Fault-tolerant Computation in the Full Information Model*
(extended abstract)

Oded Goldreich[†] Shafi Goldwasser[‡] Nathan Linial[§]

July 15, 1991

Abstract

We initiate an investigation of general fault-tolerant distributed computation in the *full-information* model. In the full information model no restrictions are made on the computational power of the faulty parties or the information available to them. (Namely, the faulty players may be infinitely powerful and there are no private channels connecting pairs of honest players).

Previous works, in this model, have concentrated on the particular problem of simulating a single bounded-bias global coin flip (e.g. Ben-Or and Linial [4] and Alon and Naor [1]). We widen the scope of investigation to the general question of how well can arbitrary fault-tolerant computations be performed in this model. The results we obtain should be considered as first steps in this direction.

We present efficient two-party protocols for fault-tolerant computation of any two-argument function. We prove that the influence of dishonest player in these protocols is the minimum one possible (up to polylogarithmic factors).

We also present efficient m-party fault-tolerant protocols for sampling a general distribution ($m \geq 2$). We present efficient m-party protocols for computation of any m-argument function, and prove for these protocols that for "most" functions, the influence of any t dishonest players on the outcome of the protocol is the minimum one possible (upto polylogarithmic factors).

1 Introduction

The problem of how to perform general distributed computation in an unreliable environment has been extensively addressed. Two types of models have been considered. The first model assumes that one-way functions exist and considers adversaries (faults) which are computationally restricted to probabilistic polynomial time [18, 10, 19, 11, 9, 2]. The second model postulates that private channels exist between every pair of players [3, 7, 8, 15, 13]. Hence, in both models fault-tolerance is achieved at the cost of restricting the type of faults.

We want to avoid any such assumption and examine the problem of fault-tolerant distributed computation where the faults are computationally unrestricted, and no private channels are available. Clearly the assumption that one-way functions exist is of no use here. The situation here corresponds to games of complete information.

The general problem can be described informally as follows: m players are interested in globally computing $v = f(x_1, ..., x_m)$ where f is a pre-determined m-argument function and x_i is an input given to party i (and initially known only to it). The input x_i is assumed to have been drawn from probability distribution D_i (which without loss of generality can be assumed to be uniform). A coalition F of faulty players may favor a particular value v for f and play any strategy to maximize the probability of such an outcome. We want to bound, for each value v in the range of f, the probability (under the best strategy for the faults) that the outcome of the protocol (used to distributively compute f) is v. How good can this bound be?

Regardless of the protocol under consideration, there is always one avenue that is open for the faulty

*Supported in part by NSF grant No. CCR-8657527, DARPA grant No. DAAL03-86-K-0171, and grants No. 89-00312 and 89-00126 from the United States - Israel Binational Science Foundation (BSF), Jerusalem, Israel.

[†] Computer Science Dept., Technion, Haifa, Israel.

[‡] Laboratory for Computer Science, MIT, Cambridge, USA.

[§] Computer Science Dept., Hebrew University, Jerusalem, Israel.

CH3062-7/91/0000/0447$01.00 © 1991 IEEE

players, namely, alter their input values to ones under which the value v is most likely. This is always possible, since players' inputs are not visible by others. That is, $q_v := max_{x_i, i \in F} \text{Prob}(f(\vec{x}) = v$ where $x_j \in_R D_j, j \notin F)$ is a lower bound on the influence of coalision F towards value v, no matter what protocol is used.

Consider the simple procedure in which each player announces its x_i, and the global output is taken to be $f(x_1, ..., x_m)$. If all players (including the faulty ones!) act simultaneously, then for every v, the probability of v being the outcome is indeed q_v. Unfortunately, in a distributed network simultaneity cannot be guaranteed, and a delayed action by the faults can result in much better results for them (e.g., for $f = \sum_{i=1}^{m} x_i \bmod N$ with $x_i \in \{0, 1, ..., N-1\}$, $q_0 = \frac{1}{N}$, but a single faulty player acting last has complete control of the outcome).

In both of the previously studied models (private channels or computationally bounded faults) protocols were developed where for all values v and all *minority* coalitions F, the probability of outcome v is as close to q_v as desired. The key to these protocols is the notion of *simultaneous commitment.* The protocols start in a stage where each player P_i commits its input x_i. It should be stressed that a faulty party may alter its input in this stage but not later and that a party's commitment is "independent" of the inputs of the other parties.

Obviously, in the full-information model such a qualitative notion of commitment cannot be implemented (even if the faulty parties are in minority). Instead, we need to look for quantitative results. Malicious players can and will be able to "alter their inputs" throughout the execution of the protocol in order to influence the outcome. Yet, we can bound the influence gained by their malicious behaviour.

The main focus of this article is the two-player case of this problem. Even this restricted case provides interesting problems and techniques. We resolve the main problems in this case, showing:

1. A lower bound: for every function f and every value v, no protocol can guarantee that the probability for outcome v be below $\max(q_v, \sqrt{p_v})$, where $p_v = \text{Prob}(f(\vec{x}) = v | x_i \in_R D_i))$.

2. More interestingly, we show a matching (upto poly-logarithmic factor) constructive upper bound. We describe a (generic) probabilistic polynomial time protocol that computes f, given a single oracle access to f, such that for all v, the probability that f evaluates to v is bounded above by

$$O(\log^4(\frac{1}{p_v})\ max(q_v, \sqrt{p_v}))$$

The spirit of our protocol is best illustrated by the following example.

Example: Define $id(x, y) = 1$ if $x = y$ and 0 otherwise. Suppose that the local inputs x, y are chosen uniformly in $\{0,1\}^n$. Clearly, $p_1 = \frac{1}{N}$, and $p_0 = 1 - \frac{1}{N}$, where $N = 2^n$. A protocol in which the first player declares x and then the second player declares y allows the second player complete control on the value of *id*. A protocol in which the two players alternatingly exchange bits in the description of their inputs is no better if these bits are exchanged in the same order (i.e., both parties send their respective ith bit in round i). A much better idea is for the two players to alternate in describing the bits of their inputs but do so from opposite directions (i.e., in round i the first party sends its ith bit whereas the second party sends its $(n-i+1)$th bit). Clearly, whichever player is faulty, the probability that the outcome of this protocol is "1" is bounded by $\frac{1}{\sqrt{N}}$. In light of the lower bound, this is the best result possible. This idea of gradually revealing appropriately chosen "bits of information" is the key to the general problem of two-party computation.

The problem of m-party computations, where a subset of $t < m$ faults may exist, is more involved than the two-party case (even for $m = 3$). (For more elaborate discussion see beginning of Section 5). Here, we prove the following results:

1. First we consider the problem of collectively sampling a given distribution, say without loss of generality the uniform distribution on strings in $\{0,1\}^l$. We provide a probabilistic polynomial time sampling protocol such that for every $S \subset \{0,1\}^l$, for every t faults,

$$\text{Pr}(\text{sample } \in S) \leq (\frac{|S|}{2^l})^{1-c(\frac{t}{m})}$$

for some constant $c > 0$. This result is the best possible (up to better constant c), and is superior to the bound obtained by the trivial protocol which consists of l repeated applications of "collective coin flipping".

2. Based on the above sampling protocol, we present a (generic probabilistic polynomial-time) protocol that works well for computing *almost all* functions: Fix a probability distribution π on the range set R, and let $\mathcal{F}$ be the set of all mappings $f : D_1 \times D_2 \times \ldots \times D_m \mapsto R$ where each $v \in R$ is obtained with probability $p_v = \pi(v)$. Then, for every $\epsilon > 0$, for all but an ϵ fraction of the functions $f \in \mathcal{F}$, for every $v \in R$, and any set of faults F, the probability that f evaluates to v is bounded above by

$$O(\log(1/p_v) + \log(1/\varepsilon)) \cdot {p_v}^{1-O(\frac{|F|}{m})}$$

Previous Work in the Full Information Model

Collective coin flipping, i.e., common bounded-biased sampling in $\{0,1\}$ has been considered in this full-information model before [4, 5, 14, 1]. Matching lower and (constructive) upper bounds of $\frac{1}{2} + \theta(\frac{t}{n})$ for $t < \frac{n}{3}$ has been shown (by Ben-Or and Linial [4] and Alon and Naor [1], respectively). Our work can be viewed as an extension of these investigations of influences of players on Boolean functions (i.e., $Range(f) = \{0,1\}$). Some differences with the general situation should be pointed out. First, since $Range(f)$ is arbitrary, we must be able to sample in sets with more than two elements. We call this the *sampling problem.* Note that the following obvious approach to the problem fails - i.e., repeatedly apply a given coin tossing procedure to select a random member from a larger set. The performance of this method is inferior to our protocols[1]. Secondly, in the present study f may depend on inputs supplied to the players seperatey and prior to the protocol, and it is demanded that the final value v be legally computed with respect to the inputs of non-faulty players.

Relation to works on Slightly-Random Sources

In this paper we present a multi-party protocol for sampling a set of strings $\{0,1\}^l$. In "sampling" we mean producing a single string in $\{0,1\}^l$ so that, for every subset $S \subset \{0,1\}^l$, the probability that the sample hits S is related to the density of S. Our protocol uses the collective coin flipping of [1] as a subroutine. In fact, our sampling protocol can be viewed as a deterministic reduction to the problem of collective coin tossing. The collective coin can be viewed as a slightly random source in the sense of Santha and Vazirani [16], i.e., a *SV-source.* Hence, our result can be interpreted as presenting a sampling algorithm which uses a SV-source. Our sampling algorithm performs much better than the obvious algorithm which uses as sample a sequence of coins produced by the source. In the obvious sampling algorithm the probability of hitting S may depend (exponentially) also on l and not only on the density of S, as in our algorithm. Furthermore, in the obvious algorithm very sparse sets may be hit with probability ≈ 1, whereas such sets are hit with small probability in our algorithm.

Our sampling algorithm provides an alternative way of recognizing languages in BPP by polynomial-time algorithms which use a SV-source. First, reduce the error probability in the BPP-algorithm so that it is negligible (i.e. smaller than any polynomial fraction). Next use our sampling algorithm to produce a sequence of coin tosses for a *single run* of the BPP-algorithm. This method is different from the original method of Vazirani and Vazirani [17] (adopted also in [6]) where the BPP-algorithm is invoked many times, each time with a different sequence of coin tosses.

2 Preliminaries

Throughout the paper we represent the function $f : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}^*$ by an N-by-N table, where $N \overset{\text{def}}{=} 2^n$. An entry, (x, y), in the table which has value v (i.e., $f(x,y) = v$) is called a *v-entry.* The following quantaties, related to the function f and a value v in its range, are central to our analysis.

Notation: The *density of v*, denoted p_v, is the fraction of v-entries in the table of f (i.e., $p_v = |\{(x,y) : f(x,y) = v\}|/2^{2n}$). The *maximum row density of v*, denoted r_v, is the maximum, taken over all rows, of the fraction of v-entries in a row of f (i.e., $r_v = \max_{x \in \{0,1\}^n}\{|\{y : f(x,y) = v\}|/2^n\}$). The *maximum column density of v* is denoted $c_v = \max_{y \in \{0,1\}^n}\{|\{x : f(x,y) = v\}|/2^n\}$. The *maximum a-priori influence towards v*, denoted q_v, equals $\max\{r_v, c_v\}$.

Throughout the paper, we consider the case of uniform input distribution. Namely, we assume that each input is selected uniformly from $\{0,1\}^n$ and independently of the other input(s). The more general case,

[1] An alternative method which performs even worse is to try to generalize the work of Alon and Naor [1] as follows: the method of [1] consists of randomly selecting one of the players who is appointed to flip a fair coin. Letting this player select a random string is a natural idea, but it is obvious that this approach performs very poorly for a sample space of non-constant size.

where each input is selected for an arbitrary distribution (yet independently of the other inputs) can be reduced to the uniform (see our technical report [12]).

We call a player *honest* if it follows the protocol. *Dishonest* players which may deviate arbitrarily from the protocol. In presenting our possitive results (i.e., our protocols) we assume, without loss of generality, that dishonest players do not deviate from the protocol in a manner which may be detected. This assumption can be easily removed by augmenting our protocols with simple detection and recovery procedures.

3 Lower Bounds

Theorem 1 : *Let $f: D_1 \times D_2 \times \ldots \times D_m \mapsto R$ be a function of m variables, Π an m-party protocol for computing f, and $v \in R$ be a value in the range of f. Consider performing Π where players in the set S are dishonest, while all other players are honest. Let ϕ_S be the maximum, over all strategies of coalition S of the probability of the outcome being v. For any $1 \leq t \leq m$ there is a coalition T of t players with $\phi_T \geq p_v^{1-t/m}$.*

The proof appears in our technical report [12].

Corollary 1 : *Let $f: \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}^*$ be an arbitrary two-argument function, Π an arbitrary two-party protocol for computing the function f, and v a value in the range of f. Then there exists at least one of the players which by playing (possibly) dishonestly can force the outcome to be v with probability at least $\max\{q_v, \sqrt{p_v}\}$ (the other party is played honestly).*

4 Two-Party Protocols

In this section we present protocols which meet the lower bounds presented in section 3, up to a polylogarithmic factor. We first present a general framework for the construction of such protocols (subsection 4.1), argue that this framework does indeed yield protocols meeting the lower bound (subsection 4.2), and finally use the framework to present *efficient* protocols meeting the lower bound (subsection 4.3).

Without loss of generality, we assume throughout this section that, for every value v in the range of f, each row and column in the table of f has at least $\frac{p_v}{4} \cdot 2^n$ entries of value v. Intuitively, the influence towards value v can not decreased if the number of v-entries is increased.

4.1 Framework for Protocols Meeting the Lower Bounds

The goal of the protocol is to enable the parties to gradually reveal their inputs to each other in a manner guaranteeing the minimum possible influence of one party on the value of the function.

The protocol proceeds in n rounds, each consisting of two steps. In each step one party sends one bit of information about its input to the other party. In the next step the other party sends such a bit. The bits sent by each party specify in which side, of a 2-partition of the residual input-space, its actual input lies. These partitions must satisfy the "value-balance" property to be discussed below. Following is the code of the generic protocol.

Inputs: $x \in X_0 \overset{\text{def}}{=} \{0,1\}^n$ for the row player, $y \in Y_0 \overset{\text{def}}{=} \{0,1\}^n$ for the column player.

Round i**:** Let (X_i^0, X_i^1) be a partition of X_i, and (Y_i^0, Y_i^1) a partition of Y_i.
The row player sends $\sigma \in \{0,1\}$ such that $x \in X_i^\sigma$.
Let $X_{i+1} \overset{\text{def}}{=} X_i^\sigma$.
The column player sends $\sigma \in \{0,1\}$ such that $y \in Y_i^\sigma$. Let $Y_{i+1} \overset{\text{def}}{=} Y_i^\sigma$.

For the protocol to achieve its goal of minimum possible influence of each party it employs 2-partitions satisfying the following *value-balance* property. The two sides, of the column (resp. row) partition, are balanced with respect to each value v in the range of f. Namely, the v-entries appearing in specific subsets of the columns (resp. rows) are distributed almost equally in the two sides of the column (resp. row) partition. (see Definition 2 below.) The v-balance property is used to argue that as long as submatrices of the residual table contains sufficiently many v-entries, the density of the value v remains as its density in these submatrices of the original table, no matter how the players act.

Motivation to the analysis of the protocol

In analyzing the influence of a dishonest party we consider, w.l.o.g., the probability that the row player (following an arbitrary adversarial strategy) succeeds in having the protocol yield a particular value v (in the range of f). For simplicity, we consider first the special case where $q_v = p_v$. In this case there are exactly

$K \overset{\text{def}}{=} p_v \cdot N$ entries of value v in each row of the function table. The analysis proceeds in three stages:

1) Consider the first $\log_2 K$ rounds. If all column (resp. row) partitions employed were halving the number of v-entries in each row (resp. column), then at the end of this stage the residual $\frac{1}{p_v}$-by-$\frac{1}{p_v}$ table would have contained a single v-entry in each row (resp. column), thus preserving the density of v-entries in each row and column. Using the v-balance property of the partitions, we show that this is roughly the situation (see Corollary 2).

2) Consider the next $\frac{1}{2} \cdot \log_2(1/p_v)$ rounds. If each row (resp. column) partition employed was halving the number of v-entries in the residual table, then at the end of this stage the residual $\frac{1}{\sqrt{p_v}}$-by-$\frac{1}{\sqrt{p_v}}$ table would have contained a single v-entry, thus preserving the density of v-entries. Using the v-balance property of the partitions, we show that this is roughly the situation (see Lemma 2).

3) At the last $\frac{1}{2} \cdot \log_2(1/p_v)$ rounds the row player can force the outcome to be v only if the input of the column player is a column containing a v-entry. Clearly, if there are at most Δ v-entries before this stage begins then the probability that the input of the column player is in a column containing a v-entry is bounded by $\Delta\sqrt{p_v}$.

Analysis of the Protocol: The Special Case of $q_v = p_v$

First we define the "value-balance" property. The definition is phrased for column partition. An analogous definition holds for row partitions.

Definition 1 (almost unbiased partitions): *Let (Y_i^0, Y_i^1) be a (column) partition, S a subset of Y_i, and $b > 1$. Denote $k \overset{\text{def}}{=} |S|$, and $k^\sigma \overset{\text{def}}{=} |Y_i^\sigma \cap S|$ (for $\sigma \in \{0,1\}$). We say that the partition (Y_i^0, Y_i^1) is* at most b-biased *with respect to S if the following hold:*
1) If $k > b^4$ then $|k^0 - \frac{k}{2}| < k^{3/4}$.
2) If $b < k < b^4$ then $|k^0 - \frac{k}{2}| < \frac{k}{20}$.

In our analysis of the protocol, we assume that it utilizes partitions which are at most $O(\log(1/p_v))$-biased with respect to specific sets. The constant in the O-notation will be denoted δ and will be determined as a function of other constants which appear in the analysis (see subsections 4.2 and 4.3). We denote $\Delta_v \overset{\text{def}}{=} \delta \log_2(1/p_v)$. Whenever obvious from the context, we abbreviate Δ_v by Δ.

Definition 2 (value-balanced partitions): *Let (Y_i^0, Y_i^1) be a (column) partition, and v be a value in the range of f. We say that the partition (Y_i^0, Y_i^1) is* v-balanced *if it has the following two properties:*
P1*) The partition is* v-balanced with respect to rows*: for every (remaining) row $x \in X_i$, the partition is at most Δ_v-biased with respect to set of columns having v-entries in row x (i.e., the set $S_x \overset{\text{def}}{=} \{y \in Y_i : f(x,y) = v\}$).*
P2*) If $|Y_i| < 2/p_v$ then the partition is* v-balanced with respect to the standard colouring*: Consider a standard minimum colouring, $\xi : X_i \times Y_i \mapsto \{0,1\}^*$, of the v-entries of the $X_i \times Y_i$ table so that no two v-entries in the same column or row are assigned the same colour. For every colour α, the partition is at most Δ_v-biased with respect to the set of columns containing a v-entry of colour α (i.e., the set $S^\alpha \overset{\text{def}}{=} \{y \in Y_i : \exists x \in X_i \text{ s.t. } f(x,y) = v \text{ and } \xi(x,y) = \alpha\}$).*

An elementary technical claim that we use in the analysis follows

Claim 1 : *Let $\alpha < 1$. Suppose that $z_{i+1} < \frac{z_i}{2} + (z_i)^\alpha$, for every i $(0 \leq i \leq T)$. Then, there exists a constant c_α, so that $z_t < \frac{z_0}{2^{t-1}}$, for every $t < \min\{T, (\log_2 z_0) - c_\alpha\}$.*

Lemma 1 : *Let v be a value in the range of f, and suppose that the protocol uses v-balanced column partitions. Let K_x denote the number of v entries in the original row x. Then after the first $\log_2 K_x$ rounds, no matter how the row player and the column player play, the number of v entries in the residual x row is at most Δ_v $(= O(\log 1/p_v)$.*

proof idea: The analysis uses only the fact that the column partitions satisfy Property (P1) (i.e., are v-balanced with respect to each row). □

Corollary 2 : *Let v be a value in the range of f, and suppose that $q_v = p_v$. Suppose that v-balanced column (resp. row) partitions are used. Then after the first $n - \log_2(1/p_v)$ rounds, no matter how the row player and the column player play, the number of v entries* *in each residual row (resp. column) is at most Δ_v $(= O(\log 1/p_v))$.*

Lemma 2 : *Let $M < 2/p_v$. Consider an M-by-M table so that B is an upper bound on the number of v-entries in each row and in each column. Suppose that the protocol is applied to this table, using v-balanced row and column partitions. Then after the first $\frac{1}{2}\log_2 M$ rounds, no matter how the row player and the column player play, the number of columns (in the resulting $\sqrt{M}$-by-$\sqrt{M}$ table) containing a v-entry is $O(B\Delta_v)$.*

proof sketch: The analysis uses only the fact that the row and column partitions satisfy Property (P2) (i.e., are v-balanced with respect to the standard colouring). Note that the standard colouring uses at most $2B + 1$ colours since the underlying graph has maximum degree $\leq 2B$. Let α be a colour. In each row and column there is at most one v-entry of colour α, hence each row/column partition effect the number of remaining v-entries of colour α exactly as the column partition effects the number of v-entries in a specific row. Hence, using the same arguments as in Lemma 1, we see that after $\frac{1}{2}\log_2 M$ rounds the residual table contains at most Δ_v v-entries of colour α. The lemma follows. □

Using Corollary 2 and Lemma 2, we get

Corollary 3 : *Let v be a value in the range of f, and suppose that $q_v = p_v$. Suppose that the protocol uses v-balanced partitions. Then no matter how the row player plays, the probability that v is the output of the protocol is bounded by $O(\Delta_v^2\sqrt{p_v})$ $(= O((\log 1/p_v)^2\sqrt{p_v}))$.*

Analysis of the Protocol: The General Case

The analysis of the general case (where q_v does not necessarily equal p_v) is more cumbersome. In this extended abstract we provide only a sketch of the analysis. We assume, for simplicity, that $1/p_v$ is a power of 2, and analyze the influence of the row player. Throughout the analysis we introduce additional restrictions on the partitions used in the protocol. These restrictions are in fact properties that we latter prove to hold for random (and even "slightly random") partitions.

We classify the rows according to their density and apply the analysis separately to each class. Let $\rho_v(x)$ denote the density of v-entries in row x of the original table (i.e., $\rho_v(x) = |\{y \in Y_0 : f(x,y) = v\}|/|Y_0|$). For $0 \leq j \leq \log_2(1/p_v) - 1$, the class $R^j \overset{\text{def}}{=} \{x \in X_0 : \lfloor\log_2(1/\rho_v(x))\rfloor = j\}$ contains all rows with v-entry density between 2^{-j} and 2^{-j-1}. The class $R^{\log_2(1/p_v)} \overset{\text{def}}{=} \{x \in X_0 : \lfloor\log_2(1/\rho_v(x))\rfloor \geq \log_2(1/p_v) - 1\}$ contains all rows with v-entry density smaller than $p_v/2$. Recall that, without loss of generality, there are no rows of density less than $\frac{p_v}{4}$.

Clearly, the influence of the row player towards value v is bounded by the sum of her influences when restricting herself to inputs of a certain class. The classes are partitioned into two categories: those of density above $\sqrt{p_v}$ and those below this density.

First we bound the influence of the row player when restricting herself to inputs of the first category (i.e. of density $> \sqrt{p_v}$). There are at most $\sqrt{p_v}N$ such rows. In analyzing the situation after $\log_2(\sqrt{p_v}N)$ rounds of the protocol, we use assume that the partitions posses the following additional v-balance property, denoted **P3**: the row partitions are at most Δ-bias with respect to the set of rows of density greater than $\sqrt{p_v}$. One can show that after $\log_2(\sqrt{p_v}N)$ rounds at most Δ of the above ("heavy") rows will remain and furthermore that each such row maintains its original density up to a multiplicative factor of Δ. Hence, the influence of the row player when restricting herself to these rows is bounded by $\Delta^2 q_v = O((\log_2(1/p_v))^2 q_v)$.

We now bound the influence of the row player when restricting her input to rows of density $\approx \sqrt{p_v}2^{-i}$ (i.e., the class $R \overset{\text{def}}{=} R^{i+\frac{1}{2}\log_2(1/p_v)}$), for $1 \leq i \leq \frac{1}{2}\log_2(1/p_v)$. There are at most such $\sqrt{p_v}2^i N$ such rows. Consider the situation after $\log_2(\sqrt{p_v}2^{-i}N)$ rounds. In analyzing the situation, we further assume that the partitions posses the additional v-balance property, denoted **P4**: the row partitions are at most Δ-bias with respect to the set of rows of density $\approx \sqrt{p_v}2^{-i}$ (i.e., R). One can show that after these $\log_2(\sqrt{p_v}2^{-i}N)$ rounds at most $\max\{\Delta, 2^{2i}\}$ such rows will remain, and furthermore that each such row has at most Δ entries of value v. For simplicity, we assume without loss of generality, that we are left with a 2^{2i}-by-$\frac{2^i}{\sqrt{p_v}}$ table having at most Δ v-entries in each row. We classify the columns in this table by their approximate density. For $0 \leq k \leq 2i$, let $C^k \overset{\text{def}}{=} \{y \in Y_0 : \lfloor\log_2(1/\mu_v(y,R))\rfloor = k\}$, where $\mu_v(y,R)$ is the density of v-entries in the portion of column y restricted to rows R (i.e., $\mu_v(y) = |\{x \in R : f(x,y) = v\}|/|R|$). Clearly the influence of the row player towards v when restricting her input to R is the sum, over all k, of the probability that the input of the column player happens to be in C^k and the output of the protocol is v when the row player restricts her input to be in R.

Let us now bound the probability that the input of the column player happens to be in $C \stackrel{\text{def}}{=} C^k$. The number of v-entries in a column of C is $\approx 2^{-k}2^{2i}$ and hence the number of such columns is bounded by $\frac{2^{2i}\Delta}{2^{2i-k}} = 2^k\Delta$. Hence, the probability that the input of the column player is in such a column is bounded by $\frac{2^k\Delta}{2^i/\sqrt{p_v}} = \Delta 2^{k-i}\sqrt{p_v}$. Hence, if $k \leq i$ the probability that the input of the column player happens to be in C and the output of the protocol is v when the row player restricts her input to be in R, is bounded by $\Delta\sqrt{p_v}$.

We are left with the case $i < k \leq 2i$. Consider the situation in the $R \times C$ *rectangle* after additional $2i - k$ rounds. In analyzing the situation, we assume that the partitions posses the following additional v-balance properties, denoted P5 and P6. In **P5**, we require that the column partitions are at most Δ-bias with respect to the columns in C. In **P6**, we require that, for every $y \in C$, the row partitions are at most Δ-bias with respect to the set of rows in R having a v-entry in column y. One can show that after these additional rounds at most $2^{k-(2i-k)}\Delta$ columns are left, each having at most Δ entries of value v. For simplicity we assume, w.l.o.g., that exactly $2^{k-(2i-k)}\Delta$ columns are left.

Finally, consider the situation after another additional $k-i+\frac{1}{2}\log_2\Delta$ rounds. To analyze this situation, we augment the v-balance property for the last time. In this auxiliary property, denoted **P7**, we consider a standard minimum colouring of the v-entries in the $R \times C$ rectangle and also require that, for every colour, the row (resp. column) partitions are at most Δ-bias with respect to the set of rows (resp. columns) having v-entries coloured by a this specific colour. Using an argument similar to the one used in Lemma 2, one can show that after these $k - i + \frac{1}{2}\log_2\Delta$ rounds we are left with Δ v-entries in the resulting $\frac{2^i}{\sqrt{\Delta}}$-by-$2^{k-i}\sqrt{\Delta}$ table. Hence, the probability that the input of the column player happens to be in C and the output of the protocol is v when the row player restricts her input to be in R, is bounded by $\Delta 2^{k-i}\sqrt{p_v} \cdot \frac{\Delta}{2^{k-i}\sqrt{\Delta}} < \Delta^{3/2}\sqrt{p_v}$. Summing the probability over all k's we get $2i \cdot \Delta^2\sqrt{p_v}$ and summing over all possible i's we get a bound of $(\log_2(1/p_v))^{3/2}\Delta^{3/2}\sqrt{p_v} = O((\log(1/p_v))^{3.5} \cdot \sqrt{p_v})$. Using Definition 3 (below), we conclude

Theorem 2 : *Let $f : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}^*$ be an arbitrary two-argument function. Then, a two-party protocol for computing the function f which uses generalized value-balanced partitions satisfies the following: for every value v in the range of f, if one party plays honestly then the outcome of the protocol is v with probability bounded above by $O(\log^{3.5}(1/p_v) \cdot \max\{q_v, \sqrt{p_v}\})$.*

It is likely that some logarithmic factors (but not all) can be eliminated by a more careful analysis.

Definition 3 (generalized value-balanced partitions): *Let v be a value in the range of f. We say that a column (row) partition, of the residual M-by-M table, is generalized v-balanced if it satisfies properties (P1) through (P7) listed above. (Recall that property (P2) holds vacuasly if $M \geq 2/p_v$.) Properties (P5) through (P7) are also defined to hold whenever $M \geq 2/p_v$.*

4.2 On the Existence of generalized Value-balanced Partitions

In this subsection we prove the existence of generalized value-balanced partitions. We first bound the probability that a random column partition is not balanced with respect to a specific set of columns. In the analysis we use an unspecified constant, denoted c_1. The constant δ (in the definition of Δ_v) is determined by c_1 (in fact $\delta > 50 \cdot c_1$ will do, and $c_1 = 3$ suffices for all the results of this section).

Lemma 3 : *Let S be a set of cardinality k. Then, for every $c_1 > 0$ there exists δ ($= 50c_1$), so that the probability that a random partition is not Δ_v-biased with respect to S is bounded above by $(p_v/k)^{c_1}$.*

The proof appears in our technical report [12].

For the rest of the analysis we assume, without loss of generality, that the number of rows (columns) in the original N-by-N table with "v-entry density $\approx 2^{-j}$" is at least $\frac{2^j p_v}{4\log_2(1/p_v)} \cdot N$.

Proposition 1 : *Let $X_i \times Y_i$ be the residual table after i rounds of the protocol. Then there exist column partitions (of Y_i) that are generalized value-balanced. Furthermore, $\forall c_1 \exists \delta$,*

1) if $|Y_i| > 2/p_v$ then the probability that a random partition is not generalized v-balanced is bounded above by $(1/|Y_i|)^{c_1-2} \cdot p_v$.
2) if $|Y_i| \leq 2/p_v$ then the probability that a random partition is not generalized v-balanced is bounded above by $(\frac{p_v}{\log_2(1/p_v)})^{c_1-1}$.

The proof appears in our technical report [12].

Combining Theorem 2 and Proposition 1, we get

Corollary 4 : *Let f be as in Theorem 2. Then, there exists a (deterministic) two-party protocol for computing the function f satisfying the following: for every value v in the range of f, if one party plays honestly then the outcome of the protocol is v with probability bounded above by $O(\log^{3.5}(1/p_v) \cdot \max\{q_v, \sqrt{p_v}\})$.*

4.3 Efficient Protocols Meeting the Lower Bounds

The protocols guranteed by Corollary 4 are not efficient. In particular, merely specifying the partitions used by the protocol takes exponential space (in size of the inputs), not to mention that the proof is nonconstructive and that a naive construction requires double exponential time. Efficient implementation of the protocols are possible by using partitions which can be specified by polynomially many bits. These partitions will not be hard-wired into the protocol but rather selected online by the two parties. Namely, in the beginning of each step both parties will first execute a sampling protocol to select a partition for that step. The partition will be specified by an m-degree ($m = \text{poly}(n)$) polynomial over the field $F \overset{\text{def}}{=} GF(2^n)$ and a fixed partition of the elements of F to two equal parts F^0 and F^1. For example, suppose polynomial P (over F) is chosen to specify a partition of Y_i, then Y_i^σ is the set of all points $y \in Y_i$ satisfying $P(y) \in F^\sigma$. In order to prove the validity of this proposal it suffices to present a two-party protocol for sampling these partitions and to prove that a partition selected using this protocol is generalized v-balanced with probability at least $1 - p_v$. To this end we first bound the probability that, for an appropriately chosen $m = \text{poly}(n)$, a random m-degree polynomial does not induce a (generalized) v-balanced partition. Next, we present a two-party protocol for sampling l-bit strings and bound the influence of each party towards any set as a function of the density of that set.

Bounding the Probability of Non Balanced Partitions

We start by bounding the probability that a random $(\delta n)^4$-degree polynomial does not induce a (generalized) v-balanced partition.

Lemma 4 : *For every $c_1 > 0$ there exists δ, so that for every set S of cardinality k, the probability that a partition induced by a random $(\delta n)^4$-degree polynomial is not Δ_v-biased with respect to S is bounded above by $(p_v/k)^{c_1}$.*

The proof appears in our technical report [12].

Proposition 2 : *Let $X_i \times Y_i$ be the residual table after i rounds of the protocol. Then, for every $c_1 > 0$ there exists δ, so that*
1) if $|Y_i| > 2/p_v$ then the probability that a partition induced by a random $(\delta n)^4$-degree polynomial is not generalized v-balanced is bounded above by $(1/|Y_i|)^{c_1-2} \cdot p_v$.
2) if $|Y_i| \leq 2/p_v$ then the probability that a partition induced by a random $(\delta n)^4$-degree polynomial is not generalized v-balanced is bounded above by $(\frac{p_v}{\log_2(1/p_v)})^{c_1-1}$.

proof sketch: The proof is identical to the proof of Proposition 1, except that Lemma 4 is used instead of Lemma 3.□

Protocol for String Sampling

We now present a two-party protocol for sampling l-bit strings and bound the influence of each party towards any set as a function of the density of the set. The protocol is a simplification of the protocol for computing a function. The parties proceed in l rounds. In each round one party selects a ("random") partition of the residual sample space and the other party selects at random one of the sides of this partition. In the next round the parties switch roles. The partitions selected by each party must be provable fair in the sense that they divide the residual space into two sets of equal cardinality. To be specific, the partition is defined by a linear combination of the bits in the representation of the sample point. Following is the code of the protocol (the parties are called P_0 and P_1).

Round i: $P_{i \bmod 2}$ selects at random an l-dimensional binary vector v_i, from the set of vectors which are

linearly independent of the vectors used in previous rounds, and sends v_i to $P_{(i+1)\bmod 2}$.
$P_{(i+1)\bmod 2}$ selects at random $\sigma_i \in \{0,1\}$ and sends it to $P_{i \bmod 2}$.
(The residual sample space after round i is the set of l-dimensional binary vectors x satisfying, for every $j \leq i$, the inner-product-mod-2 of x and v_j equals σ_i.)

Proposition 3 : *Let S be an arbitrary subset of $\{0,1\}^l$ and $p \overset{\text{def}}{=} |S|/2^l$. Then the above two-party protocol satisfies the following: if one party plays honestly then the outcome of the protocol is in S with probability bounded above by $O(p^{\frac{1}{4}})$.*

proof sketch: Let U_i denote the residual sample space after round i, and $S_i \overset{\text{def}}{=} S \cap U_i$ ($U_0 = \{0,1\}^l$ and $S_0 = S$). The idea underlying the proof is to consider the cardinality of S_i and treat differently "small" and "large" S_i. In case S_i is "small', we get a sufficiently good bound for the probability of hitting S_i by multiplying the probability of hitting a specific element by the cardinality of S_i. In case S_i is "large", with sufficiently high probability $S_{i+1} \approx |S_i|/2$ and hence the density, $|S_i|/|U_i|$, is approximately preserved. Details appears in our technical report [12].

Remark 1 : The result claimed above is not the best one possible. Yet it suffices for sampling partitions for the generic protocol. Much better protocols can be obtained - see Theorem 4.

Main Result

Combining Propositions 2 and 3 with Theorem 2, we get

Theorem 3 : *There exists a (generic) two-party protocol, that when given oracle access to any two-argument function $f : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}^*$ satisfies the following properties :*
1) The protocol consists of a pair of uniform probabilistic polynomial-time programs (with one oracle call to the function f). If both parties play honestly and their inputs are x and y respectively, then the output is $f(x,y)$.
2) For every value v in the range of f, if one party plays honestly then the outcome of the protocol is v with probability bounded above by $O(\log^{3.5}(1/p_v) \cdot \max\{q_v, \sqrt{p_v}\})$. In case $q_v = p_v$, the probability bound can be improved to $O(\log\log(1/p_v) \cdot \sqrt{p_v})$.

proof sketch: The protocol is an implementation of the generic protocol in which the partitions are determined by poly(n)-degree polynomials which are selected using the sampling protocol described above. Using Propositions 2 and 3, we can bound the probability that the selected partition of Y_i is not generalized v-balanced by $p_v/|Y_i|$ if $|Y_i| > 2/p_v$ and by $p_v/\log_2(1/p_v)$ otherwise. Hence, the probability that one of the partitions used in the protocol is not generalized v-balanced can be bounded by $\sum_{i=\log_2(2/p_v)}^{n} \frac{p_v}{|Y_i|} + \log_2(2/p_v) \cdot (p_v/\log_2(1/p_v)) = O(p_v)$. Using Theorem 2, the theorem follows (for the general case). A bound of $O(\sqrt{p_v}\log^2(1/p_v))$ for the special case of $q_v = p_v$ can be obtained by using Corollary 3 instead of Theorem 2. The better bound of $O(\sqrt{p_v}\log\log(1/p_v))$ requires a slightly more careful analysis (see our technical report [12]).□

As stated in Remark 1, better sampling protocols (than the one referred in Proposition 3) can be obtained. This can be done either directly (by using the techniques used in the proof of Theorem 3) or as a corollary to Theorem 3 when applied to the function $f(x,y) \overset{\text{def}}{=}$ the bit-by-bit XOR of x and y. Both resulting sampling protocols use the simple sampling protocol and the bound presented in Proposition 3 as a bootstraping step.

Theorem 4 : *There exists a sampling protocol having the following properties:*
1) For every set S (of density p) if one party plays honestly then the outcome of the protocol is in S with probability bounded above by $O(\sqrt{p}\log\log(1/p))$.
2) The protocol consists of a pair of uniform probabilistic polynomial-time programs.

5 Towards the Multi-party Case

We believe that the ideas developed in the two-party case will prove useful also for the multi-party case. However, even the problem of computing an arbitrary 3-argument function by a 3-party protocol in the presence of one dishonest party is much more involved than the problem of computing an arbitrary 2-argument function by a 2-party protocol (i.e., the problem treated in the previous sections). The function can be represented as a three dimensional cube. One type of difficulty which arises is that a vertical layer (i.e. the subspace defined by $Z = z$) may contain many v-entries and yet there is no v-balanced partition along the Y-axis (e.g. $\{(x,y) : f(x,y,z) = v\} = \{x : f(x,1,z) = v\}$). For example, consider $f(x,y,z) = g(y,z)$. Interestingly, the above difficulty occurs only in "very few" functions (whereas in "most" functions this difficulty does not occur - see Theorem 5). Another difficulty which arises is that we cannot afford to let the parties reveal information in a predetermined order (as done in the two-party case). This difficulty is best demonstrated when considering the special case of where each input is one bit (i.e., $f : \{0,1\} \times \{0,1\} \cdots \times \{0,1\} \mapsto \{0,1\}^*$). The influence of parties which are last to reveal their input is more substential than the influence of parties which reveal their input first. This calls for choosing a random permutation to determine the order of playing. Thus, the role of a sampling protocol in the multi-party case is more central than in the two-party case. (Recall that in the two-party case sampling protocols where introduced only for efficiency purposes.)

Following is a generic protocol for multi-party computations.

Inputs: $x^{(j)} \in \{0,1\}^n$ for party j $(1 \leq j \leq m)$.

Round i**:** The parties use a sampling protocol (see below) for selecting at random an ordering of the parties (i.e. permutation of $\{1, 2, ..., m\}$).

Following the selected order, each party, at its turn, reveals its i^{th} bit.

Theorem 5 : *Let n and m be integers, and $\varepsilon > 0$ be a real number. Let $p_1, p_2, p_3, ..., p_V$ be positive real numbers such that $\sum_{i=1}^{V} p_i = 1$. Denote by F the set of functions from the m-fold Cartisian product of $\{0,1\}^n$ to $\{0,1\}^*$, in which fraction p_v of the domain maps to value v, for $1 \leq v \leq V$. Suppose that the above protocol uses a sampling protocol satisfying condition (1) of Theorem 6 below. Then for all but an ε fraction of the functions f in F the above m-party protocol satisfies: for every value v in the range of f, if $m - t$ parties play honestly then the outcome of the protocol is v with probability bounded above by $O(\log(1/p_v) + \log(1/\varepsilon)) \cdot {p_v}^{1-O(\frac{t}{m})}$.*

proof sketch: For most functions, there are at most two rounds in which the order of the parties is of any importance. These are the rounds during which the cardinality of the residual set of possible inputs is approximately $1/p_v$ (this may happen at the "middle" of one round or at the "end" of one round and the "beginning" of its proceeding round). In the previous rounds, in most functions, the density of the value v is about the same in all 2^m subcubes defined by all possible values of the bits revealed in this round. In each of the last rounds, there are very few (if at all) v-entries in the residual m-dimensional cube and the probability that a specific v-entry (in the cube at the beginning of that round) remains in the residual subcube at the end of that round is at most $2^{-(m-t)}$. The two crucial rounds (in which the number of entries is about $1/p_v$) are analyzed with more care, and in particular we take advantage on the randomness of the order of players in these rounds. For simplicity we assume that at the beginning of some round, and for $2^{m/3} < K < 2^{2m/3}$, there are K entries of value v in the residual cube. In this case we cannot guarantee that after $\log_2 K$ steps (bits revealed by different players), no matter which players make these steps, the density of the v-entries is preserved (not even for most functions). However, we can prove that for most functions, and for most choices of a sequence of $\log_2 K$ parties for these steps, the density of the v-entries is preserved during these steps. Another concern in the last $m - \log_2 K$ steps. In a random choice of ordering of the parties, these steps are expected to have the same proportion of honest players as in the set of all player (i.e., $(m - t)/m$). Details are omitted in this extended abstract. □

Theorem 6 : *There exists an m-party sampling protocol satisfying the following:*
1) for every set $S \subset \{0,1\}^l$, if $m - t$ parties plays honestly then the outcome of the protocol is in S with probability bounded above by $p^{1-O(\frac{t}{m})} \log(1/p)$, where $p \overset{\text{def}}{=} |S|/2^l$.
2) the protocol consists of m (identical) uniform probabilistic polynomial-time programs.

proof idea: The idea is to implement the proof of Theorem 4 in the multi-party context. The protocol proceeds in l rounds each. In each round, the m-parties first select at random (using a simpler sampling protocol) a poly(n,m)-degree polynomial specifying a partition of the residual sample space, and next use the collective-coin tossing protocol of Alon and Naor [1] to choose one side of this partition. The sampling protocol used to choose poly(n, m)-degree polynomials is similar except that the partitions are specified by linear transformations (as in the protocol of Proposition 3). These linear transformations are selected using a trivial sampling algorithm which consists of selecting each bit using the the collective-coin tossing protocol of Alon and Naor [1].□

Acknowledgment

Oded Goldreich would like to thank the Computer Science Department of the Hebrew University for providing him shelter in times of war; surprisingly some shelters are more pleasant than home! The authors wish to thank the Computer Science Department of Tel-Aviv University for use of its computing facilities.

References

[1] Alon N., and M. Naor, "Coin-flipping games immune against linear-sized coalitions", *SIAM J. on Computing*, to appear. Extended abstract in *Proc. of 31st FOCS*.

[2] Beaver D., and S. Goldwasser, D. Beaver., and S. Goldwasser, "Distributed Computation with Faulty Majority ", *FOCS*, 1989.

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

[4] Ben-or M., and N. Linial, "Collective coin flipping", *Randomness and Computation*, (S. Micali, ed.), JAI Press, pp. 91-115, 1989.

[5] Ben-Or, M., Linial, N., Saks, M., "Collective coin flipping and other models of imperfect randomness", *Colloq. Math Soc. Janos Bolyai* no. 52, Combinatorics Eger 1987, pp. 75-112.

[6] Chor, B., and O. Goldreich, "Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity", *SIAM J. Computing*, Vol. 17, pp. 230-261, April 1988.

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

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

[9] Galil, Haber, and Yung., "Cryptographic Computation: Secure Faulty-Tolerant Protocols and the Public Key Model", *CRYPTO*, 1989.

[10] Goldreich, O., S. Micali, and A. Wigderson, "Proofs that Yield Nothing but their Validity and a Methodology for Cryptographic Protocol Design", *27th FOCS*, 1986, pp. 174-187.

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

[12] Goldreich, O., S. Goldwasser, and N. Linail, "Fault-tolerant Computation in the Full Information Model", TR-682, Computer Science Dept., Technion, Haifa, Israel, 1991.

[13] Goldwasser, S. and L.A. Levin, "Fair Computation of General Functions in Presence of Immoral Majority", Crypto1990.

[14] J. Kahn, G. Kalai and N. Linial, "The influence of variables on boolean functions", *29th FOCS*, 1988, pp. 68-80.

[15] Kilian, J., "Founding Cryptography on Oblivious Transfer", 20^{th} STOC (1988), 20-29.

[16] Santha, M., and U.V. Vazirani, "Generating Quasi-Random Sequences from Slightly-Random Sources", *25th Symp. on Foundation of Computer Science*, pp. 434-440, 1984.

[17] U.V. Vazirani, and V.V. Vazirani, "Random Polynomial Time is equal to Slightly-Random Polynomial Time", *26th Symp. on Foundation of Computer Science*, pp. 417-428, 1985.

[18] Yao, A. C., "Protocols for Secure Computations", *23th FOCS*, pp. 160-164, 1982.

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