Randomness in Private Computations

Eyal Kushilevitz* Yishay Mansour†

Abstract

We consider the amount of randomness used in *private* distributed computations. Specifically, we show how n players can compute the exclusive-or (xor) of n boolean inputs t-privately, using only $O(t^2 \log(n/t))$ random bits (the best known upper bound is $O(tn)$). We accompany this result by a lower bound on the number of random bits required to carry out this task; we show that any protocol solving this problem requires at least t random bits (again, this significantly improves over the known lower bounds).

For the upper bound, we show how, given m subsets of $\{1, \ldots, n\}$, to construct in (deterministic) polynomial time a probability distribution of n random variables such that (1) the parity of random-variables in each of these m subsets is 0 or 1 with equal probability; and (2) the support of the distribution is of size at most $2m$. This construction generalizes previously considered types of sample spaces (such as k-wise independent spaces and Schulman's spaces [S92]). We believe that this construction is of independent interest and may have various applications.

1 Introduction

There has been a great effort devoted to the study of *randomization*. Initially, the main application of randomization was for solving problems to which deterministic solutions are impossible (e.g., in distributed computing and in cryptography) or unknown (e.g., efficient primality testing). Furthermore, randomization is used to construct both more efficient and, not less significant, much simpler algorithms.

Randomness as a resource was extensively studied in the last decade. One line of research was devoted to a quantitative study of the role of randomness in specific contexts, e.g., [RS89, KPU88, BGG90, CG90, BGS94, BSV94, KR94]; another direction was developing general-purpose methods for saving random bits. These methods range over pseudo-random generators [Y82, BM84, N90], techniques for re-cycling random bits [IZ89, CW89], sources of weak randomness [CG85, VV85, Z91], and construction of various kinds of small probability spaces [NN90, AGHP90, S92, KM93, KM94b, KK94]. A particular goal was to allow *derandomization*; i.e., to completely eliminate the use of randomness. For some problems, the best known deterministic algorithms are randomized algorithm which are latter derandomized (see, e.g., [KM94a]).

We consider the role of randomness in *t-private* protocols. Informally, a t-private protocol $\mathcal{P}$ for computing a function f is a protocol that allows n players, P_i $(1 \leq i \leq n)$, each holding an individual secret input, x_i, to compute the value of $f(\vec{x})$ in a way that no coalition of at most t players learns about the initial inputs of other players more than what is revealed by the value of $f(\vec{x})$ and their own inputs. The players are assumed to be honest but curious; namely, they all follow the prescribed protocol $\mathcal{P}$ but they could try to get additional information from the messages they receive during the execution of the protocol. The study of private computations in this setting was initiated by [BGW88, CCD88] and was the subject of a considerable amount of work, e.g., [BB89, CK89, K89, B89, FY92, CK92, CGK90, CGK92, KMO94, KOR96].[1] The use of *randomness* is a crucial ingredient in private protocols; without randomness only degenerate functions can be computed privately.

*Department of Computer Science, Technion, Israel. This research was supported in part by MANLAM Fund 120-865. (e-mail: eyalk@cs.technion.ac.il. http:/cs.technion.ac.il/~eyalk)

†Dept. of Computer Science, Tel-Aviv University. This research was supported in part by The Israel Science Foundation administered by The Israel Academy of Science and Humanities and by a grant of the Israeli Ministry of Science and Technology. (e-mail: mansour@cs.tau.ac.il. http://moon.math.tau.ac.il/~mansour)

[1] This setting is different than the one studied in [Y86, GMW87] in which the computational power of the players is restricted and hence intractability assumptions can be used.

The *xor* (*exclusive-or*) function (and more generally, modular sum function) is a basic building block in all private-protocols. As a result (and due to its relative simplicity), the task of computing *xor* t-privately was the subject of previous research [FY92, CK92, KR94, BDPV95]. We investigate the amount of randomness required to compute the exclusive-or of n input bits t-privately. The known upper bound uses $O(tn)$ random bits. Better upper bounds were known only for the case $t = 1$ (see [KR94]), in which a single random bit is sufficient. As for lower bounds, Blundo et. al. [BDPV95] proved that if $t \geq n - c$ (for some constant c) then $\Omega(n^2)$ random bits are required and if $t \geq (2 - \sqrt{2})n$ then $\Omega(n)$ bits are required. For smaller values of t it was only known that no deterministic protocol for this task exists.

We significantly improve both the upper bound and the lower bound for this problem. We present a protocol for this task that uses only $O(t^2 \log(n/t))$ random bits. On the other hand, we prove that any t-private protocol for *xor* requires at least t random bits. This is the best known lower bound for most values of t (i.e., excluding the case where t is very close to n).

For our upper bound, we develop a new construction of small sample spaces that naturally generalizes k-wise independent sample spaces, and sample spaces of the type studied by Schulman [S92].[2] More precisely, given subsets $T_1, \ldots, T_m \subseteq \{1, \ldots, n\}$ we look for a small sample space in which the parity of the variables in every T_j gets the values 0 and 1 with equal probability, i.e. 1/2. We present a deterministic polynomial-time algorithm that allows constructing such a space for *any* collection of such subsets (in particular, it is important for our application that there is no restriction on the size of the sets). This is a *uniform* space[3] whose size is linear in m ($2m$ vectors). We also show how to find such spaces in parallel (in NC).

Relation to Other Work

The abovementioned result of [S92] was generalized in various ways [KM93, KM94b, KK94]. In particular, Karger and Koller [KK94], presented a construction that can handle parity requirements as in our work. However, there is a significant difference between their construction (as well as those in [KM93, KM94b]) and the construction presented in this paper: our construction builds a *uniform* sample space while the constructions of [KM93, KM94b, KK94] build a *non-uniform* sample spaces (i.e., each $s \in S$ is selected with a different probability p_s). While non-uniformity does not matter if the goal is derandomization, it has many disadvantages if we have other applications in mind. In particular, if we want to *sample* in the space then non-uniform sample spaces may not be useful for saving random bits (the goal of the current paper): for sampling in a uniform space $\lceil \log_2 |S| \rceil$ random bits are enough, while for sampling in a non-uniform space we may need much more, and in some cases will not be able to create exactly the same distribution (see [KY76]). Sampling is needed if, for example, we do not want to pay the $|S|$ penalty in the running-time involved with derandomizing an algorithm but just want to run the algorithm once. In such a case a non-uniform space does not necessarily reduce the number of random bits required.

In addition, as mentioned, randomness is not only used for achieving efficient algorithms. For example, in cryptographic protocols randomness is used to maintain the secrecy of information. In such applications, we cannot completely eliminate the use of randomness while maintaining the secrecy so we are just interested in reducing the number of random bits used. For such applications, other constructions such as ϵ-bias (uniform) sample spaces [NN90, AGHP90] are not useful, as they only guarantee that parities are "almost" balanced. In cryptographic settings this could mean information leakage.

Organization

In Section 2 we present the construction of small sample spaces required for our results (its parallel version appears in Appendix A and other extensions appear in Appendix B). In Section 3 we study the question of randomness in private computations. The upper bound, that builds on the construction of small sample spaces, appears in Section 3.2, while the lower bound appears in Section 3.3. (Section 3.1 includes some required definitions.)

[2] Schulman [S92] initiated the study of sample spaces that satisfy a list of specific independence constraints. He showed that if we are interested in having n random variables such that only m subsets $H_1, \ldots, H_m$ of them, each of size at most k, will behave uniformly, then one can get a sample space of size $O(m2^k)$ (in particular, if $k = O(\log n)$ then the sample space is of polynomial size).

[3] I.e., a space that consists of a set S and the uniform distribution over S.

2 Constructing Small Spaces Immune to Parity Tests

2.1 Preliminaries

Let $S \subseteq \{0,1\}^n$ be a collection of (not necessarily distinct) binary vectors of length n. We denote by $s \in_R S$ a choice of an element of S uniformly at random. The distribution generated by S is the distribution induced by picking $s \in_R S$.
We say that a set S is *immune* to a parity-test $T \subseteq \{1,\ldots,n\}$ if,

$$\Pr_{s \in_R S}[\oplus_{i \in T} s_i = 1] = \frac{1}{2}.$$

Informally, this means that when considering the distribution generated by S, the parity of variables in T is unbiased, i.e. the probability that the parity is 0 (or 1) is exactly 1/2. Let $T_1,\ldots,T_m$ be m (non-empty) subsets of $\{1,2,\ldots,n\}$. A set S is immune to $T_1,\ldots,T_m$ if it is immune to each T_i.

In the sequel whenever there are operations between elements of $\{0,1\}^n$ they are done in $GF(2^n)$. For example, $x+y$ is simply the bitwise exclusive or of x and y; the inner product is defined by $\langle x,y\rangle = \sum_{i=1}^{n} x_i y_i \bmod 2$. The following definition and lemma play a central role in our constructions:

Definition 1: Let M be an $n \times \ell$ 0-1 matrix. We define the following multi-set

$$space(M) = \left\{ M \cdot \vec{v}^T \,|\, \vec{v} \in \{0,1\}^\ell \right\}.$$

That is, $space(M)$ contains 2^ℓ (not necessarily distinct) vectors/strings where each is n-bit long. The matrix M is referred to as the *generating matrix* of $space(M)$. We denote the rows of this matrix by $M_1,\ldots,M_n$ and its columns by $M^1,\ldots,M^\ell$.

Lemma 1: Consider a set T_i, and let M be a matrix as above such that $\sum_{j \in T_i} M_j \neq \vec{0}$. Then, $space(M)$ is immune to the set T_i.

Proof: We need to show that the probability of the parity of the bits in T_i to be zero is exactly 1/2. Note that the parity of the bits of the vector $M\vec{v}^T$ whose indices are in T_i is simply $1_{T_i} M \vec{v}^T$, where 1_{T_i} stands for the characteristic vector of the set T_i. Since $\sum_{j \in T_i} M_j \neq \vec{0}$, then $1_{T_i} M = \vec{u}$, where $\vec{u} \neq \vec{0}$. The probability that the parity of the bits in T_i is zero is the probability that $\langle \vec{u}, \vec{v}\rangle = 0$, where $\vec{v} \in \{0,1\}^\ell$. Since $\vec{u} \neq \vec{0}$, this probability is exactly 1/2. □

Using this lemma, we reduce the problem of constructing a space which is immune to $T_1,\ldots,T_m$ to the problem of constructing a matrix M such that the rows corresponding to each T_i do not sum up to $\vec{0}$. We call such a matrix M *good*. All our constructions make use of this observation; i.e., their aim is to find a good matrix M.

2.2 Randomized Construction

Here we present a randomized construction of immune spaces. While a randomized construction is not very useful in the applications, this construction exhibits the possibility of finding such spaces.

Let $\ell = \lceil \log m \rceil + k$, for some parameter k. The construction is simply to select a random $0-1$ matrix M of size $n \times \ell$. Note that the construction depends on the number of sets, m, but it does *not* depend on the specific sets. Also note that $space(M)$ is of size at most $2 \cdot m \cdot 2^k$ and that to verify that the randomly chosen matrix M is good takes polynomial-time (in n and m).

Lemma 2: Let $T_1,\ldots,T_m$ be a collection of sets, and let M be constructed as above. With probability at least $1 - \frac{1}{2^k}$ $space(M)$ is immune to all sets $T_1,\ldots,T_m$.

Proof: For convenience, we view the construction as selecting at random, one-by-one, the n rows of the matrix $M_1,\ldots,M_n \in \{0,1\}^\ell$. Fix a set $T_i \subseteq \{1,\ldots,n\}$ and denote by t the maximal element in T_i. By Lemma 1, it is enough that

$$\sum_{j \in T_i} M_j \neq \vec{0}.$$

Consider, the t-th step in the construction of M, when $M_1,\ldots,M_{t-1}$ were already fixed. The only row in the above sum that still should be chosen is M_t and there is exactly *one* choice, among the 2^ℓ possibilities, that will make this sum equal $\vec{0}$. Hence, the probability of failure for T_i is $\frac{1}{2^\ell} \leq \frac{1}{m2^k}$. Therefore, the probability that there exists a set T_i for which we fail is at most m times larger, i.e. at most $1/2^k$. □

2.3 Deterministic Construction

In this section we describe a *deterministic* construction. The idea is similar to what we did in the randomized case, but instead of choosing the rows at random we will construct them deterministically entry-by-entry. This time the construction does look

at the specific sets $T_1, \ldots, T_m$ in question. Let $\ell = \lceil \log(m+1) \rceil$.

- For $k = 1, \ldots, n$ construct the k-th row of M as follows:
  We say that a set T_i is *relevant* for the k-th step if k is the maximal element of T_i. The goal in the k-th step is to make sure that each of the relevant sets has the property that $\sum_{j \in T_i} M_j \neq \vec{0}$ (the fact that the set is relevant means that k is the only row in the sum that was not fixed yet). For each of these sets there is exactly one value (in $\{0,1\}^\ell$) for the row that violates this property (i.e., with this value $\sum_{j \in T_i} M_j = \vec{0}$). This implies that there are at most m illegal choices for the k-th row. However, there are $2^\ell \geq m+1$ possible values for this row hence at least one of them satisfies the property with respect to all relevant sets.

Lemma 3: The above algorithm constructs in time $poly(n,m)$ a matrix M such that $space(M)$ is immune to all sets $T_1, \ldots, T_m$.

Proof: The correctness follows from the above discussion (particularly, the choice of ℓ and Lemma 1). The time complexity is obvious. □

In the appendices we discuss various extensions of our construction and its relationship to previous constructions.

3 Privacy

In this section we consider the problem of computing *xor* t-privately. We prove bounds on the (total) number of random bits used by the n players in order to perform this task. We first present the upper bound (Section 3.2) for which we use the results of the previous section. Then (in Section 3.3) we give a lower bound for this problem. We start with some required definitions.

3.1 Preliminaries

A set of n players, $P_1, \ldots, P_n$, each possessing a single bit x_i (known only to P_i), collaborate in a protocol to compute *xor* (i.e., $\oplus(x_1, \ldots, x_n)$). Each player may toss coins during the protocol. We formalize this as follows: the player P_i holds an infinite tape R_i of random bits. The number of random bits he uses is the position of the rightmost cell read by the player on his tape. The number of random bits used by the protocol is the total number of bits used by all players (note that in different executions P_i may use different number of bits). The randomness complexity is the worst case (over all inputs and all executions) number of random bits.

Next, we define the notion of privacy (we follow, e.g., [BGW88, CK89]). For a set of players T (sometimes called *a coalition*), denote by C_T the communication seen by the players in T (i.e., all message they receive during the execution of the protocol). A protocol is said to be *t-private* if every coalition T of size at most t sees the same distribution of communication on inputs that look the same for players in T. Formally, for every two inputs $\vec{x}$ and $\vec{y}$ such that $xor(\vec{x}) = xor(\vec{y})$ and $x_i = y_i$ for all $i \in T$, for every sequence of messages C, and for every choice of random tapes R_i for players in T, the protocol satisfies

$$\Pr[C_T = C | \{R_i\}_{i \in T}, \vec{x}] \quad = \quad \Pr[C_T = C | \{R_i\}_{i \in T}, \vec{y}],$$

where the probability ranges over the random tapes of the players not in T.

To simplify the presentation, we also consider a non-standard model in which, in addition to the n players $P_1, \ldots, P_n$, there is some trusted party Q. This party has a multi-set of vectors S which are just strings in Σ^n for some set Σ. The trusted party participates in the protocol in a very restricted way: Q first tosses some coins and use their outcome to pick a random string from S; he then sends to P_i the i-th coordinate of that vector (the P_i's have no other source of randomness); the players $P_1, \ldots, P_n$ proceed without receiving any more messages from Q.[4] We assume that each vector in S has a positive probability to be picked, and we call the set S the *support* of Q.

The notion of t-privacy is defined in a way similar to the standard model, but here the definition is somewhat simpler since we assume that Q does not participate in any coalition (he is a trusted party) and also all other players do not toss coins. That is, for every $T \subseteq \{P_1, \ldots, P_n\}$, every $\vec{x}, \vec{y}$ as above and every sequence of messages C, the protocol satisfies

$$\Pr[C_T = C | \vec{x}] \quad = \quad \Pr[C_T = C | \vec{y}],$$

where the probability ranges over the random choices made by Q (and C_T includes the random bits received from Q by players in T).

[4] If we allow the trusted party Q to be active he can collect the inputs of all players and compute the answer. However, we will not be able to transform such a solution to the standard model (with no trusted party).

3.2 Upper Bound

In this section we present a t-private protocol which computes *xor* while using small amount of random bits. We first show how, in the trusted-party model, t-private computation of *xor* can be performed using only $O(t\log(n/t))$ random bits. Then, we modify the protocol to work in the standard model (where no such trusted party exists) with a penalty of $O(t)$ in the randomness complexity. Namely, we use $O(t^2\log(n/t))$ random bits (compared to the previously known upper bound which is $O(tn)$). We start with the protocol in the trusted-party model. We first assume that the trusted party Q uses random bits which are uniformly distributed and completely independent. We will analyze this protocol and as a result note that for the proof to go through much weaker requirements regarding the random bits are needed. These requirements are of the type satisfied by the sample spaces constructed above and hence Q will be able to choose its random bits from such a space. The protocol goes as follows:

1. Q chooses at random $n-1$ random bits $z_1,\ldots,z_{n-1}$. He sends z_i to P_i $(1\leq i\leq n-1)$. In addition, he sends $z_n=\sum_{i=1}^{n-1} z_i$ to P_n.[5]

2. In round i $(i=1,\ldots,n-1)$, player P_i sums up the message (bit) m_{i-1} he received from P_{i-1} in round $i-1$, the random bit z_i received from Q, and its input x_i,[6] and it sends the result, m_i, to P_{i+1}. Similarly, in the n-th round, P_n computes $m_n=m_{n-1}+x_n+z_n$ and announces m_n as the output.

A simple induction shows that, for $1\leq i\leq n-1$, the message m_i sent by P_i satisfies:

$$m_i=x_i+z_i+m_{i-1}=\sum_{j=1}^{i}x_j+\sum_{j=1}^{i}z_j.$$

This, together with the fact that $z_n=\sum_{i=1}^{n-1}z_i$ implies that $m_n=m_{n-1}+x_n+z_n=\sum_{j=1}^{n}x_j$ and so the protocol is always correct. It remains to prove the privacy of the protocol. Namely, that every coalition, given the output of the protocol and the input of coalition members, sees the same distribution of communication. We distinguish between two types of coalitions, depending on whether or not P_n is in the coalition. Consider a coalition $T=\{i_1,i_2,\ldots,i_p\}$, of size $p\leq t$, which does *not* include P_n. The view of this coalition, consists of the random bits $z_{i_1},\ldots,z_{i_p}$ they received from Q in the first step of the protocol, and of messages $m_{j_1},\ldots,m_{j_r}$ $(r\leq t)$ sent from players not in the coalition to players in the coalition. We claim that, for all $\vec{x}$, each assignment of values to these $p+r$ messages is obtained with probability $1/2^{p+r}$. Since this is true for all $\vec{x}$, the privacy with respect to this kind of coalitions follows. For each player $P_{j_k}\notin T$ that sends a message to a member of the coalition, denote by $P_{i(j_k)}$ the last player of the coalition before him (formally, $i(j_k)$ is the maximum value which is smaller than j_k and $P_{i(j_k)}\in T$; if no such index exists then $i(j_k)=0$). Now, we can express each message m_{j_k} as $m_{i(j_k)}+\sum_{i=i(j_k)+1}^{j_k}x_i+\sum_{i=i(j_k)+1}^{j_k}z_i$, where the last sum is denoted by Y_{j_k}. Clearly, given $\vec{x}$, the values of $z_{i_1},\ldots,z_{i_p}$ and $Y_{j_1},\ldots,Y_{j_r}$ are completely determined by the view of T. Since each of these $p+r$ values depends on different z_i's they are all equally distributed and independent. Therefore, every communication has probability $1/2^{p+r}$, as claimed.

The case of coalitions that include the player P_n is similar. There, however, the value of the message z_n received by P_n is already determined by the other elements of the view and *the output*. Hence, each communication which is consistent with the output has probability of $1/2^{p-1+r}$.

By the above analysis, it is already clear how to define the sets to which the sample space should be immune: for every $1\leq i\leq j\leq n-1$, let $T_{i,j}=\{i,\ldots,j\}$. This guarantees that each z_{i_j} is uniformly distributed (as we have the singleton T_{i_j,i_j} as a special case) and so is each Y_{j_k} (as it is the parity corresponding to the set $T_{i(j_k)+1,j_k}$). To get the independence among these $p+r\leq 2t$ values, it is sufficient to consider[7] the collection $\mathcal{T}$ of all sets which are composed from taking unions of at most $2t$ disjoint sets as above. Each such union has at most $2t$ intervals, therefore the number of sets in $\mathcal{T}$ can be bounded by $|\mathcal{T}|\leq\sum_{i=0}^{2t}\binom{n}{2i}=\left(\frac{n}{t}\right)^{O(t)}$. In other words, we replace step (1) of the above protocol by:

1'. Q chooses, using $O(\log|\mathcal{T}|)=O(t\log(n/t))$ random bits, a vector $\vec{z}=z_1,\ldots,z_{n-1}$ from a space which is immune to the sets in $\mathcal{T}$. He sends z_i to P_i $(1\leq i\leq n-1)$ and $z_n=\sum_{i=1}^{n-1}z_i$ to P_n.

The same proofs of correctness and privacy remain valid. Hence, we have just proved,

[5] all additions are modulo 2.

[6] The first player P_1 has to sum only z_1 and x_1 as he receives no other message. Alternatively, define $m_0=0$.

[7] this is a standard fact; for details, see the proof of Lemma 7 in the Appendix.

Theorem 1: There exists an n-party, t-private protocol in the trusted-party model to compute *xor* using $O(t\log(n/t))$ random bits.

To transform the above protocol to the standard model, we let the players $P_1,\ldots,P_t$ simulate the role of Q. That is, we replace (1') by the following:

1". Each P_j $(1 \le j \le t)$ chooses, using $O(t\log(n/t))$ random bits, a vector $\vec{z^j} = z_1^j,\ldots,z_{n-1}^j$ from a space which is immune to the sets in $\mathcal{T}$. Player P_j sends z_i^j to P_i $(1 \le i \le n-1)$ and $z_n^j = \sum_{i=1}^{n-1} z_i^j$ to P_n. Each P_i computes $z_i = \sum_{j=1}^{t} z_i^j$.

The key observation now is that if the coalition is $P_1,\ldots,P_t$ then the coalition gets no messages (from non-coalition players) during the protocol and hence gets no additional information. For any other coalition, there exists at least one player P_m, $1 \le m \le t$, whose random string is not known to the coalition. The same proof above, regarding the distribution of communications, can be repeated using only the choices of P_m in the argument. To conclude, we have proved:

Theorem 2: There exists an n-party, t-private protocol to compute *xor* using $O(t^2\log(n/t))$ random bits.

Remark: Note that the size of the sets $T_{i,j}$ defined above may be very large (up to $n-1$), hence the sample spaces of [S92] are not enough. It should also be clear that just by using a t-wise independent sample space the protocol fails. This is because large sets of variables may be dependent and hence their sum (i.e., the Y_{j_k}'s) is not equally distributed.

3.3 Lower Bound

In this section we prove the lower bound for the privacy problem. Again, we first consider the trusted-party model. We start with a simple combinatorial lemma.

Lemma 4: Let S be a set of at most K distinct vectors of n coordinates. Then, there are $k = \lfloor \log K \rfloor$ coordinates, and an assignment to these k coordinates such that there is a unique vector $\vec{s} \in S$ which is consistent with this assignment.

Proof: The proof is by induction on K. It can be easily verified for small values of K (e.g., $K = 2, 3$). Now, given a set S, we find a coordinate i in which not all the vectors have the same value. Choose this coordinate, and take a value which appears in at most half of the vectors (but does appear at least once). After this we are left with at most $K/2$ vectors and, by induction hypothesis, we can fix $\lfloor \log(K/2) \rfloor = k-1$ more coordinates. □

We are now ready to show the lower bound theorem for the trusted party model.

Theorem 3: Let $\mathcal{P}$ be a protocol for $n \ge 3$ players that allows computing *xor* t-privately, $t \le n-2$, using a trusted party Q. Then, the support of Q, S, has at least 2^t vectors.

Proof: By contradiction, assume that the trusted party has only $2^t - 1$ distinct vectors in its support set. By Lemma 4, there are $t-1$ processors, $P_{i_1},\ldots,P_{i_{t-1}}$, such that if each processor P_{i_j} receives a certain value s_{i_j} from Q, the trusted party, then they uniquely determine that the entire vector used by Q is $\vec{s}$. Thus, in this event (which occurs with positive probability), the coalition of the $t-1$ players knows all the values that Q distributed. Intuitively, in such a case, the protocol becomes deterministic. The impossibility result follows from the fact that there is no deterministic 1-private protocol for 3 or more parties (see [KR94]). To be more formal, consider the protocol for $n-(t-1) \ge 3$ players that is obtained by running $\mathcal{P}$ while giving each P_{i_j} input 0, and each P_i (in this set and out of it) the corresponding coordinate of $\vec{s}$. It can be seen that the modified protocol is deterministic (the random choices were fixed), 1-private (using the t-privacy of $\mathcal{P}$) and always correct. (Full details are provided in the proof of Theorem 4 below.) □

The above theorem implies that in the trusted party model t random bits are required by the trusted-party to allow t-private computation of *xor*. This is quite close to the $O(t\log(n/t))$ upper bound proven for this model. The same lower bound holds for the standard model (where no such trusted party exists):

Theorem 4: Let $\mathcal{P}$ be a protocol for $n \ge 3$ players that allows computing *xor* t-privately, $t \le n-2$. Then, $\mathcal{P}$ requires at least t random bits.[8]

Proof: Suppose, towards a contradiction, that $\mathcal{P}$ uses at most (over all inputs and all random choices) $d \le t-1$ random bits. One of the difficulties in transforming the proof from the trusted-party model (Theorem 3) to the standard model, is the possibility that

[8] by this we mean, that there exists an input $\vec{x}$ and an execution (i.e., an assignment for the random choices) in which a total of at least t bits are used.

in different executions different players toss the coins. Assume, without loss of generality, that $\vec{0}$ is an input on which d random bits are used in some executions, and that $P_1, \ldots, P_s$ is a set of $s \leq d$ players which in some execution (on input $\vec{0}$) make all the d random choices. Denote by $R_1, \ldots, R_s$ possible random choices for them in such an execution.

We will construct a new protocol $\mathcal{P}'$ that computes *xor* for $n - s \geq 3$ players, $P_{s+1}, \ldots, P_n$, 1-privately and deterministically. It is well known that such a protocol does not exist (see [KR94]) and hence we will get the desired contradiction. To do so, let $P_{s+1}, \ldots, P_n$ (who wish to compute the *xor* of $\vec{x}' = (x_{s+1}, \ldots, x_n)$) execute the protocol $\mathcal{P}$ on $\vec{x} = (0, \ldots, 0, x_{s+1}, \ldots, x_n)$; in addition, if any of these players needs to send a message to one of $P_1, \ldots, P_s$ he informs its value to everybody, and if he needs to receive a message from one of $P_1, \ldots, P_s$ he computes it himself by taking 0 as the input of each of them, the corresponding R_i (as fixed above), and the messages they received in previous rounds (which are known to all).

First, we need to prove that $\mathcal{P}'$ is deterministic; that is, for all $\vec{x}'$, the players never reach a point in their code that they are required to flip coins and to use it (to be formalized below). Note, that if $\vec{x}'$ is a 0-vector then so is $\vec{x}$. In this case, by the assumptions, none of $P_{s+1}, \ldots, P_n$ flips coins and the whole communication is fixed to some vector of values C. The problem is that it is not obvious that this is the case for all vectors $\vec{x}$ as above (we will prove that this is the case). Let $step_0$ be the first step in which for some input, some of $P_{s+1}, \ldots, P_n$ tosses a coin. In all previous steps the communication is deterministic (for all inputs) which implies that the messages sent are identical to those in C. To see this, suppose this is false and consider the first step for which on some input P_i sends to P_j the message 0 and on others he sends 1. Since this is the *first* such step, this implies that P_i behaves differently when the input $\vec{x}'$ is the all-0 vector, and when it is a vector in which only P_i and P_k (where P_k is any player different than $P_i, P_j, P_1, \ldots, P_s$) have input 1. This implies that in the original protocol, $\mathcal{P}$, the coalition $P_1, \ldots, P_s$ and P_j (which is of size at most t) could be able to distinguish between these two inputs (when $P_1, \ldots, P_s$ have randomness $R_1, \ldots, R_s$). But, as in both cases they have the same input and the output is the same, they should not be able to do that. So we proved that up to step $step_0$ for all inputs the communication is identical to C. Now, by assumption, in this step some players toss coins and "use" them. That is, for some input $\vec{x}'$, some player P_i sends to some player P_j either a 0 or 1 both with positive probability (it can be seen that it must be that $x_i = 1$). Again, as so far the communication was the same for all inputs, this is true for the vector in which only P_i and P_k have 1, which again allows the coalition $P_1, \ldots, P_s$ and P_j to distinguish between $\vec{0}$ (on which the communication must be C) and this vector (on which, with positive probability, the communication is different than C). Thus, contradicting the t-privacy of the original protocol. This implies that indeed $\mathcal{P}'$ is deterministic.

Now, as we know that no player will be required to toss coins in $\mathcal{P}'$, we can claim that $\mathcal{P}'$ is correct; each execution of it on input $x_{s+1}, \ldots, x_n$ has a corresponding execution of the original protocol (on input $0, \ldots, 0, x_{s+1}, \ldots, x_n$ and with randomness $R_1, \ldots, R_s$ and where none of $P_{s+1}, \ldots, P_n$ tosses coins) which by assumption is always correct. Note that the inputs given to $P_1, \ldots, P_s$ contribute nothing to the outcome (had we choose different inputs for these players we may need to flip the outcome).

Finally, we claim that $\mathcal{P}'$ is 1-private; the (deterministic) view in $\mathcal{P}'$ of any single player P_j in $\{P_{s+1}, \ldots, P_n\}$ is the same as the view of $P_1, \ldots, P_s$ together with P_j in the original protocol (where $P_1, \ldots, P_s$ have inputs $0, \ldots, 0$ and randomness $R_1, \ldots, R_s$). Note that inputs for which we have the privacy requirement in $\mathcal{P}'$ remain so if we extend them with the inputs of $P_1, \ldots, P_s$. Hence, the 1-privacy of $\mathcal{P}'$ follows from the t-privacy of $\mathcal{P}$. □

Remark: In fact the statement shown by the proof is stronger than what is required in few ways: (1) it shows that for *every* input there is an execution that uses at least t random bits; (2) it shows that not only t coins must be tossed but that the number of players that toss coins has to be at least t (even if each of them tosses many coins); and (3) it shows that the *support* of S is large no matter what is the probability distribution over S.

Acknowledgment

We wish to thank Noga Alon and Adi Rosén for discussions concerning Randomness and Privacy, and Amos Beimel, Ran Canetti, Benny Chor, and Oded Goldreich for very useful comments and suggestions. In particular, Ran Canetti came up with an alternative construction to the one appears in Section 3.2.

References

[AGHP90] N. Alon, O. Goldreich, J. Hastad, and R. Peralta, "Simple Constructions of Almost k-wise Independent Random Variables", Proc. of 31st FOCS, 1990, pp. 544–553.

[AS92] N. Alon, and J. Spencer, "The Probabilistic Method", John Wiley & Sons, 1992.

[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.

[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.*

[BSV94] C. Blundo, A. De-Santis, and U. Vaccaro, "Randomness in Distribution Protocols", *Proc. of 21st ICALP*, 1994.

[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*, 1995.

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

[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, April 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.

[CG85] 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.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson, "How to Play Any Mental Game", Proc. of 19th STOC, 1987, pp. 218-229.

[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, pp. 252-263.

[KM94a] D. Karger, and R. Motwani, "Derandomization through Approximation: An NC Algorithm for Minimum Cuts", Proc. of 26th STOC, 1994, pp. 497–506.

[KM94b] H. Karloff, and Y. Mansour, "On Construction of k-wise Independent Random Variables", Proc. of 26th STOC, 1994, pp. 564–573.

[KY76] D. E. Knuth, and A. C. Yao, "The Complexity of Non-Uniform Random Number Generation", Algorithms and Complexity, pp. 357-428, ed. J. Traub, Academic Press, 1976.

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

[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.

[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, May 1992, pp. 273–284.

[KR94] E. Kushilevitz, and A. Rosén, "A Randomness-Rounds Tradeoff in Private Computation", *Advances in Cryptology*, Proceedings of Crypto '94, Y. Desmedt, ed., Springer-Verlag, Lecture Notes in Computer Science, Vol. 839, pp. 397-410, 1994.

[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.

[Y86] A. C. Yao, "How to Generate and Exchange Secrets", Proc. of 27th FOCS, 1986, pp. 162-167.

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

A Parallel Construction

In this section we show how immune sample spaces can be constructed in parallel. Obviously, the randomized construction (Section 2.2) can be parallelized. Our goal however, is to show how this can be done deterministically. That is, by an NC-algorithm.

To do so, we construct a matrix M as in Section 2.3 but this time in a column-by-column fashion. Also, we think of each set T_i as a vector in $\{0,1\}^n$, which is simply its characteristic vector. Note that if we find a column $M^j \in \{0,1\}^n$ such that the inner product $\langle T_i, M^j \rangle$ is 1 (i.e., $1 = \sum_{k=1}^{n} T_{i,k} M^j_k = \sum_{k \in T_i} M^j_k$) then T_i has the desired property that $\sum_{j \in T_i} M_j \neq \vec{0}$. We will use for the construction ϵ-bias spaces. These are sets $B \subset \{0,1\}^n$ of size polynomial in n and $1/\epsilon$ such that, for every $x \in \{0,1\}^n$ (and in particular every T_i), $\Pr_{b \in B}(\langle x, b \rangle = 1)$ is at least $\frac{1}{2} - \epsilon$. These spaces were studied in [NN90, AGHP90], which in particular proved that they can be constructed in NC^1. By a simple counting argument, for every collection of t vectors in $\{0,1\}^n$ there exists $b \in B$ whose inner product with $\frac{1}{2} - \epsilon$ of the vectors gives 1. The idea of the construction is that this vector can be found in parallel. Let $\epsilon = 1/m$ and $\ell = \left\lceil \log_{\frac{2}{1+2\epsilon}}(m+1) \right\rceil$ (again, the size of the space is $2^\ell = O(m)$).

- Construct an ϵ-bias space B.
- For $j = 1, \ldots, \ell$

  Consider all the sets $T_{i_1}, \ldots, T_{i_t}$ which were not marked as "done" yet. Find (in parallel) a vector $b \in B$ which is good for at least $\frac{1}{2} - \epsilon$ these sets. Mark these sets as "done". Let $M^j = b$.

As argued above, there exists some vector $b \in B$ which is good for $1/2 - \epsilon$ of the sets. In order to find it in parallel, we can check for each b and each T_i whether b is good for T_i (this can be done in $O(\log n)$ time on EREW PRAM[9]). For each b, we need to count the number of T_i's for which it is good (this takes $O(\log m)$ time on an EREW PRAM). Finally, we need to chose the b with the maximum number of good sets (this takes $O(\log n)$ time on EREW PRAM). Therefore, each iteration takes $O(\log n + \log m)$ time on EREW PRAM. Since there are at most $O(\log m)$ iterations, the total time is $O(\log m \log n + \log^2 m)$, which implies that this is an NC^2 algorithm.

Lemma 5: Let the matrix M be constructed as above. Then, *space*(M) is immune to all sets $T_1, \ldots, T_m$.

Proof: By the above discussion, no matter what is the set of un-marked T_i's, there exists b which is good for at least $\frac{1}{2} - \epsilon$ of them. Hence ℓ columns are enough. □

[9] EREW PRAM = Exclusive Read Exclusive Write Parallel Random Access Machine

B Extensions of the Construction.

Many times we are guaranteed some additional properties of the sets $T_1, \ldots, T_m$. Of a particular interest is the case where for every $p \in \{1, \ldots, n\}$ it is known that p belongs to at most d of the sets (see [S92]). In this case, we are able to generate a space of size at most $2d$ (instead of size $2m$ guaranteed by the construction of Section 2.3).

Lemma 6: Let $T_1, \ldots, T_m$ be a collection of sets such that each element appears in at most d of them. Construct a matrix M using the construction of Section 2.3 with the exception that now $\ell = \lceil \log(d+1) \rceil$. Then, $space(M)$ is immune to all sets $T_1, \ldots, T_m$.

Proof: The proof is similar to the proof of Lemma 3, except that now the number of relevant sets for each row k can obviously be bounded by d (instead of m). Since $2^\ell \geq d+1$, ℓ columns are sufficient. □

Schulman [S92] considered the following problem: given sets $H_1, \ldots, H_p$ construct a space whose projection on every H_i yields a uniform distribution. Using our constructions, we get the following corollary, which is a new proof for the results of [S92]:[10]

Corollary 5: [11] Let $H_1, \ldots, H_p$ be a collection of sets of size at most h. Then (1) A space whose projection on every H_i is uniform, of size $O(p \cdot 2^h)$, can be constructed in polynomial time; and (2) If in addition every element in $\{1, \ldots, n\}$ appears in at most d of the H_i's then the construction can be made of size $O(d \cdot 2^h)$.

Proof: Given sets $H_1, \ldots, H_p$ define $\mathcal{T} = \{T | \text{ for some } i, T \subseteq H_i\}$. Observe that if a space is immune to the sets of $\mathcal{T}$ then its projection on every H_i is uniform. Again this is a standard fact (see, e.g., [AS92]); to see this, consider a set H_i. When specifying the probability of the parity for each $T \subseteq H_i$, we essentially determine the Fourier transform of the distribution, hence we uniquely determine the probability distribution over H_i. Since we require that all the sets are immune it implies that the probability distribution over the H_i is the uniform one.
The two parts of the corollary now follow from Lemma 3 and Lemma 6 respectively. □

It is sometimes useful to have a sample space which is not only immune to a single test T_i but is t-wise immune; namely, if we take $k \leq t$ tests then we get each combination of the k parities with probability 2^{-k}. This of course is possible only if the tests are independent (for example, if T_1 and T_2 are disjoint sets, then the result of the parity test $T_3 = T_1 \cup T_2$ is always the sum of the parity tests T_1 and T_2). We show here how to transform our construction into a t-wise immune sample space.[12] Let $T = T_1 \oplus T_2 \oplus \ldots \oplus T_k$ denotes the set of elements that appear in an even number of sets in $T_1, \ldots, T_k$. This operation can best be viewed by looking at the characteristic vectors of the sets. Then, the characteristic vector of T is the sum (over $GF(2^n)$) of the characteristic vectors of $T_1, \ldots, T_k$. We say that $T_1, \ldots, T_k$ are *independent* if no subset of them gives $T = \emptyset$ (alternatively, if their characteristic vectors are linearly independent over $GF(2^n)$).

Lemma 7: Given $T_1, \ldots, T_m$, let $\mathcal{T}$ be the collection of all (non-empty) sets of the form $T_{i_1} \oplus \ldots \oplus T_{i_k}$, where $k \leq t$. If a sample space is immune to $\mathcal{T}$, it is t-wise immune to $T_1, \ldots, T_m$.

Proof: Let $b_1, \ldots, b_m$ be any boolean values and consider the event in which, for all i, the parity of the inputs in T_i is b_i. Note that such an event is equivalent to the event $\prod_{i=1}^{m} (\chi_{T_i}(x) - b_i)/2 \neq 0$, where $\chi_{T_i}(x) = (-1)^{\sum_{j \in T_i} x_i}$ (these are just the Fourier basis functions). Observe that once we multiply out the product, in the above expression, we have only terms of the form $\chi_A(x)$, where $A \in \mathcal{T}$, i.e. the event depends only on the parity of sets of the form $A = T_{i_1} \oplus \ldots \oplus T_{i_k}$, where $k \leq t$. □

We can now use each of the constructions presented above. For example,

Corollary 6: Let $T_1, \ldots, T_m$ be a collection of sets. There exists a sample space which is t-wise immune to the sets $T_1, \ldots, T_m$. The size of the space is bounded by the number of different sets of the form $T_{i_1} \oplus \ldots \oplus T_{i_k}$ (where $k \leq t$) which is at most $O(\sum_{i=0}^{t} \binom{m}{i}) = O(m^t)$. This space can be constructed in time polynomial in n and m^t.

[10] The construction obtained is *different* than that of [S92]. In [S92] the vectors of the space are constructed directly without going through, what we call, the generating matrix. We believe that the construction based on our approach is somewhat simpler.

[11] Both this corollary and Lemma 6 are not used in the present paper and appear here only to exemplify the power of our construction.

[12] this transformation is standard and described here for the sake of completeness.