Privacy and Communication Complexity*

Eyal Kushilevitz†

Abstract

Each of two parties $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ holds an n-bit input, x and y, respectively. They wish to *privately* compute the value of $f(x,y)$. That is, $P_{\mathcal{X}}$ should not learn any additional information about y (in the information-theoretic sense) other than what follows from its input x and the function value, $f(x,y)$, and similarly, $P_{\mathcal{Y}}$ should not learn any additional information about x.
In this paper we consider two basic questions in the theory of private computations:

1. Which functions can be privately computed ?
2. What is the communication complexity of protocols that privately compute a function f (in the case that such protocols exist) ?

A complete combinatorial characterization of privately computable functions is given. We use this characterization to derive tight bounds on the rounds complexity of any privately computable function, and to design optimal private protocols that compute these functions.

We show that for every $1 \leq g(n) \leq 2 \cdot 2^n$ there are functions that can be privately computed with $g(n)$ -rounds of communication, but not with $g(n)-1$ -rounds of communication. This implies that the communication costs of private protocols can be exponentially higher than the communication costs of non-private protocols.

Interestingly, randomization helps neither to increase the set of privately computable functions, nor to improve the rounds complexity of these functions.

1 Introduction

The topic of this paper is private computations and their communication complexity. To exemplify the issue, we consider the "Two millionaires" problem presented by Yao [17]: Two millionaires wish to know who is the richer. However they want to do this in a way such that neither of them will receive any additional information about the other's wealth. Can the two millionaires solve their problem ?

The general question that we address is which functions of two arguments can be computed in such a way that no party learns any additional knowledge, other than what follows from

*An early version of this paper appeared in *Proc. of 30th FOCS*, 1989, pp. 416–421. Research supported by US-Israel Binational Science Foundation grant 88-00282.

†e-mail: eyalk@techunix.bitnet . Department of Computer Science, Technion, Haifa 32000, Israel.

the value of the function and its input. It turns out that the answer for this question heavily relies on the assumptions that are made regarding the computational power of the parties. One possible approach is to assume that the parties are limited to efficient computations (i.e. in polynomial-time). The issue of privacy, using this approach, was resolved in [19, 11], under (unproven) intractability assumptions.

Another possible approach is the information theoretic approach: The computational power of the parties is not restricted, and no intractability assumptions are made. Thus, the notion of privacy is much stronger. It is not only that the parties cannot obtain additional information using polynomial-time computations, but that such information cannot be obtained at all. This approach was studied in [5, 6]. It was shown that any function f of N arguments can be computed $\left\lfloor\frac{N-1}{2}\right\rfloor$-privately: No coalition of $t \leq \left\lfloor\frac{N-1}{2}\right\rfloor$ parties learns anything, other than the value of the function from the execution of the protocol (Note that for the case $N = 2$ this claim is not meaningful). On the other hand, for $t > \left\lfloor\frac{N-1}{2}\right\rfloor$, Ben-or, Goldwasser and Wigderson [5] showed that some functions cannot be computed t-privately. In [8] a complete characterization of the t-private Boolean functions, for $\left\lfloor\frac{N-1}{2}\right\rfloor < t \leq N$ is given. The basis for this characterization was a reduction from the multi-party case to the two-party case. In the two-party case it is shown that a *Boolean* function f is 1-private if and only if it can be expressed as $f(x,y) = h(x) \oplus g(y)$, where h and g are also Boolean functions, and $\oplus$ denotes exclusive-or [8].

In this paper we solve the question of privacy with respect to *arbitrary* two-argument functions: A *complete combinatorial characterization* of the privately computable functions is given. In particular, this provides a necessary condition for privacy in the general *multi-party* case. This characterization was independently found by Beaver [3].

A new facet of privacy, considered in this paper, is the communication complexity of computing functions privately. Two measures of complexity are considered in this paper: communication complexity (number of bits) and rounds complexity. The communication complexity of computing functions in a two-party system was extensively studied in previous works. Yao, for example, investigated the communication complexity of computing arbitrary Boolean functions in such a system (both in a deterministic model and a probabilistic model) [16, 18]. Tight $\Theta(n)$ bounds on the communication complexity of explicit functions and of random functions were given both for the case of deterministic protocols [16] and for randomized protocols [1, 7, 12]. Rounds complexity was studied by Papadimitriou and Sipser [15] and by Duris, Galil and Schnitger [9]. They showed that for certain functions there is an exponential gap between the number of bits that must be exchanged using k-round protocols and $k+1$ -round protocols.

We use the characterization of privately computable functions, to derive tight bounds on the communication complexity and rounds complexity of these functions. We show that the privately computable functions form a very dense rounds-complexity hierarchy. For every $1 \leq g(n) \leq 2 \cdot 2^n$ there exists a function which is privately computable by a $g(n)$-round protocol but cannot be privately computed by any $g(n) - 1$ -round protocol. In particular, certain functions require $2 \cdot 2^n$ communication rounds and $\Theta(2^n)$ bits to be privately computed. This should be contrasted with the fact that in a regular computation (that is, without privacy constraints), any function f can be computed using $O(n)$ bits and two communication rounds ($P_{\mathcal{X}}$ sends x; $P_{\mathcal{Y}}$ computes and sends $f(x,y)$). Comparing these two bounds we conclude that even when a function is privately computable, the "cost" of privacy may be exponentially larger.

In distributed computing, randomization often increases the computation power [13, 2], or significantly decreases computational costs [4, 10, 14]. Interestingly, this is not the case for private two-party computation. For every privately computable function f, we present a *deterministic* protocol, which privately computes f in an *optimal* number of rounds (and deterministically-optimal number of bits). We conclude that randomization helps neither to increase the set of privately computable functions, nor to improve the rounds complexity of privately computable functions.

The rest of this paper is organized as follows: In section 2 we present the model and the definitions of privacy. In section 3 we give the characterization of the privately computable functions, and in section 4 we deal with the communication complexity and rounds complexity of these functions. Finally, in section 5 we discuss some connections of the results with multi-party private protocols.

2 Preliminaries

2.1 Two-Party Protocols and Communication Complexity (Without Privacy)

In this section we define the model of two-party computation, and the notions of complexity which are used in this paper. These definitions are originally due to Yao [16].

The model consists of two parties, $P_{\mathcal{X}}$ - holding n-bit input x, and $P_{\mathcal{Y}}$ - holding n-bit input y. $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ wish to compute the value of $f(x,y)$ ($f : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1,\ldots,m-1\}$) using a probabilistic protocol. A *probabilistic protocol* [16] for computing $f(x,y)$ is a predetermined probabilistic program. That is, the parties are allowed to toss coins during their local computations. The parties are alternately sending messages to each other. The message q_i a party sends in the i-th round, is a function of its input, its coin tosses, and the messages it has received so far $(q_1,\ldots,q_{i-1})$. We assume that the parties are honest, that is, they follow their predetermined programs. The last message sent in the protocol $\mathcal{A}$ is assumed to contain the value of the function, and is denoted $\mathcal{A}(x,y)$. We say that the protocol $\mathcal{A}$ computes the function f, if

$$\forall x,y \in \{0,1\}^n : \quad Pr(\mathcal{A}(x,y) = f(x,y)) > \frac{1}{2}$$

where the probability is taken over the random coin tosses of the two parties.

The *communication string* passed in the protocol is the concatenation of all the messages $q_1 \cdot q_2 \cdot \ldots \cdot q_k$ sent in the course of the protocol. We assume that in every round, the set of all possible messages forms a prefix-free code. Thus, the communication string can be uniquely decomposed to its messages. It also enables a party which receives a message to recognize its end. The influence of this assumption on the communication complexity of the computation is bounded by a constant.

The *communication complexity* of a protocol $\mathcal{A}$, is the maximal number of bits transmitted during the execution of the protocol $\mathcal{A}$ (where the maximum is taken over all the possible inputs (x,y), and all the possible coin tosses). Similarly, the *rounds complexity* of a protocol $\mathcal{A}$, is the maximal number of rounds in any run of the protocol $\mathcal{A}$.

The *communication complexity* (*rounds complexity*) of a function f, is the minimum communication complexity (rounds complexity) over all protocols which compute f.

Remark 1: Since the last message of the protocol should contain the value of $f(x, y)$ it was assumed that if the image of f is of size m then the image is the set $\{0, 1, 2, \ldots, m-1\}$. If this is not the case we can use, in the last message, some encoding of $f(x, y)$ into the set $\{0, 1, \ldots, m-1\}$. Therefore the length of the last message is at most $\log_2 m$ bits (note that $m \leq 2^{2n}$ and therefore $\log_2 m \leq 2n$).
Remark 2: The assumption that $|x| = |y|$ is not essential for any of our results, and can be easily relaxed.

It is convenient to visualize any function $f : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0, 1, \ldots, m-1\}$ as a $2^n \times 2^n$ matrix with entries in $\{0, 1, \ldots, m-1\}$. We denote this matrix by M_f. Each row of M_f represents an input x held by $P_{\mathcal{X}}$, and each column of M_f represents an input y held by $P_{\mathcal{Y}}$. The entry (x, y) of the matrix M_f contains the value $f(x, y)$. A submatrix of M_f is called *monochromatic*, if f is constant over it.

2.2 The Privacy Constraints

In this section we formally define the notions of weak and strong privacy in a two-party distributed system. The definitions are taken from [8], and the definition of strong privacy is equivalent to the definition of [5].

Informally, a protocol is private if each party does not learn anything from the execution of the protocol other than the value of the function. This means that the parties do not gain even a probabilistic advantage over what they know by themselves.

Formally, we call a protocol *weakly private with respect to* $P_{\mathcal{X}}$ (Similar definition holds for $P_{\mathcal{Y}}$) if for every two inputs (x, y_1) and (x, y_2) satisfying $f(x, y_1) = f(x, y_2)$, and for every communication string s

$$Pr(s|(x, y_1)) = Pr(s|(x, y_2))$$

where the probability is taken over all the possible coin tosses of both parties.

Now, we make the notion of privacy stronger in two ways. First, we require that the protocols will not make errors, and second, we require that the parties not learn anything from the execution of the protocol even when taking into account their random coin tosses. Formally, we say that a protocol $\mathcal{A}$ is *strongly private with respect to* $P_{\mathcal{X}}$ (Similar definition holds for $P_{\mathcal{Y}}$) if $\mathcal{A}$ always computes the correct value of the function (that is $\mathcal{A}(x, y) = f(x, y)$), and if for every two inputs (x, y_1) and (x, y_2) satisfying $f(x, y_1) = f(x, y_2)$, every string of coin tosses $r_{\mathcal{X}}$ held by $P_{\mathcal{X}}$, and for every communication string s

$$Pr(s|r_{\mathcal{X}}, (x, y_1)) = Pr(s|r_{\mathcal{X}}, (x, y_2))$$

where the probability is taken over all the possible coin tosses of $P_{\mathcal{Y}}$.

A protocol $\mathcal{A}$ is called *weakly/strongly private* if it is weakly/strongly private with respect to both parties. A function f is called *weakly/strongly private* if there exists a weakly/strongly private protocol which computes it.

Finally, we remark that when dealing with the communication/rounds complexity of private functions, we consider only private protocols.

	y_1	y_2
x_1	0	0
x_2	0	1

	y_1	y_2	y_3
x_1	0	0	1
x_2	2	4	1
x_3	2	3	3

Figure 1: Forbidden Matrices

3 Characterization of Private Functions

In this section we give a complete combinatorial characterization of the privately computable functions. This characterization is used to derive a deterministic private protocol for any privately computable function. We start with some definitions:

Definition 1*:* *Let $M = C \times D$ be a matrix (C is the set of rows and D is the set of columns). The relation $\sim$ on the rows of the matrix M is defined as follows: $x_1, x_2 \in C$ satisfy $x_1 \sim x_2$ if there exists $y \in D$ such that $M_{x_1,y} = M_{x_2,y}$. The equivalence relation $\equiv$ on the rows of the matrix M is defined as the transitive closure of the relation $\sim$. That is, $x_1, x_2 \in C$ satisfy $x_1 \equiv x_2$ if there exist $z_1, z_2, \ldots, z_\ell \in C$ such that $x_1 \sim z_1 \sim z_2 \sim \ldots \sim z_\ell \sim x_2$. Similarly the relations $\sim$ and $\equiv$ are defined on the columns of the matrix.*

Definition 2*:* *A matrix M is called* forbidden *if it is not monochromatic, all its rows are equivalent, and all its columns are equivalent. That is, every $x_1, x_2 \in C$ satisfy $x_1 \equiv x_2$, and every $y_1, y_2 \in D$ satisfy $y_1 \equiv y_2$. (For examples of forbidden matrices see Figure 1).*

The first theorem claims that if a function f is represented by a matrix M_f which contains a forbidden submatrix, then f is not privately computable.

Theorem 1*:* *Let f be a function. If M_f contains a forbidden submatrix $M = C \times D$ then f is not (weakly) private.*

Proof: The idea of the proof is to show that any protocol $\mathcal{A}$, trying to compute f privately, fails (with "high" probability) in computing the correct value of the function for some inputs. Intuitively, this is done by showing that for any $x_1, x_2 \in C$, and any message q, the probability that $P_{\mathcal{X}}$ sends q on input x_1 equals to the probability that $P_{\mathcal{X}}$ sends q on input x_2. An analogous argument holds for $P_{\mathcal{Y}}$ with respect to any $y_1, y_2 \in D$. Thus, the probability distribution on communication strings is the same for any $(x, y) \in M$. Since M is not monochromatic, and since the protocol is required to compute the correct value of f for every input with probability greater than $\frac{1}{2}$, then for some of the inputs the protocol fails.

Let $C = \{x_1, \ldots, x_\ell\}$ and $D = \{y_1, \ldots, y_p\}$. Let $s = q_1 \cdot q_2 \cdot \ldots \cdot q_k$ be any communication string. We prove, by induction on the round number t, that for every $(x_i, y_j) \in M$

$$Pr(q_1, \ldots, q_t | (x_i, y_j)) = Pr(q_1, \ldots, q_t | (x_1, y_1)). \quad (1)$$

This implies that for every $(x, y) \in M$ the probability $Pr(s|(x, y))$ is the same.
The induction assumption implies that for every $(x, y) \in M$ the probability $Pr(q_1, \ldots, q_{t-1} | (x, y))$ is the same. If this probability is 0 then for every $(x, y) \in M$ we have $Pr(s|(x, y)) = 0$. Otherwise, we consider the t-th round $(1 \leq t \leq k)$ and assume, without loss of generality, that t is odd. Therefore the message q_t is sent by $P_{\mathcal{X}}$ (a similar argument holds for the messages of $P_{\mathcal{Y}}$). The submatrix $M = C \times D$ is forbidden, hence all the rows of M are equivalent. Therefore, we can also assume that the set C is ordered in a way such that for every i $(i \geq 2)$ there exists $j < i$ such that $x_i \sim x_j$. We now prove by induction on i that for every $1 \leq i \leq \ell$,

$$Pr(q_t | x_i, q_1, \ldots, q_{t-1}) = Pr(q_t | x_1, q_1, \ldots, q_{t-1}) \; . \quad (2)$$

Given $i > 1$ there is some $j < i$ with $x_i \sim x_j$. By the definition of $\sim$, there exists $y \in D$ such that $f(x_i, y) = f(x_j, y)$. According to the definition of (weak) privacy, this implies that for every communication string s, $Pr(s|(x_i, y)) = Pr(s|(x_j, y))$. In particular, $Pr(q_1, \ldots, q_t | (x_i, y)) = Pr(q_1, \ldots, q_t | (x_j, y))$. But,

$$Pr(q_1, \ldots, q_t | (x, y)) = Pr(q_1, \ldots, q_{t-1} | (x, y)) \cdot Pr(q_t | x, q_1, \ldots, q_{t-1}). \quad (3)$$

By the assumptions, $Pr(q_1, \ldots, q_{t-1} | (x_i, y)) = Pr(q_1, \ldots, q_{t-1} | (x_j, y)) \neq 0$. Therefore, by (3) and the privacy condition, $Pr(q_t | x_i, q_1, \ldots, q_{t-1}) = Pr(q_t | x_j, q_1, \ldots, q_{t-1})$, which implies by induction hypothesis (2) that $Pr(q_t | x_i, q_1, \ldots, q_{t-1}) = Pr(q_t | x_1, q_1, \ldots, q_{t-1})$. This completes the proof of claim (2).
By induction hypothesis (1) and claim (2) we get that for every $(x_i, y_j) \in M$

$$Pr(q_1, \ldots, q_t | (x_i, y_j)) = Pr(q_1, \ldots, q_t | (x_1, y_1)) \; .$$

This completes the proof of (1). Therefore, for every input $(x_i, y_j) \in M$ the probability distributions of communication strings sent on these inputs are all equal. In particular, for every $(x_i, y_j) \in M$ the last message of the communication string, which is $\mathcal{A}(x_i, y_j)$, is distributed in the same way. On the other hand, the correctness of $\mathcal{A}$ implies that for every (x_i, y_j),

$$Pr(\mathcal{A}(x_i, y_j) = f(x_i, y_j)) > \frac{1}{2} \; .$$

Thus, the same function value is computed for all the inputs in M, in contradiction to the assumption that M is not monochromatic. $\square$

The special case of theorem 1, where M is a 2×2 submatrix, is useful for proving that particular functions are not private. It says that if there exist $x_1, x_2, y_1, y_2 \in \{0,1\}^n$ such that $f(x_1, y_1) = f(x_1, y_2) = f(x_2, y_1) = a$, and $f(x_2, y_2) \neq a$ then f is not private. We can now prove, for example, that the "greater equal" function ($f(x, y) = 1 \iff x \geq y$) is not private. It is enough to observe that for every integer a we have $f(a-1, a) = f(a-1, a+1) = f(a, a+1) = 0$ but $f(a, a) = 1$. Note that finding a private protocol for the "greater than" function is actually solving the "Two millionaires" problem. We remark that in the computational model this problem can be solved [17].

We will now prove that the condition of theorem 1 is not only necessary but also sufficient. Namely, if the matrix M_f does not contain a forbidden submatrix then f is privately computable. Before proving this claim, we introduce some new definitions.

Definition 3: *A matrix $C \times D$ is called* rows decomposable *if there exist non-empty sets $C_1, C_2, \ldots, C_t$ ($t \geq 2$) such that:*

1. *$C_1, C_2, \ldots, C_t$ are a partition of C. That is, $\bigcup_{i=1}^{t} C_i = C$ and $\forall i \neq j : C_i \cap C_j = \emptyset$.*
2. *For every $x_1, x_2 \in C$, if $x_1 \sim x_2$ then x_1 and x_2 are in the same C_i.*

Similarly, one can define when a matrix $C \times D$ is *columns decomposable* . When we say *the optimal rows (columns) decomposition* of a matrix we mean the rows (columns) decomposition which maximizes t (the number of sets). Note that the optimal decomposition is unique. This is because the sets of the optimal decomposition are exactly the equivalence classes of the relation $\equiv$ (In general, the sets of any decomposition are unions of equivalence classes of $\equiv$).

Definition 4: *A matrix $C \times D$ is called* decomposable *if one of the following conditions hold:*

1. *$C \times D$ is monochromatic.*
2. *$C \times D$ is rows decomposable to sub-matrices $C_1 \times D, C_2 \times D, \ldots, C_t \times D$ which are all decomposable.*
3. *$C \times D$ is columns decomposable to sub-matrices $C \times D_1, C \times D_2, \ldots, C \times D_t$ which are all decomposable.*

An example of a decomposable matrix is given in Figure 2 (section 4).

Given a matrix M it is easy to check whether it is decomposable, or rows (columns) decomposable, and to find the optimal decomposition. Such an algorithm will be needed for the design of efficient private protocols. For example, the following algorithm checks if M ($n \times m$ matrix) is rows decomposable and finds the optimal rows decomposition:

1. For every row $1 \leq i \leq n$ create a set $C_i = \{i\}$
2. If there exist $x_1 \in C_i, x_2 \in C_j$ ($i \neq j$) and y such that $f(x_1, y) = f(x_2, y)$ (i.e. $x_1 \sim x_2$) then $C_i = C_i \cup C_j$ (that is unite C_i and C_j to a single set) and return to step (2).
3. If the number of sets is 1 - there is no rows decomposition. Else the current sets are the optimal rows decomposition of M.

In order to prove the sufficiency of our condition we will use the following lemma which claims that a matrix is decomposable if and only if it does not contain a forbidden submatrix.

Lemma 1: *Let $M = C \times D$. M is decomposable iff for every submatrix M' of M, M' is not forbidden.*

Proof: If M contains a forbidden submatrix M' then by induction we can show that the rows and columns of this submatrix always remain in the same submatrix of the decomposition. As a decomposition must end with monochromatic submatrices, M is not decomposable. Conversely, if M is not decomposable then we have a submatrix M' which is not monochromatic, not rows decomposable and not columns decomposable. This means that M' is a forbidden submatrix of M. □

Theorem 2: *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1,\ldots,m-1\}$ be an arbitrary function. If M_f does not contain a forbidden submatrix then f is (strongly) private.*

Proof: If M_f does not contain a forbidden submatrix then according to lemma 1, M_f is decomposable. We use the optimal decomposition of M_f for presenting a deterministic private protocol $\mathcal{A}_f$ which computes f. We assume that the two parties have a common way of numbering the sets in the decomposition. The protocol is as follows:
Protocol $\mathcal{A}_f$:

1. $P_\mathcal{X}$ and $P_\mathcal{Y}$ both set $C = D = \{0, 1, \ldots, 2^n - 1\}$
2. While $C \times D$ is not monochromatic:
   (a) $P_\mathcal{X}$ sends the j such that $x \in C_j$ in the optimal rows decomposition of $C \times D$. Both $P_\mathcal{X}$ and $P_\mathcal{Y}$ set $C = C_j$.
   (b) $P_\mathcal{Y}$ sends the j such that $y \in D_j$ in the optimal columns decomposition of $C \times D$. Both $P_\mathcal{X}$ and $P_\mathcal{Y}$ set $D = D_j$.
3. $P_\mathcal{X}$ sends the constant value in $C \times D$ as the value of $f(x, y)$.

From the definition of a decomposable matrix it is clear that during this algorithm $C \times D$ is always decomposable, and that the protocol $\mathcal{A}_f$ terminates. Since the input (x, y) always belongs to $C \times D$ then it follows that $\mathcal{A}_f$ terminates with the correct value of $f(x, y)$. (Note that the first time step (2a) is performed, it is possible that $C \times D$ is not rows decomposable. In such a case $C \times D$ must be columns decomposable, and we can start from step (2b). From this point on, since we use the optimal decomposition, $C \times D$ is rows decomposable and columns decomposable alternately). Since $\mathcal{A}_f$ is deterministic then for every input (x, y), the communication string is unique. Therefore, for showing that $\mathcal{A}_f$ is (strongly) private, one should verify that for every two inputs (x, y_1) and (x, y_2) that satisfy $f(x, y_1) = f(x, y_2)$ the same communication string is exchanged. This is true since the messages sent in step (2a) depend only on x, while according to the definition of columns decomposition, the messages sent in step (2b), are uniquely determined by x and the function value (Note that $y_1 \sim y_2$ in every submatrix $C \times D$ that contains (x, y_1) and (x, y_2), and thus they will never be decomposed). Similarly, we can verify that for every two inputs (x_1, y) and (x_2, y) that satisfy $f(x_1, y) = f(x_2, y)$, the same communication string is exchanged. □

From theorems 1 and 2 we get the following corollaries:

Corollary 3: *If f is weakly private then f is also strongly private.*

If f is weakly private then, according to theorem 1, M_f does not contain a forbidden submatrix. Therefore, according to theorem 2, f is strongly private. In particular this means that the ability of the protocols to make errors does not help to achieve privacy.

Corollary 4: *If f is private then it can be privately computed using a deterministic protocol.*

According to the proof of theorem 2 every private function is privately computable using a deterministic protocol. That is, randomization does not help to achieve privacy in the two-party case. We emphasize that this is not the case for multi-party protocols (see [5, 8]).

4 The Communication Complexity of Private Functions

In this section we deal with the communication complexity of private functions. We show a correspondence between *every* protocol $\mathcal{A}$, which computes a function f privately, and a decomposition of the matrix M_f. This correspondence is used for proving lower bounds on the number of rounds, and the number of communication strings needed for computing f privately.

It may be convenient to look at the decomposition of a matrix M, as a tree containing submatrices in its nodes. The root (level 0) of the tree contains the given matrix M. The children of a node v in an even level of the tree, contain a rows decomposition of the submatrix in v, while the children of a node in an odd level contain a columns decomposition of the submatrix. All the leaves of the tree are monochromatic submatrices. Such a tree is called a *decomposition tree* of the matrix M. The tree corresponding to the optimal decomposition is called the *optimal decomposition tree*. (We assume without loss of generality that M itself is rows decomposable. Otherwise in the even levels we should use columns decomposition and in the odd levels rows decomposition).

Following Yao [16], we visualize every protocol that computes f as a decision tree. We show that the decision tree of any private protocol, is also a decomposition tree. Given $\mathcal{A}$, a (weakly) private protocol for computing f, we describe how to create a decomposition tree that corresponds to $\mathcal{A}$. The root (level 0) contains the matrix M_f. Let v be a node in level ℓ of the tree containing a submatrix $M = C \times D$. We define its children as follows: Assume that the message in round $\ell + 1$ of the protocol should be sent by $P_{\mathcal{X}}$ (a similar definition holds for the case where the next message should be sent by $P_{\mathcal{Y}}$). Every $x_1, x_2 \in C$, for which $P_{\mathcal{X}}$ behaves "similarly" will be in the same C_i, where "similarly" means that for every message $q_{\ell+1}$ and for every sequence of messages $q_1, q_2, \ldots, q_\ell$ passed in previous rounds (with a positive probability):

$$Pr(q_{\ell+1}|x_1, q_1, q_2, \ldots, q_\ell) = Pr(q_{\ell+1}|x_2, q_1, q_2, \ldots, q_\ell).$$

Since it is given that $\mathcal{A}$ is (weakly) private, then $f(x_1, y) = f(x_2, y)$ implies that for every communication string s we have $Pr(s|(x_1, y)) = Pr(s|(x_2, y))$. Thus, $Pr(q_{\ell+1}|x_1, q_1, q_2, \ldots, q_\ell)$ is equal to $Pr(q_{\ell+1}|x_2, q_1, q_2, \ldots, q_\ell)$. Therefore, if $x_1 \sim x_2$ then x_1 and x_2 are in the same C_i, and the conditions for rows decomposition hold. Finally, we have to show that the leaves contain monochromatic submatrices. This follows from the fact that the same probability distribution of messages is sent for every element of the submatrix, and that for every input the protocol should compute the correct value of the function with probability greater then $\frac{1}{2}$. (Note that if the protocol $\mathcal{A}$ is not efficient it is possible that there are rounds that give decomposition with $t = 1$. In such a case we can omit these rounds).

Going on with this correspondence, it is easy to see that the number of communication rounds in the protocol is equal to the depth of the corresponding decomposition tree. To derive our lower bounds we use this equality together with the next lemma, which claims that the optimal decomposition tree has the minimal depth.

Lemma 2*:* *Let M be a decomposable matrix. Let T_{opt} be the optimal decomposition tree for M. For every T, a decomposition tree of M, the depth of T_{opt} is less than or equal to the depth of T.*

Proof: Given a non-optimal decomposition tree T, we can use the following bottom-up process in order to get the optimal tree T_{opt} without increasing the depth: In each step find an internal node v, for which the decomposition is not optimal, but the decomposition in any node of its subtree is optimal. Changing the decomposition in the node v to the optimal one, does not increase the depth of the tree (and may even decrease it).
More formally, let v be an internal node of T as above and denote by $C_v \times D_v$ the matrix corresponding to v. Assume without loss of generality that in v we have rows decomposition. That is, C_v is decomposed to $C_1, C_2, \ldots, C_t$. Since the decomposition is not optimal then there exists a set C_i in the decomposition that can be replaced by C_i^1 and C_i^2 (i.e. $C_1, \ldots, C_{i-1}, C_i^1, C_i^2, C_{i+1}, \ldots, C_t$ are the sets of a row decomposition of $C_v \times D_v$). Let $C \times D$ be the matrix in any node of the subtree corresponding to $C_i \times D_v$, then in the subtree corresponding to $C_i^j \times D_v$ $(j = 1, 2)$ this matrix will be replaced by $(C \cap C_i^j) \times D$. In the case that this matrix is monochromatic then its subtree will be omitted. It is easy to verify that the modified tree is indeed a decomposition tree, and clearly the depth is not increased. $\Box$

Corollary 5*:* *For every private function f, the protocol $\mathcal{A}_f$ (described in the proof of theorem 2) achieves the optimal number of rounds.*

This is because every private protocol that computes f corresponds to a decomposition tree. According to Lemma 2, the optimal decomposition tree, which is exactly the tree corresponding to $\mathcal{A}_f$, has the minimal depth. Note that Lemma 2 implies that $\mathcal{A}_f$ is optimal in a very strong sense: the depth of *every* input (x, y) in the corresponding tree is minimal, and not only of the worst-case input.

Corollary 6*:* *For every $1 \leq g(n) \leq 2 \cdot (2^n - 1)$ there exists a function f which is privately computable using a $g(n)$-round protocol but is not privately computable using any $g(n)-1$ -round protocol.*

This corollary implies the existence of a rounds-complexity hierarchy for privately computable functions. To prove it, we present below a function that is privately computable using a $2 \cdot (2^n - 1)$ -round protocol, but is not privately computable using any $2 \cdot (2^n - 1) - 1$ -round protocol. This example can be easily generalized to any $1 \leq g(n) \leq 2 \cdot (2^n - 1)$. We consider the following function f (see M_f in Figure 2):

$$f(x, y) = \begin{cases} 2\text{x} & \text{if } x \leq y \\ 2\text{y+1} & \text{if } x > y \end{cases}$$

The optimal decomposition of M_f gives us the following protocol $\mathcal{A}_f$:
For $i = 0, 1, 2, \ldots$

f	0	1	2	3	$\cdots$	2^n-2	2^n-1
0	0	0	0	0	$\cdots$	0	0
1	1	2	2	2	$\cdots$	2	2
2	1	3	4	4	$\cdots$	4	4
3	1	3	5	6	$\cdots$	6	6
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
2^n-2	1	3	5	7	$\cdots$	$2^{n+1}-4$	$2^{n+1}-4$
2^n-1	1	3	5	7	$\cdots$	$2^{n+1}-3$	$2^{n+1}-2$

Figure 2: Forbidden Matrices

1. $P_{\mathcal{X}}$ compares its input, x, with i. If $x = i$ it sends $f(x, y) = 2 \cdot i$ and the protocol is terminated. Otherwise it sends a GO-ON message (one bit).

2. $P_{\mathcal{Y}}$ compares its input, y, with i. If $y = i$ it sends $f(x, y) = 2 \cdot i + 1$ and the protocol is terminated. Otherwise it sends a GO-ON message (one bit).

It is easy to see that this protocol requires $2 \cdot (2^n - 1)$ rounds (and $\Theta(2^n)$ bits) for the input $(2^n - 1, 2^n - 1)$. Moreover, the average number of rounds (and bits) is also $\Theta(2^n)$. Note that in a regular computation every function f can be computed using at most $n + \log |range(f)|$ bits and two communication rounds (Simply, $P_{\mathcal{X}}$ sends x and $P_{\mathcal{Y}}$ computes and sends $f(x, y)$). Since in every round of the optimal protocol we decompose our matrix into at least two sub-matrices, then the function f defined above is the worst function. In other words, every private function f can be privately computable using at most $2 \cdot (2^n - 1)$ rounds.

Corollary 7: *Let m be the size of f's range (i.e. $f : \{0,1\}^n \times \{0,1\}^n \to \{0, 1, \ldots, m-1\}$) then the rounds complexity of f is bounded above by $2 \cdot (m - 1)$.*

Here again we use the optimality of $\mathcal{A}_f$ to claim that in every round the current submatrix is decomposed into at least two sub-matrices. In addition, in a rows decomposition, for every row x all the y's such that $M_{x,y}$ has the same value, belong to the same D_i in the decomposition. (Similar property holds for columns decomposition). Thus, if the original matrix contains m different values then (by a simple induction) we can show that after $2 \cdot i$ rounds of the protocol, every row and every column of the submatrix $C \times D$ contains at most $m - i$ different values. Therefore the number of rounds is bounded by $2 \cdot (m - 1)$.

In order to give lower bounds on the communication complexity of privately computable functions we present the following claim which relates rounds complexity to communication complexity in private computations.

Claim 1: *Let f be a privately computable function. Denote its communication complexity by $C_{private}(f)$ and its rounds complexity by $R_{private}(f)$. Then the following holds:*

$$R_{private}(f) \leq C_{private}(f) \leq n \cdot R_{private}(f)$$

Proof: $R_{private}(f) \leq C_{private}(f)$ since in any protocol the parties send at least one bit per round. On the other hand, for every privately computable function f the bound $R_{private}(f)$ is achieved by the protocol $\mathcal{A}_f$. In any round of this protocol the parties send a message with size at most n. (Note that in steps (2a) and (2b) of $\mathcal{A}_f$ we may save one bit per round by omitting the most significant bit which is always '1'). The claim follows. $\Box$

Corollary 8*: Randomization does not help to reduce the number of rounds needed in private computations.*

This is true since the protocol $\mathcal{A}_f$ is a deterministic protocol. We remark that $\mathcal{A}_f$ besides being optimal in its rounds complexity, is also optimal in its communication complexity among the deterministic protocols. To prove that we observe that if $\mathcal{A}$ is a deterministic protocol then the number of different communication strings is equal to the number of leaves in the decomposition tree that corresponds to $\mathcal{A}$. We conclude not only that randomization does not help to compute more functions privately, but also that it does not help to reduce the number of rounds needed in private computations.

5 Discussion

As it is pointed out in [8], the characterization of two-party private functions can be used to give necessary conditions for privacy in a multi-party scenario. That is, $f(x_1, x_2, \ldots, x_N)$ is t-private, for $t \geq \left\lfloor \frac{N-1}{2} \right\rfloor + 1$, if for *any* partition of the N parties into two sets of participants T and $\bar{T}$, where T is of size t, the two-argument function $\hat{f}$ defined as

$$\hat{f}(\{x_i\}_{i \in T}, \{x_i\}_{i \in \bar{T}}) \stackrel{def}{=} f(x_1, x_2, \ldots, x_N)$$

is private.

Using the above observation and Theorem 1, we get from any such partition a necessary condition for f to be t-private ($t \geq \left\lfloor \frac{N-1}{2} \right\rfloor + 1$). In addition, from any such partition and using Corollary 5, we get lower bounds on the rounds complexity and the communication complexity of computing f t-privately.

The problem of giving an exact characterization of the t-private (general) functions, for $\left\lfloor \frac{N-1}{2} \right\rfloor < t \leq N$ is still open.

6 Acknowledgments

I would like to thank Benny Chor and Oded Goldreich for helpful discussions on the topics of this paper, and Avi Wigderson for asking the question of corollary 7. I would also like to thank Shai Ben-David, Hugo Krawczyk, Alon Lavie, Lior Moscovici and Dan Simon for their useful comments on this paper.

References

[1] Alon N., P. Frankl, and V. Rodl, "Geometric Realization of Set Systems and Probabilistic Communication Complexity", *Proc. of 26th FOCS*, pp. 277-280, 1985.

[2] Ben-or M., "Another Advantage of Free Choice: Complete Asynchronous Agreement Protocols", *Proc. of 2nd PODC*, pp. 27-30, 1982.

[3] Beaver D., "Perfect Privacy for Two-Party Protocols", TR-11-89, Harvard University, 1989.

[4] Bracha G., "An $O(\log n)$ Expected Rounds Randomized Byzantine Generals Protocol", *Proc. of 17th STOC*, pp. 316-326, 1985.

[5] Ben-or M., S. Goldwasser, and A. Wigderson, "Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation", *Proc. of 20th STOC*, pp. 1-10, 1988.

[6] Chaum D., C. Crepeau, and I. Damgard, "Multiparty Unconditionally Secure Protocols", *Proc. of 20th STOC*, pp. 11-19, 1988.

[7] Chor B., and O. Goldreich, "Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity", *SIAM J. Computing*, Vol 17, No 2, 1988, pp. 230-261.

[8] Chor B., and E. Kushilevitz, "A Zero-One Law for Boolean Privacy", *SIAM J. Discrete Math.*, Vol 4, No 1, 1991, pp. 36-47.

[9] Duris P., Z. Galil, and G. Schnitger, "Lower Bounds on Communication Complexity", *Proc. of 16th STOC*, pp. 81-91, 1984.

[10] Feldman P., and S. Micali, "Optimal Algorithms for Byzantine Agreement", *Proc. of 20th STOC*, pp. 148-161, 1988.

[11] Goldreich, O., S. Micali, and A. Wigderson, "How to Play Any Mental Game", *Proc. of 19th STOC*, pp. 218-229, 1987.

[12] El-Gamal, A., and A. Orlitsky, "Randomized Communication Complexity", preprint, 1985.

[13] Lehman, D., and M.O. Rabin, "On the Advantage of Free Choice: A symmetric and Fully Distributed Solution to the Dining Philosophers Problem", *Proc. of 8th POPL*, pp. 133-138, 1981.

[14] Mehlhorn, K., and E. Schmidt, "Las-Vegas is better than Determinism in VLSI and Distributed Computing", *Proc. of 14th STOC*, pp. 330-337, 1982.

[15] Papadimitriou, C., and M. Sipser, "Communication Complexity", *JCSS*, 28 (1984), pp. 260-269.

[16] Yao, A. C., "Some Complexity Questions Related to Distributed Computing", *Proc. of 11th STOC*, pp. 209-213, 1979.

[17] Yao, A. C., "Protocols for Secure Computations", *Proc. of 23th FOCS*, pp. 160-164, 1982.

[18] Yao, A. C., "Lower Bounds by Probabilistic Arguments", *Proc. of 24th FOCS*, pp. 420-428, 1983.

[19] Yao, A. C., "How to Generate and Exchange Secrets", *Proc. of 27th FOCS*, pp. 162-167, 1986.