A Theorem on Sensitivity and Applications in Private Computation *

Anna Gál† Adi Rosén‡

Abstract

In this paper we prove a theorem that gives an (almost) tight upper bound on the sensitivity of a multiple-output Boolean function in terms of the sensitivity of its coordinates, and the size of the range of the function. We apply this theorem to get improved lower bounds on the time (number of rounds) to compute Boolean functions by private protocols. These bounds are given in terms of the sensitivity of the function being computed and the amount of randomness used by the private protocol. These lower bounds are tight (up to constant factors) for the case of the `xor` function, and together with the results in [25] establish a tight (up to constant factors) tradeoff between randomness and time in private computation.

1 Introduction

An important characteristic of a Boolean function is its sensitivity. Informally, the sensitivity of a function is the maximum number of input variables, such that changing the value of just one variable at a time, changes the value of the function as well. The sensitivity of Boolean functions and its relation to other complexity measures have been studied extensively. A number of important results have been achieved via arguments about sensitivity. For example, lower bounds on the time required for CREW PRAM computation [33, 10, 28], and lower bounds on the size of reliable circuits from unreliable gates [11, 30, 31, 14, 15] were given in terms of the sensitivity of the function being computed. A generalization of sensitivity, block sensitivity, was defined by Nisan [28]. Studying block sensitivity revealed that sensitivity provides lower bounds for several other measures, including Boolean decision tree complexity [28] and the degree of real polynomials representing Boolean functions [29]. The relation between sensitivity and block sensitivity has been studied in a number of papers [28, 17, 32, 19]. In several settings, average

*An early version of this paper appeared in the Proceedings of the 31st STOC (1999), pp. 348–357.

†Dept. of Computer Science, The University of Texas at Austin, Austin, TX 78712, USA. Part of this work was done while on leave at the Dept. of Computer Science of the University of Toronto and the Fields Institute. Partially supported by ITRC, an Ontario Centre of Excellence and NSF CAREER Award CCR-9874862. e-mail: panni@cs.utexas.edu.

‡Dept. of Computer Science, University of Toronto, Toronto, Canada. Current affiliation: Dept. of Computer Science, Technion, Haifa 32000, Israel. e-mail: adiro@cs.technion.ac.il.

sensitivity is an important measure. It has been shown that the average sensitivity of a function is related to its Fourier coefficients [20], and that the average sensitivity of functions computable by constant depth circuits must be low [26]. Bounds on the sensitivity of various classes of functions were given in [34, 35].

In this paper we prove a theorem on the sensitivity of multiple-output Boolean functions. We give an almost tight upper bound on the sensitivity of such functions, in terms of the sensitivity of each coordinate, and the size of the range of the function. More formally, we prove the following theorem:

Let $F : \{0,1\}^n \to \{0,1\}^m$ be an m-output Boolean function with coordinate functions $f_j : \{0,1\}^n \to \{0,1\}$, such that $s(f_j) \leq k$ for each $1 \leq j \leq m$, where $s(f)$ is the sensitivity of f. If the range of F contains D different values, then the sensitivity of F is at most $k \cdot 4(\log_2 D + 2)$.

Note that the restriction on the size of the range of F can be interpreted as a condition on the "correlation" between the coordinate functions f_j. Without this restriction, the number of possible values of an m-output Boolean function is 2^m. It is easy to see that $s(F) \leq km$ must always hold. Similarly, if we restrict the range of F to be a q-dimensional subcube of $\{0,1\}^m$, that is, we require that $m - q$ of the m coordinate functions are constants (and the other q coordinate functions can be arbitrary functions), then $s(F) \leq kq$ must hold. Our results show that even if the range of F is an *arbitrary subset* of size 2^q of $\{0,1\}^m$, the sensitivity of F cannot be much larger. Our bound is almost tight, as for q independent coordinates the sensitivity kq is achieved.

We use the above theorem to prove lower bounds in information-theoretic private computation. We believe, however, that the theorem is of independent interest, and may find additional applications. Using the above theorem and the machinery of [25] we prove improved lower bounds on the number of rounds required to privately compute a Boolean function. The lower bound is given in terms of the sensitivity of the function being computed, and the amount of randomness used by the protocol overall. For the case of the function `xor` (exclusive or) these lower bounds are tight, up to a small constant factor. A private protocol to compute a Boolean function f allows a number of players, each possessing a single input bit, to compute the value of the function f on their combined input, in a way that no single player learns any "unnecessary" information (in particular, the inputs of the other players).[1] Private computation in this setting was first considered by Yao [36] and has been the subject of considerable amount of research [1, 2, 4, 5, 7, 8, 9, 12, 13, 16, 21, 22, 23].

Using randomness any function can be computed privately if the number of players is at least 3. On the other hand, for most functions (except very simple ones), randomness is necessary in order to compute the function privately. Randomness as a resource has been the subject of extensive research in the past decade. In the context of private computation, the main questions addressed about randomness as a resource have been the minimum number of random bits required for private computation of different functions, and tradeoffs between the amount of randomness and the amount of time (i.e., number of rounds) required for the computation [5, 21, 25, 24, 6]. It is worthwhile to note that one can also characterize the class of functions computable by linear size circuits in terms of the amount of randomness required for their private

[1] In the literature, a more general notion of t-privacy is used, requiring that no coalition of t players learns extra information. Here we discuss the case of $t = 1$. See Section 3 for a formal definition of private protocols.

computation [24]. The question whether private computations in general can be carried out in constant number of rounds was addressed in [1, 3, 18].

A lower bound on the number of rounds required for the private computation of Boolean functions was given by Kushilevitz and Rosén [25]. They proved that it takes at least $\Omega(\log s(f)/d)$ rounds to privately compute a function f of sensitivity $s(f)$ using d random bits overall. They also gave protocols to compute the function `xor`, that use a small number of random bits, and at the same time are efficient in terms of rounds: for any $d \geq 2$, they provided a protocol to privately compute the `xor` of n bits, using $O(\log n/\log d) = O(\log s(\texttt{xor})/\log d)$ rounds and d random bits. However, the exact tradeoff between randomness and rounds was left as an open question. Using our theorem on sensitivity we prove that it takes $\Omega(\log s(f)/\log d)$ rounds to privately compute a Boolean function f using d random bits overall. In view of the above mentioned upper bounds this bound is tight (up to a small constant factor).

2 A Theorem on Sensitivity

In this section we prove a theorem that gives an upper bound on the sensitivity of a Boolean function $F : \{0,1\}^n \rightarrow \{0,1\}^m$, in terms of the sensitivity of the coordinate functions $f_j : \{0,1\}^n \rightarrow \{0,1\}$, and the size of the range of F.

Definition 1 ***(Sensitivity)***

- *For a binary vector y, denote by $y^{(i)}$ the binary vector obtained from y by flipping the i'th entry.*
- *A function f is* sensitive *to its i'th variable on input y, if $f(y) \neq f(y^{(i)})$.*
- *$s_y(f)$ is the number of variables to which the function f is sensitive on input y.*
- *The* sensitivity *of a function f is $s(f) = \max_y s_y(f)$.*
- *The* average sensitivity *of a function f is $as(f) = \frac{1}{2^n}\sum_{y\in\{0,1\}^n} s_y(f)$.*

Note that for a multiple output Boolean function $F : \{0,1\}^n \rightarrow \{0,1\}^m$ the value of F on each input $x \in \{0,1\}^n$ is a binary vector of length m, and $F(x) \neq F(y)$ if and only if $f_j(x) \neq f_j(y)$ for at least one of the coordinate functions f_j, $1 \leq j \leq m$.

Fact 1 *Let $F : \{0,1\}^n \rightarrow \{0,1\}^m$ be an m-output Boolean function with coordinate functions $f_j : \{0,1\}^n \rightarrow \{0,1\}$, such that $s(f_j) \leq k$ for each $1 \leq j \leq m$. Then $s(F) \leq km$.*

Proof: The statement follows from the simple fact that $s(F) \leq \sum_{j=1}^m s(f_j)$. □

Note that this bound is tight: taking $f_j(x) = x_j$, we have $s(f_j) = 1 = k$ and $s(F) = m = km$.

The proof of Fact 1 shows that if we restrict the range of F to a q-dimensional subcube of $\{0,1\}^m$ then $s(F) \leq kq$ must hold. In the following theorem we prove that the sensitivity of F cannot be much larger, even if the range of F is an *arbitrary subset* of size 2^q of $\{0,1\}^m$.

Theorem 1 *Let $F : \{0,1\}^n \rightarrow \{0,1\}^m$ be an m-output Boolean function with coordinate functions $f_j : \{0,1\}^n \rightarrow \{0,1\}$, such that $s(f_j) \leq k$ for each $1 \leq j \leq m$. If the range of F contains D different values, then the sensitivity of F is at most $k \cdot 4(\log_2 D + 2)$.*

We observe that this bound is almost tight. Consider again the functions $f_j(x) = x_j$, each of sensitivity $k = 1$. In this case the sensitivity of F is $k \cdot \log_2 D$, as $D = 2^n = 2^m$. Note that our bound holds for every k and D, independent of the values of n and m.

In the following the function F and the functions f_j are as defined in the above theorem. In particular, we always assume that $s(f_j) \leq k$.

For our argument it is convenient to use the following definition.

Definition 2 ***(Sensitivity restricted to a set)*** *For $F(x) = (f_1(x), \ldots, f_m(x))$ and $x \in \{0,1\}^n$, let $S(x)$ denote the set of vectors $y \in \{0,1\}^n$ that are at Hamming distance 1 from x and for which $F(y) \neq F(x)$. Let $U \subseteq \{0,1\}^n$. We use the notation $S(x,U) = S(x) \cap U$, and say that $|S(x,U)|$ is the sensitivity of F on x restricted to the set U.*

Notation Let $V \subseteq \{0,1\}^n$ be a subset of input vectors. We partition the set V into levels, with respect to a fixed vector $v_0 \in V$. The set V_i is the set of vectors in V that are at Hamming distance i from v_0, and is called the i'th level of V with respect to v_0. Note that $V_0 = \{v_0\}$. The notation used for the levels V_i does not indicate their dependence on the choice of v_0. This should nevertheless be clear from the context. For $x \in V_i$ we use the notation $\sigma_{V,v_0}(x) = |S(x, V_{i+1})|$ and $\sigma_{V,v_0}(V_i) = min_{x \in V_i} \sigma_{V,v_0}(x)$. Note that σ_{V,v_0} depends on the choice of both V and $v_0 \in V$, since the levels V_i are defined with respect to v_0, and if $x \in V_i$ then $\sigma_{V,v_0}(x)$ is the sensitivity of F on x restricted to the set V_{i+1}. We omit from our notation the dependence of the set $S(x)$ and $\sigma_{V,v_0}(x)$ on F.

2.1 The Proof

Our argument proceeds in two stages. First we choose a subset $V \subseteq \{0,1\}^n$ of inputs with certain properties that we define below. Informally these properties say that if F has sensitivity s on some input v_0, then we can start with v_0 and build a set of inputs V such that for "many" levels V_i of V (partitioning V into levels with respect to v_0) the following holds: on each input from V_i the sensitivity of F restricted to V_{i+1} is "high" (in terms of s). That is, not only the sensitivity of F is high on each selected input, but it remains high even when restricted to the next level of the set V.

In the second stage we use the set V to demonstrate a large number (in terms of s) of different values for F. This will give us the relation between the sensitivity s and the number of possible values D.

We start with a lemma stating that if the sensitivity of F on an input x is s, then the sensitivity must be "almost" s on many inputs from the set $S(x)$. Moreover, this holds even if we consider the sensitivity with respect to a partition of the inputs into levels.

Lemma 1 *Let $F : \{0,1\}^n \rightarrow \{0,1\}^m$ be an m-output Boolean function with coordinate functions $f_j : \{0,1\}^n \rightarrow \{0,1\}$, such that $s(f_j) \leq k$, for $1 \leq j \leq m$. Let $S(x,U)$ be defined with respect* *to F as above. Let $B = \{0,1\}^n$, and let $v_0 \in B$. Let B_i denote the i'th level of B with respect to v_0, that is, the set of vectors in B that are at Hamming distance i from v_0. Let $x \in B_i$ and let $S \subseteq S(x, B_{i+1})$ such that $|S| = \sigma$. Then there are at least $\sigma/2$ nodes $v \in S$, such that $|S(v, B_{i+2})| \geq \sigma - 4k$.*

Proof: We think of B as the n dimensional hypercube, with an edge connecting two nodes if and only if their Hamming distance is exactly 1. We number the nodes in S from 1 to σ, and name them v_ℓ, $1 \leq \ell \leq \sigma$. (Recall that $S \subseteq B_{i+1}$.) Let $v_{(\ell,j)}$ for $1 \leq \ell, j \leq \sigma$, $\ell \neq j$, be the node in B_{i+2} whose Hamming distance to both v_ℓ and v_j is 1 (note that by our notation $v_{(\ell,j)}$ and $v_{(j,\ell)}$ denote the same node). Let $e_{\ell,j}$ be the edge that connects v_ℓ to $v_{(\ell,j)}$.

Call an edge *sensitive* if it connects nodes x and y such that $F(x) \neq F(y)$. First we show that among the $\sigma(\sigma-1)$ edges $e_{\ell,j}$ at most $\sigma 2(k-1)$ are not sensitive edges. To this end we partition the $\sigma(\sigma-1)$ edges into σ sets. The set E_ℓ contains the edges $e_{j,\ell}$ for $j \neq \ell$, $1 \leq j \leq \sigma$, that is, the set E_ℓ contains for each node v_j, $j \neq \ell$, the edge that connects v_j to the node $v_{(j,\ell)}$.

Now we claim that in each set E_ℓ there are at most $2(k-1)$ edges which are not sensitive. Since $v_\ell \in S$, there is some coordinate t such that $f_t(v_\ell) \neq f_t(x)$. Without loss of generality assume that $f_t(x) = 0$, and $f_t(v_\ell) = 1$. For an edge $e_{j,\ell}$ which is not sensitive, we have that $F(v_j) = F(v_{(\ell,j)})$, and in particular $f_t(v_j) = f_t(v_{(\ell,j)})$. There can be at most $k-1$ such edges with $f_t(v_j) = 1$ since the sensitivity of f_t is at most k, and together with v_ℓ there are at most k nodes adjacent to x on which f_t is 1. Similarly, together with x there are at most k nodes adjacent to v_ℓ on which f_t is 0. It follows that in E_ℓ there are at most $2(k-1)$ edges which are not sensitive. Thus, among the $\sigma(\sigma-1)$ edges $e_{\ell,j}$ at most $\sigma 2(k-1)$ are not sensitive. By a simple averaging argument there are at most $\sigma/2$ nodes in S which are adjacent to at least $4(k-1)$ non-sensitive edges in $\cup_{\ell=1}^{\sigma} E_\ell$.

It follows that there are at least $\sigma/2$ nodes in S which are adjacent to more than $\sigma - 1 - 4(k-1) \geq \sigma - 4k$ sensitive edges in $\cup_{\ell=1}^{\sigma} E_\ell$. That is, there are at least $\sigma/2$ nodes in S such that the sensitivity of F restricted to B_{i+2} is at least $\sigma - 4k$ on each of them. $\square$

Now we are ready to prove the existence of a set V of input vectors with the desired properties.

Lemma 2 *If the sensitivity of F on v_0 is s, then there is a set $V \subseteq B = \{0,1\}^n$ such that $\sigma_{V,v_0}(V_i) \geq s/4$ for $i = 0, \ldots, \lceil s/(8k) \rceil - 1$.*

Proof: We construct the set V by building its levels V_i inductively. (Recall that V_i denotes the i'th level of V with respect to v_0, that is, V_i is the set of vectors from V that are at Hamming distance i from v_0.) We have $V_0 = \{v_0\}$. We also partition the set B into levels with respect to v_0.

We prove by induction on i that we can build sets $V_0, \ldots, V_i$, such that:

- for $0 \leq j < i$ and any $x \in V_j$, we have $\sigma_{V,v_0}(x) \geq (1/2)s - j2k$;
- for any $x \in V_i$, we have $\sigma_{B,v_0}(x) \geq s - i4k$.

If the sensitivity of F on v_0 is s, then by Lemma 1 there is a set $V_1 \subseteq S(v_0, B_1)$ of at least $s/2$ nodes such that for any $x \in V_1$ we have $|S(x, B_2)| = \sigma_{B,v_0}(x) \geq s - 4k$. This proves the statement for $i = 1$.

Now assume we have built the set V by levels up to level i. We now build level $i+1$. Consider a node $x \in V_i$. By the induction hypothesis $\sigma_{B,v_0}(x) \geq s - i4k$ for $x \in V_i$, that is, $|S(x, B_{i+1})| \geq s - i4k$. By Lemma 1 there is a set G_x of at least $(1/2)(s - i4k)$ nodes $v \in S(x, B_{i+1})$, such that $|S(v, B_{i+2})| \geq (s - i4k) - 4k = s - (i+1)4k$.

To build the set V_{i+1} we let

$$V_{i+1} = \cup_{x \in V_i} G_x \ .$$

It follows that each node $x \in V_i$ has at least $(1/2)s - i2k$ neighbors y in V_{i+1}, for which $F(x) \neq F(y)$, that is, for any $x \in V_i$ it holds that

$$\sigma_{V,v_0}(x) \geq (1/2)s - i2k \ .$$

Moreover, for all the nodes $y \in V_{i+1}$, it holds that $\sigma_{B,v_0}(y) \geq s - (i+1)4k$. Thus our induction step is complete. Now, as long as $i \leq s/(8k)$ we have that $\sigma_{V,v_0}(x)$, for any $x \in V_i$, is at least $(1/2)s - (s/(8k))2k = (1/4)s$. □

This completes the first stage of our proof. Next we will show, using the set V guaranteed by the above lemma, that F must take many different values. We start by a claim that selects a number of different values among the neighbors of a single input vector.

Claim 1 *Let $x \in \{0,1\}^n$ and $S \subseteq S(x)$. If $|S| \geq \xi$ then for some ℓ, such that $\xi/k \leq \ell \leq m$, we can find ℓ vectors $y_1, \ldots, y_\ell$ among the vectors in the set S, and ℓ coordinates $j_1, \ldots, j_\ell$ among the m coordinates of F, such that $f_{j_a}(y_a) \neq f_{j_a}(x)$ for $a = 1, \ldots, \ell$, and $f_{j_a}(y_b) = f_{j_a}(x)$ for $1 \leq a < b \leq \ell$.*

The above claim says, for example, that if $F(x)$ is 0 in all coordinates, then we can find ℓ vectors ,$y_1, \ldots, y_\ell$ in S, for $\xi/k \leq \ell \leq m$, and a set of ℓ coordinates, such that if we appropriately reorder the coordinates of F, then the values of F on these vectors have the form: $F(y_1) = (1.....)$, $F(y_2) = (01....)$, $F(y_3) = (001...)$, $F(y_4) = (0001..)$, and so on.

Proof of Claim 1. We can take any vector from S as y_1, and let j_1 be the first coordinate where $F(y_1)$ and $F(x)$ differ, that is $f_{j_1}(y_1) \neq f_{j_1}(x)$. Since the sensitivity of each f_j is at most k, there are at most $k-1$ other vectors in S on which the value of F differs from $F(x)$ in the same coordinate j_1. Pick any of the remaining vectors from S as y_2, and let j_2 be the first coordinate where $F(y_2)$ and $F(x)$ differ. Thus we can continue this process for at least $\ell \geq \xi/k$ steps, and find the vectors and coordinates required by the claim. Note that $\ell \leq m$, since there are only m coordinates. □

Selecting different values We now define a procedure to select input vectors from the set V guaranteed by Lemma 2, in a way that F has a different value on each selected vector.

Definition 3 *:*

- *A* signature *of a node x is a subset $T \subseteq [1...m]$ of coordinates, and a set of cardinality $|T|$ of binary values $\epsilon_j \in \{0,1\}$, such that $f_j(x) = \epsilon_j$ for $j \in T$. The signature of x serves as a witness of the value of F on x. We say that the length of the signature is $|T|$.*

- *We say that the signatures of the nodes x and y are* inconsistent *if there is a coordinate j that participates in both signatures but $f_j(x) \neq f_j(y)$.*

- *We say that a node z* preserves *the signature of the node x if for each coordinate j participating in the signature of x it holds that $f_j(z) = f_j(x)$.*

- *For $x \in V_i$, let $S_x \subseteq S(x, V_{i+1})$ be the subset of $S(x, V_{i+1})$ consisting of the nodes that preserve the signature of x.*

The following observation follows directly from the definitions above.

Observation 1 *For any two nodes x and y, if the signatures of x and y are inconsistent then $S_x \cap S_y = \emptyset$.*

The next observation requires a simple proof.

Observation 2 *If the length of the signature of a node $x \in V_i$ is ν, then $|S_x| \geq |S(x, V_{i+1})| - \nu k$.*

Proof: Since the sensitivity of each f_j is at most k, there are at most νk vectors in $S(x, V_{i+1})$ on which F takes a value that differs from $F(x)$ in at least one of the ν coordinates that belong to the signature of x. $\Box$

We now describe the procedure for selecting input vectors with different values. In this process we will assign to each selected vector a *signature* (defined above) and an *address*, which is a sequence of integers, representing the way the vector was selected, as defined in the sequel. The procedure selects from each level V_i a subset of the nodes that we denote by Z_i. The procedure starts with selecting the node v_0 to which we assign a signature of size 0 and address of size 0 (i.e., an empty signature and an empty address). We thus have $Z_0 = \{v_0\}$. For any i, the procedure selects the set Z_{i+1} after the set Z_i has been determined. To determine the set Z_{i+1} we start with the sets $S_x \subseteq S(x, V_{i+1})$ for $x \in Z_i$. (Recall that the nodes in S_x preserve the signature of x by definition.) If S_x is not empty, we apply Claim 1 to x and S_x, to get nodes to be included in Z_{i+1}. We get a set Y_x of at least $\ell \geq |S_x|/k$ vectors, $Y_x = \{y_1, \ldots, y_\ell\}$. Note that Claim 1 selects ℓ coordinates, such that for every $1 \leq a \leq \ell$ the value of $F(y_a)$ differs from the value of $F(x)$ on the a'th selected coordinate, and the value of a coordinates are fixed. On the other hand, since for each vector $y \in S_x$ the value $F(y)$ must be consistent with the signature of x, none of the ℓ coordinates chosen by Claim 1 participates in the signature of x. We thus set the *signature* of y_a by adding the additional a coordinates and their values fixed by Claim 1 to the signature of x (which is consistent with $F(y_a)$). Thus, if the length of the signature of x is ν, then the length of the signature of y_a is $\nu + a$. We obtain the *address* of y_a by appending the integer a to the address of x. The set Z_{i+1} is defined to be the union of the vectors selected for each node $x \in Z_i$, that is, $Z_{i+1} = \cup_{x \in Z_i} Y_x$. We continue this procedure as long as the last set Z_i is not empty.

Next we analyze the properties of our procedure. First note that the address of a node $x \in Z_i$ consists of exactly i integers. We also have the following simple observation.

Observation 3 *For $x \in Z_i$, the length of the signature of x is exactly $\sum_{u=1}^{i} t_u$ if the address of x is $(t_1, t_2, \ldots, t_i)$.*

The following two claims show that the values of F on all inputs in $Z = \cup_{i \geq 0} Z_i$ are all different.

Claim 2 *For any two vectors y and z from the same level Z_i ($i \geq 1$), the signatures of y and z are inconsistent.*

Proof: We prove the claim by induction on i. The statement holds for $i = 1$ since the vectors in Z_1 and their signatures were selected by applying Claim 1 to v_0. Assume now that we have proved the statement for i, that is, for any $x_1, x_2 \in Z_i$ the signatures of x_1 and x_2 are inconsistent. Then by Observation 1 we have $S_{x_1} \cap S_{x_2} = \emptyset$. Recall that every $y \in Z_{i+1}$ must belong to a set S_x for some $x \in Z_i$. Thus for every $y \in Z_{i+1}$ we have a unique $x \in Z_i$ such that $y \in S_x$. Let $y, z \in Z_{i+1}$. If $y, z \in S_x$ for $x \in Z_i$ then the signatures of y and z must be inconsistent in the part that was appended to the signature of x after applying Claim 1 to x and S_x. If $y \in S_{x_1}$ and $z \in S_{x_2}$, for $x_1 \neq x_2$, then the signatures of y and z must be inconsistent in the part that was "inherited" from x_1 and x_2, respectively. □

Corollary 1 *For any two vectors y and z from the same level Z_i ($i \geq 1$), we have $F(y) \neq F(z)$.*

We will think of the set $Z = \cup_{i \geq 0} Z_i$ as a tree rooted at v_0. For a node $z \in Z_{i+1}$, we define its *parent* to be the node $x \in Z_i$, such that $z \in S_x$. Note that there is exactly one such node since for any $x, y \in Z_i$ we have $S_x \cap S_y = \emptyset$ by Observation 1 and Claim 2. Thus Z indeed forms a rooted tree. We say that y is an *ancestor* of z if there is a path from z to y in Z such that each step along the path leads from a node to its parent.

Claim 3 *For any two vectors from different levels $y \in Z_{i_1}$ and $z \in Z_{i_2}$, $i_1 \neq i_2$, we have $F(y) \neq F(z)$.*

Proof: Assume that there are vectors $y \in Z_i$ and $z \in Z_j$ for $i < j$ such that $F(y) = F(z)$. This is only possible if y is an ancestor of z, since each vector we select must preserve the signature of its parent and by Claim 2 the signatures are all inconsistent within each level. For the case that y is an ancestor of z, note that z must preserve the signature of each of its ancestors on the path in Z from z to y, in particular the signature of y' for some $y' \in S_y \subseteq S(y, V_{i+1})$. But this is not possible if $F(y) = F(z)$, because y' and its signature were selected by applying Claim 1 to y, thus the signature of y' contains a coordinate $\ell \in [1...m]$ such that $f_\ell(y') \neq f_\ell(y)$. □

Counting the number of selected values Since no two selected input vectors are mapped to the same value by F, it remains to show that we select a "large" number of vectors. We will prove the following theorem.

Theorem 2 *If for a set $V \subseteq \{0,1\}^n$ and vector $v_0 \in V$ the sensitivity of F restricted to the levels of V satisfies $\sigma_{V,v_0}(V_i) \geq \xi = \Delta k$ for $i = 0, \ldots, i^* - 1$, then F takes at least $\sum_{i=0}^{i^*} \binom{\Delta}{i}$ different values.*

Proof: Let V be a set that satisfies the conditions set forth in the theorem. We will show that by applying the procedure described above to the set V, we select at least $\sum_{i=0}^{i^*} \binom{\Delta}{i}$ input vectors. Let $Z = \cup_{i \geq 0} Z_i$ be the set obtained by applying our procedure to the set V.

We say that the *degree* of a node $z \in Z$ is the number of vectors selected by our procedure from the set S_z (by applying Claim 1 to z and S_z), that is, the number of z's children in the tree.

Observation 4 *If for some $0 \leq i \leq i^* - 1$ and $x \in Z_i$ the length of the signature of x is ν, then the degree of x is at least $\Delta - \nu$.*

Proof: Since $\sigma_{V,v_0}(V_i) \geq \xi = \Delta k$ for $i = 0, \ldots, i^* - 1$, we have that $|S(x, V_{i+1})| \geq \xi$. Thus, $|S_x| \geq \xi - \nu k = k(\Delta - \nu)$ must hold by Observation 2. It follows from Claim 1 that the degree of x is at least $\Delta - \nu$. $\square$

As described in the above selection procedure, we number the children of a given node by the number of additional coordinates we fix when applying Claim 1 to this node. Thus the address $(t_1, t_2, \ldots, t_i)$ of a node in Z_i specifies the path to follow on the tree from v_0 to reach this node. The path contains one node from each level Z_ℓ for $0 \leq \ell \leq i$, such that the node from Z_j on the path is the t_j'th child of the node from Z_{j-1}.

The following lemma is helpful in counting the number of vertices in Z_i.

Lemma 3 *If $t_u \geq 1$ is integer for $1 \leq u \leq i \leq i^*$ and $(t_1, t_2, \ldots, t_i)$ satisfies*

$$\sum_{u=1}^{i} t_u \leq \Delta \tag{1}$$

then $(t_1, t_2, \ldots, t_i)$ is a valid address in the tree Z.

Proof: We prove by induction on i that if $(t_1, t_2, \ldots, t_i)$ is a solution of (1) such that $t_u \geq 1$ is integer for $1 \leq u \leq i \leq i^*$, then $(t_1, t_2, \ldots, t_i)$ is a valid address in the tree.

To prove the statement for $i = 1$, we need to observe that v_0 has at least Δ children, thus if $t_1 \leq \Delta$ then (t_1) is a valid address.

Assume the statement is true for $i \leq i^* - 1$, and let $(t_1, t_2, \ldots, t_i, t_{i+1})$ be a solution to $\sum_{u=1}^{i+1} t_u \leq \Delta$. It follows that

$$\sum_{h=1}^{i} t_h \leq \Delta - t_{i+1} < \Delta \ ,$$

thus by our induction hypothesis $(t_1, t_2, \ldots, t_i)$ is a valid address that leads to a node v. By Observation 3, the length of the signature of v is $\sum_{h=1}^{i} t_h$, and by Observation 4 its degree in the tree is at least $\Delta - \sum_{h=1}^{i} t_h$. If $(t_1, t_2, \ldots, t_i, t_{i+1})$ is a solution to $\sum_{u=1}^{i+1} t_u \leq \Delta$, then $1 \leq t_{i+1} \leq \Delta - \sum_{h=1}^{i} t_h$, thus v has indeed at least t_{i+1} children and $(t_1, t_2, \ldots, t_{i+1})$ is a valid address. $\square$

We now prove the following lemma.

Lemma 4 *The number of nodes selected in level $i \leq i^*$, $|Z_i|$, satisfies*

$$|Z_i| \geq \binom{\Delta}{i} \ .$$

Proof: The number of nodes $|Z_i|$ at level i is the number of different sequences of length i $(t_1, t_2, \ldots, t_i)$, where $t_u \geq 1$ is an integer for $1 \leq u \leq i$, and such that $(t_1, t_2, \ldots, t_i)$ corresponds to a valid address in the tree.

There are exactly $\binom{\Delta}{i}$ different solutions for (1) where $t_u \geq 1$ is integer for $1 \leq u \leq i$ (cf. [27]). To see this, note that we can specify a solution for (1) by choosing i distinct integers between 1 and Δ, and this gives a bijection between solutions of (1) with the required properties and sets of i distinct integers between 1 and Δ. Suppose that the i distinct integers are $1 \leq n_1 < \ldots < n_i \leq \Delta$. Then by taking $t_1 = n_1$ and $t_u = n_u - n_{u-1}$ we get a solution for (1) with the required properties. Thus by Lemma 3 we have that the number of nodes selected at level $i \leq i^*$ is $|Z_i| \geq \binom{\Delta}{i}$. □

Since we have shown that F takes different values on each input in Z, Theorem 2 follows from the above lemma. □

We can now conclude this section with the following proof.

Proof of Theorem 1. Let the sensitivity of F be s. Then there is at least one node v_0 that has sensitivity s. Using Lemma 2 we construct a set $V \subseteq \{0,1\}^n$ such that $\sigma_{V,v_0}(V_i) \geq s/4$ for any $0 \leq i \leq \lceil s/(8k) \rceil - 1$.

We apply Theorem 2 to the set V with parameters $i^* = \lceil s/(8k) \rceil$, and $\Delta = \lfloor s/(4k) \rfloor$.

We get that the number of values that F takes is at least

$$\sum_{i=0}^{\lceil s/(8k) \rceil} \binom{\lfloor s/(4k) \rfloor}{i} \geq 2^{\lfloor s/(4k) \rfloor - 1} \ .$$

Since by the hypothesis of the theorem the number of values that F takes is D, we get

$$2^{\lfloor s/(4k) \rfloor - 1} \leq D \ ,$$

and

$$s \leq k \cdot 4(\log_2 D + 2) \ .$$

□

3 Applications to Private Computation

In this section we apply the theorem on sensitivity of the previous section to give lower bounds on the number of rounds required for the private computation of Boolean functions, given a certain amount of randomness. Comparing our results with known protocols [25], shows that our lower bounds are tight (up to a small constant factor) in terms of the sensitivity of the functions and the amount of randomness. We first briefly define private protocols and the complexity measures that we are interested in.

Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be any Boolean function. A set of n players P_i $(1 \leq i \leq n)$, each possessing a single input bit x_i (known *only* to P_i), collaborate in a protocol to compute the value of $f(x)$. The protocol operates in rounds. In each round each player may toss some coins, and then sends messages to the other players (messages are sent over private channels so that other than the intended receiver no other player can listen to them). After sending its messages, each player receives the messages sent to it by the other players in the current round. In addition, each player chooses to locally output the value of the function at a certain round. We say that the protocol computes the function $f : \{0,1\}^n \rightarrow \{0,1\}$ if for every input $x \in \{0,1\}^n$, the output produced by each player is $f(x)$.

Sometimes it is more convenient to model the coin tossing done by each player, as a set of binary random tapes R_i, each R_i being provided to player P_i. The number of random coins tossed by player P_i is the number of random bits it reads from its random tape.

Notation We denote by R_i a specific random tape provided to player P_i, and by $\vec{R} = (R_1, \ldots, R_n)$ the vector of the random tapes of all the players. By $\vec{r} = (r_1, \ldots, r_n)$ we denote the random variable for these tapes and vector of tapes. Note that if we fix $\vec{R}$, we obtain a deterministic protocol. By C_i we denote a specific sequence of messages received by P_i, and c_i denotes the random variable (depending on $\vec{r}$) for the sequence of messages received by P_i.

Informally, *privacy* with respect to player P_i means that player P_i cannot learn anything (in particular, the inputs of the other players) from the messages it receives, except what is implied by its input bit, and the value of the function computed. Formally,

Definition 4 ***(Privacy)*** *A protocol $\mathcal{A}$ for computing a function f is* private *with respect to player P_i if for any two input vectors x and y, such that $f(x) = f(y)$ and $x_i = y_i$, for any sequence of messages C_i, and for any random tape R_i provided to P_i,*

$$Pr[c_i = C_i | R_i, x] = Pr[c_i = C_i | R_i, y],$$

where the probability is over the random tapes of all other players.

A protocol is said to be private if it is private with respect to all players.

Definition 5 ***(Randomness Complexity)*** *A d-random protocol is a protocol such that for any input assignment, the total number of coins tossed by all players in* any *execution is at most d.*

Let $View_i^t$ be the view of player i at round t, that is, the input bit to this player and all the messages it has seen until round t. Note that $View_i^t$, for any i and t, is a function of the input assignment x, and the random tapes of all the players. We can thus write it as $View_i^t(x, \vec{r})$. We denote by $T_i(x, \vec{r})$ the round number in which player P_i outputs its result, as function of the input to all players and of the random tapes given to all players.

Definition 6 ***(Rounds Complexity)*** *A ρ-round protocol is a protocol such that for all i, x, $\vec{R}$, we have $T_i(x, \vec{R}) \leq \rho$.*

We will also make use of the following definition.

Definition 7 ***(Expected Rounds Complexity)***[2] *An expected ρ-round protocol is a protocol such that there exists a player P_i such that for all x, $E_{\vec{r}}[T_i(x, \vec{r})] \leq \rho$.*

[2]We adopt this weak definition, rather than requiring the property to hold for *all* players, as we are interested here in lower bounds.

We are interested in tradeoffs between the rounds complexity (and expected rounds complexity) and the randomness complexity of private protocols.

Our proof follows the line of proof of [25]. We get (almost) tight lower bounds by using our result of the previous section on the sensitivity of multiple-output Boolean functions. For completeness we give below the full proof. We will prove the following theorem.

Theorem 3 *Let $\mathcal{A}$ be a ρ-round d-random ($d \geq 2$) private protocol to compute a Boolean function f. Then, $\rho \geq \Omega(\frac{\log s(f)}{\log d})$, where $s(f)$ is the sensitivity of f.*

To prove the above theorem we use the following lemma from [25].

Lemma 5 ([25] Lemma 4.11) *Consider a private d-random protocol to compute a Boolean function f. Fix random tapes $\vec{R} = (R_1, \ldots, R_n)$. Then, for any P_i, $View_i^t(x, \vec{R})$ can assume at most 2^{d+2} different values (over the values of x).*

Using the above lemma we can prove our next lemma, following the proof in [25], but using our tight bound on the sensitivity of multiple-output Boolean functions.

Lemma 6 *Consider a private d-random protocol to compute a Boolean function f, and consider a specific vector of random tapes $\vec{R}$, and the deterministic protocol derived by it. Then for every player P_i, the function $View_i^t(x, \vec{R})$ (as a function of x only) has sensitivity at most $Q(t) \triangleq (4(d+4))^{t-1}$.*

Proof: First note that since we fix the random tapes, the views of the players are functions of the input assignment x only. (We regard each bit of the view as a Boolean function of x.) We prove the lemma by induction on t. For $t = 1$ the view of any player depends only on its single input bit. Thus, the claim is obvious. For $t > 1$ assume the claim holds for any $\ell < t$, and for any player. For $t > 1$ the view of any P_i is composed of the input bit of P_i, and bits that were received as messages, each one in some round $\ell < t$. Each such message bit is a function of the view of the player that sent it, at the round in which it was sent. That is, each such message bit is a function of a view with sensitivity at most $Q(t-1)$, and hence the sensitivity of each such message bit (regarded as a Boolean function of x) is at most $Q(t-1)$. Clearly the input bit of P_i has sensitivity 1 which is at most $Q(t-1)$. Thus, the view under consideration is composed of coordinates each having sensitivity at most $Q(t-1)$. Moreover, by Lemma 5 the view can assume at most 2^{d+2} values. It follows from Theorem 1 that the sensitivity of the view under consideration is at most $Q(t-1) \cdot 4(d+4) = Q(t)$. $\Box$

We can now give the proof of the main theorem of this section.

Proof of Theorem 3. Consider the deterministic protocol obtained from a d-random private protocol ($d \geq 2$) by fixing the vector of random tapes to be a given vector $\vec{R}$.

We prove that for any player P_i there is at least one input assignment x such that

$$T_i(x, \vec{R}) \geq \log s(f)/(2 + \log(d+4)) + 1 \ .$$

This proves our theorem, since for every player it shows a run of the protocol where the player outputs its value only after $\log s(f)/(2+\log(d+4))+1$ rounds.

Let y be an input assignment on which the sensitivity $s(f)$ is obtained. That is, y has $s(f)$ neighbors (in the inputs hypercube) where the value of f is different from $f(y)$. Denote this set of neighbors by $S(y)$. Denote by t the value of $T_i(y, \vec{R})$, i.e., the time step at which P_i outputs the value of f when the input to all players is y.

Now consider the view of P_i at round t, denoted $View^t(x, \vec{R})$, and the sensitivity of this view. Assume towards a contradiction that the sensitivity of $View^t(x, \vec{R})$ is less than $s(f)$. Then in particular the sensitivity of this view on y is less than $s(f)$. It follows that for at least one input assignment $z \in S(y)$, $View^t(y, \vec{R}) = View^t(z, \vec{R})$, and P_i would output the same value for f on inputs y and z, contradicting the fact that it is a correct protocol.

Thus the sensitivity of $View^t(x, \vec{R})$ on input y is at least $s(f)$, and t is such that $s(View_i^t(x, \vec{R})) \geq s(f)$. By Lemma 6, we get $(4(d+4))^{t-1} \geq s(f)$, i.e., $t \geq \frac{\log s(f)}{2+\log(d+4)} + 1$. It follows that $T_i(y, \vec{R}) \geq \frac{\log s(f)}{2+\log(d+4)} + 1$. □

Using similar techniques, that also follow the proofs from [25], we can obtain the following improved bound on the expected number of rounds of private protocols.

Theorem 4 *Let $\mathcal{A}$ be an expected ρ-round, d-random ($d \geq 2$) private protocol to compute a Boolean function f. Then, $\rho \geq \Omega(\frac{as(f)}{n} \cdot \frac{\log as(f)}{\log d})$.*

Proof: To prove the theorem we consider a protocol $\mathcal{A}$ and fix any player P_i. We say that the protocol is *late* on input x and vector of random tapes $\vec{R}$ if $T_i(x, \vec{R}) \geq \frac{\log as(f)}{4+2\log(d+4)} + 1$. We define a $0-1$ random variable $L(x, \vec{r})$ to be 1 if and only if the protocol is late on x and $\vec{r}$. For the purpose of this proof we also define a uniform distribution on the 2^n input assignments (this is not to say that the input are actually drawn according to such distribution).

We first show that for any deterministic protocol to compute f, derived, by fixing $\vec{R}$, from a private protocol, not only there is at least one input on which the protocol is late, but that this happens for a large fraction of the inputs.

Lemma 7 *Consider a player P_i and any fixed vector of random tapes $\vec{R} = (R_1, \ldots, R_n)$. Then*

$$E_X[L(x, \vec{R})] \geq \frac{as(f) - \sqrt{as(f)}}{2n} .$$

Proof: Consider the views of P_i, $View_i^t$, given the vector of random tapes $\vec{R}$. For any round t such that $t < \frac{\log as(f)}{4+2\log(d+4)} + 1$, by Lemma 6, we get that $s(View_i^t) < (4(d+4))^{\frac{\log as(f)}{4+2\log(d+4)}} = \sqrt{as(f)}$. Any function g computed from such a view can have at most the same sensitivity, and thus clearly an average sensitivity of at most $\sqrt{as(f)}$. Such function g can have the same values as f on at most $2^n(1 - \frac{as(f)-\sqrt{as(f)}}{2n})$ input assignments. It follows that at least $2^n\frac{as(f)-\sqrt{as(f)}}{2n}$ input assignments are late. □

The lower bound on the expected number of rounds now follows. Since at least $2^n\frac{as(f)-\sqrt{as(f)}}{2n}$ input assignments are late for any set of random tapes, $E_{\vec{r},x}[L(x, \vec{r})] \geq \frac{as(f)-\sqrt{as(f)}}{2n}$. Hence, there is at least one input assignment x for which $E_{\vec{r}}[L(x,\vec{r})] \geq \frac{as(f)-\sqrt{as(f)}}{2n}$. For such x we get

$$E_{\vec{r}}[T_i(x,\vec{r})] \geq \left(\frac{as(f)-\sqrt{as(f)}}{2n}\right) \cdot \left(\frac{\log as(f)}{4+2\log(d+4)}+1\right),$$

as needed. $\square$

4 Conclusions

In this paper we prove an almost tight upper bound on the sensitivity of multiple-output Boolean functions, in terms of the sensitivity of each output coordinate, and the size of the range of the function. Using this bound, we establish improved lower bounds on the number of rounds of private protocols, in terms of the sensitivity of the function that they compute, and the amount of randomness that they use. These lower bounds are tight (up to a small constant factor) for the function `xor`.

We believe that the theorem on the sensitivity is of independent interest, and it would be interesting to see if it can find additional applications. Also, it would be interesting to close the remaining (small constant factor) gap in our bound on sensitivity. In fact, we conjecture that the right bound is $k\lceil \log_2 D \rceil$, which can be achieved for $D = 2^q$ by a construction of q independent coordinate functions.

Acknowledgments We thank Eyal Kushilevitz for many useful discussions on privacy and randomness, and on the theorem on sensitivity presented here. We also thank Dimitri Achlioptas for useful discussions.

References

[1] J. Bar-Ilan, and D. Beaver, "Non-Cryptographic Fault-Tolerant Computing in a Constant Number of Rounds", Proc. of 8th PODC, pp. 201–209, 1989.

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

[3] D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway, "Security with Low Communication Overhead", *Advances in Cryptology*, Proceedings of Crypto '90, Springer-Verlag, Lecture Notes in Computer Science, Vol. 537, pages 62–76, 1991.

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

[5] C. Blundo, A. De-Santis, G. Persiano, and U. Vaccaro, "On the Number of Random Bits in Totally Private Computations", Proc. of 22nd ICALP, Springer-Verlag, Lecture Notes in Computer Science, Vol. 944, pp. 171–182, 1995.

[6] R. Canetti, E. Kushilevitz, R. Ostrovsky, and A. Rosén, "Randomness vs. Fault-Tolerance", Proc. of 16th PODC, pp. 35–44, 1997.

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

[8] B. Chor, and E. Kushilevitz, "A Zero-One Law for Boolean Privacy", *SIAM J. Disc. Math.* Vol. 4, 36–47, 1991.

[9] B. Chor, M. Geréb-Graus, and E. Kushilevitz, "Private Computations Over the Integers", *SIAM Jour. on Computing*, Vol. 24, No. 2, pp. 376-386, 1995.

[10] S. A. Cook, C.Dwork, and R. Reischuk, "Upper and lower time bounds for parallel random access machines without simultaneous writes", *SIAM J. Computing* Vol. 15 (1986), pp. 87-97.

[11] R. L. Dobrushin and S. I. Ortyukov, "Lower bound for the redundancy of self-correcting arrangements of unreliable functional elements," *Prob. Inf. Trans.* 13, 1977, pp. 59-65.

[12] U. Feige, J. Kilian, and M. Naor, "A Minimal Model for Secure Computation", Proc. of 26th STOC, pp. 554-563, 1994.

[13] M. Franklin, and M. Yung, "Communication Complexity of Secure Computation", Proc. of 24th STOC, pp. 699–710, 1992.

[14] P. Gács, A. Gál: "Lower bounds for the complexity of reliable Boolean circuits with noisy gates", IEEE Transactions on Information Theory, Vol. 40, No. 2, 1994, pp. 579-583.

[15] A. Gál, "Lower Bounds for the Complexity of Reliable Boolean Circuits with Noisy Gates," Proc. of 32nd FOCS, 1991, pp. 594-601.

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

[17] C. Gotsman and N. Linial, "The equivalence of two problems on the cube", *Journal of Combinatorial Theory*, Series A, Vol. 61, No. 1, 1992, pp. 142-146.

[18] Y. Ishai and E. Kushilevitz, " Private Simultaneous Messages Protocols with Applications", ISTCS97.

[19] Claire Kenyon, "Sensitivity vs. block sensitivity of Boolean functions", DIMACS Technical Report 90-18, 1990.

[20] J. Kahn, G. Kalai, and N. Linial, "The influence of variables of Boolean functions", Proc. of 29th FOCS, 1988, pp. 68-80.

[21] E. Kushilevitz, and Y. Mansour, "Randomness in Private Computations", Proc. of 15th PODC, pp. 181–190, 1996.

[22] E. Kushilevitz, S. Micali, and R. Ostrovsky, "Reducibility and Completeness in Multi-Party Private Computations", Proc. of 35th FOCS, pp. 478-489, 1994.

[23] E. Kushilevitz, "Privacy and Communication Complexity", FOCS 89, and SIAM Jour. on Disc. Math., Vol. 5, No. 2, pp. 273–284, 1992.

[24] E. Kushilevitz, R. Ostrovsky, and A. Rosén, "Characterizing Linear Size Circuits in Terms of Privacy", Proc. of 28th STOC, 1996, pp. 541-550.

[25] E. Kushilevitz, and A. Rosén, "A Randomness-Rounds Tradeoff in Private Computation", Crypto94, and SIAM J. on Discrete Mathematics, 20 (1), 1998, pp. 61–80.

[26] N. Linial. Y. Mansour, and N. Nisan, "Constant depth circuits, Fourier transform, and learnability", *Journal of the ACM*, Vol. 40, No. 3, 1993, pp. 607-620.

[27] J. H. van Lint, and R. M. Wilson, "A course in combinatorics", Cambridge University Press, 1992.

[28] N. Nisan, "CREW PRAM's and decision trees", *SIAM J. on Computing*, 20 (6), 1991, pp. 999-1007.

[29] N. Nisan and M. Szegedy, "On the degree of Boolean functions as real polynomials", Proc. of 24th STOC, 1992, pp. 462-467.

[30] N. Pippenger, G. D. Stamoulis and J. N. Tsitsiklis, "On a lower bound for the redundancy of reliable networks with noisy gates," *IEEE Trans. Inform. Theory*, vol. 37, no. 3, 1991, pp. 639-643.

[31] R. Reischuk, B. Schmeltz, "Reliable Computation with Noisy Circuits and Decision Trees – A General n log n Lower Bound," Proc. of 32nd FOCS, 1991, pp. 602-611.

[32] D. Rubinstein, "Sensitivity vs. block sensitivity of Boolean functions", *Combinatorica*, 15 (2), 1995, pp. 297-299.

[33] H. U. Simon, "A tight $\Omega(\log\log n)$ bound on the time for parallel RAM's to compute nondegenerate Boolean functions" Proc. of FCT'83, Springer Lecture Notes in Computer Science 158 (1983), pp. 439-444.

[34] G. Turán, "The critical complexity of graph properties" *Information Processing Letters*, 18, 1984, pp. 151-153.

[35] I. Wegener, "The critical complexity of all (monotone) Boolean functions and monotone graph properties", *Information and control*, 67, 1985, pp. 212-222.

[36] A. C. Yao, "Protocols for secure computations", Proc. of 23rd FOCS, 1982, pp. 160-164.