A RANDOMNESS-ROUNDS TRADEOFF IN PRIVATE COMPUTATION*

EYAL KUSHILEVITZ† AND ADI ROSÉN‡

Abstract. We study the role of randomness in multi-party private computations. In particular, we give several results that prove the existence of a randomness-rounds tradeoff in multi-party private computation of `xor`. We show that with a single random bit, $\Theta(n)$ rounds are necessary and sufficient to privately compute `xor` of n input bits. With $d \geq 2$ random bits, $\Omega(\log n/d)$ rounds are necessary, and $O(\log n/\log d)$ are sufficient.

More generally, we show that the private computation of a boolean function f, using $d \geq 2$ random bits, requires $\Omega(\log S(f)/d)$ rounds, where $S(f)$ is the sensitivity of f. Using a single random bit, $\Omega(S(f))$ rounds are necessary.

Keywords: Private distributed computations, Randomness, Lower-bounds, Sensitivity.

1. Introduction. A *1-private* (or simply, *private*) protocol $\mathcal{A}$ for computing a function f is a protocol that allows n players, P_i, $1 \leq i \leq n$, each possessing an individual secret input, x_i, to compute the value of $f(\vec{x})$ in a way that no *single* player learns more about the initial inputs of other players than what is revealed by the value of $f(\vec{x})$ and its own input[1]. The players are assumed to be honest but curious. Namely, they all follow the prescribed protocol $\mathcal{A}$ but they could try to get additional information by considering the messages they receive during the execution of the protocol. Private computations in this setting were the subject of a considerable amount of work, e.g., [BGW88, CCD88, BB89, CK89, K89, B89, FY92, CK92, CGK90, CGK92, KMO94]. One crucial ingredient in private protocols is the use of *randomness*. Quantifying the amount of randomness needed for computing functions privately is the subject of the present work.

Randomness as a resource was extensively studied in the last decade. Methods for saving random bits range over pseudo-random generators [BM84, Y82, N90], techniques for re-cycling random bits [IZ89, CW89], sources of weak randomness [CG88, VV85, Z91], and constructions of different kinds of small probability spaces [NN90, AGHP90, S92, KM93, KM94, KK94] (which sometimes even allow to eliminate the use of randomness). A different direction of research is a quantitative study of the role of randomness in specific contexts, e.g., [RS89, KPU88, BGG90, CG90, BGS94, BSV94]. In this work, we initiate a quantitative study of randomness in private computations. We mainly concentrate on the specific task of computing the `xor` of n input bits. However, most of

* An early version of this paper appeared in *Advances in Cryptology*, Proceedings of Crypto '94, Y. Desmedt, ed., Springer-Verlag, Lecture Notes in Computer Science, Vol. 839, pp. 397—410, 1994.

† Dept. of Computer Science, Technion, Haifa, Israel. e-mail: eyalk@cs.technion.ac.il; http://www.cs.technion.ac.il/~eyalk; Work on this research was supported by the E. and J. Bishop Research Fund, and by the Fund for the Promotion of Research at the Technion. Part of this research was performed while the author was at Aiken Computation Laboratory, Harvard University, supported by research contracts ONR-N0001491-J-1981 and NSF-CCR-90-07677.

‡ Dept. of Computer Science, Tel-Aviv University, Tel-Aviv, Israel. e-mail: adiro@math.tau.ac.il

[1] In the literature a more general definition of t-privacy is given. The above definition is the case $t = 1$.

our results extend to any boolean function. The task of computing `xor` was the subject of previous research due to its being a basic linear operation and its relative simplicity [FY92, CK92].

It is known as a "folklore theorem" (and is not difficult to show) that private computation of `xor` cannot be carried out deterministically (for $n \geq 3$). On the other hand, with a single random bit such a computation becomes possible: At the first round player P_n chooses a random bit r and sends to P_1 the bit $x_n \oplus r$. Then, in round i $(2 \leq i \leq n)$ player P_{i-1} xors its bit x_{i-1} with the message it received in the previous round, and sends the result to P_i. Finally, P_n xors the message it received with the random bit r. Both the correctness and privacy of this protocol are easy to verify. The main drawback of this protocol is that it takes n rounds. Another protocol for this task computes `xor` in two rounds but requires a linear number of random bits: In the first round each player P_i chooses a random bit r_i. Then, player P_i sends $x_i \oplus r_i$ to P_1 and r_i to P_2. In the second round P_2 xors all the (random) bits it received in the first round and sends the result to P_1 which xors all the messages it received during the protocol to get the value of the function. Again, both the correctness and privacy of this protocol are not hard to verify.

In this work we prove that there is a tradeoff between the amount of randomness and the number of rounds in private computations of `xor`. For example, we show that while with a single random bit $\Theta(n)$ rounds are necessary and sufficient[2], with two random bits $O(\log n)$ rounds suffice.[3] Namely, with a single additional random bit, the number of rounds is significantly reduced. Additional bits give a much more "modest" saving. More precisely, we prove that with $d \geq 2$ random bits $O(\log n / \log d)$ rounds suffice and $\Omega(\log n/d)$ rounds are required. Our upper bound is achieved using a new method that enables us to use linear combinations of random bits again and again (while preserving the privacy). The lower bounds are proved using combinatorial arguments, and they are strong in the sense that they also apply to protocols that are allowed to make errors, and that they actually show a lower bound on the *expected* number of rounds. We also show that if protocols are restricted to certain natural types (that include, in particular, the protocol that achieves the upper bound) we can even improve the lower bound and show that $\Theta(\log n / \log d)$ rounds are necessary and sufficient.

Our lower bound techniques apply not only to the `xor` function, but in fact give lower bounds on the number of rounds for any boolean function in terms of the sensitivity of the function. Namely, we prove that with $d \geq 2$ random bits $\Omega(\log S(f)/d)$ rounds are necessary to privately compute a boolean function f, whose sensitivity is

[2] More precisely, $\lceil n/2 \rceil$ rounds. This upper bound is achieved by a slight modification of the first protocol above. Assume, for simplicity, that n is even. At the first round, player P_n sends $x_n \oplus r$ to player P_1, and at the same time sends r to player P_{n-1}. The players then continue as in the above protocol, forwarding messages in parallel until the two messages meet. More precisely, in round i, $2 \leq i \leq \frac{n}{2}$, player P_{i-1} xors the message it received with its own input and sends it to player P_i and player $P_{n-(i-1)}$ xors the message it received with its own input and sends it to player P_{n-i}. In round $\frac{n}{2}$, player $P_{\frac{n}{2}}$ receives two messages and can compute the value of the function by xoring the two messages together with its own input.

[3] All logarithms are base 2, unless otherwise indicated.

$S(f)$. With a single random bit ($d = 1$) $\Omega(S(f))$ rounds are necessary.

The question whether private computations in general can be carried out in constant number of rounds was previously addressed [BB89, BFKR90]. In light of our results, a promising approach to investigate this question may be by proving that if a constant number of rounds is sufficient then a large number of random bits is required.

Subsequent to our work, several other works were done regarding the amount of randomness in privacy. In particular, the amount of randomness required for computing the function `xor` t-privately, for $t \geq 2$, was studied in [BDPV95, KM96]; in [KOR96] it is shown that the boolean functions that can be computed privately with a constant number of random bits are exactly the functions that have linear size boolean circuits. Further results on randomness in private computations appear in [CKOR97].

The rest of the paper is organized as follows: In Section 2 we give some definitions. In Section 3 we give an upper bound on the number of rounds required to privately compute `xor`. In section 4 we give lower bounds on the number of rounds to privately compute a boolean function, in terms of its sensitivity. We conclude in Section 5 with lower bounds on the expected number of rounds in terms of the average sensitivity of the function being computed. The appendix contains the improved lower bounds for restricted types of protocols.

2. Preliminaries. We give here a description of the protocols we consider, and define the *privacy* property of protocols. More rigorous definitions of the protocols are given in Section 4.1.

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 it), collaborate in a protocol to compute the value of $f(\vec{x})$. 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). It then receives the messages sent to it by the other players. In addition, each player at a certain round chooses to output the value of the function. We assume that each player knows its serial number and the total number of players n. We may also assume that each player P_i is provided with a read-only random tape R_i from which it reads random bits (rather than toss coins).

Each player P_i receives during the execution of the protocol a sequence of messages. Let c_i be a random variable of the communication string seen by player P_i, and let C_i be a particular communication string seen 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 C_i, except what is implied by its input bit, and the value of the function computed. Formally,

DEFINITION 2.1. **(Privacy)** *A protocol $\mathcal{A}$ for computing a function f is* private *with respect to player P_i if for any two input vectors $\vec{x}$ and $\vec{y}$, such that $f(\vec{x}) = f(\vec{y})$ and $x_i = y_i$, for any sequence of messages C_i, and for any random tape R_i provided to P_i,*

$$Pr[c_i = C_i | R_i, \vec{x}] = Pr[c_i = C_i | R_i, \vec{y}],$$

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

3. Upper Bound. This section presents a protocol which allows n players to use $d \geq 2$ random bits for computing `xor` privately. This protocol takes $O(\log_d n)$ rounds. (For the case $d = 1$ a similar protocol that uses $\lceil n/2 \rceil$ rounds was already described in the introduction.) All arithmetic operations in this section are done modulo 2.

Consider the following protocol (which we call the *basic protocol*): First organize the n players in a tree. The degree of the root of the tree is $d+1$, and the degree of any other internal node is d (assume for simplicity that n is such that this forms a complete tree). The computation starts from the leaves and goes towards the root by sending messages (each of them of a single bit) as follows: Each leaf player P_i sends its input bit x_i to its parent in the tree. Each internal node, after receiving messages from its d children, sums them up (modulo 2) together with its input bit x_i and sends the result to its parent. Finally, the root player sums up the $d+1$ messages it receives together with its input bit and the result is the output of the protocol.

While a simple induction shows the correctness of this protocol, and it clearly runs in $O(\log_d n)$ rounds, it is obvious that it does not maintain the required privacy. The second idea will be to "mask" each of the messages sent in the basic protocol by an appropriate random bit (constructed using the d random bits available), in a way that these masks will disappear at the end, and we will be left with the (un-masked) output. To do so we assign the nodes of the above tree vectors in $GF[2^d]$ as follows (the meaning of those vectors will become clear soon): Assign to the root the vector $(0, \ldots, 0)$. The children of the root will be assigned $d+1$ (non-zero) vectors such that the vectors in any d-size subset of them are linearly independent and the sum of all the $d+1$ vectors is $(0, \ldots, 0)$ (for example, the d unit vectors together with the $(1, \ldots, 1)$ vector satisfy these requirements). Finally, in a recursive way, given an internal node which is assigned a vector v, we assign to its d children d linearly independent vectors whose sum is v (note that in particular none of these vectors is the $\vec{0}$ vector)[4].

We now show how to use the vectors we assigned to the nodes, so as to get a private protocol. We will assume that the random bits $b_1, \ldots, b_d$ are chosen by some external processor. We will later see that this assumption can be eliminated easily. Let v be the vector assigned to some player which is a *leaf* in the tree. We will give this player a single bit $r_v = v \cdot b$, where $b = (b_1, \ldots, b_d)$ is the vector consisting of the d random bits, and the product is an inner product (modulo 2). The players will use the basic protocol, described above, with the modification that a player in a leaf also xors its message with the bit r_v it received (the other players behave exactly as before). We claim that for every player P_i, if in the basic protocol it sends the message m when the input vector is $\vec{x}$, then in the modified protocol it sends the message $m + (v_i \cdot b)$, where v_i is the vector assigned to this player. The proof goes by induction:

[4] For example, such a collection of d vectors can be constructed as follows: Since $v \neq \vec{0}$ there exists an index i such that $v_i = 1$. The first $d-1$ vectors will be the $d-1$ unit vectors $e_1, \ldots, e_{i-1}, e_{i+1}, \ldots, e_d$. The last vector will be $v - \sum_{j \neq i} e_j$. Obviously the sum of these d vectors is v and they are linearly independent.

It is trivially true for the leaf players. For internal nodes the message is calculated by adding the input of the players to the sum of the incoming messages. Using, the induction hypothesis this sum is $\sum_{k=1}^{d}[m^k + (v^k \cdot b)]$, where m^k is the message received from the k'th child in the basic protocol, and v^k is the vector assigned to the k'th child. Since the construction is such that v_i, the vector assigned to P_i, satisfies $v_i = \sum_{k=1}^{d} v^k$, then a simple algebraic manipulation proves the induction step. In particular, since the root is assigned the vector $(0, \ldots, 0)$, its output equals the output of the basic protocol. Hence, the correctness follows.

We now prove the privacy property of the protocol. The leaf players do not receive any message, hence there is nothing to prove. Let P_j be an internal node in the tree. Denote by $s^1, \ldots, s^d$ the messages it receives. We claim that for every vector $w = (w_1, \ldots, w_d) \in GF[2^d]$, and for any input vector, we have

$$Pr[(s^1 = w_1) \wedge \ldots \wedge (s^d = w_d)] \;=\; \frac{1}{2^d},$$

where the probability is over the random choice of $b_1, \ldots, b_d$ (note that in this protocol the players do not make internal random choices). In other words, fix any specific input vector $\vec{x}$, then for every vector w, there exists exactly one choice of values for $b_1, \ldots, b_d$, such that the messages that P_j receives, when the protocol is executed with input $\vec{x}$, are the vector w. Denote by $\vec{v}^1, \ldots, \vec{v}^d$ the vectors corresponding to the d children of P_j in the tree, and let $m^1, \ldots, m^d$ be the messages they have to send in the basic protocol given the input vector $\vec{x}$. As claimed, for every $1 \leq k \leq d$, the message that the k-th child sends in the modified protocol can be expressed as $s^k = m^k + (\vec{v}^k \cdot \vec{r})$. With this notation, for having $s^1 = w_1, \ldots, s^d = w_d$ the following linear system has to be satisfied:

$$\begin{bmatrix} \vec{v}^1 \\ \vdots \\ \vec{v}^d \end{bmatrix} \cdot \vec{r} = \begin{bmatrix} w_1 - m^1 \\ \vdots \\ w_d - m^d \end{bmatrix} .$$

Since $\vec{v}^1, \ldots, \vec{v}^d$ are linearly independent, this system has exactly one solution, as needed.

As for the root player the same argument can be applied to any fixed d-size subset of the $d+1$ messages it receives. This gives us that given any input vector $\vec{x}$, for all d-size messages vectors $\vec{w}$,

$$Pr[(s^1 = w_1) \wedge \ldots \wedge (s^d = w_d)] \;=\; \frac{1}{2^d} .$$

Now take two input vectors $\vec{x}$ and $\vec{y}$ such that $x_{\mathrm{root}} = y_{\mathrm{root}}$ and such that $f(\vec{x}) = f(\vec{y})$. Then by the correctness of the protocol, given a specific d-size messages-vector, the $d+1$'st message is the same for $\vec{x}$ and $\vec{y}$. Thus the privacy property holds with respect to the root too.

Finally, note that we assumed that the random choices were made by some external processor. However, we can let one of the leaf players randomly choose the bits $b_1, \ldots, b_d$ and supply each of the leaf players with the appropriate bit r_v. As the leaf players only send messages in the protocol, the special processor that selects the random bits gets no advantage.

Note that if a player is non-honest it can easily prevent the other players from computing the correct output. However, it cannot get any additional information in the above protocol, since the only message each player gets after sending its own message is the value of the function. We have thus proved the following theorem:

THEOREM 3.1. *The function* `xor` *on n input bits can be computed privately using $d \geq 2$ random bits in $O(\log n/\log d)$ rounds.*

4. Lower Bounds. In this section we prove several lower bounds on the number of rounds required to privately compute a boolean function, given that the total number of random bits the players can toss is d. The lower bound is given in terms of the sensitivity of the function. In Section 4.1 we give some formal definitions. In Section 4.2 we introduce the notion of *sensitivity* and present a lemma, central to our proofs, about sensitivity of functions. The proof of the lower bound appears in Section 4.3.

4.1. Preliminaries. We first give a formal definition for the protocols. A protocol operates in rounds. In each round each player P_i, based on the value of its input bit x_i, the values of the messages it received in previous rounds, and the values of the coins it tossed in previous rounds, tosses a certain number of additional coins, and sends messages to the other players. The values of these messages may depend on all of the above, including the coins just tossed by P_i. The player may also choose to output the value of the function as calculated by itself (this is done only once by each player). Then, each player P_i receives the messages sent to it by the other players. To define the protocol more formally, we give the following definition:

DEFINITION 4.1. **(View)**

- *A time-t* partial *view of player P_i consists of its input bit x_i, the messages it has received in the first $t-1$ rounds, and the coins it tossed in the first $t-1$ rounds. We denote it by $PView_i^t$.*
- *A time-t view of player P_i consists of its input bit x_i, the messages it has received in the first $t-1$ rounds, and the coins it tossed in the first t rounds. We denote it by $View_i^t$.*

Intuitively, the *partial view* of a player P_i in round t determines how many coins (if at all) P_i will toss in round t. Then, its *view* (which includes those newly tossed coins) determines the messages P_i will send in round t, and whether and which value it will output as the value of f. The formal definition of a protocol is given below:

DEFINITION 4.2. *A protocol consists of a set of functions $R_i^k : PView_i^k \to \mathcal{N}$ which determine how many coins are tossed by P_i in round k, and a set of functions $M_{i\to j}^k : View_i^k \to M$, $1 \leq i, j \leq n$ (where M is a finite domain of possible message values), which determine the message sent by P_i to P_j at round k.*

To quantify the amount of randomness used by a protocol we give the following definition:

DEFINITION 4.3. *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.

Note that the definition allows that in different executions different players will toss the coins. This may depend on both the input of the players, and previous coin tosses. Next, we define the correctness of a protocol. We usually consider protocols that are always correct; protocols that are allowed to err will be considered in Section 5.1.

DEFINITION 4.4. *A protocol to compute a function f is a protocol such that for any input vector $\vec{x}$ and every i, player P_i always correctly outputs the value of $f(\vec{x})$.*

It is sometime convenient to assume that each player P_i is provided with a random tape R_i, from which it reads random bits (rather than to assume that the player tosses random coins). The number of random coins tossed by player P_i is thus the rightmost position of this tape that it reads. We thus denote by R_i a specific random tape provided to player P_i, and by $\vec{R} = (R_1, \ldots, R_n)$ the vector or the random tapes of all the players ($\vec{r} = (r_1, \ldots, r_n)$ will denote the random variable for these tapes and vector of tapes). Note that if we fix $\vec{R}$, we obtain a deterministic protocol. Furthermore, $View_i^t$, for any i and t, is a function of the input assignment $\vec{x}$, and the random tapes of the players. We can thus write it as $View_i^t(\vec{x}, \vec{r})$. We denote by $T_i(\vec{x}, \vec{R}))$ the round number in which player P_i outputs its result given input assignment $\vec{x}$ and random tapes for the players $\vec{R}$.

DEFINITION 4.5. **(Rounds Complexity)** *An r-round protocol to compute a function f is a protocol to compute f such that for all i, $\vec{x}$, $\vec{R}$, we have $T_i(\vec{x}, \vec{R})) \leq r$.*

For the purpose of our proofs, we slightly modify our view of the protocol in the following way. Fix an arbitrary binary encoding for the messages in M. We will consider a protocol where each player sends instead of a single message from M, a set of boolean messages that represent the binary encoding of the message to be sent in the original protocol. These messages are sent "in parallel" in the same round. Henceforth when we refer to messages we refer to these binary messages. Clearly, the number of rounds remains the same.

4.2. Sensitivity of Functions. In this section we include some definitions related to functions $f : \{0,1\}^n \to D$, where D is some finite domain. Then, we present some useful properties related to these definitions.

DEFINITION 4.6. **(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 assignment Y, if $f(Y) \neq f(Y^{(i)})$.*
- *$\mathcal{S}_f(Y)$ is the set of variables to which the function f is sensitive on assignment Y.*
- *The* sensitivity *of a function f, denoted $S(f)$, is $S(f) \triangleq \max_Y |\mathcal{S}_f(Y)|$.*
- *The* average sensitivity *of a function f, denoted $AS(f)$, is the average of $|\mathcal{S}_f(Y)|$. That is, $AS(f) \triangleq \frac{1}{2^n} \sum_{Y \in \{0,1\}^n} |\mathcal{S}_f(Y)|$.*
- *The set of variables on which f* depends, *denoted $\mathcal{D}(f)$, is $\mathcal{D}(f) \triangleq \{i : \exists Y s.t.\ i \in \mathcal{S}_f(Y)\}$. If $i \in \mathcal{D}(f)$ we say that f depends on its i-th variable.*

The following claim gives a lower bound on the degree of error if we evaluate a function f by means of another function g, in terms of their average sensitivities. We use this property in our proofs.

CLAIM 4.7. *Consider any two functions $f, g : \{0,1\}^n \to D$. Then $f(\vec{x}) = g(\vec{x})$ for at most $2^n \cdot (1 - \frac{AS(f)-AS(g)}{2n})$ input assignments $\vec{x}$.*

Proof. Consider the n-dimensional hypercube. An *f-good edge* is an edge $e = (\vec{x}, \vec{y})$ such that $f(\vec{x}) \neq f(\vec{y})$. By the definitions, the number of f-good edges is exactly $\frac{2^n AS(f)}{2}$. Therefore, there are at least $2^n \frac{AS(f)-AS(g)}{2}$ edges which are f-good but not g-good. For each such edge $e = (\vec{x}, \vec{y})$ either $f(\vec{x}) \neq g(\vec{x})$ or $f(\vec{y}) \neq g(\vec{y})$. Since the degree of each vertex in the hypercube is n there must be at least $2^n \cdot \frac{AS(f)-AS(g)}{2n}$ inputs on which f and g do not agree. □

Next, we prove a lemma that bounds the growth of the sensitivity of a combination of functions. This lemma plays a central role in the proofs of our lower bounds, and any improvement on it will immediately improve our lower bounds.

LEMMA 4.8. *Let $\mathcal{F} = \{f_j\}, 1 \leq j \leq m$ be a set of m functions $f_j : \{0,1\}^n \to \{0,1\}$, for some n. Assume $S(f_j) \leq C$ for all j. Define the function $F(Y) \triangleq (f_1(Y), \ldots, f_m(Y))$. If F assumes at most 2^d different values (different vectors), then the sensitivity of F is at most $C \cdot (2^d - 1)$.* [5]

Proof. Let Y be the assignment on which F has the largest sensitivity, i.e. $|\mathcal{S}_f(Y)| \geq |\mathcal{S}_f(Y')|$ for any assignment Y'. Without loss of generality assume that $F(Y) = (0, \ldots, 0)$. Consider the set of neighbors of Y on which F has a value different than $(0, \ldots, 0)$ (the cardinality of this set is the sensitivity of F). There are at most $2^d - 1$ values of F attained on the assignments in this set. Consider one such value $q \in \{0,1\}^m$. There is at least one index j such that $q_j = 1$, and since the sensitivity of f_j is at most C, there can be at most C assignments $Y^{(i)}$ with the value q. We get that the total number of assignments $Y^{(i)}$ for which F has a value other than $(0, \ldots, 0)$ is at most $C \cdot (2^d - 1)$. □

4.3. Lower Bound on the Number of Rounds. In this subsection we prove the following theorem.

THEOREM 4.9. *Let $\mathcal{A}$ be an r-round d-random ($d \geq 2$) private protocol to compute a boolean function f. Then, $r = \Omega(\log S(f)/d)$.*

The first step of our proof uses the d-randomness property of the protocol to show that the number of views a player can see on a fixed input $\vec{x}$ is at most 2^d (over the different random tapes of all the players). Note that this is not obvious; although only d coins are tossed during every execution, the identity of the players that toss these coins may depend on the outcome of previous coin tosses.

LEMMA 4.10. *Consider a private d-random protocol to compute a boolean function f. Fix an input $\vec{x}$. Let $C_i^k(\vec{x}, \vec{r})$ be the communication string seen by player P_i up to round k on input $\vec{x}$ and vector of random tapes $\vec{r}$. Then, for every player P_i, $C_i^k(\vec{x}, \vec{r})$ can assume at most 2^d different values (over the different vectors of random tapes $\vec{r}$).*

[5] An obvious bound is $S(F) \leq C \cdot m$. However, for reasons that will become clear soon we are interested in bounds which are independent of m.

Proof. For each execution we can order the coin tosses (i.e., readings from the local random tapes) according to the rounds of the protocol and within each round according to the index of the players that toss them. The identity of the player to toss the first coin is fixed by $\vec{x}$. The identity of the player to toss any next coin is determined by $\vec{x}$, and the outcome of the previous coins. Therefore, the different executions on input $\vec{x}$ can be described using the following binary tree: In each node of the tree we have a name of a player P_j that tosses a coin. The two outgoing edges from this node, labeled 0 and 1 according to the outcome of the coin, lead to two nodes labeled P_k and P_ℓ respectively (k, ℓ, and j need not be distinct) which is the identity of the player to toss the next coin. If no additional coin toss occurs, the node is labeled "nil"; there are no outgoing edges from a nil node. By the d-randomness property of the protocol, the depth of the above tree is at most d, hence it has at most 2^d root-to-leaf paths. Every possible run of the protocol is described by one root-to-leaf path. Such a path determines all the messages sent in the protocol, which player tosses coins and when, and the outcome of these coins. In particular each path determines for any P_i the value of $C_i^k(\vec{x}, \vec{r})$ (for any k). Hence, $C_i^k(\vec{x}, \vec{r})$ can assume at most 2^d different values. □

In the following proof we restrict our attention to a specific deterministic protocols derived from the original protocol by fixing specific vector of random tapes $\vec{R} = (R_1, \ldots, R_n)$ for the n players. In such a deterministic protocol the views of the players are functions of only the input assignment $\vec{x}$.

LEMMA 4.11. *Consider a private d-random protocol to compute a boolean function f. Fix random tapes $\vec{R} = (R_1, \ldots, R_n)$. Recall that $View_i^k(\vec{x}, \vec{r})$ is the view of player P_i at round k on input $\vec{x}$ and vector of random tapes $\vec{r}$. Then, for any P_i, $View_i^k(\vec{y}, \vec{R})$ can assume at most 2^{d+2} different values (over the values of $\vec{y}$).*

Proof. Partition the input assignments $\vec{x}$ into 4 groups according to the value of x_i (0 or 1), and the value of $f(\vec{x})$ (0 or 1). We argue that the number of different values the view can assume within each such group is at most 2^d. Fix an input $\vec{x}$ in one of these 4 groups and consider any other input $\vec{y}$ pertaining to the same group. Recall that $C_i^k(\vec{y}, \vec{R})$ is the communication string seen by player P_i until round k on input $\vec{y}$ and when the random tapes of the players are $\vec{R}$. If the value of $C_i^k(\vec{y}, \vec{R})$ is some communication string C_i, then by the privacy requirement[6], communication C_i must also occur by round k when the input is $\vec{x}$, and the vector of random tapes is some $\vec{R}' = (R'_1, \ldots, R'_n)$, where $R'_i = R_i$. Thus, the value of $C_i^k(\vec{y}, \vec{R})$ must also appear as $C_i^k(\vec{x}, \vec{r})$ for some vector of random tapes. However, by Lemma 4.10, $C_i^k(\vec{x}, \vec{r})$ can assume at most 2^d values (over the values of $\vec{r}$). Thus, $C_i^k(\vec{y}, \vec{R})$ can assume at most 2^d values over the possible input assignments that pertain to the same group.

Now, observe that $View_i^k(\vec{y}, \vec{r})$ is determined by the input bit y_i, the communication string $C_i^k(\vec{y}, \vec{r})$, and the random tape r_i. Therefore, on $\vec{R}$ and on two input assignments $\vec{y}$ and $\vec{y}'$ of the same group (in particular $y_i = y'_i$), if $View_i^k(\vec{y}, \vec{R}) \neq View_i^k(\vec{y}', \vec{R})$ then $C_i^k(\vec{y}, \vec{R}) \neq C_i^k(\vec{y}', \vec{R})$. Thus, $View_i^k(\vec{y}, \vec{R})$ can assume at most 2^d different values over the input assignments that pertain to the same group. □

[6] The privacy requirement is defined on the final communication string, but this clearly implies the same requirement on any prefix of it.

The following lemma gives an upper bound on the sensitivity of the view of a player at a given round, in terms of the number of random bits and the round number. This will enable us to give a lower bound on the number of necessary rounds.

LEMMA 4.12. *Consider a private d-random protocol to compute a boolean function f, and consider specific vector of random tapes $\vec{R}$, and the deterministic protocol derived by it. Then for every player P_i, the function $View_i^k(\vec{x}, \vec{R})$ (as a function of $\vec{x}$ only) has sensitivity of at most $Q(k) \triangleq (2^{d+2})^{k-1}$.*

Proof. First note that since we fix the random tapes, the views of the players are functions of the input assignment $\vec{x}$ only. We prove the lemma by induction. For $k = 1$ the view of any player depends only on its single input bit. Thus, the claim is obvious. For $k > 1$ assume the claim holds for any $\ell < k$. This implies in particular that all messages received by player P_i and included in the view under consideration have sensitivity of at most $Q(k-1)$. Clearly the input bit itself has sensitivity 1 which is at most $Q(k-1)$. Thus the view under consideration is composed of bits each having sensitivity at most $Q(k-1)$. Moreover, by Lemma 4.11 the view can assume at most 2^{d+2} values. It follows from Lemma 4.8 that the sensitivity of the view under consideration is at most $Q(k-1) \cdot (2^{d+2} - 1) \leq Q(k)$. (Note that Lemma 4.8 allows us to give a bound which does not depend on the *number* of messages received by P_i.) $\square$

We can now give the lower bound on the number of rounds, in terms of the sensitivity of the function and the number of random bits.

THEOREM 4.13. *Given a private d-random protocol ($d \geq 2$) to compute a boolean function f, consider the deterministic protocol derived from it by any given random tapes $\vec{R}$. For any player P_i, there is at least one input assignment $\vec{x}$ such that $T_i(\vec{x}, \vec{R}) = \Omega(\log S(f)/d)$.*

Proof. Consider a fixed but arbitrary player P_i. Denote by t the largest round number in which P_i outputs a value, i.e., $t = \max_{\vec{x}}\{T_i(\vec{x}, \vec{R})\}$. We claim that as long as the sensitivity of the view of P_i does not reach $S(f)$, there is at least one input assignment for which P_i cannot output the correct value of f. Let Y be an input assignment on which the sensitivity $S(f)$ is obtained. That is, the value of $F(Y)$ is different than the value of F on $S(f)$ of Y's "neighbors". Hence, if the sensitivity of the view of P_i is less than $S(f)$, then the output of P_i must be wrong on either Y or on at least one of these "neighbors" (as the sensitivity of the view is an upper bound on the sensitivity of the output). Thus, t is such that $S(View_i^t(\vec{x}, \vec{R})) \geq S(f)$. By Lemma 4.12, we get $2^{(d+2)(t-1)} \geq S(f)$, i.e., $t \geq \frac{\log S(f)}{(d+2)} + 1$. $\square$

This proves Theorem 4.9; moreover, it shows not only that there is an input assignment $\vec{x}$ and random tapes $\vec{R}$ for which the protocol runs "for a long time", but also that for each vector of random tapes $\vec{R}$ there is such input assignment. The following corollary follows for the function `xor` (using the fact that $S(\texttt{xor}) = n$).

COROLLARY 4.14. *Let $\mathcal{A}$ be an r-round d-random private protocol ($d \geq 2$) to compute* `xor` *of n bits. Then $r = \Omega(\log n/d)$.*

4.3.1. Lower Bound for a Single Random Bit ($d = 1$). For the case of a single random bit ($d = 1$), we have the following lower bound:

THEOREM 4.15. *Let $\mathcal{A}$ be an r-round 1-random private protocol to compute a boolean function f. Then, $r = \Omega(S(f))$.*

To prove the theorem, we restrict our attention to one of the two deterministic protocols derived from the original protocol by a fixing the value of the random bit [7]. The messages and views in this protocol are functions of the input vector, $\vec{x}$, only. Let Y be an assignment on which $S(f)$, the sensitivity of f, is obtained. For a given function m, a variables x_j is called *good* for m on Y if both m and f are sensitive to x_j on Y. We denote by $\mathcal{G}_m(Y)$ the set of good variables on Y, i.e., $\mathcal{G}_m(Y) \triangleq \mathcal{S}_m(Y) \cap \mathcal{S}_f(Y)$. We first prove the following two lemmas:

LEMMA 4.16. *Consider any player P_i. Denote by m_1 a message that P_i receives such that $|\mathcal{G}_{m_1}(Y) \setminus \{x_i\}| \geq 1$. Then for any other message m_2 received by P_i such that $|\mathcal{G}_{m_2}(Y) \setminus \{x_i\}| \geq 1$, either (a) $\mathcal{G}_{m_1}(Y) \setminus \{x_i\} = \mathcal{G}_{m_2}(Y) \setminus \{x_i\}$ or (b) $|\mathcal{G}_{m_1}(Y) \cup \mathcal{G}_{m_2}(Y)| \geq S(f) - 1$.*

Proof. Assume towards a contradiction, that both (a) and (b) do not hold. First, since $|\mathcal{G}_{m_1}(Y) \setminus \{x_i\}| \geq 1$ and $|\mathcal{G}_{m_2}(Y) \setminus \{x_i\}| \geq 1$ there are two variables $x_k \in \mathcal{G}_{m_1}(Y)$, $x_\ell \in \mathcal{G}_{m_2}(Y)$ such that $k \neq i$ and $\ell \neq i$. Moreover, by the assumption that (a) does not hold, we can assume without loss of generality (as to the names of m_1 and m_2) that $x_\ell \notin \mathcal{G}_{m_1}(Y)$ (in particular, $\ell \neq k$). By the assumption that (b) does not hold, there is a variable x_j ($j \neq i$) such that f is sensitive to x_j on Y, but both m_1 and m_2 are not sensitive to x_j on Y. Now consider the following three input assignments:

- $Y_0 = Y^{(j)}$.
- $Y_1 = Y^{(k)}$.
- $Y_2 = Y^{(\ell)}$.

Consider $View_i$ on the above 3 inputs and assume, without loss of generality, that $m_1(Y) = m_2(Y) = 0$. Since both m_1 and m_2 are *not* sensitive to x_j on Y, then $m_1(Y^{(j)}) = 0$ and $m_2(Y^{(j)}) = 0$. Since m_1 is sensitive to x_k on Y, then $m_1(Y^{(k)}) = 1$. Since m_2 is sensitive to x_ℓ on Y, but m_1 is not, then $m_1(Y^{(\ell)}) = 0$ and $m_2(Y^{(\ell)}) = 1$. Hence, $View_i$ assumes 3 different values for Y_0, Y_1, and Y_2. The function f is sensitive on Y to all of j, k and ℓ, therefore, $f(Y_0) = f(Y_1) = f(Y_2)$, and x_i is equal in all three assignments. However, in the proof of Lemma 4.11, it is shown that the number of values of $View_i$ corresponding to inputs with the same value of f and the same value of x_i is at most $2^d = 2$ — a contradiction. □

The following lemma gives an upper bound on the sensitivity of the view of the player in terms of the round number. We then use this lemma to give a lower bound on the number of necessary rounds.

LEMMA 4.17. *Let $t \leq (S(f)-1)/2$ be a round number and P_i be any player. Then, $|\mathcal{G}_{View_i^t}(Y)| \leq t$.*

Proof. We prove the claim by induction on t. For $t = 1$ clearly $|\mathcal{G}_{View_i^t}(Y)| \leq 1$ (since before getting any messages, the view depends only on x_i). For $1 < t \leq (S(f) - 1)/2$ assume the claim holds for any $k < t$. Denote by M the set of messages

[7] We let the identity of the player that tosses this coin to possibly depend on the input $\vec{x}$. However note that if we want that the privacy and 1-randomness properties hold, this cannot be the case.

received by P_i and included in the view under consideration. Clearly $\mathcal{G}_{View_i^t}(Y) \subseteq \{x_i\} \cup (\cup_{m \in M} \mathcal{G}_m(Y))$. There could be one of three cases:

1. For any message $m \in M$, $|\mathcal{G}_m(Y) \setminus \{x_i\}| = 0$. In this case the claim clearly holds.
2. Any two messages $m_1, m_2 \in M$, such that $|\mathcal{G}_{m_1}(Y) \setminus \{x_i\}| \geq 1$, and $|\mathcal{G}_{m_2}(Y) \setminus \{x_i\}| \geq 1$, satisfy $\mathcal{G}_{m_1}(Y) \setminus \{x_i\} = \mathcal{G}_{m_2}(Y) \setminus \{x_i\}$. It follows that $|\mathcal{G}_{View_i^t}(Y)| \leq |\{x_i\} \cup (\cup_{m \in M} \mathcal{G}_m(Y))| \leq |\mathcal{G}_{m_1}(Y)| + 1$. Since by the induction hypothesis $|\mathcal{G}_{m_1}(Y)| < t$, then $|\mathcal{G}_{View_i^t}(Y)| \leq t$.
3. There are two messages $m_1, m_2 \in M$, such that $|\mathcal{G}_{m_1}(Y) \setminus \{x_i\}| \geq 1$, and $|\mathcal{G}_{m_2}(Y) \setminus \{x_i\}| \geq 1$, but $\mathcal{G}_{m_1}(Y) \setminus \{x_i\} \neq \mathcal{G}_{m_2}(Y) \setminus \{x_i\}$. By Lemma 4.16, $|\mathcal{G}_{m_1}(Y) \cup \mathcal{G}_{m_2}(Y)| \geq S(f) - 1$, and (without loss of generality) $|\mathcal{G}_{m_1}(Y)| \geq (S(f) - 1)/2$. This contradicts the induction hypothesis as m_1 was received in some round $k < t \leq (S(f) - 1)/2$, and therefore generated by a view of round k. By the induction hypothesis $|\mathcal{G}_{m_1}(Y)| \leq k < (S(f) - 1)/2$.

□

We can now give the proof of Theorem 4.15.

Proof of Theorem 4.15.. Consider any player P_i. Denote by t the largest round number in which P_i outputs a value, i.e. $t = \max_{\vec{x}}\{T_i(\vec{x}, 0)\}$. As in the proof of Theorem 4.13, it must be that $|\mathcal{G}_{View_i^t}(Y)| \geq S(f)$, and therefore, by Lemma 4.17, we have $t > (S(f) - 1)/2$. □

For the function `xor` we have the following corollary.

COROLLARY 4.18. *Let $\mathcal{A}$ be an r-round 1-random private protocol to compute* `xor` *of n bits. Then $r = \Omega(n)$.*

5. Lower Bounds on the Expected Number of Rounds. As the protocols we consider are randomized, it is possible that for the same input $\vec{x}$, different random tapes for the players will result in executions that run for different number of rounds. Hence, it is natural to consider not only the *worst case* running time but also the *expected* running time. Usually, saying that a protocol has expected running time r means that for every input $\vec{x}$ the expected time until *all* players finish the execution is bounded by r (where the expectation is over the choices of the random tapes of the players). Here we consider a weaker definition, which requires only the *existence* of a player i whose expected running time is bounded by r. As we are proving a lower bound, this only makes our result stronger: It would mean that for *every* player there is an input assignment for which the expected running time is high. Note that it is not necessarily the case that the first player that computes the value of the function can announce this value (and thus all players compute the value within one round). The reason is that the fact that a certain player computes the function at a certain round may reveal some information on the inputs, and hence such announcing may violate the privacy requirement (see [CGK90]). We first define the expected rounds complexity of a protocol.

DEFINITION 5.1. **(Expected Rounds Complexity)** *An* expected *r-round protocol to compute a function f is a protocol to compute f such that there exists a player P_i such that for all $\vec{x}$, $E_{\vec{r}}[T_i(\vec{x}, \vec{r})] \leq r$.*

The lower bound that we prove in this section is in terms of the average sensitivity of the computed function. In particular, we prove an $\Omega(\log n/d)$ lower bound on the expected number of rounds required by protocols that privately compute `xor` of n bits. We will prove the following theorem:

THEOREM 5.2. *Let f be a boolean function and let $\mathcal{A}$ be an* expected *r-round d-random private protocol ($d \geq 2$) to compute the function f. Then, $r = \Omega(AS(f) \cdot \log AS(f)/nd)$.*

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 $\vec{x}$ and vector of random tapes $\vec{R}$ if $T_i(\vec{x}, \vec{R}) \geq \frac{\log AS(f)}{2(d+2)} + 1$. We define a $0-1$ random variable $L(\vec{x}, \vec{r})$ to be 1 if and only if the protocol is late on $\vec{x}$ and $\vec{r}$. For the purpose of our proofs in this section we define a uniform distribution on the 2^n input assignments (this is not to say that the input are actually drawn by such distribution). Moreover, note that the domain of vectors of random tapes is enumerable.

We first show that for any deterministic protocol derived from a private protocol to compute f, 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 5.3. *Consider a player P_i and any fixed vector of random tapes $\vec{R} = (R_1, \ldots, R_n)$. Then*

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

Proof. Consider the views of P_i, $View_i^t$, given the vector of random tapes $\vec{R}$. For any round t such that $t < \frac{\log AS(f)}{2(d+2)} + 1$, by Lemma 4.12, we get $S(View_i^t) < 2^{(d+2)\frac{\log AS(f)}{2(d+2)}} = \sqrt{AS(f)}$. Any function g computed from such a view can have at most the same sensitivity, and thus clearly an average sensitivity of at most $\sqrt{AS(f)}$. By Claim 4.7, such a function g can have the correct value for the function f for at most $2^n(1 - \frac{AS(f)-\sqrt{AS(f)}}{2n})$ input assignments. Since we assume that $\mathcal{A}$ is correct for *all* input assignments, it follows that at least $2^n \frac{AS(f)-\sqrt{AS(f)}}{2n}$ input assignments are late. □

We can now give a lower bound on the expected number of rounds.

LEMMA 5.4. *Consider a player P_i. There is at least one input assignment $\vec{x}$ for which*

$$E_{\vec{r}}[T_i(\vec{x}, \vec{r})] \geq \left(\frac{AS(f) - \sqrt{AS(f)}}{2n}\right) \left(\frac{\log AS(f)}{2(d+2)} + 1\right) = \Omega(AS(f) \cdot \log AS(f)/nd) .$$

Proof. By Lemma 5.3, $E_{\vec{r},\vec{x}}[L(\vec{x},\vec{r})] \geq \frac{AS(f)-\sqrt{AS(f)}}{2n}$. Hence, there is at least one input assignment $\vec{x}$ for which $E_{\vec{r}}[L(\vec{x},\vec{r})] \geq \frac{AS(f)-\sqrt{AS(f)}}{2n}$. For such $\vec{x}$ we get

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

as needed. □

Theorem 5.2 follows from the above lemma. The following corollary applies to the function `xor`.

COROLLARY 5.5. *Let $\mathcal{A}$ be an* expected *r-round d-random private protocol ($d \geq 2$) to compute* `xor` *of n bits. Then, $r = \Omega(\log n/d)$.*

Proof. Follows from Theorem 5.2 and the fact that $AS(\texttt{xor}) = n$. □

5.1. Weakly Correct Protocols. In this section we consider protocols that are allowed to make a certain amount of errors. Given a protocol $\mathcal{A}$, denote by $\mathcal{A}_i(\vec{x},\vec{r})$ the output of the protocol in player P_i, given input assignment $\vec{x}$ and vector of random tape $\vec{r} = (r_1,\ldots,r_n)$.

DEFINITION 5.6. *For $\delta < 1/2$, a $(1-\delta)$-correct protocol to compute a function f is a protocol that for every player P_i and every input vector $\vec{x}$ satisfies $Pr_{\vec{r}}[\mathcal{A}_i(\vec{x},\vec{r}) = f(\vec{x})] \geq (1-\delta)$.* Note that while designing a protocol one usually wants a stronger requirement; that is, with high probability *all* players compute the correct value. With the above definition, it is possible that in every execution of the protocol at least one of the players is wrong. However, as our aim now is to prove a lower bound this weak definition only makes our result stronger.

In the following theorem we give lower bounds on the number of rounds and on the expected number of rounds for weakly correct protocols.

THEOREM 5.7. *Let f be a boolean function.*

- *Let $\mathcal{A}$ be a $(1-\delta)$-correct r-round d-random private protocol ($d \geq 2$) to compute f. If $\delta < \frac{AS(f)-\sqrt{AS(f)}}{2n}$ then $r = \Omega(\log AS(f)/d)$.*
- *Let $\mathcal{A}$ be a $(1-\delta)$-correct* expected *r-round d-random private protocol ($d \geq 2$) to compute f. Then $r = \Omega((1-\sqrt{2\delta}) \cdot (\frac{AS(f)-\sqrt{AS(f)}}{2n} - \sqrt{\frac{\delta}{2}}) \cdot \frac{\log AS(f)}{d})$.*

Proof. We first prove the lower bound on the number of rounds, and then turn our attention to the expected number of rounds. The correctness requirement implies that for any player P_i, $Pr_{\vec{r}}[\mathcal{A}_i(\vec{x},\vec{r}) = f(\vec{x})] \geq 1-\delta$, for all $\vec{x}$. This implies that there exists a vector of random tapes $\vec{R}$ such that for at least $2^n(1-\delta)$ input assignments $\vec{x}$, $\mathcal{A}_i(\vec{x},\vec{R}) = f(\vec{x})$. As in the proof of Lemma 5.3 (using Claim 4.7), it follows that before round number $\frac{\log AS(f)}{2(d+2)} + 1$, the protocol can be correct on at most $2^n(1 - \frac{AS(f)-\sqrt{AS(f)}}{2n})$ inputs (with random tapes $\vec{R}$). Since we require that at least $2^n(1-\delta)$ are correct, we have that at least

$$2^n(1-\delta) - 2^n\left(1 - \frac{AS(f)-\sqrt{AS(f)}}{2n}\right) = 2^n\left(\frac{AS(f)-\sqrt{AS(f)}}{2n} - \delta\right)$$

inputs are late. To get a lower bound on r for an r-round protocol, it is sufficient to have a single input vector $\vec{x}$ such that the execution on $(\vec{x}, \vec{R})$ is "long". For this, note that if $\delta < \frac{AS(f)-\sqrt{AS(f)}}{2n}$ then (for random tapes $\vec{R}$) the number of late inputs is greater than 0. This gives us a lower bound of $r = \Omega(\log AS(f)/d)$ for any $(1-\delta)$-correct r-round d-random protocol, with δ as above.

We now turn to the lower bound on the *expected* number of rounds of $(1-\delta)$-correct protocols. Consider a player P_i. Define a $0-1$ random variable $G(\vec{x}, \vec{r})$ to be 1 if and only if $\mathcal{A}_i(\vec{x}, \vec{r}) = f(\vec{x})$. Then, the correctness requirement implies that $E_{\vec{r}}[G(\vec{x}, \vec{r})] \geq 1-\delta$, for all $\vec{x}$. It follows that for any $\vec{R}$ the probability that $\vec{R}$ satisfies $E_{\vec{x}}[G(\vec{x}, \vec{R})] \geq 1 - \sqrt{\frac{\delta}{2}}$ is at least $1-\sqrt{2\delta}$.[8] For any such vector of random tapes, $\vec{R}$, consider the deterministic protocol derived from it. In such a protocol there are at least

$$2^n\left(1-\sqrt{\frac{\delta}{2}}\right) - 2^n\left(1 - \frac{AS(f)-\sqrt{AS(f)}}{2n}\right) = 2^n\left(\frac{AS(f)-\sqrt{AS(f)}}{2n} - \sqrt{\frac{\delta}{2}}\right)$$

late input assignments; that is, $E_{\vec{x}}[L(\vec{x}, \vec{R})] \geq (\frac{AS(f)-\sqrt{AS(f)}}{2n} - \sqrt{\frac{\delta}{2}})$. Thus

$$E_{\vec{r},\vec{x}}[L(\vec{x}, \vec{r})] \geq (1-\sqrt{2\delta}) \cdot \left(\frac{AS(f)-\sqrt{AS(f)}}{2n} - \sqrt{\frac{\delta}{2}}\right) .$$

It follows that there is at least one input assignment $\vec{x}$ for which

$$E_{\vec{r}}[L(\vec{x}, \vec{r})] \geq (1-\sqrt{2\delta}) \cdot \left(\frac{AS(f)-\sqrt{AS(f)}}{2n} - \sqrt{\frac{\delta}{2}}\right) ,$$

which implies that

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

as claimed. □

The following gives the lower bounds for the function `xor`.

Corollary 5.8. *For fixed $\delta < 1/2$ let $\mathcal{A}$ be a $(1-\delta)$-correct d-random* expected *r-round private protocol to compute* `xor` *of n bits. Then $r = \Omega(\log n/d)$. (Obviously the same lower bound holds for r-round protocols.)*

Proof. Follows from Theorem 5.7 and the fact that $AS(\texttt{xor}) = n$. Note that the expression $(1-\sqrt{2\delta})(\frac{1}{2} - \sqrt{\frac{\delta}{2}} - \frac{1}{2\sqrt{n}})$ is greater than 0 for any $\delta < 1/2$ (and sufficiently large n). □

[8] Otherwise, $E_{\vec{r},\vec{x}}[G(\vec{x}, \vec{r})] < (1-\sqrt{2\delta}) \cdot 1 + \sqrt{2\delta} \cdot (1-\sqrt{\frac{\delta}{2}}) = 1 - \sqrt{2\delta} + \sqrt{2\delta} - \delta = 1-\delta$. Thus there is at least one input assignment $\vec{x}$ such that $E_{\vec{r}}[G(\vec{x}, \vec{r})] < 1-\delta$, which is a contradiction to the protocol being $1-\delta$ correct.

6. Conclusion. In this paper we initiate the quantitative study of randomness in private computations. As mentioned in the introduction, our work was already followed by additional work on this topic [BDPV95, KM96, KOR96, CKOR97].

We give upper bounds and lower bounds on the number of rounds required for computing `xor` privately with a given number of random bits. Alternatively, we give bounds on the number of random bits required for computing `xor` privately within a given number of rounds. Our lower bounds extend to other functions in terms of their sensitivity (and average sensitivity).

An obvious open problem is to close the gap between the upper bound and the lower bound for computing `xor` using d random bits. One possible way of doing this is to improve the bound given by Lemma 4.8.

Acknowledgments. We thank Gábor Tardos for improving the constant and simplifying the proof of Lemma 4.8, and Demetrios Achlioptas for his help in simplifying the proof of Lemma A.1. We also thank Benny Chor for useful comments.

REFERENCES

[AGHP90] N. Alon, O. Goldreich, J. Hastad, and R. Peralta, "Simple constructions of almost k-wise independent random variables", FOCS 90 and *Random Structures & Algorithms*, Vol 4, 1993.

[BB89] J. Bar-Ilan, and D. Beaver, "Non-Cryptographic Fault-Tolerant Computing in a Constant Number of Rounds", Proc. of 8th PODC, 1989, pp. 201–209.

[B89] D. Beaver, "Perfect Privacy for Two-Party Protocols", TR-11-89, Harvard University, 1989.

[BFKR90] 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.

[BGG90] M. Bellare, O. Goldreich, and S. Goldwasser, "Randomness in Interactive Proofs", Proc. of 31st FOCS, 1990, pp. 563–571.

[BGW88] 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.

[BM84] M. Blum, and S. Micali "How to Generate Cryptographically Strong Sequences Of Pseudo-Random Bits", FOCS 82 and *SIAM J. on Computing,* Vol 13, 1984, pp. 850–864.

[BGS94] C. Blundo, A. Giorgio Gaggia, and D. R. Stinson, "On the Dealer's Randomness Required in Secret Sharing Schemes", *Proc. of EuroCrypt94*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 950, 1995, pp. 35–46.

[BSV94] C. Blundo, A. De-Santis, and U. Vaccaro, "Randomness in Distribution Protocols", *Proc. of 21st ICALP*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 820, 1994, pp. 568–579.

[BDPV95] 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, 1995, pp. 171–182.

[CG90] R. Canetti, and O. Goldreich, "Bounds on Tradeoffs between Randomness and Communication Complexity", FOCS 90 and *Computational Complexity* Vol. 3, (1993), 141-167.

[CKOR97] R. Canetti, E. Kushilevitz, R. Ostrovsky, and A. Rosén, "Randomness vs. Fault-Tolerance", Proc. of 16th PODC, 1997.

[CCD88] D. Chaum, C. Crepeau, and I. Damgard, "Multiparty Unconditionally Secure Protocols", Proc. of 20th STOC, 1988, pp. 11–19.

[CK89] B. Chor, and E. Kushilevitz, "A Zero-One Law for Boolean Privacy", STOC 89 and *SIAM J. Disc. Math.* Vol. 4, (1991), 36–47.

[CK92] B. Chor, and E. Kushilevitz, "A Communication-Privacy Tradeoff for Modular Addition", *Information Processing Letters*, Vol. 45, 1993, pp. 205–210.

[CGK90] B. Chor, M. Geréb-Graus, and E. Kushilevitz, "Private Computations Over the Integers", FOCS 90 and *SIAM Jour. on Computing*, Vol. 24, No. 2, 1995, pp. 376-386.

[CGK92] B. Chor, M. Geréb-Graus, and E. Kushilevitz, "On the Structure of the Privacy Hierarchy", *Journal of Cryptology*, Vol. 7, No. 1, 1994, pp. 53-60.

[CG88] B. Chor, and O. Goldreich, "Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity", FOCS 85 and *SIAM J. Computing* Vol. 17, (1988), 230–261.

[CW89] A. Cohen, and A. Wigderson, "Dispersers, Deterministic Amplification, and Weak Random Sources", Proc. of 30th FOCS, 1989, pp. 14–19.

[FY92] M. Franklin, and M. Yung, "Communication Complexity of Secure Computation", Proc. of 24th STOC, 1992, pp. 699–710.

[IZ89] R. Impagliazzo, and D. Zuckerman, "How to Recycle Random Bits", Proc. of 30th FOCS, 1989, pp. 248–253.

[KK94] D. Karger, and D. Koller, "(De)randomized Construction of Small Sample Spaces in NC", Proc. of 35th FOCS, 1994.

[KM93] D. Koller, and N. Megiddo, "Constructing Small Sample Spaces Satisfying Given Constraints", Proc. of 25th STOC, 1993, pp. 268–277.

[KM94] H. Karloff, and Y. Mansour, "On Construction of k-wise Independent Random Variables", Proc. of 26th STOC, 1993, pp. 564–573.

[KPU88] D. Krizanc, D. Peleg, and E. Upfal, "A Time-Randomness Tradeoff for Oblivious Routing", Proc. of 20th STOC, 1988, pp. 93–102.

[K89] E. Kushilevitz, "Privacy and Communication Complexity", FOCS 89 and SIAM Jour. on Disc. Math., Vol. 5, No. 2, 1992, pp. 273–284.

[KM96] E. Kushilevitz, and Y. Mansour, "Randomness in Private Computations", Proc. of 15th PODC, 1996, pp. 181–190. To appear in *SIAM J. Disc. Math.*

[KMO94] E. Kushilevitz, S. Micali, and R. Ostrovsky, "Reducibility and Completeness in Multi-Party Private Computations", Proc. of 35th FOCS, 1994, pp. 478-489.

[KOR96] E. Kushilevitz, R. Ostrovsky, and A. Rosén, "Characterizing Linear Size Circuits in Terms of Privacy", Proc. of 28th STOC, 1996, pp. 541-550.

[NN90] J. Naor, and M. Naor, "Small-Bias Probability Spaces: Efficient Constructions and Applications", STOC 90, and *SIAM J. on Computing,* Vol 22, No. 4, 1993, pp. 838–856.

[N90] N. Nisan, "Pseudorandom Generator for Space Bounded Computation", Proc. of 22nd STOC, 1990, pp. 204–212.

[RS89] P. Raghavan, and M. Snir, "Memory vs. Randomization in On-Line Algorithms", Proc. of 16th ICALP, 1989, pp. 687–703.

[S92] L. J. Schulman, "Sample Spaces Uniform on Neighborhoods", Proc. of 24th STOC, 1992, pp. 17–25.

[VV85] U. Vazirani, and V. Vazirani, "Random Polynomial Time is Equal to Slightly-Random Polynomial Time", Proc. of 26th FOCS, 1985, pp. 417–428.

[Y82] A. C. Yao, "Theory and Applications of Trapdoor Functions", Proc. of 23rd FOCS, 1982, pp. 80–91.

[Z91] D. Zuckerman, "Simulating BPP Using a General Weak Random Source", Proc. of 32nd FOCS, 1991, pp. 79–89.

A. Improved Lower Bounds for Some Special Cases.

A.1. A Special Case – Each Player Sends a Single Message. In Section 3 we show that `xor` can be computed privately with d random bits, in $O(\log n/\log d)$ rounds.

Corollary 4.14 shows a lower bound of $\Omega(\log n/d)$ rounds for such a computation. In this section we prove a stronger lower bound than the one proved in Section 4 but in a weaker model. In this model, each player is allowed to send a single non-constant message. More precisely, each player sends only a single non-constant message to a specific other player, and this other player is the same in all runs. Note that the protocol presented in Section 3 has this property.

Again, we consider the protocol obtained by fixing a *specific* vector of random tapes $\vec{R} = (R_1, \ldots, R_n)$. The main observation is that the above property implies that if a player receives two messages then the sets of variables on which the two messages depend are disjoint.

In the following we prove an extension of Lemma 4.12, which gives an upper bound on the sensitivity of the view of the players. To do so, we first prove a stronger version of Lemma 4.8 for the case that each function depends on a different set of variables. We then give a slight variation of the proof of Lemma 4.12, using the new lemma instead of Lemma 4.8.

LEMMA A.1. *Let $\mathcal{F} = \{f_j\}, 1 \leq j \leq m$ be a set of m functions $f_j : \{0,1\}^n \rightarrow \{0,1\}$, for some n. Assume $S(f_j) \leq C$ for all j. Further assume that, for all $i \neq j$, $\mathcal{D}(f_i) \cap \mathcal{D}(f_j) = \emptyset$. Define $F(Y) = (f_1(Y), \ldots, f_m(Y))$. If F assumes at most 2^d different values (different vectors), then the sensitivity of F is at most $C \cdot d$.*

Proof. Let $\mathcal{F}' \subseteq \mathcal{F}$ be the set of functions which are not constant functions. Let $m' = |\mathcal{F}|$ be the cardinality of this set. Since for every $i \neq j$ such that $f_i, f_j \in \mathcal{F}'$, we have $\mathcal{D}(f_i) \cap \mathcal{D}(f_j) = \emptyset$, then the number of values assumed by F is $2^{m'}$. Therefore $m' \leq d$.

Let Y be the assignment on which F has the largest sensitivity, i.e., $|\mathcal{S}_f(Y)| \geq |\mathcal{S}_f(Y')|$ for any assignment Y'. If F is sensitive to i on Y, then there is some functions $f_j \in \mathcal{F}'$ such that f_j is sensitive to i on Y. However every function f_j has sensitivity at most C. Therefore F is sensitive on Y to at most $C \cdot m' \leq C \cdot d$ variables. $\square$

We remark that the bound proved by the above lemma is tight.

LEMMA A.2. *Consider a private d-random protocol to compute a boolean function f with the additional property under consideration, and consider a specific vector of random tapes $\vec{R}$. Then $View_i^k(\vec{x}, \vec{R})$, for any player P_i, has sensitivity of at most $Q(k) \triangleq (d+1)^{k-1}$.*

Proof. First note that since we fix the random tapes the views of the players are functions of the input assignment $\vec{x}$ only. We prove the lemma by induction. For $k=1$ the view of any player depends only on its single input bit. Thus the claim is obvious. For $k > 1$ assume the claim holds for any $\ell < k$. This implies in particular that all messages received by player P_i and included in the view under consideration have sensitivity of at most $Q(k-1)$. Consider the view without the input bit x_i and denote it by F; this is composed of only the messages received by P_i and the bits read from the (local) random tape. Using Lemma 4.11, and Lemma A.1 with $C = Q(k-1)$, we have that F has sensitivity at most $Q(k-1) \cdot d = (d+1)^{k-2} \cdot d$. The view under consideration, $View_i^k(\vec{x}, \vec{R})$, has an additional single bit, which has clearly sensitivity at most 1. We thus have that the sensitivity of $View_i^k(\vec{x}, \vec{R})$ is at most $(d+1)^{k-2} \cdot d + 1 \leq Q(k)$. $\square$

THEOREM A.3. *Given a private d-random protocol ($d \geq 2$) to compute a boolean function f that has the special property under consideration, consider the deterministic protocol derived from it by any given vector of random tapes $\vec{R}$. For every player P_i, there is at least one input assignment $\vec{x}$ such that $T_i(\vec{x}, \vec{R}) = \Omega(\log S(f)/\log d)$.*

The proof follows the proof of Theorem 4.13, using Lemma A.2 instead of Lemma 4.12. The lower bound of this section is stated in the following corollary.

COROLLARY A.4. *Let $\mathcal{A}$ be an r-round d-random private protocol ($d \geq 2$) to compute a boolean function f that has the property under consideration. Then $r = \Omega(\log S(f)/\log d)$.*

The following lower bound applies for the function `xor`.

COROLLARY A.5. *Let $\mathcal{A}$ be an r-round d-random private protocol ($d \geq 2$) to compute* `xor` *of n bits that has the property under consideration. Then $r = \Omega(\log n/\log d)$.*

A.2. A Special Case – Protocols with Messages of a Special Type. As proved in Section 3, `xor` can be computed privately with d random bits in $O(\log n/\log d)$ rounds. For general protocols, Corollary 4.14 shows a lower bound of $\Omega(\log n/d)$ rounds, while in a special case (in which the upper bound falls) Corollary A.5 gives a tight $\Omega(\log n/\log d)$ lower bound on the number of rounds. Here we consider another special case in which each message m can be expressed as a boolean function $f(\vec{x}) \oplus g(\vec{r})$, of the n-entry input vector $\vec{x}$, and a d-entry binary random tape, $\vec{r}$. Again, our upper bound satisfies this restriction, and the lower bound we prove in this case is tight (i.e., $\Omega(\log n/\log d)$). One can think of such protocols as those which are designed (as we do in Section 3) by first designing a non-private protocol, and then "masking" the messages with random noise (which is canceled at the end). In particular, each player can toss all of its coins before the protocol starts (the number of coins tossed may depend however on the outcome of previous coin tosses). The particular masking bit that is used for each message is not to depend on the input to the players.

The idea of the proof is to consider the number of input variables which are *good* (with respect to the computed function f) in the view of a player at some round k (as in Section 4.3.1[9]). Clearly, the input variable of the player may be good. Other good variables can be those that are good for any of the messages that this player received in a previous round. The key ingredient in the proof will be to show that the number of messages received by any player, that are "beneficial" to increase the number of good variables in its view, is at most d (otherwise, the privacy requirement is violated). From this we get that the number of good variables at round k is $O((d+1)^{k-1})$. In order for a player to output the correct value of f its view must have at least $S(f)$ good variables. The result will follow.

More precisely, consider a private protocol (with the property under consideration) to compute a boolean function f. Consider the messages $M_1, \ldots, M_m$ received by some player P_j during the protocol. By assumption, we can write each of these messages, M_ℓ, as $M_\ell(\vec{x}, \vec{r}) = f_\ell(\vec{x}) \oplus g_\ell(\vec{r})$. Denote by Y an input assignment on which f achieves

[9] Let Y be an assignment and f be the computed function. Recall that for a given function m, a variables x_j is called *good* for m on Y if both m and f are sensitive to x_j on Y.

its sensitivity $S(f)$. Let $\mathcal{G}_{M_\ell}(Y)$ denote the set of *good* variables of the message M_ℓ on Y; henceforth we refer by *good* to "good on Y". Note that for any message M_ℓ, due to the special structure of messages in this protocol, $\mathcal{G}_{M_\ell}(Y) = \mathcal{G}_{f_\ell}(Y)$. First, ignore those messages M_ℓ for which $\mathcal{G}_{M_\ell}(Y) \subseteq \{x_j\}$. Assume that the rest of the messages are ordered in a way such that each message M_i (where $i \leq m' \leq m$) has at least one good variable that is not good for any of $M_1, \ldots, M_{i-1}$ (i.e., $\mathcal{G}_{M_i}(Y) \setminus \cup_{i'<i} \mathcal{G}_{M_{i'}}(Y) \neq \emptyset$). Further assume that $|\cup_{i=1}^{m'} \mathcal{G}_{M_i}(Y) \cup (\mathcal{S}_f(Y) \cap \{x_j\})| \leq S(f) - 1$. We first present a claim which gives a property of $g_1, \ldots, g_{m'}$. This claim is then used to prove that $m' \leq d$.

Claim A.6. *Let $M_1, \ldots, M_{m'}$ be as above, and let $g_1, \ldots, g_{m'}$ be the corresponding g_i's. Then, the vector $(g_1(\vec{r}), \ldots, g_{m'}(\vec{r}))$ assumes all $2^{m'}$ values (over the choices of the random tapes $\vec{r}$).*

Proof. Suppose this is not the case. Then there exists a minimal k such that $(g_1(\vec{r}), \ldots, g_{k-1}(\vec{r}))$ assumes all 2^{k-1} values but $(g_1(\vec{r}), \ldots, g_k(\vec{r}))$ does not assume all 2^k values. In particular there are $k-1$ bits $a_1, \ldots, a_{k-1}$ for which (without loss of generality) the values vector $(a_1, \ldots, a_{k-1}, 0)$ is assumed but $(a_1, \ldots, a_{k-1}, 1)$ is not.

By the assumptions on the M_i's, there exists a variable x_{i_k}, $i_k \neq j$, such that x_{i_k} is good for M_k (and hence for f_k), but is not good for any of $f_1, \ldots, f_{k-1}$. This implies that both f and f_k are sensitive to x_{i_k} (on Y). In particular, $f_k(Y) \neq f_k(Y^{(x_{i_k})})$. On the other hand, for any $\ell < k$, $f_\ell(Y) = f_\ell(Y^{(x_{i_k})})$. Since $|\cup_{i=1}^{m'} \mathcal{G}_{M_i}(Y) \cup (\mathcal{S}_f(Y) \cap \{x_j\})| \leq S(f) - 1$, there is a variable x_s, $s \neq j$, such that f is sensitive to x_s on Y, but none of the f_i, $i \leq k$, is.

Now consider the two input assignments $Y^{(x_s)}$ and and $Y^{(x_{i_k})}$. Without loss of generality assume that $(f_1(Y), \ldots, f_k(Y)) = (0, \ldots, 0)$. Then, $(f_1(Y^{(x_s)}), \ldots, f_k(Y^{(x_s)}) = (0, \ldots, 0)$, and $(f_1(Y^{(x_{i_k})}), \ldots, f_k(Y^{(x_{i_k})}) = (0, \ldots, 0, 1)$. Therefore, on the assignment $Y^{(x_s)}$, player P_j can see (with a positive probability) the vector of messages $(a_1, \ldots, a_{k-1}, 0)$, but not $(a_1, \ldots, a_{k-1}, 1)$, and vice versa for the input assignment $Y^{(x_{i_k})}$. Since $f(Y^{(x_{i_k})}) = f(Y^{(x_s)})$, and x_j is equal in both of them, the privacy requirement (with respect to P_j) is violated. $\square$

The important consequence of the above claim is that $m' \leq d$. This is because, by Lemma 4.10, there are at most 2^d different vectors for $(g_1(\vec{r}), \ldots, g_{m'}(\vec{r}))$. We are now ready to prove the theorem:

Theorem A.7. *Let $\mathcal{A}$ be an r-round d-random private protocol ($d \geq 2$) to compute a boolean function f, such that $\mathcal{A}$ has the property that each message M_ℓ can be expressed as $f_\ell(\vec{x}) \oplus g_\ell(\vec{r})$. Then $r = \Omega(\log S(f)/\log d)$.*

Proof. Recall that Y is an input assignment on which f has sensitivity $S(f)$ and that by "good" we refer to "good on Y". We first prove, by induction, that the view of any player in round k, on input Y, can have at most $(d+1)^{k-1}$ good variables, or at least $S(f)$ good variables. This is certainly true in the first round, where the view can have at most one good variable. Now consider the view of P_j in round k. Denote by M the set of messages received by the player. If the view has less than $S(f)$ good variables, then $|\cup_{m \in M} \mathcal{G}_m(Y) \cup (\mathcal{S}_f(Y) \cap \{x_j\})| \leq S(f) - 1$. By Claim A.6, it follows that there can be at most d messages each of which has at least one good variable which is not x_j, and such that each one has at least one such good variable that the previous ones (in the order we defined) do not have. Each of these messages was received in one of the previous rounds, and has less than $S(f)$ good variables, hence by the induction hypothesis has at most $(d+1)^{k-2}$ good variables. Thus, the view of P_j at round k can have at most $d \cdot (d+1)^{k-2} + 1 \leq (d+1)^{k-1}$ good variables.

To prove the theorem, fix some vector of random tapes $\vec{R}$ and consider the views obtained when running the deterministic protocol derived by it. As remarked before, due to the special type of message, the set of good variables is the same for all random tapes. Obviously, the view of a player that outputs the value f must have $S(f)$ good variables (on Y). Let t be the *first* round in which some player P_j has view with at least $S(f)$ good variables. Order the messages received by P_j in round t in an arbitrary order. Then, there is the first message m after which the view of P_j has at least $S(f)$ good variables. This message m has at most $(d+1)^{t-2}$ good variables, by the above claim (as m was produced from a view of a previous round, which has less than $S(f)$ good variables). On the other hand, just before this message is received, the view of the player does not have $S(f)$ good variables; therefore, by the above claim, it has at most $(d+1)^{t-1}$ good variables. We thus have,

$$S(f) \leq (d+1)^{t-1} + (d+1)^{t-2} ,$$

which gives the required lower bound on t. □

COROLLARY A.8. *Let $\mathcal{A}$ be an r-round d-random private protocol ($d \geq 2$) to compute* `xor` *of n bits that has the property under consideration. Then $r = \Omega(\log n / \log d)$.*