Private Computations Over the Integers*

Benny Chor[†] Mihály Geréb-Graus[‡] Eyal Kushilevitz[§]

Abstract

The subject of this work is the possibility of private distributed computations of n-argument functions defined over the integers. A function f is t-private if there exists a protocol for computing f, so that no coalition of at most t participants can infer any additional information from the execution of the protocol. It is known that over *finite* domains, every function can be computed $\lfloor (n-1)/2 \rfloor$-privately. Some functions, like addition, are even n-private.

We prove that this result cannot be extended to infinite domains. The possibility of privately computing f is shown to be closely related to the *communication complexity* of f. Using this relation, we show, for example, that n-argument addition is $\lfloor (n-1)/2 \rfloor$-private over the non-negative integers, but not even 1-private over all the integers.

Finally, a complete characterization of t-private *Boolean* functions over countable domains is given. A Boolean function is 1-private if and only if its communication complexity is *bounded*. This characterization enables us to prove that every Boolean function falls into one of the following three categories: It is either n-private, $\lfloor (n-1)/2 \rfloor$-private but not $\lceil n/2 \rceil$-private, or not 1-private.
Keywords: private distributed computations, communication complexity
AMS (MOS): 94A15, 94A60, 68R05

1 Introduction

A set of $n \geq 3$ computationally unbounded parties, each holding an input x_i taken from a domain D, wishes to cooperate in distributively computing the value $f(x_1, x_2, ..., x_n)$ of a predetermined function f. These parties are honest, namely they all follow their prescribed protocol. They communicate over a complete point-to-point communication network, where eavesdropping is not possible. A function f is called *t-private* if there is a communication protocol for computing f, so that no coalition of at most t participants gets any additional information from the execution of the protocol. Ben-Or, Goldwasser, and Wigderson [5], and

*An early version of this paper appeared in *Proc. of 31th FOCS*, 1990, pp. 335–344. Research supported by US-Israel Binational Science Foundation grant 88-00282.

†Department of Computer Science, Technion, Haifa 32000, Israel. e-mail: benny@cs.technion.ac.il . Part of this research was done while visiting the Computer Science Department in the University of Toronto.

‡Department of Computer Science, Tufts University, Medford, MA 02155. e-mail: gereb@cs.tufts.edu .

§Department of Computer Science, Technion, Haifa 32000, Israel. e-mail: eyalk@cs.technion.ac.il .

Chaum, Crepeau, and Damgard [8], have shown that if the domain D of f is *finite*, then f is $\lfloor (n-1)/2 \rfloor$-private. Specific functions, like addition, are even n-private over finite domains [6], while certain functions, like Boolean OR, are not $\lceil n/2 \rceil$-private[1] [5]. However, functions of interest are typically defined not over finite domains, but rather over all strings, over the integers, or more generally over some countable domain. To apply the protocol of [5, 8], one has to (implicitly or explicitly) assume an upper bound on inputs size. If the bound does not hold then the protocol, which depends not only on f, but also on the size of the domain, has to be adjusted. This adjustment amounts to revealing additional information on the magnitude of the inputs.

The question we deal with in this work is private computations of functions defined over countable domains. In other words, is there a private protocol for computing f which can be applied to all inputs, regardless of their sizes. Before going any further, we remark that the $\lfloor (n-1)/2 \rfloor$-private protocol of [5, 8] cannot be used in the infinite domain. Its first step is to apply a secret-sharing scheme to every input. Secret sharing schemes strongly rely on the finiteness of the domain, and indeed do not exist over countable domains [7, 11].

We show that the privacy of f is closely related to the *communication complexity* of f, as defined by Yao [17], namely to the number of bits that need to be communicated among n parties in order to compute f (using any protocol, not necessarily a private one). The connection between these two notions enables us to show that a wide class of functions are $\lfloor (n-1)/2 \rfloor$-private. This class includes, among others, maximum, minimum, addition over the non-negative integers, and multiplication over all integers. The relations to communication complexity are further used in our impossibility results. For example, we show that over the (negative and non-negative) integers, addition is not even 1-private. That is, there is no protocol which computes the sum of n integers and preserves privacy even with respect to single participants. This contrasts with the $\lfloor (n-1)/2 \rfloor$-privacy of addition over the non-negative integers, and the n-privacy of addition over any finite domain.

We give a complete characterization of private Boolean functions: A Boolean function is 1-private if and only if its communication complexity is *bounded*. As many Boolean functions have unbounded communication complexity, this proves that there are Boolean functions which are not even 1-private. Furthermore, we show that if a Boolean f is 1-private, then it is also $\lfloor (n-1)/2 \rfloor$-private. In [10] it was shown that if a Boolean f is $\lceil n/2 \rceil$-private, then it is also n-private. Hence there is a three level privacy hierarchy for Boolean functions: Every Boolean function (defined over a countable domain) is either n-private, $\lfloor (n-1)/2 \rfloor$-private but not $\lceil n/2 \rceil$-private, or not 1-private.

The remainder of this paper is organized as follows: In section 2 we describe the model and give the needed definitions. In section 3 we apply communication complexity arguments to produce private protocols for various functions. Section 4 contains the impossibility results for the functions addition and integer gcd. In section 5 we prove our characterization for the Boolean case, and its implications. Section 6 introduces a model of non-terminating protocols which compute f "in the limit", and demonstrate some intriguing private protocols in this model. Finally, section 7 contains concluding remarks and open problems.

[1]The study of private distributed computing was initiated in [18, 13]. The model used there is different – the communication lines are not secure against eavesdropping, and on the other hand the participants are computationally bounded, so cryptographic techniques can be employed.

2 Model and Definitions

The system consists of a synchronous network of n computationally unbounded parties $P_1, P_2, \ldots, P_n$. Each pair of parties is connected by a secure (no eavesdropping) and reliable communication channel. At the beginning of an execution, each party P_i has an input $x_i \in \{0,1\}^*$ (no probability space is associated with the inputs). In addition, each party has a random input r_i taken from a source of randomness R_i (the random inputs are independent). The parties wish to compute the value of a function $f(x_1, x_2, \ldots, x_n)$. To this end, they exchange messages as prescribed by a protocol $\mathcal{F}$. The parties are honest, namely they follow the protocol $\mathcal{F}$, and there are no failures of any kind. Messages are sent in rounds, where in each round every processor sends a (non-empty) message to every other processor. The protocol specifies n functions, one per party, which determine the messages sent by each party. The arguments to the messages producing function of each processor are its input, its random input, the round number, the messages it received so far, and the identity of the receiver.

The communication S sent in an execution of the protocol is the concatenation of all messages sent in the execution, parsed according to sender, receiver, and round number. For a subset $T \subseteq \{1, 2, \ldots, n\}$ we denote by S_T the communication S with the exception of messages sent between parties in $\bar{T}$. The communication length of the protocol $\mathcal{F}$ on inputs $x_1, \ldots, x_n$ is the maximum length of all communications S, sent in $\mathcal{F}$ on these inputs, over all random inputs. We say that a communication string S, parsed as above, is *consistent* with the protocol $\mathcal{F}$, party P_i and input x_i if the following holds: There is a positive probability that if P_i has input x_i and receives the messages in the string S that are destined to it, P_i will send messages identical to those messages in S that emanate from P_i. (Notice that to be consistent with just one party, the communication string need not be an actual string sent in some execution.)

We say that a protocol $\mathcal{F}$ computes the function f with *ε-advantage* ($0 < \varepsilon \leq \frac{1}{2}$) if for every input $\vec{x} = (x_1, \ldots, x_n)$ the last message in the protocol, is an identical message sent by party P_1 to all parties, indicating that this is the last message in the execution, and containing a value $\mathcal{F}(\vec{x}, \vec{r})$ which satisfies

$$Pr_{\vec{r}}\left(\mathcal{F}(\vec{x}, \vec{r}) = f(\vec{x})\right) \;\geq\; \frac{1}{2} + \varepsilon \; .$$

Denote by $p_k(\vec{x})$ the probability that more than k bits are exchanged between the parties during the execution of the protocol $\mathcal{F}$ with input $\vec{x}$ (the probability space is over the random inputs of all parties). The protocol is *terminating* if for every input $\vec{x}$

$$\lim_{k \to \infty} p_k(\vec{x}) = 0 \; .$$

We say that a coalition (i.e. a set of parties) T *does not learn any additional information* (other than what follows from its input and the function value) from the execution of a randomized protocol $\mathcal{F}$, which computes f, if the following holds: For every two inputs $\vec{x}, \vec{y} \in (\{0,1\}^*)^n$ that agree in their T entries (i.e. $\forall i \in T : x_i = y_i$) and satisfy $f(\vec{x}) = f(\vec{y})$, the messages passed between T and $\bar{T}$ are identically distributed. That is, for every communication S,

$$Pr_{\vec{r}}(S_T | \vec{x}) = Pr_{\vec{r}}(S_T | \vec{y}) \; ,$$

where the probability space is over the random inputs of all parties.

We say that a protocol $\mathcal{F}$ for computing f is *t-private* if any coalition T of at most t parties does not learn any additional information from the execution of the protocol. We say that a function f is *t-private* if there exists a t-private terminating protocol that computes it with ε-advantage, for some $0 < \varepsilon \leq \frac{1}{2}$.

We end this section with some standard communication complexity definitions. Let $\{0,1\}^{\leq m}$ denote the collection of binary strings whose length is at most m. The ε-advantage *communication complexity* of f, when restricted to $\{0,1\}^{\leq m}$, denoted $C_\varepsilon(f_m)$, is the minimum over all n-party protocols which compute f with ε-advantage over $\{0,1\}^{\leq m}$, of the worst case communication length of the protocol, over all n-tuples of inputs that are all at most m-bit long. (Notice that party P_i has input x_i. This is the "regular" definition, as in [17, 16], and should not be confused with the one of Chandra, Furst and Lipton [9].) The ε-advantage *communication complexity* of a function f, denoted $C_\varepsilon(f)(m)$, is defined as $C_\varepsilon(f_m)$. For *deterministic* protocols, $C_{det}(f)(m)$ is defined similarly. We say that f has *bounded* communication complexity if there is some positive ε $(0 < \varepsilon \leq 1/2)$ and an integer d, such that for all m, $C_\varepsilon(f)(m) \leq d$.

3 Functions That Can Be Privately Computed

In this section we present a sufficient condition, based on $C_\varepsilon(f)$, for $\lfloor (n-1)/2 \rfloor$-privacy of f. This enables us to derive $\lfloor (n-1)/2 \rfloor$-private error-free protocols for a wide family of "natural" functions. We start with a lemma:

Lemma 1: *Let $B \subseteq \{0,1\}^*$ and let $f : (\{0,1\}^*)^n \to B$. If for every $b \in B$ the function*

$$f_b(\vec{x}) = \begin{cases} 1 & \text{if } f(\vec{x}) = b \\ 0 & \text{otherwise} \end{cases}$$

can be computed t-privately, using an error-free protocol, then f is t-private.

Proof: The value of $f_b(\vec{x})$, for every $b \in B$, is determined by the value of $f(\vec{x})$. Thus, to compute $f(\vec{x})$, the parties can go over every $b \in B$ (say in lexicographic order) and compute $f_b(\vec{x})$ using the given t-private protocol for f_b, without revealing any additional information. The protocol terminates when for some $b \in B$ the value of $f_b(\vec{x})$ is 1. From the definition of f_b this implies that $f(\vec{x}) = b$. As every function f_b is computed by an error-free, t-private protocol, it is not hard to verify that the resulting protocol is error-free and is t-private. □

Theorem 2: *Let $f : (\{0,1\}^*)^n \to B$ and $0 < \varepsilon \leq \frac{1}{2}$. If for every $b \in B$ the communication complexity $C_\varepsilon(f_b)$ is bounded then f is $\lfloor (n-1)/2 \rfloor$-private.*

Proof: Using lemma 1 it is enough to prove that for every $b \in B$, the function f_b is $\lfloor (n-1)/2 \rfloor$-private. Fix $b \in B$. By the assumption, $C_\varepsilon(f_b)$ is bounded. Extending a theorem of Yao [17] from the two-party to the n-party case, it follows that $C_{det}(f_b)(m)$ is at most exponential in $C_\varepsilon(f_b)(m)$. In particular, if $C_\varepsilon(f_b)$ is bounded, then so is $C_{det}(f_b)$. If for all m, $C_{det}(f)(m) \leq d$, this means that for every m, there is a d-bit protocol for f on $\left(\{0,1\}^{\leq m}\right)^n$. Using König's lemma, this implies the existence of one deterministic protocol, $\mathcal{F}$, defined over $(\{0,1\}^*)^n$, which computes f and for every input exchanges at most d bits. (To prove this, consider the following tree: The nodes in level m of the tree are the protocols that computes f on $\left(\{0,1\}^{\leq m}\right)^n$, that use at most d bits of communication. A protocol in level $m+1$ is the son of a protocol in level m if they have the same communication on all inputs in $\left(\{0,1\}^{\leq m}\right)^n$. Since for every m there exists such a protocol, the tree is infinite. In addition, each protocol in level m has finitely many sons – at most the number of possibilities to map the strings of length $m+1$ into the set of d-bit communications. By König's lemma, this implies the existence of an infinite branch in the tree. This branch defines a protocol that computes f over $(\{0,1\}^*)^n$, and uses at most d bits of communication.)

Let $\{S_1, S_2, \ldots, S_k\}$ $(k \leq 2^d)$ be the set of all possible communications (in this deterministic protocol) with last message (the output) consisting of "1". We describe an $\lfloor (n-1)/2 \rfloor$-private protocol, $\mathcal{F}'$, which computes f_b. In the first step of $\mathcal{F}'$, each party P_i locally computes the set y_i as follows:

$$y_i = \{ j \mid 1 \leq j \leq k, \text{the communication } S_j \text{ is consistent with } x_i \}.$$

Let

$$f_b^*(y_1, \ldots, y_n) = \begin{cases} 1 & \text{if } \cap y_i \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

That is, $f_b^*(\vec{y}) = 1$ if and only if there is a communication S_j with "output" 1, which is consistent with the inputs of all n participants. One can verify that $f_b^*(\vec{y}) = f_b(\vec{x})$ and that f_b^*'s domain is finite (every y_i is a subset of $\{1, 2, \ldots k\}$). Thus, using the protocols of [5, 8] the value of $f_b^*(\vec{y})$ can be computed $\lfloor (n-1)/2 \rfloor$-privately, which implies that the value of $f_b(\vec{x})$ can be computed $\lfloor (n-1)/2 \rfloor$-privately. By Lemma 1 this completes the proof of the theorem. □

Corollary 3*:* *Let $f : (\{0,1\}^*)^n \to B$ and $0 < \varepsilon \leq \frac{1}{2}$. If $C_\varepsilon(f)$ is bounded then f is $\lfloor (n-1)/2 \rfloor$-private.*

Corollary 4*:* *Let $f : (\{0,1\}^*)^n \to B$. If for every $b \in B$ the set $f^{-1}(b)$ is finite then f is $\lfloor (n-1)/2 \rfloor$-private.*

These results imply that many "natural" functions can be computed $\lfloor (n-1)/2 \rfloor$-privately. This includes functions as addition over the positive integers, maximum, minimum, and multiplication over the integers. Note that for the minimum and maximum, $f^{-1}(b)$ is infinite for every b. However, these two functions satisfy the condition of theorem 2 and thus are $\lfloor (n-1)/2 \rfloor$-private.

Finally, we remark that the above protocols satisfy a stronger definition of privacy, as defined in [10].

4 Impossibility Results

In this section we apply communication complexity arguments to prove the existence of functions that are not even 1-private. Specifically, we show that addition over $\mathcal{Z}$, the ring of integers, is not 1-private. The same result holds for integer gcd.

Theorem 5*:* *Let $0 < \varepsilon \leq \frac{1}{2}$ and $n \geq 3$. Let $SUM_n : \mathcal{Z}^n \to \mathcal{Z}$ be defined as $SUM_n(\vec{x}) = \sum_{i=1}^{n} x_i$. There is no ε-advantage protocol which computes SUM_n 1-privately.*

The high level structure of the proof is as following: Assume, towards a contradiction, that there exists an ε-advantage, 1-private, protocol $\mathcal{F}$ which computes SUM_n. Privacy arguments imply that, with high probability, the number of bits communicated by $\mathcal{F}$ on all inputs $\vec{x}$ which satisfy $SUM_n(\vec{x}) = 0$ is bounded. We then show how to transform $\mathcal{F}$ into a two-argument protocol which computes the predicate "identity" using bounded communication. This contradicts a known lower bound.

Lemma 6*:* *Let $0 < \varepsilon \leq \frac{1}{2}$ and $0 < \delta \leq 1$ be two constants. Suppose there is an ε-advantage, 1-private protocol $\mathcal{F}$ for computing SUM_n. Then there exists a constant d such that for every input vector $\vec{x}$ with $SUM_n(\vec{x}) = 0$, the probability that fewer than d bits are transmitted during the execution of $\mathcal{F}$ is at least $1 - \delta$.*

Proof: We first show that for every $\vec{x}$ with $SUM_n(\vec{x}) = 0$, the distribution of messages on every link (P_i, P_j) $(i < j)$ of the network, when the input is $\vec{x}$, is the same as the distribution of messages on this link, when the input is $(0, 0, \ldots, 0)$. Then, we prove the existence of a constant d as above for the input vector $(0, 0, \ldots, 0)$. Combining these two claims we get the proof of the lemma.

Let $\vec{x}$ be an input vector satisfying $SUM_n(\vec{x}) = 0$. To simplify the notations, we restrict attention to the link (P_1, P_2). Consider the following three input vectors:

$$\begin{array}{rcl} \vec{x} & = & (x_1, \quad x_2, \quad x_3, \quad x_4, \quad \ldots, \quad x_n) \\ \vec{x'} & = & (x_1, \quad 0, \quad x_2 + x_3, \quad x_4, \quad \ldots, \quad x_n) \\ \vec{0} & = & (0, \quad 0, \quad 0, \quad 0, \quad \ldots, \quad 0) \end{array}$$

By the assumptions, SUM_n for all three vectors is 0. By 1-privacy, the distribution of messages on the link (P_1, P_2) must be the same for the two vectors $\vec{x}$ and $\vec{x'}$. Otherwise P_1 will be able to distinguish between the two. Similarly, the distribution of messages on the link (P_1, P_2) must be equal for the two vectors $\vec{x'}$ and $\vec{0}$, as otherwise P_2 will be able to distinguish between them. Thus, the distribution of messages on the link (P_1, P_2) is equal for $\vec{x}$ and $\vec{0}$.

In the second part of the proof, we use the termination condition for the input vector $\vec{0}$. It implies that for each link (P_i, P_j) there exists a constant $d_{i,j}$ such that the probability that more then $d_{i,j}$ bits are exchanged over (P_i, P_j) during the execution of the protocol on $\vec{0}$ is at most $\delta / \binom{n}{2}$.

Let $\vec{x}$ be any input vector satisfying $SUM_n(\vec{x}) = 0$. By the first part of the proof, the probability that more then $d_{i,j}$ bits are exchanged over (P_i, P_j) during the execution of the protocol on $\vec{x}$ is also bounded by $\delta / \binom{n}{2}$. Thus, the probability that more than $d = \sum_{i<j} d_{i,j}$ bits are exchanged (over all $\binom{n}{2}$ communication lines) during the execution of $\mathcal{F}$ on $\vec{x}$ is at most δ. This completes the proof of the lemma. $\square$

The next lemma is a known result in communication complexity. Let $ID : \{0,1\}^m \times \{0,1\}^m \rightarrow \{0,1\}$ be the following function:

$$ID(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$$

Lemma 7*:* ***(Yao [17])*** *For every constant $0 < \varepsilon \leq \frac{1}{2}$, $C_\varepsilon(ID)(m) = \Theta(\log m)$.*

Proof of Theorem 5: Assume, towards a contradiction, that there exists a protocol $\mathcal{F}$ which computes SUM_n 1-privately with ε-advantage. Let $\delta = \frac{\varepsilon}{2}$. Using Lemma 6, there exists a constant d such that for every input vector $\vec{x}$ satisfying $SUM_n(\vec{x}) = 0$, the probability that more than d bits are transmitted during the execution of $\mathcal{F}$ is bounded by δ. We use this property of $\mathcal{F}$ to construct a randomized protocol $\mathcal{A}$ to compute the ID function over $\{0,1\}^m \times \{0,1\}^m$, with $\varepsilon/2$ - advantage, using only a constant $(d+1)$ number of bits. This will be a contradiction to Lemma 7.

The protocol $\mathcal{A}$ works as following: On input $(a,b) \in \{0,1\}^m \times \{0,1\}^m$, the two parties (denoted by P_1^* and P_2^*) will simulate the protocol $\mathcal{F}$:

- P_1^* will simulate P_1 with $x_1 = a$ (the integer in the range $[0, 2^m - 1]$ whose binary representation is the string a).
- P_2^* will simulate $P_2, P_3, \ldots, P_n$ with $x_2 = -b$ and $x_3 = x_4 = \ldots = x_n = 0$.

The parties P_1^* and P_2^* will simulate $\mathcal{F}$ until it either terminates, or d bits are transmitted. If $\mathcal{F}$ terminates with final value 0, the parties output $ID(a,b) = 1$ (namely, they claim that $a = b$). If $\mathcal{F}$ terminates with value $\neq 0$, or does not terminate after exchanging d bits, then the parties output $ID(a,b) = 0$ (namely, they claim that $a \neq b$).

Clearly, the protocol $\mathcal{A}$ transmits at most $d+1$ bits. What we now show is that $\mathcal{A}$ computes the value of the ID function over $\{0,1\}^m \times \{0,1\}^m$ with probability $\geq \frac{1}{2} + \frac{\varepsilon}{2}$. If $a = b$, then the constructed $\vec{x}$ satisfies $SUM_n(\vec{x}) = 0$. By Lemma 6, the probability that more than d bits are exchanged on such input is at most δ. The other possible source of error is an incorrect output of $\mathcal{F}$. The probability that this happens is no greater than $\frac{1}{2} - \varepsilon$. Thus, the overall error probability of $\mathcal{A}$ on (a,b) is no greater than $\frac{1}{2} - \varepsilon + \delta$. By the choice of δ this is equal to $\frac{1}{2} - \frac{\varepsilon}{2}$. If $a \neq b$ then $\vec{x}$ satisfies $SUM_n(\vec{x}) \neq 0$. If more then d bits are transmitted, then $\mathcal{A}$ outputs 0 which is the correct value of ID in this case. As the probability of error in $\mathcal{F}$ is bounded by $\frac{1}{2} - \varepsilon$ then clearly this is so in case that the execution terminates with at most d communicated bits. Thus, on such (a,b) the protocol $\mathcal{A}$ computes the correct value of $ID(a,b)$ with probability at least $\frac{1}{2} + \varepsilon$. This completes the proof of the theorem. □

Corollary 8: *Multiplication over the rationales is not 1-private.*

Proof: Addition over the integers can be reduced to multiplication in the cyclic group $\{2^i \mid i \in \mathcal{Z}\}$, by mapping the integer i to the rational number 2^i. Thus a 1-private protocol for multiplication over the rationales would translate into a 1-private protocol for addition over the integers, contradicting Theorem 5. □

The technique relating communication complexity and privacy is applicable for various other functions. For example:

Theorem 9: *Let $0 < \varepsilon \leq \frac{1}{2}$ and $n \geq 2$. Let $\mathcal{N}$ denote the set of natural numbers $\{1, 2, \ldots\}$, and $GCD_n : \mathcal{N}^n \to \mathcal{N}$ be defined as the greatest common divisor of the n arguments. There is no ε-advantage protocol which computes GCD_n 1-privately.*

Proof: The proof of this theorem is very similar to the proof for Theorem 5. We sketch here the differences: the analogue of Lemma 6 will be proved for input vectors satisfying

$GCD_n(\vec{x}) = 1$. In the proof of the analogous lemma we use the following triples of n-tuples:

$$\begin{array}{rcl} \vec{x} & = & (\; x_1, \;\; x_2, \;\; x_3, \;\; x_4, \;\; \ldots, \;\; x_n \;) \\ \vec{x'} & = & (\; x_1, \;\; 1, \;\; x_3, \;\; x_4, \;\; \ldots, \;\; x_n \;) \\ \vec{1} & = & (\; 1, \;\; 1, \;\; 1, \;\; 1, \;\; \ldots, \;\; 1 \;) \end{array}$$

Now, let $PRIMES = \{q_1, q_2, q_3, \ldots\}$ denote the sequence of prime numbers. In the proof of the theorem we will construct a protocol for ID as follows: On input (a, b) (two natural numbers ≥ 1)

- P_1^* will simulate P_1 with $x_1 = q_a$.
- P_2^* will simulate $P_2, P_3, \ldots, P_n$ with $x_2 = x_3 = \ldots = x_n = q_b$.

Clearly, $GCD_n(\vec{x}) = 1$ if and only if $a \neq b$. Thus the original argument holds here as well. □

5 Characterization For Boolean Functions

In this section we characterize the privacy of any Boolean function f in terms of its communication complexity. Corollary 3 implies that any Boolean function f with bounded communication complexity is $\lfloor (n-1)/2 \rfloor$-private. The main theorem in this section states the reverse implication. More precisely, a Boolean function whose communication complexity is unbounded is not even 1-private. We set the stage for the theorem by introducing some definitions and claims.

Let $A \subseteq \{0,1\}^* \times \{0,1\}^*$. Define a relation $\sim_A$ on $A \times A$ by $(x_1, y_1) \sim_A (x_2, y_2)$ if $x_1 = x_2$ or $y_1 = y_2$. The equivalence relation $\equiv_A$ is defined as the transitive closure of $\sim_A$. $Class(A)$ denotes the number of equivalence classes of $\equiv_A$ (it can be infinite). For any set $S \subseteq \{0,1\}^* \times \{0,1\}^*$, define S_1 and S_2 as the projections of S on the first and second coordinates respectively. We now present two simple properties of the equivalence classes of the relation defined above:

Claim 1: *Let $A \subseteq \{0,1\}^* \times \{0,1\}^*$, and let K and L be two disjoint (non-empty) equivalence classes of $\equiv_A$. Then, $K_1 \cap L_1 = K_2 \cap L_2 = \emptyset$.*

Proof: Follows directly from the definition of the equivalence relation $\equiv_A$. □

Claim 2: *Let $A \subseteq \{0,1\}^* \times \{0,1\}^*$, and let K and L be two disjoint (non-empty) equivalence classes of $\equiv_A$. If $(x, y) \in (K_1 \times L_2) \cup (L_1 \times K_2)$ then $(x, y) \notin A$.*

Proof: Assume, without loss of generality, that $(x, y) \in (K_1 \times L_2)$. This implies that there exist y' such that $(x, y') \in K$ and x' such that $(x', y) \in L$. Suppose that $(x, y) \in A$. By the definition of $\sim_A$ these pairs satisfy $(x, y') \sim_A (x, y) \sim_A (x', y)$, contradicting the disjointness of the two $\equiv_A$ equivalence classes K and L. □

The next lemma states that if A and B cover $\{0,1\}^* \times \{0,1\}^*$ (A and B need not be disjoint), then either $Class(A) \leq 2$ or $Class(B) \leq 2$. Furthermore, if $Class(B) \geq 3$ then $Class(A) = 1$. Notice that it is possible to have $Class(A) = Class(B) = 2$, e.g. by taking $A = \{(x, y) | x \equiv y \pmod 2\}$ and $B = \{(x, y) | x \not\equiv y \pmod 2\}$.

Lemma 10: *Suppose $\{0,1\}^* \times \{0,1\}^* \subseteq A \cup B$. If $Class(B) \geq 3$ then $Class(A) = 1$.*

Proof: [2] Let C, D, E be three distinct equivalence classes of $\equiv_B$. Let $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ be arbitrary elements of C, D and E respectively (see Figure 1). By claim 1, $x_i \neq x_j$ and $y_i \neq y_j$ for every $i \neq j$ $(1 \leq i, j \leq 3)$. Consider the element (x_1, y_2). By claim 2, $(x_1, y_2) \in A$. Similarly $(x_i, y_j) \in A$ for every $i \neq j$ $(1 \leq i, j \leq 3)$. Furthermore, by claim 1, they all belong to the same equivalence class of $\equiv_A$. Denote this class by F.

Suppose, towards a contradiction, that $\equiv_A$ has another equivalence class G, and let $(x_4, y_4) \in G$. By claim 1, $x_4 \neq x_j$ and $y_4 \neq y_j$ for every $1 \leq j \leq 3$. Consider the element (x_4, y_1). By claim 2, $(x_4, y_1) \notin A$. Therefore, $(x_4, y_1) \in B$ and by claim 1 it must belong to C. By a similar argument, (x_4, y_2) must belong to D. This implies that $C_1 \cap D_1 \neq \emptyset$, contradicting claim 1. □

	y_1	y_2	y_3	y_4
x_1	C	F	F	-
x_2	F	D	F	-
x_3	F	F	E	-
x_4	?	?	-	G

Figure 1

Theorem 11: *Let $f : (\{0,1\}^*)^n \to \{0,1\}$ be 1-private. Then the communication complexity of f is bounded.*

Proof: Let $f : (\{0,1\}^*)^n \to \{0,1\}$, and $\mathcal{F}$ be a 1-private protocol which computes f with ε-advantage. We show that $\mathcal{F}$ can be modified to a communication protocol which computes f with $\frac{\varepsilon}{2}$-advantage while exchanging only a bounded number of bits between every pair of parties (where the bound does not depend on the input).

In order to carry out this transformation, it would be desirable to be in a situation similar to that occurring in SUM_n, where every $\vec{z} \in f^{-1}(0)$ has message distribution similar to some *fixed* input ($\vec{0}$ for SUM_n). While this desired situation may not occur, something weaker, but sufficient for our purposes, does: For every link (P_i, P_j), there is a value $v \in \{0,1\}$ and there are at most two *fixed* inputs $\vec{x}(i,j), \vec{w}(i,j) \in (\{0,1\}^*)^n$ so that for every $\vec{z} \in f^{-1}(v)$, the distribution of messages exchanged on the link (P_i, P_j), given input $\vec{z}$, is *identical* to the same distribution either given input $\vec{x}(i,j)$ or given input $\vec{w}(i,j)$.

Let $1 \leq i < j \leq n$. Define the set $A_{i,j}$ (resp. $B_{i,j}$) by $(x_i, x_j) \in A_{i,j}$ if there exists an n-tuple $\vec{x} = (x_1, x_2, \ldots, x_n)$ extending (x_i, x_j) such that $f(\vec{x}) = 0$ (resp. $f(\vec{x}) = 1$). Notice that $A_{i,j} \cup B_{i,j}$ covers $\{0,1\}^* \times \{0,1\}^*$. We now relate the 1-privacy of f to the sets $A_{i,j}, B_{i,j}$.

Suppose $f(\vec{x}) = f(\vec{w}) = 0$ and $(x_i, x_j) \equiv_{A_{i,j}} (w_i, w_j)$. Any 1-private protocol which computes f induces identical distribution of messages exchanged between P_i and P_j, given the input $\vec{x}$ and the input $\vec{w}$. (Similarly for $f(\vec{x}) = f(\vec{w}) = 1$ and $(x_i, x_j) \equiv_{B_{i,j}} (w_i, w_j)$.) The proof of this claim is immediate from the definitions of the sets $A_{i,j}, B_{i,j}$, the relations $\sim_{A_{i,j}}, \sim_{B_{i,j}}, \equiv_{A_{i,j}}$ and $\equiv_{B_{i,j}}$, and 1-privacy.

By lemma 10, for each (i,j), at least one of the sets $A_{i,j}, B_{i,j}$ has no more than two equivalence classes in the corresponding $\equiv$ equivalence relation. Without loss of generality $A_{i,j}$ has at most two equivalence classes, and let (x_i, x_j) and (w_i, w_j) be fixed representatives of the two classes. (Notice that for a different pair i', j', the set $B_{i',j'}$ can be the set with $Class(\cdot) \leq 2$.)

[2]This proof simplifies our original proof, which appeared in the early version of this paper. It is due to Freddy Bruckstein and Alon Orlitsky, and it appears here with their kind permission.

Let $\vec{x}(i,j)$ and $\vec{w}(i,j)$ denote two inputs in $f^{-1}(0)$ whose projections on the i and j coordinates yield the above representatives (take $\vec{x}(i,j) = \vec{w}(i,j)$ if there is only one equivalence class).

Let $\mathcal{F}$ be a 1-private protocol which computes f with ε-advantage ($0 < \varepsilon \leq \frac{1}{2}$). Consider runs of $\mathcal{F}$ on the $2 \cdot \binom{n}{2}$ inputs $\vec{x}(1,2), \vec{w}(1,2), \ldots, \vec{x}(n-1,n), \vec{w}(n-1,n)$. Let d be the minimal integer such that the probability that P_i and P_j exchange more then d bits, given that the input is either $\vec{x}(i,j)$ or $\vec{w}(i,j)$, is less than $\frac{\varepsilon}{n^2}$ (for all (i,j) pairs).

We modify the protocol $\mathcal{F}$ as follows: All parties run $\mathcal{F}$ until it either terminates, or some pair attempts to exchange more than d bits. Suppose, without loss of generality, that the first such pair is $(1,2)$, and further that $\vec{x}(1,2), \vec{w}(1,2) \in A_{1,2}$. In this case, the pair aborts the execution of $\mathcal{F}$, and announces that the value of f is 1. (In case of conflicting announcements in the same round, the one resulting from the minimal pair in lexicographical order is adopted by all parties).

The modified protocol exchanges at most $d \cdot \binom{n}{2}$ bits on all inputs. We argue that it computes the correct answer with probability at least $\frac{1}{2} + \frac{\varepsilon}{2}$. Suppose, without loss of generality, that $f(\vec{z}) = 0$. For all pairs i,j with representatives $\vec{x}(i,j), \vec{w}(i,j) \in A_{i,j}$, the messages exchanged between P_i and P_j on input $\vec{z}$ are distributed identically to one of the two representatives, and thus the probability that more than d bits will be exchanged in $\mathcal{F}$ is bounded above by $\frac{\varepsilon}{n^2}$. Thus the probability that some i,j as above will abort the execution of $\mathcal{F}$ and announce "1" as the final outcome is smaller then $\frac{\varepsilon}{2}$. Any pair i,j with $\vec{x}(i,j), \vec{w}(i,j) \in B_{i,j}$ cannot announce "1" as the final outcome by aborting $\mathcal{F}$'s execution. The other source of errors stems from errors in $\mathcal{F}$'s final outcome. The probability that this occurs is smaller than $\frac{1}{2} - \varepsilon$. Thus the overall probability of error in the modified protocol is smaller than $\frac{1}{2} - \frac{\varepsilon}{2}$. $\square$

Combining Theorem 11 and Corollary 3, we get

Corollary 12: *A Boolean function $f : (\{0,1\}^*)^n \to \{0,1\}$ is 1-private if and only if it has bounded communication complexity. If f is 1-private then it is $\lfloor (n-1)/2 \rfloor$-private.*

Together with the characterization of $\lceil n/2 \rceil$-private Boolean functions of [10], we conclude:

Theorem 13: *Any Boolean function $f : (\{0,1\}^*)^n \to \{0,1\}$ is either*

- *n-private.*
- *$\lfloor (n-1)/2 \rfloor$-private private but not $\lceil n/2 \rceil$-private.*
- *not 1-private.*

6 Non-Terminating Protocols

So far, we considered protocols that are required to be private, compute the correct value (with probability $\geq \frac{1}{2} + \varepsilon$), and terminate (with probability 1). In this section we show that if we abandon the termination condition (and the requirement that a distinguished "last message" is sent), and require instead only "correctness in the limit", then private computations which were not possible before become possible. We exemplify the model via two examples. Consider the following non-terminating protocol for computing SUM_n:

For $k = 1, 2, 3, \ldots$

- Each party P_i locally computes $y_i \in [-k, k]$ such that $y_i \equiv x_i \pmod{2k+1}$.
- The parties compute n-privately, using the protocol of [6], s_k such that $s_k \equiv \sum_{i=1}^{n} y_i \pmod{2k+1}$ and $s_k \in [-k, k]$. The value s_k is the outcome of the k-th round.

It is easy to see that no coalition (of any size $t \leq n$) gets any additional information from the execution of this protocol. After a *finite* number of iterations (which depends on the input vector) the output s_k stabilizes on the correct value. Thus, SUM_n is n-private in the non-terminating model.

The next example shows that we can demand even more. Consider the following non-terminating protocol for computing GCD_n:

For $k = 1, 2, 3, \ldots$

- Each party P_i computes $y_i = 1$ if $k \mid x_i$ and $y_i = 0$ otherwise.
- The parties compute, $\lfloor (n-1)/2 \rfloor$-privately, the AND of the y_i's. This bit indicates whether k divides all the x_i's or not. The output of the k-th round is the maximal $g \leq k$ that divides all the x_i's.

The protocol computes GCD_n, and it is $\lfloor (n-1)/2 \rfloor$-private in the non-terminating model. In the protocol for SUM_n, no participant knows when the result stabilizes. Here, however, the gcd can never exceed x_i. Thus after x_i many iterations, party P_i knows that he knows the final outcome. After $\max_{i=1}^{n} x_i$ iterations, all the parties know that they know the final outcome. However, the protocol must continue, since the parties do not know whether the other parties know that the result is already final. In other words, the privacy requirements prevent the parties from achieving *common knowledge* [14] on the fact that the result is already final, and an infinite execution is enforced.

7 Concluding Remarks

In case that a function f is not 1-private, it can still be computed by first revealing the lengths $|x_i|$ of all participants inputs, and then using the protocols for the finite case of [8, 5]. However, $f(\vec{x})$ can always be computed while revealing only smaller amount of additional information (for a formal definition see [3, 4]) – the *maximal* input length, i.e., $\max_{i=1}^{n} |x_i|$ – in two steps:

- Compute $\lfloor (n-1)/2 \rfloor$-privately the value $m = \max_{i=1}^{n} |x_i|$.
- Compute $\lfloor (n-1)/2 \rfloor$-privately the value of f over the *finite* domain of strings with length at most m.

For functions with low circuit complexity, the second step can be performed efficiently using the protocols described in [5, 8, 1]. The first step can be performed using the protocol described in the proof of theorem 2 (section 3): In the ℓ-th round, party P_i sets $y_i = 1$ if $|x_i| \leq \ell$ and $y_i = 0$ otherwise. Then, the parties compute $\lfloor (n-1)/2 \rfloor$-privately the AND of the y_i's and stop with output ℓ if the AND is 1. This step is therefore polynomial-time in the length of the input $\vec{x}$. It is possible to substantially reduce the number of iterations in this part while preserving the polynomial complexity, by checking in the ℓ-th round whether $\max_{i=1}^{n} |x_i| \leq 2^{\ell}$. Thus if the maximum is m, this fact is found in $O(\log m)$ rounds (instead of $O(m)$ rounds).

A possible direction for extending Theorem 11 to functions with non-Boolean range might be to generalize Lemma 10 to coloring of the two dimensional plane by more than 2 colors. However, such generalization of the lemma is not true: There are 3 colorings of the plane (say by colors 0,1,2), where $Class(0) = Class(1) = Class(2) = \infty$. Such example was constructed by Nati Linial, and is included, with his kind permission, in the appendix. It remains open whether the corresponding generalization of Theorem 11 is true.

One of the interesting questions in the area of private computations has been to investigate the structure of the *privacy hierarchy*. Recently, we have resolved this question [12] for n argument functions which are defined over finite domains. We have shown that for finite domains, the privacy hierarchy has exactly $\lfloor n/2 \rfloor$ levels. For any $\lceil n/2 \rceil \leq t \leq n-2$, an n–argument function which is t–private but not $t+1$–private was constructed. The major tool used in this proof was to partition the n inputs into two sets of appropriate sizes, and applying the criteria for privacy of two argument functions. This approach is not useful in determining t-privacy for t which is at most $\lfloor (n-1)/2 \rfloor$, because one of the two sizes will be larger than t. Indeed, the results in [12] leave open the structure of the privacy hierarchy for functions defined over countable domains.

The communication complexity techniques used in the present work are of very different nature than the partition techniques which we mentioned above. They enabled us to show that the privacy hierarchy contains a new, additional level (0-privacy). We believe that these techniques will also play an important role in future research towards resolving the exact structure of the privacy hierarchy over countable domains.

Acknowledgments

We are grateful to Shai Ben-David, Josh Benaloh, and Charlie Rackoff for very helpful discussions and contributions to this work. Thanks to Freddy Bruckstein and Alon Orlitsky for simplifying the proof of lemma 10, and to Nati Linial for his 3-coloring example. We would also like to thank Oded Goldreich, who has the good property of commenting on early drafts fast.

References

[1] Bar-Ilan J., and D. Beaver, "Non-Cryptographic Fault-Tolerant Computing in a Constant Number of Rounds", *Proc. of 8th ACM Symposium on Principles of Distributed Computing*, 1989, pp. 201-209.

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

[3] R. Bar-Yehuda, B. Chor, and E. Kushilevitz, "Privacy, Additional Information, and Communication", *Proc. of 5th IEEE Structure in Complexity Theory*, July 1990, pp. 55-65.

[4] R. Bar-Yehuda, B. Chor, E. Kushilevitz, and A. Orlitsky, "Privacy, Additional Information, and Communication", *IEEE Transactions on Information Theory*, In Press.

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

[6] Benaloh (Cohen), J.D., "Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret", *Advances in Cryptography - Crypto86 (proceedings)*, A.M. Odlyzko (ed.), Springer-Verlag, Lecture Notes in Computer Science, Vol. 263, pp. 251-260, 1987.

[7] Blakley, G.R., and L. Swanson, "Security Proofs for Information Protection Systems", *Proc. of IEEE Symposium on Security and Privacy*, 1981, pp. 75-88.

[8] Chaum, D., C. Crépeau, and I. Damgård, "Multiparty Unconditionally Secure Protocols" *Proc. of 20th ACM Symposium on Theory of Computing*, 1988, pp. 11-19.

[9] Chandra, A. K., M. L. Furst, and R. J. Lipton, "Multi-Party Protocols" *Proc. of 15th ACM Symposium on Theory of Computing*, 1983, pp. 94-99.

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

[11] Chor, B., and E. Kushilevitz, "Secret Sharing over Infinite Domains", *Jour. of Cryptology*, Vol. 6, No. 2, 1993, pp. 87-96.

[12] B. Chor, M. Geréb-Graus, and E. Kushilevitz, "On the Structure of the Privacy Hierarchy", *J. Cryptology*, In Press.

[13] Goldreich, O., S. Micali, and A. Wigderson, "How to Play Any Mental Game", *Proc. of 19th ACM Symposium on Theory of Computing*, 1987, pp. 218-229.

[14] Halpern, J. Y., and Y. Moses, "Knowledge and Common Knowledge in Distributed Environment", *Journal of the Association for Computing Machinery*, Vol. 37, No. 3, July 1990, 549-587.

[15] Kushilevitz, E., "Privacy and Communication Complexity", *SIAM Jour. on Disc. Math.* Vol. 5, No. 2, May 1992, pp. 273-284.

[16] Tiwari, P., "Lower Bounds on Communication Complexity in Distributed Computer Networks", *Proc. of 25th Annual Symposium on Foundations of Computer Science*, 1984, pp. 109–117.

[17] Yao, A. C., "Some Complexity Questions Related to Distributive Computing", *Proc. of 11th ACM Symposium on Theory of Computing*, 1979, pp. 209-213.

[18] Yao, A. C., "Protocols for Secure Computations", *Proc. of 23th Annual Symposium on Foundations of Computer Science*, 1982, pp. 160-164.

8 Appendix

In this appendix we describe an example, due to Nati Linial, of a function $f : \mathcal{N} \times \mathcal{N} \rightarrow \{0, 1, 2\}$ with the property that $Class(0) = Class(1) = Class(2) = \infty$. This example implies that

Lemma 10 cannot be extended to non-Boolean functions. The function f is defined as follows (see Figure 2):

$$f(x,y)=\begin{cases} x \bmod 3 & \text{if } x=y \\ x+1 \bmod 3 & \text{if } x\neq y \wedge (x\equiv y) \bmod 3 \\ 2(x+y) \bmod 3 & \text{otherwise} \end{cases}$$

	0	3	6	$\cdots$	1	4	7	$\cdots$	2	5	8	$\cdots$
0	0	1	1	$\cdots$	2	2	2	$\cdots$	1	1	1	$\cdots$
3	1	0	1	$\cdots$	2	2	2	$\cdots$	1	1	1	$\cdots$
6	1	1	0	$\cdots$	2	2	2	$\cdots$	1	1	1	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
1	2	2	2	$\cdots$	1	2	2	$\cdots$	0	0	0	$\cdots$
4	2	2	2	$\cdots$	2	1	2	$\cdots$	0	0	0	$\cdots$
7	2	2	2	$\cdots$	2	2	1	$\cdots$	0	0	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
2	1	1	1	$\cdots$	0	0	0	$\cdots$	2	0	0	$\cdots$
5	1	1	1	$\cdots$	0	0	0	$\cdots$	0	2	0	$\cdots$
8	1	1	1	$\cdots$	0	0	0	$\cdots$	0	0	2	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Figure 2

It can be verified that every diagonal point (x,x) forms a singleton equivalence class. Since for each value in $\{0,1,2\}$ there are infinitely many preimages of the form (x,x), it follows that $Class(0)=Class(1)=Class(2)=\infty$.