Introducing Reputation Systems to the Economics of Outsourcing Computations to Rational Workers

Jassim Aljuraidan[1], Lujo Bauer[1], Michael K. Reiter[2], and Matthias Beckerle[1]

[1] Carnegie Mellon University, Pittsburgh, PA, USA
[2] University of North Carolina at Chapel Hill, Chapel Hill, NC, USA

Abstract. Outsourcing computation to remote parties ("workers") is an increasingly common practice, owing in part to the growth of cloud computing. However, outsourcing raises concerns that outsourced tasks may be completed incorrectly, whether by accident or because workers cheat to minimize their cost and optimize their gain. The goal of this paper is to explore, using game theory, the conditions under which the incentives for all parties can be configured to efficiently disincentivize worker misbehavior, either inadvertent or deliberate. By formalizing multiple scenarios with game theory, we establish conditions to discourage worker cheating that take into account the dynamics of multiple workers, workers with limited capacity, and changing levels of trust. A key novelty of our work is modeling the use of a reputation system to decide how computation tasks are allocated to workers based on their reliability, and we provide insights on strategies for using a reputation system to increase the expected quality of results. Overall, our results contribute to make outsourcing computation more reliable, consistent, and predictable.

1 Introduction

A powerful recent trend in computing has been the outsourcing of computation tasks to other parties. This trend has spanned the government and commercial sectors, with examples of outsourcing ranging from scientific computation (e.g., CERN's grid computing[3], and the Folding@home distributed computing project for disease research[4]) to commercial web and content delivery services (e.g., Netflix's use of Amazon's EC2[5]) and sensitive government computing tasks (e.g., the U.S. Central Intelligence Agency's use of Amazon's cloud[6] and the U.S. Department of State's use of Datamaxx[7]). The use of cloud infrastructure brings many advantages, including a high degree of flexibility and cost savings.

[3] http://home.web.cern.ch/about/computing/worldwide-lhc-computing-grid
[4] https://folding.stanford.edu/
[5] http://www.theatlantic.com/technology/archive/2014/07/the-details-about-the-cias-deal-with-amazon/374632/
[6] https://aws.amazon.com/cloudfront/
[7] https://www.datamaxx.com/

At the same time, outsourcing computation carries inherent risks for the party doing the outsourcing (which we call the *outsourcer*). Since the compute infrastructure is not under control of the outsourcing party, the correctness of these computations cannot necessarily be trusted. In particular, the parties to whom tasks have been outsourced (*workers*) may have an economic incentive to perform sub-standard work (e.g., to guess at answers, saving computational effort; or to use sub-standard hardware, increasing chance of data loss). To reassure customers, service level agreements (SLAs) and increasingly precise and complex certification requirements for cloud providers (e.g., the FBI's CJIS security policy requirements[8]) are often devised.

In parallel with such steps, crucial to increasing the trustworthiness of outsourcing is to understand how to leverage technical mechanisms for verifying that outsourced tasks are being performed correctly and to appropriately reward correct and penalize incorrect behavior. To that end, researchers have used game-theoretic models of outsourcing to determine the optimal strategies and parameters to be used by the party interested in outsourcing (e.g., [4, 18]). More specifically, recent work in the context of single-round games has shown how to wield incentives such as fines, budgets, and auditing rates to design optimal outsourcing contracts with one or two workers [18].

In this paper we extend this line of work on game-theoretic analysis of economic incentives to encompass richer, more realistic scenarios. In particular, our models consider two additional important factors. First, we use infinite-round (rather than single- or limited-round) games to model the realistic long-term interaction between the outsourcer and the workers. Second, we explore the use of two forms of simple reputation systems as a mechanism for outsourcers to choose how to direct future outsourcing tasks based on past performance and to model the realistic incentive of the outsourcing of future tasks depending on the successful completion of earlier ones. Using these models, we show which values of the game parameters (e.g., job cost, how much workers value future transactions as opposed to current ones) ensure that the workers have no incentive to cheat, i.e., that an equilibrium exists only when all workers play honestly. We calculate these values for five games representing different scenarios. In addition, we demonstrate the practical implications of our results.

This paper proceeds as follows. We describe related work in Section 2 and provide a more detailed problem description in Section 3. Section 4 presents our main results. Section 5 summarizes our findings and concludes the paper.

2 Background and Related Work

The problem of outsourcing computations to untrusted workers is an active area of research, with multiple general approaches being advanced simultaneously. One direction, surveyed by Walfish and Blumberg [24], uses advances in probabilistically checkable proofs (e.g., [10]) and/or interactive proofs (e.g., [8]) to

[8] https://www.fbi.gov/about-us/cjis/cjis-security-policy-resource-center

enable an outsourcer to determine that a worker probably performed an outsourced task correctly, at a cost less than that of performing the task itself. This approach to verification is largely complementary to the incentive frameworks that we propose here, in the sense that an outsourcer could use this approach to verify a worker's computation when it chooses to do so in the context of our framework. However, since this approach requires changes to the worker software that can increase the worker's computational cost by orders of magnitude (e.g., see [24, Fig. 5]), we do not consider its use here.

An alternative approach is to leverage trusted computing technologies to improve the trustworthiness of workers. For example, Trusted Platform Modules [23] permit a worker machine to attest to the software it runs; provided that the outsourcer trusts the worker's hardware, it can have confidence that the worker will execute the attested software to perform the outsourcer's task. The emergence of Intel's SGX [2, 9, 14] should make this approach more practical (e.g., [20]). Again, however, these approaches are complementary to ours; rather than leverage trusted hardware primitives to gain confidence in workers, we apply incentives, instead.

Use of game theory to reason about allocation of jobs to distributed entities has been considered previously, such as in the context of grid computing (e.g., [17, 26, 3]). The work that is most related to ours, however, is on incentive-based approaches that utilize rewards and fines to ensure the honesty of workers when outsourcing computations. Belenkiy et al. first analyzed the idea of using rewards, fines, and bounties (given to workers who catch other workers cheating) to incentivize (rational) workers to behave honestly [4]. In their work on computation-outsourcing contracts, Pham et al. explore the best options when making such contracts to ensure workers' honesty while minimizing the cost to the outsourcer [18]. While these works [4, 18] do cover scenarios with more than two workers, neither models multi-round interactions or the use of reputation systems as a mechanism to incentivize workers to be honest.

More recent work by Nojoumian et al. [16] uses a specific reputation system [15] to incentivize players in a secret-sharing scheme to cooperate. However, the mechanics of the secret-sharing game are different than the computation outsourcing game we are considering here. For example, the concept of testing or verifying results at a certain frequency and its effect on detecting cheaters in computation outsourcing does not have an equivalent in the secret-sharing game. In addition, the reputation system used in that work would be a poor fit in our scenario. That reputation system gives significant leeway to cheaters, which does not interfere with assuring honesty in the secret-sharing game but would make it significantly harder to guarantee honesty in our setting.

The implementation of a reputation system as used in our game of outsourcing has similarities to, but also important differences from, those of *game triggers* [7, 13]. Game triggers are strategies that make non-optimal choices for a fixed period after a trigger event (e.g., prices falling below some threshold or another player choosing to not cooperate) as a punishment, and are used by the players themselves to influence the behavior of other players in repeated

W_i	the worker i
IR_i	the initial reputation of W_i
N_i	the capacity of W_i, i.e., how many jobs W_i can handle per round
C	the cost for a worker to calculate the results for one job correctly ($C > 0$)
m	the markup or profit for each worker ($m > 0$)
α	the discount factor that reduces the utility of each round; the effective utility of round i, assuming the first round is round 0, is the round's utility multiplied by α^i ($0 < \alpha < 1$, except in Game 2.2, where $\alpha = 1$)
q_i	the probability, per job, that W_i will cheat ($0 < q_i \leq 1$)
t_i	the probability that the outsourcer will verify a job done by W_i
p	the probability that a worker who cheated on a job will be detected, given that the outsourcer verify the job
n	number of workers in group one (when there are two groups of workers)
G	total number of workers
N	number of jobs available in each round

Table 1. Summary of notation. We assume that cost (C), markup (m), the discount factor (α), and detection probability (p) are the same for all workers.

noncooperative games. For example, they are often used in games that model cartels [19, 5], where cartel members use these strategies, e.g., by selling goods at lower prices for a period of time to maintain higher prices overall. In our setting, reputation systems are (and are modeled as) an integral part of an infinite-round game and are imposed by the outsourcer, not the players (i.e., workers). With game triggers, punishments usually affect all players, while in our setting punishment affects only the offending player. Another difference between game triggers and our work is that with game trigger strategies each player in the game can choose whether to use them, while in our setting the outsourcer enforces the reputation system on all players. Finally, while game triggers are used in repeated games, the games in our setting are single games with multiple rounds with explicit state (i.e., player reputations) that influence the game at each round.

3 Problem Description and Notation

In this section we describe the aspects of our models and the assumptions we make, common to all the games that we explore. We also provide a brief explanation of our formal notation, which is summarized in Table 1.

All the games described in the paper share two main concepts. First, we include a simple *reputation* system, which assigns (dynamic) ratings to the workers and analyze how such systems can deter workers from cheating. Second, all the games in this paper have an infinite number of rounds; considering multiple rounds is an essential requirement for a reputation system to be beneficial.

3.1 Basic Game Structure

We have two classes of entities in our games. The first is the outsourcer. The outsourcer provides *jobs* in the form of computations. The second class of entities

are the workers that perform the computations and send the results back to the outsourcer. All our games have at least two workers. Moreover, in all these games the outsourcer is not an active player; that is, all the outsourcer's moves are based on a pre-specified algorithm, which we describe below, that decides how to distribute jobs to workers in each round of the game based on the game state (worker ratings).

The goal of all workers in the described games is to optimize their utility (i.e., benefits or gain). All parties are considered perfectly rational and all the games are noncooperative. That is, the workers are only interested in maximizing their own expected utility and are not cooperating with other players. Moreover, they are capable of performing any complex reasoning in order to achieve the maximum utility. The players only act maliciously (i.e., cheat) if it improves their expected utility. In addition, we don't take into account the risk-awareness of the workers, and we assume that all workers are risk neutral. This implies that they only care about the *expected* utility, and won't prefer one outcome over the other if both result in the same utility.

In order to make the total expected utility for each player finite, we employ a discount factor α ($0 < \alpha < 1$; except for Game 2.2 where $\alpha = 1$) per round, as is common. That is, the effective utility at round i is the round's utility multiplied by α^i. This discount factor is applied not only to calculate the utility but also to model the fact that in real life, people tend to value immediate gains more than gains in the future.

In each game we analyze, the goal will be to find the sufficient conditions to ensure that the only equilibrium that is possible is the one where all workers are honest. More specifically, the equilibrium we are considering is the Nash equilibrium, which requires that no single player be able to increase her utility by only changing her strategy, with strategies of all other players being the same.

We next provide a general description how the games in this paper are played out. The concrete games in Sections 4.1–4.2 provide additional details like the number and grouping of workers, the initial setup, and the specific strategies of the players.

- At the beginning of each round the outsourcer offers N jobs to all or some of the G available workers and expects correct results in return. Performing the computation for a single job costs each worker C ($C > 0$). For each job offered, a worker can either play honest or decide to cheat. If a worker cheats on a job, e.g., by guessing at the result instead of computing it, the cost of that job to the worker is considered to be 0. On returning a job's result, except as discussed below, each worker is paid $C + m$ by the outsourcer; we call m the markup or profit ($m > 0$).
  All strategies are mixed strategies; i.e., each choice is assigned a probability. For example, the probability of a worker to cheat on any given job might be 0.6. Each worker decides on a probability to cheat q_i ($0 \leq q_i \leq 1$).
- The outsourcer will (randomly) verify some of the results returned by workers; the probability that the outsourcer will verify a particular job's result from worker W_i is t_i. If a worker cheats and the outsourcer verifies the result,

then the outsourcer detects the cheating with probability p. If the outsourcer detects that a returned result is wrong, i.e., the worker cheated, the worker gets penalized through a loss of reputation (see Sec. 3.2). If the outsourcer does not detect that a worker cheated in a round, either because it did not verify any of that worker's responses or because its verification attempts for that worker failed to detect misbehavior, then (and only then) does the outsourcer pay the worker for all the jobs it completed in that round. That is, a worker is given only *post-payments* and, in particular, does not receive any payment in a round where its cheating is detected.

The the decisions to verify a result or to cheat are made randomly (Bernoulli distributions with parameters t_i and q_i, respectively) and each decision is independent from the others. Also, in the remainder of this paper, when we refer to a worker being honest, we mean that $q_i = 0$ for the entire game, as opposed to deciding not to cheat on a single job. Similarly, when refering to a worker cheating or being dishonest, we specifically mean that $0 < q_i \leq 1$ for an entire game. Moreover, we don't distinguish between a worker being honest because cheating will decrease her utility and her being honest because other external (e.g., moral) incentives. In both cases, we will refer to her as an honest worker.

A key aspect that we examine in this paper is penalizing workers through the loss of future work. For this approach a reputation system is needed.

3.2 Reputation Systems

Reputations systems, especially those for on-line markets, are usually designed with one goal in mind: to share and compare results of former interactions with different entities. That gives customers or buyers a way to compare the quality of, e.g., products, service providers, or sellers [6, 11].

Reputation systems differ in the assumptions made about service providers (or workers), with some assuming that all workers are inherently good or inherently bad [22], or that workers have a certain quality to be estimated [21, 25]. Kerr et al. showed that many reputation systems will fail in the presence of cheaters who may offer good service on some occasions but cheat on others [12]. In our work we only assume that all workers are (infinitely) rational players, and as such will cheat if that increases their utility, but not otherwise. We believe that this threat model is more suitable for our setting of outsourcing potentially security sensitive computations to external workers whose incentives are not usually aligned a priori with the outsourcer.

One potential weakness of some reputation systems, particularly in the presence of opportunistic cheaters, is that they are not very reactive regarding recent rating changes, since they often only calculate a mean score over all interactions. In such systems, a long-time seller with excellent reputation might not care about a single bad rating since it might have negligible impact on his total score. Hence, even if sellers try to maintain a good overall reputation they might still cheat from time to time.

In this work, we employ a reputation system with variants that we call Zero-Reset and Knock-Out. If the Zero-Reset penalty is applied, the reputation of the cheating worker is reset to zero. The reputation then increases by 1, as for workers who had not been penalized, each time the worker completes a round without being caught cheating. Jobs are assigned to the worker W_i with the highest reputation first, up to its capacity N_i. If more jobs are available than the workers with the highest reputation have capacity to handle, the workers with the next highest reputation will be assigned jobs, until all jobs are assigned. In case of a tie in reputation, the jobs are allocated randomly among the workers with the same reputation. If the Knock-Out penalty is applied, the cheating worker will no longer get jobs assigned by the outsourcer in any future round.

Both variants of the reputation system we study here are too strict for actual implementations. In reality, reputation systems should incorporate some leeway for accidental failures, such as user errors or network failures. Otherwise, workers will be discouraged from participating in the system. However, care must be taken to not give too much leeway as this will render systems susceptible to abuse by determined cheaters [12]. It is outside the scope of this paper to specify how much leeway should be given or what is the best reputation system to choose for actual implementations (which, we suspect, is application-dependent). For the purpose of this paper, we will assume that the actual implementation will be strict enough such that the conditions we derive for the various games we propose can serve as guidelines to narrow the space within which the incentives to all parties are aligned.

4 Improving Outsourcing with Reputation Systems

In this section, we show how a reputation system can be used to induce workers to maintain a high level of quality. All calculations are based on rational acting players that make optimal decisions regarding their own expected utility.

Table 2 shows the main differences between the games we explore. In general, we try to vary only one factor at a time between games, so as to make it possible to attribute different outcomes to specific variations in factors or assumptions.

In Section 4.1 we introduce the first set of games, which focuses on two workers that have, potentially different, initial reputations and play for an infinite number of rounds. The first game uses the Zero-Reset reputation system. The second game assumes that one worker has higher initial reputation, while the third game explores the effect of changing the reputation system to Knock-Out instead. After that, we investigate games with more than two workers in Section 4.2. We analyse two games, one with the workers split into two groups with a common choice for the cheating probability, and another game where workers are not grouped but no discount factor is considered (i.e., $\alpha = 1$).

Proofs for all the propositions can be found in our technical report [1].

Game	Num. workers	Unlim. capacity	Rep. system	α	Notable assumptions
G1.1	2	Yes	Zero-Reset	< 1	$IR_1 = IR_2$; $N_1 \geq N$; $N_2 \geq N$
G1.2		Worker 2	Zero-Reset	< 1	Same as G1.1 but: $IR_1 > IR_2$; $N \geq N_1$; $q_2 = 0$
G1.3		Yes	Knock-Out	< 1	Same as G1.2 but with Knock-Out
G2.1	any	No	Zero-Reset	< 1	2 groups of workers; workers in the same group have the same q; $N = 1$
G2.2			Zero-Reset	n/a	Same as G2.1 but each worker has individual q_i; $N = 1$

Table 2. Summary of games examined in this paper and their main assumptions.

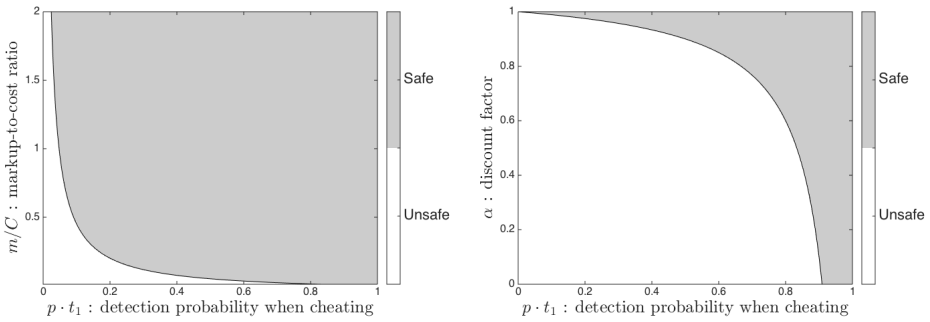
4.1 First Set of Games: Two-worker Games

We start our analysis with three games in which we consider scenarios with exactly two workers. We vary the capacity of the workers (we consider the case when both workers have more capacity than there is work available, as well as when one worker has insufficient capacity to perform all available work) and the reputation system we try.

Game 1.1: Symmetric initial reputation In this first game we test our reputation system by using a simple scenario with only two workers. Both workers have the capacity to handle all the available jobs in each round ($N_1 \geq N$; $N_2 \geq N$), both workers start with the same initial reputation ($IR_1 = IR_2$), and both have the option to cheat ($q_1 \geq 0$; $q_2 \geq 0$).

Proposition 1 *For Game 1.1, with parameters N, m, C, α, p, t_1, and t_2, if one worker (W_2) is always honest and one of the following conditions holds, then the only possible Nash equilibrium in the game will be the one where both workers are honest.*

- *$N = 1$, and*
$$\frac{m}{C} > (1-\alpha)\frac{1 - p \cdot t_1}{p \cdot t_1} \tag{1}$$
- *Or, $N = 2k$, for some positive integer k, and*
$$\frac{m}{C} > (1-\alpha)(1 - p \cdot t_1)\frac{(1 - p \cdot t_1)^{N/2-1}}{1 - (1 - p \cdot t_1)^{N/2}} \tag{2}$$
- *Or, $N = 2k + 1$, for some positive integer k, and*
$$\frac{m}{C} > (1-\alpha)(1 - p \cdot t_1)\frac{(1 - p \cdot t_1)^{k-1}(1 - \frac{1}{2}p \cdot t_1)}{1 - (1 - p \cdot t_1)^{k}(1 - \frac{1}{2}p \cdot t_1)} \tag{3}$$



(a) Discount factor (α) set to 0.95. (b) Markup-to-cost ratio ($\frac{m}{C}$) set to 0.1.

Fig. 1. The safe region for the case of $N = 1$ in Game 1.1.

Discussion The introduction of a reputation system leads to a clear benefit: as long as one worker (W_2) is honest and one of the conditions in Prop. 1 holds, the other worker (W_1) is forced to be honest too. Only the case where both workers decide to cheat stays unresolved for this game. In this case a repeating cycle of two stages takes place. Each cycle begins with both workers having equal reputations, and thus the jobs will be distributed equally (50%/50%). The first stage ends when one of the workers is detected cheating. In the second stage of a cycle the other worker will get all the jobs (100%/0% or 0%/100%); this stage lasts until this worker is detected cheating, which leads to an equal reputation of zero for both workers. The cycle repeats. This case results in a *noncentral hypergeometric distribution* for the job assignments, for which we have not yet found a closed-form solution. However, we will discuss the solution for the special case when $N = 1$ within the discussion of Game 1.3.

In Fig. 1a we plot the *safe* region, i.e., where not cheating is the only equilibrium, in terms of detection probability when cheating ($p \cdot t_1$) and the markup-to-cost ratio (m/C), with the discount factor set to 0.95. We do the same in Fig. 1b, but this time we fix the value of the markup-to-cost ratio to $m/C = 0.1$.

From the figures we can see that if the detection probability, i.e., the product of $p \cdot t_1$, is very small, then condition in Prop. 1 will be invalid for reasonable values of the markup-to-cost ratio and the discount factor. For example, if $p \cdot t_1 = 0.005$, then even the slightly extreme values of 0.99 and 1 for the discount factor and the markup, respectively, will not result in a safe outsourcing. In addition, it seems that discount factor α should be high enough, i.e., well above 0.90, for the the valid range of $p \cdot t_1$ to be reasonable.

Let's imagine a company (outsourcer) wants to outsource scanning of incoming email attachments to a third party (worker). The results of this game show that, unless there is a large penalty or a high markup-to-cost ratio, a worker will have no incentive to be honest, because the probability of detection is very low in this case; an arbitrary email attachment has a very low chance of being mali-

cious [9]. However, actively increasing p, for example by sending known malicious files to the third party, can significantly improve the chances that a worker will have no incentive to cheat.

In the next game we investigate asymmetric initial reputation combined with limited capacity of a worker.

Game 1.2: One worker with higher initial reputation but limited capacity The assumption in game 1.1 that workers have equal initial reputation and unlimited capacity is a limitation that we remove in this game. We already considered the case when the initial reputations are equal, so we will assume that they are strictly different here.

Without loss of generality, we give W_1 a higher initial reputation compared to W_2. In addition, W_1's capacity is limited, i.e., W_1's capacity can always be saturated ($N > N_1$). We also tested the setup where both workers have limited capacity but since the interesting parts of the results were similar to Game 1.1, we only describe the former case here. As in game 1.1, both workers have the option to cheat ($q_1 \geq 0$; $q_2 \geq 0$).

Proposition 2 *For Game 1.2, with parameters m, C, α, p, t_1, and N_1, if worker 2 is always honest and one of the following conditions holds, then the only possible Nash equilibrium in the game will be the one where worker 1 is also honest.*

- $N_1 = 1$, *and*

$$\frac{m}{C} > (1-\alpha)\frac{1-p\cdot t_1}{p\cdot t_1} \tag{4}$$

- $N_1 > 1$, *and*

$$\frac{m}{C} > (1-\alpha)(1-p\cdot t_1)\frac{(1-p\cdot t_1)^{N_1-1}}{1-(1-p\cdot t_1)^{N_1}} \tag{5}$$

Discussion This game shows that even with initially asymmetric ratings, the workers can still deter each other from cheating, if one of them is honest. Again the case where both workers decide to cheat was not resolved for the same reason as in Game 1.1. In fact, after both workers are detected cheating for the first time, they both will have a reputation of zero, which is almost exactly the situation in game 1.1. The only difference will be that W_2 will always have a number of jobs assigned to her ($N - N_1$), regardless of what happens in game because of the limited capacity of W_1. However, these job assignments won't affect the equilibrium, because they are fixed and won't change during the entire game, and won't affect the rest of the assignments.

In the next game we investigate how changing our reputation system to the second variant (Knock-Out) influences the results.

[9] 2.3% and 3.9% in the second and third quarters of 2013, respectively, according to a Kaspersky Lab study (http://www.kaspersky.ca/internet-security-center/threats/spam-statistics-report-q2-2013)

Game 1.3: A Knock-Out reputation system In the third game we want to test the Knock-Out variant of our reputation system (see Sec. 3 for more details). The intuition is that the Knock-Out variant is more effective in deterring workers from cheating in comparison to the Zero-Reset variant tested in former games. This game is exactly like Game 1.1, except for the reputation system. In Game 1.1, we showed the result for both even and odd number of jobs. In this game we will only show the result for an even number of jobs, in adition to the case where there is only one job. As we saw in Game 1.1, the result for an odd number of jobs will be only slightly different.

Proposition 3 *For Game 1.3, with parameters m, C, p, t_1, t_2 and N, if the following condition holds, then the only possible Nash equilibrium in the game will be the one where both workers are honest.*

Either $N = 1$ or $N = 2k$, for some positive integer k, and for both workers, and for all possible values of q_1 and q_2, the following holds ($i = 1, 2$) [10]

$$\frac{\mathrm{d}f_i}{\mathrm{d}q_i} \cdot h_i < \frac{\mathrm{d}h_i}{\mathrm{d}q_i} \cdot f_i \tag{6}$$

where

$$f_i = u_i(1-\beta_i)\left[\beta_i(1-\beta_j)[1-\alpha(1-\hat{\beta}_i)] + \beta_j[1+\alpha-\alpha(2B+\hat{\beta}_i-2\hat{\beta}_i \cdot B)]\right]$$

$$h_i = (1-B)(1-\alpha \cdot B)[1-\alpha(1-\hat{\beta}_i)]$$

$$B = (1-\beta_1)(1-\beta_2)$$

$$\beta_i = \begin{cases} q_i \cdot p \cdot t_i & N = 1 \\ 1-(1-q \cdot p \cdot t_i)^{N/2} & N \text{ is even} \end{cases}$$

$$\hat{\beta}_i = \begin{cases} \beta_i & N = 1 \\ 1-(1-q_i \cdot p \cdot t_i)^{N} & N \text{ is even} \end{cases}$$

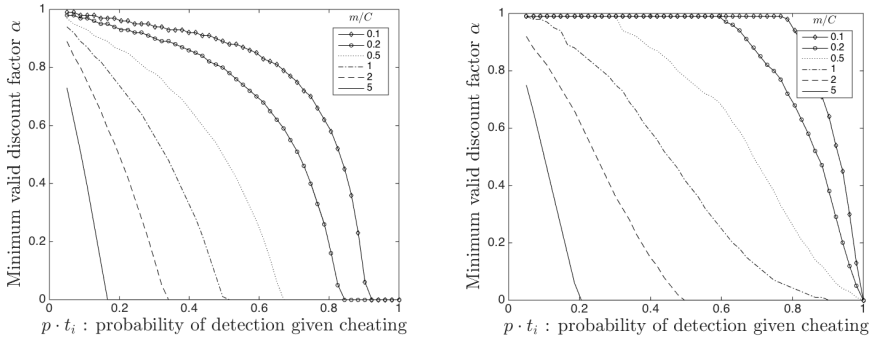
$$u_i = N(m + C \cdot \hat{q}_i)$$

$$j = 3 - i$$

Discussion The switch to the second variant of the reputation system did not change the general results from Game 1.1 for the case where at least one of the workers is honest. In essence, the fact that one worker is honest makes the effect of both reputations systems the same from the point of view of the other worker. Unfortunately, since we were not able to get a result for the case where both workers are dishonest and $N > 1$ in Game 1.1 (and 1.2), we cannot compare it with the result of this game. However, we can compare the results for the case when $N = 1$.

Regarding the condition for the case of two dishonest workers, in Fig. 2a we plot the minimum required discount factor (α) versus the probability of detection given a cheating worker for several values of the markup-to-cost ratio. We obtained the minimum values for α using numerical minimization. Notice

[10] $\frac{\mathrm{d}y}{\mathrm{d}q_i}$ is the derivative of y with respect to q_i.



(a) Knock-Out (Game 1.3). (b) Zero-Reset (Game 1.1).

Fig. 2. The minimum discount factor vs $p \cdot t_i$ (when $N = 1$ and $t_1 = t_2$). For example, at a markup-to-cost ratio of 0.2 and a detection probability (given cheating) of 0.8, the minimum discount factor needed in order to deter cheaters is 0.2 with the Knock-Out strategy and 0.7 with Zero-Reset.

that the required values for a discount factor are still quite high. Although we did not have a general solution for the same case in the Zero-Reset setting, we did find the solution for the special case of $N = 1$. Fig. 2b shows the results for that case. Comparing the two figures confirms the intuition that smaller values of α are needed with the stricter Knock-Out system than with Zero-Reset.

4.2 Second Set of Games: Many-worker Games

The previous games considered only two workers. In this section, we investigate games with more workers. To derive a closed form expression for the expected utility for each worker, we had to limit the number of available jobs per round to one, i.e., $N = 1$. In the first game, we consider a special case, where the workers are split into two groups with a common cheating probability. In the second game, we relax this condition and allow every worker to chose her own cheating probability, but to retain our ability to calculate the expected utility, we will assume that the discount factor is not applied, i.e., $\alpha = 1$.

Game 2.1 This is the first game where we test the reputation system where we allow more than two workers. The workers are divided into two groups. Each group will chose a cheating probability at the beginning, and adhere to it through out the game. The members of groups 1 and 2 will cheat with probabilities q_1 and q_2, and their sizes are n and $G - n$, respectively. Where G is the total number of workers. In addition, we assume that $N = 1$ and $\alpha < 1$. The reputation system will be Zero-Reset in this game.

Proposition 4 *In Game 2.1, with the parameters G, m, C, α, p, and t ($t_1 = t_2 = t$), and one of the following conditions hold, then the only possible Nash equilibrium in the game will be the one where both group of workers are honest. Also let $e = \frac{m}{C}$, $\gamma = \frac{1-\alpha}{\alpha}$, and $\omega = -e \cdot p \cdot t + 1 - p \cdot t$.*

- *One group of workers (of size $G - n$) is honest and the following holds for the other group (of size n):*

$$\frac{n}{G} < 1 - \frac{\gamma}{e}(1 - p \cdot t) + p \cdot t \tag{7}$$

- *Either $\omega \leq 0$ or the following holds:*

$$\frac{\omega \cdot p \cdot t + \omega \cdot \gamma}{e \cdot p \cdot t + \omega \cdot p \cdot t} < \frac{n}{G} < \frac{e \cdot p \cdot t - \omega \cdot \gamma}{e \cdot p \cdot t + \omega \cdot p \cdot t} \tag{8}$$

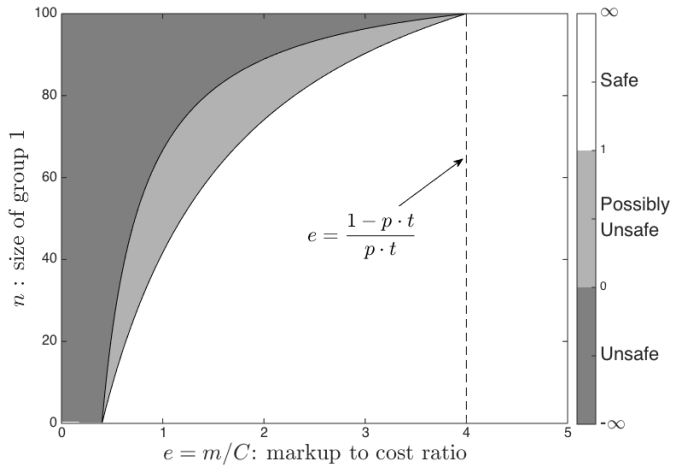


Fig. 3. The safe, unsafe, and possibly unsafe regions, w.r.t. group 1 size and the markup-to-cost ratio, when group 2 is always honest. ($G = 100$ and $\alpha = 0.95$)

Discussion The following equations are the precursor for condition (8) (refer to the proof in [1] for details).

$$q_2 < F(n) = \frac{e}{\omega} \cdot \left(\frac{G - n}{n}\right) - \frac{\gamma}{p \cdot t} \cdot \frac{G}{n} \tag{9}$$

$$q_1 < F(G - n) = \frac{e}{\omega} \cdot \left(\frac{n}{G - n}\right) - \frac{\gamma}{p \cdot t} \cdot \frac{G}{G - n} \tag{10}$$

To understand the behavior of $F(n)$ we will use the plot in Fig. 3. The value of the discount factor (α) is chosen to be 0.9, because, as we saw earlier, values below 0.9 for α will render the game too biased toward cheating in games with many (or infinite) number of rounds. In addition, we will choose the detection probability to be 0.2, i.e., $p \cdot q = 0.2$, which will result in $\omega = 0$ at $e = 4$.

There are three regions in this plot (Fig. 3). The first region, colored in dark grey, covers the range of values of n and e where $F(n) < 0$. This is the unsafe region; since $F(n) < 0$, then there is no valid nonzero value for q_2 ($0 < q_2 \leq 1$) that will cause (9) to be always true, i.e., $\frac{d\mathrm{E}[u_1]}{dq_1} > 0$. In other words, group 1 workers will have the incentive to always cheat, since more cheating will result in an increased utility. In this case, equilibrium is only possible when group 1 always cheats.

The second region, colored in white, represent the range of values where $F(n) > 1$. This is the safe region where, regardless of the value of q_2, the utility of group 1 workers will *decrease* when q_1 is increased, which will incentivize them not to cheat. In the safe region, equilibrium is only possible when group 1 is honest. In the grey region ($0 \leq F(n) \leq 1$) whether group 1's incentive is to cheat or not depends on q_2, which makes it possibly unsafe.

Ideally we want to control the parameters of the game to be always in the safe region to guarantee that the (untrusted) workers in group 1 have no incentive to cheat. However, this does not say anything about group 2 workers. This is fine if we either think that group 1 is trustworthy enough, or somehow can ensure their honesty. It can be seen in Fig. 3 that even at the relatively high markup-to-cost ratio of 1, we need the untrusted group size to be less than 50%.

An interesting scenario to consider is when we do not have any guarantees of honesty about both groups. Can we still achieve similar results that guarantee that they have no incentive to cheat? For such guarantees we need both (9) and (10) to hold, i.e., we need to be in the safe region of both $F(n)$ and $F(G - n)$. Figure 4 shows the intersection of both regions. Note that the conditions in Prop. 4 correspond to this intersection. It can be seen from the plot that a very high markup-to-cost ratio is needed to have reasonable guarantees that there is no incentive to cheat for both groups (at $\alpha = 0.95$ and $p \cdot t = 1$ nonetheless). Intuitively, this means that the outsourcer needs to promise to pay the workers enough to ensure that they value their future transactions more than the potential gains of cheating. Moreover, in order to have these guarantees, the relative sizes of the two groups need to be balanced. In this case, if the size of one of the groups is less than one fourth the size of the other group ($< 20\%$ of the total), then the guarantees cannot hold. In addition, the lower the markup-to-cost ratio, the more balanced these groups need to be. The same applies to α: with lower values for α, more balanced groups are required in order for them to deter each other from cheating.

Game 2.2 In the previous games we divided the workers into two groups, where each group shares a common cheating probability. In this game, each worker is allowed to chose her own cheating probability. In addition, we consider the case

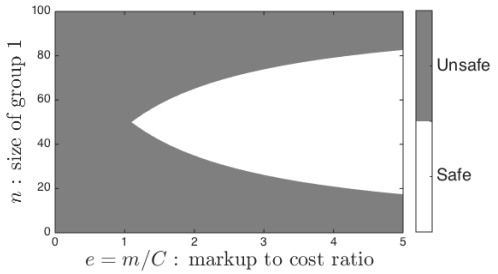


Fig. 4. The safe region, w.r.t. group size and markup-to-cost ratio, when both groups are not assumed to be honest. ($G = 100$ and $\alpha = 0.95$)

were there is no discount factor (i.e., $\alpha = 1$). Since in this case it is not possible to find a Nash equilibrium considering the total utility, due to the infinite number of rounds, we analyze the expected utility per round. In addition to having no discount factor, we also restrict the number of jobs per round to one, otherwise finding an equilibrium will be too complex of a problem, probably without a closed form solution for the expected utility.

Proposition 5 *In Game 2.2, with parameters G, m, C, p, and t_i, for $i = 1, .., G$, if the condition below holds for at least one worker then the only equilibrium (as defined above) will be the one where every player is honest.*

$$\frac{C}{m} \cdot \frac{1 - p \cdot t_i}{p \cdot t_i} - 1 < \frac{1}{p} \cdot \sum_{j \in [1,G], j \neq i} \frac{1}{t_j} \tag{11}$$

Discussion Due to the lack of discounting ($\alpha = 1$) we cannot directly compare the result here with the conditions in previous games. However, the same basic observations are true here. That is, we need a high enough markup-to-cost ratio and a reasonably high detection probability, i.e., $p \cdot t_i$. Of note here is that the sum $\sum_{j \in [1,G], j \neq i} \frac{1}{t_j}$ has a minimum value of $G - 1$, which means that the more workers we have available, the easier it is to incentivize the workers to be honest. Another observation, which might not be obvious, is that because we only need to incentivize one worker to be honest to force the others to also be honest, it actually makes sense to test those other workers less frequently. This is true because the more frequently we test the other workers, the more utility they will gain (from cheating) at the expense of the cheating utility of this one worker we are trying to incentivize.

5 Conclusion

In this paper we explored several games that model the outsourcing of computation and the long-term interaction between the outsourcer and the workers

using models with infinite number of rounds. Instead of using penalties to deter workers from cheating, we used a reputation system and the potential loss of future work as way to incentivize workers to not cheat. Such a reputation system could not have been modeled with a one- or two-round game, or even (stateless) repeated games. While we have not utilized penalties in our games, a natural direction of future work is to formulate and analyze games where penalties are used in conjuction with reputations to ensure worker honesty.

We demonstrated that if specified conditions are met, then workers are compelled to behave honestly in our models. Moreover, these conditions enable us to calculate parameter settings that suffice to compel worker honesty. For example, these conditions enable us to calculate the outsourcer's detection probability ($p \cdot t_i$) that suffices to compel workers, from which we can then adjust p and/or t_i to ensure that this detection probability is met. Doing so is important for scenarios where the *a priori* probability p of detecting the worker cheating (when checked) is low, e.g., because the distribution over honest worker responses is so biased that a cheating worker can guess the correct response with high probability without performing the computation (e.g., guessing that an email attachment is benign, without actually checking it). In such cases, our conditions provide guidelines for increasing the testing rate t_i or the detection probability p when response are checked (e.g., by sending a larger fraction of malicious attachments for checking) to ensure worker honesty.

Acknowledgments

This work was supported in part by NSF grant 1330599 and by the Army Research Laboratory under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

1. Aljuraidan, J., Bauer, L., Reiter, M.K., Beckerle, M.: Introducing reputation systems to the economics of outsourcing computations to rational workers. Tech. Rep. CMU-CyLab-16-001, Carnegie Mellon University (Jan 2016)
2. Anati, I., Gueron, S., Johnson, S., Scarlat, V.: Innovative technology for CPU based attestation and sealing. In: Workshop on Hardware and Architectural Support for Security and Privacy (2013)
3. Awerbuch, B., Kutten, S., Peleg, D.: Competitive distributed job scheduling (extended abstract). In: 24th ACM Symposium on Theory of Computing. pp. 571–580 (1992)

4. Belenkiy, M., Chase, M., Erway, C.C., Jannotti, J., Küpçü, A., Lysyanskaya, A.: Incentivizing outsourced computation. In: Proceedings of the 3rd International Workshop on Economics of Networked Systems. NetEcon '08 (2008)
5. Briggs, III, H.: Optimal cartel trigger strategies and the number of firms. Review of Industrial Organization 11(4), 551–561 (1996)
6. Commerce, B.E., Jsang, A., Ismail, R.: The beta reputation system. In: 15th Bled Electronic Commerce Conference (2002)
7. Friedman, J.W.: A non-cooperative equilibrium for supergames. The Review of Economic Studies 38(1), pp. 1–12 (1971)
8. Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: Delegating computation: Interactive proofs for muggles. In: 40th ACM Symposium on Theory of Computing (May 2008)
9. Hoekstra, M., Lal, R., Pappachan, P., Rozas, C., Phegade, V., del Cuvillo, J.: Using innovative instructions to create trustworthy software solutions. In: Workshop on Hardware and Architectural Support for Security and Privacy (2013)
10. Ishai, Y., Kushilevitz, E., Ostrovsky, R.: Efficient arguments without short PCPs. In: 22nd IEEE Conference on Computational Complexity (Jun 2007)
11. 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)
12. Kerr, R., Cohen, R.: Smart cheaters do prosper: Defeating trust and reputation systems. pp. 993–1000. AAMAS '09 (2009)
13. Krugman, P.R.: Trigger strategies and price dynamics in equity and foreign exchange markets. Tech. Rep. 2459, National Bureau of Economic Research (1987)
14. Mckeen, F., Alexandrovich, I., Berenzon, A., Rozas, C., Shafi, H., Shanbhogue, V., Savagaonkar, U.: Innovative instructions and software model for isolated execution. In: Workshop on Hardware and Architectural Support for Security and Privacy (2013)
15. Nojoumian, M., Lethbridge, T.: A new approach for the trust calculation in social networks. In: Filipe, J., Obaidat, M. (eds.) E-Business and Telecommunication Networks, pp. 64–77 (2008)
16. Nojoumian, M., Stinson, D.: Socio-rational secret sharing as a new direction in rational cryptography. In: Grossklags, J., Walrand, J. (eds.) Decision and Game Theory for Security, pp. 18–37 (2012)
17. Penmatsa, S.: Game Theory Based Job Allocation/Load Balancing in Distributed Systems with Applications to Grid Computing. Ph.D. thesis, The University of Texas at San Antonio (2007)
18. Pham, V., Khouzani, M., Cid, C.: Optimal contracts for outsourced computation. In: Poovendran, R., Saad, W. (eds.) Decision and Game Theory for Security. LNCS (2014)
19. Porter, R.H.: Optimal cartel trigger price strategies. Journal of Economic Theory 29(2), 313–338 (1983)
20. Schuster, F., Costa, M., Fournet, C., Gkantsidis, C., Peinado, M., Mainar-Ruiz, G., Russinovich, M.: VC3: Trustworthy data analytics in the cloud using SGX. In: 36th IEEE Symposium on Security and Privacy (May 2015)
21. Teacy, W., Patel, J., Jennings, N., Luck, M.: Travos: Trust and reputation in the context of inaccurate information sources. Autonomous Agents and Multi-Agent Systems 12(2), 183–198 (2006)
22. Tran, T., Cohen, R.: Improving user satisfaction in agent-based electronic marketplaces by reputation modelling and adjustable product quality. pp. 828–835. AAMAS '04 (2004)
23. Trusted Computing Group: Trusted platform module main specification, version 1.2, revision 103 (2007)

24. Walfish, M., Blumberg, A.J.: Verifying computations without reexecuting them. Communications of the ACM 58(2) (Feb 2015)
25. Whitby, A., Jøsang, A., Indulska, J.: Filtering out unfair ratings in bayesian reputation systems. In: Proc. 7th Int. Workshop on Trust in Agent Societies. vol. 6, pp. 106–117 (2004)
26. Yagoubi, B., Medebber, M.: A load balancing model for grid environment. In: 22nd International Symposium on Computer and Information Sciences. pp. 1–7 (2007)