$\Omega(\log N)$ LOWER BOUNDS ON THE AMOUNT OF RANDOMNESS IN 2-PRIVATE COMPUTATION*

ANNA GÁL † AND ADI ROSÉN‡

Abstract. We consider the amount of randomness necessary in information-theoretic private protocols. We prove that at least $\Omega(\log n)$ random bits are necessary for the t-private computation of the function xor by n players, for any $t \geq 2$. In view of the upper bound of $O(t^2 \log(n/t))$ [23], this bound is tight, up to constant factors, for any fixed t. For a class of protocols obeying certain restrictions, we give a stronger lower bound of $\Omega(t \log(n/t))$. We note that all known randomness efficient private protocols designed specifically for xor belong to this class. In fact we prove slightly stronger statements: we prove that on *every* input there is a run where the number of random bits used is large, rather than only proving that on some input there is a run where the number of random bits used is large. All our lower bounds hold for the "trusted dealer" model as well, and the $\Omega(t \log(n/t))$ lower bound for restricted protocols is tight, up to constant factors, for any $t \geq 2$ in this model.

In comparison, the previous lower bounds on the amount of randomness required by t-private computation of explicit functions did not grow with n for constant values of t, and our results improve the previous lower bounds for xor for any $2 \leq t = o(\log n)$. Our results also show that already for $t = 2$, $\Omega(\log n)$ random bits are necessary, while it is known that for the case of $t = 1$ a single random bit is sufficient for privately computing xor for any number of players.

Our proofs use novel techniques by which we extract random variables from a t-private protocol, and then use the t-privacy property of the protocol to prove properties of these random variables. These properties in turn imply that the number of random bits used by the players is large.

Key words. private computation, randomness, lower bounds

AMS subject classifications. 68R05, 94A60, 68M10

1. Introduction.

A *t-private* protocol for computing a function f is a distributed 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 does not reveal any "unnecessary" information to any coalition of at most t players. The players proceed in rounds, where in each round each player can send a private message to any other player (i.e., each player sends to each other player a message that cannot be seen by any of the remaining players). The t-privacy property means that any coalition of at most t players cannot learn anything from the execution of the protocol, except what is implied by the value of $f(\vec{x})$ and the inputs of the members of the coalition. In particular, the members of the coalition do not learn anything about the inputs of the other players. Private computation in this setting was the subject of considerable research, see e.g. [2, 3, 4, 6, 7, 11, 13, 14, 15, 16, 17, 19, 23, 21, 24, 25, 30]. Randomness is necessary to perform private computations involving more than two players (except for the computation of very degenerate functions). That is, the players must have access to a random source. As randomness is regarded as a scarce resource, methods for saving random bits in various contexts have been suggested in the literature, see e.g. [29, 18] for a survey. Thus, an important research topic is the design of randomness-efficient private protocols, and the quantification of the amount of randomness needed to perform private computations of various functions and under

[*] An early version of this paper appeared in the Proceedings of the 35th STOC (2003), pp. 659-666.

[†] Dept. of Computer Science, University of Texas at Austin, Austin, TX 78712-1188, USA. e-mail: panni@cs.utexas.edu. Supported in part by NSF CAREER Award CCR-9874862 and an Alfred P. Sloan Research Fellowship.

[‡] Dept. of Computer Science, Technion, Haifa 32000, Israel. email: adiro@cs.technion.ac.il.

various constraints. This line of research has received considerable attention in recent years, see e.g. [28, 23, 26, 17, 7, 8, 10, 27, 5, 20]. This study also showed that the randomness complexity of the private computation of a function is related to other complexity measures, such as sensitivity and circuit size [28, 26, 17, 7, 27]. The specific function xor (addition modulo 2) was the subject of considerable research in this context due to its being a basic operation and its relative simplicity [28, 27, 8, 23].

Previous work on the randomness complexity of private computations revealed that there is a tradeoff between randomness and time (i.e., number of communication rounds) for the 1-private computation of the function xor [28, 17]. These works also gave lower bounds on the number of rounds necessary to 1-privately compute any function, in terms of the sensitivity of the function and the amount of randomness used. If one is allowed an arbitrary number of rounds for the computation, there are no known lower bounds on the number of random bits for 1-private protocols computing explicit functions (except that randomness is necessary, i.e., no deterministic private protocol exists). In fact, Kushilevitz et al. [26] gave a relation between the number of random bits necessary to 1-privately compute a function, and the Boolean circuit size necessary to compute it; it is proved that the class of functions that have $O(1)$-random, 1-private, protocols is equal to the class of functions that have linear size circuits. This surprising connection explains the lack of $\omega(1)$ lower bounds on the number of random bits for explicit functions in the case of 1-privacy, as such results would imply superlinear lower bounds on circuit size.

Before our work, $\omega(1)$ lower bounds on the number of random bits of t-private protocols (without limiting the number of rounds) have been proved for explicit functions only for values of t that grow with n, and no such bounds have been known if t itself is constant. More precisely, Kushilevitz and Mansour [23] proved that any t-private protocol for xor requires at least t random bits. Blundo et al. [7] gave lower bounds for two special cases. Namely, they proved that if $t = n - c$, for some constant c, then $\Omega(n^2)$ random bits are necessary, and if $t \geq (2 - \sqrt{2})n$, then $\Omega(n)$ random bits are necessary. As to upper bounds, Canetti et al. [10] gave randomness-efficient generic protocols to t-privately compute (for $t < n/2$) any Boolean function f. They showed that any function f with circuit size of m gates can be computed by a t-private protocol ($t < n/2$) using $O(t^2 \log n + (m/n)t^5 \log t)$ random bits. Kushilevitz and Mansour [23] gave protocols that compute the function xor t-privately, for any t, using $O(t^2 \log(n/t))$ random bits.

In the present paper we develop new techniques for proving lower bounds on the number of random bits necessary in t-private computations (for $t \geq 2$), and obtain $\Omega(\log n)$ lower bounds on the number of random bits necessary to t-privately compute the function xor for any $t \geq 2$.[1] More precisely, we prove the following theorem. (See Section 2 for a formal definition of a d-random protocol).

THEOREM 1.1. *Let $t \geq 2$, and let $\mathcal{A}$ be a d-random, t-private, protocol for computing $f(\vec{x}) = x_1 + \ldots + x_n (\text{mod} 2)$. Then $d = \Omega(\log n)$.* In fact, we prove a slightly stronger statement: we prove that $\Omega(\log n)$ random bits are necessary on *every* input.

In view of the upper bound of $O(t^2 \log(n/t))$ of [23], our lower bound is tight, up to constant factors, for any fixed t. This is the first result showing that the number of random bits necessary for t-private computation grows with n for constant values of t, and it improves the lower bound of [23] for any $t = o(\log n)$. It is interesting

[1] Blundo et al. [9] recently reported obtaining similar results independently, using a different approach.

to note that our $\Omega(\log n)$ lower bound holds already for $t = 2$, while for the case of $t = 1$, it is known that the function `xor` can be computed 1-privately, for any number of players n, with only 1 random bit.

All known randomness-efficient private protocols designed specifically for the function `xor` [28, 23, 27, 8] are built in the following special way. They are based on a deterministic, non-private, protocol for `xor`. Then, this protocol is modified by changing any message so it is the sum (modulo 2) of the original message, and a value which is a function of the random bits only. Thus, the private protocol is built by masking the original messages of the non-private protocol. We give stronger lower bounds for protocols of this class (see Section 4 for a formal definition of this class). Namely, we give a lower bound of $\Omega(t \log(n/t))$ on the number of random bits required by any protocol of this class, to compute `xor` for n players.

All our lower bounds hold also in the "trusted dealer" model, considered in [23, 10]. In this model, the n players are deterministic, and there is an additional player, the "trusted dealer", who does not get any input, and whose role is limited to "deal" random bits to the other players (hence a "dealer"). This player never participates in any coalition (hence it is "trusted"). For this model, our lower bound of $\Omega(t \log(n/t))$ for protocols of the above restricted class is tight up to constant factors for every $t \geq 2$, as [23] gave a protocol (of this class) in the trusted dealer model using $O(t \log(n/t))$ random bits.

Our proofs use novel techniques by which we extract from a private protocol random variables that depend on the randomness that the players use. We then use the t-privacy property of the protocol to prove that these random variables must have certain properties (for example, linear independence or t-wise independence). Based on these properties we show that the amount of randomness used by the players must be large. We believe that these new techniques may prove useful for proving other properties of private protocols.

2. Preliminaries. In this paper we consider information-theoretic privacy (as in [4, 11]), where the players have unlimited computational power, no intractability assumptions are made, and messages are sent over private channels.

Let $f : \{0,1\}^n \to \{0,1\}$ be an arbitrary Boolean function. A set of n players P_i ($1 \leq i \leq n$), each possessing a single private input bit x_i (i.e., x_i is known *only* to P_i), collaborate in a protocol to compute the value of $f(\vec{x})$. The protocol is probabilistic. During the course of the protocol each player can toss random coins, where the coin tosses are unbiased and independent. 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 access them). The player then receives the messages sent to it by the other players. Each player chooses to output the value of the function at a certain round. In a correct protocol, each player must output the correct value $f(\vec{x})$, and stop its operation in a finite number of steps (it may output $f(\vec{x})$ before stopping). That is, for every input assignment $\vec{x}$ and for every outcome of the coin tosses of all players, each player outputs $f(\vec{x})$, and stops in a finite number of steps.

In Claim 1 below we formally argue that for a given correct protocol involving n players, there is a finite upper bound ℓ on the number of coin tosses any single player performs during the course of the protocol. We in fact show that there is a finite upper bound ℓ on the total number of coin tosses performed by all players. Claim 1 gives a formal argument for the intuition that a protocol for which such upper bound does not exist is not a correct protocol. For example, consider a protocol where some player keeps tossing coins until it gets a 1, before sending any message. For such protocol, the property claimed does not hold. But, such a protocol does not satisfy the definition of correctness either, as there are possible outcomes of the coin tosses for which some player does not stop in a finite number of steps. We note that since we prove lower bounds on the number of random bits used, we could avoid using Claim 1 by the following argument: we could argue that if on some input, some player may toss more than ℓ coins, then a lower bound of ℓ is obtained; otherwise one can assume that no player ever tosses more than ℓ coins. This would be sufficient to prove lower bounds on the randomness complexity of the protocol (see Definition 2.1). However, we prove stronger statements. Claim 1 is useful in proving that on *every* input there is a run where the number of coin tosses performed is large, rather than only proving that on some input there is a run where the number of coin tosses performed is large.

Claim 1. *Given a correct protocol involving n players, there is a finite upper bound ℓ on the total number of coin tosses performed by all players in any run of the protocol.*

Proof. We show below that for any input assignment $\vec{x}$ there is a finite upper bound $\ell(\vec{x})$ on the total number of coin tosses performed by all players in any run of the protocol in which the input assignment is $\vec{x}$. Since there is a finite number of input assignments $\vec{x} \in \{0,1\}^n$, the claim follows by letting $\ell = \max_{\vec{x} \in \{0,1\}^n}\{\ell(\vec{x})\}$.

Fix an input assignment $\vec{x} \in \{0,1\}^n$. As in the proof of Lemma 4.10 in [28], we build a binary tree $T_{\vec{x}}$ representing the coin tosses of the players on a given input $\vec{x}$. Each node of the tree is labeled by the name of a player P_i, which tosses a coin. The two outgoing edges from a node are labeled 0 and 1 according to the outcome of the coin toss. Coin tosses in the run of the protocol are ordered by round number, then by player number, and then by a serial number (for that player in that round). Note that the identity of the player to toss the first coin on $\vec{x}$, that is, the label of the root, is determined by $\vec{x}$, and the identity of any subsequent player to toss a coin is determined by $\vec{x}$ and the outcomes of the previous coin tosses, that is, by the path leading to a given node of the tree.

Observe that a path of length k from the root to another node represents a run (or a prefix of a run) of the protocol in which k coin tosses occur. Assume towards a contradiction that there is no finite upper bound $\ell(\vec{x})$ on the number of coin tosses performed by the players on input $\vec{x}$. Then, for every finite k there is a path of length k in the tree. Since the outdegree of each node of the tree is at most 2, this means that the tree $T_{\vec{x}}$ must contain an infinite path starting from the root. (cf. König's lemma in [22].) This path corresponds to a possible run of the protocol (defined by $\vec{x}$ and the results of the coin tosses as defined by the edges along the path). In this run at least one player tosses an infinite number of coins, i.e., this player does not stop in a finite number of steps, contradicting the correctness of the protocol. □

We thus model the players in a correct protocol as being provided with finite binary random tapes. That is, in a correct protocol involving n players to compute a function f each player P_i is provided with a local binary random tape R_i of length ℓ. Note that the value of ℓ may be different for different protocols, and different number of players n. The bits in the random tapes are unbiased and independent. We denote by $\vec{R} = (R_1, \ldots, R_n)$ a given vector of random tapes of all players, and we think of $\vec{R}$ as a binary vector of length $n\ell$.

The following definition is used to measure the amount of randomness used in a protocol.

Definition 2.1. *(***Randomness complexity of a protocol***) A d-random pro-*

tocol is a protocol such that for any input assignment and any vector of local random tapes, the total number of random bits read from the local random tapes by all players is at most d.

We emphasize that the definitions allow, for example, that in different executions of a protocol (i.e., different input assignments and different local random tapes), a given player reads a different number of random bits from its local random tape. The number of random bits read by the player may depend on both the inputs of the players, and the random bits read by all players.

We now proceed to consider the messages exchanged by the players. Each player P_i receives during the execution of the protocol a sequence of messages. In different runs of a protocol the various players may receive different messages. These depend on the input to the players and on the random tapes. We denote the communication seen by a player as follows.

Definition 2.2. *The* communication $c_i(\vec{x}, \vec{R})$, *of player P_i, on input $\vec{x}$ and vector of random tapes $\vec{R} = \{R_i\}_{1 \leq i \leq n}$, is the sequence of messages that player P_i receives during the execution of the protocol, when the input is $\vec{x}$ and the vector of random tapes of all players is $\vec{R}$.*

Thus, c_i is the (random) variable of the sequence of messages received by P_i. For a subset of the players S, we denote by c_S the (random) variable of the sequences of messages received by all the players in S. Informally, *t-privacy* means that any coalition of up to t players cannot learn anything (in particular, the inputs of the other players) from the communication that the members of the coalition receive, except what is implied by the input bits of the members of that coalition, and the value of the function computed. Formally,

Definition 2.3. **(Privacy)** *A protocol for computing a function f is private with respect to a subset of the players $S \subseteq [n]$ if the following holds. 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 $i \in S$, and for any sequence of messages C_S, and for any vector of random tapes for the subset S, $\{R_i\}_{i \in S}$,*

$$Pr[c_S = C_S | \{R_i\}_{i \in S}, \vec{x}] = Pr[c_S = C_S | \{R_i\}_{i \in S}, \vec{y}] \ ,$$

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

A protocol is said to be t-private if it is private with respect to any subset of players S, such that $|S| \leq t$.

It will be convenient in our proofs to use a weaker privacy requirement, directly implied by the t-privacy property, as stated in the following lemma. (Note that since we prove lower bounds, this makes our results only stronger.)

Lemma 2.4. *Consider any t-private protocol. For any subset S of the players $S \subseteq [n]$ such that $|S| \leq t$, and 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 $i \in S$, the following holds.*

1. *For any sequence of messages C_S,*

$$Pr[c_S = C_S | \vec{x}] = Pr[c_S = C_S | \vec{y}] \ ,$$

where the probability is over the vectors $\vec{R}$ chosen uniformly from $\{0,1\}^{n\ell}$.

2. *For any function ϕ^S of c_S, and for any value Φ in the range of ϕ^S,*

$$Pr[\phi^S = \Phi | \vec{x}] = Pr[\phi^S = \Phi | \vec{y}] \ ,$$

where the probability is over the vectors $\vec{R}$ chosen uniformly from $\{0,1\}^{n\ell}$.

Proof. Let s be the size of S, i.e., $s = |S|$. Fixing a vector of random tapes for the subset S, $\{R_i\}_{i\in S}$, is equivalent to fixing a binary vector of length $s\ell$. The probability of each of these $2^{s\ell}$ vectors is $2^{-s\ell}$, and the events corresponding to the various vectors are disjoint. Therefore we have

$$Pr[c_S = C_S|\vec{x}] = 2^{-s\ell} \sum_{\{R_i\}_{i\in S}\in\{0,1\}^{s\ell}} Pr[c_S = C_S|\{R_i\}_{i\in S}, \vec{x}] \ .$$

Using the same arguments, applied to $\vec{y}$ instead of $\vec{x}$, we have that

$$Pr[c_S = C_S|\vec{y}] = 2^{-s\ell} \sum_{\{R_i\}_{i\in S}\in\{0,1\}^{s\ell}} Pr[c_S = C_S|\{R_i\}_{i\in S}, \vec{y}] \ .$$

But, by the privacy property of the protocol we have that for any vector of random tapes for the subset S, $\{R_i\}_{i\in S}$,

$$Pr[c_S = C_S|\{R_i\}_{i\in S}, \vec{x}] = Pr[c_S = C_S|\{R_i\}_{i\in S}, \vec{y}] \ .$$

We therefore obtain that

$$Pr[c_S = C_S|\vec{x}] = Pr[c_S = C_S|\vec{y}] \ .$$

The second statement of the lemma follows by observing that the value of ϕ^S is fixed given any communication C_S. □

From the point of view of an observer of the protocol one can define the *transcript* of a given run of the protocol, which is the set of all messages sent between all players during the execution of the protocol on input $\vec{x}$ and vector of random tapes $\vec{R}$. The transcript is in fact the ordered vector of the communication of all players.

Definition 2.5. *The* transcript $Trans(\vec{x}, \vec{R})$ *of a protocol on input $\vec{x}$ and vector of random tapes $\vec{R} = \{R_i\}_{1\leq i\leq n}$, is $(c_1(\vec{x}, \vec{R}), c_2(\vec{x}, \vec{R}), \ldots, c_n(\vec{x}, \vec{R}))$.*

The following lemma follows immediately from the arguments of the proof of Lemma 4.10 in [28]. We will use this lemma in our proofs.

Lemma 2.6. *[28] For a given input $\vec{x}$, let d be the maximum, over all runs on input $\vec{x}$, of the total number of random bits read from the random tapes by all players during a given run. Then, the number of different transcripts of runs on input $\vec{x}$ is at most 2^d.*

It is convenient in our proofs to consider the messages sent by the players as being messages of single bits. This is done by "breaking" each message into the bits of its binary representation. Formally, for a given protocol involving n players, let M be the set of all different messages that can be sent in the protocol in all different runs (over all possible inputs $\vec{x} \in \{0,1\}^n$ and all possible vectors of random tapes $\vec{R} \in \{0,1\}^{n\ell}$). Fix an arbitrary one-to-one binary encoding of fixed length for the messages in M. We note that the empty message is one of the elements of M. We consider a protocol where each player sends instead of a given message from M, a sequence of single bit messages that represent the binary encoding of the original message from M. Henceforth, when we refer to *messages* we refer to these single bit messages. It is important for our argument that the number of transcripts of the protocol, on any given input $\vec{x}$, remains the same. This follows since we use a one-to-one encoding.

Since we consider each message as being a single bit, we can think of a given message m as a Boolean function of the input $\vec{x}$, which is a binary vector of length n, and the random tapes of all players, which is a binary vector of length $n\ell$. We therefore can write m as $m = m(\vec{x}, \vec{R})$.

Our lower bound exploits the fact that the function xor has large sensitivity on every input. Sensitivity is defined as follows.

Definition 2.7. **(Sensitivity)**

- *Given $\vec{x} \in \{0,1\}^n$, we denote by $\vec{x}^{(i)}$ the vector $\vec{x}$ with its i-th bit flipped. (Similarly, $\vec{x}^{(i,j)}$ is $\vec{x}$ with its i-th and j-th bits flipped.)*
- *A function f is* sensitive *to its i-th variable on input $\vec{x}$, if $f(\vec{x}) \neq f(\vec{x}^{(i)})$.*
- *$s(f,\vec{x})$ is the number of variables to which the function f is sensitive on input $\vec{x}$.*
- *The* sensitivity *of a function f is $s(f) = \max_{\vec{x}} s(f,\vec{x})$.*

Note that the function xor (addition modulo 2) of n binary variables is sensitive to all its n variables on any input. This immediately follows since for any $\vec{x} \in \{0,1\}^n$, and for any $i \in [n]$, $\texttt{xor}(\vec{x}) \neq \texttt{xor}(\vec{x}^{(i)})$.

We will further need the following definitions.

Definition 2.8.

- *A message m depends on the variable x_i if there exist $\vec{x}$ and $\vec{R}$, such that $m(\vec{x},\vec{R}) \neq m(\vec{x}^{(i)},\vec{R})$. In other words, m depends on the variable x_i, if m is sensitive to x_i on some $\vec{x}$ and $\vec{R}$.*
- *For $i \leq j$, a message m depends on a variable x_j under the partial assignment $x_1 = \alpha_1, \ldots, x_{i-1} = \alpha_{i-1}$, if there exists an assignment to the remaining variables $x_i, \ldots, x_n$, and there exists $\vec{R}$, such that $m(\alpha_1, \ldots, \alpha_{i-1}, x_i, \ldots, x_j, \ldots, x_n, \vec{R}) \neq m(\alpha_1, \ldots, \alpha_{i-1}, x_i, \ldots, \bar{x_j}, \ldots, x_n, \vec{R})$.*

We will use the following simple observation.

Observation 1. *Let $m = m(\vec{x},\vec{R}) = \phi(f_1(\vec{x},\vec{R}), \ldots, f_u(\vec{x},\vec{R}))$. If m depends on a variable x_j under the partial assignment $x_1 = \alpha_1, \ldots, x_{i-1} = \alpha_{i-1}$, then at least one of the functions $f_1, \ldots, f_u$ depends on the variable x_j under the partial assignment $x_1 = \alpha_1, \ldots, x_{i-1} = \alpha_{i-1}$.*

3. Lower Bound for General Protocols.

In this section we give a lower bound that applies to any t-private protocol for xor, for $t \geq 2$. We first outline our approach, which is common to the general lower bound and to the stronger lower bound for the restricted class of protocols (given in the next section), and then proceed to give the proof of Theorem 1.1.

3.1. Our Approach.

We state informally the approach we use in our proofs. Our proofs proceed in two stages. First, we prove that in any t-private protocol for xor we can identify $q = \Omega(n)$ distinct messages $m_1, \ldots, m_q$ with certain properties. Informally these properties are that

(1) We can permute the input vector (and accordingly the set of players), such that for any i, message m_i depends on input x_i, but does not depend on any input x_j, $j > i$.

(2) The set of receivers of these messages is disjoint from the set of players that have access to the inputs x_i, $1 \leq i \leq q$.

In the second stage of our proofs we consider the values of these selected messages on a given input assignment. That is, we consider the vectors representing the values of these messages over the possible random tapes to all players, when the input assignment is fixed. Using the properties of the private protocol and the properties of the special set of selected messages, we then prove, in the case of a general t-private protocol, that these vectors are linearly independent. In the case of a protocol of the restricted class, we prove that all the vectors obtained from sums of at most $t/2$ original vectors are linearly independent. In each case, this allows us to conclude that the number of different columns in the matrix obtained from the vectors as rows, is "large" (where the extent to which this number is large is different in each case). It follows that the number of transcripts of the protocol on the given input is "large", and hence using Lemma 2.6 the randomness complexity of the protocol is "high".

3.2. Proof of Theorem 1.1. In this section, as well as in the next section, we always assume that the computed function is `xor`. We now proceed to prove Theorem 1.1. In fact we prove here a stronger claim than the claim of Theorem 1.1. We prove that for any input assignment $\vec{\alpha} = \alpha_1, \ldots, \alpha_n$, there is a run in which the number of random bits read by the players from their random tapes is $\Omega(\log n)$.

3.2.1. Selecting the messages. Let $\vec{\alpha} = \alpha_1, \ldots, \alpha_n$ be an arbitrary input assignment. Given a fixed $\vec{\alpha}$, we will define a sequence of messages. (We will not indicate in our notation that the choice of this sequence depends on $\vec{\alpha}$, but this should be clear from the context.)

We define an ordering of all the messages sent during the protocol, in order to be able to refer to *the first* message with a given property. Then, based on this ordering, we select a sequence of messages with certain properties. The choice of these messages will induce a particular permutation of the input bits; and since each input bit belongs to a given player, this induces a permutation of the players as well.

DEFINITION 3.1. *We define an ordering of all messages, such that in this ordering, any message sent in round i precedes any message sent in round j, for $i < j$. For the messages sent within the same round we choose an arbitrary ordering.*

When we refer to the first message with a given property, we mean the first message according to the above ordering, that satisfies that property.

During the process of selecting the sequence of messages, we also assign a particular numbering to the input bits and to the players. To this end, when selecting a given message, a variable and a player are also selected, and both are given the same number as the message. That is, when the first i messages $m_1, \ldots, m_i$ have been selected, the variables $x_1, \ldots, x_i$ and the players $P_1, \ldots, P_i$ are also already selected. (We assume an arbitrary permutation of the remaining indices $i + 1, \ldots, n$ for the remaining variables and players.) When the process ends (after selecting n messages) a permutation of the variables and a permutation of the players is fixed.

We now proceed to the selection process. Let m_1 be the first message in the protocol that depends on at least one input variable. We will argue below that since this is the first such message, it can depend on only one input variable, and without loss of generality, we denote this input variable by x_1, and the player that has access to it by P_1.

Let m_2 be the first message in the protocol that depends on at least one input variable under $x_1 = \alpha_1$. We will argue below that since this is the first such message, there is only one such input variable, and without loss of generality we denote it by x_2, and the player that has access to it by P_2.

Inductively, let m_i be the first message sent in the protocol that depends on some input variable under $x_1 = \alpha_1, \ldots, x_{i-1} = \alpha_{i-1}$. We prove the following claim.

CLAIM 2. *Let x_k be any input variable on which m_i depends under $x_1 = \alpha_1, \ldots, x_{i-1} = \alpha_{i-1}$. Then the sender of the message m_i must be the player that has access to the variable x_k, that is, player P_k.*

Proof. Suppose that the sender of the message m_i is a player P_j such that $j \neq k$ (i.e., P_j is not the owner of x_k). Note that a message m sent in a given round by player P_j is a function of only the communication to player P_j in previous rounds, its input x_j and its random tape R_j. Thus, by Observation 1, if P_j is not the owner of the variable x_k, it can send a message that depends on x_k under $x_1 = \alpha_1, \ldots, x_{i-1} = \alpha_{i-1}$, only if it received in an earlier round a message that depends on x_k under $x_1 = \alpha_1, \ldots, x_{i-1} = \alpha_{i-1}$. But this contradicts the assumption that m_i is the first such message. $\square$

The above claim implies that there is only one input variable on which m_i depends under $x_1 = \alpha_1, \ldots, x_{i-1} = \alpha_{i-1}$. Without loss of generality we denote it by x_i, and the player that has access to it by P_i. Thus we derive the following.

CLAIM 3.

1. *The message m_i is sensitive to x_i on the input $\vec{\alpha}$ and some vector of random tapes $\vec{R}$.*
2. *Let $\vec{\beta}$ be any input that agrees with $\vec{\alpha}$ in the first $i-1$ coordinates, and let $j > i$. Then the message m_i is not sensitive to the variable x_j on $\vec{\beta}$ and $\vec{R}$, for any $\vec{R}$.*

Proof. We selected m_i such that m_i depends on x_i under $x_1 = \alpha_1, \ldots, x_{i-1} = \alpha_{i-1}$. This means that there is some assignment to the remaining variables and some $\vec{R}$ such that m_i is sensitive to x_i on the input obtained by the additional assignment and $\vec{R}$. But since x_i is the only input variable on which m_i depends under $x_1 = \alpha_1, \ldots, x_{i-1} = \alpha_{i-1}$ this also means that m_i is sensitive to x_i on $\vec{\alpha}$ and $\vec{R}$.

We obtain the second statement using the observation that x_i is the only input variable on which m_i depends under $x_1 = \alpha_1, \ldots, x_{i-1} = \alpha_{i-1}$. $\square$

CLAIM 4. *We can continue the above procedure of selecting messages for n steps, and define the sequence of messages $m_1, \ldots, m_n$.*

Proof. In a correct protocol to compute the function f, the output of each player has to be equal to $f(\vec{\alpha})$ on any input $\vec{\alpha}$. Note that the output of a given player depends only on the communication it received, its input bit and its random tape. Since the sensitivity $s(f, \vec{\alpha})$ of the function $f(\vec{x}) = x_1 + \ldots + x_n (\bmod 2)$ is n on every input $\vec{\alpha}$, we have by Observation 1, for each player P_i and each variable x_j such that $j \neq i$, that the communication received by P_i must contain at least one message that is sensitive to x_j on the input $\vec{\alpha}$ and some $\vec{R}$. Thus, on any input $\vec{\alpha}$, there exists at least one message for each variable x_j, that is sensitive to x_j on $\vec{\alpha}$ and some $\vec{R}$. If our procedure cannot be continued after $k < n$ steps on some input $\vec{\alpha}$, that would mean by Claim 3 that no message is sensitive to any of the remaining variables on $\vec{\alpha}$ and $\vec{R}$ for any $\vec{R}$, which would be a contradiction. $\square$

As argued above, the senders of the messages $m_1, \ldots, m_n$ are $P_1, \ldots, P_n$, respectively. Denote by $Q_1, \ldots, Q_n$ the receivers of these messages. Note that the n receivers are not necessarily n distinct players. We now select a subset of the above n messages, $m_{i_1}, \ldots, m_{i_q}$, with the property that $\{P_{i_j} : j \in [q]\} \cap \{Q_{i_j} : j \in [q]\} = \emptyset$. That is, none of the receivers of the selected messages is a sender of a selected message.

LEMMA 3.2. *There is a subset of size $q \geq \frac{n}{4}$ of the above n messages, denoted $m_{i_1}, \ldots, m_{i_q}$, such that the receivers of these messages, $Q_{i_1}, \ldots, Q_{i_q}$ are disjoint from the senders of these messages, $P_{i_1}, \ldots, P_{i_q}$.*

Proof. For the purpose of the proof we define an undirected graph. The set of nodes consists of n nodes, each node v_i representing a message m_i of the original set of n messages. Recall that each distinct message m_i is sent by a distinct player P_i. Therefore we can also think of the nodes as representing n distinct players. For each message m_i, we put an edge between node v_i and node v_j, if player P_j is the receiver of message m_i (i.e., if $Q_i = P_j$).

We now have a graph of n nodes and at most n edges. The graph therefore contains an independent set of size at least $\frac{n}{4}$ (cf. [1], Theorem 3.2.1). We select the messages that correspond to the nodes of this independent set. □

To simplify notation, in what follows we denote by $m_1, \ldots, m_q$, $P_1, \ldots, P_q$, $x_1, \ldots, x_q$, and $Q_1, \ldots, Q_q$, the selected messages, their senders, the input variables these senders have access to, and the receivers of the messages, respectively.

3.2.2. Properties of the vectors defined by the selected messages. We will now consider the $2^{n\ell}$-bit binary vectors that represent these messages on input $\vec{\alpha}$. We denote by $\vec{m}(\vec{\alpha})$ the binary vector of length $2^{n\ell}$ that consists of the bits $m(\vec{\alpha}, \vec{R})$. Thus, for any i, the vector $\vec{m}_i(\vec{\alpha})$ consists of the bits $m_i(\vec{\alpha}, \vec{R})$. For $\emptyset \neq S \subseteq [q]$ we denote by $\vec{m}_S(\vec{\alpha})$ the bitwise mod2 sum of the vectors $\vec{m}_i(\vec{\alpha})$, for $i \in S$. That is, $m_S(\vec{\alpha}, \vec{R}) = \oplus_{i \in S} m_i(\vec{\alpha}, \vec{R})$.

LEMMA 3.3. *Let $m_1, \ldots, m_q$ be selected as above in a t-private protocol. For any $\emptyset \neq S \subseteq [q]$ of size at most t, and any $i, j \in [q]$ (where $i \neq j$),*

$$Pr_{\vec{R}}[m_S(\vec{\alpha}, \vec{R}) = 1] = Pr_{\vec{R}}[m_S(\vec{\alpha}^{(i,j)}, \vec{R}) = 1] \ .$$

Proof. Consider the set of players $S' = \{Q_i | i \in S\}$, that is, the coalition formed by the receivers of the messages m_i, for $i \in S$. Then $\phi^{S'}(c_{S'}) = \oplus_{i \in S} m_i(\vec{\alpha}, \vec{R}) = m_S(\vec{\alpha}, \vec{R})$ is a function of the sequence of messages received by the players in S'. Recall that the set of senders of the messages m_i is disjoint from the set of the receivers, that is $\{P_i : i \in [q]\} \cap \{Q_i : i \in [q]\} = \emptyset$, which implies that $\{P_i : i \in [q]\} \cap S' = \emptyset$, and therefore for any $i, j \in [q]$, P_i and P_j are not in S'. This means that for any $i, j \in [q]$ and for any $l \in S'$, $\alpha_l = \alpha_l^{(i,j)}$, that is, the input bits held by the members of the coalition S' are not changed when one flips the i-th and j-th bits of $\vec{\alpha}$. Note also that flipping two bits does not change the value of the `xor` function, that is $f(\vec{\alpha}) = f(\vec{\alpha}^{(i,j)})$, for any $i, j \in [q]$, when f is the `xor` function. Thus, we can apply Lemma 2.4 to S' and $\phi^{S'} = m_S$, and the second statement of Lemma 2.4 directly implies the statement of the present lemma. □

For $i = 1, \ldots, q$, we denote by $\vec{h}_i(\vec{\alpha})$ the bitwise mod2 sum of the vectors $\vec{m}_i(\vec{\alpha})$ and $\vec{m}_i(\vec{\alpha}^{(i)})$. Thus, the vector $\vec{h}_i(\vec{\alpha})$ is 1 in the coordinates corresponding to $\vec{R}$ such that $m_i(\vec{\alpha}, \vec{R})$ and $m_i(\vec{\alpha}^{(i)}, \vec{R})$ differ, and 0 where they agree. We denote by $h_i(\vec{\alpha}, \vec{R})$ the entry of $\vec{h}_i(\vec{\alpha})$ in the coordinate corresponding to $\vec{R}$.

CLAIM 5. *The vectors $\vec{h}_i(\vec{\alpha})$, $i = 1, \ldots, q$, are not identically* 0.

Proof. Follows by the definition of the messages m_i and the first statement of Claim 3. □

LEMMA 3.4. *Let $m_1, \ldots, m_q$ be selected as above in a t-private protocol. Let $\emptyset \neq S \subseteq [q-1]$ be a subset of size at most t, and let k be the largest element of S. Then*

$$Pr_{\vec{R}}[m_S(\vec{\alpha}, \vec{R}) = 1 | h_k(\vec{\alpha}, \vec{R}) = 1] = 1/2 \ .$$

Proof. Since k is the largest element of S and q is larger than any element in S, we get by Claim 3 that $\vec{m}_S(\vec{\alpha}^{(k,q)})$ is the bitwise mod 2 sum of the vectors $\vec{m}_S(\vec{\alpha})$ and $\vec{h}_k(\vec{\alpha})$. To see this observe that for any $1 \leq i < k$, $m_i(\vec{\alpha}^{(k)}, \vec{R}) = m_i(\vec{\alpha}, \vec{R})$, by the second statement of Claim 3, and $m_i(\vec{\alpha}^{(k,q)}, \vec{R}) = m_i(\vec{\alpha}^{(k)}, \vec{R})$ again by the second statement of Claim 3. At the same time, $m_k(\vec{\alpha}^{(k,q)}, \vec{R}) = m_k(\vec{\alpha}^{(k)}, \vec{R})$ by the second statement of Claim 3, and $m_k(\vec{\alpha}^{(k)}, \vec{R}) = m_k(\vec{\alpha}, \vec{R}) + h_k(\vec{\alpha}, \vec{R})$ by the definition of h_k.

Thus, $m_S(\vec{\alpha}^{(k,q)}, \vec{R})$ and $m_S(\vec{\alpha}, \vec{R})$ are complements of each other in the coordinates where $h_k(\vec{\alpha}, \vec{R})$ is 1, and agree where $h_k(\vec{\alpha}, \vec{R})$ is 0. We therefore have

$$\begin{aligned} Pr_{\vec{R}}[m_S(\vec{\alpha}, \vec{R}) = 1] = Pr_{\vec{R}}[m_S(\vec{\alpha}, \vec{R}) = 1 \wedge h_k(\vec{\alpha}, \vec{R}) = 1] \\ + Pr_{\vec{R}}[m_S(\vec{\alpha}, \vec{R}) = 1 \wedge h_k(\vec{\alpha}, \vec{R}) = 0] \ , \end{aligned}$$

and,

$$\begin{aligned} Pr_{\vec{R}}[m_S(\vec{\alpha}^{(k,q)}, \vec{R}) = 1] = Pr_{\vec{R}}[m_S(\vec{\alpha}, \vec{R}) = 0 \wedge h_k(\vec{\alpha}, \vec{R}) = 1] \\ + Pr_{\vec{R}}[m_S(\vec{\alpha}, \vec{R}) = 1 \wedge h_k(\vec{\alpha}, \vec{R}) = 0] \ . \end{aligned}$$

Since by Lemma 3.3

$$Pr_{\vec{R}}[m_S(\vec{\alpha}, \vec{R}) = 1] = Pr_{\vec{R}}[m_S(\vec{\alpha}^{(k,q)}, \vec{R}) = 1] \ ,$$

we have that

$$Pr_{\vec{R}}[m_S(\vec{\alpha}, \vec{R}) = 1 \ \wedge \ h_k(\vec{\alpha}, \vec{R}) = 1] = Pr_{\vec{R}}[m_S(\vec{\alpha}, \vec{R}) = 0 \ \wedge \ h_k(\vec{\alpha}, \vec{R}) = 1] \ .$$

Since $Pr_{\vec{R}}[h_k(\vec{\alpha}, \vec{R}) = 1] \neq 0$ by Claim 5, this implies the statement of the lemma. □

For $i = 1, \ldots, q$ we denote by $\vec{\omega}_i(\vec{\alpha})$ the binary vector of length $2^{n\ell}$ that we get by replacing each 0 in $\vec{m}_i(\vec{\alpha})$ by 1, and replacing each 1 in $\vec{m}_i(\vec{\alpha})$ by -1. Similarly, for $\emptyset \neq S \subseteq [q]$ we denote by $\vec{\omega}_S(\vec{\alpha})$ the binary vector of length $2^{n\ell}$ that we get by replacing each 0 in $\vec{m}_S(\vec{\alpha})$ by 1, and replacing each 1 in $\vec{m}_S(\vec{\alpha})$ by -1. That is, we move from the domain $\{0,1\}$ to the domain $\{1,-1\}$, and obtain the vectors $\vec{\omega}_i(\vec{\alpha})$, for $i = 1, \ldots, q$, from the vectors $\vec{m}_i(\vec{\alpha})$ using the standard transformation that replaces each value b by $(-1)^b$. The vectors $\vec{\omega}_S(\vec{\alpha})$, for $\emptyset \neq S \subseteq [q]$, are obtained from the vectors $\vec{m}_S(\vec{\alpha})$ in the same way.

Lemma 3.5. *Let $\vec{\omega}_1, \ldots, \vec{\omega}_q$ be selected as above in a t-private protocol, for $t \geq 2$. Then, the vectors $\vec{\omega}_i(\vec{\alpha})$, $i = 1, \ldots, q-1$ are linearly independent over the reals.*

Proof. As we will see in the next section, our job would be much easier (and we could obtain stronger bounds) if the vectors $\vec{h}_k(\vec{\alpha})$ were the same for each k. In that case we could show that the vectors $\vec{\omega}_i(\vec{\alpha})$, $i = 1, \ldots, q-1$ (or some projections of them) are pairwise orthogonal. However, in general the vectors $\vec{h}_k(\vec{\alpha})$ may not be the same. Nevertheless, we can show that a given projection of each vector $\vec{\omega}_k(\vec{\alpha})$ is orthogonal to the same projection of each preceding vector $\vec{\omega}_i(\vec{\alpha})$ for $i < k$. This will let us show that for any k such that $2 \leq k \leq q-1$, the vector $\vec{\omega}_k(\vec{\alpha})$ cannot be obtained as a linear combination of the vectors $\vec{\omega}_1(\vec{\alpha}), \ldots, \vec{\omega}_{k-1}(\vec{\alpha})$.

Consider an arbitrary $k \in \{2, \ldots, q-1\}$, and consider the following projection of the vectors $\vec{\omega}_1(\vec{\alpha}), \ldots, \vec{\omega}_k(\vec{\alpha})$. Note that our choice of the projection depends on k (via $\vec{h}_k(\vec{\alpha})$) and this is indicated in the notation by the superscript k. For $i = 1, \ldots, k$, denote by $\vec{v}_i^k$ the projection of the vector $\vec{\omega}_i(\vec{\alpha})$ to only those coordinates where $h_k(\vec{\alpha}, \vec{R}) = 1$. Note that $\vec{h}_k(\vec{\alpha})$ is not identically 0, as stated in Claim 5, so there is always at least one such coordinate.

Since $t \geq 2$, by applying Lemma 3.4 to the sets $\{i, k\}$ for $i < k$ we get that the inner product of $\vec{v}_k^k$ with any of the vectors $\vec{v}_i^k$ for $i < k$ is 0. To see this, consider $\vec{\omega}_S(\vec{\alpha})$ for $S = \{i, k\}$. Considering $\{1,-1\}$ vectors instead of $\{0,1\}$ vectors, Lemma 3.4 states that

$$Pr_{\vec{R}}[\omega_S(\vec{\alpha}, \vec{R}) = 1 | h_k(\vec{\alpha}, \vec{R}) = 1] = Pr_{\vec{R}}[\omega_S(\vec{\alpha}, \vec{R}) = -1 | h_k(\vec{\alpha}, \vec{R}) = 1] = 1/2 \ .$$

Thus, $\sum_{\{\vec{R}:h_k(\vec{\alpha},\vec{R})=1\}} \omega_S(\vec{\alpha},\vec{R}) = 0$. Notice that for $S = \{i,k\}$, the above sum is exactly the inner product of the vectors $\vec{v}_i^k$ and $\vec{v}_k^k$, since $\omega_S(\vec{\alpha},\vec{R}) = \omega_i(\vec{\alpha},\vec{R})\cdot\omega_k(\vec{\alpha},\vec{R})$. Therefore, we get that the inner product of $\vec{v}_k^k$ with any of the vectors $\vec{v}_i^k$ for $i < k$ is 0, as claimed.

Suppose that $\vec{\omega}_k(\vec{\alpha})$ can be obtained as a linear combination of the vectors $\vec{\omega}_1(\vec{\alpha}),\ldots,\vec{\omega}_{k-1}(\vec{\alpha})$. Then $\vec{v}_k^k$ can be obtained as a linear combination of the vectors $\vec{v}_1^k,\ldots,\vec{v}_{k-1}^k$. But since the inner product of $\vec{v}_k^k$ with each $\vec{v}_i^k$ for $i < k$ is 0, this would imply that the inner product of $\vec{v}_k^k$ with itself is 0. Since $\vec{v}_k^k$ has only 1 or -1 entries, this is not possible. □

Let us now consider the $(q-1)\times 2^{n\ell}$ matrix, formed by the vectors $\vec{\omega}_i(\vec{\alpha})$, $i = 1,\ldots,q-1$ as row vectors. Since the vectors $\vec{\omega}_i(\vec{\alpha})$, $i = 1,\ldots,q-1$ are linearly independent, this matrix has at least $q-1$ different columns. This implies that the protocol has at least $q-1$ different transcripts on input $\vec{\alpha}$. Since $q = \Omega(n)$, Theorem 1.1 follows by Lemma 2.6.

4. Restricted Protocols.

In this section we consider a class of restricted protocols that we define below. Our motivation to consider this class is that all known randomness-efficient protocols designed specifically for xor obey this restriction, or can be easily brought to this form without changing the number of coin tosses performed [28, 23, 27, 8]. Informally one can describe the protocols of this class in the following way. First, a deterministic, non-private, protocol to compute f is defined. Then this protocol is modified by masking each message with randomness by adding to it (modulo 2) a value that depends on the randomness only. This restriction was previously considered in [28]. Formally the restriction we consider here is defined as follows:

Definition 4.1. *We say that a given protocol involving n players has a restricted form if each message of the protocol can be obtained as a mod 2 sum of a Boolean function $u : \{0,1\}^n \to \{0,1\}$ that depends on the input variables only, and a Boolean function $v : \{0,1\}^{n\ell} \to \{0,1\}$ that depends on the random tapes only. That is, each message m can be written as $m(\vec{x},\vec{R}) = u(\vec{x}) + v(\vec{R}) \bmod 2$.*

Theorem 4.2. *Let $t \geq 2$, and let $\mathcal{A}$ be a d-random, t-private, protocol, obeying the above restriction, for computing $f(\vec{x}) = x_1 + \ldots + x_n(\text{mod}2)$. Then $d = \Omega(t\log(n/t))$.*

As in our proof for general protocols, here too we prove in fact a stronger claim. We prove that for any input assignment $\vec{\alpha} = \alpha_1,\ldots,\alpha_n$, there is a run of the protocol in which the number of random bits read by the players from their random tapes is $\Omega(t\log(n/t))$.

Proof. Let $\vec{\alpha} = \alpha_1,\ldots,\alpha_n$ be an arbitrary input assignment. We select a sequence of messages $m_1,\ldots,m_q$ and define the corresponding vectors as in the previous section. Recall that $\vec{h}_i(\vec{\alpha})$ denotes the bitwise mod2 sum of the vectors $\vec{m}_i(\vec{\alpha})$ and $\vec{m}_i(\vec{\alpha}^{(i)})$. The restriction on the protocols we consider allows us to have the following claim.

Claim 6. *The vectors $\vec{h}_i(\vec{\alpha})$, $i = 1,\ldots,q$ are identically* 1.

Proof. We know by Claim 5 that the vectors $\vec{h}_i(\vec{\alpha})$ are not identically 0. That is, they have at least one entry with value 1. This means that $m_i(\vec{\alpha},\vec{R}) \neq m_i(\vec{\alpha}^{(i)},\vec{R})$ for at least one $\vec{R}$. But $m_i(\vec{x},\vec{R})$ can be written as $m_i(\vec{x},\vec{R}) = u_i(\vec{x}) + v_i(\vec{R}) \bmod 2$. It follows that $u_i(\vec{\alpha}) \neq u_i(\vec{\alpha}^{(i)})$, and therefore for any $\vec{R}$, $m_i(\vec{\alpha},\vec{R}) \neq m_i(\vec{\alpha}^{(i)},\vec{R})$. □

The above claim allows us to obtain a stronger bound using the machinery of the previous section. We now prove the following lemma.

LEMMA 4.3. *Let $\vec{\omega}_S$ be defined as above in a t-private protocol obeying the above restriction, for $t \geq 2$. Then, the vectors $\vec{\omega}_S(\vec{\alpha})$, such that $\emptyset \neq S \subseteq [q-1]$ and $|S| \leq \lfloor t/2 \rfloor$ are linearly independent over the reals.*

Proof. First note that since $t \geq 2$, there exist sets S, such that $\emptyset \neq S \subseteq [q-1]$ and $|S| \leq \lfloor t/2 \rfloor$, thus the statement of the lemma is meaningful (assuming $n \geq 5$; otherwise $q-1$ could be less than 1, since the guarantee of Lemma 3.2 is that $q \geq \frac{n}{4}$). Since by Claim 6 for each $i \in [q]$ the vector $\vec{h}_i(\vec{\alpha})$ is identically 1, Lemma 3.4 gives that $Pr_{\vec{R}}[m_T(\vec{\alpha}, \vec{R}) = 1] = 1/2$, for any $\emptyset \neq T \subseteq [q-1]$ of size at most t. This implies that the sum of entries of the vector $\vec{\omega}_T(\vec{\alpha})$ is 0, for any $\emptyset \neq T \subseteq [q-1]$ of size at most t. Notice that for any two sets S_1 and S_2, we have $\omega_{S_1}(\vec{\alpha}, \vec{R}) \cdot \omega_{S_2}(\vec{\alpha}, \vec{R}) = \omega_{S_1 \triangle S_2}(\vec{\alpha}, \vec{R})$. Thus, for sets $\emptyset \neq S_1 \subseteq [q-1]$ and $\emptyset \neq S_2 \subseteq [q-1]$, each of size at most $\lfloor t/2 \rfloor$, the inner product of $\vec{\omega}_{S_1}(\vec{\alpha})$ and $\vec{\omega}_{S_2}(\vec{\alpha})$ must be 0. We get that the vectors $\vec{\omega}_S(\vec{\alpha})$, for $\emptyset \neq S \subseteq [q-1]$ and $|S| \leq \lfloor t/2 \rfloor$ are pairwise orthogonal, and therefore they must be linearly independent over the reals. □

We denote by $\binom{a}{\leq b}$ the sum $\sum_{i=1}^{\min(a,b)} \binom{a}{i}$, for integers $a, b \geq 1$. Let us now consider the $\binom{q-1}{\leq \lfloor t/2 \rfloor} \times 2^{n\ell}$ matrix, formed by the vectors $\vec{\omega}_S(\vec{\alpha})$, such that $\emptyset \neq S \subseteq [q-1]$ and $|S| \leq \lfloor t/2 \rfloor$, as row vectors. Since the vectors $\vec{\omega}_S(\vec{\alpha})$ are linearly independent, this matrix has at least $\binom{q-1}{\leq \lfloor t/2 \rfloor}$ different columns. Note that each column of the matrix is associated with a fixed vector of random tapes $\vec{R}$, and each column is completely determined by the transcript of the protocol on the given $\vec{R}$ and $\vec{\alpha}$. This implies that the protocol has at least $\binom{q-1}{\leq \lfloor t/2 \rfloor}$ different transcripts on input $\vec{\alpha}$. Since $q = \Omega(n)$, the theorem follows by Lemma 2.6. □

We find it worthwhile to note that our proof also implies that the random variables associated with the messages $m_1, \ldots, m_q$ on input $\vec{\alpha}$ are t-wise independent in any t-private protocol that obeys the above restriction. Thus, we could conclude the proof of Theorem 4.2 by referring to the known lower bounds on the size of sample spaces with t-wise independent random variables [12, 1]. In fact, part of the proof of Lemma 4.3 is analogous to the corresponding part of the argument used in [12]. The t-wise independence property of the random variables associated with the messages $m_1, \ldots, m_q$ on input $\vec{\alpha}$ follows by Lemma 3.4, which, as we have shown above, gives in the case of the restricted protocols that $Pr_{\vec{R}}[m_S(\vec{\alpha}, \vec{R}) = 1] = 1/2$, for any $\emptyset \neq S \subseteq [q-1]$ of size at most t. This implies t-wise independence of the corresponding random variables by the results of [12].

REFERENCES

[1] N. Alon, J. H. Spencer, *The Probabilistic Method* (second edition), John Willey & Sons, 2000.

[2] 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.

[3] D. Beaver, “Perfect Privacy for Two-Party Protocols”, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 2, 1991, pp. 65-77.

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

[5] M. Bläser, A. Jakoby, M. Liskiewicz, B. Siebert, “Private Computation - k-connected vs. 1-connected networks” Proc. of 22nd CRYPTO, 2002, pp. 194-209.

[6] C. Blundo, A. De Santis, G. Persiano, and U. Vaccaro, “On the Number of Random Bits in Totally Private Computations”, Proc. of 22nd ICALP, 1995, pp. 171-182.

[7] C. Blundo, A. De Santis, G. Persiano, U. Vaccaro, “Randomness Complexity of Private Computation”. Computational Complexity, Vol. 8, pp. 145–168, 1999.

[8] C. Blundo, C. Galdi, P. Persiano, "Randomness Recycling in Constant-Round Private Computations". DISC 1999. pp. 138–150.

[9] C. Blundo, C. Galdi, P. Persiano, "Low-Randomness Constant-Round Private Computations". Manuscript, 2003.

[10] R. Canetti, E. Kushilevitz, R. Ostrovsky, and A. Rosén, "Randomness vs. Fault-Tolerance", *Journal of Cryptology*, Vol. 13, pp. 107–142, 2000.

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

[12] B. Chor, J. Friedmann, O. Goldreich, J. Hastad, S. Rudich, and R. Smolensky, "The bit extraction problem and t-resilient functions" in Proc. of 26th FOCS, 1985, pp. 396-407.

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

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

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

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

[17] A. Gál, and A. Rosén, "A Theorem on Sensitivity and Applications in Private Computation". SIAM Jour. on Computing, Vol. 31, No. 5, pp. 1424–1437, 2002.

[18] O. Goldreich, "Modern Cryptography, Probabilistic Proofs and Pseudorandomness", Springer Verlag, Algorithms and Combinatorics, Vol 17, 1998.

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

[20] A. Jakoby, M. Liskiewicz, R. Reischuk, "Private Computations in Networks: Topology vs. Randomness", Proc. of 21st STACS, 2003, pp. 121-132.

[21] J. Kilian, E. Kushilevitz, S. Micali, and R. Ostrovsky, "Reducibility and Completeness in Private Computations", SIAM Jour. on Computing, Vol. 28, No. 4, pp. 1189-1208, 2000.

[22] S. C. Kleene, "Mathematical Logic", Dover, New York, 2002.

[23] E. Kushilevitz, and Y. Mansour, "Randomness in Private Computations", SIAM Jour. on Disc. Math., Vol. 10, No. 4 , pp. 647-661, 1997.

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

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

[26] E. Kushilevitz, R. Ostrovsky, and A. Rosén, "Characterizing Linear Size Circuits in Terms of Privacy", Proc. of STOC 96, and JCSS, Vol. 58 pp. 129–136, 1999.

[27] E. Kushilevitz, R. Ostrovsky, and A. Rosén, "Amortizing Randomness in Private Multiparty Computations". PODC 1998, pp. 81–90.

[28] E. Kushilevitz, and A. Rosén, "A Randomness-Rounds Tradeoff in Private Computation", *SIAM Journal on Discrete Mathematics*, Vol. 27, pp. 1531–1549, 1998.

[29] N. Nisan, A. Ta-Shma, "Extracting Randomness: A Survey and New Constructions" JCSS, Vol.58, No. 1, Feb. 1999, pp. 148-173.

[30] A. Yao, "Protocols for Secure Computation", Proc. of 23rd FOCS, 1982, pp. 160–164.