X_i; M_p^{<\Pi^p} \mid X_{-i} M_i^{<\Pi^p}) = 0 \) and the inequality holds. Similarly, if \( M_p^{<\Pi^p} \) is sent by player \( i \), \( I(X_{-i}; M_p^{<\Pi^p} \mid X_i M_i^{<\Pi^p}) = 0 \) and applying lemma 8, we get that \( I(X_i; M_p^{<\Pi^p} \mid X_{-i} M_i^{<\Pi^p}) \leq I(X_i; M_i^{<\Pi^p}). \) ◀