Incentive Engineering for Outsourced Computation in the Face of Collusion

MHR. Khouzani, Viet Pham, and Carlos Cid

Information Security Group
Royal Holloway, University of London, U.K.
arman.khouzani,viet.pham.2010,carlos.cid@rhul.ac.uk

Abstract. One of the major security concerns in outsourced computation is whether the computation is performed honestly. Although cryptographically verifiable computation promises (almost) certain detection of dishonest computations, its computational complexity is still a distance away from practical, and only justifies its use in mission-critical applications, e.g., where there is suspicion of malicious actors. An alternative approach is to provide the right economic incentives for honest computation, assuming that contractors are lazy rather than malicious. This assumption finds applications in less critical settings, e.g., most civil-purpose computations. Previous works on this approach have established feasible and optimal (in terms of the outsourcer's expected costs) contracts for single and non-colluding multiple contractors. In this paper we review these results, and take a further step by considering the effect of side channel information and collusion among contractors. Through careful design of incentives, we demonstrate that it can still be optimal to use multiple contractors even if they are able to collude.

Keywords: Outsourced Computing, Moral Hazard, Collusion, Game Theory

1 Introduction

Outsourcing of computational tasks have appeared in research projects on large-scale data, such as SETI@Home: Search for Extra-Terrestrial Intelligence, and Folding@Home: protein folding simulations and computational drug design. Businesses from different sections ranging from energy infrastructure to life sciences and healthcare have realized the benefits of outsourcing their data and computation, and "moving to the cloud" [1]. Cloud-based computing for a fee have become a reality [2]. This trend is expected to continue with increasing ubiquity of need for computational power, for instance as Internet of Things (IoT) will become a reality, where capacity-constrained devices face the needs to becoming more and more intelligent.

Meanwhile, one of the biggest challenges with outsourcing computational tasks is to guarantee the honesty of the contractor(s). In mission-critical environments like in the military, a computation often requires a stringent level of security against malicious adversaries. With such level of security in mind, cryptographic verification methods, such as homomorphic encryption [3] and Probabilistically Checkable Proofs (PCPs) [4, 5], have been developed (and are being improved upon) that provide a proof of correctness for each and every outsourced computation task. In general, these techniques require a heavy and expensive preparation step, followed by a relatively less expensive computation. The argument for their efficiency, although not always applicable, is that since the preparation phase can be "reused" over multiple computations with the same representation, the time average cost for each result becomes asymptotically affordable. However, the computation overhead of these methods is still an issue for large scale practicability.

For less critical applications, a more relaxed notion of security may be applicable: that contractors/agents are not necessarily malicious, but are rather just *lazy*. That is, they are tempted to return random guesses instead of the result of honest computation in order to save their computing cost – or equivalently, reap in more reward by serving more tasks than their computational capacity would have allowed, were they computing honestly. This applies, for example, when contractors are cloud providers with no business conflicts with the outsourcer. For such cases, economic approaches can be used to create incentives for honest computation, at a possibly much lower cost on the outsourcer than a heavy-weight cryptographic method. This is especially true when computation cannot be reused. Specifically, the honesty of contractors can be encouraged through plausible use of rewards, random auditing (for instance, through occasional re-computation of the tasks) and penalties for incorrect results, as discussed in [6].

An alternative to the direct random auditing of tasks by the outsourcer is the simple technique of *redundancy* as proposed by [7]: (sometimes) the same task is assigned to multiple agents and the returned results are compared. This scheme is widely used in large research projects like Folding@Home. Assuming that the likelihood that guessed results are the same is negligible (by e.g. requiring snapshots of intermediate steps), the redundancy scheme provides a strong means of verification and lifts the burden of auditing from the outsourcer. This can be specially helpful as the whole paradigm for outsourcing the computations in the first place is the computational limitation of the outsourcer.

However, the redundancy scheme is susceptible to two potential threats that previous research has neglected to address. First, note that irrespective of the method of auditing (directly or through redundancy), it is critical that the tasks for which the auditing occurs are not earmarked, or else the contractor would know when it could get away with cheating. In cases where the contractors cannot communicate, this is a good assumption. However, the agents may be able to find a *side channel* that enables them to find out whether the same task is outsourced to multiple agents for redundancy check or not. Second, the contractors may further collude to report the same guessed result and hence undermine the whole scheme. It is the subject of this paper to examine whether the outsourcing scheme can still provide any benefit in the face of these two challenges, and if so, how exactly. Specifically, we consider an outsourcer that can use a hybrid of direct auditing and auditing through redundancy. The outsourcer designs a contract comprised on probabilistic auditing, probabilistic redundancy, rewards and punishments, while being wary of its limitations such as bounded computation capacity for auditing, cost of auditing, maximum fine that can be practically levied on a cheater and its maximum budget. We develop the optimal contracts in closed-form in the presence of a side channel (Proposition 2) and compare its characteristics with optimal contracts in the absence of such side information. Moreover, we develop two "bounty" schemes and provide sufficient conditions to make redundancy scheme a preferred method over direct auditing even in the presence of collusion (Corollaries 2, 3 and 4). This paper provides valuable insights in the potentials and limitations of redundancy scheme as a means of honesty enforcement in outsourced computation.

Literature Review

A line of research is focused on designing reliable verification techniques for outsourcing of special-purpose computations. For instance, [8] investigates outsourcing of linear optimizations. Another notable examples are queries on outsourced databases, including typical queries [9, 10] and aggregation [11]. Their main paradigm is for the querier to rely on trusted information directly given by the data owner (outsourcer) to verify the results returned by the servers.

Verification methods for general-purpose computing also appear in several remarkable works. In [12] verification is performed by re-executing parts of the computation. A variation is presented in [13] in which the authors utilize redundancy over multiple agents, assuming that at least one of them is honest. Outsourced computation has also caught attraction in cryptographic research: in a seminal work, the authors of [3] formally define verifiable computation and give a non-interactive solution. Their solution uses Yao's garbled circuits to represent the computation and homomorphic encryption to hide such circuits from the agents. More efficient but interactive solutions that use *probabilistically-checkable proofs* (PCPs) have since been developed such as Pepper [4] and Ginger [5]. Such cryptographic methods, however, usually introduce a large computational overhead on the prover or the verifier, or both.

Incentive-based solutions such as [6, 7] have studied contracts that the outsourcer may offer to the agents and through a combination of auditing, fines and rewards, honest computation is enforced. These works mostly focus on establishing feasibility of individually rational honesty-enforcing outsourcing contracts through direct probabilistic auditing in [6] and redundancy scheme in [7], respectively. In our previous work [14], we investigated the cost of the outsourcer and developed optimal hybrid contracts that minimize the aggregate cost of the outsourcer, as a combination of compensation of the agents and auditing overheads, while capturing many practical constraints such as boundedness of penalties in practice, limited computational capacity of the outsourcer, its capped budget, *etc.* In [14], the agents were assumed to be non-communicating and non-colluding. In the current paper, we explicitly relax these two assumptions and re-examine the optimal contracts. We provide provably optimal contracts that enforce honesty even in the face of side information and collusion.

Structure of the Paper

In Section 2, we set up the problem and introduce the notations used in the model. In Section 3, we overview our previous results from [14] and describe the new challenges addressed in the current manuscript. In Section 4, we develop optimal contracts when the outsourcer suspects that the agents may find out about whether the same task is sent to another agent. Subsequently in Section 5, we focus on the case where the agents potentially collude with each other.

2 Problem Set-up

This section explains the general set up of the problem and the basic assumptions behind the modelling. A summary of the notations is provided in Table 1.

The outsourcer, which we will call the *principal*, has one (normalized) unit rate of deterministic *computation tasks (jobs).* Throughout the paper, we will use the terms outsourcer and principal interchangeably. Instead of executing the tasks itself, the principal hires a set of *agents* (cloud providers) to do them. The principal has zero tolerance for incorrect computations and aims to enforce *fully honest* computation through setting a contract, involving rewards, auditing, job replication (redundancy) and fines. [1]

The principal and the agents are assumed to be rational expected utility maximizers. We further assume that the parties involved are risk-neutral, i.e., they have no strict preference between their expected utility and their utility of expected reward, and hence [15, ch.2.4], their utilities are linear functions of costs (with negative sign) and rewards (with positive sign). Moreover, we assume that agents are "lazy but not malicious", that is, they do not gain by dishonest computations other than through saving in their computation cost.[2] Generating a guessed output according to its range and distribution has zero computation cost. We assume that the range of the output is "expanded" such that a randomly guessed result is correct with only a negligible probability. Expansion can be done by requiring the intermediate steps of the computation be part of the returned output.[3] For instance, the computation task of whether a number is prime has only two possible (yes/no) outputs. But if the intermediate steps of the algorithm (e.g. the AKS primality test) must be included in the result, then guessing the correct output is practically impossible. Another consequence of the "expansion" of the output range is that two "uncoordinated" outputs are the same if and only if they are honestly computed. The "only if" part means that if the returned outputs from two independent agents are the same (in the absence of collusion), then they are correct.

[1] The no tolerance for incorrect results can be thought of as a negative infinity in the utility of the outsourcer for an accepted incorrect result.

[2] Or potentially increasing their profit, as they can produce guesses faster than honest computation, but they do not have a direct interest in sabotaging the outsourcer.

[3] A one-way hashing of the intermediate steps can be used to significantly reduce the "volume" of the data required to be transmitted without tangibly increasing the correctness probability of a guessed result.

Table 1. List of main notations

parameter	definition
α	probability of using two agents for the same computation (redundancy)
ρ	probability of auditing by the principal when only one agent is assigned
λ	probability that only one agent is assigned *and* audited, i.e., $\lambda = (1-\alpha)\rho$
β	probability of auditing by the principal if the task is assigned to two agents and the returned results are different
ν	probability of auditing by the principal if the task is assigned to two agents and the returned results are the same
Λ	maximum auditing capacity of the principal
γ	cost of an instance of auditing (incurred by the principal)
f	fine levied on an agent by the principal
F	the maximum enforceable fine
r	reward paid to the agent
R	the maximum feasible reward (instantaneous budget of the principal)
$\mathcal{C}$	the expected cost of the contract to the principal
c	cost of honest computation of a task to an agent

To efficiently enforce correctness of the results, the principal has a few options: (a) it can assign the task to one agent and randomly *audits* the returned result independently (i.e., on its own). We will simply refer to this method as *auditing*. The most straightforward method of auditing is re-computation of the task. We assume that auditing is perfect, i.e., the audit always confirms a correct output (no "false positives"), and definitely detects an incorrect output (no "false negatives"); (b) it can with some probability assign the task to multiple agents and compare the returned results (*redundancy*).[4]

The contract can involve a hybrid of auditing and redundancy. We consider the case that the redundancy scheme is done only with two agents, i.e., the same task is assigned to at most two agents at a time. Let $\alpha \in [0,1]$ be the probability that the same task is assigned to two agents. With probability $1-\alpha$, the principal employs only one of the agents selected equally likely between the two.[5] When only one agent is assigned, the principal audits the output with probability $\rho \in [0,1]$. When two agents are assigned, the principal audits the output with probability $\beta \in [0,1]$ when the returned results are different, and with probability $\nu \in [0,1]$ when the returned results are the same (ref. Table 1). The principal decides on the probability of using redundancy and

[4] The outsourcer can also with some probability compute the task itself (*partial outsourcing*). However, we showed in our prequel manuscript that it is never optimal to do partial outsourcing: it should be "all or nothing". In particular, it is best for the outsourcer to use its entire computation capacity for random auditing, except in trivial cases, e.g., when the computing capacity of the outsourcer is sufficient for computation of all of the tasks on its own *and* the cost of honest computation for the agents is higher than the cost of auditing.

[5] We formally show in [14, prop. 5.3] that equal randomization is optimal. Intuitively, this removes any information that the agents may exploit upon receiving a task.

the (conditional) probabilities of auditing, sets the value of penalty (fine) f for detected erroneous answers and reward r otherwise.

The assumption is that independent auditing of *all* of the results is either *infeasible* or *costly.* When independent auditing by the principal is through recomputation of the tasks, this assumption trivially holds, as otherwise, there is no point in outsourcing the tasks in the first place. The infeasibility of auditing all tasks is due to the limited computational capacity of the outsourcer. Specifically, let Λ be the maximum rate of auditing that the outsourcer can carry out. This issue is of particular importance, as the main reason behind outsourcing the tasks by the principal is its limited computational capacity in the first place. Moreover, auditing may also be costly since it adds a computational burden on the principal's machine and slows it down, or it will require obtaining additional hardware. For simplicity of exposition, we assume a linear relation for auditing cost and let γ be the per instance audit cost incurred by the principal.

As we argue in [14], if there is no bound on the fine that can be practically levied on an agents, then as long as there is even a tiniest probability of detection, the problem of outsourced computation could become trivial. This is because the principal then can make the expected utility of the agents negative for even the smallest likelihood of cheating by setting the fine for erroneous results arbitrarily large. However, in practice, such a fine may be extremely large, becoming an *incredible threat*, in that, if the cheating of an agent is indeed caught, the fine is practically uncollectible.[6] Thus, existence (feasibility) results of honesty enforcement that rely on choosing a "large enough" fine are rather straightforward and uninteresting. In particular, such approaches leave unanswered the question of whether honest computation is still attainable for a bounded enforceable fine below their prescriptive threshold. Moreover, such results do not provide a metric of comparison between alternative incentive schemes, or across different choice of parameters for a particular scheme for a given level of enforceable fine. In our model, we will explicitly introduce $F \geq 0$ as an exogenous parameter to represent the maximum enforceable fine and obtain the optimal contracts subject to $f \leq F$. This can be the "security deposit", prepaid by the agent to the principal, that is collectible upon a provable detection of an erroneous result. A special case of interest is $F = 0$, i.e., when the only means of punishment is refusal to pay the reward. As with the maximum enforceable fine to make it a credible threat, the maximum instantaneous "budget" of the principal leads to a bound on the reward to make it a credible promise. Let the maximum instantaneous payable reward by the principal be R. Thus, we require that $2r \leq R$. Note the multiplicand of 2: for the contract to ever consider the redundancy scheme, the principal should be able to reward both of the two agents when there is no indication of wrongdoing. This in turn implies feasibility requirement of $R \geq 2c$, since c is the cost of honest computation by an agent and hence the minimum compensation for participation.

Let $\mathcal{C}$ represent the expected cost of the principal. Due to the intolerance of the outsourcer to any erroneous results, it needs to ensure that all tasks are

[6] There can also be legal bounds on how much fine can be levied for a wrong computation, protected by liability laws.

computed honestly. Assuming honest computation of agents is established, then with the parameters introduced in the previous section, $\mathcal{C}$ is as follows:

$$\mathcal{C} = [r(1-\alpha) + 2\alpha r] + [\gamma(1-\alpha)\rho + \gamma\alpha\nu] \tag{1}$$

The first two terms refer to the expected rewards paid and the second two terms are the expected cost of auditing: with probability $1-\alpha$, the task is assigned to only one agent, hence the expected reward of $(1-\alpha)r$, and the task is audited at probability ρ, hence the expected auditing cost of $\gamma(1-\alpha)\rho$. The principal assigns two agents with probability α, hence the reward of $2r\alpha$ to be paid, and audits the returned results (that will be the same if honesty is enforced) at probability ν, which gives the last term of $\gamma\alpha\nu$. The principal's goal is to minimize the above cost provided that the contract can indeed enforce honest computation and moreover, is accepted by agent(s).

3 The Benchmark: No Side-Channel and No Collusion

In this section, we overview our previous results in [14], where we developed optimal contracts assuming (a) no *side channel* and (b) no *collusion* between the agents. The assumption of no side channel information means that none of the agents can find out whether the same task is being assigned to any other agent. In other words, the only information state that an agent has is that it has been assigned the task or not. The no collusion assumption means that not only the agents know the state of each other with respect to task assignment, they can further coordinate their reported results so that the returned results can be the same even when they are just guessed.

Recall that expansion of the range of the output (through requiring a hash of intermediate steps) implies that the two returned results are the same only if (a) they are honestly computed; or (b) the agents are colluding and coordinating to return the same guessed results. Hence, when there is no collusion, sameness of the returned results is a necessary and sufficient condition for its correctness and honest computation of the task by both agents. Hence, in this scenario, there is no gain in further auditing by the principal if (redundancy scheme is used, and) the returned results are the same. This is while, the cost of such auditing will enter the total cost of the principal. Hence, $\nu^* = 0$.

Consider two agents labelled agent 1 and 2. The choice of action for each agent is between honest computation, which we represent by $\mathscr{H}$, and cheating, which we denote by $\mathscr{C}$. Since the agents have no information about the state of the other agent, the set of their (pure) strategies and actions are the same.

The expected utility of each agent depends in part on the action of its own and of the other agent. Let $u_A(a_1, a_2)$ represent the utility of agent 1 when it chooses action a_1 and agent 2 chooses a_2, where $a_1, a_2 \in \{\mathscr{H}, \mathscr{C}\}$. The principal wants to enforce honest computation with probability one. If $u_A(\mathscr{H}, \mathscr{H}) \geq u_A(\mathscr{C}, \mathscr{H})$, then given that agent 2 is going to be computing honestly, agent 1 will prefer to do the same too, and due to symmetry, likewise for agent 2. In the game theoretic lingo, this means that $(\mathscr{H}, \mathscr{H})$ is a (Nash) equilibrium.

When only one agent is selected, the agent is rewarded r if there is no indication of wrongdoing, and is punished f if audited and caught wrong. When the redundancy scheme is selected and the returned results are equal, both agents are rewarded r. If the results are different, the principal fines both of them at f. Let λ be the probability that only one agent is assigned *and* the result is audited. Note that λ is simply $(1-\alpha)\rho$, but we introduce it for easier formulation. Also let c denote the cost of honest computation of the task, where $c > 0$. With the model so described, the expected utilities are computed as follows:[7]

$$u_A(\mathscr{H},\mathscr{H}) = r - c, \qquad u_A(\mathscr{C},\mathscr{H}) = (1-\alpha-\lambda)r/2 - (\alpha+\lambda/2)f.$$

The expression for $u_A(\mathscr{H},\mathscr{H})$ is easy to see: the agent incurs the cost of honest computation and is rewarded r for it. When the agent cheats given the other agent is honest, it receives a reward only when it is the only agent assigned *and* it is not audited. This happens with probability of $(1-\alpha-\lambda)/2$. The agent is fined otherwise: either if redundancy scheme is used (probability α) or it is the only one assigned and audited (probability $\lambda/2$), hence the expression for $u_A(\mathscr{H},\mathscr{H})$. The condition $u_A(\mathscr{H},\mathscr{H}) \geq u_A(\mathscr{C},\mathscr{H})$ therefore becomes:

$$r \geq (1+\alpha)c/(\lambda+2\alpha) - f : \qquad \textit{Incentive Compatibility Constraint.}$$

Subject to making $(\mathscr{H},\mathscr{H})$ an equilibrium, the contract is accepted if the expected utility of it to the agents is above their *reservation utility*,[8] which we assume here to be zero for simplicity:

$$r - c \geq 0 : \qquad \textit{Participation (i.e., Individual Rationality) Constraint.}$$

Then the expected cost of the contract to the principal is: $\mathcal{C} = (1+\alpha)r + \gamma\lambda$, which comes from (1) by setting $\nu = 0$. The principal chooses λ, α, f, r such that honest computation is enforced, the contract is accepted, and the expected cost of the principal is minimized. λ and α must satisfy the structural condition $0 \leq \alpha \leq 1$, $0 \leq \lambda \leq \Lambda$ and $\alpha + \lambda \leq 1$. The instantaneous budget of the principal imposes $r \leq R$ if $\alpha = 0$, and $2r \leq R$ if $\alpha > 0$. We assume $R \geq 2c$, since otherwise, as the compensation for each agent should be at least the cost of honest computation, the principal can never employ both of the agents without violating its instantaneous budget constraint, and hence, the problem would be simplified by setting $\alpha = 0$ a priori. Now, to simplify this "provisional" budget constraint, we use the following trick: we keep the less restrictive inequality, i.e.,

[7] Since the only information state to an agent is whether it receives the task, the analysis of the incentive of an agent before and after reception of the task becomes equivalent. We present the viewpoint before reception of the task for simplicity.

[8] The *reservation utility* (also referred to as the *fall-back utility* or *aspiration wage*) is the minimum utility that the agent aspires to attain or can obtain from other offers. Note that an implicit assumption here is that the agent is replaceable by any other agent with the same fall-back utility, i.e., there are many agents available with the same reservation utility. Without this assumption, the agent has negotiation power by refusing the contract knowing that it cannot be replaced.

$r \leq R$ irrespective of the value of α and solve the optimization. We will check later whether the more restrictive constraint is also met.[9] Therefore, the optimal contracts for two agents that make $(\mathscr{H}, \mathscr{H})$ a Nash equilibrium are solutions of the following optimization problem:

$$\min_{r,f,\alpha,\lambda} r(1+\alpha) + \gamma\lambda \ \text{ subject to:} \tag{2a}$$

$$r \leq R,\ f \leq F,\ 0 \leq \lambda \leq \varLambda,\ \lambda \leq 1-\alpha,\ \alpha \geq 0,\ r \geq c,\ r \geq \frac{c(1+\alpha)}{\lambda + 2\alpha} - f. \tag{2b}$$

The solution of the above (non-convex) optimization is (Proposition 5.1 in [14]):

Proposition 1. *Let $F_0 = c/\varLambda - c$ and $F_1 = c[c-\gamma]^+/[2\gamma - c]^+$,*[10] *the optimal two-agent contract that makes $(\mathscr{H}, \mathscr{H})$ a Nash equilibrium chooses $f^* = F$ and:*

$$\begin{cases} F_1 \leq F: & \alpha^* = \dfrac{c}{2F+c},\ \lambda^* = 0, r^* = c,\ \mathcal{C}^* = c(1 + \dfrac{c}{2F+c}) \\ F_0 \leq F < F_1: & \alpha^* = 0,\ \lambda^* = \dfrac{c}{c+F},\ r^* = c,\ \mathcal{C}^* = c(1+\dfrac{\gamma}{F+c}) \\ F < \min(F_0, F_1): & \alpha^* = \dfrac{c - \varLambda(c+F)}{c+2F},\ \lambda^* = \varLambda,\ r^* = c,\ \mathcal{C}^* = \dfrac{c(c+F)(2-\varLambda)}{c+2F} + \gamma\varLambda \end{cases}$$

Note that the solution always picks $r^* = c$, hence the more restrictive constraint of $2r \leq R$ for $\alpha > 0$ is also satisfied.

We have the following observation (Corollary 5.2 in [14]):
If auditing is more expensive than the cost of honest computation ($\gamma \geq c$), the optimal contract only uses the redundancy scheme. When $\gamma \leq c/2$, either there is no redundancy scheme ($\alpha = 0$) or the whole auditing capacity is used ($\lambda^* = \varLambda$). The first part of this corollary is quite intuitive: when $\gamma > c$, any instance of outsourcing to a single agent and performing independent auditing can be replaced by the redundancy scheme and strictly lower the cost by $\gamma - c$.

Note that in our optimal two-agent contract, as long as $R \geq 2c$, there is no infeasible region: for any level of enforceable fines and auditing capacity, there is always a contract that makes $(\mathscr{H}, \mathscr{H})$ a Nash equilibrium. Both incentive compatibility and the participation constraints are binding, in particular, the payment to any of the agents is never more than the cost of honest computation.

When the enforceable fine is large, the redundancy scheme is preferable. This is despite the fact that the redundancy scheme is more expensive than auditing: it costs an extra c as opposed to $\gamma < c$, however. In other words, for high values of fine, the redundancy scheme is a more effective threat against cheating than independent auditing. When F is less than F_1, the independent auditing becomes the preferred method. For lower values of F, when the auditing capacity is all used up, the redundancy scheme is added to compensate the low value of fine to maintain incentive compatibility. When $\varLambda = 0$, redundancy scheme is the only

[9] In fact, as long as $R \geq 2c$, the budget constraint is never binding since the optimal r turns out to be always c, irrespective of the other parameters (ref. Proposition 1).

[10] The notation $x^+ := \max\{0, x\}$. Also, we take $x/0 = +\infty$ for $x > 0$.

means to enforce honest computation. If furthermore no fine can be enforced ($F = 0$), then α must be one: the job must be always duplicated.

Note that the incentive compatibility constraint of $u_A(\mathscr{H}, \mathscr{H}) \geq u_A(\mathscr{C}, \mathscr{H})$ only makes $(\mathscr{C}, \mathscr{H})$ a Nash Equilibrium: each agent prefers to be honest if the other agent is honest. In particular, $(\mathscr{C}, \mathscr{C})$ is also another Nash equilibrium. If, however, $u_A(\mathscr{H}, \mathscr{H}) \geq u_A(\mathscr{C}, \mathscr{C})$, then one can argue that if the agents indeed sign up for the contract, they would rather follow the better Nash Equilibrium. This is indeed the case: $u_A(\mathscr{H}, \mathscr{H}) = r - c$ and $u_A(\mathscr{C}, \mathscr{C}) = (1 - \alpha - \lambda)r/2 - (\alpha + \lambda/2)f$, and hence the $u_A(\mathscr{H}, \mathscr{H}) \geq u_A(\mathscr{C}, \mathscr{C})$ condition becomes exactly the incentive compatibility constraint.[11]

Now consider the case that the principal is restricted to assigning only one agent. Note that when $c \leq R < 2c$, this is the only feasible option, since if the principal assigns more than one agent with any positive probability, then each agent needs to be compensated at least equal to the the cost of honest computation, c.[12] Then $\alpha^* = 0$, and the optimal contract will be obtained from the following optimization:

$$\min_{r,f,\lambda} \mathcal{C} := r + \gamma\lambda \tag{3a}$$

$$s.t. \quad r \leq R,\ 0 \leq f \leq F,\ 0 \leq \lambda \leq \mathit{\Lambda}, \tag{3b}$$

$$r \geq c,\ r\lambda + f\lambda \geq c \tag{3c}$$

The solution ([14, Prop. 4.1]) is given by setting $f^* = F$ and:

$$\begin{aligned}
&\gamma \leq \frac{c}{\mathit{\Lambda}^2}: \begin{cases} [\frac{c}{\mathit{\Lambda}} - c]^+ \leq F: & \lambda^* = \frac{c}{c+F},\ r^* = c,\ \mathcal{C}^* = c + \frac{\gamma c}{c+F} \\ [\frac{c}{\mathit{\Lambda}} - R]^+ \leq F < [\frac{c}{\mathit{\Lambda}} - c]^+: & \lambda^* = \mathit{\Lambda},\ r^* = \frac{c}{\mathit{\Lambda}} - F,\ \mathcal{C}^* = \frac{c}{\mathit{\Lambda}} + \gamma\mathit{\Lambda} - F \end{cases} \\
&\gamma > \frac{c}{\mathit{\Lambda}^2}: \begin{cases} [\sqrt{c\gamma} - c]^+ \leq F: & \lambda^* = \frac{c}{c+F},\ r^* = c,\ \mathcal{C}^* = c + \frac{\gamma c}{c+F} \\ [\sqrt{c\gamma} - R]^+ \leq F < [\sqrt{c\gamma} - c]^+: & \lambda^* = \sqrt{\frac{c}{\gamma}}, r^* = \sqrt{c\gamma} - F, \mathcal{C}^* = 2\sqrt{c\gamma} - F \\ [\frac{c}{\mathit{\Lambda}} - R]^+ \leq F < [\sqrt{c\gamma} - R]^+: & \lambda^* = \frac{c}{R+F},\ r^* = R,\ \mathcal{C}^* = R + \frac{\gamma c}{R+F} \end{cases}
\end{aligned} \tag{4}$$

For $F < [c/\mathit{\Lambda} - R]^+$, the optimization is infeasible, i.e., there is no honesty-enforcing contract that is also accepted by the agent.

Note that the optimal contracts fully utilize the maximum enforceable fine and punish at no less than F. When auditing is cheap ($\gamma \leq c/\mathit{\Lambda}^2$), increasing the auditing rate is the better option to compensate for lower values of F to maintain incentive compatibility (honest computation). This is unless the auditing rate is at its maximum $\mathit{\Lambda}$, in which case, reward must increase above c (unlike the two-agent case) to maintain incentive compatibility and compensate for the low

[11] A more strict notion of implementation is the following: if $u_A(\mathscr{H}, \mathscr{C}) \geq u_A(\mathscr{C}, \mathscr{C})$, then $(\mathscr{H}, \mathscr{H})$ will be the dominant (Nash) equilibrium, i.e., honest computation is the preferred action irrespective of the action of the other agent.

[12] Note that for $R < c$, there is no feasible contract.

value of F. Note that in this case, the participation constraint is not binding and is satisfied with a slack, while the incentive compatibility constraint is binding (satisfied tightly). For yet lower values of enforceable fine F, even maximum reward $r = R$ and auditing rate $\lambda = \Lambda$ might not impose a strong enough threat against cheating, hence the infeasibility region. When auditing is expensive ($\gamma > c/\Lambda^2$), in order to retain incentive compatibility in the situation of very low fine F, the principal should increase reward, and only consider more frequent auditing if the reward budget R has been reached.

Note that even when $c \leq R < 2c$, the infeasible region does not have to exist. Specifically, when the principal's instantaneous budget R is larger than c/Λ, then there is always a feasible contract. Then even for $F = 0$, i.e., no enforceable fine, a contract that enforces honest computing is feasible, albeit by using high values of reward and/or auditing rate. In such cases, the principal "punishes" audited erroneous computations only through not rewarding the agent. However, here, honesty cannot be enforced with zero auditing rate, and consequently, when $R < 2c$ the case of $\Lambda = 0$ leads to infeasibility.

4 Side-Channel (Information Leakage)

One of the important assumptions we made in developing our optimal hybrid contract was that the two agents do not communicate, and hence, upon receiving a task, an agent is not aware whether the same task is assigned to another agent or not. The principal uses this ambiguity in its favour to enhance the threat of auditing through redundancy. However, if agents somehow gain access to this information, the threat loses its efficacy. Specifically, if redundancy scheme is used, an agent can selectively be honest if it finds out that the task is outsourced to another agent (hence the name *side channel*), and be lax when it knows it is the only recipient of the task. If the principal supposes such "information leakage", then the contract optimization problem must be modified. Note that the agents now have two distinct information states: one in which they are the only assignee and another in which, both of them have received the task. We refer to them as *lone recipient* and *redundancy* information states, respectively. The new incentive compatibility constraint (preferring honest computation over cheating) for an agent in the lone recipient information state is:

$$r - c \geq r(1-\rho) - f\rho \Leftrightarrow r\lambda \geq c(1-\alpha) - f\lambda \tag{5}$$

For the redundancy information state, the incentive compatibility is: $r - c \geq -f$, because if the agent cheats, the results will be different and the agent will definitely be punished.[13] This constraint is redundant, because the participation constraint is still $r - c \geq 0$, which also implies $r - c \geq -f$. Here, we assume $R \geq 2c$. This will allow us to ignore the budget constraint, which is: $r \leq R$ if $\alpha = 0$, and $2r \leq R$ if $\alpha > 0$. This is because the optimal contract turns out

[13] Note that even when the agents know that the redundancy scheme is being used, unless they coordinate their reported results, guessed results will be the same only with negligible probability. We will consider the case of collusion in the next section.

to choose $r^* = c$ and hence, for $R \geq 2c$, the budget constraint is automatically satisfied. [14] Hence, the new optimization problem is the following:

$$\min_{r,f,\alpha,\lambda} r(1+\alpha) + \gamma\lambda \text{ subject to:} \tag{6a}$$

$$f \leq F,\ 0 \leq \lambda \leq \varLambda,\ \lambda \leq 1-\alpha,\ \alpha \geq 0,\ r \geq c,\ r\lambda + f\lambda \geq c(1-\alpha). \tag{6b}$$

The solution is given as the following proposition:

Proposition 2. *The optimal two-agent contract with information leakage, i.e., where the agents have access to the information of whether the same task is outsourced to the other agent or not, enforces honesty in that makes $(\mathscr{H},\mathscr{H})$ a Nash equilibrium sets $f^* = F$, $r^* = c$, and:*

$$\gamma \geq \frac{c}{\varLambda}\colon \begin{cases} F \geq [\gamma - c]^+ : & \lambda^* = \dfrac{c}{c+F},\ \alpha^* = 0,\ \mathcal{C}^* = c + \dfrac{\gamma c}{c+F} \\ F < [\gamma - c]^+ : & \lambda^* = 0,\ \alpha^* = 1,\ \mathcal{C}^* = 2c \end{cases}$$

$$\gamma < \frac{c}{\varLambda}\colon \begin{cases} F \geq [c/\varLambda - c]^+ : & \lambda^* = \dfrac{c}{c+F},\ \alpha^* = 0,\ \mathcal{C}^* = c + \dfrac{\gamma c}{c+F} \\ [\gamma - c]^+ \leq F < [c/\varLambda - c]^+ : & \lambda^* = \varLambda,\ \alpha^* = 1 - \varLambda(1+\dfrac{F}{c}),\ \mathcal{C}^* = c(2 - \varLambda(1+\dfrac{F}{c})) + \gamma\varLambda \\ F < [\gamma - c]^+ : & \lambda^* = 0,\ \alpha^* = 1,\ \mathcal{C}^* = 2c \end{cases}$$

The proof is provided in Appendix A.

Discussion Firstly, note that the cost of the above contract is clearly higher than that of the contract with no information leakage, but lower than the cost of single-agent contract. The latter is because the single-agent contract in (3) is a feasible solution of the above optimization by setting $\alpha = 0$. In fact, the cost of the contract is capped at $2c$: when both agents are hired at all times (i.e., with probability one), the incentive compatibility constraint and participation constraint are clearly satisfied with $r = c$, giving the contract cost of $2c$. This makes the redundancy scheme still an appealing option specially when the cost of auditing is high or there is little (or zero) capacity for auditing. Secondly, note that for $\gamma < c$, the redundancy scheme is never used for any value of maximum enforceable fine. When the cost of independent auditing γ is higher than the cost of honest computation c, if the enforceable fine is below the threshold of $\gamma - c$, it is best to only use the redundancy scheme, but it has to be done with certainty, i.e., $\alpha^* = 1$. Note that with information leakage, never probabilistic usage of redundancy scheme *alone* is optimal, because the agents will choose to be lazy when they *know* they are not audited. Also like the two-agent contract with no information leakage (but unlike the single-agent), the optimal reward r^* is never higher than the cost of honest computation c, irrespective of the value of the maximum enforceable fine. Moreover, we observe that for large values of enforceable fine, in contrast to the no information leakage case, it is the independent auditing mechanism that is now the preferred method.

[14] Recall that for $R < 2c$, the outsourcer can never assign more than one agent with any positive probability, and hence, the issue of information leakage is irrelevant.

5 Colluding Agents

Suppose the agents not only know the state of the other agents with respect to the task assignment, but they can also coordinate their response to report the same guessed result. This can save them from the cost of honest computation and at the same time, go undetected. Hence, unlike before, returned results from multiple agents that are the same may not be correct. The principal can audit the returned results (through re-computation of the task) when they are the same. Consider two agents. As in the information leakage setting, each agent has two distinct information state: being the sole recipient, and being one of the two recipients. The difference is that in the second information state, the set of (pure) actions for the agents is computing honestly, denoted by $\mathscr{H}$ as before, and colluding with the other agent, which we represent by $\underline{\mathscr{C}}$ (to differentiate it from just cheating, which we denoted by $\mathscr{C}$).

Note that similar to the case of information leakage, the principal still has to enforce honesty in the information state of an agent in which it knows it is the lone recipient of the task. This implies (ref. (5)):

$$r \geq \frac{c}{\rho} - f \tag{7}$$

One way to dissuade the agents from colluding is to make collusion a less attractive equilibrium than honesty. For non-colluding agents with no information leakage, this meant the same as the incentive compatibility constraint, because there, $u_A(\mathscr{C}, \mathscr{H}) = u_A(\mathscr{C}, \mathscr{C})$. For non-colluding agents with information leakage in the information state of the redundancy scheme, this translated to $r - c \geq -f$, which is trivially satisfied following the participation constraint, i.e., $r - c \geq 0$. Here, though, this constraint is not automatically satisfied, since the redundancy scheme "alone" is fundamentally susceptible to collusion as coordinated guessed results will be indistinguishable from honestly computed ones. Hence – unless the tasks are sufficiently obfuscated so that the colluding agents cannot (economically) tell whether they have received the same task or not – the principal must add another threat: the returned results from the two agents are audited by the principal with probability ν, (even) when they are the same. Note that the value of ν enters the cost of the principal even if honest computation is indeed enforced. This is because when honest computation is enforced, the returned results are the same too. Specifically, there will be an additional term of $\gamma\alpha\nu$.

With the introduction of ν, we have: $u_A(\underline{\mathscr{C}}, \underline{\mathscr{C}}) = r(1-\nu) - F\nu$. Therefore, to make honesty a more attractive equilibrium than collusion, in the redundancy scheme information state, we must have:

$$r - c \geq r(1-\nu) - F\nu \Leftrightarrow r \geq \frac{c}{\nu} - f \tag{8}$$

Hence, the corresponding optimal contract is given by the following optimization:

$$\min_{r,f,\alpha,\lambda,\nu} r(1+\alpha) + \gamma\lambda + \gamma\alpha\nu, \quad \text{s. t.} : r \leq R,\ f \leq F,\ 0 \leq \rho \leq 1,\ 0 \leq \nu \leq 1 \tag{9a}$$

$$\text{and: } \rho(1-\alpha)+\alpha\nu \leq \mathit{\Lambda},\ \alpha \geq 0,\ r \geq c,\ r \geq \frac{c}{\rho} - f,\ r \geq \frac{c}{\nu} - f. \tag{9b}$$

We have the following proposition:

Proposition 3. *The optimal contract that enforces honesty in lone information state and makes collusion a less attractive equilibrium than honest computation in the redundancy information state sets $\alpha^* = 0$, i.e., never uses the redundancy scheme at all. The rest of the parameters of the contract are also according to the optimal contract for a single agent provided in* (4).

The result can be derived directly by examining the KKT conditions of the optimization problem in (9) after establishing that KKT conditions are indeed applicable. However, we provide a simpler proof that delivers more intuition.

Proof. Consider a claimed optimal contract that selects an $\alpha > 0$. We will construct an alternative feasible contract that employs only one agent ($\alpha_{\text{alt}} = 0$) and strictly improves the cost, hence reaching a contradiction.

The claimed contract has to satisfy inequalities (7) and (8) to be feasible. Now consider an alternative contract alt that only selects one agent and audits it with probability $\lambda_{\text{alt}} = \rho(1-\alpha) + \alpha\nu$. The values of the reward and fine are the same. Fist we examine the change in the contract cost: $C_{\text{alt}} - C = [r + \gamma(\rho(1-\alpha) + \nu\alpha)] - [r(1+\alpha) + \gamma\rho(1-\alpha) + \gamma\nu\alpha] = \alpha r$, which is strictly positive based on the assumption that $\alpha > 0$. Now if we show that this alternative contract is feasible, then we have reached the contradiction we are after.

The only non-trivial constraint that we need to verify to establish the feasibility of the alternative contract is the incentive compatibility: we must have:

$$r - c \geq r(1-\lambda_{\text{alt}}) - f\lambda_{\text{alt}} \Rightarrow r \geq \frac{c}{\lambda_{\text{alt}}} - f \quad \text{that is: } \ r \geq \frac{c}{\rho(1-\alpha)+\nu\alpha} - f$$

The last inequality can be inferred from (7) and (8) and the fact that for any $\alpha \geq 0$, we have: $\rho(1-\alpha) + \nu\alpha \leq \min(\rho, \nu)$. This completes the proof. □

Intuitively, whenever the two agents are to be assigned, the principal can save the reward to the second agent by assigning the task to only one of them. The principal will audit the only agent as it would have audited the two agents. This works since two colluding agents (so far) act as though a single agent anyway.

The above proposition is a negative result: the benefits of redundancy scheme seem to be all lost if the principal suspects collusion between the agents. However, in what follows, we introduce two schemes based on the idea of offering "bounties" that, at least partially, save the redundancy scheme. These bounty schemes better utilize the incentive of the agents against each other, creating a prisoner's dilemma-like situation to undermine collusion. That is, instead of trying to make collusion a less attractive equilibrium, which we observed is futile in Proposition 3, these schemes make collusion a non-equilibrium. The idea is as follows: when the returned results are different, the principal can randomly audit the task and reward the bounty in such cases to the agent with the correct result (if any). The value of the bounty should be the largest credible promise, i.e., R. The difference between the two schemes is how they treat the unaudited cases when the returned results are different: in *bounty scheme one*, when the results are different and auditing does not occur, both agents are punished at f. In contrast, in *bounty scheme two*, when the returned results are different and the task is not audited, both agents are rewarded at r. A nice feature of both schemes is that, if they indeed succeed to enforce honesty, the bounties will never in fact be paid: All that is necessary is the credible promise of the bounties. In what follows we analyse these two schemes.

For both schemes, we have:

$$u(\underline{\mathscr{C}},\underline{\mathscr{C}}) = r(1-\nu) - f\nu, \qquad\qquad u(\mathscr{H},\mathscr{H}) = r - c$$

In bounty scheme one:

$$u(\mathscr{H},\underline{\mathscr{C}}) = -c + R\beta - f(1-\beta) \qquad\qquad u(\underline{\mathscr{C}},\mathscr{H}) = -f$$

In bounty scheme two:

$$u(\mathscr{H},\underline{\mathscr{C}}) = -c + R\beta + r(1-\beta), \qquad u(\underline{\mathscr{C}},\mathscr{H}) = r(1-\beta) - f\beta$$

Making $(\mathscr{H},\mathscr{H})$ an equilibrium is automatic in scheme one: $r - c \geq -f$ for any $f \geq 0$, following the participation constraint $r - c \geq 0$. Making $(\underline{\mathscr{C}},\underline{\mathscr{C}})$ a non-equilibrium requires the following:

$$-c + R\beta - f(1-\beta) \geq r(1-\nu) - f\nu \tag{10}$$

Similarly, making $(\underline{\mathscr{C}},\underline{\mathscr{C}})$ a non-equilibrium for scheme two requires:

$$-c + R\beta + r(1-\beta) \geq r(1-\nu) - f\nu. \tag{11}$$

Moreover, to have $(\mathscr{H},\mathscr{H})$ an equilibrium in scheme two, one must ensure:

$$r - c \geq r(1-\beta) - f\beta \tag{12}$$

In both cases, the value of β does not directly enter the cost of the contract to the principal if honesty is indeed enforced. Hence, the principal can choose the maximum possible value. In order to make it credible, the principal must have enough auditing capacity. Specifically, $\lambda + \alpha\beta \leq \varLambda$. Hence the maximum value of β is given as $(\varLambda - \lambda)/\alpha$. Replacing in (10) we obtain the following extra (incentive compatibility) constraint for scheme one:

$$(R+f)(\varLambda - \lambda) + (r+f)\alpha\nu - (c+r+f)\alpha \geq 0. \tag{13}$$

Similarly, replacing $\beta = (\varLambda - \lambda)/\alpha$ in (11) and (12) yields the following for scheme two:

$$(r+f)(\varLambda - \lambda) - \alpha c \geq 0, \qquad (R-r)(\varLambda - \lambda) + \nu(r+f)\alpha - c\alpha \geq 0 \tag{14}$$

Hence, the contract optimization for bounty schemes one and two are the same as in (9) except that the last constraint in (9b), i.e., $r \geq c/\nu - f$, is replaced with (13) and (14), respectively. It turns out, however, that these two innocuous-looking optimization problems do not lend easily to closed-form solutions.

In what follows we obtain partial solutions of these optimizations, which provide insight on the applicability of redundancy scheme in the presence of collusion. First, note that for any given set of parameters c, F, R, $\varLambda$, the best contract for the information leakage setting yields a better cost than any feasible contract in the collusion scenario (both bounty schemes). This is because, compared to (6), the optimization problem of finding the best contract for bounty schemes one and two each have: (A) an additional non-negative term in the cost: $\gamma\alpha\nu$; and (B) an *extra* incentive compatibility constraint, (13) in scheme one and (14) in scheme two. Therefore, in particular, if a solution of the optimization in (6) is a feasible solution for the optimization of schemes one and two, then it is also optimal for them as well. This happens for example when the optimal information leakage contract chooses $\alpha = 0$, as then, the extra incentive compatibility constraint in (13) and (14) are trivially satisfied. Hence, in the light of Proposition 2, we have the following result:

Corollary 1. *For both schemes one and two, the optimal contract chooses $\alpha^* = 0$ for $F \geq [\max(\gamma, c/\varLambda) - c]^+$. The rest of the parameters for such cases are $f^* = F$, $r^* = c$ and $\lambda^* = c/(c + F)$.*

The corollary shows that for large values of the enforceable fine F, assigning a single agent is the preferred method of outsourcing. However, the corollary leaves out the question of whether redundancy scheme is ever the preferred method in the presence of collusion with the introduction of the bounty schemes. The next result provides a positive answer. In particular, we derive sufficient conditions under which, the redundancy scheme is the preferred method even in the presence of collusion:

Corollary 2. *In bounty scheme two, for $F < [\gamma - c]^+$, if $\varLambda \geq c/\min(c + F, R - c)$, the optimal contract chooses redundancy $\alpha^* = 1$. The rest of the parameters for such a case are: $r^* = c$, $\lambda^* = \nu^* = 0$, $f^* = F$.*

The corollary is intuitive: the auditing capacity should be large enough to make the promise of checking for bounty when the results are different a credible one.

Proof. The corollary follows from a similar logic as in the previous corollary: we will find cases that the optimal solution of the information leakage contract optimization in (6) are feasible solutions of the optimization problem for scheme two. An alternative to $\alpha = 0$ is $\alpha = 1$: if $\nu = 0$ is a feasible choice for scheme two with the parameters that make $\alpha = 1$ an optimal solution for the information leakage setting, then the corresponding contract is optimal for scheme two. From Proposition 2, $\alpha^* = 1$ when $F < [\gamma - c]^+$. The rest of the parameters are $\lambda^* = 0$, $r^* = c$ and $f^* = F$. We should investigate whether these parameters and $\nu = 0$ satisfy (14), which becomes: $(c + F)\varLambda \geq c$ & $(R - c)\varLambda \geq c \Leftrightarrow \varLambda \geq c/\min(c + F, R - c)$, hence the corollary. □

The following corollary provides a sufficient condition for scheme one to use the redundancy scheme.

Corollary 3. *In bounty scheme one, for $F < [\gamma - c]^+$, if $\varLambda \geq 2c/R$, the optimal contract chooses redundancy $\alpha^* = 1$. The rest of the parameters for such a case are: $r^* = c$, $\lambda^* = \nu^* = 0$, and notably $f^* = 0$.*

Proof. The proof is similar to that of Corollary 2, with the following exception: Note from (13) that f plays a double edge sword role, and it is no more a priori clear that maximum fine is the best option. In fact for this case it turns out to be exactly the opposite. From Proposition 2, for $F \leq [\gamma - c]^+$, optimal contract is given by $r^* = c$, $\lambda^* = 0$, $\alpha* = 1$ and $f^* = F$. However, the value of $f \geq 0$ for this region does not affect the cost and feasibility of the contract, and hence, any $f \geq 0$ is in fact also optimal. Replacing these parameters with a general f and along with $\nu = 0$ in (13), we obtain: $(R + f)\varLambda \geq 2c + f$. Hence a sufficient condition for feasibility (and hence optimality) is $(R + f)\varLambda \geq 2c + f$. The value of f is arbitrarily, the best result is obtained for $f = 0$, that is $\varLambda \geq 2c/R$. □

5.1 Generalizing the two bounty schemes

In this subsection, we propose a way to unite (and strengthen) the two previous bounty schemes. The distinction between the two schemes were in treatment of un-matching returned results from the two agents that are not audited by the principal: scheme one penalizes the two at f, while scheme two rewards them both at r. A new approach that can capture both of these schemes as special cases is to define a new variable x to represent the amount that is "paid" to each one of the two agents when the returned results are different and are not audited by the principal. The maximum enforceable fine and instantaneous budget constraints must be applied to this new variable: $-F \leq x \leq R/2$. Note that $x = -f$ and $x = r$ retrieves the previous schemes respectively.

With this new variable defined, the agent utilities are modified as follows:

$$u(\mathscr{H}, \underline{\mathscr{C}}) = -c + R\beta + x(1 - \beta) \qquad u(\underline{\mathscr{C}}, \mathscr{H}) = -f\beta + x(1 - \beta)$$

Ensuring $(\underline{\mathscr{C}}, \underline{\mathscr{C}})$ is a non-equilibrium, replacing $\beta = (\varLambda - \lambda)/\alpha$, translates to:

$$(R - x)(\varLambda - \lambda) + (r + f)\alpha\nu - (r + c - x)\alpha \geq 0.$$

which captures both (10) and (11) as special cases. Moreover, to ensure that $(\mathscr{H}, \mathscr{H})$ is an equilibrium, we must have (with $\beta = (\varLambda - \lambda)/\alpha$ replaced):

$$(x + f)(\varLambda - \lambda) - (x + c - r)\alpha \geq 0.$$

Note that Corollary 1 still holds as before. Introduction of the new variable x allows us to generalize the results of Corollaries 2 and 3 as follows:

Corollary 4. *In the generalized bounty scheme, for $F < [\gamma - c]^+$, if $\varLambda \geq \max\{2c/(R + F), (4c - R)/R\}$, the optimal contract chooses redundancy $\alpha^* = 1$. The rest of the parameters for such a case are: $r^* = c$, $\lambda^* = \nu^* = 0$, and notably $f^* = F$ and $x^* = \min\{2cF/(R + F - 2c), R/2\}$.*

The proof is similar to those of Corollaries 2 and 3, and is omitted for brevity. Note that unlike bounty scheme one, the penalty is always at the maximum value and the two agents are paid a positive amount for those results that are returned different but are not audited by the principal. The condition given for Λ is looser (and hence better) than either one of Corollaries 2 and 3.

Corollaries 2, 3 and 4 show that the bounty scheme can make the redundancy scheme the preferred method, even in the face of colluding agents, specially when the value of the enforceable fines are low, and the cost of auditing is high.

6 Conclusion

In this paper, we designed and analysed incentive schemes that an outsourcer of computation tasks can utilize to enforce honest computation and participation of the agents. The focus of the paper was on the effect of side information and collusion on the optimal contracts involving a hybrid of direct computation and redundancy scheme (duplication of the same task to two agents and comparing the returned results). In particular, we explicitly developed conditions in which the redundancy scheme fails to be the preferred method (Propositions 2 and 3) and conditions in which it will be the preferred method (Proposition 2 and Corollaries 2, 3 and 4) even under such adverse conditions. Notably, we showed that that making collusion a less attractive equilibrium is *not* an effective way at all to save the redundancy scheme in the face of collusion (Proposition 3). Instead, an effective way is bounty-like schemes that attempt to make collusion a dis-equilibrium (Corollaries 2, 3 and 4). Overall, we noted that preference for redundancy in the presence of side information or collusion occurs for high values of auditing cost (expected), and low values of maximum enforceable fines, where the latter is in sharp contrast with the cases that collusion or side information is absent. This work in part provided insights on potentials and limitations of redundancy scheme as a method of auditing.

Our work opens a number of potential avenues for future investigation. One of the major scenarios which we have simplified in this paper is the possible interactions among the agents. Here we assumed that agents share accurate information about their state with respect to the job assignments to each others, and then each individually and independently decides its action. However, the agents may be able to deceive their peers by giving them wrong signals about their state with the objective of winning the bounty. Also, we assumed cheating agents, although able to collude, cannot have a means of commitment among themselves. If enforceable commitments among colluding agents are assumed, the analysis can become more complicated: the agents may agree to pass the honest result to one another, or intentionally plan for one of them to get the bounty, only to share it among themselves later. In addition, we have not considered global optimality, i.e., optimality among all possible contract designs. In [14], we established that when agents are non-colluding and non-communicating, the optimal contracts developed assuming at most two agents per each task are in fact globally optimal among all contracts involving any number of agents per task. In the presence of information leakage and collusion, this becomes more challenging, as more parameters (e.g., the bounty) can be involved to build contracts. We leave this investigation for future research.

A Appendix: Proof of Proposition 2

We first argue that we can safely assume that the fine is at its maximum value, i.e., $f^* = F$: the principal can manipulate r, λ, α or f in order to enforce the incentive compatibility constraint in the lone recipient information state. Among these variables, only increasing the fine is costless to the principal. Moreover, the only two constraints that f appears in is $f \leq F$ and the incentive compatibility constraint. Hence, any optimal contract can be transformed to one in which $f = F$, keeping all other parameters fixed. We use the Karush-Kuhn-Tucker (KKT) conditions [16] to solve the above non-linear (non-convex) programming. The non-convexity arises due to the incentive compatibility constraint (the last constraint in (6b)). Note that our cost and constraint functions are all continuously differentiable. We first use the Mangasarian–Fromovitz constraint qualification (MFCQ) to establish that any minimum must satisfy the KKT conditions, i.e., KKT are necessary conditions of optimality. In the absence of equality constraints, the MFCQ condition means that the gradients of the active inequality constraints are positive-linearly independent at optimum points.

As we mentioned before, we assume $R \geq 2c$, since otherwise, never more than one agent can be hired. It will turn out that the optimal contract will always choose $r^* = c$, hence the budget constraint of $r \leq R$ for $\alpha = 0$ and $r \leq 2R$ for $\alpha > 0$ is automatically satisfied. The remaining inequalities (written in standard form) are $-\lambda \leq 0$, $\lambda - \Lambda \leq 0$, $\lambda + \alpha - 1 \leq 0$, $-\alpha \leq 0$, $c - r \leq 0$, $c(1 - \alpha) - F\lambda - r\lambda \leq 0$. The gradients of these inequality constraints with the order of variables as (λ, α, r) are: $(-1, 0, 0)$, $(1, 0, 0)$, $(1, 1, 0)$, $(0, -1, 0)$, $(0, 0, -1)$ and $(-F - r, -c, -\lambda)$. We will consider the cases of $\alpha = 1$ and $\alpha < 1$ separately.

If $\alpha = 1$, we must have $\lambda = 0$, which means the only possible active inequalities are $-\lambda \leq 0$, $\lambda + \alpha \leq 1$ and $c - r \leq 0$, with gradients $(-1, 0, 0)$, $(1, 1, 0)$ and $(0, 0, -1)$. These gradients are clearly linearly independent and the MFCQ condition holds. If $\alpha < 1$, from the last constraint, we must have $\lambda > 0$. Now consider two cases: $\Lambda = 1$ or $\Lambda < 1$. When $\Lambda = 1$, then the constraint of $\lambda \leq \Lambda$ is implied by $\lambda + \alpha \leq 1$ and $\alpha \geq 0$, and can be removed. Now, if $\alpha = 0$, then the last inequality $c(1 - \alpha) - F\lambda - r\lambda$ is implied by $c - r \geq 0$ and hence can be removed. The standing inequalities will therefore be $\lambda + \alpha \leq 1$, $c - r \leq 0$, along with the active inequality of $-\alpha \leq 0$. The gradients of these inequalities are respectively $(1, 1, 0)$, $(0, 0, -1)$ and $(0, -1, 0)$, and are clearly linearly independent. If, on the other hand, $0 < \alpha < 1$, then the standing constraints are: $\lambda + \alpha - 1 \leq 0$, $c - r \leq 0$ and $c(1 - \alpha) - F\lambda - r\lambda \leq 0$, with the gradients: $(1, 1, 0)$, $(0, 0, -1)$ and $(-F - r, -c, -\lambda)$. The last three vectors are linearly independent because all the elements of the last vector are non-zero given $\lambda > 0$. When $\Lambda < 1$, first suppose $\alpha = 0$. Then, the constraint of $\lambda + \alpha \leq 1$ is inferred from $\lambda \leq \Lambda$, and hence can be removed. The standing constraints are $\lambda - \Lambda \leq 0$, $c - r \leq 0$, $c(1-\alpha) - F\lambda - r\lambda \leq 0$, along with the active constraint of $-\alpha < 0$. The gradients of these constraints are $(1, 0, 0)$, $(0, 0, -1)$, $(-F - r, -c, -\lambda)$ and $(0, -1, 0)$. Note that except for the singleton point of $F = c(1/\Lambda - 1)$, never all four of these constraints are active.[15] Now note that any three of the gradients are linearly independent given $\lambda > 0$. Finally, when $\Lambda < 1$ and $0 < \alpha < 1$, the standing constraints are $\lambda - \Lambda \leq 0$, $\lambda + \alpha - 1 \leq 0$, $c - r \leq 0$ and $c(1+\alpha) - F\lambda - r\lambda \leq 0$, with gradients $(1, 0, 0)$, $(1, 1, 0)$, $(0, 0, -1)$ and $(-F - r, -c, -\lambda)$, respectively. As before, the only case that all four of these constraints can be active is the single point of $F = c - r$. For all other values of F, at most three of these constraints are active, whose gradients are linearly independent given $\lambda > 0$. In summary, the MFCQ normality condition holds.

To systematically obtain the KKT conditions, we introduce the dual multipliers μ_1 to μ_6, and transform the problem in (6) as follows:

$$\begin{aligned}\min_{r,\alpha,\lambda,\mu_i} \bar{\mathcal{C}} = r(1+\alpha) + \gamma\lambda - \mu_1\lambda + \mu_2(\lambda - \Lambda) + \mu_3(\lambda + \alpha - 1) \\ - \mu_4\alpha + \mu_5(c - r) + \mu_6\left(c(1-\alpha) - F\lambda - r\lambda\right) \quad (15)\end{aligned}$$

subject to:

$$\begin{aligned}\text{primal feasibility: } & 0 \leq \lambda \leq \Lambda,\ \lambda \leq 1 - \alpha, \\ & \alpha \geq 0,\ r \geq c,\ r\lambda + F\lambda \geq c(1-\alpha) && (16a) \\ \text{dual feasibility: } & \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6 \geq 0, && (16b) \\ \text{complementary slackness: } & \mu_1\lambda = 0,\ \mu_2(\lambda - \Lambda) = 0, && (16c) \\ & \mu_3(\lambda + \alpha - 1) = 0, \mu_4\alpha = 0, && (16d) \\ & \mu_5(c - r) = 0, \mu_6\left(c(1-\alpha) - F\lambda - r\lambda\right) = 0. && (16e)\end{aligned}$$

The first order conditions of optimality are:

$$\frac{\partial\bar{\mathcal{C}}}{\partial\lambda} = 0 \Leftrightarrow \gamma - \mu_1 + \mu_2 + \mu_3 - \mu_6(F + r) = 0, \quad (17)$$

$$\frac{\partial\bar{\mathcal{C}}}{\partial\alpha} = 0 \Leftrightarrow r + \mu_3 - \mu_4 - c\mu_6 = 0, \quad (18)$$

$$\frac{\partial\bar{\mathcal{C}}}{\partial r} = 0 \Leftrightarrow (1 + \alpha) - \mu_5 - \lambda\mu_6 = 0. \quad (19)$$

The full solution is now derived as in the statement of the proposition by straightforward investigation of the conditions (16) through (19). □

References

1. Fullbright, N.R.: Outsourcing in a brave new world: An international survey of current outsourcing practice and trends. Technical report (2011)
2. Online: Cloud Computing Price Comparison Engine. `http://www.cloudorado.com/` Accessed: 2014-05-11.
3. Gennaro, R., Gentry, C., Parno, B.: Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In: CRYPTO. Springer (2010)

[15] The optimal contract for such a point can be derived using a continuity argument.

4. Setty, S., McPherson, R., Blumberg, A.J., Walfish, M.: Making argument systems for outsourced computation practical (sometimes). In: NDSS. (2012)
5. Setty, S., Vu, V., Panpalia, N., Braun, B., Blumberg, A.J., Walfish, M.: Taking proof-based verified computation a few steps closer to practicality. In: USENIX Security. (2012)
6. Belenkiy, M., Chase, M., Erway, C.C., Jannotti, J., Küpçü, A., Lysyanskaya, A.: Incentivizing outsourced computation. In: NetEcon, ACM (2008)
7. Nix, R., Kantarcioglu, M.: Contractual agreement design for enforcing honesty in cloud outsourcing. In: GameSec. Springer (2012)
8. Wang, C., Ren, K., Wang, J.: Secure and practical outsourcing of linear programming in cloud computing. In: INFOCOM, 2011. (2011)
9. Atallah, M.J., Cho, Y., Kundu, A.: Efficient data authentication in an environment of untrusted third-party distributors. In: IEEE ICDE. (2008)
10. Chen, H., Ma, X., Hsu, W., Li, N., Wang, Q.: Access control friendly query verification for outsourced data publishing. In: ESORICS. (2008)
11. Yi, K., Li, F., Cormode, G., Hadjieleftheriou, M., Kollios, G., Srivastava, D.: Small synopses for group-by query verification on outsourced data streams. ACM TODS (2009)
12. Monrose, F., Wyckoff, P., Rubin, A.D.: Distributed execution with remote audit. In: NDSS. (1999)
13. Canetti, R., Riva, B., Rothblum, G.N.: Practical delegation of computation using multiple servers. In: ACM CCS. (2011)
14. Pham, V., Khouzani, M., Cid, C.: Optimal contracts for outsourced computation. In: online manuscript. (2014) `http://www.isg.rhul.ac.uk/~ccid/publications/outsourced_computation-contracts.pdf`.
15. Gintis, H.: Game Theory Evolving: A Problem-Centered Introduction to Modeling Strategic Interaction. Princeton University Press (2009)
16. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear programming: theory and algorithms. John Wiley & Sons (2013)