

Manipulating Scrip Systems: Sybils and Collusion

Ian A. Kash¹, Eric J. Friedman², and Joseph Y. Halpern¹

¹ Computer Science Dept., Cornell University
{kash, halpern}@cs.cornell.edu

² Sch. of Oper. Res. and Inf. Eng., Cornell University
ejf27@cornell.edu

Abstract. Game-theoretic analyses of distributed and peer-to-peer systems typically use the Nash equilibrium solution concept, but this explicitly excludes the possibility of strategic behavior involving more than one agent. We examine the effects of two types of strategic behavior involving more than one agent, sybils and collusion, in the context of scrip systems where agents provide each other with service in exchange for scrip. Sybils make an agent more likely to be chosen to provide service, which generally makes it harder for agents without sybils to earn money and decreases social welfare. Surprisingly, in certain circumstances it is possible for sybils to make all agents better off. While collusion is generally bad, in the context of scrip systems it actually tends to make all agents better off, not merely those who collude. These results also provide insight into the effects of allowing agents to advertise and loan money.

1 Introduction

Studies of filesharing networks have shown that more than half of participants share no files [1,11]. Creating a currency with which users can get paid for the service they provide gives users an incentive to contribute. Not surprisingly, scrip systems have often been proposed to prevent such free-riding, as well as to address resource-allocation problems more broadly. For example, KARMA used scrip to prevent free riding in P2P networks [17] and Mirage [4] and Egg [3] use scrip to allocate resources in a wireless sensor network testbed and a grid respectively.

Chun et al. [16] studied user behavior in a deployed scrip system and observed that users tried various (rational) manipulations of the auction mechanism used by the system. Their observations suggest that system designers will have to deal with game-theoretic concerns. Game-theoretic analyses of scrip systems (e.g., [2,8,10,14]) have focused on Nash equilibrium. However, because Nash equilibrium explicitly excludes strategic behavior involving more than one agent, it cannot deal with many of the concerns of systems designers. One obvious concern is collusion among sets of agents, but there are more subtle concerns. In a P2P network, it is typically easy for an agent to join the system under a number of different identities, and then mount what has been called a *sybil attack* [7]. While these concerns are by now well understood, their impact on a Nash equilibrium is not, although there has been some work on the effects of multiple identities in auctions [19]. In this paper we examine the effects of sybils and collusion on scrip systems. We show that if such strategic behavior is not taken into

account, the performance of the system can be significantly degraded; indeed the scrip system can fail in such a way that all agents even stop providing service entirely. Perhaps more surprisingly, there are circumstances where sybils and collusion can improve social welfare. Understanding the circumstances that lead to these different outcomes is essential to the design of stable and efficient scrip systems.

In scrip systems where each new user is given an initial amount of scrip, there is an obvious benefit to creating sybils. Even if this incentive is removed, sybils are still useful: they can be used to increase the likelihood that an agent will be asked to provide service, which makes it easier for him to earn money. This means that, in equilibrium, those agents who have sybils will tend to spend less time without money and those who do not will tend to spend more time without money, relative to the distribution of money if no one had sybils. This increases the utility of sybiling agents at the expense of non-sybiling agents. The overall effect is such that, if a large fraction of the agents have sybils (even if each has only a few), agents without sybils typically will do poorly. From the perspective of an agent considering creating sybils, the first few sybils can provide him with a significant benefit, but the benefits of additional sybils rapidly diminish. So if a designer can make sybiling moderately costly, the number of sybils actually created by rational agents will usually be relatively small.

If a small fraction of agents have sybils, the situation is more subtle. Agents with sybils still do better than those without, but the situation is not zero-sum. In particular, changes in the distribution of money can actually lead to a greater total number of opportunities to earn money. This, in turn, can result in an increase in social welfare: everyone is better off. However, exploiting this fact is generally not desirable. The same process that leads to an improvement in social welfare can also lead to a crash of the system, where all agents stop providing service. The system designer can achieve the same effects by increasing the average amount of money or biasing the volunteer selection process, so exploiting the possibility of sybils is generally not desirable.

Sybils create their effects by increasing the likelihood that an agent will be asked to provide service; our analysis of them does not depend on why this increase occurs. Thus, our analysis also applies to other factors that increase demand for an agent's services, such as advertising. In particular, our results suggest that there are tradeoffs involved in allowing advertising. For example, many systems allow agents to announce their connection speed and other similar factors. If this biases requests towards agents with high connection speeds even when agents with lower connection speeds are perfectly capable of satisfying a particular request, then agents with low connection speeds will have a significantly worse experience in the system. This also means that such agents will have a strong incentive to lie about their connection speed.

While collusion is generally a bad thing, in the context of scrip systems with fixed prices, it is almost entirely positive. Without collusion, if a user runs out of money he is unable to request service until he is able to earn some. However, a colluding group can pool their money so that all members can make a request whenever the group as a whole has some money. This increases welfare for the agents who collude because agents who have no money receive no service.

Furthermore, collusion tends to benefit the non-colluding agents as well. Since colluding agents work less often, it is easier for everyone to earn money, which ends up

making everyone better off. However, as with sybils, collusion does have the potential of crashing the system if the average amount of money is high.

While a designer should generally encourage collusion, we would expect that in most systems there will be relatively little collusion and what collusion exists will involve small numbers of agents. After all, scrip systems exist to try and resolve resource-allocation problems where agents are competing with each other. If they could collude to optimally allocate resources within the group, they would not need a scrip system in the first place. However, many of the benefits of collusion come from agents being allowed to effectively have a negative amount of money (by borrowing from their collusive partners). These benefits could also be realized if agents are allowed to borrow money, so designing a loan mechanism could be an important improvement for a scrip system. Of course, implementing such a loan mechanism in a way that prevents abuse requires a careful design.

2 Model

Our model of a scrip system is essentially that of [14]. There are n agents in the system. One agent can request a service which another agent can volunteer to fulfill. When a service is performed by agent j for agent i , agent i derives some utility from having that service performed, while agent j loses some utility for performing it. The amount of utility gained by having a service performed and the amount lost by performing it may depend on the agent. We assume that agents have a *type* t drawn from some finite set T of types. We can describe the entire system using the tuple (T, \mathbf{f}, n, m) , where f_t is the fraction with type t , n is the total number of agents, and m is the average amount of money per agent. In this paper, we consider only *standard agents*, whose type we can characterize by a tuple $t = (\alpha_t, \beta_t, \gamma_t, \delta_t, \rho_t, \chi_t)$, where

- α_t reflects the cost of satisfying a request;
- β_t is the probability that the agent can satisfy a request
- γ_t measures the utility an agent gains for having a request satisfied;
- δ_t is the rate at which the agents discounts utility (so a unit of utility in k steps is worth only $\delta_t^{k/n}$ as much as a unit of utility now)—intuitively, δ_t is a measure of an agent’s patience (the larger δ_t the more patient an agent is, since a unit of utility tomorrow is worth almost as much as a unit today);
- ρ_t represents the (relative) request rate (since not all agents make requests at the same rate)—intuitively, ρ_t characterizes an agent’s need for service; and
- χ_t represents the relative likelihood of being chosen to satisfy a request. This might be because, for example, agents with better connection speeds are preferred.

The parameter χ_t did not appear in [14]; otherwise the definition of a type is identical to that of [14].

We model the system as running for an infinite number of rounds. In each round, an agent is picked with probability proportional to ρ_t to request service: a particular agent of type t is chosen with probability $\rho_t / \sum_{t'} f_{t'} n \rho_{t'}$. Receiving service costs some amount of scrip that we normalize to \$1. If the chosen agent does not have enough scrip, nothing will happen in this round. Otherwise, each agent of type t is able to satisfy this

request with probability β_t , independent of previous behavior. If at least one agent is able and willing to satisfy the request, and the requester has type t' , then the requester gets a benefit of $\gamma_{t'}$ utils (the job is done) and one of the volunteers is chosen at random (weighted by the χ_t) to fulfill the request. If the chosen volunteer has type t , then that agent loses α_t utils, and receives a dollar as payment. The utility of all other agents is unchanged in that round. The total utility of an agent is the discounted sum of the agent's round utilities. To model the fact that requests will happen more frequently the more agents there are, we assume that the time between rounds is $1/n$. This captures the intuition that things are really happening in parallel and that adding more agents should not change an agent's request rate.

In our previous work [8,14], we proved a number of results about this model:

- There is an (ϵ -) Nash equilibrium where each agent chooses a threshold k and volunteers to work only when he has less than k dollars. For this equilibrium, an agent needs no knowledge about other agents, as long as he knows how often he will make a request and how often he will be chosen to work (both of which he can determine empirically).
- Social welfare is essentially proportional to the average number of agents who have money, which in turn is determined by the types of agents and the average amount of money the agents in the system have.
- Social welfare increases as the average amount of money increases, up to a certain point. Beyond that point, the system “crashes”: the only equilibrium is the trivial equilibrium where all agents have threshold 0.

Our proofs of these results relied on the assumption that all types of agents shared common values of β , χ , and ρ . To model sybils and collusion, we need to remove this assumption. The purpose of the assumption was to make each agent equally likely to be chosen at each step, which allows entropy to be used to determine the likelihood of various outcomes, just as in statistical mechanics [12]. When the underlying distribution is no longer uniform, entropy is no longer sufficient to analyze the situation, but, as we show here, *relative entropy* [5] (which can be viewed as a generalization of entropy to allow non-uniform underlying distributions) can be used instead. The essential connection is that where entropy can be interpreted as a measure of the number of ways a distribution can be realized, relative entropy can be interpreted as a weighted measure where some outcomes are more likely to be seen than others. While this connection is well understood and has been used to derive similar results for independent random variables [6], we believe that our application of the technique to a situation where the underlying random variables are not independent is novel and perhaps of independent interest. Using relative entropy, all our previous results can be extended to the general case. The details of this extension are omitted for space, but are available in the full version [13].

This model does make a number of simplifying assumptions, but many can be relaxed without changing the fundamental behavior of the system (albeit at the cost of greater strategic and analytic complexity). For example, rather than all requests having the same value γ , the value of a request might be stochastic. This would mean that an agent may forgo a low-valued request if he is low on money. This fact may impact the

threshold he chooses and introduces new decisions about which requests to make, but the overall behavior of the system will be essentially unchanged.

A more significant assumption is that prices are fixed. However, our results provide insight even if we relax this assumption. With variable prices, the behavior of the system depends on the value of β . For large β , where there are a significant number of volunteers to satisfy most requests, we expect the resulting competition to effectively produce a fixed price, so our analysis applies directly. For small β , where there are few volunteers for each request, variable prices can have a significant impact. In particular, sybils and collusion are more likely to result in inflation or deflation rather than a change in utility.

However, allowing prices to be set endogenously, by bidding, has a number of negative consequences. For one thing, it removes the ability of the system designer to optimize the system using monetary policy. In addition, for small β , it makes it possible for colluding agents to form a cartel to fix prices on a resource they control. It also greatly increases the strategic complexity of using the system: rather than choosing a single threshold, agents need an entire pricing scheme. Finally, the search costs and costs of executing a transaction are likely to be higher with variable prices. Thus, in many cases we believe that adopting a fixed price or a small set of fixed prices is a reasonable design decision.

3 Sybils

Unless identities in a system are tied to a real world identity (for example by a credit card), it is effectively impossible to prevent a single agent from having multiple identities [7]. Nevertheless, there are a number of techniques that can make it relatively costly for an agent to do so. For example, Credence uses cryptographic puzzles to impose a cost each time a new identity wishes to join the system [18]. Given that a designer can impose moderate costs to sybiling, how much more need she worry about the problem? In this section, we show that the gains from creating sybils when others do not diminish rapidly, so modest costs may well be sufficient to deter sybiling by typical users. However, sybiling is a self-reinforcing phenomenon. As the number of agents with sybils gets larger, the cost to being a non-sybiling agent increases and so the incentive to create sybils becomes stronger. Therefore measures to discourage or prevent sybiling should be taken early before this reinforcing trend can start. Finally, we examine the behavior of systems where only a small fraction of agents have sybils. We show that under these circumstances a wide variety of outcomes are possible (even when all agents are of a single type), ranging from a crash (where no service is provided) to an increase in social welfare. This analysis provides insight into the tradeoffs between efficiency and stability that occur when controlling the money supply of the system's economy.

When an agent of type t creates sybils, he does not change anything about himself but rather how other agents perceive him. Thus the only change to his type might be an increase in χ_t if sybils cause other agents to choose him more often. We assume that each sybil is as likely to be chosen as the original agent, so creating s sybils increases χ_t by $s\chi_t$. (Sybils may have other impacts on the system, such as increased search costs, but we expect these to be minor.) The amount of benefit he derives from this depends on two probabilities: p_e (the probability he will earn a dollar this round if he volunteers)

and p_s (the probability he will be chosen to spend a dollar and there is another agent willing and able to satisfy his request). When $p_e < p_s$, the agent has more opportunities to spend money than to earn money, so he will regularly have requests go unsatisfied due to a lack of money. In this case, the fraction of requests he has satisfied is roughly p_e/p_s , so increasing p_e results in a roughly linear increase in utility. When p_e is close to p_s , the increase in satisfied requests is no longer linear, so the benefit of increasing p_e begins to diminish. Finally, when $p_e > p_s$, most of the agent's requests are being satisfied so the benefit from increasing p_e is very small. Figure 1 illustrates an agent's utility as p_e varies for $p_s = .0001$.¹ We formalize the relationship between p_e , p_s , and utility in the following theorem.

Theorem 1. *Let p_s be the probability that a particular agent is chosen to make a request in a given round and there is some other agent willing and able to satisfy it, p_e be the probability that the agent earns a dollar given that he volunteers, $r = p_e/p_s$, and k be the agent's strategy (i.e., threshold). In the limit as the number of rounds goes to infinity, the fraction of the agent's requests that have an agent willing and able to satisfy them that get satisfied is $(r - r^{k+1})/(1 - r^{k+1})$ if $r \neq 1$ and $k/(k + 1)$ if $r = 1$.*

Proof. Since we consider only requests that have another agent willing and able to satisfy them, the request is satisfied whenever the agent has a non-zero amount of money. Since we have a fixed strategy and probabilities, consider the Markov chain whose states are the amount of money the agent has and the transitions describe the probability of the agent changing from one amount of money to another. This Markov chain satisfies the requirements to have a stationary distribution and it can be easily verified that the distribution gives the agent probability $r^i(1 - r)/(1 - r^{k+1})$ of having i dollars if $r \neq 1$ and probability $1/(k + 1)$ if $r = 1$ [5]. This gives the probabilities given in the theorem.

Theorem 1 also gives insight into the equilibrium behavior with sybils. Clearly, if sybils have no cost, then creating as many as possible is a dominant strategy. However, in practice, we expect there is some modest overhead involved in creating and maintaining a sybil, and that a designer can take steps to increase this cost without unduly burdening agents. With such a cost, adding a sybil might be valuable if p_e is much less than p_s , and a net loss otherwise. This makes sybils a self-reinforcing phenomenon. When a large number of agents create sybils, agents with no sybils have their p_e significantly decreased. This makes them much worse off and makes sybils much more attractive to them. Figure 2 shows an example of this effect. This self-reinforcing quality means it is important to take steps to discourage the use of sybils before they become a problem. Luckily, Theorem 1 also suggests that a modest cost to create sybils will often be enough to prevent agents from creating them because with a well chosen value of m , few agents should have low values of p_e .

We have interpreted Figures 1 and 2 as being about changes in χ due to sybils, but the results hold regardless of what caused differences in χ . For example, agents may choose

¹ Except where otherwise noted, this and other figures assume that $m = 4$, $n = 10000$ and there is a single type of rational agent with $\alpha = .08$, $\beta = .01$, $\gamma = 1$, $\delta = .97$, $\rho = 1$, and $\chi = 1$. These values are chosen solely for illustration, and are representative of a broad range of parameter values. The figures are based on calculations of the equilibrium behavior. The algorithm used to find the equilibrium is described in [14].

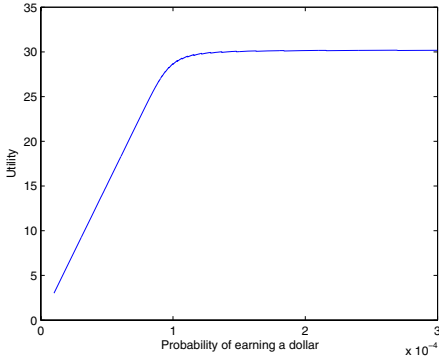


Fig. 1. The effect of p_e on utility

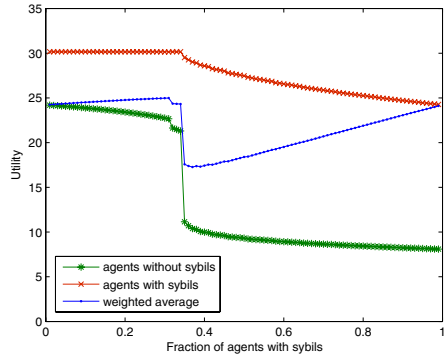


Fig. 2. The effect of sybils on utility

a volunteer based on characteristics such as connection speed or latency. If these characteristics are difficult to verify and do impact decisions, our results show that agents have a strong incentive to lie about them. This also suggests that the decision about what sort of information the system should enable agents to share involves tradeoffs. If advertising legitimately allows agents to find better service or more services they may be interested in, then advertising can increase social welfare. But if these characteristics impact decisions but have little impact on the actual service, then allowing agents to advertise them can lead to a situation like that in Figure 2, where some agents have a significantly worse experience.

We have seen that when a large fraction of agents have sybils, those agents without sybils tend to be starved of opportunities to work. However, as we saw in Figure 2, when a small fraction of agents have sybils this effect (and its corresponding cost) is small. Surprisingly, if there are few agents with sybils, an increase in the number of sybils these agents have can actually result in a *decrease* of their effect on the other agents. Because agents with sybils are more likely to be chosen to satisfy any particular request, they are able to use lower thresholds and reach those thresholds faster than they would without sybils, so fewer are competing to satisfy any given request. Furthermore, since agents with sybils can almost always pay to make a request, they can provide more opportunities for other agents to satisfy requests and earn money. Social welfare is essentially proportional to the number of satisfied requests (and is exactly proportional to it if everyone shares the same values of α and γ), so a small number of agents with a large number of sybils can improve social welfare, as Figure 3 shows. Note that this is not necessarily a Pareto improvement. For the choice of parameters in this example, social welfare increases when the agents create at least two sybils, but agents without sybils are worse off unless the agents with sybils create at least eight sybils. As the number of agents with sybils increases, they are forced to start competing with each other for opportunities to earn money and so are forced to adopt higher thresholds and this benefit disappears. This is what causes the discontinuity in Figure 2 when approximately a third of the agents have sybils.

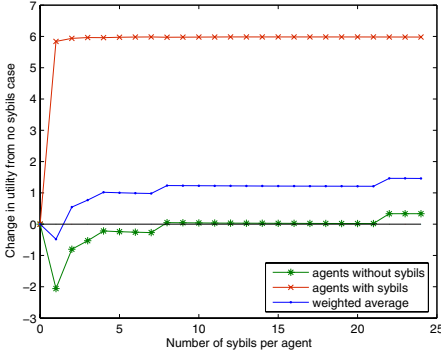


Fig. 3. Sybils can improve utility

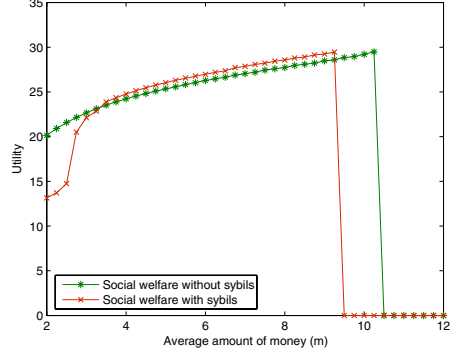


Fig. 4. Sybils can cause a crash

This observation about the discontinuity also suggests another way to mitigate the negative effects of sybils: increase the amount of money in the system. This effect can be seen in Figure 4, where for $m = 2$ social welfare is very low with sybils but by $m = 4$ it is higher than it would be without sybils.

Unfortunately, increasing the average amount of money has its own problems. Recall from Section 2 that, if the average amount of money per agent is too high, the system will crash. It turns out that just a small number of agents creating sybils can have the same effect, as Figure 4 shows. With no sybils, the point at which social welfare stops increasing and the system crashes is between $m = 10.25$ and $m = 10.5$. If one fifth of the agents each create a single sybil, the system crashes if $m = 9.5$, a point where, without sybils, the social welfare was near optimal. Thus, if the system designer tries to induce optimal behavior without taking sybils into account, the system will crash. Moreover, because of the possibility of a crash, raising m to tolerate more sybils is effective only if m was already set conservatively.

This example shows that there is a significant tradeoff between efficiency and stability. Setting the money supply high can increase social welfare, but at the price of making the system less stable. Moreover, as the following theorem shows, whatever efficiencies can be achieved with sybils can be achieved without them, at least if there is only one type of agent. The theorem does require a technical condition similar in spirit to *N-replica economies* [15] to rule out transient equilibria that exist only for limited values of n . In our results, we are interested in systems (T, \mathbf{f}, n, m) where T , \mathbf{f} , and m are fixed, but n varies. This leads to a small technical problem: there are values of n for which \mathbf{f} cannot be the fraction of agents of each type nor can m be the average amount of money (since, for example, m must be a multiple of $1/n$). This technical concern can be remedied in a variety of ways; the approach we adopt is one used in the literature on *N-replica economies*.

Definition 1. A strategy profile \mathbf{k} is an *asymptotic equilibrium* for a system (T, \mathbf{f}, n, m) if for all n' such that $n' = cn$ for integer $c > 0$, \mathbf{k} is a Nash equilibrium for (T, \mathbf{f}, n', m) . ■

Consider a system where all agents have the same type t . Suppose that some subset of the agents have created sybils, and all the agents in the subset have created the same number of sybils. We can model this by simply taking the agents in the subsets to have a new type s , which is identical to t except that the value of χ increases. Thus, we state our results in terms of systems with two types of agents, t and s .

Theorem 2. *Suppose that t and s are two types that agree except for the value of χ , and that $\chi_t < \chi_s$. If (k_t, k_s) is an asymptotic equilibrium for $(\{t, s\}, \mathbf{f}, m)$ with social welfare x , then there exists an m' and n' such that (k_s) is an asymptotic equilibrium for $(\{t\}, \{1\}, n', m')$ with social welfare at least x .*

Proof. We show this by finding an m' such that agents in the second system that play some strategy k get essentially the same utility that an agent with sybils would by playing that strategy in the first system. Since k_s was the optimal strategy for agents with sybils in the first system, it must be optimal in the second system as well. Since agents with sybils have greater utility than those without (they could always avoid volunteering some fraction of the time to effectively lower χ_s), social welfare will be at least as high in the second system as in the first.

Once strategies are fixed, an agent's utility depends only on α_t , γ_t , δ_t , p_s , and p_e . The first three are constants. Because the equilibrium is asymptotic, for sufficiently large cn , almost every request by an agent with money is satisfied (the expected number of agents wishing to volunteer is a constant fraction of cn). Therefore, p_s is essentially $1/cn$, his probability of being chosen to make a request. With fixed strategies, any value of p_e between 0 and β_t can be achieved by taking the appropriate m' . Take m' such that, if every agent in the second system plays k_s , the resulting p_e will be the same as the agents with sybils had in the original equilibrium. Note that p_s may decrease slightly because fewer agents will be willing to volunteer, but we can take $n' = cn$ for sufficiently large c to make this decrease arbitrarily small.

The analogous result for systems with more than one type of agent is not true. Consider the situation shown in Figure 2, where forty percent of the agents have two sybils. With this population, social welfare is lower than if no agents had sybils. However, the same population could be interpreted simply as having two different types, one of whom is naturally more likely to be chosen to satisfy a request. In this situation, if the agents less likely to be chosen created exactly two sybils each, the each agent would then be equally likely to be chosen and social welfare would increase. While changing m can change the relative quality of the two situations, a careful analysis of the proof of Theorem 2 shows that, when each population is compared using its optimal value of m , social welfare is greater with sybils. While situations like this show that it is theoretically possible for sybils to increase social welfare beyond what is possible by adjusting the average amount of money, this outcome seems unlikely in practice. It relies on agents creating just the right number of sybils. For situations where such a precise use of sybils would lead to a significant increase in social welfare, a designer could instead improve social welfare by biasing the algorithm agents use for selecting which volunteer will satisfy the request.

4 Collusion

Agents that collude gain two major benefits. The primary benefit is that they can share money, which simultaneously makes them less likely to run out of money and be unable to make a request and allows them to pursue a joint strategy for determining when to work. The secondary benefit, but important in particular for larger collusive groups, is that they can satisfy each other's requests. The effects of collusion on the rest of the system depend crucially on whether agents are able to volunteer to satisfy requests when they personally cannot satisfy the request but one of their colluding partners can. In a system where a request is for computation, it seems relatively straightforward for an agent to pass the computation to a partner to perform and then pass the answer back to the requester. On the other hand, if a request is a piece of a file it seems less plausible that an agent would accept a download from someone other than the person he expects and it seems wasteful to have the chosen volunteer download it for the sole purpose of immediately uploading it. If it is possible for colluders to pass off requests in this fashion, they are able to effectively act as sybils for each other, with all the consequences we discussed in Section 3. However, if agents can volunteer only for requests they can personally satisfy, the effects of collusion are almost entirely positive.

Since we have already discussed the consequences of sybils, we will assume that agents are able to volunteer only to satisfy requests that they personally can satisfy. Furthermore, we make the simplifying assumption that agents that collude are of the same type, because if agents of different types collude their strategic decisions become more complicated. For example, once the colluding group has accumulated a certain amount of money it may wish to have only members with small values of α volunteer to satisfy requests. Or when it is low on money it may wish to deny use of money to members with low values of γ . This results in strategies that involve sets of thresholds rather than a single threshold, and while nothing seems fundamentally different about the situation, it makes calculations significantly more difficult.

With these assumptions, we now examine how colluding agents will behave. Because colluding agents share money and types, it is irrelevant which members actually perform work and have money. All that matters is the total amount of money the group has. This means that when the group needs money, everyone in the group volunteers for a job. Otherwise no one does. Thus, the group essentially acts like a single agent, using a threshold which will be somewhat less than the sum of the thresholds that the individual agents would have used, because it is less likely that c agents will make ck requests in rapid succession than a single agent making k . Furthermore, some requests will not require scrip at all because they can potentially be satisfied by other members of the colluding group. When deciding whether the group should satisfy a member's request or ask for an outside volunteer to fulfill it, the group must decide whether it should pay a cost of α to avoid spending a dollar. Since not spending a dollar is effectively the same as earning a dollar, the decision is already optimized by the threshold strategy; the group should always attempt to satisfy a request internally unless it is in a temporary situation where the group is above its threshold.

Figure 5 shows an example of the effects of collusion on agents' utilities as the size of collusive groups increases. As this figure suggests, the effects typically go through three phases. Initially, the fraction of requests colluders satisfy for each other is small. This

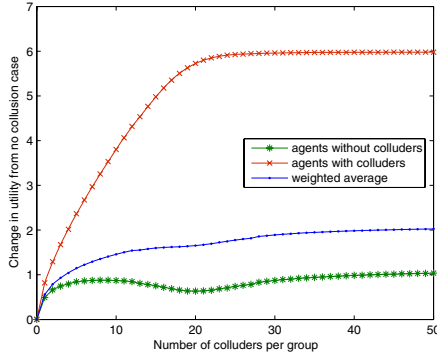


Fig. 5. The effect of collusion on utility

means that each collusive group must work for others to pay for almost every request its members make. However, since they share money, the colluders do not have to work as often as individuals would. Thus, other agents have more opportunity to work, and every agent's p_e increases, making all agents better off.

As the number of colluders increases, the fraction of requests they satisfy internally grows significant. We can think of p_s as decreasing in this case, and view these requests as being satisfied “outside” the scrip system because no scrip changes hands. This is good for colluders, but is bad for other agents whose p_e is lower since fewer requests are being made. Even in this range, non-colluding agents still tend to be better off than if there were no colluders because the overall competition for opportunities to work is still lower. Finally, once the collusive group is large enough, it will have a low p_s relative to p_e . This means the collusive group can use a very low threshold which again begins improving utility for all agents. Since collusion is difficult to maintain (the problem of incentivizing agents to contribute is the whole point of using scrip), we would expect the size of collusive groups seen in practice to be relatively small. Therefore, we expect that for most systems collusion will be Pareto improving. Note that, as with sybils, this decreased competition can also lead to a crash. However, if the system designer is monitoring the system and encouraging and expecting collusion she can reduce m appropriately and prevent a crash.

These results also suggest that creating the ability to take out loans (with an appropriate interest rate) is likely to be beneficial. Loans gain the benefits of reduced competition without the accompanying cost of fewer requests being made in the system. However, implementing a loan mechanism requires addressing a number of other incentive problems. For example, *whitewashing*, where agents take on a new identity (in this case to escape debts) needs to be prevented [9].

Acknowledgements. EF, IK, and JH are supported in part by NSF grant ITR-0325453. JH is also supported in part by NSF grant IIS-0812045 and by AFOSR grants FA9550-08-1-0438 and FA9550-05-1-0055. EF is also supported in part by NSF grant CDI-0835706.

References

1. Adar, E., Huberman, B.A.: Free riding on Gnutella. *First Monday* 5(10) (2000)
2. Aperjris, C., Johari, R.: A peer-to-peer system as an exchange economy. In: *GameNets 2006: Proceeding from the 2006 Workshop on Game Theory for Communications and Networks*, p. 10 (2006)
3. Brunelle, J., Hurst, P., Huth, J., Kang, L., Ng, C., Parkes, D., Seltzer, M., Shank, J., Youssef, S.: Egg: An extensible and economics-inspired open grid computing platform. In: *Third Workshop on Grid Economics and Business Models (GECON)*, pp. 140–150 (2006)
4. Chun, B., Buonadonna, P., AuYung, A., Ng, C., Parkes, D., Schneidman, J., Snoeren, A., Vahdat, A.: Mirage: A microeconomic resource allocation system for sensor network testbeds. In: *Second IEEE Workshop on Embedded Networked Sensors*, pp. 19–28 (2005)
5. Cover, T., Thomas, J.: *Elements of Information Theory*. John Wiley & Sons, Inc., New York (1991)
6. Csiszár, I.: Sanov property, generalized i-projection and a conditional limit theorem. *The Annals of Probability* 12(3), 768–793 (1984)
7. Douceur, J.R.: The sybil attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) *IPTPS 2002*. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002)
8. Friedman, E.J., Halpern, J.Y., Kash, I.A.: Efficiency and Nash equilibria in a scrip system for P2P networks. In: *Proc. Seventh ACM Conference on Electronic Commerce (EC)*, pp. 140–149 (2006)
9. Friedman, E.J., Resnick, P.: The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy* 10(2), 173–199 (2001)
10. Hens, T., Schenk-Hoppe, K.R., Vogt, B.: The great Capitol Hill baby sitting co-op: Anecdote or evidence for the optimum quantity of money? *J. of Money, Credit and Banking* 9(6), 1305–1333 (2007)
11. Hughes, D., Coulson, G., Walkerdine, J.: Free riding on Gnutella revisited: The bell tolls? *IEEE Distributed Systems Online* 6(6) (2005)
12. Jaynes, E.T.: Where do we stand on maximum entropy? In: Levine, R.D., Tribus, M. (eds.) *The Maximum Entropy Formalism*, pp. 15–118. MIT Press, Cambridge (1978)
13. Kash, I.A., Friedman, E.J., Halpern, J.Y.: Manipulating scrip systems: Sybils and collusion. [arXiv:0903.2278v1](https://arxiv.org/abs/0903.2278v1)
14. Kash, I.A., Friedman, E.J., Halpern, J.Y.: Optimizing scrip systems: Efficiency, crashes, hoarders and altruists. In: *Proc. Eighth ACM Conference on Electronic Commerce (EC)*, pp. 305–315 (2007)
15. Mas-Colell, A., Whinston, M.D., Green, J.R.: *Microeconomic Theory*. Oxford University Press, Oxford (1995)
16. Ng, C., Buonadonna, P., Chun, B., Snoeren, A., Vahdat, A.: Addressing strategic behavior in a deployed microeconomic resource allocator. In: *Third Workshop on Economics of Peer-to-Peer Systems (P2PECON)*, pp. 99–104 (2005)
17. Vishnumurthy, V., Chandrakumar, S., Siler, E.G.: KARMA: a secure economic framework for peer-to-peer resource sharing. In: *First Workshop on Economics of Peer-to-Peer Systems (P2PECON)* (2003)
18. Walsh, K., Siler, E.G.: Experience with an object reputation system for peer-to-peer file-sharing. In: *Third Symp. on Network Systems Design & Implementation (NSDI)*, pp. 1–14 (2006)
19. Yokoo, M., Sakurai, Y., Matsubara, S.: The effect of false-name bids in combinatorial auctions: new fraud in internet auctions. *Games and Economic Behavior* 46(1), 174–188 (2004)