Algorithmic Mechanisms for Internet Supercomputing under Unreliable Communication

Evgenia Christoforou
Univ. of Cyprus
evgenia.christoforou@gmail.com

Antonio Fernández Anta
Institute IMDEA Networks and URJC
antonio.fernandez@imdea.org

Chryssis Georgiou
Univ. of Cyprus
chryssis@cs.ucy.ac.cy

Miguel A. Mosteiro
Rutgers Univ. and URJC
mosteiro@cs.rutgers.edu

***Abstract*—This work, using a game-theoretic approach, considers Internet-based computations, where a master processor assigns, over the Internet, a computational task to a set of untrusted worker processors, and collects their responses. The master must obtain the correct task result, while maximizing its benefit. Building on prior work, we consider a framework where altruistic, malicious, and rational workers co-exist. In addition, we consider the possibility that the communication between the master and the workers is not reliable, and that workers could be unavailable; assumptions that are very realistic for Internet-based master-worker computations. Within this framework, we design and analyze two algorithmic mechanisms that provide, when necessary, appropriate incentives to rational workers to act correctly, despite the malicious' workers actions and the unreliability of the network. These mechanisms are then applied to two realistic Internet-based master-worker settings, a SETI-like one and a contractor-based one, such as Amazon's mechanical turk.**

***Keywords*-mechanism design; task performance; Internet-based computing; malicious, altruistic, rational workers; unreliable communication.**

I. Introduction

Motivation and prior work: In the last few years we have witnessed the Internet becoming a viable platform for processing complex computational jobs. Several Internet-oriented systems and protocols have been designed to operate on top of this global computation infrastructure; examples include Grid systems (e.g., [5], [17]), the "@home" projects [2], such as SETI [12], and peer-to-peer computing–P2PC (e.g., [8], [18]). Although the potential is great, the use of Internet-based computing is limited by the untrustworthiness nature of the platform's components (see, e.g., [2], [9]–[11], [18]).

In SETI, data is distributed for processing to millions of voluntary machines around the world. At a conceptual level, in SETI there is a machine, call it the *master*, that sends jobs, across the Internet, to these computers, call them the *workers*. These workers execute and report back the result of the computation task. However, these workers are not trustworthy, and hence might report incorrect results. In SETI, the master attempts to minimize the impact of these bogus results by assigning the same task to several workers and comparing their outcomes (that is, *redundant* task allocation is employed [2]). Another popular master-worker Internet-based application is Amazon's mechanical turk [1]. Here the master and the workers can be in fact humans that contribute time for solving problems in exchange to economic rewards. A person who wishes to have a problem (task) solved can act as a master processor and "hire" worker processors (other persons) through the mechanical turk platform and have its task computed.

In [6], an Internet-based master-worker framework was considered where a master processor assigns, over the Internet, a computational task to a set of untrusted worker processors and collects their responses. Three type of workers were assumed: *altruistic, malicious,* and *rational*. Altruistic workers (aka the "good" workers) always compute and return the correct result of the task, malicious workers (aka the "bad" workers) always return an incorrect result, and rational (selfish) workers act based on their self-interest. In other words, the altruistic and malicious workers have a predefined behavior: the first are *honest* and the latter are *cheaters*. Rational workers decide to be honest or to cheat based on which strategy would increase their benefit (utility). (In a massive computation platform, such as the Internet, one cannot preclude the co-existence of all three worker types.) Under this framework, a game-theoretic mechanism was designed that provided necessary incentives to the rational workers to compute and report the correct task result despite the malicious workers' actions. The design objective of the mechanism is for the master to force a desired *Nash Equilibrium* (NE) [13], i.e., a strategy choice by each rational worker such that none of them has incentive to change it. The NE is the one in which the master achieves a desired probability of obtaining the correct task result, while maximizing its benefit. The utility of the mechanism was demonstrated by applying it to the abovementioned paradigmatic applications: a SETI-like volunteer computing system and a contractor-based system, such as Amazon's mechanical turk.

Contributions: This work extends the master-worker framework of [6] by additionally considering the possibility that the communication between the master and the workers is not reliable. That is, we consider the possibility that messages exchanged may get lost or arrive late. This communication uncertainty can either be due to communication-

This work is supported in part by the Cyprus Research Promotion Foundation grant ΤΠΕ/ΠΛΗΡΟ/0609(ΒΕ)/05, Comunidad de Madrid grant S2009TIC-1692, Spanish MICINN grant TIN2008–06735-C02-01, and NSF grant 0937829.

related failures or due to workers being slow in processing messages (or even crashing while doing so). For instance, Heien at al. [10] have found that in BOINC only around 5% of the workers are available more than 80% of the time, and that half of the workers are available less than 40% of the time. This fact, combined with the length of the computation incurred by a task [11], justifies the interest of considering in the Internet-based master-worker framework the possibility of workers not replying.

In order to introduce this possibility in the framework, we consider that there is some positive probability that the master does not receive a reply from a given worker. Since it is now possible for a worker's reply not to reach the master, we additionally extend the framework of [6] by allowing workers to abstain from the computation. (In [6] workers did not have the choice of abstaining.) Imagine the situation where a rational worker decides to compute and truthfully return the task result but its reply is not received by the master. As we explain in Section II, in this case the master provides no reward to the worker, while the worker has incurred the cost of performing the task. Hence, it is only natural to provide to the workers the choice of not replying (especially when the reliability of the network is low). This makes the task of the master even more challenging, as it needs to provide the necessary incentives to encourage rational workers to reply and do so truthfully, even in the presence of low network reliability.

Within this extended framework, we develop and analyze two game-theoretic mechanisms, a time-based mechanism and a reply-based one, that provide the necessary incentives for the rational workers to truthfully compute and return the task result, despite the malicious workers' actions and the network unreliability. Furthermore, we apply our mechanisms to two realistic settings: SETI-like volunteer computing applications and contractor-based applications such as Amazon's mechanical turk. More details can be found in [16].

II. Model and Definitions

Master-workers framework: We consider a distributed system consisting of a master processor that assigns, over the Internet, a computational task to a set of n workers to compute and return the task result. The master, based on the received replies, must decide on the value it believes is the correct outcome of the task. The tasks considered in this work are assumed to have a unique solution (although such limitation reduces the scope of application of the presented mechanisms, there are plenty of computations where the correct solution is unique: e.g., any mathematical function).

Worker types: Each worker has one of the following types: *rational*, *malicious*, or *altruistic*. The exact number of workers of each type is unknown, but a type probability distribution is known: each worker is independently of one of the three types with probabilities p_ρ, p_μ, p_a, respectively, where $p_\rho + p_\mu + p_a = 1$. In this paper, a worker being honest means that it truthfully computes and returns the correct task result, while a cheating worker does not compute the task but returns a bogus result to the master. Malicious and altruistic workers always cheat and are honest, respectively, without caring on how such a behavior impacts their utilities. On the other hand, rational workers are assumed to be selfish in a game-theoretic sense, that is, their aim is to maximize their benefit (utility) under the assumption that other workers do the same. So, a rational worker decides to be honest, cheat or not reply to the master (unlike the work in [6], workers can abstain and choose not to reply) depending on which strategy maximizes its utility. As a result, each rational worker cheats with probability $p_{\mathcal{C}}$, it is honest with probability $p_{\mathcal{H}}$, and does not reply with probability $p_{\mathcal{N}}$, such that $p_{\mathcal{C}} + p_{\mathcal{H}} + p_{\mathcal{N}} = 1$. It is understood that if a worker decides not to reply, then it does not perform the task.

Network unreliability: Unlike the work in [6], the communication network is considered to be unreliable, and workers could be unavailable, which are very realistic assumptions for Internet-based master-worker computations, as suggested, for example, by the work of Heien at al. [10]. We model this shortcoming assuming that the communication with each worker fails stochastically and independently of other workers. Furthermore, we assume two settings, one where the probability of communication failure depends on time (the more the master waits for replies the larger the probability of obtaining more replies), and a second one where the probability of communication failure is fixed (hence, the more workers the master hires the larger the number of replies). As we will see in the next section, the first setting leads to a *time-based* mechanism and the second one to a *reply-based* mechanism. In our analysis, we let d_1 be the probability of any worker being available and receiving the task assignment message by the master, d_2 be the probability of the master receiving the worker's response (has the worker chosen to reply), and $d = d_1 \cdot d_2$ be the probability of a round trip, that is, the probability that the master receives the reply from a given worker. We also assume that there is some chance of a message being delivered to its destination, i.e. $d > 0$, a realistic assumption for today's Internet's infrastructure.

Master's objectives: The main objective of the master is to guarantee that the decided value is correct with probability at least $1 - \varepsilon$, for a desired constant $0 \leq \varepsilon \leq 1$. Then, having achieved this, the master wishes to maximize its own benefit (utility). As, for example, in [15], [7], and [6], while it is assumed that rational workers make their decision individually, it is assumed that all the (malicious and rational) workers that cheat return the same incorrect answer; this yields a worst case scenario, and hence analysis, for the master with respect to its probability of obtaining the correct result.

Auditing, payoffs and reward models: To achieve its objectives, the master employs, if necessary, *auditing* and *reward/penalizing* schemes. The master might decide to audit the responses of the workers (with a cost). In this work, auditing means that the master computes the task itself and checks which workers have been truthful or not. We denote by $p_{\mathcal{A}}$ the probability of the master auditing the responses of the workers.

Furthermore, the master can reward and punish workers, which can be used (possibly combined with auditing) to encourage rational workers to be honest (altruistic workers need no encouragement, and malicious workers do not care about their utility). When the master audits, it can accurately reward and punish the workers. When the master does not audit, it decides on the majority of the received replies and may apply different reward/penalizing schemes. (From the assumptions that cheaters send the same incorrect answer and that tasks have unique solutions, it follows that there can be only two kind of replies: a correct and an incorrect one). In this work we consider the three reward models shown below:

$\mathcal{R}_{\mathrm{m}}$	the master rewards the majority only
$\mathcal{R}_{\mathrm{a}}$	the master rewards all workers whose reply was received
$\mathcal{R}_{\emptyset}$	the master does not reward any worker

Auditing or not, the master neither rewards nor punishes a worker from whom it did not receive its response. Due to the unreliability of the network, when the master does not receive a reply from a worker it can not distinguish whether the worker decided to abstain, or there was a communication failure in the round trip (it could be the case that the worker did not even receive the task assignment message). Hence, it would be unfair to punish a worker for not getting its response; imagine the case where the worker received the request, performed the task and replied to the master, but this last message got lost! On the other hand, if it is indeed the case that a worker received the task assignment message but decided to abstain, then it gets no reward. If the reward is much bigger than the worker's cost for computing the task, this alone can be a counter incentive to such a strategy.

The payoff parameters considered in this work are shown below. All parameters are non-negative. Note that there are different parameters for the reward $WB_{\mathcal{Y}}$ to a worker and the cost $MC_{\mathcal{Y}}$ of this reward to the master; this models the fact that the cost for the master might be different from the benefit for a worker (in some applications they could in fact be completely unrelated). Although workers are not penalized for not replying, our model allows the possibility for the master to be penalized for not getting any replies (parameter $MC_{\mathcal{S}}$). This provides an incentive for the master to choose (when it can) more workers to assign the task (especially if d is small) or to increase their incentives for replying. (If convenient, $MC_{\mathcal{S}}$ could be set to zero.) Among the parameters involved, we assume that

$WP_{\mathcal{C}}$	worker's punishment for being caught cheating
$WC_{\mathcal{T}}$	worker's cost for computing the task
$WB_{\mathcal{Y}}$	worker's benefit from master's acceptance
$MP_{\mathcal{W}}$	master's punishment for accepting a wrong answer
$MC_{\mathcal{Y}}$	master's cost for accepting the worker's answer
$MC_{\mathcal{A}}$	master's cost for auditing worker's answers
$MC_{\mathcal{S}}$	master's cost for not getting any reply
$MB_{\mathcal{R}}$	master's benefit from accepting the right answer

the master has the freedom of choosing $WB_{\mathcal{Y}}$ and $WP_{\mathcal{C}}$; by tuning these parameters and choosing n, the master can achieve the desired trade-offs between correctness and cost. All other parameters can either be fixed because they are system parameters or may also be chosen by the master.

III. ALGORITHMIC MECHANISMS

In this section we present the mechanisms we design and show their analysis.

Algorithms: As mentioned, we consider two different settings for modeling network unreliability, which yield two different protocols.

Figure 1 presents the *time-based* protocol. Based on how the probability of communication failure depends on time, the master fixes a time T, it sends the specification of the task to be computed to n workers, and waits for replies. Once time T is reached, the master gathers all received replies, and chooses to audit the answers with probability $p_{\mathcal{A}}$. If the answers were not audited, it accepts the result of the majority (ties are broken at random). Then, it applies the corresponding reward model.

Figure 2 presents the *reply-based* protocol. Here the master, by appropriately choosing n, fixes k, an estimate of the minimum number of replies that wants to receive with high probability. (We discuss in the next subsection how k is computed and what is the probability of not receiving at least that many answers). The master sends the task specification to the n workers and gets replies. If at least k replies are received, then the master chooses to audit the answers with probability $p_{\mathcal{A}}$ and proceeds as the other protocol. In case that less than k replies are received, then the master does nothing and it incurs penalty $MC_{\mathcal{S}}$.

Notice that both protocols are one-shot, in the sense that they terminate after one round of communication between the master and the workers. This enables fast termination and avoids using complex cheater detection and worker reputation mechanisms. The benefit of one-round protocols is also partially supported by the work of Kondo et al. [11] that have demonstrated experimentally that tasks may take much more than one day of CPU time to complete.

Each of the above protocols basically comprises a *game*, that the master designs, and the rational workers play looking for a *Nash Equilibrium* (NE) in an effort to maximize their benefit. Therefore, based on the type distribution, the master must choose a value of $p_{\mathcal{A}}$ that would yield a *unique* NE that best serves its purposes. The reason for uniqueness

```
send(task, p_A, certificate) to n workers
wait time T for replies
upon expire of time T do
audit the answers with probability p_A
if the answers were not audited then
    accept the majority
end if
apply the reward model
```

Figure 1. Master Algorithm for the Time-based Mechanism

```
send(task, p_A, certificate) to n workers
if at least k replies are received then
    audit the answers with probability p_A
    if the answers were not audited then
        accept the majority
    end if
    apply the reward model
end if
```

Figure 2. Master Algorithm for the Reply-based Mechanism

is to force all workers to the same strategy; this is similar to *strong implementation* in Mechanism Design, cf., [3], [14]. For computational reasons, the master, along with the task specification and the chosen value of $p_{\mathcal{A}}$, also sends a *certificate* to the workers. The certificate includes the strategy that the workers must play to achieve the unique NE together with the appropriate data (system parameters/payoff values and reward model) to demonstrate this fact (more about the certificate can be found in [6]).

Recall that the main objective of the master is to achieve probability of accepting the correct task result of at least $1-\varepsilon$. Once this is achieved, then it seeks to maximize its utility as well. Based on the type distribution, it could be the case that the master may achieve this without relying on actions of the rational workers (e.g., the vast majority of workers are altruistic). Following the terminology of [6], such cases fall into the *free rationals* scenario. The cases in which the master needs to enforce the behavior of rational workers fall into the *guided rationals* scenario. In this scenario, the master must choose $p_{\mathcal{A}}$ such that the benefit of the rational workers is maximized when $p_{\mathcal{C}} = p_{\mathcal{N}} = 0$; in other words, rational workers choose to be honest ($p_{\mathcal{H}} = 1$) and hence they compute and truthfully return the correct task result.

The protocol ran by the master for choosing $p_{\mathcal{A}}$ is presented in Figure 3. Together with each of the protocols in Figures 1 and 2 comprise our mechanisms. The analysis of the mechanisms and the lemmas referenced in Figure 3 are given in the next subsection.

We now provide a couple of examples that demonstrate that both mechanisms are useful:

(a) As discussed in Section II, we consider two settings with respect to the probability of the communication failure: one in which it depends on time, and one in which it is fixed. The master could have knowledge (e.g., based on statistics) of only one of the two settings. In such a case, it has no choice other than using the mechanism designed for that setting.

(b) It is not difficult to see that the time-based mechanism is more likely to use auditing than the other one, on the other hand, the reply-based mechanism runs the risk of not receiving enough replies. Hence, the time-based mechanism would be more preferable in case the cost of auditing is low, and the reply-based mechanism in case the cost of auditing is high and the value of parameter $MC_{\mathcal{S}}$ is small.

Equilibria Conditions and Analysis: We begin the analysis of our mechanisms by elucidating the following probabilities, expected utilities, and equilibria conditions. For succinctness, the analysis of both mechanisms is presented for a minimum number of replies k, where $k = 1$ for the time-based mechanism and $k \geq 1$ for the reply-based mechanism. For the latter, for a given worker type distribution, the choice of n workers, and d, even if all rational workers choose not to reply, the master will receive at least $\boldsymbol{E} = nd(p_\alpha + p_\mu)$ replies in expectation. Thus, using Chernoff bounds, it can be shown that the master will receive at least $k = \boldsymbol{E} - \sqrt{2\boldsymbol{E}\ln(1/\zeta)}$ replies with probability at least $1-\zeta$, for $0 < \zeta < 1$ and big enough n (e.g., $\zeta = 1/n$).

Pr(worker cheats|worker replies): $q = \frac{p_\mu + p_\rho p_{\mathcal{C}}}{1-p_\rho p_{\mathcal{N}}}$

Pr(worker does not cheat|worker replies): $\overline{q} = \frac{p_\alpha + p_\rho p_{\mathcal{H}}}{1-p_\rho p_{\mathcal{N}}} = 1-q$

Pr(reply received): $r = d(1-p_\rho p_{\mathcal{N}})$

Pr(reply not received): $\overline{r} = 1 - r$

Then, $r(q+\overline{q}) + \overline{r} = 1$.

Pr(i out of n replies received): $r_i = \binom{n}{i} r^i \overline{r}^{n-i}$

Pr(majority honest | i replies received):

$$h_i = \sum_{j=0}^{\lfloor \frac{i}{2} \rfloor - 1} \binom{i}{j} q^j \overline{q}^{i-j} + (1 + \lceil \frac{i}{2} \rceil - \lfloor \frac{i}{2} \rfloor) \frac{1}{2} \binom{i}{\lfloor \frac{i}{2} \rfloor} q^{\lfloor \frac{i}{2} \rfloor} \overline{q}^{\lceil \frac{i}{2} \rceil}.$$

Pr(majority cheats | i replies received):

$$c_i = \sum_{j=\lceil \frac{i}{2} \rceil + 1}^{i} \binom{i}{j} q^j \overline{q}^{i-j} + (1 + \lceil \frac{i}{2} \rceil - \lfloor \frac{i}{2} \rfloor) \frac{1}{2} \binom{i}{\lceil \frac{i}{2} \rceil} q^{\lceil \frac{i}{2} \rceil} \overline{q}^{\lfloor \frac{i}{2} \rfloor}.$$

Pr(master obtains correct answer):

$$P_{succ} = \sum_{i=k}^{n} r_i \left(p_{\mathcal{A}} + (1-p_{\mathcal{A}}) h_i\right) \quad (1)$$

E(utility of master):

$$U_M = -\sum_{i=0}^{k-1} r_i MC_{\mathcal{S}} + \sum_{i=k}^{n} r_i \big(p_{\mathcal{A}} \alpha_i + (1-p_{\mathcal{A}}) \beta_i\big) \quad (2)$$

where,

$$\alpha_i = MB_{\mathcal{R}} - MC_{\mathcal{A}} - nd(p_\alpha + p_\rho p_{\mathcal{H}}) MC_{\mathcal{Y}}$$
$$\beta_i = MB_{\mathcal{R}} h_i - MP_{\mathcal{W}} c_i - MC_{\mathcal{Y}} \gamma_i$$

and where, $\gamma_i = 0$ for $\mathcal{R}_\emptyset$, $\gamma_i = i$ for $\mathcal{R}_{\mathrm{a}}$, and for $\mathcal{R}_{\mathrm{m}}$ is,

$$\gamma_i = \sum_{j=\lceil \frac{i}{2} \rceil + 1}^{i} \binom{i}{j} j (\overline{q}^j q^{i-j} + q^j \overline{q}^{i-j}) +$$
$$(1 + \lceil \frac{i}{2} \rceil - \lfloor \frac{i}{2} \rfloor) \frac{1}{2} \binom{i}{\lceil \frac{i}{2} \rceil} \lceil \frac{i}{2} \rceil (\overline{q}^{\lceil \frac{i}{2} \rceil} q^{\lfloor \frac{i}{2} \rfloor} + q^{\lceil \frac{i}{2} \rceil} \overline{q}^{\lfloor \frac{i}{2} \rfloor}).$$

if *Pr*[*majority honest* | *all rationals honest*] $< 1-\varepsilon$ **then** /* P_{succ} is small, even if $p_{\mathcal{H}} = 1$ */
    $p_{\mathcal{C}} \leftarrow 1$; $p_{\mathcal{N}} \leftarrow 0$; $p_{\mathcal{A}} \leftarrow 1-\varepsilon/\sum_{i=k}^{n} r_i c_i$; /* cf. Lemma 2 */
elseif *Pr*[*majority honest* | *all rationals cheat*] $\geq 1-\varepsilon$ **then** /* P_{succ} is big, even if $p_{\mathcal{C}} = 1$ */
    $p_{\mathcal{C}} \leftarrow 1$; $p_{\mathcal{N}} \leftarrow 0$; $p_{\mathcal{A}} \leftarrow 0$; /* cf. Lemma 3 */
elseif *Pr*[*majority honest* | *all rationals honest*] $\geq 1-\varepsilon$ **and**
    $\Delta U_{\mathcal{HC}}(p_{\mathcal{H}} = 1, p_{\mathcal{A}} = 0) \geq 0$ **and** $\Delta U_{\mathcal{HN}}(p_{\mathcal{H}} = 1, p_{\mathcal{A}} = 0) \geq 0$ **then** /* $p_{\mathcal{H}} = 1$, even if $p_{\mathcal{A}} = 0$ */
    $p_{\mathcal{C}} \leftarrow 0$; $p_{\mathcal{N}} \leftarrow 0$; $p_{\mathcal{A}} \leftarrow 0$; /* cf. Lemma 3 */
else /* $p_{\mathcal{C}} = 0$ and $p_{\mathcal{N}} = 0$ enforced */
    $p_{\mathcal{C}} \leftarrow 0$; $p_{\mathcal{N}} \leftarrow 0$; *set* $p_{\mathcal{A}}$ *as in* Lemma 4; /* cf. Lemma 4 */
if $U_M(p_{\mathcal{A}}, p_{\mathcal{N}}, p_{\mathcal{C}}) < U_M\left(p_{\mathcal{A}} = (1-\varepsilon)/\sum_{i=k}^{n} r_i, p_{\mathcal{N}} = 1, p_{\mathcal{C}} = 0\right)$ **then**
    $p_{\mathcal{N}} \leftarrow 1$; $p_{\mathcal{A}} \leftarrow (1-\varepsilon)/\sum_{i=k}^{n} r_i$; /* cf. Lemma 1 */

Figure 3. Master protocol to choose $p_{\mathcal{A}}$. The expressions of k, r_i, and c_i are defined in Section III

We denote by $\Delta U_{S_1 S_2}$ the difference on the expected utilities of a rational worker when choosing strategy S_1 over strategy S_2. Then, for any rational worker, the equilibria conditions are:

$$\begin{cases} \Delta U_{\mathcal{HC}} = \boldsymbol{\pi}_{\mathcal{H}} \cdot \boldsymbol{w}_{\mathcal{H}} - \boldsymbol{\pi}_{\mathcal{C}} \cdot \boldsymbol{w}_{\mathcal{C}} & \geq 0 \\ \Delta U_{\mathcal{HN}} = \boldsymbol{\pi}_{\mathcal{H}} \cdot \boldsymbol{w}_{\mathcal{H}} - \boldsymbol{\pi}_{\mathcal{N}} \cdot \boldsymbol{w}_{\mathcal{N}} & \geq 0 \end{cases} \tag{3}$$

The components of the vectors denoted by $\boldsymbol{w}_\bullet$ in (3) correspond to the different payoffs received by the given worker for each of the various events that may outcome from the game when the worker has chosen strategy $\bullet$, and the components of the vectors denoted by $\boldsymbol{\pi}_\bullet$ to the probabilities that those events occur. A thorough detail of their specific values is left to the full version [16] of this extended abstract for brevity. These conditions are defined for the guided rationals case so that a pure NE where $p_{\mathcal{H}} = 0$ is precluded. We now proceed to analyze the different cases, first considering the free rationals scenario and then the guided rationals one. *Proofs can be found in [16].*

1) Free Rationals: Here we study the various cases where the behavior of rational workers does not need to be enforced. As mentioned before the main goal is to carry out the computation obtaining the correct output with probability at least $1-\varepsilon$. Provided that this goal is achieved, it is desirable to maximize the utility of the master. Hence if, for a given instance of the problem, the expected utility of the master utilizing the mechanism presented is smaller than the utility of just setting $p_{\mathcal{A}}$ big enough to guarantee the desired probability of correctness, independently of the outcome of the game, the latter is used. We establish this observation in the following lemma.

Lemma 1. *In order to guarantee $P_{succ} \geq 1-\varepsilon$, it is enough to set $p_{\mathcal{A}} = (1-\varepsilon)/\sum_{i=k}^{n} r_i$, making $p_{\mathcal{N}} = 1$.*

We consider now pessimistic worker-type distributions, i.e., distributions where p_μ is so large that, even if all rationals choose to be honest, the probability of obtaining the correct answer is too small. Hence, the master has to audit with a probability big enough, perhaps bigger than the minimum needed to ensure that all rationals are honest. Nevertheless, for such $p_{\mathcal{A}}$, rational workers still might use some NE where $p_{\mathcal{H}} < 1$. Thus, the worst case for P_{succ} has to be assumed. Formally,

Lemma 2. *In order to guarantee $P_{succ} \geq 1-\varepsilon$, it is enough to set $p_{\mathcal{A}} = 1-\varepsilon/\sum_{i=k}^{n} r_i c_i$, making $p_{\mathcal{C}} = 1$ and $p_{\mathcal{N}} = 0$.*

Now, we consider cases where no audit is needed to achieve the desired probability of correctness. I.e., we study conditions under the assumption that $p_{\mathcal{A}} = 0$. The first case occurs when the type-distribution is such that, even if all rational workers cheat, the probability of having a majority of correct answers is at least $1-\varepsilon$. A second case happens when the particular instance of the parameters of the game force a unique NE such that rationals are honest, even if they know that the result will not be audited. We establish those cases in the following lemma.

Lemma 3. *In order to guarantee $P_{succ} \geq 1-\varepsilon$, if $\sum_{i=k}^{n} r_i h_i \geq 1-\varepsilon$ making $p_{\mathcal{C}} = 1$ and $p_{\mathcal{N}} = 0$; or the same condition holds but making $p_{\mathcal{C}} = 0$ and $p_{\mathcal{N}} = 0$ and there is a unique NE for $p_{\mathcal{H}} = 1$ and $p_{\mathcal{A}} = 0$, then it is enough to set $p_{\mathcal{A}} = 0$.*

2) Guided Rationals: We now study worker-type distributions such that the master can take advantage of a specific NE to achieve the desired bound on the probability of error. Given that the scenario where all players cheat was considered in Section III-1, in this section it is enough to study $\Delta U_{\mathcal{HC}}$ and $\Delta U_{\mathcal{HN}}$ for each reward model, conditioning $\Delta U_{\mathcal{HC}}(p_{\mathcal{C}} = 1) \geq 0$ and $\Delta U_{\mathcal{HN}}(p_{\mathcal{N}} = 1) \geq 0$ to obtain appropriate values for $p_{\mathcal{A}}$. As proved in the following lemma, the specific value $p_{\mathcal{A}}$ assigned depends on the reward model, and it is set so that a unique pure NE is forced at $p_{\mathcal{H}} = 1$ and the correctness probability is achieved.

Lemma 4. *In order to guarantee $P_{succ} \geq 1-\varepsilon$, if $\sum_{i=k}^{n} r_i h_i < 1-\varepsilon$ making $p_{\mathcal{C}} = 1$ and $p_{\mathcal{N}} = 0$, and $\sum_{i=k}^{n} r_i h_i \geq 1-\varepsilon$ making $p_{\mathcal{C}} = 0$ and $p_{\mathcal{N}} = 0$, then it is enough to set $p_{\mathcal{A}}$ as follows.*

For $\mathcal{R}_\emptyset$,

$$p_{\mathcal{A}} = \frac{WC_{\mathcal{T}}}{d_2\, WB_{\mathcal{Y}} \sum_{i=k-1}^{n-1} r_i'} \tag{4}$$

For $\mathcal{R}_{\mathrm{a}}$,

$$p_{\mathcal{A}} = \frac{WC_{\mathcal{T}}}{d_2(WB_{\mathcal{Y}} + WP_{\mathcal{C}}) \sum_{i=k-1}^{n-1} r_i'},\ d_2\, WB_{\mathcal{Y}} \sum_{i=k-1}^{n-1} r_i' \geq WC_{\mathcal{T}} \tag{5}$$

For $\mathcal{R}_{\mathrm{m}}$,

$$p_{\mathcal{A}} = \frac{WC_{\mathcal{T}}/d_2 - WB_{\mathcal{Y}} \sum_{i=k-1}^{n-1} r_i'(h_i' - c_i')}{(WB_{\mathcal{Y}} + WP_{\mathcal{C}}) \sum_{i=k-1}^{n-1} r_i' - WB_{\mathcal{Y}} \sum_{i=k-1}^{n-1} r_i'(h_i' - c_i')} \tag{6}$$

$$p_{\mathcal{A}} = \frac{WC_{\mathcal{T}}/d_2 - WB_{\mathcal{Y}} \sum_{i=k-1}^{n-1} r_i' h_i'}{WB_{\mathcal{Y}} \sum_{i=k-1}^{n-1} r_i' - WB_{\mathcal{Y}} \sum_{i=k-1}^{n-1} r_i' h_i'} \tag{7}$$

Where

$$r_i' = \binom{n-1}{i} r^i \overline{r}^{n-1-i},$$

$$h'_i = \sum_{j=0}^{\lfloor \frac{i}{2} \rfloor} \binom{i}{j} q^j \overline{q}^{i-j} + \left(\lceil \tfrac{i}{2} \rceil - \lfloor \tfrac{i}{2} \rfloor\right) \tfrac{1}{2} \binom{i}{\lceil \frac{i}{2} \rceil} q^{\lceil \frac{i}{2} \rceil} \overline{q}^{\lfloor \frac{i}{2} \rfloor},$$
$$c'_i = \sum_{j=\lceil \frac{i}{2} \rceil}^{i} \binom{i}{j} q^j \overline{q}^{i-j} + \left(\lceil \tfrac{i}{2} \rceil - \lfloor \tfrac{i}{2} \rfloor\right) \tfrac{1}{2} \binom{i}{\lfloor \frac{i}{2} \rfloor} q^{\lfloor \frac{i}{2} \rfloor} \overline{q}^{\lceil \frac{i}{2} \rceil},$$

for $p_{\mathcal{C}} = 1$ in conditions (5)(a) and (6), and for $p_{\mathcal{N}} = 1$ in conditions (4), (5)(b) and (7).

3) Correctness and Optimality: The following theorem summarizes the previous analyses, and proves the correctness of the mechanisms designed.

Theorem 5. *For any given system parameters, the values of $p_{\mathcal{A}}$ obtained in Sections III-1 and III-2 satisfy that $P_{succ} \geq 1 - \varepsilon$.*

Furthermore, it turns out that the strategy enforced by our mechanisms is optimal.

Theorem 6. *In order to achieve $P_{succ} \geq 1 - \varepsilon$, the only feasible approaches are either to enforce a NE where $p_{\mathcal{H}} = 1$ or to use a $p_{\mathcal{A}}$ as shown in Lemma 2.*

IV. Application of the 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 how to choose $p_{\mathcal{A}}$ and n in the case where the behavior of rational workers is enforced, i.e., under the conditions of Lemma 4.

SETI-like Scenario: We first consider a volunteering computing system such as SETI@home. 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. Hence, we assume that $WB_{\mathcal{Y}} > WC_{\mathcal{T}} = 0$. The master incurs in a (possibly small) cost $MC_{\mathcal{Y}}$ when rewarding a worker. The master may audit the values returned by the workers with cost $MC_{\mathcal{A}} > 0$. We also assume that the master obtains a benefit $MB_{\mathcal{R}} > MC_{\mathcal{Y}}$ if it accepts the correct result, and suffers a cost $MP_{\mathcal{W}} > MC_{\mathcal{A}}$ if it accepts an incorrect value. Also, as stressed before, $d > 0$.

Plugging $WC_{\mathcal{T}} = 0$ in the lower bounds of Lemma 4 it can be seen that, for this scenario and conditions, in order to achieve the desired P_{succ}, it is enough to set $p_{\mathcal{A}}$ arbitrarily close to 0 for all three models. So, we want to choose $\delta \leq p_{\mathcal{A}} \leq 1$, with $\delta \to 0$, so that the utility of the master is maximized. Using calculus, it can be seen that U_M is monotonic in such range, but the growth of such function depends on the specific instance of the master-payoff parameters. Thus, it is enough to choose one of the extreme values of $p_{\mathcal{A}}$. Replacing in Eq. (2),

$$U_M \approx -\sum_{i=0}^{k-1} r_i MC_{\mathcal{S}} + \sum_{i=k}^{n} r_i \max\{\alpha_i, \beta_i\} \tag{8}$$

For $p_{\mathcal{N}} = 0$ and α_i, β_i as in Eq. (2). The approximation given in Eq. (8) provides a mechanism to choose $p_{\mathcal{A}}$ and n so that U_M is maximized for $P_{succ} \geq 1 - \varepsilon$ for any given worker-type distribution, reward model, and set of payoff parameters in the SETI scenario.

Contractor Scenario: The second scenario considered is a company that buys computational power from Internet users and sells it to computation-hungry costumers. An example is Amazon's Mechanical Turk [1]. In this case the company pays the users an amount $S = WB_{\mathcal{Y}} = MC_{\mathcal{Y}}$ for using their computing capabilities, and charges the consumers another amount $MB_{\mathcal{R}} > MC_{\mathcal{Y}}$ for the provided service. Since the users are not volunteers in this scenario, we assume that computing a task is not free for them (i.e., $WC_{\mathcal{T}} > 0$), and that rational workers must have incentives to participate, that is, their utility must be positive. As in the previous case, we assume that the master verifies and has a cost for accepting a wrong value, such that $MP_{\mathcal{W}} > MC_{\mathcal{A}} > 0$. Also as before we assume that $d > 0$.

Using similar reasoning as before (and calculus), for example, for the $\mathcal{R}_\emptyset$ model, using Lemma 4 and conditioning that workers must have positive utility, we get,

$$\begin{aligned} U_M = &-\sum_{i=0}^{k-1} r_i MC_{\mathcal{S}} \\ &+ \sum_{i=k}^{n} r_i \max\left\{\alpha_i, \beta_i + (\alpha_i - \beta_i)\frac{WC_{\mathcal{T}}}{d_2\, WB_{\mathcal{Y}} \sum_{i=k-1}^{n-1} r'_i}\right\}. \end{aligned}$$

References

[1] Amazon's Mechanical Turk, `https://www.mturk.com`.

[2] D. Anderson. BOINC: A system for public-resource computing and storage. In *proc. of GRID 2004*, pp. 4–10.

[3] M. Babaioff, M. Feldman, and N. Nisan. Combinatorial agency. In *proc. of ACM EC 2006*, pp. 18–28.

[4] K. Eliaz. Fault tolerant implementation. *Review of Economic Studies*, 69:589–610, 2002.

[5] "Enabling Grids for E-sciencE", `http://www.eu-egee.org`.

[6] A. Fernández Anta, Ch. Georgiou and M. A. Mosteiro. Algorithmic Mechanisms for Internet-based Master-Worker Computing with Untrusted and Selfish Workers. In *proc. of IPDPS 2010*, pp. 378-388.

[7] A. Fernández Anta, Ch. Georgiou, and M. A. Mosteiro. Designing mechanisms for reliable Internet-based computing. In *proc. of NCA 2008*, pp. 315–324.

[8] I.T. Foster and A. Iamnitchi. On death, taxes, and the convergence of P2P and grid computing. In *proc. of IPTPS 2003*, pp. 118–128.

[9] P. Golle and I. Mironov. Uncheatable distributed computations. In *proc. of CT-RSA 2001*, pp. 425–440.

[10] E.M. Heien, D.P. Anderson, and K. Hagihara. Computing low latency batches with unreliable workers in volunteer computing environments. *Journal of Grid Computing*, 7:501–518, 2009.

[11] D. Kondo, F. Araujo, P. Malecot, P. Domingues, L. Silva, G. Fedak, and F. Cappello. Characterizing result errors in Internet desktop grids. In *proc. of Euro-Par 2007*, pp. 361–371.

[12] E. Korpela, D. Werthimer, D. Anderson, J. Cobb, and M. Lebofsky. SETI@home: Massively distributed computing for SETI. *Comp. in Science and Engineering*, 3(1):78–83, 2001.

[13] J.F. Nash. Equilibrium points in n-person games. *National Academy of Sciences*, 36(1):48–49, 1950.

[14] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[15] L. Sarmenta. Sabotage-tolerance mechanisms for volunteer computing systems. *Future Generation Computer Systems*, 18(4):561–572, 2002.

[16] Technical Report of this work, `http://www.cs.ucy.ac.cy/ric/MARfailTR.pdf`.

[17] "TeraGrid", `http://www.teragrid.org`.

[18] M. Yurkewych, B.N. Levine, and A.L. Rosenberg. On the cost-ineffectiveness of redundancy in commercial P2P computing. In *proc. of CCS 2005*, pp. 280–288.