institute IMDEA networks

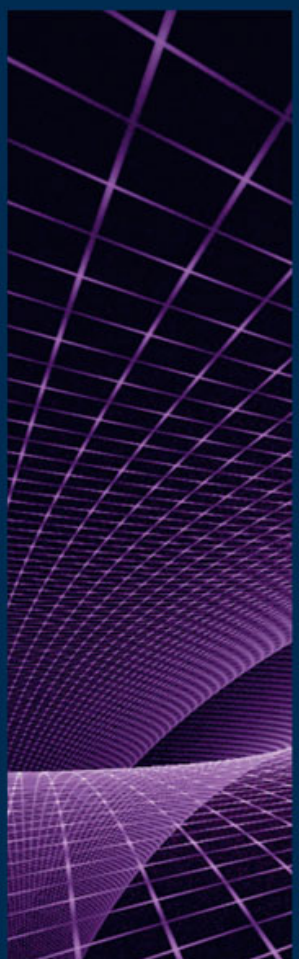
technical report

TR-IMDEA-Networks-2014-2

Reputation-Based Mechanisms for Reliable Crowdsourcing Computation

Evgenia Christoforou
Antonio Fernández Anta
Chryssis Georgiou
Miguel A. Mosteiro
Ángel (Anxo) Sánchez

August 2014



Reputation-Based Mechanisms for Reliable Crowdsourcing Computation ⋆

Evgenia Christoforou[1], Antonio Fernández Anta[2], Chryssis Georgiou[3], Miguel A. Mosteiro[4], and Angel (Anxo) Sánchez[5]

[1] Institute IMDEA Networks, Madrid, Spain & Universidad Carlos III de Madrid, Madrid, Spain
evgenia.christoforou@imdea.org
[2] Institute IMDEA Networks, Madrid, Spain
antonio.fernandez@imdea.org
[3] University of Cyprus, Nicosia, Cyprus
chryssis@cs.ucy.ac.cy
[4] Kean University, Union, NJ, USA & Univ. Rey Juan Carlos, Madrid, Spain
mmosteir@kean.edu
[5] Universidad Carlos III de Madrid, Madrid, Spain & BIFI Institute, Zaragoza, Spain
anxo@math.uc3m.es

Abstract. We consider an Internet-based Master-Worker framework, for machine-oriented computing tasks (i.e. SETI@home) or human intelligence tasks (i.e. Amazon's Mechanical Turk). In this framework a master sends tasks to unreliable workers, and the workers execute and report back the result. We model such computations using evolutionary dynamics and consider three type of workers: *altruistic*, *malicious* and *rational*. Altruistic workers always return the correct result, malicious workers always return an incorrect result, and rational (selfish) workers decide whether to be truthful depending on what increases their benefit. The goal of the master is reaching *eventual correctness*, that is, a stable state of the system in which it always obtains the correct results. To this respect, we propose a mechanism that uses *reinforcement learning* to induce a correct behavior to rational workers; coping with malice leveraging *reputation schemes*. We analyze our system as a Markov chain and we give provable guarantees under which truthful behavior can be ensured. Simulation results, obtained using parameter values similar to the values observed in real systems, reveal interesting trade-offs between various metrics and parameters, such as cost, time of convergence to a truthful behavior, tolerance to cheaters and the type of reputation metric employed.

Keywords: Volunteer computing, crowdsourcing, evolutionary dynamics, reinforcement learning, reputation.

1 Introduction

1.1 Motivation and prior work

The need for high-performance computing and the growing use of personal computers and their capabilities (i.e. CPU and GPU), and the wide access to the Internet, have led to the development of Internet-based computing. At present, Internet-based computing is embraced by the scientific community mostly in the form of volunteer computing, where computing resources are volunteered by the public to help solve scientific problems. Among the most popular volunteering projects is SETI@home [27] running on the BOINC [5] platform.

Besides users volunteering computational resources, humans themselves connected to the Internet are a source of computational power. Luis von Ahn pioneered games with a purpose [40] as a mean of mining useful data out of the players, while keeping them entertained through playing the online game. The word crowdsourcing was introduced by Jeff Howe [21] to describe the situation where human intelligence tasks are executed over the Internet by humans that are given monetary, social or other kind of incentives. A profit-seeking computation platform has been developed by Amazon, called Mechanical Turk [4] (MTurk). Users sign in to the platform and choose to perform human

⋆ This work is supported by the Cyprus Research Promotion Foundation grant ΤΠΕ/ΠΛΗΡΟ/0609(BE)/05, the National Science Foundation (CCF-0937829, CCF-1114930), Kean University UFRI grant, Comunidad de Madrid grants S2009TIC-1692 and MODELICO-CM, and MINECO/MICINN grants TEC2011-29688-C02-01 and PRODIEVO, and National Natural Science Foundation of China grant 61020106002.

intelligence tasks (HIT). In return they get a monetary reward. The most common tasks encountered in MTurk are closed class questions (following the categorization in [17]), meaning that the range of answers is a limited predefined set. Another profit-seeking platform, Bitcoin mining [11], has recently produced a sky-rocketing interest from users and the financial industry. In Bitcoin mining workers carry out complex computations to validate transactions based on Bitcoins, a virtual currency. The computational paradigm is peer-to-peer, that is, there is no centralized authority. Nevertheless, the whole system can be viewed as the master assigning tasks to workers, or in this case, the miners. Given that miners may be deceitful, the system must include measures to prevent or minimize this drawback. However, so far, Bitcoin relies on the complexity of the computation and Bitcoin payments to guarantee trustworthiness [10]. So, although the potential is great, the use of Internet-based computing is limited by the untrustworthy nature of the platform's components [5, 20].

In Internet-based Master-Worker task computing systems a master process sends tasks, across the Internet, to worker processes, that execute and report back the result. However, these workers are not trustworthy, and hence might report incorrect results [5, 6, 25]. Prior work has considered different approaches in tackling the problem. A classical Distributing Computing approach is to model the malfunctioning (due to a hardware or a software error) or cheating (intentional wrongdoer) as *malicious* workers that wish to hamper the computation and thus always return an incorrect result. The non-faulty workers are viewed as *altruistic* ones [32] that always return the correct result. Under this view, malicious-tolerant protocols have been considered, e.g., [31, 26, 8], where the master decides on the correct result based on majority voting. A Game-theoretic approach is to assume that workers are *rational* [1, 19, 33], that is, a worker decides whether to truthfully compute and return the correct result or return a bogus result, based on the strategy that best serves its self-interest (increases its benefit). Under this view, incentive-based algorithmic mechanisms have been devised, e.g., [42, 9], that employ reward/punish schemes to "enforce" rational workers to act correctly.

In [15], all three types were considered, and both approaches were combined in order to produce an algorithmic mechanism that provides incentives to rational workers to act correctly, while alleviating the malicious workers' actions. All the solutions described are *one-shot (or stateless)* in the sense that the master decides about the outcome of an interaction with the workers involving a specific task, without using any knowledge gained by prior interactions. In [16], we took advantage of the repeated interactions between the master and the workers, assuming the presence of *only* rational workers. For this purpose, we studied the *dynamics of evolution* [35] of such master-worker computations through *reinforcement learning* [37] where both the master and the workers adjust their strategies based on their prior interaction. The objective of the master is to reach a state in the computation after which it always obtains the correct results, while the workers attempt to increase their benefit. Hence, prior work either considered all three types of workers in one-shot computations, or multi-round interactions assuming only rational workers.

In volunteer computing workers join projects to support a scientific goal and/or to gain prestige [6], while in non-volunteer computing workers expect payment. Whatever the reason, Internet-based computing can not be considered a reliable platform [5, 6, 20, 25, 17]. Thus provable guarantees must be given that the designed mechanism provides a reliable platform, especially in commercial platforms where one can not consider altruistic workers. The existence of all three types of workers must be assumed since workers can have a predefined behavior (malicious or altruistic) or not (rational) as we observe from the survey conducted by SETI@home [32], the behavior of its users [12] and studies conducted in crowdsourcing platforms [17, 22]. A mechanism must be designed that benefits from the repeated interaction with the workers and thus detaches the knowledge of the distribution over the type of workers from the assumptions (in comparison with [15]).

1.2 Our contributions

In this paper we study a master-worker crowdsourcing model where we assume the existence of a master that assigns tasks in an online fashion to a fixed set of workers. Workers on the other hand are active and always reply to the master. Workers can be one of three types, altruistic, malicious or rationals. Our mechanisms reinforces the behavior of the rational workers by providing the appropriate incentives in order to return the correct reply. The first goal of the system is to achieve a stable state where the master will always receive the correct task reply with the minimum auditing, from there forth. On top of that, in order to deal with malicious workers while keeping the auditing probability minimum, we implement a reputation scheme. Our specific contributions follow.

– We design such an algorithmic mechanism that uses reinforcement learning to induce a correct behavior to rational workers while coping with malice using *reputation*. We consider a centralized reputation scheme controlled by the master that may use three different reputation metrics to calculate each worker's reputation. The first is adopted from [36], the second, which we introduce, allows for a more drastic change of reputation and the third is inspired by BOINC's reputation scheme [7].
– We analyze our reputation-based mechanism modeling it as a Markov chain and we identify conditions under which truthful behavior can be ensured. We analytically prove that by using the second reputation type (i.e. the one we introduce) reliable computation is eventually achieved.
– Simulation results, obtained using parameter values extracted by BOINC-operated applications (such as [18, 3]), reveal interesting trade-offs between various metrics and parameters, such as cost, time of convergence to a truthful behavior, tolerance to cheaters and the type of reputation metric employed. Simulations also reveal better performance of our reputation type (second type) in several realistic cases.

1.3 Background and related work

As part of our mechanism we use reinforcement learning to induce the correct behavior of rational workers. Reinforcement learning [37] models how system entities, or *learners*, interact with the environment to decide upon a strategy, and use their experience to select or avoid actions according to the consequences observed. Positive payoffs increase the probability of the strategy just chosen, and negative payoffs reduce this probability. Payoffs are seen as parameterizations of players' responses to their experiences. There are several models of reinforcement learning. A well-known model is that of Bush and Mosteller [13]; this is an aspiration-based reinforcement learning model where negative effects on the probability distribution over strategies are possible, and learning does not fade with time. The learners adapt by comparing their experience with an *aspiration* level. In our work we adapt this reinforcement learning model and we consider a simple aspiration scheme where aspiration is fixed by the workers and does not change during the evolutionary process.

The master reinforces its strategy as a function of the reputation calculated for each worker. Reputation has been widely considered in on-line communities that deal with untrustworthy entities, such as online auctions (e.g., eBay) or P2P file sharing sites (e.g., BitTorrent); it provides a mean of evaluating the degree of trust of an entity [23]. Reputation measures can be characterized in many ways, for example, as objective or subjective, centralized or decentralized. An objective measure comes from an objective assessment process while a subjective measure comes from the subjective belief that each evaluating entity has. In a centralized reputation scheme a central authority evaluates the entities by calculating the ratings received from each participating entity. In a decentralized system entities share their experience with other entities in a distributed manner. In our work, we use the master as a central authority that objectively calculates the reputation of each worker, based on its interaction with it; this centralized approach is also used by BOINC .

The BOINC system itself uses a form of reputation [7] for an optional policy called adaptive replication. This policy avoids replication in the event that a job has been sent to a highly reliable worker. The philosophy of this reputation scheme is to require a long time for the worker to gain a good reputation but a short time to lose it. Our proposed mechanism differs significantly from the one that is used in BOINC. One important difference is that we use auditing to check the validity of the worker's answers while BOINC uses only replication; in this respect, we have a more generic mechanism that also guarantees reliability of the system. Notwithstanding inspired by the way BOINC handles reputation we have designed a BOINC-like reputation type in our mechanism (called Type 3).

Sonnek et al. [36] use an adaptive reputation-based technique for task scheduling in volunteer setting (i.e., projects running BOINC). Reputation is used as a mechanism to reduce the degree of redundancy while keeping it possible for the master to verify the results by allocating more reliable nodes. In our work we do not focus on scheduling tasks to more reliable workers to increase reliability but rather we design a mechanism that forces the system to evolve to a reliable state. We also demonstrate several tradeoff between reaching a reliable state fast and master's cost. We have created a reputation function (called reputation Type 1) that is analogous to the reputation function used in [36] to evaluate this function's performance in our setting.

Aiyer et al. [2] introduce the BAR model to reason about systems with Byzantine (malicious), Altruistic, and Rational participants. They also introduce the notion of a protocol being BAR-tolerant, that is, the protocol is resilient

to both Byzantine faults and rational manipulation. As an application, they designed a cooperative backup service for P2P systems, based on a BAR-tolerant replicated state machine. Li et al [29] also considered the BAR model to design a P2P live streaming application based on a BAR-tolerant gossip protocol. Both works employ incentive-based game theoretic techniques (to remove the selfish behavior), but the emphasis is on building a reasonably practical system (hence, formal analysis is traded for practicality). Recently, Li et al [28] developed a P2P streaming application, called FlightPath, that provides a highly reliable data stream to a dynamic set of peers. FlightPath, as opposed to the above-mentioned BAR-based works, is based on mechanisms for *approximate equilibria* [14], rather than strict equilibria. In particular, ϵ-Nash equilibria are considered, in which rational players deviate if and only if they expect to benefit by more than a factor of ϵ. As the authors claim, the less restrictive nature of these equilibria enables the design of incentives to limit selfish behavior rigorously, while it provides sufficient flexibility to build practical systems. More recent works have considered other problems in the BAR model (e.g., data transfer [39]). Although the objectives and the model considered are different, our reputation-based mechanism can be considered, in some sense, to be BAR-tolerant.

Various surveys focus on the obstacles and challenges that current crowdsourcing approaches face [41, 24, 34]. One of the most crucial issues that crowdsourcing faces is the cheating behaviour of workers. In [43] reputation-based incentive protocols are presented. The analysis is done over the presence of many masters that are matched with workers. The task is assigned to a single worker and payments are ex-ante (i.e. the master pays before the worker performs the task). They design an algorithm that prevents workers from not putting any effort in performing the task (i.e. this can be perceived as cheating). Eickhoff and de Vries [17] investigate the nature and the causes for the cheating behavior of workers in crowdsourcing platforms. They categorize HITs in two categories: (1) *closed class questions* where the workers selects the correct answer from a limited list of options, and (2) *open class questions* where the answer is not a strict list and/or the task is of a creative nature. In their work they propose ways to discourage cheaters from performing HITs by transforming the task to be less appealing to them. Our approach also maintains a reputation for each worker (in order to know how reliable his answers are), but it does not discourage workers from participating. Instead, it tries to reinforce their good behavior to use them in future computations. We do not consider here the problem of matching masters and workers, which we assume solved, and focus on the problem with one master and multiple workers, all of them used by the master.

2 Model

In this section we characterize our model and we present the concepts of auditing, payoffs, rewards and aspiration. We also give a formal definition of the three reputation types used by our mechanism.

Master-Worker Framework We consider a master and a set W of n workers. The computation is broken into *rounds*, and in each round the master sends a task to the workers to compute and return the result. Based on the workers' replies, the master must decide which is the value most likely to be the correct result for this round. Crowdsourcing applications provide the possibility of selecting workers [4, 30].

Tasks From the perspective of crowdsourcing, tasks are closed class questions, whereas from the perspective of BOINC-operated applications, they are computations. These tasks have a unique solution; although such limitation reduces the scope of application of the presented mechanism [38], there are plenty of computations where the correct solution is unique: e.g., any mathematical function.

Worker types We consider three type of workers: *rational, altruistic* and *malicious*. Rational workers are selfish in a game-theoretic sense and their aim is to maximize their utility (benefit). In the context of this paper, a worker is *honest* in a round, when it truthfully computes and returns the correct result, and it *cheats* when it returns some incorrect value. Altruistic and malicious workers have a predefined behavior, to always be honest or cheat, respectively. Instead, a rational worker decides to be honest or cheat depending on which strategy maximizes its utility. We denote by $p_{Ci}(r)$ the probability of a rational worker i cheating in round r. This probability is not fixed and the worker adjusts it over the course of the computation. The master is not aware of the worker types, neither of a distribution of types (our mechanism does not rely on any statistical information).

Type 1: $\rho_i(r) = (v_i(r) + 1)/(aud(r) + 2)$.
Type 2: $\rho_i(r) = \varepsilon^{aud(r)-v_i(r)}$, for $\varepsilon \in (0, 1)$, when $aud(r) > 0$, and $\rho_i(r) = 1/2$, otherwise.
Type 3: Here we define $\beta_i(r)$ as the error rate of worker i at round r. Reputation for this type is calculated as follows:

$$\beta(r) = \begin{cases} 0.1, & \text{if } r = 0. \\ 0.95\beta(r-1), & \text{if } r > 0 \text{ and worker is truthful in round } r. \\ \beta(r-1) + 0.1, & \text{otherwise.} \end{cases}$$

$$\rho(r) = \begin{cases} 0.5, & \text{if } r = 0. \\ 0, & \text{if } \beta(r) > 0.05. \\ 1 - \sqrt{\frac{\beta(r)}{0.05}}, & \text{otherwise.} \end{cases}$$

Fig. 1: Reputation types.

While workers make their decision individually and with no coordination, following [31] and [8], we assume that all the workers that cheat in a round return the same incorrect value; this yields a worst case scenario (and hence analysis) for the master with respect to obtaining the correct result using mechanisms where the result is the outcome of voting. It subsumes models where cheaters do not necessarily return the same answer. (This can be seen as a weak form of collusion.)

For simplicity, unless otherwise stated, we assume that workers do not change their type over time. In practice it is possible that changes occur. For example, a rational worker might become malicious due to a bug, or a malicious worker (e.g., a worker under the influence of a virus) become altruistic (e.g., if an antivirus software reinstates it). If this may happen, then all our results still apply for long enough periods between two changes. (In our simulation study we do consider scenarios where the workers change their type dynamically.)

Auditing, Payoffs, Rewards and Aspiration To induce the rational workers to be honest, the master employs, when necessary, *auditing* and *reward/punish* schemes. The master, in a round, might decide to audit the response of the workers, at a cost. In this work, auditing means that the master computes the task by itself, and checks which workers have been honest. We denote by $p_{\mathcal{A}}(r)$ the probability of the master auditing the responses of the workers in round r. The master can change this auditing probability over the course of the computation, but restricted to a minimum value $p_{\mathcal{A}}^{min} > 0$. When the master audits, it can accurately reward and punish workers. When the master does not audit, it rewards only those in the weighted majority (see below) of the replies received and punishes no one.

In this work we consider three worker payoff parameters: (a) $WP_{\mathcal{C}}$: worker's punishment for being caught cheating, (b) $WC_{\mathcal{T}}$: worker's cost for computing a task, and (c) $WB_{\mathcal{Y}}$: worker's benefit (typically payment) from the master's reward. Also, following [13], we assume that, in every round, a worker i has an *aspiration* a_i: the minimum benefit it expects to obtain in a round. In order to motivate the worker to participate in the computation, the master usually ensures that $WB_{\mathcal{Y}} \geq a_i$; in other words, the worker has the potential of its aspiration to be covered. We assume that the master knows the aspirations. Finally, we assume that the master has the freedom of choosing $WB_{\mathcal{Y}}$ and $WP_{\mathcal{C}}$ with goal of eventual correctness.

Eventual Correctness The goal of the master is to eventually obtain a reliable computational platform: After some finite number of rounds, the system must guarantee that the master obtains the correct task results in every round with probability 1 and audits with probability $p_{\mathcal{A}}^{min}$. We call such property *eventual correctness*.

Reputation The reputation of each worker is measured by the master; a centralized reputation mechanism is used. In fact, the workers are unaware that a reputation scheme is in place, and their interaction with the master does not reveal any information about reputation; i.e., the payoffs do not depend on a worker's reputation.

In this work, we consider three reputation metrics. The first one, called *Type 1* is analogous to a reputation metric used in [36] and the third one, called *Type 3* is inspired by BOINC [7]. We also define our own type called *Type 2* that is not influenced by any other reputation type, and as we show in Section 4 it possesses beneficial properties. In all types, the reputation of a worker is determined based on the number of times it was found truthful. Hence, the master may update the reputation of the workers only when it audits. We denote by $aud(r)$ the number of rounds the master audited up to round r, and by $v_i(r)$ we refer to the number of auditing rounds in which worker i was found truthful up to round r. We let $\rho_i(r)$ denote the *reputation* of worker i after round r, and for a given set of workers $Y \subseteq W$ we let $\rho_Y(r) = \sum_{i \in Y} \rho_i(r)$ be the aggregated reputation of the workers in Y, by aggregating we refer to summing the reputation values. Then, the reputation types we consider are detailed in Figure 1.

These reputation types satisfy the natural properties that if a worker is honest when the master audits, then the reputation of the worker cannot decrease. Conversely, if a worker cheats when the master audits, the reputation of the worker cannot increase. (The proofs of these facts can be found in Appendix A.)

In each round, when the master *does not audit*, the result is obtained from the *weighted majority* as follows. Consider a round r. Let $F(r)$ denote the subset of workers that returned an incorrect result, i.e., the rational workers who chose to cheat plus the malicious ones; recall that we assume as a worst case that all cheaters return the same value. Then, $W \setminus F(r)$ is the subset of workers that returned the correct value, i.e., the rational workers who chose to be truthful plus the altruistic ones. Then, if $\rho_{W \setminus F(r)}(r) > \rho_{F(r)}(r)$, the master will accept the correct value, otherwise it will accept an incorrect value. The mechanism, presented in the next section, employs auditing and appropriate incentives so that rational workers become truthful with high reputation, while malicious workers (alternatively altruistic workers) end up having very low (altr. very high) reputation after a few auditing rounds.

3 Reputation-based Mechanism

We now present our reputation-based mechanism. The mechanism is composed by an algorithm run by the master and an algorithm run by each worker.

Master's Algorithm The algorithm begins by choosing the initial probability of auditing and the initial reputation (same for all workers). The initial probability of auditing will be set according to the information the master has about the environment (e.g., workers' initial p_C). For example, if it has no information about the environment, a possibly safe approach is to initially set $p_{\mathcal{A}} = 0.5$. The master also chooses the reputation type to use (e.g., Type 1, 2 or 3).

After that, at each round, the master sends a task to all workers and, when all answers are received, the master audits the answers with probability $p_{\mathcal{A}}$. In the case the answers are not audited, the master accepts the value returned by the weighed majority, and continues to the next round with the same probability of auditing and the same reputation values for each worker. In the case the answers are audited, the value $p_{\mathcal{A}}$ of the next round is reinforced (i.e., modified according to the accumulated reputation of the cheaters) and the reputations of the workers are updated based on their responses. Then, the master rewards/penalizes the workers appropriately. Specifically, if the master audits and a worker i is a cheater (i.e., $i \in F$), then $\Pi_i = -WP_{\mathcal{C}}$; if i is honest, then $\Pi_i = WB_{\mathcal{Y}}$. If the master does not audit, and i returns the value of the weighted majority (i.e., $i \in W_m$), then $\Pi_i = WB_{\mathcal{Y}}$, otherwise $\Pi_i = 0$.

We include a threshold, denoted by τ, that represents the master's *tolerance* to cheating (typically, we will assume $\tau = 1/2$ in our simulations). If the ratio of the aggregated reputation of cheaters with respect to the total is larger than τ, $p_{\mathcal{A}}$ is increased, and decreased otherwise. The amount by which $p_{\mathcal{A}}$ changes depends on the difference between these values, modulated by a *learning rate* α_m. This latter value determines to what extent the newly acquired information will override the old information. (For example, if $\alpha_m = 0$ the master will never adjust $p_{\mathcal{A}}$.) A pseudocode of the algorithm described is given as Algorithm 1.

Workers' Algorithm This algorithm is run only by rational workers (recall that altruistic and malicious workers have a predefined behavior).[1] The execution of the algorithm begins with each rational worker i deciding an initial probability of cheating p_{Ci}. In each round, each worker receives a task from the master and, with probability $1 - p_{Ci}$ computes the task and replies to the master with the correct answer. Otherwise, it fabricates an answer, and sends the incorrect

[1] Since the workers are not aware that a reputation scheme is used, this algorithm is the one considered in [16]; we describe it here for self-containment.

ALGORITHM 1: Master's Algorithm

$p_{\mathcal{A}} \leftarrow x$, *where* $x \in [p_{\mathcal{A}}^{min}, 1]$
$aud = 0$
// initially all workers have the same reputation
$\forall i \in W : v_i = 0; \beta_i = 0.1; \rho_i = 0.5$
for $r \leftarrow 1$ **to** ∞ **do**
  send *a task* T *to all workers in* W
  upon *receiving all answers* **do**
    audit *the answers with probability* $p_{\mathcal{A}}$
    if *the answers were not audited* **then**
      // weighted majority, coin flip in case of a tie
      accept *the value returned by workers in* $W_m \subseteq W$,
          where $\rho_{W_m} > \rho_{W \setminus W_m}$
    else *// the master audits*
      $aud \leftarrow aud + 1$
      Let $F \subseteq W$ *be the set of workers that cheated.*
      $\forall i \in W$:
        // honest workers
        if $i \notin F$ **then** $v_i \leftarrow v_i + 1$ *or* $\beta_i \leftarrow \beta_i \cdot 0.95$
        // cheater workers
        else $v_i \leftarrow v_i$ *or* $\beta_i \leftarrow \beta_i + 0.1$
        update *reputation* ρ_i *of worker* i *as defined by reputation type used*
      if $\rho_W = 0$ **then** $p_{\mathcal{A}} \leftarrow \min\{1, p_{\mathcal{A}} + \alpha_m\}$ **else**
        $p_{\mathcal{A}} \leftarrow \min\{1, \max\{p_{\mathcal{A}}^{min}, p_{\mathcal{A}} + \alpha_m(\frac{\rho_F}{\rho_W} - \tau)\}\}$
    $\forall i \in W$: **return** *payoff* Π_i *to worker* i

ALGORITHM 2: Algorithm for Rational Worker i

$p_{Ci} \leftarrow y$, *where* $y \in [0, 1]$
for $r \leftarrow 1$ **to** ∞ **do**
  receive *a task* T *from the master*
  $S_i \leftarrow -1$ *with probability* p_{Ci},
  and $S_i \leftarrow 1$ *otherwise*
  if $S_i = 1$ **then**
    $\sigma \leftarrow compute(T)$,
  else
    $\sigma \leftarrow$ *arbitrary solution*
  send *response* σ *to the master*
  get payoff Π_i
  if $S_i = 1$ **then**
    $\Pi_i \leftarrow \Pi_i - WC_{\mathcal{T}}$
  $p_{Ci} \leftarrow \max\{0, \min\{1, p_{Ci} - \alpha_w(\Pi_i - a_i)S_i\}\}$

response to the master. We use a flag S_i to model the stochastic decision of a worker i to cheat or not. After receiving its payoff, each worker i changes its p_{Ci} according to payoff Π_i, the chosen strategy S_i, and its aspiration a_i.

The workers have a *learning rate* α_w. In this work, we assume that all workers have the same learning rate, that is, they learn in the same manner (see also the discussion in [37]; the learning rate is called step-size there); note that our analysis can be adjusted to accommodate also workers with different learning rates. A pseudocode of the algorithm is given as Algorithm 2.

4 Analysis

We now analyze the reputation-based mechanism. We model the evolution of the mechanism as a Markov Chain, and then discuss the necessary and sufficient conditions for achieving eventual correctness. Modeling a reputation-based mechanism as a Markov Chain is more involved than previous models that do not consider reputation (e.g. [16]).

The Markov Chain Let the state of the Markov chain be given by a vector s. The components of s are: for the master, the probability of auditing $p_{\mathcal{A}}$ and the number of audits before state s, denoted as aud; and for each *rational* worker i, the probability of cheating p_{Ci}, the number of *validations* (i.e., the worker was honest when the master audited) before state s, denoted as v_i, and the error rate β_i. To refer to any component x of vector s we use $x(s)$. Then,

$$s = \big\langle p_{\mathcal{A}}(s), aud(s), p_{C1}(s), p_{C2}(s), \ldots, p_{Cn}(s), v_1(s), v_2(s), \ldots, v_n(s), \beta_1(s), \beta_2(s), \ldots, \beta_n(s) \big\rangle.$$

In order to specify the transition function, we consider the execution of the protocol divided in rounds. In each round, probabilities and *counts* (i.e. numbers of validations and audits) are updated by the mechanism as defined in Algorithms 1 and 2. The state at the end of round r is denoted as s_r. Abusing the notation, we will use $x(r)$ instead of $x(s_r)$ to denote component x of vector s_r. The workers' decisions, the workers' error rate, the number of cheaters, and the payoffs of each round $r > 0$ are the stochastic outcome of the probabilities and counts at the end of round $r-1$. We specify the transition from s_{r-1} to s_r by the actions taken by the master and the workers during round r.

In the definition of the transition function that follows, the probabilities are limited to $p_{\mathcal{A}}(s) \in [p_{\mathcal{A}}^{min}, 1]$ and for each rational worker i to $p_{Ci}(s) \in [0,1]$, for any state s. The initial state s_0 is arbitrary but restricted to the same limitations. Let $P_F(r)$ be the probability that the set of cheaters in round r is exactly $F \subseteq W$. (That is, $P_F(r) = \prod_{j \in F} p_{Cj}(r-1) \prod_{k \notin F} (1 - p_{Ck}(r-1))$.) Then, the transition from state s_{r-1} to s_r is as follows.

- Malicious workers always have $p_C = 1$ and altruistic workers always have $p_C = 0$.
- With probability $p_{\mathcal{A}}(r-1) \cdot P_F(r)$, the master audits when the set of cheaters is F. Then, according to Algorithms 1 and 2, the new state is as follows.

  For the master: $aud(r) = aud(r-1)+1$ and, if $\rho_W(r) > 0$ then $p_{\mathcal{A}}(r) = p_{\mathcal{A}}(r-1) + \alpha_m\,(\rho_F(r)/\rho_W(r) - \tau)$ and $p_{\mathcal{A}}(r) = \min\{1, p_{\mathcal{A}} + \alpha_m\}$ otherwise.

  (1) For each worker $i \in F$: $v_i(r) = v_i(r-1)$ and $\beta_i(r) = \beta_i(r-1) + 0.1$ and, if i is rational, then $p_{Ci}(r) = p_{Ci}(r-1) - \alpha_w(a_i + WP_{\mathcal{C}})$.

  (2) For each worker $i \notin F$: $v_i(r) = v_i(r-1) + 1$ and $\beta_i(r) = \beta_i(r-1) \cdot 0.95$ and, if i is rational, then $p_{Ci}(r) = p_{Ci}(r-1) + \alpha_w(a_i - (WB_{\mathcal{Y}} - WC_{\mathcal{T}}))$.
- With probability $(1 - p_{\mathcal{A}}(r-1))P_F(r)$, the master does not audit when the set of cheaters is F. Then, according to Algorithms 1 and 2, the following updates are carried out.

  For the master: $p_{\mathcal{A}}(r) = p_{\mathcal{A}}(r-1)$ and $aud(r) = aud(r-1)$.

  For each worker $i \in W$: $v_i(r) = v_i(r-1)$.

  For each rational worker $i \in F$,

  (3) if $\rho_F(r) > \rho_{W \setminus F}(r)$ then $p_{Ci}(r) = p_{Ci}(r-1) + \alpha_w(WB_{\mathcal{Y}} - a_i)$,

  (4) if $\rho_F(r) < \rho_{W \setminus F}(r)$ then $p_{Ci}(r) = p_{Ci}(r-1) - \alpha_w \cdot a_i$,

  For each rational worker $i \notin F$,

  (5) if $\rho_F(r) > \rho_{W \setminus F}(r)$ then $p_{Ci}(r) = p_{Ci}(r-1) + \alpha_w(a_i + WC_{\mathcal{T}})$,

  (6) if $\rho_F(r) < \rho_{W \setminus F}(r)$ then $p_{Ci}(r) = p_{Ci}(r-1) + \alpha_w(a_i - (WB_{\mathcal{Y}} - WC_{\mathcal{T}}))$.

Recall that in case of a tie in the weighted majority, the master flips a coin to choose one of the answers, and assigns payoffs accordingly. If that is the case, transitions (3)–(6) apply according to that outcome.

Conditions for Eventual Correctness We show now the conditions under which the system can guarantee eventual correctness. The analysis is carried out for a universal class of reputation functions characterized by two properties. Property 1 states that if the master audits in consecutive rounds, the aggregated reputation of the honest workers will be larger than that of malicious workers in a bounded number of rounds. Property 2 states that if the aggregated reputation of a set $X \subset W$ is larger than that of a set $Y \subset W$, then it remains so if the master audits and all workers are honest. The two properties are formally stated below.

Property 1: For any constant $\delta > 0$, there is a bounded value $\gamma(\delta)$ such that, for any non-empty $X \subseteq W$ and any initial state s_r in which $v_i(r) = 0, \forall i \notin X$, if the Markov chain evolves in such a way that $\forall k = 1, \ldots, \gamma(\delta)$, it holds that $aud(r+k) = aud(r) + k$, $\forall i \in X, v_i(r+k) = v_i(r) + k$ and $\forall j \in W \setminus X, v_j(r+k) = v_j(r)$, then $\rho_X(r + \gamma(\delta)) > \delta \cdot \rho_{W \setminus X}(r + \gamma(\delta))$.

Property 2: For any $X \subset W$ and $Y \subset W$, if $aud(r+1) = aud(r)+1$ and $\forall j \in X \cup Y$ it is $v_j(r+1) = v_j(r)+1$ then $\rho_X(r) > \rho_Y(r) \Rightarrow \rho_X(r+1) > \rho_Y(r+1)$.

Reputation Type 1 and Type 2 (cf. Section 2) satisfy Property 1, while reputation Type 3 as defined does not. However, if a constant upper bound in the value of $\beta_i(r)$ is established, we obtain an adapted version of Type 3 reputation that satisfies Property 1. Regarding Property 2, while reputation Type 2 satisfies it, reputation Type 1 and 3 do not. (The proofs of these facts can be found in Appendix B.) As we show below, this *makes a difference* with respect to guaranteeing eventual correctness.

The following terminology will be used throughout. For any given state s, a set X of workers is called a *reputable set* if $\rho_X(r) > \rho_{W \setminus X}(r)$. In any given state s, let a worker i be called an *honest worker* if $p_{Ci}(s) = 0$. Let a state s be called a *truthful state* if the set of honest workers in state s is reputable. Let a *truthful set* be any set of truthful states. Let a worker be called a *covered worker* if the payoff of returning the correct answer is at least its aspiration plus the computing cost. I.e., for a covered worker i, it is $WB_{\mathcal{Y}} \geq a_i + WC_{\mathcal{T}}$. We refer to the opposite cases as *uncovered worker* ($WB_{\mathcal{Y}} < a_i + WC_{\mathcal{T}}$), *cheater worker* ($p_{Ci}(s) = 1$), *untruthful state* (the set of cheaters in that state is reputable), and *untruthful set*, respectively. Let a set of states S be called *closed* if, once the chain is in any state $s \in S$, it will not move to any state $s' \notin S$. (A singleton closed set is called an *absorbing* state.) For any given set of states S, we say that the chain *reaches* (resp. *leaves*) the set S if the chain reaches some state $s \in S$ (resp. reaches some state $s \notin S$).

In the master's algorithm, a non-zero probability of auditing is always guaranteed. This is a necessary condition. Otherwise, unless the altruistic workers outnumber the rest, a closed untruthful set is reachable, as we show in Lemma 1.

Lemma 1. *Consider any set of workers $Z \subseteq W$ such that $WB_{\mathcal{Y}} > a_i$, for every rational worker $i \in Z$. Consider the set of states*

$$S = \{s | (p_{\mathcal{A}}(s) = 0) \wedge (\forall w \in Z : p_{Cw}(s) = 1) \wedge (\rho_Z(s) > \rho_{W-Z}(s))\}.$$

Then,

(i) *S is a closed untruthful set, and*
(ii) *if $p_{\mathcal{A}}(0) = 0$, $\rho_Z(0) > \rho_{W-Z}(0)$, and for all $i \in Z$ it is $p_{Ci}(0) > 0$, then, S is reachable.*

Proof. (i) Each state in S is untruthful, since the workers in Z are all cheaters and Z is a reputable set. Since $p_{\mathcal{A}} = 0$, the master never audits, and the reputations are never updated. From transition (3) it can be seen that, if the chain is in a state of the set S before round r, for each worker $i \in Z$, it holds $p_{Ci}(r) \geq p_{Ci}(r-1) = 1$. Hence, once the chain has reached a state in the set S, it will move only to states in the set S. Thus, S is a closed untruthful set.

(*ii*) We show now that S is reachable from the initial state under the above conditions. Because $p_{\mathcal{A}}$ and the reputations only change when the master audits, we have that $p_{\mathcal{A}}(0) = 0 \Longrightarrow p_{\mathcal{A}}(s) = 0$ and $\rho_Z(0) > \rho_{W-Z}(0) \Longrightarrow \rho_Z(s) > \rho_{W-Z}(s)$, for any state s. Malicious workers always have $p_C = 1$, and no altruistic worker may be contained in Z because $p_{Ci}(0) > 0$ for all $i \in Z$. Thus, to complete the proof it is enough to show that eventually it is $p_C = 1$ for all the workers in L, which is the set of rational workers in Z. Given that for each rational worker $j \in L$, $p_{Cj}(0) > 0$ and $WB_{\mathcal{Y}} > a_j$, from transition (3) it can be seen that there is a non-zero probability of moving from s_0 to a state s_1 where the same conditions apply and $p_{Cj}(1) > p_{Cj}(0)$ for each rational worker $j \in L$. Hence, applying the argument inductively, there is a non-zero probability of reaching S.

Eventual correctness follows if we can show that the Markov chain always ends in a closed truthful set, with $p_{\mathcal{A}} = p_{\mathcal{A}}^{min}$. We prove first that having at least one worker that is altruistic or covered rational is necessary for a closed truthful set to exist. Then we prove that it is also sufficient if all rational workers are covered.

Lemma 2. *If all workers are malicious or uncovered rationals, no truthful set S is closed, if the reputation type satisfies Property 2.*

Proof. Let us consider some state s of a truthful set S. Let Z be the set of honest workers in s. Since s is truthful , then Z is reputable. Since there are no altruistic workers, the workers in Z must be uncovered rational. Let us assume that being in state s the master audits in round r. From Property 2, since all nodes in Z are honest in r, Z is reputable after r. From transition (2), after round r, each worker $i \in Z$ has $p_{Ci}(r) > 0$. Hence, the new state is not truthful , and S is not closed.

Lemma 3. *Consider a reputation type that satisfies Properties 1 and 2. If all rational workers are covered and at least one worker is altruistic or rational, a closed truthful set S is reachable from any initial state. Moreover, in every state $s \in S$, $p_{\mathcal{A}}(s) = p_{\mathcal{A}}^{min}$.*

Proof. Let X be the set of altruistic and rational workers, and consider any initial state s_r. Let us define a constant $\delta = \max\{1, (1 - \tau + \eta/\alpha_m)/(\tau + \eta/\alpha_m)\}$, for a fixed constant $\eta \in (0, \tau\alpha_m)$. We consider the following cases.

1. In state s_r not all the workers in X are truthful. Let us assume then that in the next $\lceil \frac{1}{\alpha_w(a_j - WP_C)} \rceil$ rounds the master audits and any worker i that has $p_{Ci} > 0$ in the round cheats. Then, from transition (1) and the fact that all rational workers are covered, after these $\lceil \frac{1}{\alpha_w(a_j - WP_C)} \rceil$ rounds all the workers in X are truthful. Then, we end up in one of the following three cases.
2. In state s_r all the workers in X are truthful, and $\rho_X(r) \leq \delta \cdot \rho_{W \setminus X}(r)$. Consider the value $\gamma(\delta)$ given in Property 1. Assume that in each of the following $\gamma(\delta)$ rounds the master audits. The workers in $W \setminus X$ are malicious, hence, it holds that $\forall \notin X : v_i(r) = 0$. Then, in these rounds all workers in X are honest (every worker in X remains truthful from transition (2) and the fact that all rational workers are covered) and all workers in $W \setminus X$ cheat because they are malicious. Therefore, it holds that $\forall i \notin X : \forall j \in [r, r + \gamma(\delta)] : v_i(j) = 0$. Then, from Property 1, after the $\gamma(\delta)$ rounds we have that $\rho_X(r + \gamma(\delta)) > \delta \cdot \rho_{W \setminus X}(r + \gamma(\delta))$. Then, we are in one of the following two cases.
3. In state s_r all the workers in X are truthful, $\rho_X(r) > \delta \cdot \rho_{W \setminus X}(r)$, and $p_{\mathcal{A}}(r) > p_{\mathcal{A}}^{min}$. Let us assume that in the next $\lceil p_{\mathcal{A}}(r)/\eta \rceil$ rounds the master audits. Then, as in the previous case, in these rounds all workers in X are honest and all workers in $W \setminus X$ cheat. Hence, the property that $\rho_X(r + k) > \delta \cdot \rho_{W \setminus X}(r + k)$ holds for each round $r + k$, for $k = 1, \ldots, \lceil p_{\mathcal{A}}(r)/\eta \rceil$. Then, by the definition of δ and the update of $p_{\mathcal{A}}$, in each round $p_{\mathcal{A}}$ is decremented by η (more precisely, by $\min\{\eta, p_{\mathcal{A}}\}$). Hence, by round $r + \lceil p_{\mathcal{A}}(r)/\eta \rceil$ it holds that $p_{\mathcal{A}} = p_{\mathcal{A}}^{min}$. Then, we are in the following case.
4. In state s_r all the workers in X are truthful, $\rho_X(r) > \delta \cdot \rho_{W \setminus X}(r)$, and $p_{\mathcal{A}}(r) = p_{\mathcal{A}}^{min}$. Then, all subsequent states satisfy all these properties, and define the set S, independently of whether the master audits or not (from transition (2) and (6), the fact that $\delta \geq 1$, Property 2, and the update of $p_{\mathcal{A}}$). This complete the proof.

Now, combining Lemmas 2 and 3 we obtain the following theorem.

Theorem 1. *In a system where (1) the reputation type used satisfies Properties 1 and 2, and (2) all rational workers are covered, having at least one altruist or rational worker is a necessary and sufficient condition to guarantee eventual correctness. That is, from any initial state, to eventually reach a closed truthful set S where the master audits with probability $p_{\mathcal{A}}^{min}$.*

If there is no knowledge on the distribution of the workers among the three types (altruistic, malicious and rationals), the only strategy to make sure eventual correctness is achieved, if possible, is to cover all workers. Of course, if *all* workers are malicious there is no possibility of reaching eventual correctness.

5 Simulations

This section complements our analytical results with illustrative simulations. The graphical representation of the data obtained captures the tradeoffs between reliability and cost and among all three reputation types, concepts not visible through the analysis. Here we present simulations for a variety of parameter combinations similar to the values observed in real systems (extracted from [3, 18]). We have designed our own simulation setup by implementing our

mechanism (the master's and the workers' algorithms, including the three types of reputation discussed above) using C++. The simulations were contacted on a dual-core AMD Opteron 2.5GHz processor, with 2GB RAM running CentOS version 5.3. We consider a total of 9 workers (that will be rational, altruistic or malicious in the different experiments) as an appropriate workforce, compared to SETI-like systems using three workers. The figures represent the average over 10 executions of the implementation, unless otherwise stated (when we show the behavior of typical, individual realizations). The chosen parameters are indicated in the figures. Note that, for simplicity, we consider that all workers have the same aspiration level $a_i = 0.1$, although we have checked that with random values the results are similar to those presented here, provided their variance is not very large. We consider the same learning rate for the master and the workers, i.e., $\alpha = \alpha_m = \alpha_w = 0.1$. Note that the learning rate, as discussed for example in [37] (called step-size there), is generally set to a small constant value for practical reasons. Finally we set $\tau = 0.5$ (which means that the master will not tolerate a majority of cheaters), $p_{\mathcal{A}}^{min} = 0.01$ and $\varepsilon = 0.5$ in reputation Type 2.

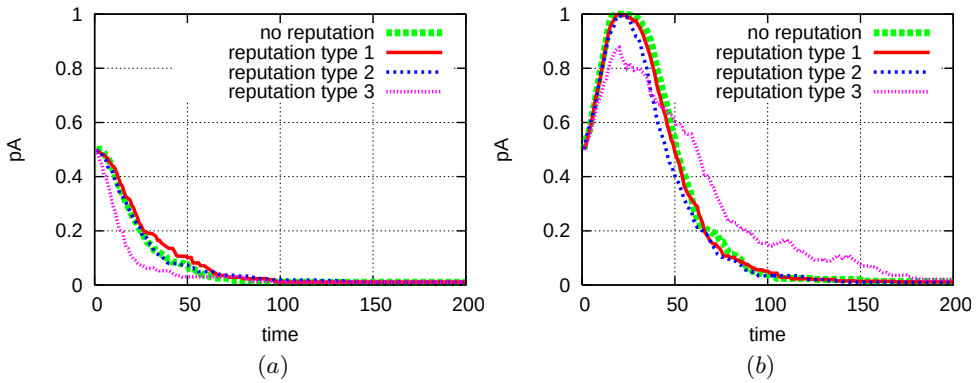
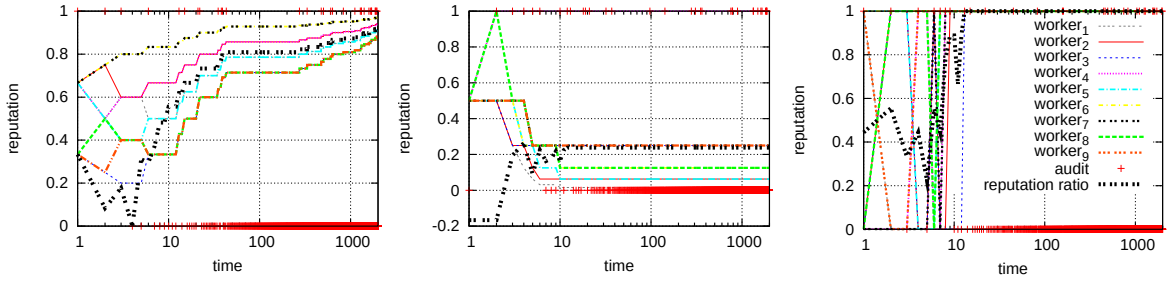


Fig. 2: Rational workers. Auditing probability of the master as a function of time (number of rounds) for parameters $p_{\mathcal{A}} = 0.5$, $\alpha = 0.1$, $a_i = 0.1$, $WB_{\mathcal{Y}} = 1$, $WP_{\mathcal{C}} = 0$ and $WC_{\mathcal{T}} = 0.1$. (a) initial $p_C = 0.5$ (b) initial $p_C = 1$.

The contents of this section can be summarized as follows: In the next paragraph we present results considering only rational workers and, subsequently, results involving all three type of workers. We continue with a discussion on the number of workers that if covered, eventual correctness is reached, and how the choice of reputation affects this. Finally, we show that our mechanism is robust even in the event of having workers changing their behavior (e.g., rational workers becoming malicious due to software or hardware errors).

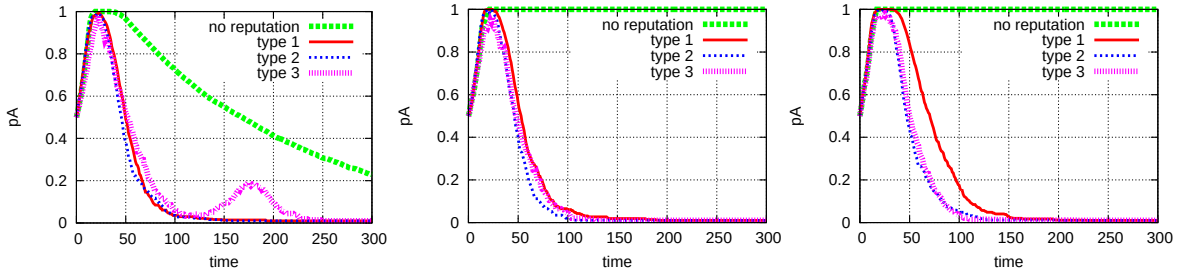
Rational workers While the main reason for introducing reputation was to cope with malicious workers, as a first step we checked whether reputation improves the algorithm performance for rational workers only. In this case as we see in Figure 2, the first two reputation types give similar results as in the case of no reputation. Reputation Type 3, on the other hand, seems to perform better in the case that the initial $p_C = 0.5$, while in the case of $p_C = 1$ the system has a slower convergence rate, but the auditing probability at the first 50 rounds is lower. This has to do with the fact that in Type 3 the reputation of a worker is reinforced indirectly, what is directly reinforced depending on the workers honesty is the error rate. Our observations in Figure 2 reveals an interesting tradeoff: depending on whether the master has information on the workers' initial behavior or on the auditing that is willing to perform it will choose to use or not reputation Type 3. From Figure 2 we can also see that the mechanism of [16] is enough to bring rational workers to produce the correct output, precisely because of their rationality. Although Figure 2 depicts the $p_{\mathcal{A}}$ of the master and not the p_C of the workers we have observed (see Figure 8 in Appendix C) that for all the initial p_C studied, by the time the master's auditing probability reaches p_A^{min}, the system had already reached eventual correctness.

Figure 3 allows to compare the behavior of the three reputation types, with reputation ratio defined as $\sum_{i \in W} \rho_i S_i / |W|$, a quantity that will prove helpful later to understand how different reputation schemes work. Reputation Type 1 leads rational workers to reputation values close to 1 (at a rate that depends on the value of the initial p_C). However, when Type 2 is applied reputation takes values between (0,0.3). This happens because when the master catches a worker



(a) (b) (c)

Fig. 3: Rational workers, for an individual realization with initially $p_C = 0.5$, $p_{\mathcal{A}} = 0.5, WB_{\mathcal{Y}} = 1$, $WC_{\mathcal{T}} = 0.1$, $WP_{\mathcal{C}} = 0$, $\alpha = 0.1$ and $a_i = 0.1$. Left, reputation Type 1. Middle, reputation Type 2. Right, reputation Type 3.



(a) (b) (c)

Fig. 4: Master's auditing probability as a function of time in the presence of rational and malicious workers. Parameters in all plots, rationals' initial $p_C = 1$, master's initial $p_{\mathcal{A}} = 0.5, WB_{\mathcal{Y}} = 1$, $WC_{\mathcal{T}} = 0.1$, $WP_{\mathcal{C}} = 0$ and $\alpha = 0.1$, $a_i = 0.1$. In (a) 4 malicious and 5 rationals, (b) 5 malicious and 4 rationals , (c) 8 malicious and 1 rational.

cheating, its reputation decreases exponentially, never increasing again. Having never increasing reputation is not a problem for our mechanism because we use the reputation as a relative value between sets of workers. Reputation Type 3, on the other hand, allows for dramatic increases and decreases of reputation. This is a result of the indirect way we calculate reputation Type 3, as we mentioned above.

Different types of workers We now move to our main case of interest and include different types of workers in our experiments. Figure 4 shows results for the extreme case, with malicious workers, no altruistic workers, and rational workers that initially cheat with probability $p_C = 1$. We observe that if the master does not use reputation and a majority of malicious workers exist, then the master is enforced by the mechanism to audit in every round. Even with a majority of rational workers, it takes a long time for the master to reach $p_{\mathcal{A}}^{min}$, if reputation is not used. Introducing reputation can indeed cope with the challenge of having a majority of malicious workers, except (obviously) when all workers are malicious. For Type 1, the larger the number of malicious workers, the slower the master reaches $p_{\mathcal{A}}^{min}$. On the contrary, the time to convergence to the $p_{\mathcal{A}}^{min}$ is independent of the number of malicious workers for reputation Type 2. This is due to the different dynamical behavior of the two reputations discussed above. For reputation Type 3, if a majority of rationals exists then convergence is slower. This is counter-intuitive, but as we mentioned before it is linked to the way reputation and error rate are calculated. On the other hand, with Type 3, $p_{\mathcal{A}}$ is slightly lower in the first rounds. We thus conclude that reputation Type 2 gives better results, as long as at least one rational worker exists and the master is willing to audit slightly more in the first rounds. We have checked, see Figure 5, that if rational workers are replaced by altruistic ones, the performance of the two reputation schemes improves, as expected.

Covering only a subset of rational workers In the previous paragraphs we considered only cases where the master was covering all workers, that is, $WB_{\mathcal{Y}} > a + WC_{\mathcal{T}}$ for all workers. For the case with malicious workers, as explained in Section 4, this is unavoidable if the worker type distribution is not known. But if we know that only rational workers

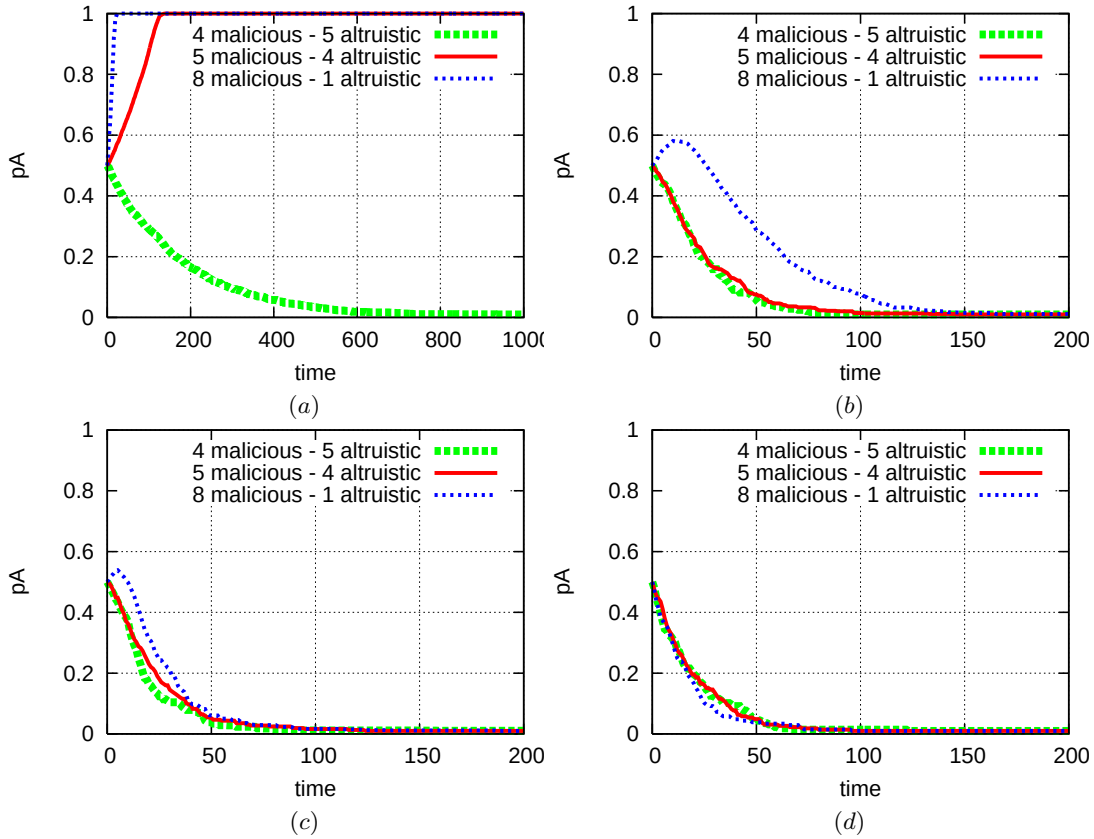


Fig. 5: Master's auditing probability as a function of time in the presence of altruistic and malicious workers. Parameters in all plots, master's initial $p_{\mathcal{A}} = 0.5$, $WC_{\mathcal{T}} = 0.1$, $WP_{\mathcal{C}} = 0$ and $\alpha = 0.1$, $a_i = 0.1$. In (a) master does not use reputation, (b) master uses reputation Type 1, (c) master uses reputation Type 2, (d) master uses reputation Type 3.

exist then maybe by covering only a subset of them the system can reach eventual correctness, a scenario that we now explore. In Figure 6(a) the extreme case of only one covered worker is presented; in Figure 7 less extreme cases are presented pointing to the same results). We see that with reputation Type 1 our system does not converge. The master tolerates a significant percentage of cheaters (since $\tau = 0.5$), creating a very unstable system, where the probability of the master receiving the correct answer varies greatly. This occurs because by tolerating more cheaters, the master creates a system where the cheating probability of the uncovered workers spikes between zero and one (see Figure 9 in Appendix C). We have found that introducing punishment or reducing the tolerance do not fix this (see Figure 10 in Appendix C). Basically, the reason is that the reputation of honest workers does not always exceed the reputation of cheaters, as indicated by the reputation ratio. In fact, we have checked that for the covered worker, p_C vanishes eventually, but for uncovered workers this is not the case: their p_C takes values usually below 0.6 but close to that value. Given that uncovered workers' p_C is greater than zero, it may occur that the master does not audit and a number of uncovered workers, with reputation higher than the rest, cheat. Because of this, even if the master maintains a high auditing probability, eventual correctness is not guaranteed.

As Figures 7(a3),(b3) and (c3) and 6(a) show, for reputation Type 3 our system does not converge, since reputation Type 3 does not satisfy Property 2. What is interesting in Type 3 is that, by decreasing tolerance (see Figure 7(a3)), results are actually worse than increasing tolerance (see Figure 7(c3)). This is a result of calculating $p_{\mathcal{A}}$ based on reputation; in Type 3 reputation in linked to error rate which is the value that is evolving based on the truthfulness of the worker (i.e, in Type 3, the reputation of a worker is only evolving indirectly). Type 3 gives even worse results, see Figures 9 and 11, than Type 1, since not even the covered workers p_C vanishes to zero eventually.

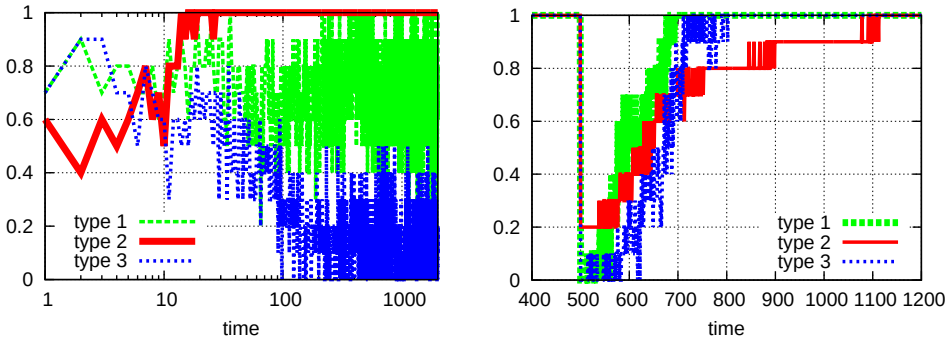


Fig. 6: Correct reply rate as a function of time. Parameters are initial $p_{\mathcal{A}} = 0.5, WB_{\mathcal{Y}} = 1$, $WC_{\mathcal{T}} = 0.1$, $WP_{\mathcal{C}} = 0$ and $\alpha = 0.1$, $a_i = 0.1$. (a) Reputation types 1 and 3 have initial $p_C = 0.5$, while in Type 2, $p_C = 1$. (b) initial $p_C = 1$.

Figure 6(a) shows that our system, in the simulated experiments, always reach eventual correctness using reputation Type 2. This does not come as a complete surprise since reputation Type 2 is an "unforgiving" reputation type. A collection of elaborative results in Figure 7 show that the exponential dynamics of this reputation type works for all the parameters considered, and the master always reaches $p_{\mathcal{A}} = p_{\mathcal{A}}^{min}$ with eventual correctness. In addition, we see that the master also decreases its auditing cost, unlike the case of reputation Type 1 where $p_{\mathcal{A}}$ goes to values close to one in order for the master to receive the correct reply with a high probability.

When a majority (5 out of 9) of workers is covered, the system converges independently of the reputation type used. However, for reputation Type 1 we have found that larger tolerances are required to reach and maintain eventual correctness with smaller auditing costs, and that punishment makes the system to converge faster and with fewer audits (we found convergence problems for $\tau = 0.1$, see Figure 12 and 13 in Appendix C). With reputation Type 2, even for the lowest tolerance and no punishment the audit probability of the master eventually converges to $p_{\mathcal{A}}^{min}$ as opposed to reputation Type 1 at a small cost. In general, however, a tolerance of 50% reduces the total auditing cost independently of the initial cheating probability of the workers and the reputation type used (see Figure 14 and 15 in Appendix C). Finally, for reputation Type 3 we have found that smaller tolerances are required to reach and maintain eventual correctness, especially in the case that the initial $p_C = 1$; this is due to the distinct way we calculate reputation in Type 3 as we mentioned before (see Figures 16 and 17 in Appendix C).

Finally, for the sake of experimentation we checked that our mechanism reaches eventual correctness (with reputation Type 2) by covering only 1 out of the 5 rational workers when the other 4 are malicious. The performance of the system to reach eventual correctness is similar to the analogous case where all workers are covered. Reputation types Type 1 and Type 3 have the same problems as before, whereas the fact that reputation becomes constant with Type 2 allows rational covered workers to form a reputable set by itself and achieve fast eventual correctness (see Figure 18 in Appendix C).

Dynamic change of roles As a further check of the stability of our procedure, we now study the case when after correctness is reached some workers change their type, possibly due to a software or hardware error. We simulate a situation in which 5 out of 9 rational workers suddenly change their behavior to malicious at time 500, a worst-case scenario. Figure 6(b) shows that after the rational behavior of 5 workers turns to malicious, convergence is reached again after a few hundred rounds and eventual correctness resumes. It takes more time for reputation Type 2 to deal with the changes in the workers' behavior (see the more elaborative results in Figure 19 in Appendix C) because this reputation can never increase, and hence the system will reach eventual correctness only when the reputation of the workers that turned malicious becomes less than the reputation of the workers that stayed rational. It also takes more time for reputation Type 3 to deal with the changes in the worker' behavior (see Figure 19 in Appendix C). In the case of reputation Type 1 not only the reputation of the workers that turned malicious decreases but also the reputation of the workers that stayed rational increases. Therefore, reputation Type 1 exhibits better performance in dealing with dynamic changes of behavior than reputation types Type 2 and Type 3.

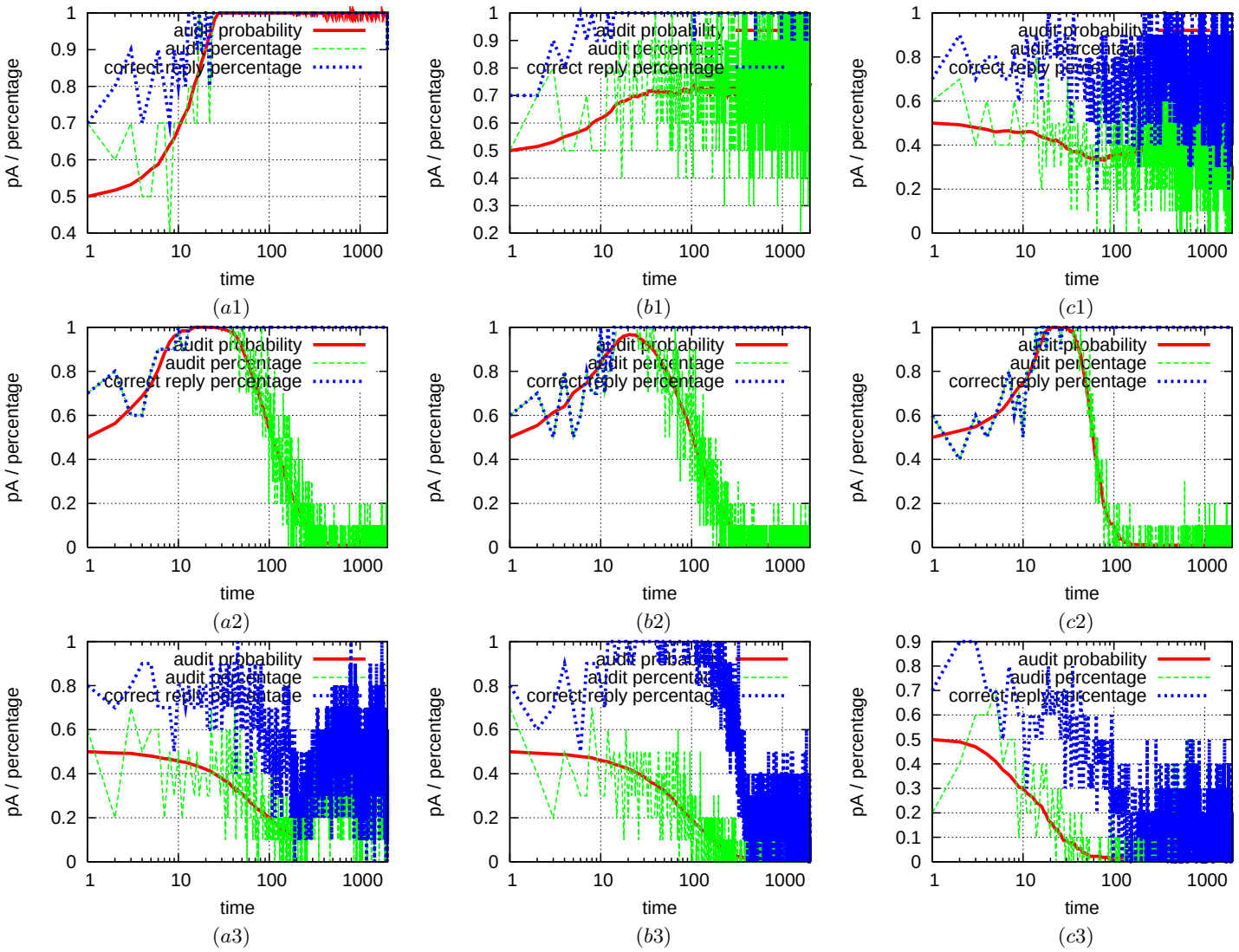


Fig. 7: One covered worker. Parameters are $WC_{\mathcal{T}} = 0.1$, $a_i = 0.1$, $\alpha = 0.1$, and $WB_{\mathcal{Y}} = 0.1$ (uncovered workers) / $WB_{\mathcal{Y}} = 1$ (covered workers). Master's auditing probability, audit percentage and correct reply percentage as a function of time. Top: reputation Type 1, initial $p_C = 0.5$. Middle: reputation Type 2, initial $p_C = 1$. Bottom: reputation Type 3, initial $p_C = 0.5$. First column $\tau = 0.1$, $WP_{\mathcal{C}} = 0$. Second column, $\tau = 0.1$, $WP_{\mathcal{C}} = 1$. Third column, $\tau = 0.5$, $WP_{\mathcal{C}} = 0$.

6 Conclusions and Future work

In this work we study a malicious-tolerant generic mechanism that uses reputation. We consider three reputation types, and give provable guarantees that only reputation Type 2 (introduced in this work) provides eventual correctness. Simulations have shown that in the case of having all rational workers covered, eventual correctness is achieved by all three types. In the case of covering only one altruistic or rational worker, simulations have shown that only reputation Type 2 can achieve eventual correctness. We show that reputation Type 2 has more potential in commercial platforms where high reliability together with low auditing cost, rewarding few workers and fast convergence are required. We believe this advances the development of reliable commercial Internet-based Master-Worker Computing services. From our simulations we make one more interesting observation: in the case when only rational workers exist and reputation Type 3 (BOINC-like) is used, although the system takes more time to converge, in every round auditing is lower. Thus, reputation Type 3 may fit better in volunteering setting where workers are most probably altruistic or rational and fast convergence can be sacrificed for lower auditing. In particular, our simulations reveal interesting tradeoffs between reputation types and parameters and show that our mechanism is a generic one that can be adjusted to various settings.

In a follow-up work we plan to investigate what happens if workers are connected to each other, forming a network (i.e, a social network through which they can communicate) or if malicious workers develop a more intelligent strategy against the system. Also the degree of trust among the players has to be considered and modeled in this scenario. Additionally, we have assumed throughout this work that workers are responsive and willing to perform the task. In a follow up work, we plan to explore the case where workers might not be responsive. Another extension we are planning to study is to assume a platform with multiple masters. The goal in this case is to match workers and masters to maximize social efficiency, constrained to masters' and workers' preferences. Finally, we plan to extend our mechanism to deal also with open class questions.

References

[1] Abraham, I., Dolev, D., Gonen, R., Halpern, J.: Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing. pp. 53–62. ACM (2006)

[2] Aiyer, A.S., Alvisi, L., Clement, A., Dahlin, M., Martin, J.P., Porth, C.: Bar fault tolerance for cooperative services. In: ACM SIGOPS Operating Systems Review. vol. 39, pp. 45–58. ACM (2005)

[3] Allen, B.: The Einstein@home project (2014), `http://einstein.phys.uwm.edu`

[4] Amazon.com: Amazon's Mechanical Turk (2014), `https://www.mturk.com`

[5] Anderson, D.P.: Boinc: A system for public-resource computing and storage. In: Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on. pp. 4–10. IEEE (2004)

[6] Anderson, D.P.: Volunteer computing: the ultimate cloud. ACM Crossroads 16(3), 7–10 (2010)

[7] Anderson, D.P.: BOINC reputation (2014), `http://boinc.berkeley.edu/trac/wiki/AdaptiveReplication`

[8] Anta, A.F., Georgiou, C., López, L., Santos, A.: Reliable internet-based master-worker computing in the presence of malicious workers. Parallel Processing Letters 22(01) (2012)

[9] Anta, A.F., Georgiou, C., Mosteiro, M.A.: Designing mechanisms for reliable internet-based computing. In: Network Computing and Applications, 2008. NCA'08. Seventh IEEE International Symposium on. pp. 315–324. IEEE (2008)

[10] Barber, S., Boyen, X., Shi, E., Uzun, E.: Bitter to betterhow to make bitcoin a better currency. In: Financial Cryptography and Data Security, pp. 399–414. Springer (2012)

[11] BitcoinMining.com: Bitcoin mining (2013), `http://www.bitcoinmining.com`

[12] boincstats.com: BOINC stats (2009), `http://boincstats.com/en/forum/10/4597`

[13] Bush, R.R., Mosteller, F.: Stochastic models for learning. (1955)

[14] Chien, S., Sinclair, A.: Convergence to approximate nash equilibria in congestion games. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. pp. 169–178. Society for Industrial and Applied Mathematics (2007)

[15] Christoforou, E., Anta, A.F., Georgiou, C., Mosteiro, M.A.: Algorithmic mechanisms for reliable master-worker internet-based computing. IEEE Trans. Computers 63(1), 179–195 (2014)

[16] Christoforou, E., Anta, A.F., Georgiou, C., Mosteiro, M.A., Sánchez, A.: Applying the dynamics of evolution to achieve reliability in master–worker computing. Concurrency and Computation: Practice and Experience 25(17), 2363–2380 (2013)

[17] Eickhoff, C., de Vries, A.P.: Increasing cheat robustness of crowdsourcing tasks. Information retrieval 16(2), 121–137 (2013)

[18] Estrada, T., Taufer, M., Anderson, D.P.: Performance prediction and analysis of boinc projects: An empirical study with emboinc. Journal of Grid Computing 7(4), 537–554 (2009)
[19] Golle, P., Mironov, I.: Uncheatable distributed computations. In: Topics in CryptologyCT-RSA 2001, pp. 425–440. Springer (2001)
[20] Heien, E.M., Anderson, D.P., Hagihara, K.: Computing low latency batches with unreliable workers in volunteer computing environments. Journal of Grid Computing 7(4), 501–518 (2009)
[21] Howe, J.: The rise of crowdsourcing. Wired magazine 14(6), 1–4 (2006)
[22] Hyman, P.: Software aims to ensure fairness in crowdsourcing projects. Commun. ACM 56(8), 19–21 (2013)
[23] Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision support systems 43(2), 618–644 (2007)
[24] Kittur, A., Nickerson, J.V., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., Lease, M., Horton, J.: The future of crowd work. In: Proceedings of the 2013 conference on Computer supported cooperative work. pp. 1301–1318. ACM (2013)
[25] Kondo, D., Araujo, F., Malecot, P., Domingues, P., Silva, L.M., Fedak, G., Cappello, F.: Characterizing result errors in internet desktop grids. In: Euro-Par 2007 Parallel Processing, pp. 361–371. Springer (2007)
[26] Konwar, K.M., Rajasekaran, S., Shvartsman, A.A.: Robust network supercomputing with malicious processes. In: Distributed Computing, pp. 474–488. Springer (2006)
[27] Korpela, E., Werthimer, D., Anderson, D.P., Cobb, J., Lebofsky, M.: Seti@ homemassively distributed computing for seti. Computing in science & engineering 3(1), 78–83 (2001)
[28] Li, H.C., Clement, A., Marchetti, M., Kapritsos, M., Robison, L., Alvisi, L., Dahlin, M.: Flightpath: Obedience vs. choice in cooperative services. In: OSDI. vol. 8, pp. 355–368 (2008)
[29] Li, H.C., Clement, A., Wong, E.L., Napper, J., Roy, I., Alvisi, L., Dahlin, M.: Bar gossip. In: Proceedings of the 7th symposium on Operating systems design and implementation. pp. 191–204. USENIX Association (2006)
[30] Microworkers.com: microWorkers, work & offer a micro job (2014), `https://microworkers.com/`
[31] Sarmenta, L.F.: Sabotage-tolerance mechanisms for volunteer computing systems. Future Generation Computer Systems 18(4), 561–572 (2002)
[32] SETI@home: SETI@home Poll Results (2000), `http://boinc.berkeley.edu/slides/xerox/polls.html`
[33] Shneidman, J., Parkes, D.C.: Rationality and self-interest in peer to peer networks. In: Peer-to-Peer Systems II, pp. 139–148. Springer (2003)
[34] Silberman, M.S., Irani, L., Ross, J.: Ethics and tactics of professional crowdwork. XRDS: Crossroads, The ACM Magazine for Students 17(2), 39–43 (2010)
[35] Smith, J.M.: Evolution and the Theory of Games. Cambridge university press (1982)
[36] Sonnek, J., Chandra, A., Weissman, J.B.: Adaptive reputation-based scheduling on unreliable distributed infrastructures. Parallel and Distributed Systems, IEEE Transactions on 18(11), 1551–1564 (2007)
[37] Szepesvári, C.: Algorithms for reinforcement learning. Synthesis Lectures on Artificial Intelligence and Machine Learning 4(1), 1–103 (2010)
[38] Taufer, M., Anderson, D.P., Cicotti, P., Brooks III, C.L.: Homogeneous redundancy: a technique to ensure integrity of molecular simulation results using public computing. In: Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International. pp. 119a–119a. IEEE (2005)
[39] Vilaça, X., Denysyuk, O., Rodrigues, L.: Asynchrony and collusion in the n-party bar transfer problem. In: Structural Information and Communication Complexity, pp. 183–194. Springer (2012)
[40] Von Ahn, L.: Games with a purpose. Computer 39(6), 92–94 (2006)
[41] Yuen, M.C., King, I., Leung, K.S.: A survey of crowdsourcing systems. In: Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom). pp. 766–773. IEEE (2011)
[42] Yurkewych, M., Levine, B.N., Rosenberg, A.L.: On the cost-ineffectiveness of redundancy in commercial p2p computing. In: Proceedings of the 12th ACM conference on Computer and communications security. pp. 280–288. ACM (2005)
[43] Zhang, Y., van der Schaar, M.: Reputation-based incentive protocols in crowdsourcing applications. In: INFOCOM, 2012 Proceedings IEEE. pp. 2140–2148. IEEE (2012)

APPENDIX

A Reputation Types and Natural Properties

In this section we show that all three reputation types (Type 1, Type 2 and Type 3) satisfy the following two natural properties:

A. If a worker is honest when the master audits, the reputation of the worker cannot decrease.
B. if a worker cheats when the master audits, the reputation of the worker cannot increase.

Lemma 4. *Natural Property A holds for reputation Type 1.*

Proof. Assume that at state s_r worker i has reputation $\rho_i(r) = \frac{v_i(r)+1}{aud(r)+2}$ and in the next state the master audits and the worker is honest then reputation becomes $\rho_i(r+1) = \frac{v_i(r)+2}{aud(r)+3}$. Since $\frac{v_i(r)+1}{aud(r)+2} \leq \frac{v_i(r)+2}{aud(r)+3}$ the lemma holds.

Lemma 5. *Natural Property B holds for reputation Type 1.*

Proof. Assume that at state s_r worker i has reputation $\rho_i(r) = \frac{v_i(r)+1}{aud(r)+2}$ and in the next state the master audits and the worker cheats then reputation becomes $\rho_i(r+1) = \frac{v_i(r)+1}{aud(r)+3}$. Since $\frac{v_i(r)+1}{aud(r)+2} \geq \frac{v_i(r)+1}{aud(r)+3}$ the lemma holds.

Lemma 6. *Natural Property A holds for reputation Type 2.*

Proof. Assume that at state s_r worker i has reputation $\rho_i(r) = \varepsilon^{aud(r)-v_i(r)}$ and in the next state the master audits and the worker is honest then reputation becomes $\rho_i(r+1) = \varepsilon^{aud(r)-v_i(r)}$, then the lemma trivially holds.

Lemma 7. *Natural Property B holds for reputation Type 2.*

Proof. Assume that at state s_r worker i has reputation $\rho_i(r) = \varepsilon^{aud(r)-v_i(r)}$ and in the next state the master audits and the worker cheats then reputation becomes $\rho_i(r+1) = \varepsilon^{aud(r)+1-v_i(r)}$. Since $\varepsilon \in (0,1)$ then $\varepsilon^{aud(r)-v_i(r)} \geq \varepsilon^{aud(r)+1-v_i(r)}$ and the lemma holds.

Lemma 8. *Natural Property A holds for reputation Type 3.*

Proof. At state s_r worker i can have reputation $\rho_i(r) \geq 0$ and in the next state the master audits and the worker is honest then $\beta_i(r) < 0.05$ and $\rho_i(r+1) \geq 0$. If in state s_r $\rho_i(r) = 0$ then the natural property holds. If in state s_r, $\rho_i(r) = 1 - \sqrt{\frac{\beta_i(r)}{0.05}}$ then in the next state $\rho_i(r+1) = 1 - \sqrt{\frac{\beta_i(r+1)}{0.05}}$. The property still holds since $\beta_i(r) < \beta_i(r+1)$ and thus the claim is proved.

Lemma 9. *Natural Property B holds for reputation Type 3.*

Proof. At state s_r worker i can have reputation $\rho_i(r) \geq 0$ and in the next state the master audits and the worker cheats then $\beta_i(r+1) > 0.05$ and $\rho_i(r+1) = 0$ and the claim holds.

B Reputation Types and Properties 1 and 2

In this section we show that Property 1 holds for reputation types Type 1 and Type 2 and under certain conditions for Type 3. Furthermore, we show that Property 2 holds only for reputation Type 2.

Lemma 10. *Property 1 holds for reputation Type 1.*

Proof. Consider any $d > 0$, any $X \subseteq W$ non empty. Without loss of generality assume $|X| = k$. Consider rounds $r+1, \dots r+j$, for some j, such that the master audits, workers in X are honest and workers not in X cheat. For $\forall i \in X$, $\rho_i(r+j) = \frac{v_i(r)+j+1}{aud(r)+j+2}$; and $\forall i \notin X$, $\rho_i(r+j) = \frac{v_i(r)+1}{aud(r)+j+2}$. Then $\rho_X(r+j) = \sum_{i\in X} \rho_i(r+j) = \frac{\sum_{i\in X} v_i(r)}{aud(r)+j+2} + \frac{k(j+1)}{aud(r)+j+2} \geq \frac{j+1}{aud(r)+j+2}$; and $p_{W\setminus X}(r+j) = \sum_{i\in W\setminus X} \rho_i(r+j) = \frac{(n+k)}{aud(r)+j+2} \leq \frac{n+1}{aud(r)+j+2}$. For any $j \leq \delta(n-1)$ we have, $\rho_X(r+j) \geq \frac{j+1}{aud(r)+j+2} > \frac{\delta(n-1)}{aud(r)+j+2} \geq \rho_{W\setminus X}(r+j)$. Hence, setting $\gamma(\delta) = \delta(n-1)$ proves the claim.

Lemma 11. *Property 2 does not hold for reputation Type 1.*

Proof. Consider any round where $aud(r+1) = aud(r)+1$ and $\forall j \in X \cup Y$ $v_j(r+1) = v_j(r)+1$. Without lose of generality assume that in state s_r, $|X| = k_r$. Then we have that if,

$$\rho_X(r) > \rho_Y(r)$$

$$\sum_{i\in X} \frac{v_i(r)+1}{aud(r)+2} > \sum_{i\in Y} \frac{v_i(r)+1}{aud(r)+2}$$

$$\frac{\sum_{i\in X} v_i(r)}{aud(r)+2} + \frac{k_r}{aud(r)+2} > \frac{\sum_{i\in Y} v_i(r)}{aud(r)+2} + \frac{n-k_r}{aud(r)+2}$$

$$\sum_{i\in X} v_i(r) + k_r > \sum_{i\in Y} v_i(r) + n - k_r$$

then,

$$\rho_X(r+1) > \rho_Y(r+1)$$

$$\sum_{i\in X} \frac{v_i(r)+2}{aud(r)+3} > \sum_{i\in Y} \frac{v_i(r)+2}{aud(r)+3}$$

$$\frac{\sum_{i\in X} v_i(r)}{aud(r)+3} + \frac{2k_{r+1}}{aud(r)+3} > \frac{\sum_{i\in Y} v_i(r)}{aud(r)+3} + \frac{2(n-k_{r+1})}{aud(r)+3}$$

$$\sum_{i\in X} v_i(r) + 2k_{r+1} > \sum_{i\in Y} v_i(r) + 2(n-k_{r+1})$$

Thus if $k_r \neq k_{r+1}$ then the entailment may not hold.

Lemma 12. *Property 1 holds for reputation Type 2.*

Proof. Consider any $\delta > 0$, any $X \subseteq W$ non empty. Without loss of generality assume $|X| = k$. Consider rounds $r+1, \dots r+j$, for some j, such that the master audits, workers in X are honest and workers not in X cheat. For $\forall i \in X$, $\rho_i(r+j) = \varepsilon^{aud(r)-v_i(r)}$ and $\forall i \notin X$, $\rho_i(r+j) = \varepsilon^{aud(r)+j-v_i(r)}$.Then $\rho_X(r+j) = \sum_{i\in X} \rho_i(r+j) = \sum_{i\in X} \varepsilon^{aud(r)-v_i(r)} \geq \varepsilon^{aud(r)}$ and $\rho_{W\setminus X}(r+j) = \sum_{i\in W\setminus X} \rho_i(r+j) = \sum_{i\in W\setminus X} \varepsilon^{aud(r)+j-v_i(r)} \leq (n-1)\varepsilon^{aud(r)+j}$. For any $j < -log(\delta(n-1))$ we have, $\rho_X(r+j) \geq \varepsilon^{aud(r)} > \delta(n-1)\varepsilon^{aud(r)+j} \geq \rho_{W\setminus X}(r+j)$. Hence, setting $\gamma(\delta) < -log(\delta(n-1))$ proves the claim.

Lemma 13. *Property 2 holds for reputation Type 2.*

Proof. Consider any $X \subset W$ and $Y \subset W$. Then if $\rho_X(r) = \sum_{i \in X} \varepsilon^{aud(r)-v_i(r)}$ then $\rho_X(r+1) = \sum_{i \in X} \varepsilon^{aud(r)+1-v_i(r)-1} = \sum_{i \in X} \varepsilon^{aud(r)-v_i(r)}$. If $\rho_Y(r) = \sum_{i \in Y} \varepsilon^{aud(r)-v_i(r)}$ then $\rho_Y(r+1) = \sum_{i \in Y} \varepsilon^{aud(r)+1-v_i(r)-1} = \sum_{i \in Y} \varepsilon^{aud(r)-v_i(r)}$. Thus trivially the condition $\rho_X(r) > \rho_Y(r) \Rightarrow \rho_X(r+1) > \rho_Y(r+1)$ holds.

Lemma 14. *Property 1 does not hold for reputation Type 3.*

Proof. Consider any $\delta > 0$, and $X \subseteq W$ non empty. Without loss of generality assume $|X| = k$. Consider rounds $r+1 \dots r+j$ for some j, such that master audits, workers in X are honest and workers not in X cheat. For $\forall i \in X$ if $\beta_i(r+j) > 0.05$ then $\rho_i(r+j) = 0$ else $\rho_i(r+j) = 1 - \sqrt{\frac{\beta_i(r) \times j \times 0.95}{0.05}}$. For $\forall i \in W \setminus X$ then $\beta_i(r+j) > 0.05$ thus $\rho_i(r+j) = 0$. If $\gamma(\delta) > 0$ then $\forall i \in W \setminus X$ $\rho_i(r+j) = 0$.To have $\rho_X(r+\gamma(\delta)) > \delta \rho_{W \setminus X}(r+\gamma(\delta))$ we need to know the state s_r where the master starts to audit to know in how many rounds $\exists i \in X$ where $\beta_i(r) \leq 0.05$. Thus the condition of Property 1 for Type 3 depends on the current state s_r where the master begins to audit. Hence Property 1 for reputation Type 3 does not hold.

Lemma 15. *Property 1 holds for reputation Type 3, given that an upper bound b of the value $\beta_i(r)$ is established.*

Proof. Consider any $\delta > 0$, and $X \subseteq W$ non empty. Without loss of generality assume $|X| = k$. Consider rounds $r+1 \dots r+j$ for some j, such that master audits, workers in X are honest and workers not in X cheat. For $\forall i \in X$ if $\beta_i(r+j) > 0.05$ then $\rho_i(r+j) = 0$ else $\rho_i(r+j) = 1 - \sqrt{\frac{\beta_i(r) \times j \times 0.95}{0.05}}$. For $\forall i \in W \setminus X$ then $\beta_i(r+j) > 0.05$ thus $\rho_i(r+j) = 0$. If $\gamma(\delta) > 0$ then $\forall i \in W \setminus X$ $\rho_i(r+j) = 0$. If a constant upper bound b in the value of $\beta_i(r)$ is established in the s_r state then in $j < \frac{0.05}{b \times 0.95}$ rounds $\forall i \in X$, $\rho_i(r+j) > 0$ and thus the condition $\rho_X(r+\gamma(\delta)) > \delta \rho_{W \setminus X}(r+\gamma(\delta))$ becomes true by setting $\gamma(\delta) < \frac{0.05}{b \times 0.95}$ and the claim is proved.

Lemma 16. *Property 2 does not hold for reputation Type 3.*

Proof. Consider any $X \subset W$ and $Y \subset W$. If $\rho_X(r) > \rho_Y(r)$ then in the next state s_{r+1} the following apply. For $\forall j \in X \setminus Y$, $v_j(r+1) = v_j(r) + 1$ (they are honest). Thus if $\beta_j(r+1) > 0.05$ then $\rho_j(r+1) = 0$ else $\rho_j(r+1) = 1 - \sqrt{\frac{\beta_j(r) \times 0.95}{0.05}}$. Consider the following case where $\forall j \in Y$, $\rho_j(r) = 0$ and $\forall j \in X'$, where X' is the set of all workers in X besides one, lets name it z, $\rho_j(r) = 0$ and $\rho_z(r) = 1 - \sqrt{\frac{\beta_z(r) \times 0.95}{0.05}}$. Then $\rho_X(r) > \rho_Y(r)$ holds. In the next round though there is a possibility that $\forall j \in Y$, $\rho_j(r+1) = 1 - \sqrt{\frac{\beta_j(r+1) \times 0.95}{0.05}}$, while only the reputation of z changes in the next state for the X set. Thus if $1 - \sqrt{\frac{\beta_z(r) \times 0.95}{0.05}} < |Y| \times (1 - \sqrt{\frac{\beta_j(r+1) \times 0.95}{0.05}})$ then Property 2 does not hold for reputation Type 3 and the claim is proved.

C Supplementary Simulation Figures

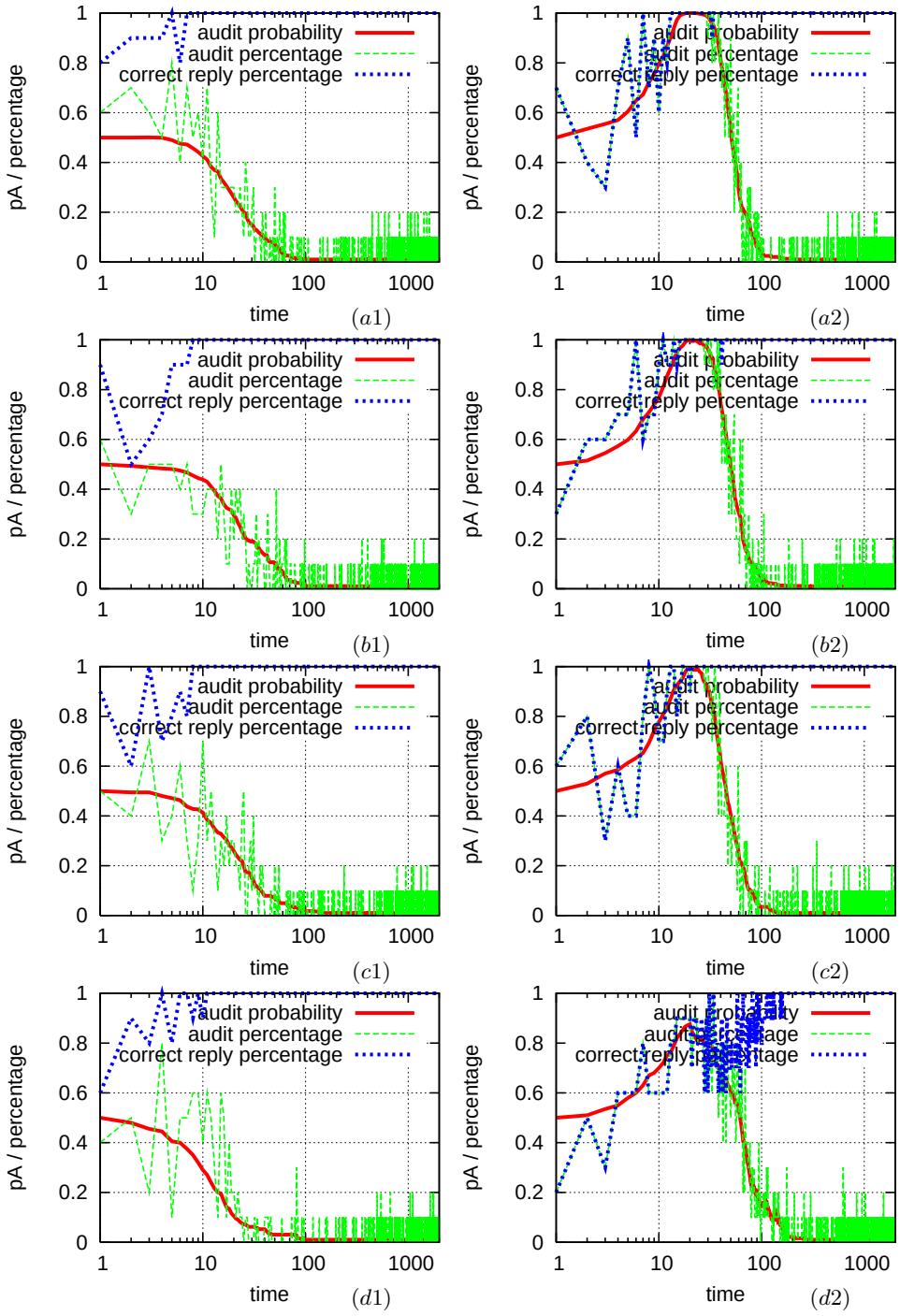


Fig. 8: Rational workers. Master's auditing probability as a function of time, audit percentage as a function of time and correct answer percentage as a function of time with initial $p_C = 0.5$ (left) or $p_C = 1$ (right). Parameters in all panels, $p_{\mathcal{A}} = 0.5$, $WC_{\mathcal{T}} = 0.1$, $WP_{\mathcal{C}} = 0$ and $\alpha = 0.1$, $a_i = 0.1$. First row, master does not use reputation. Second row, master uses reputation Type 1. Third row, master uses reputation Type 2. Fourth row, master uses reputation Type 3.

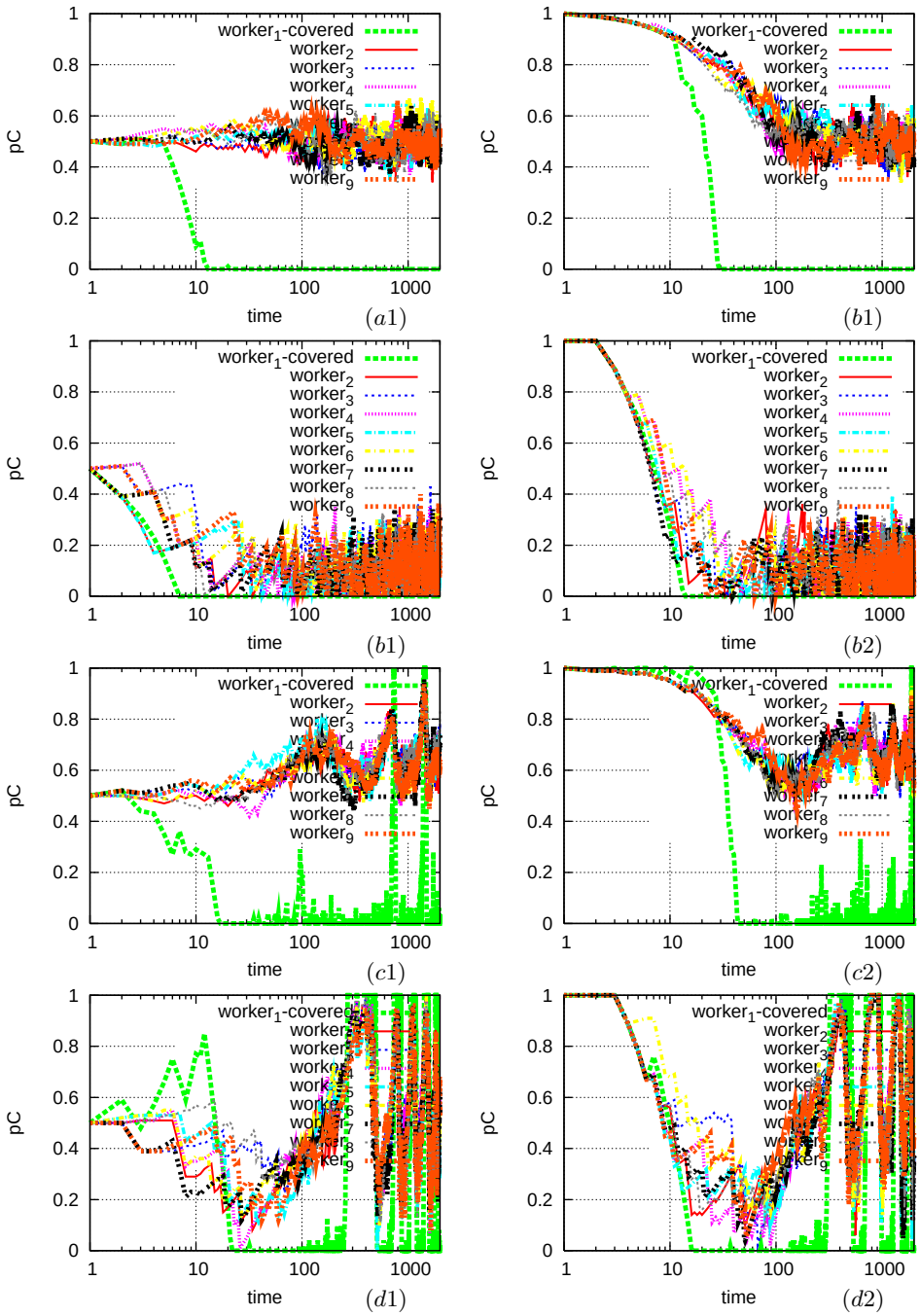


Fig. 9: One covered worker and reputation Type 1 used. Parameters in all plots $WC_{\mathcal{T}} = 0.1$, $a_i = 0.1$, $\alpha = 0.1$, and $WB_{\mathcal{Y}} = 0.1$ (for the uncovered workers) / $WB_{\mathcal{Y}} = 1$ (for the covered workers). Plots depict the cheating probability for each worker as a function of time, for an individual realization. Left, initial $p_C = 0.5$. Right, initial $p_C = 1$. First row $\tau = 0.1$, $WP_{\mathcal{C}} = 0$. Second row $\tau = 0.1$, $WP_{\mathcal{C}} = 1$. Third row $\tau = 0.5$, $WP_{\mathcal{C}} = 0$. Fourth row $\tau = 0.5$, $WP_{\mathcal{C}} = 1$.

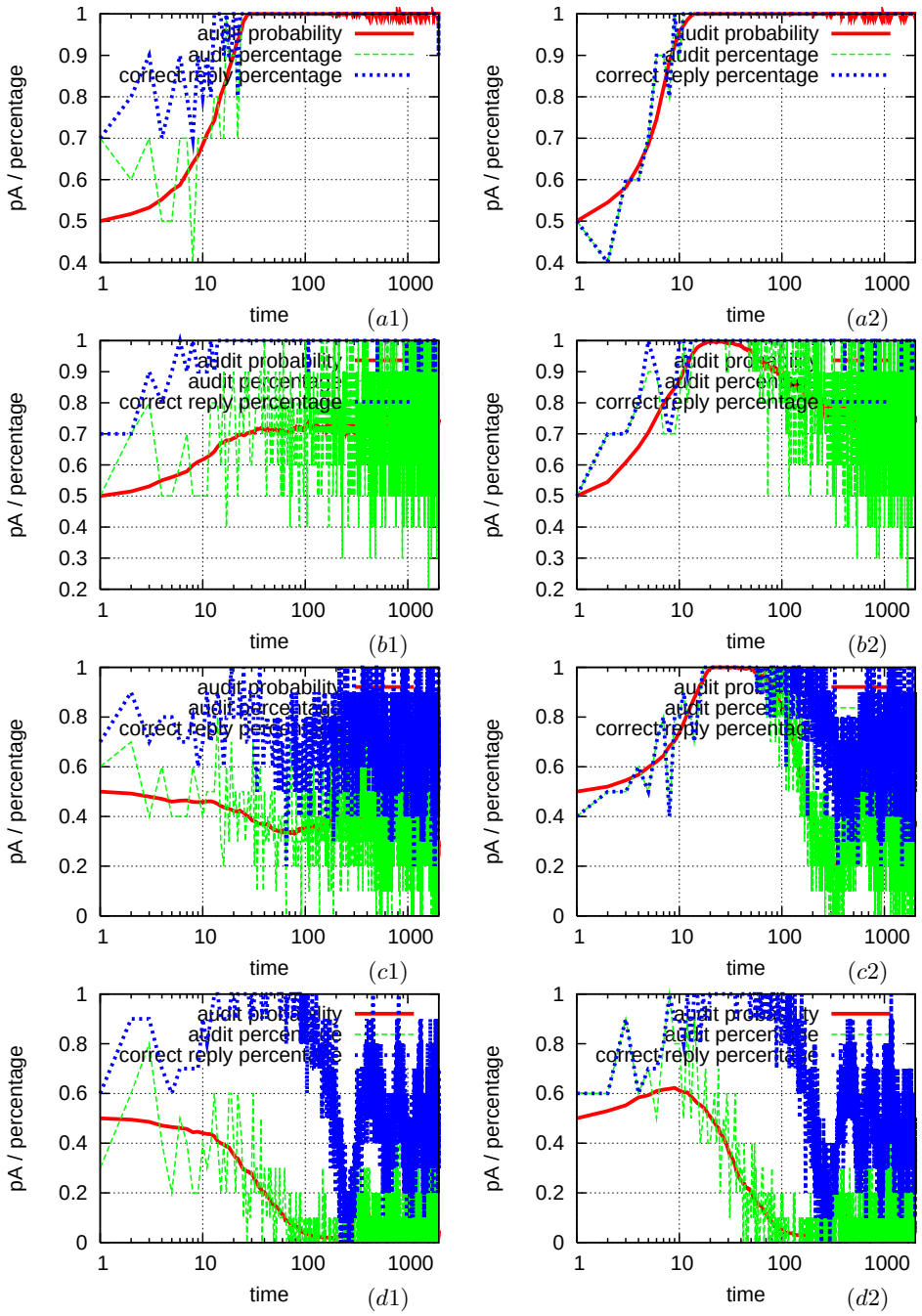


Fig. 10: One covered worker and reputation Type 1 used. Parameters in all plots $WC_{\mathcal{T}} = 0.1$, $a_i = 0.1$, $\alpha = 0.1$, and $WB_{\mathcal{Y}} = 0.1$ (for the uncovered workers) / $WB_{\mathcal{Y}} = 1$ (for the covered workers). Plots depict the master's auditing probability as a function of time, audit percentage as a function of time and correct reply percentage as a function of time. Left, initial $p_C = 0.5$. Right, initial $p_C = 1$. First row $\tau = 0.1$, $WP_{\mathcal{C}} = 0$. Second row $\tau = 0.1$, $WP_{\mathcal{C}} = 1$. Third row $\tau = 0.5$, $WP_{\mathcal{C}} = 0$. Fourth row $\tau = 0.5$, $WP_{\mathcal{C}} = 1$.

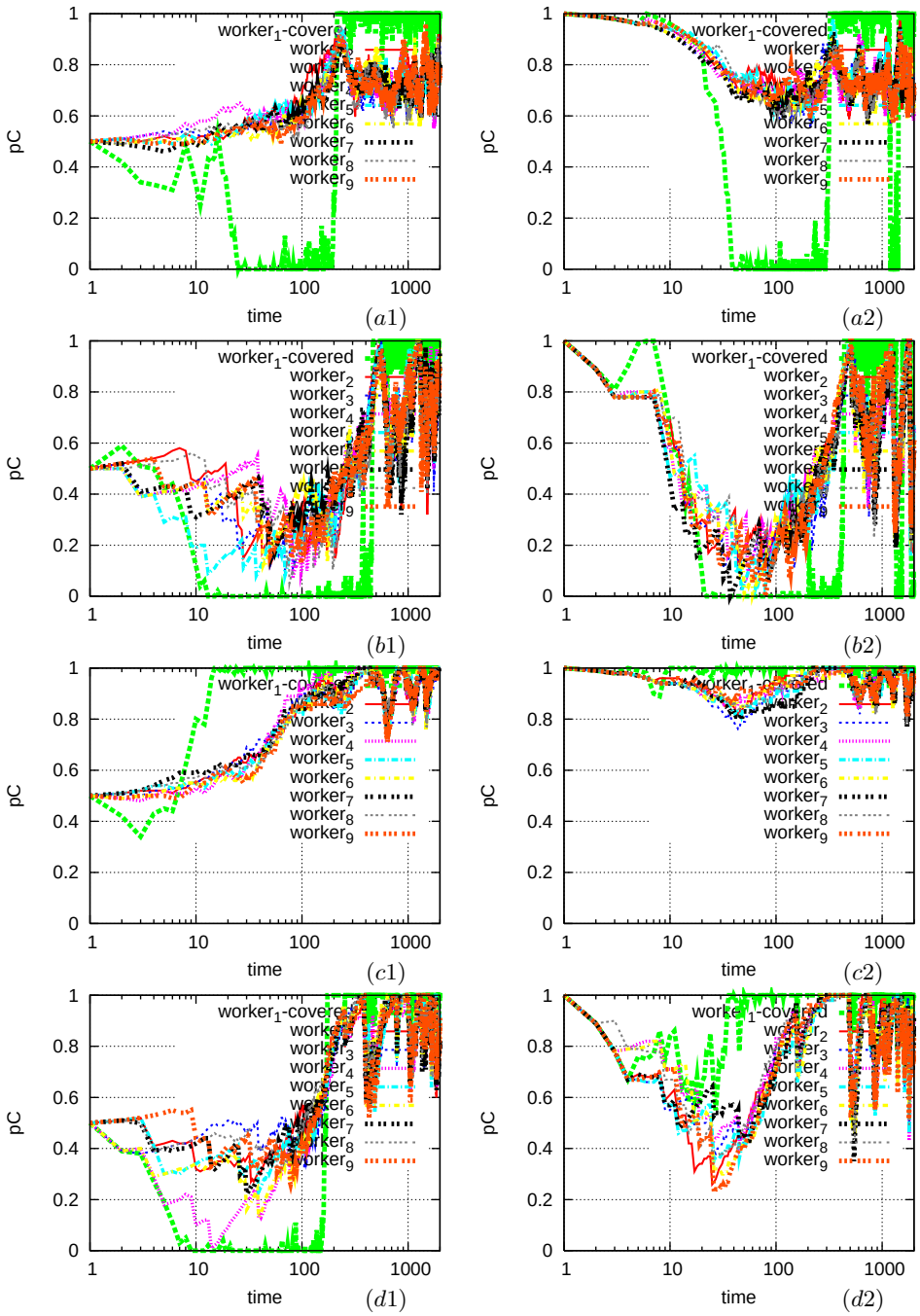


Fig. 11: One covered worker and reputation Type 3 used. Parameters in all plots $WC_{\mathcal{T}} = 0.1$, $a_i = 0.1$, $\alpha = 0.1$, and $WB_{\mathcal{Y}} = 0.1$ (for the uncovered workers) / $WB_{\mathcal{Y}} = 1$ (for the covered workers). Plots depict the cheating probability for each worker as a function of time, for an individual realization. Left, initial $p_C = 0.5$. Right, initial $p_C = 1$. First row $\tau = 0.1$, $WP_{\mathcal{C}} = 0$. Second row $\tau = 0.1$, $WP_{\mathcal{C}} = 1$. Third row $\tau = 0.5$, $WP_{\mathcal{C}} = 0$. Fourth row $\tau = 0.5$, $WP_{\mathcal{C}} = 1$.

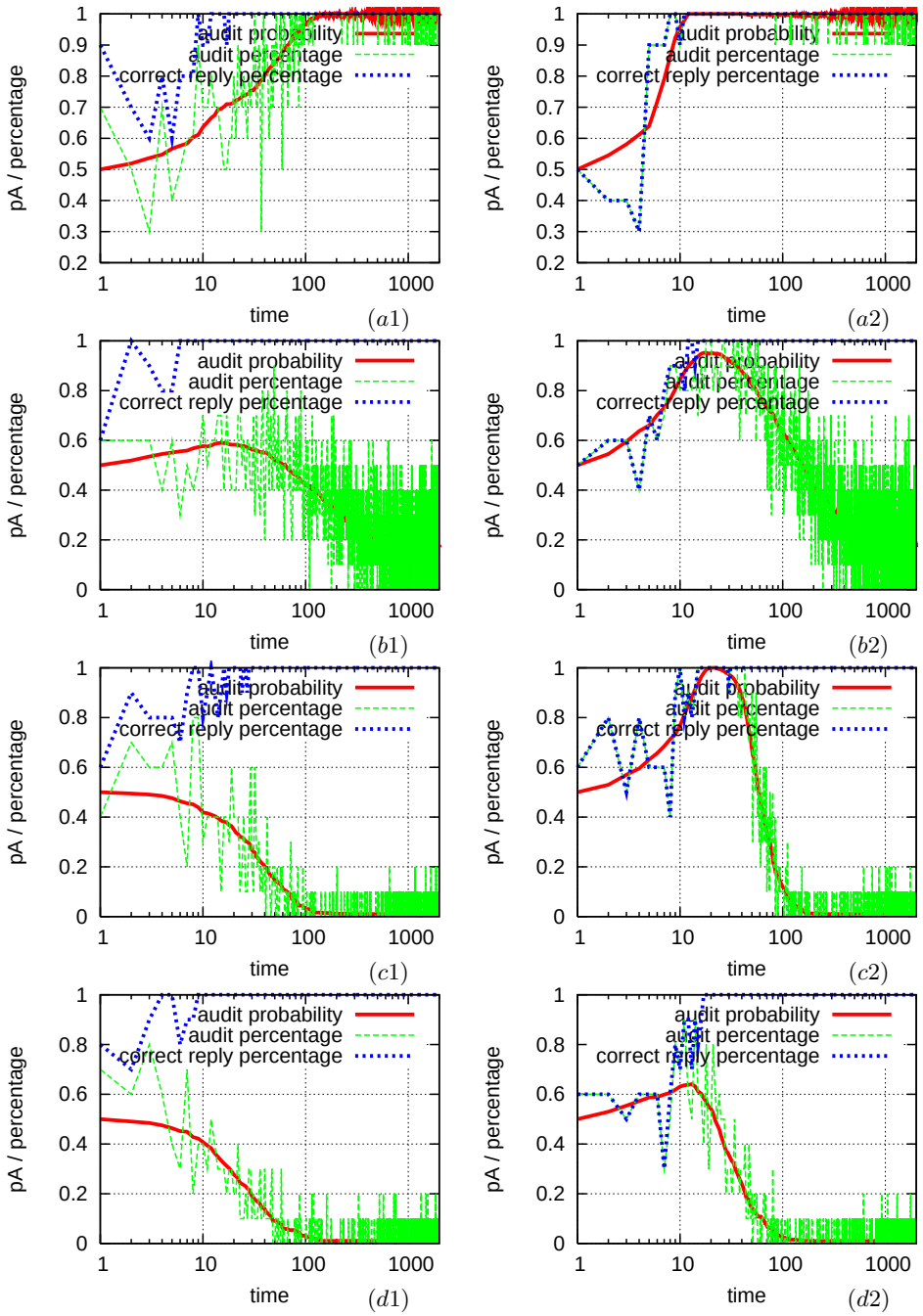


Fig. 12: Five covered workers and reputation Type 1 used. Parameters in all plots $WC_{\mathcal{T}} = 0.1$, $a_i = 0.1$, $\alpha = 0.1$, and $WB_{\mathcal{Y}} = 0.1$ (for the uncovered workers) / $WB_{\mathcal{Y}} = 1$ (for the covered workers). Plots depict the master's auditing probability as a function of time, audit percentage as a function of time and correct reply percentage as a function of time. Left, initial $p_C = 0.5$. Right, initial $p_C = 1$. First row $\tau = 0.1$, $WP_{\mathcal{C}} = 0$. Second row $\tau = 0.1$, $WP_{\mathcal{C}} = 1$. Third row $\tau = 0.5$, $WP_{\mathcal{C}} = 0$. Fourth row $\tau = 0.5$, $WP_{\mathcal{C}} = 1$.

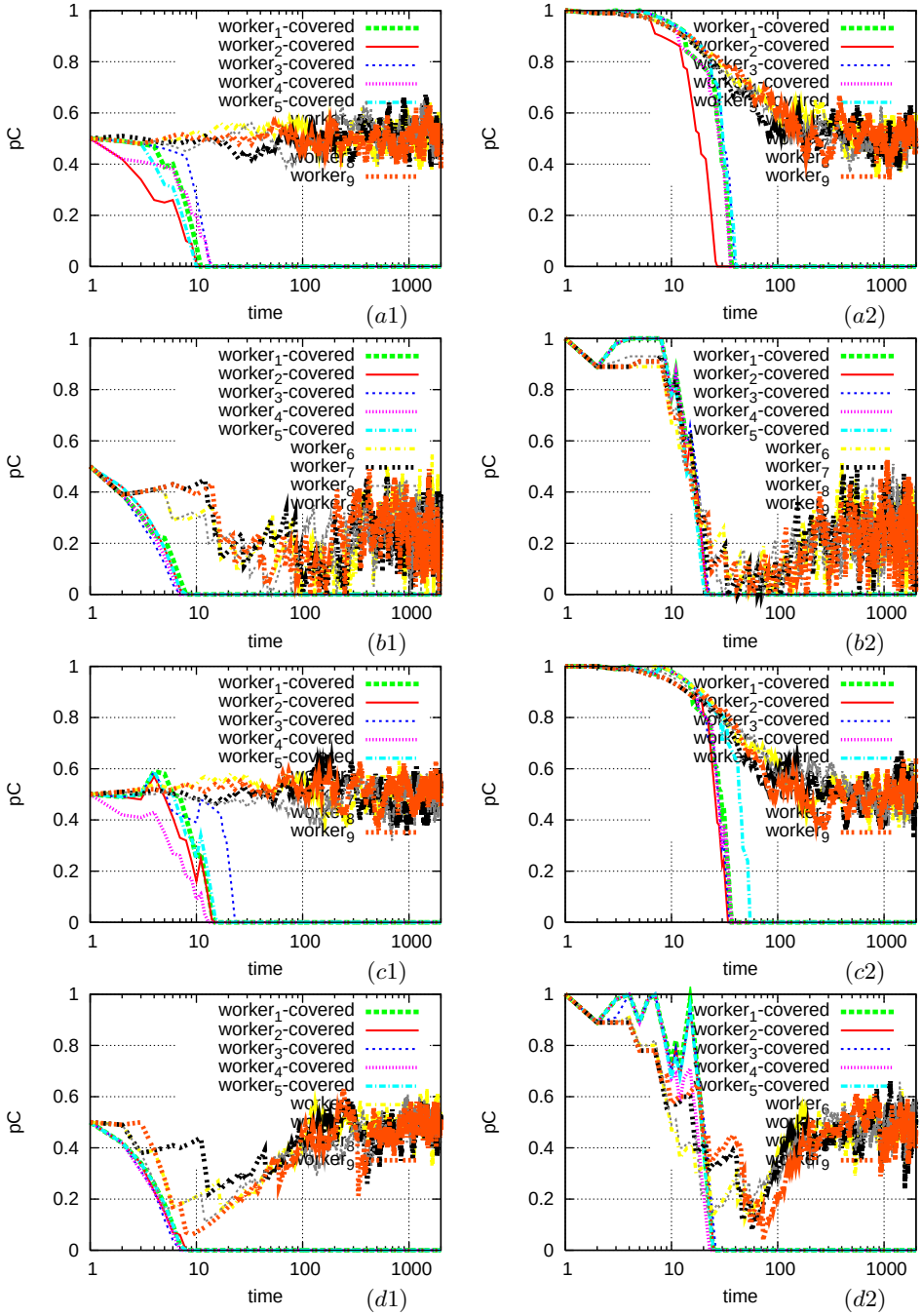


Fig. 13: Five covered workers and reputation Type 1 used. Parameters in all plots $WC_{\mathcal{T}} = 0.1$, $a_i = 0.1$, $\alpha = 0.1$, and $WB_{\mathcal{Y}} = 0.1$ (for the uncovered workers) / $WB_{\mathcal{Y}} = 1$ (for the covered workers). Plots depict the cheating probability for each worker as a function of time, for an individual realization. Left, initial $p_C = 0.5$. Right, initial $p_C = 1$. First row $\tau = 0.1$, $WP_{\mathcal{C}} = 0$. Second row $\tau = 0.1$, $WP_{\mathcal{C}} = 1$. Third row $\tau = 0.5$, $WP_{\mathcal{C}} = 0$. Fourth row $\tau = 0.5$, $WP_{\mathcal{C}} = 1$.

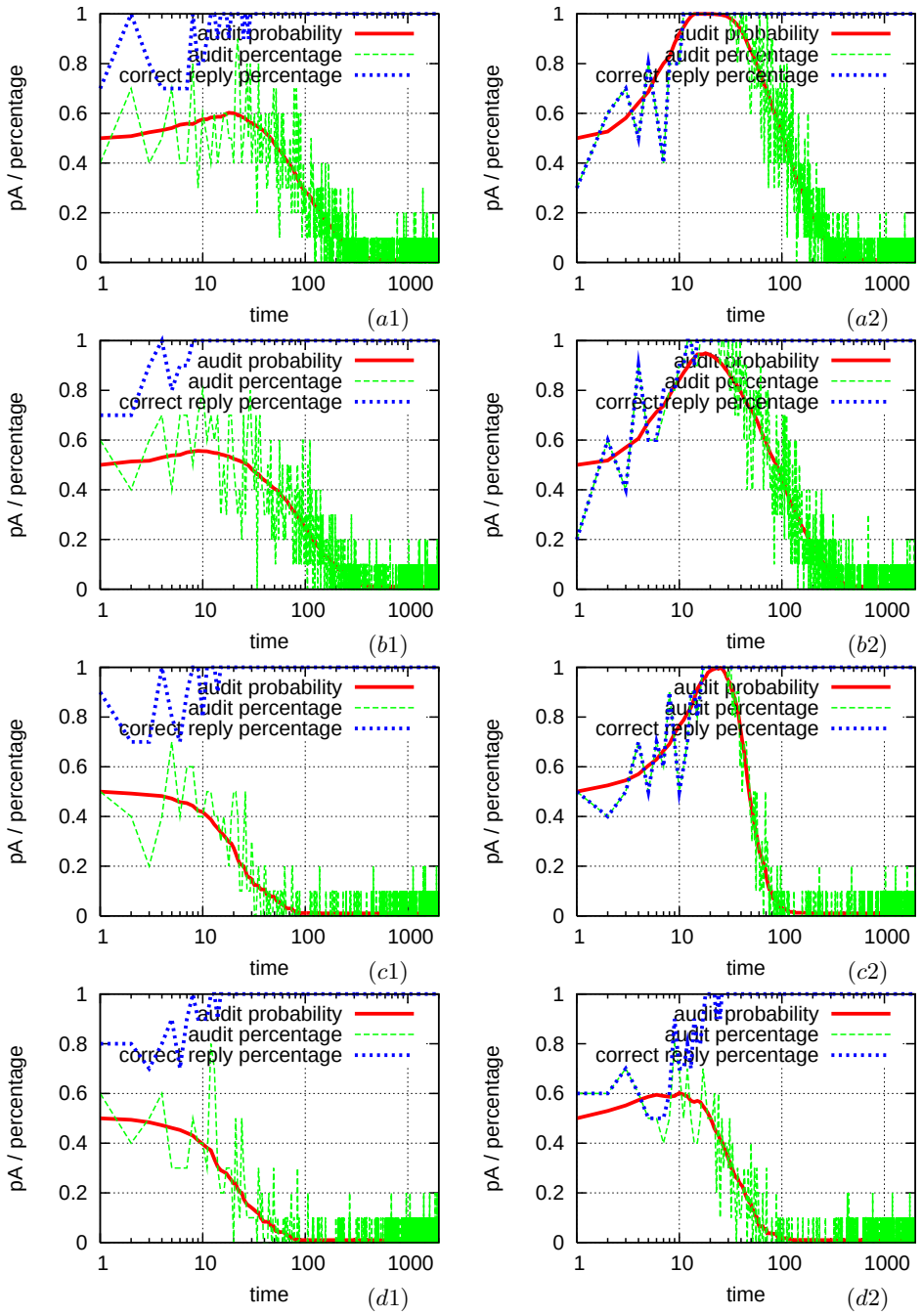


Fig. 14: Five covered workers and reputation Type 2 used. Parameters in all plots $WC_{\mathcal{T}} = 0.1$, $a_i = 0.1$, $\alpha = 0.1$, and $WB_{\mathcal{Y}} = 0.1$ (for the uncovered workers) / $WB_{\mathcal{Y}} = 1$ (for the covered workers). Plots depict the master's auditing probability as a function of time, audit percentage as a function of time and correct reply percentage as a function of time. Left, initial $p_C = 0.5$. Right, initial $p_C = 1$. First row $\tau = 0.1$, $WP_{\mathcal{C}} = 0$. Second row $\tau = 0.1$, $WP_{\mathcal{C}} = 1$. Third row $\tau = 0.5$, $WP_{\mathcal{C}} = 0$. Fourth row $\tau = 0.5$, $WP_{\mathcal{C}} = 1$.

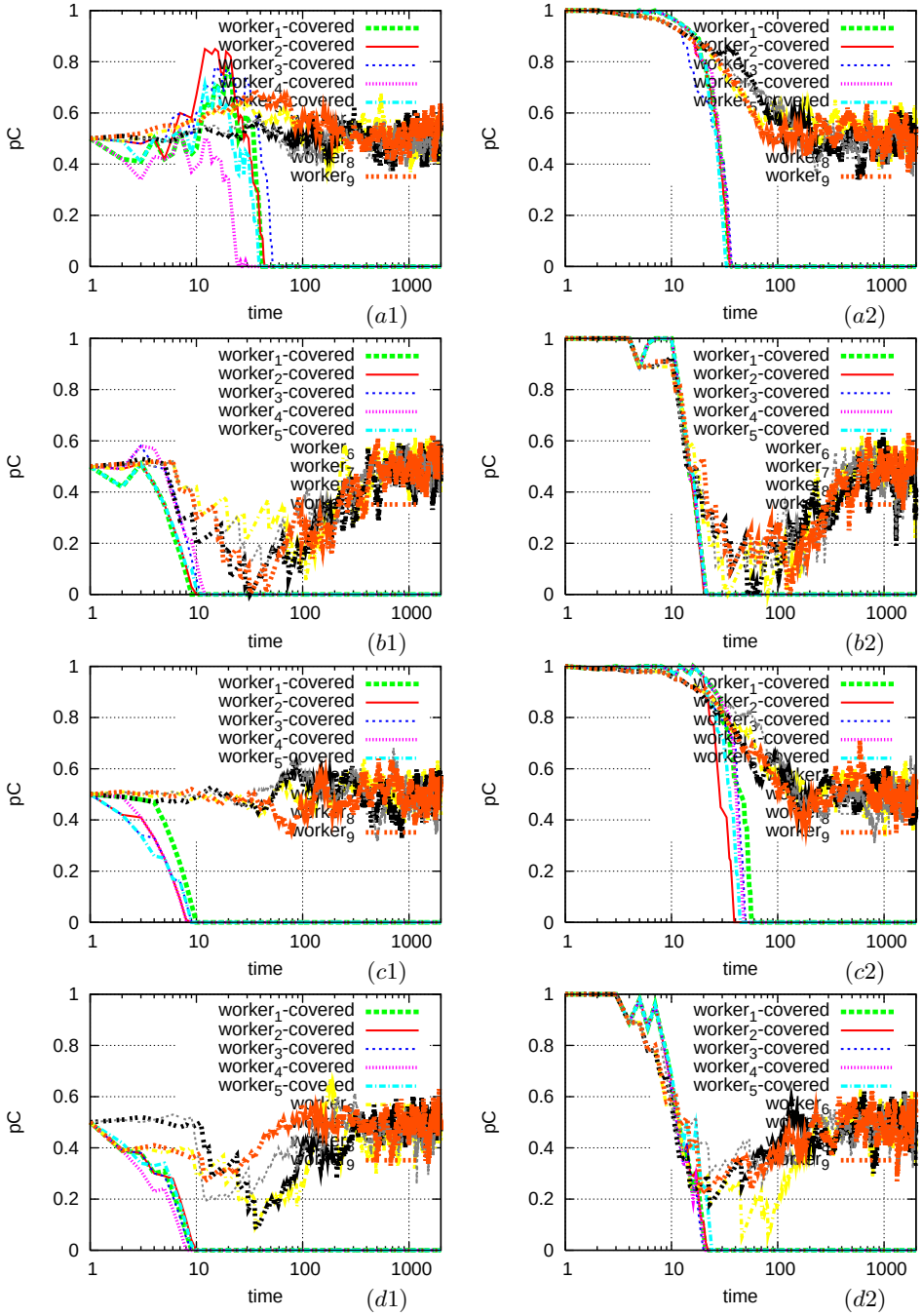


Fig. 15: Five covered workers and reputation Type 2 used. Parameters in all plots $WC_{\mathcal{T}} = 0.1$, $a_i = 0.1$, $\alpha = 0.1$, and $WB_{\mathcal{Y}} = 0.1$ (for the uncovered workers) / $WB_{\mathcal{Y}} = 1$ (for the covered workers). Plots depict the cheating probability for each worker as a function of time, for an individual realization. Left, initial $p_C = 0.5$. Right, initial $p_C = 1$. First row $\tau = 0.1$, $WP_{\mathcal{C}} = 0$. Second row $\tau = 0.1$, $WP_{\mathcal{C}} = 1$. Third row $\tau = 0.5$, $WP_{\mathcal{C}} = 0$. Fourth row $\tau = 0.5$, $WP_{\mathcal{C}} = 1$.

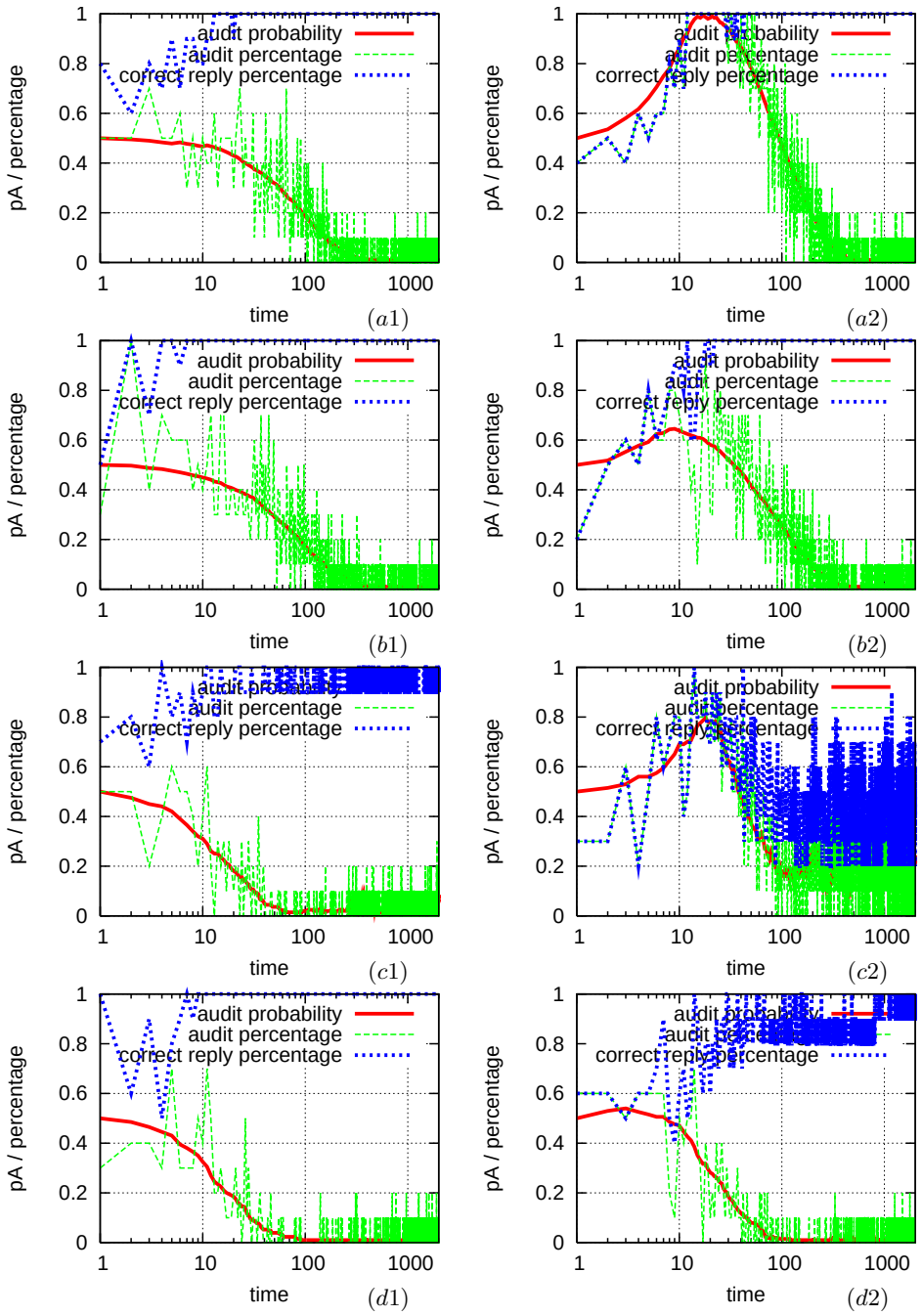


Fig. 16: Five covered workers and reputation Type 3 used. Parameters in all plots $WC_{\mathcal{T}} = 0.1$, $a_i = 0.1$, $\alpha = 0.1$, and $WB_{\mathcal{Y}} = 0.1$ (for the uncovered workers) / $WB_{\mathcal{Y}} = 1$ (for the covered workers). Plots depict the master's auditing probability as a function of time, audit percentage as a function of time and correct reply percentage as a function of time. Left, initial $p_C = 0.5$. Right, initial $p_C = 1$. First row $\tau = 0.1$, $WP_{\mathcal{C}} = 0$. Second row $\tau = 0.1$, $WP_{\mathcal{C}} = 1$. Third row $\tau = 0.5$, $WP_{\mathcal{C}} = 0$. Fourth row $\tau = 0.5$, $WP_{\mathcal{C}} = 1$.

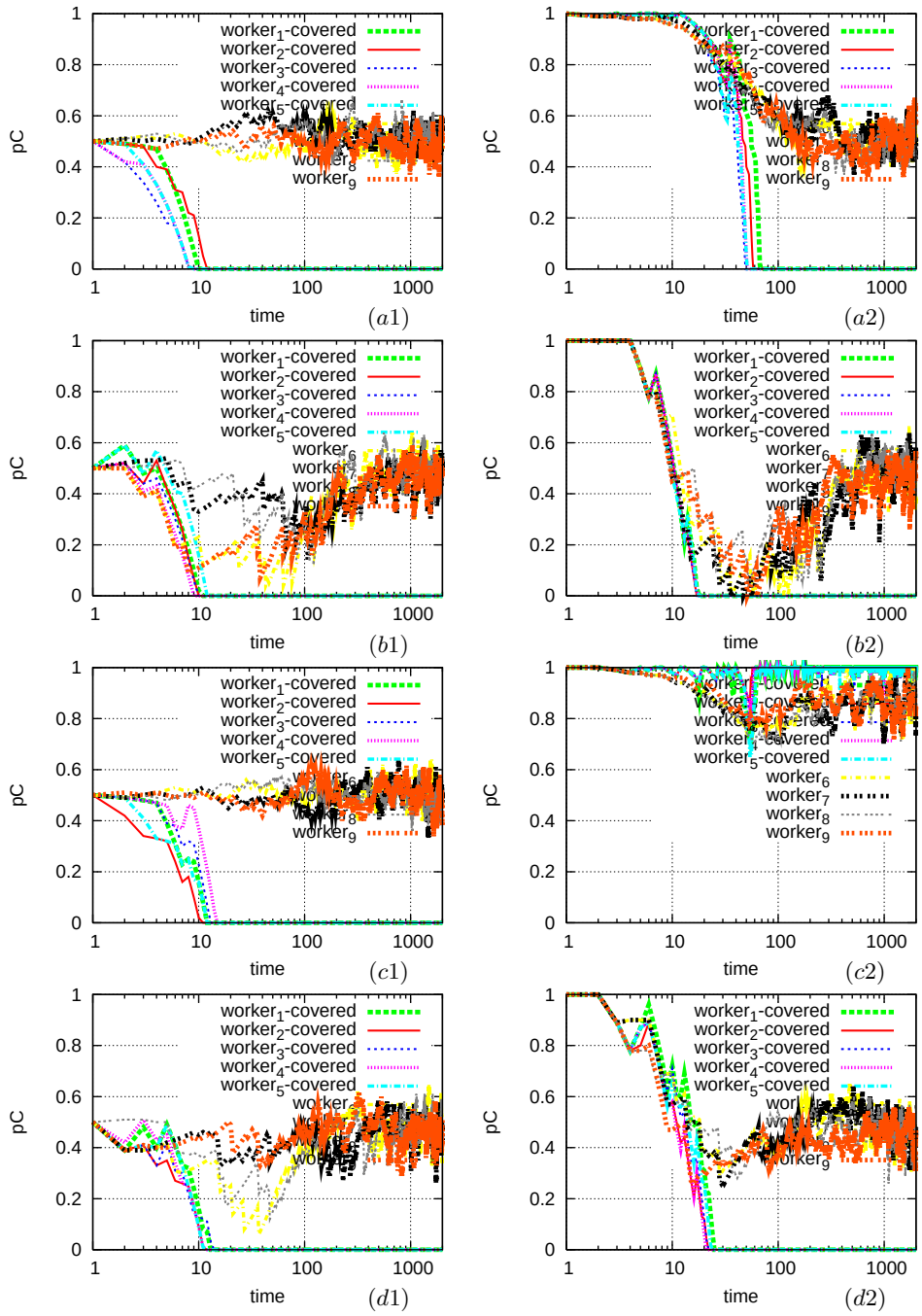


Fig. 17: Five covered workers and reputation Type 3 used. Parameters in all plots $WC_{\mathcal{T}} = 0.1$, $a_i = 0.1$, $\alpha = 0.1$, and $WB_{\mathcal{Y}} = 0.1$ (for the uncovered workers) / $WB_{\mathcal{Y}} = 1$ (for the covered workers). Plots depict the cheating probability for each worker as a function of time, for an individual realization. Left, initial $p_C = 0.5$. Right, initial $p_C = 1$. First row $\tau = 0.1$, $WP_{\mathcal{C}} = 0$. Second row $\tau = 0.1$, $WP_{\mathcal{C}} = 1$. Third row $\tau = 0.5$, $WP_{\mathcal{C}} = 0$. Fourth row $\tau = 0.5$, $WP_{\mathcal{C}} = 1$.

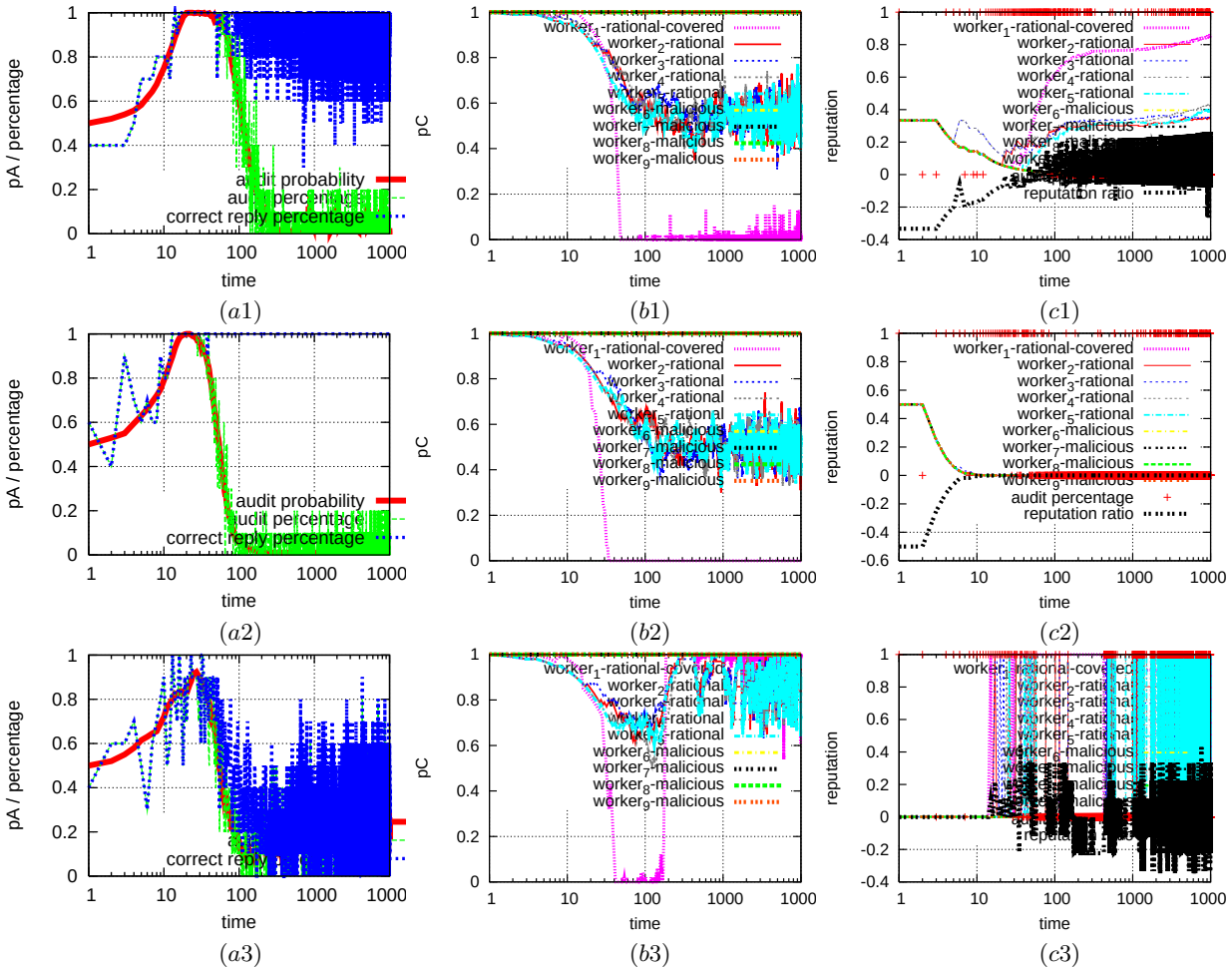


Fig. 18: Presence of 4 malicious and 5 rational workers, only 1 rational worker is covered. Top panel, reputation Type 1. Middle panel, reputation Type 2. Bottom panel, reputation Type 3. Parameters in all panels, initial $p_C = 1$, initial $p_{\mathcal{A}} = 0.5$, $WC_{\mathcal{T}} = 0.1$, $WP_{\mathcal{C}} = 0$ and $\alpha = 0.1$, $a_i = 0.1$. First column audit probability as a function of time, audit percentage as a function of time and correct reply percentage as a function of time. Second column workers' cheating probability as a function of time, for an individual realization. Third column workers' reputation as a function of time, audit occurrences as a function of time and reputation ratio as a function of time, for an individual realization.

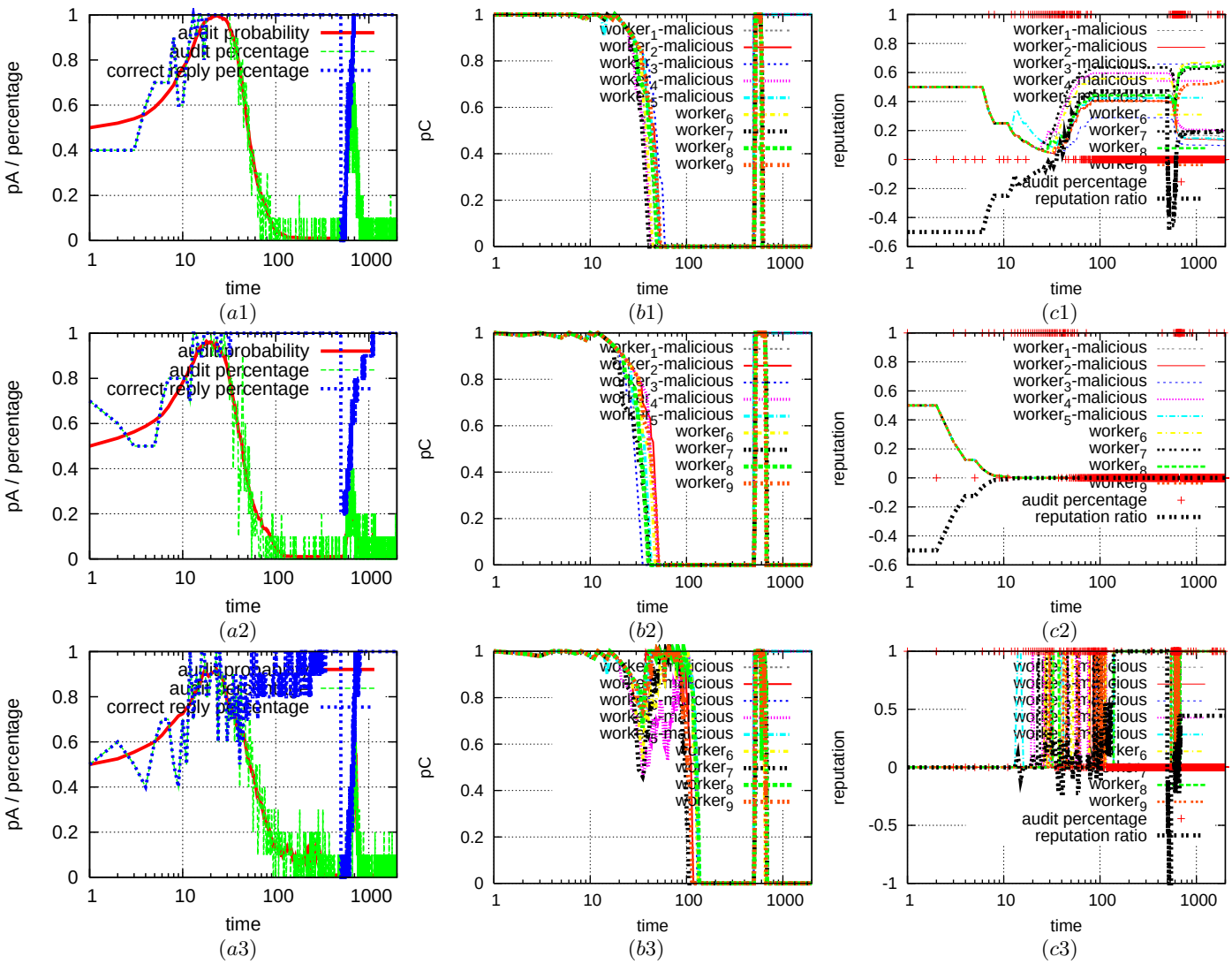


Fig. 19: Presence of 5 malicious workers on the 500th round. Top panel, reputation Type 1. Middle panel, reputation Type 2. Bottom panel, reputation Type 3. Parameters in all panels, initial $p_C = 1$, initial $p_{\mathcal{A}} = 0.5$, $WC_{\mathcal{T}} = 0.1$, $WP_{\mathcal{C}} = 0$ and $\alpha = 0.1$, $a_i = 0.1$. First column audit probability as a function of time, audit percentage as a function of time and correct reply percentage as a function of time. Second cheating probability for each worker as a function of time, for an individual realization. Third column workers' reputation as a function of time, audit occurrences as a function of time and reputation ratio as a function of time, for an individual realization.