RESEARCH ARTICLE

Algorithmic Mechanisms for Reliable Crowdsourcing Computation under Collusion

Antonio Fernández Anta[1], **Chryssis Georgiou**[2], **Miguel A. Mosteiro**[3]*, **Daniel Pareja**[3]

1 IMDEA Networks Institute, Madrid, Spain, **2** Dept. of Computer Science, University of Cyprus, Nicosia, Cyprus, **3** Dept. of Computer Science, Kean University, Union, New Jersey, United States of America

* mmosteir@kean.edu

OPEN ACCESS

Citation: Fernández Anta A, Georgiou C, Mosteiro MA, Pareja D (2015) Algorithmic Mechanisms for Reliable Crowdsourcing Computation under Collusion. PLoS ONE 10(3): e0116520. doi:10.1371/journal.pone.0116520

Academic Editor: Cheng-Yi Xia, Tianjin University of Technology, CHINA

Received: July 31, 2014

Accepted: December 10, 2014

Published: March 20, 2015

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: A. Fernández Anta is partially supported by grants from the Comunidad de Madrid (http://www.madrid.org) (Cloud4BigData-CM, S2013/ICE-2894), the Spanish Ministerio de Ciencia e Innovacion / Ministerio de Economia y Competitividad (http://www.mineco.gob.es)(TEC2011-29688-C02-01), and the National Natural Science Foundation of China (http://www.nsfc.gov.cn) (61020106002). C. Georgiou is partially supported by research funds from the University of Cyprus (http://www.ucy.ac.cy/en/) (UCY-ED2014-CG). M. A. Mosteiro is partially supported by

Abstract

We consider a computing system where a master processor assigns a task for execution to worker processors that may collude. We model the workers' decision of whether to comply (compute the task) or not (return a bogus result to save the computation cost) as a game among workers. That is, we assume that workers are rational in a game-theoretic sense. We identify analytically the parameter conditions for a unique Nash Equilibrium where the master obtains the correct result. We also evaluate experimentally mixed equilibria aiming to attain better reliability-profit trade-offs. For a wide range of parameter values that may be used in practice, our simulations show that, in fact, both master and workers are better off using a pure equilibrium where no worker cheats, even under collusion, and even for colluding behaviors that involve deviating from the game.

Introduction

Motivation and prior work

The need for high-performance computing, the partial availability of resources in personal computers (such as CPU and GPU), and the wide access to the Internet, have led to the development of Internet-based computing. Internet-based computing is mostly embraced by the scientific community in the form of volunteer computing. That is, a system where participants contribute their idle computing resources to solve scientific problems. Among the most popular volunteering projects is SETI@home [1] running on the BOINC [2] platform. Computing platforms where users contribute for profit also exist, for example, Amazon's Mechanical Turk [3]. Unfortunately, although the potential of such systems is great, the use of Internet-based computing is constrained by the untrustworthy nature of the platform's components [2, 4].

In Internet-based Master-Worker task computing systems a *master* process sends tasks, across the Internet, to *worker* processes. Workers are expected to execute and report back the result. However, these workers are not trustworthy, hence, they might report incorrect results [2, 5, 6]. Prior work has considered different approaches in tackling this shortcoming. A "classical" Distributed Computing approach is to model malfunctioning (hardware or software

grants from the National Science Foundation (http://www.nsf.gov) (CCF 1114930) and from Kean University (http://www.kean.edu) (UFRI grant). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

errors) or cheating (intentional wrongdoing) as *malice*. That is, it is assumed the presence of some workers that wish to hamper the computation returning always an incorrect result. On the other hand, non-malicious workers are modeled as *altruism*. That is, it is assumed the presence of other workers who always return the correct result. Under this classical model, malice-tolerant protocols have been considered [7–9] where the master takes the result returned by the majority of workers. A Game-theoretic approach to deal with untrustworthiness is to assume that workers are *rational* [4, 10, 11]. That is, based on the strategy that best serves its self-interest, a worker decides whether to compute and return the correct result or return a bogus result. Under this game-theoretic model, incentive-based algorithmic mechanisms have been devised [12–14]. These mechanisms employ reward/punishment schemes to "enforce" rational workers to act correctly. More recent works [15, 16] have combined these two approaches and devised mechanisms assuming the co-existence of altruistic, malicious and rational workers. For a discussion on the connection between Game Theory and (distributed) computing we refer the reader to the book by Nisan et al. [17] and the survey by Abraham et al. [18].

With respect to collusion, in [19, 20] workers are assumed to have a predefined behavior (they are always faulty or always correct). In both papers, the goal is to identify colluders statistically, which requires multiple rounds of worker computation. Rather than focusing on colluder detection, in the present paper, we design mechanisms that yield the correct task computation, *despite* collusions. Furthermore, in our work each worker computes a task at most once. The benefit of one-round per task mechanisms is partially supported by the work of Kondo et al. [21]. In such work it was demonstrated experimentally that, in master-worker computations, tasks may take much more than one day of CPU time to complete.

In a work directly related to the present paper [14], master-worker computations with rational colluding workers are also studied. In their model, the master can audit the results returned by rational workers with a tunable probability. Bounds on the audit probability that guarantee that workers have incentives to be honest are shown. The study is carried out for three scenarios: redundant allocation of tasks with and without collusion, and single-worker allocation. They conclude that, for their model, single-worker allocation is a cost-effective mechanism, specially in presence of collusion. The form of collusion considered is close to ours: within the same group, either all processors cheat or are honest. However, our analysis and simulations provide a much deeper understanding of the impact of collusions on the master-worker problem. Furthermore, our work considers a richer payoff model and probabilistic cheating, yielding interesting tradeoffs between the utility of the master and the probability of obtaining an incorrect result.

To the best of our knowledge, with the exception of the works discussed above, workers have been always assumed to interact only with the master. That is, no explicit worker collusion that models interaction among workers is considered. (In some works, it is assumed that cheating workers would return the same incorrect answer, without them interacting with each other. This is an implicit and weak form of collusion assumed as a worst case for a voting-based mechanism, as workers returning different incorrect answers would less likely reach a majority.) However, one cannot exclude the possibility of collusion, for example, as in Sybil attacks [22]. In this paper, we study the impact of collusion and the tradeoffs involved in Internet-based Master-Worker task computing, under the assumption that workers are rational in a game-theoretic sense.

Other related work in the area of cloud service and cloud computing includes [23, 24]. Although the overall setting and methodologies considered are different from ours, their objective is similar to ours: increase the trustworthiness of the system.

Contributions

Our main contributions are as follows:

1. We model the Master-Worker Internet-based task computation as a game among rational workers under collusion (cf. Section 2). Each worker chooses whether to be *honest* (i.e., compute and return the correct task result) or a *cheater* (i.e., fabricate a bogus result and return it to the master) based on which of the two strategies increases its *utility* (benefit). In addition, we consider worker *collusion*: collections of workers form groups. Within each group, workers decide on a common strategy that would provide greater expected benefit. Neither the master nor other workers outside of a colluding group are aware of the collusion. The model is enhanced with a collection of realistic payoff parameters to quantify the benefit of the master and the workers. These parameters can either be fixed because they are system parameters or be chosen by the master.

2. Under this model, we design algorithmic mechanisms that provide the necessary incentives for the workers to truthfully compute and report the correct result, despite collusion. The objective of the design is twofold: maximize the probability of obtaining the correct result (reliability) and maximize the master utility (maximizing profit or minimizing cost if profit is not possible). In short, the mechanism works as follows (cf. Section 4): The master assigns the task to n workers. Each worker i cheats with probability $p_C^{(i)}$ and the master verifies the answers with some probability p_V (verification comes at a cost–see Section 2). If the master verifies, it rewards the honest workers and penalizes the cheaters. If the master does not verify, it accepts the answer returned by the majority of workers and rewards these workers only. However, it does not penalize the workers in the minority, given that the majority may be actually cheating. Workers may collude in which case the workers belonging to the same group decide homogeneously: either all cheat or all are honest (for more details on collusion see Section 2).

3. The workers' decision of whether to cheat or not is modeled as a game. As typical in algorithmic mechanism design [10, 25], the master induces a *Nash Equilibrium* (NE) [26]. When this NE is followed by the workers, the outcome has the desired properties (reliability and utility). We analyze the game and identify the conditions under which unique NE is achieved (cf Section 3). The reason for uniqueness is to force all workers to the same strategy (this is similar to *strong implementation* in Mechanism Design [27]). Each unique NE results in a different benefit for the master and a different probability of accepting an incorrect result. Thus, the master can choose the game conditions for the unique NE that best fits its goals.

4. To demonstrate the practicality of the analysis, we design mechanisms for two specific realistic scenarios. These scenarios reflect, in their fundamental elements, (a) a system of volunteer computing (e.g., SETI), and (b) a company that buys computing cycles from participant computers and sells them to its customers in the form of a task-computation service (e.g., Amazon's Mechanical Turk). The analysis provided for these scenarios comprises implicitly a mechanism to decide how to carry out the computation. Our analysis reveals interesting tradeoffs between reliability and utility (see Section 4). A general conclusion drawn from the analysis with respect to worker collusion is that, in order to guarantee a unique equilibrium for any parameter values, all groups, colluding or singletons, must decide whether to cheat or not deterministically.

5. We carry out extensive simulations of our mechanisms for the contractor scenario in an attempt to better understand the impact of worker collusion (cf Section 5). Specifically our simulations are designed to investigate whether considering *mixed* equilibria yields better tradeoffs for the master and/or the workers, and whether the workers would benefit more if they did not follow the NE induced by the master (i.e., deviate from the game). For a wide range of parameter values (that may be used in practice), our simulations show that all parties, the master and the workers, are better-off using a pure equilibrium where no worker cheats, *even under collusion*, and even for colluding behaviors that involve deviating from the game.

Model and Definitions

Framework

Setting: We consider a distributed system consisting of a master processor that assigns a computational task to a set of workers to compute and return the result. The tasks considered in this work are assumed to have a unique solution. Although such limitation reduces the scope of application of the mechanisms presented [28], there are plenty of computations where the correct solution is unique, e.g., any mathematical function.

Master verification and worker rationality: It is assumed that the master has the possibility of *verifying* whether the value returned by a worker is the correct result of the task. It is also assumed that verifying an answer is computationally easier than computing the task [29] (e.g., numerical computations), but the correct result of the computation is not obtained if the verification fails (e.g., when all workers cheat). (Alternatively, one might assume that the master verifies by simply performing the task and checking the answers of the workers. Our analysis can easily be modified to accommodate this different model.) As in [12–14], workers are assumed to be *rational* and seek to maximize their benefit, i.e., they are not destructively malicious. We note that this assumption can conceptually be justified by the work of Shneidman and Parkes [11] where they reason on the connection of rational players (of Algorithmic Mechanism Design) and workers in realistic P2P systems. Furthermore, we do not consider unintentional errors produced by hardware or software problems. As already mentioned, the master verifies with probability p_V, and the each worker i cheats with probability $p_C^{(i)}$.

Collusion: We consider a form of collusion that covers realistic types such as Sybil attacks [22]). Within any given colluding group workers act homogeneously, i.e., either all choose to cheat, or all choose to be honest. The decision is perhaps randomized tossing a unique coin. In the case that a colluding group chooses to be honest, then only one of the workers computes the task (saving computing cost). In the case that that a colluding group chooses to cheat, then they simply agree on a bogus result. In either case they return the agreed result to the master. In addition, we assume that all "cheating groups" return the same incorrect answer if there is more than one. Both assumptions, homogeneous behavior within groups and unique incorrect answer, are adversarial. Since the master accepts the majority, this colluding strategy maximizes the chances of cheating the master. Being this the worst case (see also [9]), it subsumes models where cheaters do not necessarily return the same answer. We assume that, if a worker does not compute the task, the probability of guessing the correct answer is negligible. We also assume that the overall number of colluders is not a majority. The assumption comes naturally from the fact that the set of workers is a choice of the master, who can select arbitrarily from a large population making the likelihood of choosing many related workers low, specially for a one-shot computation. With respect to the master, we assume that it is not aware of which workers are colluding, although it is aware that collusion may occur.

Game definition

We now define formally a game "played" by the workers in a game theoretic sense. We summarize the notation in Table 1 for easy reference. We assume that the master hires an odd number of workers $n \geq 3$ to be able to apply a voting strategy and to avoid ties. In order to model collusion among workers, we view the set of workers as a set of non-empty subsets $W = \{W_1, \ldots, W_\ell\}$ such that $\sum_{i=1}^{\ell} | W_i | = n$ and $W_i \cap W_j = \emptyset$ for all $i \neq j$, $1 \leq i, j \leq \ell$. We refer to each of these subsets as a *group of workers* or a *group* for short. We also refer to groups as *players*. Workers acting individually are modeled as a group of size one. It is assumed that the size of

Table 1. Game notation.

$W = \{W_1, \ldots, W_\ell\}$	set of worker groups
$S_i = \{C, C\}$	set of pure strategies (cheat/not-cheat) available to group W_i
s	strategy profile (a mapping from players to pure strategies)
s_i	strategy used by group W_i in the strategy profile s
s_{-i}	strategy used by each player but W_i in the strategy profile s
$w_s^{(i)}$	payoff of group W_i for the strategy profile s
σ	mixed strategy profile (a mapping from players to prob. distrib. over pure strategies)
σ_i	probability distribution over pure strategies used by group W_i in σ
σ_{-i}	probability distribution over pure strategies used by each player but W_i in σ
$p_{s_i}^{(i)}$	probability that group W_i uses strategy s_i
$supp(\sigma_i)$	set of strategies of group W_i with probability > 0 (called support) in σ
$U_i(s_i, \sigma_{-i})$	expected utility of group W_i with mixed strategy profile σ

doi:10.1371/journal.pone.0116520.t001

each group is known only to the members of the group. Other than that, we assume that workers have complete information on the algorithm and the parameters involved.

For succinctness, we express a strategy profile as a collection of individual strategy choices together with collective strategy choices. For instance, $s_i = C, R_{-i}, F_{-i}, T_{-i}$ stands for a strategy profile s where group W_i chooses strategy C (to cheat), a set R_{-i} of groups (where group W_i is not included) randomize their strategy choice with probability $p_C \in (0, 1)$, a set F_{-i} of groups deterministically choose strategy C, and a set T_{-i} of groups deterministically choose strategy $\bar{C}$ (to be honest). We require that, for each group W_i, $p_C^{(i)} = 1 - p_{\bar{C}}^{(i)}$. Whenever all groups use the same probability, we drop the superindex (i) for clarity. Also, whenever the strategy is clear from context, we refer to the expected utility of group W_i as U_i.

Equilibrium definition

We define now the conditions for the equilibrium. In the context of collusion, the probability distributions are not independent among members of a group. Furthermore, the formulation of equilibrium conditions among individual workers would violate the very definition of equilibrium since the choice of a worker does change the choices of other workers. Instead, equilibrium conditions are formulated among groups. Of course, the computation of an equilibrium might not be possible since the size of the groups is unknown. But, finding appropriate conditions so that the unique equilibrium is the same independently of that size, the problem may be solved. As we will see in the analysis, knowing some bound (e.g. the trivial one) on the size of the smallest and/or largest group is enough. Moreover, as we will see in simulations, deviating from this expected behavior is against workers' interest in practice. It is important to notice that although the majority is evaluated in terms of number of single answers, this fact has no impact on correctness of the equilibrium formulation.

We recall [30] that, for any finite game, a mixed strategy profile σ^* is a *mixed-strategy Nash equilibrium* (MSNE) if, and only if, for each player π,

$$U_\pi(s_\pi, \sigma^*_{-\pi}) = U_\pi(s'_\pi, \sigma^*_{-\pi}), \forall s_\pi, s'_\pi \in supp(\sigma^*_\pi), \tag{1}$$

$$U_\pi(s_\pi, \sigma^*_{-\pi}) \geq U_\pi(s'_\pi, \sigma^*_{-\pi}), \\ \forall s_\pi, s'_\pi : s_\pi \in supp(\sigma^*_\pi), s'_\pi \notin supp(\sigma^*_\pi). \tag{2}$$

Where σ^*_π is the probability distribution over pure strategies used by π in σ^*, $\sigma^*_{-\pi}$ is the probability distribution over pure strategies used by each player but π in σ^*, $supp(\sigma^*_\pi)$ is the set of

Table 2. Master's and Workers' Payoffs.

WP_C	worker's punishment for being caught cheating
WC_T	group's cost for computing the task
WB_A	worker's benefit from master's acceptance
MP_W	master's punishment for accepting a wrong answer
MC_A	master's cost for accepting the worker's answer
MC_V	master's cost for verifying worker's answers
MB_R	master's benefit from accepting the right answer

doi:10.1371/journal.pone.0116520.t002

strategies of π with probability > 0 (called support) in σ^*, and $U_\pi(s_\pi, \sigma^*_{-\pi})$ is the expected utility of π in σ^*.

In words, given a MSNE with mixed-strategy profile σ^*, for each player π, the expected utility, assuming that all other players do not change their choice, is the same for each pure strategy that the player can choose with positive probability in σ^*, and it is not less than the expected utility of any pure strategy with probability zero of being chosen in σ^*. A *fully* MSNE is an equilibrium with mixed strategy profile σ where, for each player π, $supp(\sigma_\pi) = S_\pi$, where S_π is the whole set of pure strategies available to π.

Payoffs definition

We detail in Table 2 the workers' payoffs. We also include master payoffs to evaluate its utility. All the parameters in this table are non-negative.

Notice that we split the reward to a worker into WB_A and MC_A, to model the fact that the cost of the master might be different than the benefit of a worker. In fact, in some models they may be completely unrelated. Among the parameters involved, we assume that the master has the freedom of choosing the cheater penalty WP_C and the worker's reward WB_A. By tuning these parameters and choosing n, the master achieves the desired trade-off between reliability and utility. Recall that the master does not know the composition of groups (if there is any). Hence, benefits and punishments are applied individually to each worker, except for the cost for computing the task WC_T which is shared among all workers in the same group. Sharing the cost of computing while being paid/punished individually may provide incentive to collude, but it models the real-world situation where collusion is secret.

Equilibria Analysis

From Eqs. (1) and (2), we obtain conditions on payoffs and probabilities to attain a unique NE. A similar analysis was presented in [13] for various games and reward models, but for a model *without collusion*. The analysis presented here can be also applied to those models and games where collusion may appear. We consider a game where the master assigns a computing task to n workers that "play" (in a game-theoretic sense) among them. Intuitively, it is to the workers' advantage to be in the majority in case the master does not verify and the majority is rewarded. Given that workers know the existence of the other workers, including collusions in the analysis is in order.

For clarity, we partition the set of groups as $\{F, T, R\}$, where $F \cup T \cup R = W$. F is the set of groups that choose to cheat as a pure strategy, that is, $F = \{W_i \mid W_i \in W \wedge p_C^{(i)} = 1\}$. T is the set of groups that choose not to cheat as a pure strategy, that is, $T = \{W_i \mid W_i \in W \wedge p_C^{(i)} = 0\}$. R is the set of groups that randomize their choice, that is, $R = \{W_i \mid W_i \in W \wedge p_C^{(i)} \in (0, 1)\}$. Let $F_{-i} = F \setminus \{W_i\}$, $T_{-i} = T \setminus \{W_i\}$, and $R_{-i} = R \setminus \{W_i\}$. Let

Γ_{-i} be the set of partitions in two subsets (R_F, R_T) of R_{-i}, i.e., $\Gamma_{-i} = \{(R_F, R_T)|R_F \cap R_T = \emptyset \wedge R_F \cup R_T = R_{-i}\}$. Let $\mathbf{E}[w_s^{(i)}]$ be the expected payoff of group W_i for the strategy profile s, taking the expectation over the choice of the master of verifying or not. Then, for each group $W_i \in W$ and for each strategy profile $s_{-i} = R_{-i}, F_{-i}, T_{-i}$, we have that $U_i(s_{-i}, s_i = C) = \sum_{(R_F,R_T)\in\Gamma_{-i}} \prod_{W_f\in R_F} p_C^{(f)} \prod_{W_t\in R_T}(1-p_C^{(t)})\mathbf{E}[w_{s'}^{(i)}]$, where s' is the strategy profile $F_{-i} \cup R_F$, $T_{-i} \cup R_T$, $s_i = C$; and that $U_i(s_{-i}, s_i = \bar{C}) = \sum_{(R_F,R_T)\in\Gamma_{-i}} \prod_{W_f\in R_F} p_C^{(f)} \prod_{W_t\in R_T}(1-p_C^{(t)})\mathbf{E}[w_{s''}^{(i)}]$, where s'' is the strategy profile $F_{-i} \cup R_F, T_{-i} \cup R_T, s_i = \bar{C}$.

In order to find conditions for a desired equilibrium, we study what we call the *utility differential* of a worker, i.e. the difference on the expected utility of a worker if its group chooses to cheat with respect to the case when the group chooses to be honest. That is, $\Delta U_i(s) = U_i(s_{-i}, s_i = C) - U_i(s_{-i}, s_i = \bar{C})$.

For clarity, define $N_{F-i} = \sum_{S \in F_{-i}\cup RF} |S|$ and $N_{T-i} = \sum_{S \in T_{-i}\cup RT} |S|$, i.e. the number of cheaters and honest workers respectively except for those in group W_i. We also define what we call the *payoff differential* as the difference on the expected payoff of a worker, the expectation taken over the choice of the master, if its group chooses to cheat with respect to the case when the group chooses to be honest. Furthermore, we denote the payoff differential depending on whether the size of the group has an impact on what is the majority outcome. More precisely, for each partition $(R_F, R_T) \in \Gamma_i$, let $\Delta w_C^{(i)} = \mathbf{E}[w_{s_i=C}^{(i)}] - \mathbf{E}[w_{s_i=\bar{C}}^{(i)}]$, when $N_{F-i} - N_{T-i} > |W_i|$, $\Delta w_{\bar{C}}^{(i)} = \mathbf{E}[w_{s_i=C}^{(i)}] - \mathbf{E}[w_{s_i=\bar{C}}^{(i)}]$, when $N_{T-i} - N_{F-i} > |W_i|$, and $\Delta w_X^{(i)} = \mathbf{E}[w_{s_i=C}^{(i)}] - \mathbf{E}[w_{s_i=\bar{C}}^{(i)}]$, when $|N_{F-i} - N_{T-i}| < |W_i|$.

Given that the payoff depends only on the outcome majority, replacing this notation in the utility differential, we have that $\Delta U_i(s)$ is

$$\begin{aligned}
&\Delta w_C^{(i)} \sum_{\substack{(R_F,R_T)\in\Gamma_{-i} \\ N_{F-i}-N_{T-i}>|W_i|}} \prod_{W_f\in R_F} p_C^{(f)} \prod_{W_t\in R_T} (1-p_C^{(t)}) + \\
&\Delta w_X^{(i)} \sum_{\substack{(R_F,R_T)\in\Gamma_{-i} \\ |N_{F-i}-N_{T-i}|<|W_i|}} \prod_{W_f\in R_F} p_C^{(f)} \prod_{W_t\in R_T} (1-p_C^{(t)}) + \\
&\Delta w_{\bar{C}}^{(i)} \sum_{\substack{(R_F,R_T)\in\Gamma_{-i} \\ N_{T-i}-N_{F-i}>|W_i|}} \prod_{W_f\in R_F} p_C^{(f)} \prod_{W_t\in R_T} (1-p_C^{(t)}). \qquad (3)
\end{aligned}$$

Restating Eqs. (1) or (2) in terms of Eq. (3), the equilibrium conditions are, for each group that does not choose a pure strategy, the differential utility must be zero ($\forall i \in R, \Delta U_i(s) = 0$); for each group that chooses to cheat as a pure strategy, the differential utility must not be negative ($\forall i \in F, \Delta U_i(s) \geq 0$); and for each group that chooses to be honest as a pure strategy, the differential utility must not be positive ($\forall i \in T, \Delta U_i(s) \leq 0$).

The following lemma, which is crucially used in the rest of our analysis, shows that, if there is a given total order among the payoff differentials defined, in order to attain a unique equilibrium all groups must decide deterministically. The proof is based on an algebraic argument.

Lemma 1. *Given a game as defined, if $\Delta w_C^{(i)} \geq \Delta w_X^{(i)} \geq \Delta w_{\bar{C}}^{(i)}$ for every group $W_i \in W$, then there is no unique equilibrium where $R \neq \emptyset$ (i.e, all groups decide deterministically).*

Proof. For the sake of contradiction, assume there is a unique equilibrium σ for which $R \neq \emptyset$ and $\Delta w_C^{(i)} \geq \Delta w_X^{(i)} \geq \Delta w_{\bar{C}}^{(i)}$ for every group $W_i \in W$. Then, for every group $W_i \in R$, $\Delta U_i(s) = 0$ must be solvable. If $\Delta w_C^{(i)} \geq 0$, for all $W_i \in R$, there would be also an equilibrium where all groups in R choose to cheat and σ would not be unique, which is a contradiction. Consider now the case where there exists some $W_i \in R$ such that $\Delta w_C^{(i)} < 0$. Then, it must hold that $|R| > 1$, otherwise $\Delta U_i = 0$ is false for W_i. Given that $|R| > 1$, the probabilities given by the summations in Equation (3) for W_i are all strictly bigger than zero. Therefore, given that $\Delta U_i = 0$ must be solvable, at least one of $\Delta w_X^{(i)} > 0$ and $\Delta w_{\bar{C}}^{(i)} > 0$ must hold, which is also a contradiction with the assumption that $\Delta w_C^{(i)} \geq \Delta w_X^{(i)} \geq \Delta w_{\bar{C}}^{(i)}$.

Replacing appropriately the payoffs detailed in Table 2, we obtain for any group $W_i \in W$

$$\Delta w_C^{(i)} = -p_V \mid W_i \mid (WP_C + 2WB_A) + \mid W_i \mid WB_A + WC_T, \tag{4}$$

$$\Delta w_X^{(i)} = -p_V \mid W_i \mid (WP_C + WB_A) + WC_T, \tag{5}$$

$$\Delta w_{\bar{C}}^{(i)} = -p_V \mid W_i \mid WP_C - \mid W_i \mid WB_A + WC_T. \tag{6}$$

In the equations above, it holds that $\Delta w_C^{(i)} \geq \Delta w_X^{(i)} \geq \Delta w_{\bar{C}}^{(i)}$ for all $W_i \in W$. Then, by Lemma 1, there is no unique equilibrium where $R \neq \emptyset$. Regarding equilibria where $R = \emptyset$, unless the task assigned has a binary output (i.e. the answer can be negated), a unique equilibrium where all groups choose to cheat is not useful to the master. Then, we set up p_V so that $\Delta w_C^{(i)} < 0$, $\Delta w_X^{(i)} < 0$ and $\Delta w_{\bar{C}}^{(i)} < 0$ for all $W_i \in W$ so that $\Delta U_i \geq 0$ has no solution and no group can choose to cheat as a pure strategy. Thus, the only equilibrium is for all the groups to choose to be honest, which solves $\Delta U_i \leq 0$. Therefore, $p_C^{(i)} = 0, \forall W_i \in W$ and, hence, the probability that the master accepts a wrong answer is $\mathbf{P}_{wrong} = 0$.

To make $\Delta w_C^{(i)} < 0$ we need $p_V > (|W_i|WB_A + WC_T)/(|W_i|(WP_C + 2WB_A))$, for all $W_i \in W$. Then, the expected utilities are $U_M = MB_R - p_V MC_V - nMC_A$ and $U_{W_i} = |W_i|WB_A - WC_T$, for each $W_i \in W$. In order to maximize the master utility we would like to design games where p_V is small. Therefore, we look for a lower bound on p_V. It can be seen using calculus that the largest lower bound is given by the group of minimum size. Although at a first glance this fact seems counterintuitive, it is not surprising due to the following two reasons. On one hand, colluders are likely to be in the majority, but the unique equilibrium occurs when all workers are honest. On the other hand, the extra benefit that workers obtain by colluding is not against the master interest since it is just a saving in computation costs.

Algorithmic Mechanisms

In this section two realistic scenarios in which the master-worker model considered could be naturally applicable are proposed. For these scenarios, we determine appropriate parameters to be used by the master to obtain the correct answer and maximize its benefit.

The basic protocol (mechanism) used by the master is as follows: Given the payoff parameters (these can either be fixed by the system or be chosen by the master), the master sends to the workers the task to be computed and informs the probability of verification p_V. For computational reasons, the master also provides a certificate to the workers. This certificate includes the strategy that the workers must play to achieve the unique NE, together with the appropriate data to demonstrate this fact. (The certificate is included only for cases where resource limitations preclude the worker from computing the unique equilibrium, but it is not related to distributions over public signals as in a correlated equilibrium, since workers do not randomize their choice according to this certificate.)

After receiving the replies from all workers, and independently of the distribution of the answers, the master processor chooses to verify the answers with probability p_V. If the answers were not verified it accepts the result of the majority, and it rewards only those workers,

Table 3. Master algorithm.

```
1:  Procedure
2:    send (task, $p_V$, certificate) to all workers
3:    upon receiving all answers do
4:      verify the answers with probability $p_V$
5:      if the answers were not verified then
6:        accept the majority
7:      end if
8:      reward/penalize accordingly
9:    end upon
10: end procedure
```

doi:10.1371/journal.pone.0116520.t003

whereas if the answers were verified it rewards the correct workers and penalizes the cheaters. The protocol is detailed in Table 3.

The master, given the payoff parameters, can determine the other parameters, including the value of p_V, to force the workers into a unique NE, that would yield the correct task result while maximizing the master's benefit. Examples of specific parameters such that the master can achieve this are analyzed next.

SETI-like scenario

The first scenario considered is a volunteer computing system such as SETI@home, where users accept to donate part of their processors idle time to collaborate in the computation of large tasks. In this case, we assume that workers incur in no cost to perform the task, but they obtain a benefit by being recognized as having performed it (possibly in the form of prestige, e. g, by being included on SETI's top contributors list). In this context, we assume that $WB_A > WC_T = 0$. The master incurs in a (possibly small) cost MC_A when rewarding a worker (e.g., by advertising its participation in the project). As assumed in the general model, in this model the master may verify the values returned by the workers, at a cost $MC_V > 0$. We also assume that the master obtains a benefit $MB_R > MC_A$ if it accepts the correct result of the task, and suffers a cost $MP_W > MC_V$ if it accepts an incorrect value.

Under these constraints, replacing in the equations of Section 3, we obtain a unique equilibrium at $p_C = 0$, that can be enforced making $p_V > WB_A/(WP_C + 2WB_A)$, attaining $\mathbf{P}_{wrong} = 0$, $U_M = MB_R - p_V MC_V - nMC_A$, and $U_{W_i} = |W_i| WB_A$. Given that in this scenario p_V does not depend on n, it is better for the master to allocate the task to minimum number of workers ($n = 3$), maximizing its utility. We highlight this observation in the following theorem.

Theorem 2. *For any set of payoff parameters that can be characterized as the SETI scenario, in order to obtain the correct answer (with probability* 1*) while maximizing master's utility, it is enough to assign the task to only three workers, and verify with probability* $p_V > WB_A/(WP_C + 2WB_A)$.

Contractor scenario

The second scenario considered is a company that buys computational power from Internet users and sells it to computation-hungry costumers. In this case the company pays the workers an amount $WB_A = MC_A$ for using their computing capabilities, and charges the consumers another amount $MB_R > MC_A$ for the provided service. Since the workers are not volunteers in this scenario, we assume that computing a task is not free for them ($WC_T > 0$), and they must have incentives to participate ($U_{W_i} > 0, \forall W_i \in W$). The worker payment then must be at least the cost of computing ($WB_A \geq WC_T$). As in the previous scenario, we assume that the master verifies and has a cost for accepting a wrong value, such that $MP_W > MC_V > 0$.

Under these constraints, replacing in the equations of Section 3, we obtain a unique equilibrium at $p_C = 0$, that can be enforced making $p_V > (|W_i| WB_A + WC_T)/(|W_i|(WP_C + 2WB_A))$, $\forall W_i \in W$. Because the latter is decreasing on $|W_i|$, it is enough (a larger lower bound on $\min_i |W_i|$ could be used if available) to make $p_V > (WB_A + WC_T)/(WP_C + 2WB_A)$, attaining $\mathbf{P}_{wrong} = 0$, $U_M = MB_R - p_V MC_V - nMC_A$, and $U_{W_i} = |W_i| WB_A - WC_T$.

In this scenario, in addition to choosing the number of workers n, we assume that the master can also choose the reward WB_A and the punishment WP_C. We focus in the analysis on making only one of these parameters variable at a time. More combinations will be considered in simulations.

Tunable *n:* The utility of the master $U_M = MB_R - p_V MC_V - nMC_A$ increases as n and p_V decrease. Hence, if the number of workers is a choice, using the minimum number of workers ($n = 3$) and the minimum p_V to achieve correctness maximizes the utility. We highlight this observation in the following theorem.

Theorem 3. *For any given set of payoff parameters, such that it can be characterized as the contractor scenario, if the master gets to choose the number of workers, in order to obtain the correct answer (with probability* 1*) while maximizing the utility of the master, it is enough to verify with probability* $p_V = (WB_A + WC_T)/(WP_C + 2WB_A) + \epsilon$, *for arbitrarily small* $\epsilon > 0$, *assign the task to only three workers.*

Tunable *WP_C:* Consider the master utility $U_M = MB_R - p_V MC_V - nMC_A$. That is, $U_M = MB_R - (((WB_A + WC_T)/(WP_C + 2WB_A)) + \epsilon) MC_V - nMC_A$. It can be seen that the master utility increases with WP_C. However, a large WP_C may discourage workers to join the computation. On the other hand, a large WP_C may alleviate the impact of a large cost of verification MC_V. A trade-off between these factors must be taken into account in practice. Nevertheless, this is independent of achieving the equilibrium, as long as the appropriate p_V is used. We highlight these observations in the following theorem.

Theorem 4. *For any given sets of workers and payoff parameters, except for* WP_C *that is chosen by the master. If the set of payoffs is such that* $MC_A = WB_A > WC_T$ *and it can be characterized as the contractor scenario, in order to obtain the correct answer (with probability* 1*) while maximizing the utility of the master, it is enough to set* WP_C *as large as possible, verify with probability* $p_V = (WB_A + WC_T)/(WP_C + 2WB_A) + \epsilon$, *for arbitrarily small* $\epsilon > 0$.

Tunable *WB_A:* Using calculus, it can be seen that, if $WP_C > 2WC_T$ (resp. $2WC_T > WP_C$), U_M is decreasing (resp. increasing) on WB_A. And if on the other hand $WP_C = 2WC_T$, U_M is constant with respect to WB_A. For the parameter combinations that make the master utility not constant, because WB_A is limited as $WC_T \leq WB_A < MB_R$, we have that if $WP_C > 2WC_T$ (resp. $2WC_T > WP_C$), the utility is maximized when $WB_A = WC_T$ (resp. $WB_A = MB_R - \epsilon$, for $\epsilon > 0$). Again, this is independent of achieving the equilibrium, as long as the appropriate p_V is used. We highlight these observations in the following theorem.

Theorem 5. *For any given sets of workers and payoff parameters, except for* $WB_A = MC_A$ *that is chosen by the master. If the set of payoffs can be characterized as the contractor scenario, in order to obtain the correct answer (with probability* 1*) while maximizing the utility of the master, it is enough to set* $WB_A = WC_T$ *if* $WP_C > 2WC_T$ *or* $WB_A = MB_R - \epsilon_1$ *otherwise, for arbitrarily small* $\epsilon_1 > 0$, *verify with probability* $p_V = (WB_A + WC_T)/(WP_C + 2WB_A) + \epsilon_2$, *for arbitrarily small* $\epsilon_2 > 0$.

Simulations

Simulations design

As shown in Section 3, in order to guarantee a unique equilibrium for any parameter values, all groups must decide whether to cheat or not deterministically (Lemma 1). The reason is algebraic. (Refer to Eq. 3.) If the payoff differentials $\Delta w_C^{(i)} \geq \Delta w_X^{(i)} \geq \Delta w_{\bar{C}}^{(i)}$ are all positive (resp. negative) the worker-utility differential $\Delta U_i(s)$ is positive (resp. negative) and all workers cheat (resp. are honest) deterministically. But if not all three differentials have the same sign, there could be many values of p_C that make $\Delta U_i(s) = 0$. That is, there could be multiple mixed equilibria. Since $\Delta w_C^{(i)} \geq \Delta w_X^{(i)} \geq \Delta w_{\bar{C}}^{(i)}$, to avoid having the same sign in all three differentials, it is enough to make $\Delta w_C^{(i)} > 0$ and $\Delta w_{\bar{C}}^{(i)} < 0$ for each group $W_i \in W$. That is, from Eqs. 4 and 6, obtain a p_V that satisfies $(WC_T - |W_i|WB_A)/(|W_i|WP_C) < p_V < (|W_i|WB_A + WC_T)/(|W_i|(WP_C + 2WB_A))$. And given that $|W_i| \geq 1$ and $p_V \geq 0$, we get

$$p_V < \min_{W_i \in W} \frac{|W_i| WB_A + WC_T}{|W_i|(WP_C + 2WB_A)}. \tag{7}$$

Then, plugging the payoff differentials from Eqs. 4 to 6 in Eq. 3, up to $n - 1$ roots may result from making $\Delta U_i = 0$.

Aiming for a mixed equilibrium is promising because it might yield a cost reduction for the master on verifications. However, these multiple equilibria cannot be computed without the knowledge of *all* group sizes, because the computation of the probability of each payoff differential takes into account that each group tosses only one coin (see Eq. 3). Given that only the members of a colluding group know their group size, the computation is not feasible. In fact, although the master is aware of the possible presence of colluders, and the mechanism must take measures against this deviated behavior, the expected worker behavior is not to collude. Hence, the master only has to compute the equilibria for a platform without collusion, and provide these results to workers. That is, fix a p_V restricted to Eq. (7) for $|W_i| = 1$, and compute the equilibria as the roots of the following polynomial.

$$\begin{aligned}
&\Delta w_C \sum_{j=0}^{\lfloor n/2 \rfloor - 1} \binom{n-1}{j} p_C^{n-1-j} (1-p_C)^j + \\
&\Delta w_X \binom{n-1}{\lfloor n/2 \rfloor} p_C^{\lfloor n/2 \rfloor} (1-p_C)^{\lfloor n/2 \rfloor} + \\
&\Delta w_{\bar{C}} \sum_{j=\lceil n/2 \rceil}^{n-1} \binom{n-1}{j} p_C^{n-1-j} (1-p_C)^j = 0.
\end{aligned}$$

Where

$$\begin{aligned}
\Delta w_C &= -p_V (WP_C + 2WB_A) + WB_A + WC_T, \\
\Delta w_X &= -p_V (WP_C + WB_A) + WC_T, \text{ and} \\
\Delta w_{\bar{C}} &= -p_V WP_C - WB_A + WC_T.
\end{aligned}$$

Then, the colluding groups may use these equilibria, but still take advantage of collusion in the voting outcome, and by sharing the cost of computing when they are honest. We call this the ***compliant behavior***.

Knowing that they will act together, each colluding group could still ignore the master-provided equilibria and decide their own game strategy taking into account their group size. The members of a group do not know the size of other groups, or even their existence. Hence, they can assume nothing but that no worker outside the group colludes. Using this information, the colluding group could compute Nash equilibria, but they cannot enforce the use of such equilibria to other workers. So, they are left only with the choice of using the master-provided equilibria, or drop the aim of achieving equilibrium at all. For instance, a group of colluders may just decide whether to cheat or not deterministically to maximize their expected utility, under the assumption that all other workers are compliant. We call this the ***disobedient behavior***. This scenario can be seen as further deviation (with respect to compliant but voting together) from the expected behavior. The utility differential $\Delta U_i(s)$ (Eq. 3) for a group W_i would be the following.

$$\Delta w_C^{(i)} \sum_{j=0}^{\lfloor n/2\rfloor-|W_i|} \binom{n-\mid W_i\mid}{j} p_C^{n-|W_i|-j}(1-p_C)^j +$$

$$\Delta w_X^{(i)} \sum_{j=\lceil n/2\rceil-|W_i|}^{\lfloor n/2\rfloor} \binom{n-\mid W_i\mid}{j} p_C^{n-|W_i|-j}(1-p_C)^j +$$

$$\Delta w_{\bar{C}}^{(i)} \sum_{j=\lceil n/2\rceil}^{n-|W_i|} \binom{n-\mid W_i\mid}{j} p_C^{n-|W_i|-j}(1-p_C)^j .$$

Where $\Delta w_C^{(i)}$, $\Delta w_X^{(i)}$, and $\Delta w_{\bar{C}}^{(i)}$ are as defined in Eqs. 4 to 6, and p_C corresponds to the equilibria computed by the master (more on how to choose, if many values for p_C are possible, later). Recall that $\Delta U_i(s) = U_i(s_{-i}, s_i = C) - U_i(s_{-i}, s_i = \bar{C})$. Then, a colluding group decides to cheat (resp. be honest) if this utility differential is positive (resp. negative).

Finally, it would be interesting to study the impact of having each colluding group choosing an arbitrary p_C at random from (0, 1). We call this the ***random behavior***.

We carry out simulations for all three behaviors defined, setting up p_V so that the game may have multiple mixed equilibria. For the sake of contrast, we also carry out simulations setting up p_V as in the analysis to have a unique pure equilibrium in $p_C = 0$. The parameter combinations used are shown in Table 4. We set $MB_R = MP_W = MC_V = 100$, $MC_A = WB_A$, and $WC_T = 0.1$. For each triplet $n \in \{3, 9, 27\}$, $WB_A \in [0.1, 2]$, $WP_C \in [0, 20]$, we compute values of p_V that yield mixed and pure equilibria. Then, we proceed as follows.

Compliant behavior. We compute the payoff differentials and the roots of the worker-utility differential polynomial. We ignore the roots that are not real numbers in (0, 1). Each of the remaining roots provides a value for p_C that makes the worker-utility differential zero. That is, it is a mixed equilibrium. We then define the group sizes. Given that in practice it is not expected to have a majority of colluders, we fix $\lceil n/2\rceil$ workers to be non-colluders, and we choose the remaining group sizes at random in $[1, \lfloor n/2\rfloor]$. Armed with the set of possible p_C values,

Table 4. Simulation parameters.

behavior	p_C colluder	p_C non-colluder	p_V
pure	0	0	$> \frac{nWB_A+WC_T}{n(2WB_A+WP_C)}$
compliant	unif. at random from roots of $\Delta U_i(s) = 0$ in (0, 1)	unif. at random from roots of $\Delta U_i(s) = 0$ in (0, 1)	$< \frac{WB_A+WC_T}{2WB_A+WP_C}$
disobedient	{0, 1}	unif. at random from roots of $\Delta U_i(s) = 0$ in (0, 1)	$< \frac{WB_A+WC_T}{2WB_A+WP_C}$
random	unif. at random from (0, 1)	unif. at random from roots of $\Delta U_i(s) = 0$ in (0, 1)	$< \frac{WB_A+WC_T}{2WB_A+WP_C}$

$MB_R = MP_W = MC_V = 100$, $MC_A = WB_A$. $n \in \{3, 9, 27\}$, $WB_A \in [0.1, 2]$, $WP_C \in [0, 20]$, $WC_T = 0.1$. $\lceil n/2\rceil$ groups of size 1 and $\lfloor n/2\rfloor$ of size chosen uniformly at random from $[1, \lfloor n/2\rfloor]$.

doi:10.1371/journal.pone.0116520.t004

Table 5. Expected utilities.

verified	$\vert F\vert$	master	cheater	honest
yes	$= n$	$-MP_W - MC_V + nWP_C$	$-WP_C$	–
yes	$< n$	$MB_R - MC_V - (n - \vert F\vert)MC_A + \vert F\vert WP_C$	$-WP_C$	$WB_A - WC_T/\vert W_i\vert$
no	$> n/2$	$-MP_W - \vert F\vert MC_A$	WB_A	$-WC_T/\vert W_i\vert$
no	$< n/2$	$MB_R - (n - \vert F\vert)MC_A$	0	$WB_A - WC_T/\vert W_i\vert$

doi:10.1371/journal.pone.0116520.t005

call it P, each group W_i chooses a $p_C \in P$ uniformly at random. Then, *(i)* we simulate the computation tossing a biased coin for each group according to the p_C value chosen and for the master according to p_V, and *(ii)* we compute the expected utility of each worker and the expected utility of the master according to the outcome. Table 5 summarizes these computations.

Disobedient behavior. For this scenario we proceed in similar fashion for the non-colluding workers. Additionally, for each group, we choose a p_C at random from P, and we compute the payoff differentials and the utility differential. The rest of the procedure is the same, except that each colluding group decides deterministically to cheat (resp. be honest) if its utility differential is positive (resp. negative).

Random behavior. As before but now each colluding group chooses p_C uniformly at random from (0, 1).

Finally, for the pure equilibrium, we simulate the computation tossing a biased coin only for the master according to p_V, given that the workers follow the equilibrium and are honest deterministically. Then, we compute the expected utility of each worker and the expected utility of the master according to the outcome (detailed in Table 5).

Note that we have developed our own simulation platform that integrates Matlab and C++ code to compute roots, simulate coin tossing, and compute utilities for the different behaviors and parameter combinations.

Discussion and conclusions

Our simulations show that, for the combinations of parameters studied and assuming that the colluders are not a majority, collusion does not help the workers. That is, it is in the best interest of the master *and* also the workers, *even* if they collude, to follow the pure equilibrium $p_C = 0$. In what follows, we support this observation analyzing plots of the results obtained for $n = 9$. Each dot in these plots represents one execution for one parameter-combination. For many parameter combinations, multiple executions have been carried out obtaining similar results. Similar plots were obtained for $n = 3$ and 27 (included in Supporting Information). Other values of n were also simulated obtaining similar results.

Fig. 1 shows the expected utility of the master for all three mixed-equilibria behaviors and the pure equilibrium. Comparing all four cases, we observe that enforcing a pure equilibrium is the best for the master expected utility, even though it has to reward all workers because they never cheat (cf. Fig. 2). The master verifies more frequently for the pure equilibrium (cf. Fig. 3), but still not so frequently to impact in the expected utility (refer to Fig. 1), and on the other hand it is always correct (see Fig. 4). Having a platform where the master is always correct and its expected utility is maximized, the natural question is how good is the situation for workers. It can be seen in Fig. 5 and 6 that the expected utility of workers is also higher if the pure

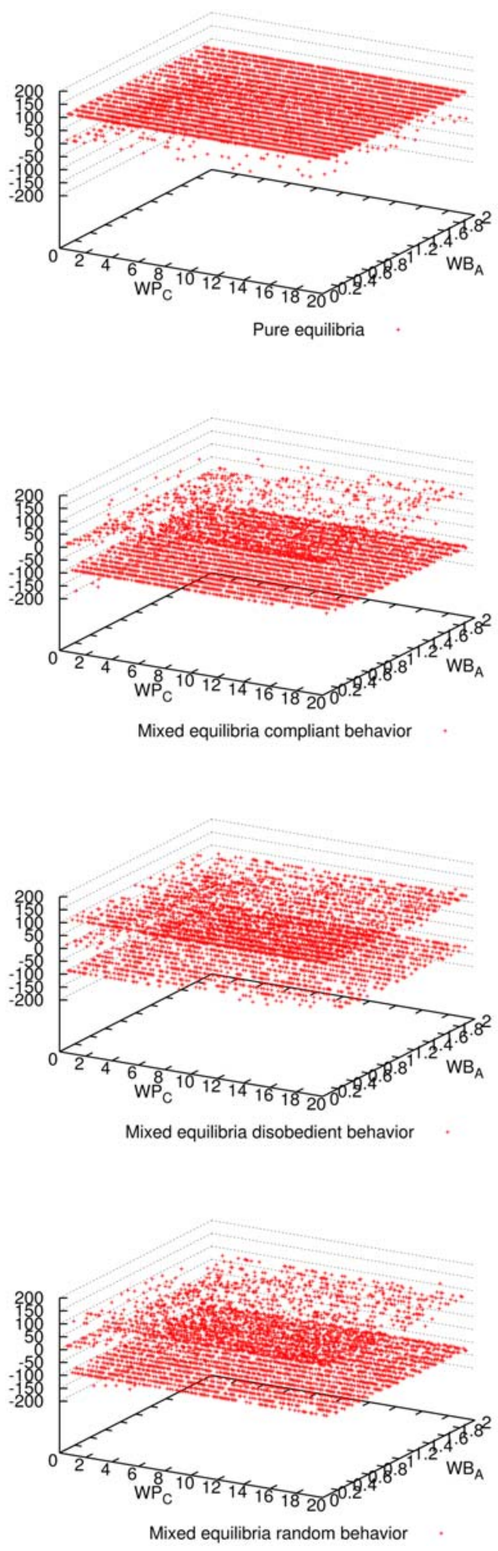

Fig 1. Expected utility of the master for all three mixed-equilibria behaviors and the pure equilibrium and for different parameter combinations when 9 workers participate. It can be seen that enforcing a pure equilibrium is the best for the master utility for most of the parameter combinations.

doi:10.1371/journal.pone.0116520.g001

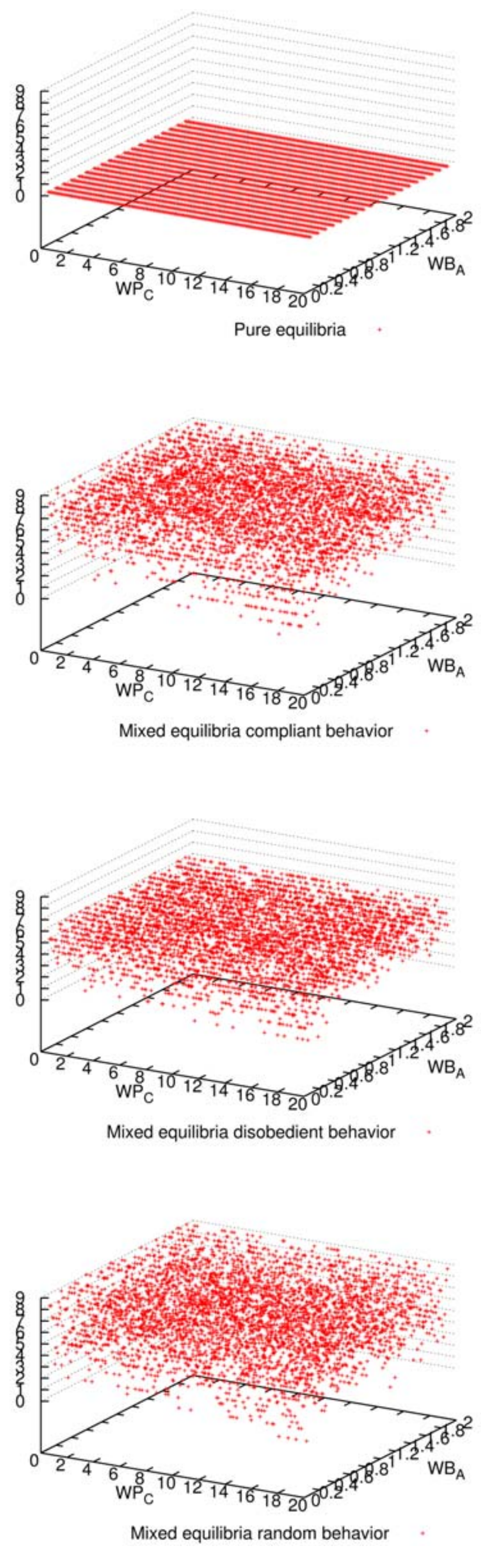

Fig 2. Number of cheaters for all three mixed-equilibria behaviors and the pure equilibrium and for different parameter combinations when 9 workers participate. As expected, enforcing the pure equilibrium ($p_C = 0$) workers are honest for all parameter combinations.

doi:10.1371/journal.pone.0116520.g002

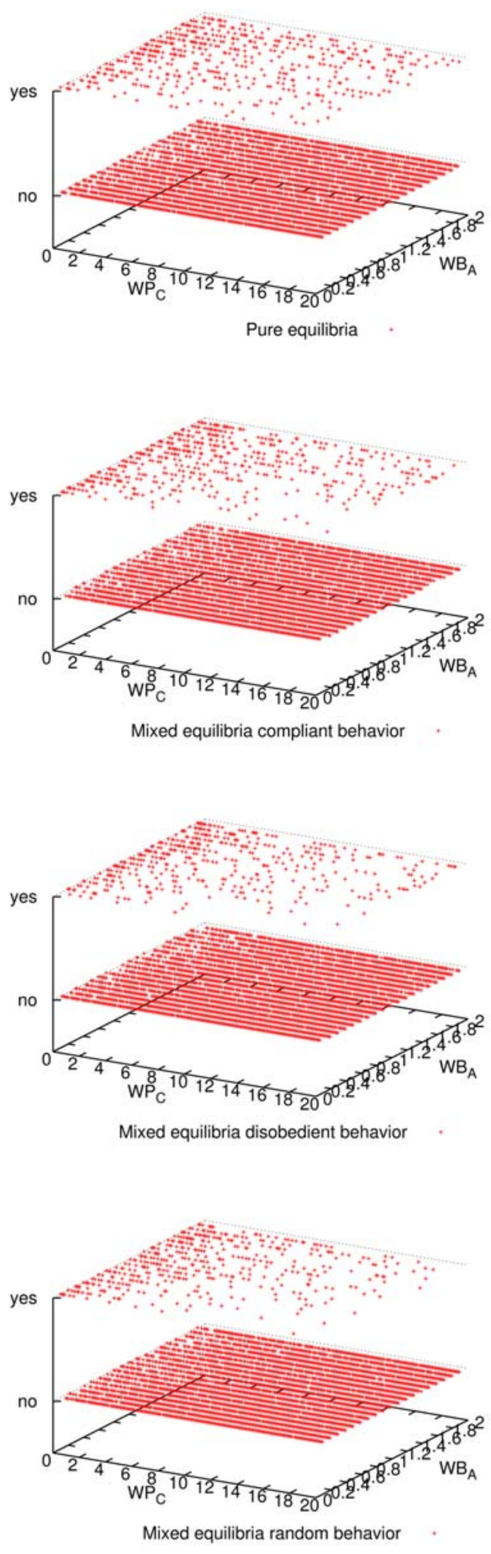

Fig 3. Outcome of master's decision (verify or not) for all three mixed-equilibria behaviors and the pure equilibrium and for different parameter combinations when 9 workers participate. Given the low probability of verification needed, the master did not verify for most of the parameter combinations in all four cases.

doi:10.1371/journal.pone.0116520.g003

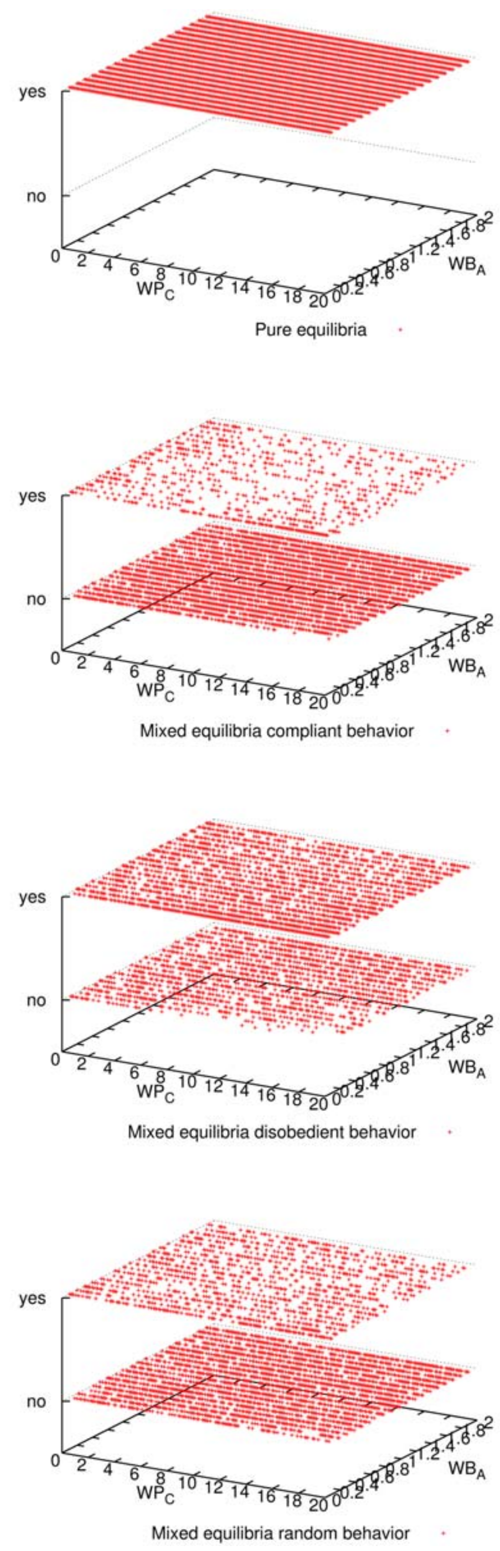

Fig 4. Outcome of the computation (correct or incorrect result) for all three mixed-equilibria behaviors and the pure equilibrium and for different parameter combinations when 9 workers participate. It can be seen that aiming for a pure equilibrium ($p_C = 0$) is the only option to guarantee correctness for any parameter combination.

doi:10.1371/journal.pone.0116520.g004

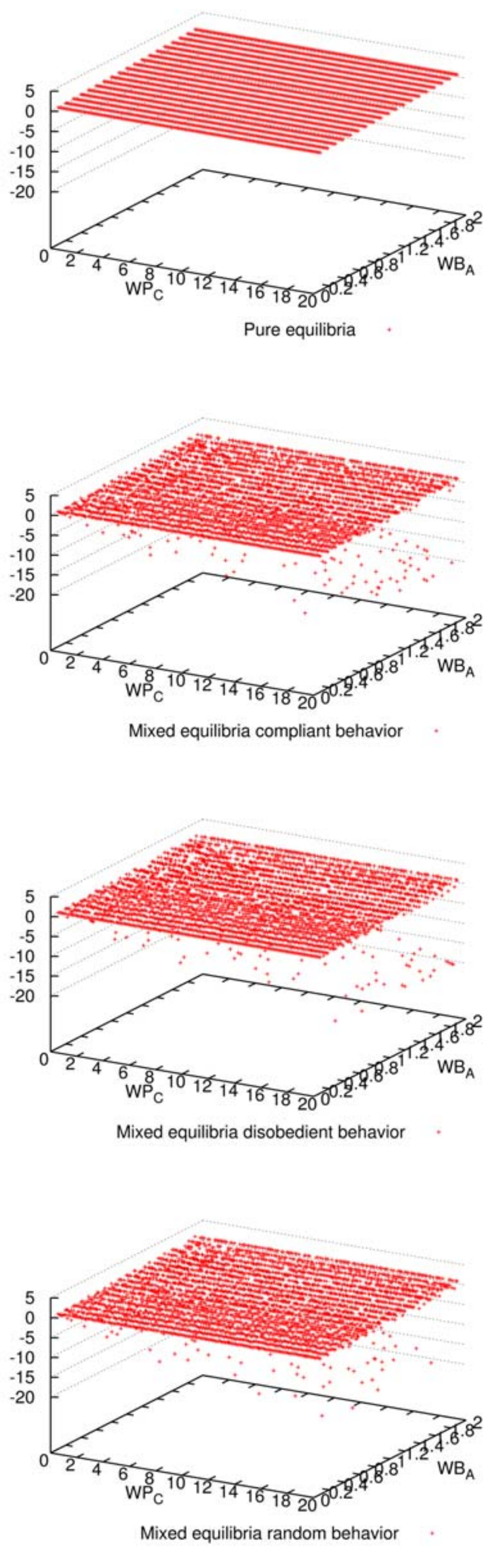

Fig 5. Expected utility of non-colluder workers for all three mixed-equilibria behaviors and the pure equilibrium and for different parameter combinations, when 9 workers participate. It can be seen that aiming for a pure equilibrium is the best for the non-colluder utility for most of the parameter combinations.

doi:10.1371/journal.pone.0116520.g005

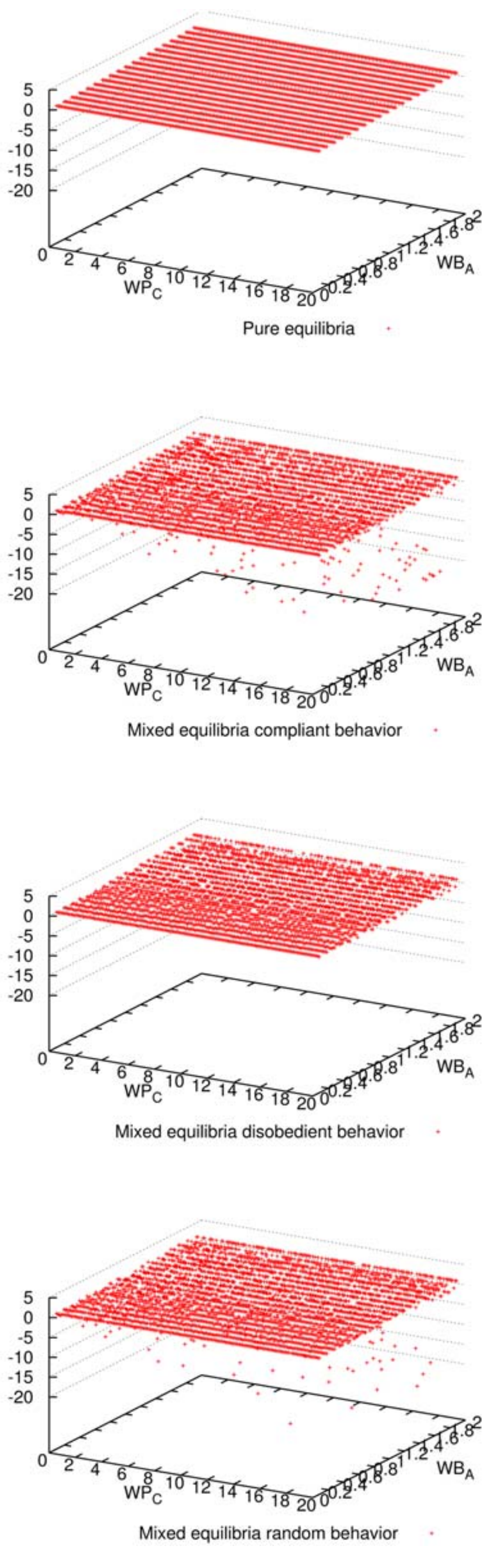

Fig 6. Expected utility of colluder workers for all three mixed-equilibria behaviors and the pure equilibrium and for different parameter combinations when 9 workers participate. It can be seen that aiming for a pure equilibrium is the best for the colluder utility for most of the parameter combinations.

doi:10.1371/journal.pone.0116520.g006

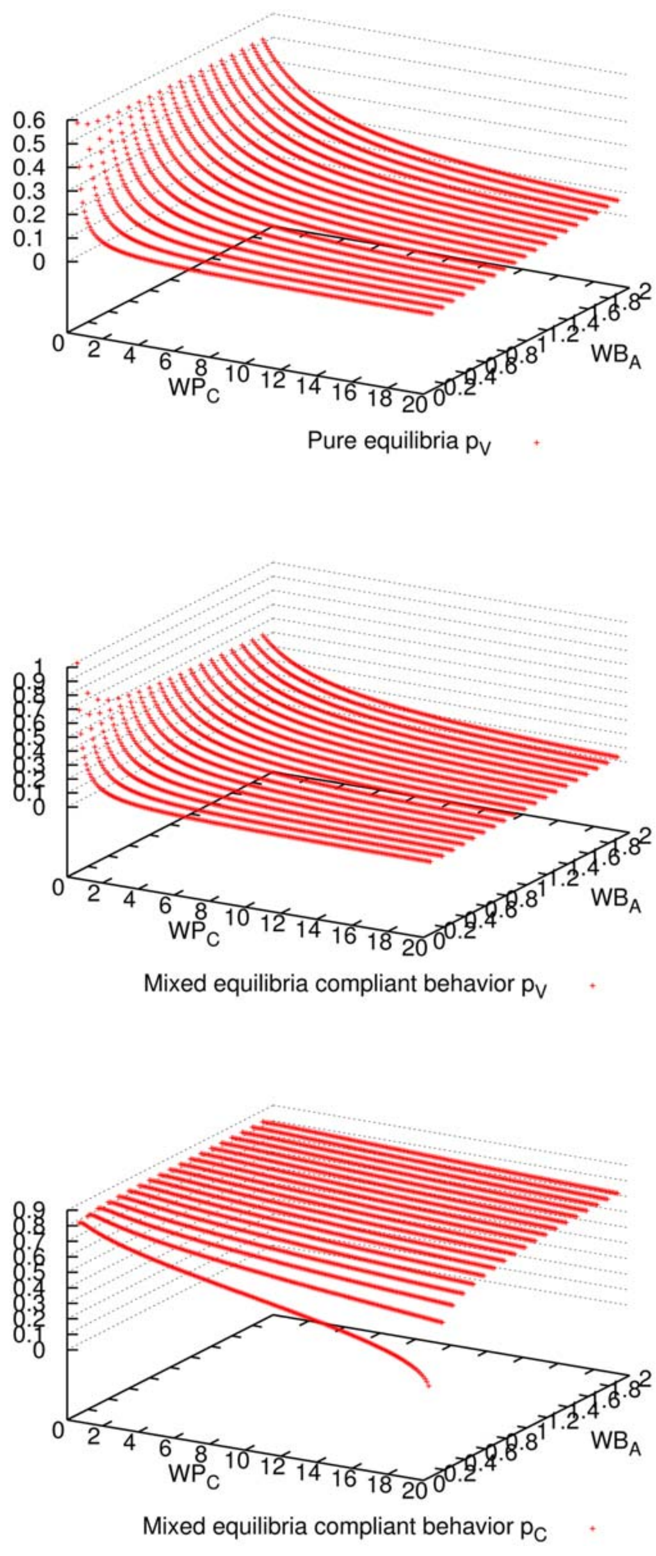

Fig 7. Probability of verification of the master for pure equilibrium and mixed equilibria, and probability of cheating of a compliant non-colluder worker, for different parameter combinations when 9 workers participate. For the master, it can be seen that, for most of the parameter combinations, the probability of verifying is very low.

doi:10.1371/journal.pone.0116520.g007

equilibrium is used. And this is the case for the non-colluders (Fig. 5) as well as for the colluders (Fig. 6). Finally, we also observed that the probability of verifying p_V and the p_C computed by the master are both significantly smaller if the ratio penalty/payoff is large as one might expect (see Fig. 7).

Supporting Information

S1 Fig. Simulation plots for 3 workers. The same observations for the case of 9 workers in Figs. 1 to 7 apply to this case.
(EPS)

S2 Fig. Simulation plots for 27 workers. The same observations for the case of 9 workers in Figs. 1 to 7 apply to this case.
(EPS)

Author Contributions

Conceived and designed the experiments: AFA CG MAM DP. Performed the experiments: AFA CG MAM DP. Analyzed the data: AFA CG MAM DP. Contributed reagents/materials/analysis tools: AFA CG MAM DP. Wrote the paper: AFA CG MAM DP.

References

1. Korpela E, Werthimer D, Anderson D, Cobb J, Lebofsky M (2001) SETI@home: Massively distributed computing for SETI. Computing in Science and Engineering 3(1):78–83.
2. Anderson D (2004) BOINC: A system for public-resource computing and storage. Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing 4–10. doi: 10.1109/GRID.2004.14
3. Amazon's Mechanical Turk. Available: https://www.mturk.com. Accessed 13 November 2014.
4. Golle P, Mironov I (2001) Uncheatable distributed computations. Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographer's Track at RSA 425–440. doi: 10.1007/3-540-45353-9_31
5. Anderson D (2010) Volunteer computing: the ultimate cloud. Crossroads 16(3):7–10. doi: 10.1145/1734160.1734164
6. Kondo D, Araujo F, Malecot P, Domingues P, Silva L, Fedak G, Cappello F (2007) Characterizing result errors in Internet desktop grids. Proceedings of the 13th International Euro-Par Conference European Conference on Parallel and Distributed Computing 361–371.
7. Fernández Anta A, Georgiou Ch, Lopez L, Santos A (2012) Reliable Internet-based computing in the presence of malicious workers. Parallel Processing Letters 22(1):1250002 doi: 10.1142/S0129626412500028
8. Konwar KM, Rajasekaran S, Shvartsman AA (2006) Robust network supercomputing with malicious processes. Proceedings of the 20th International Symposium on DIStributed Computing 474–488.
9. Sarmenta L (2002) Sabotage-tolerance mechanisms for volunteer computing systems. Future Generation Computer Systems 18(4):561–572. doi: 10.1016/S0167-739X(01)00077-2
10. Abraham I, Dolev D, Goden R, Halpern JY (2006) Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. Proceedings of the 25th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing 53–62.
11. Shneidman J, Parkes DC (2003) Rationality and self-interest in P2P networks. Proceedings of the 2nd International Workshop on Peer-to-Peer Systems 139–148. doi: 10.1007/978-3-540-45172-3_13
12. Christoforou E, Fernández Anta A, Georgiou Ch, Mosteiro MA, Sánchez A (2013) Applying the dynamics of evolution to achieve reliability in master-worker computing. Concurrency and Computation: Practice and Experience 25(17):2363–2380. doi: 10.1002/cpe.3104
13. Fernández Anta A, Georgiou Ch, Mosteiro MA (2008) Designing mechanisms for reliable Internet-based computing. Proceedings of the 7th IEEE International Symposium on Network Computing and Applications 315–324.
14. Yurkewych M, Levine BN, Rosenberg AL (2005) On the cost-ineffectiveness of redundancy in commercial P2P computing. Proceedings of the 12th ACM Conference on Computer and Communications Security 280–288.

15. Christoforou E, Fernández Anta A, Georgiou Ch, Mosteiro MA (2014) Algorithmic mechanism design for reliable master-worker Internet-based computing. IEEE Transactions on Computers 63(1):179–195. doi: 10.1109/TC.2012.186

16. Christoforou E, Fernández Anta A, Georgiou Ch, Mosteiro MA, Sánchez A (2013) Reputation-based mechanisms for evolutionary master-worker computing. Proceedings of the 17th International Conference on Principles of Distributed Systems 98–113 doi: 10.1007/978-3-319-03850-6_8

17. Nisan N, Roughgarden T, Tardos T, Vazirani VV, editors (2007) Algorithmic Game Theory. Cambridge University Press. 778 p.

18. Abraham I, Alvisi L, Halpern JY (2011) Distributed computing meets game theory: combining insights from two fields. ACM SIGACT News 42(2):69–76. doi: 10.1145/1998037.1998055

19. Cosmin G, Araujo F, Moura Silva L, Domingues P, Arenas AE (2009) Defeating colluding nodes in desktop grid computing platforms. Journal of Grid Computing 7(4):555–573. doi: 10.1007/s10723-009-9124-5

20. Staab E, Engel T (2009) Collusion Detection for Grid Computing. Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid 412–419.

21. Kondo D, Casanova H, Wing E, Berman F (2002) Models and scheduling mechanisms for global computing applications. Proceedings of the 16th IEEE International Parallel and Distributed Processing Symposium 79–86

22. Douceur JR (2002) The Sybil attack. Proceedings of the 1st International Workshop on Peer-to-Peer Systems 251–260 doi: 10.1007/3-540-45748-8_24

23. Ding S, Xia CY, Zhou KL, Yang SL, Shang JS (2014) Decision support for personalized cloud service selection through multi-attribute trustworthiness evaluation. PLoS ONE 9(6) e97762. doi: 10.1371/journal.pone.0097762 PMID: 24972237

24. Ding S, Yang SL, Zhang YT, Liang CY, Xia CY (2014) Combining QoS prediction and customer satisfaction estimation to solve cloud service trustworthiness evaluation problems. Knowledge-Based Systems 56:216–115. doi: 10.1016/j.knosys.2013.11.014

25. Nisan N, Ronen A (2001) Algorithmic mechanism design. Games and Economic Behavior 35:166'196. doi: 10.1006/game.1999.0790

26. Nash JF (1950) Equilibrium points in *n*-person games. Proceedings of the National Academy of Sciences 36(1):48–49. doi: 10.1073/pnas.36.1.48

27. Babaioff M, Feldman M, Nisan N (2006) Combinatorial agency. Proceedings of the 7th ACM Conference on Electronic Commerce 18–28.

28. Taufer M, Anderson D, Cicotti P, Brooks CL (2005) Homogeneous redundancy: a technique to ensure integrity of molecular simulation results using public computing. Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium 119a. doi: 10.1109/IPDPS.2005.247

29. Golle P, Stubblebine S (2001) Secure distributed computing in a commercial environment. Proceedings of the 5th International Conference Financial Cryptography 289–304.

30. Osborne MJ (2003) An Introduction to Game Theory. Oxford University Press. 560 p.