Using Mechanism Design to Prevent False-Name Manipulations

Vincent Conitzer
Department of Computer Science
Duke University
Durham, NC 27708, USA
conitzer@cs.duke.edu

Makoto Yokoo
Department of ISEE
Kyushu University
Fukuoka 819-0395, Japan
yokoo@is.kyushu-u.ac.jp

Abstract

When mechanisms such as auctions, rating systems, and elections are run in a highly anonymous environment such as the Internet, a key concern is that a single agent can participate multiple times by using false identifiers. Such *false-name manipulations* have traditionally not been considered in the theory of mechanism design. In this article, we review recent efforts to extend the theory to address this. We first review results for the basic concept of *false-name-proofness*. Because some of these results are very negative, we also discuss alternative models that allow us to circumvent some of these negative results.

Technologies such as the Internet allow many spatially distributed parties (or *agents*) to rapidly interact according to intricate protocols. Some of the most exciting applications of this involve making decisions based on the agents' preferences (for a more detailed discussion, see Conitzer (2010)). For example, in electronic commerce, agents can bid on items in online auctions. This results in an allocation of the items for sale to the agents bidding in the auctions; one view of this is that we decide on the allocation based on the preferences that the agents reveal through their bids. Similarly, in an online rating system, the quality of a product, article, video, etc. is decided based on the submitted ratings. In an online election, an alternative is selected based on the submitted votes. In general, a *mechanism* takes the submitted preferences (bids, ratings, votes, etc.) as input, and produces an outcome as output.

One issue with such mechanisms is that sometimes, an agent has an incentive to report her preferences insincerely, as this will result in an outcome that she prefers. Agents that respond to such incentives are said to report *strategically*. For example, in a (first-price, sealed-bid) auction, a bidder may value the item for sale at \$100, but she may strategically choose to bid only \$70 because she believes that she will still win with this bid, and pay less. Similarly, in a rating system, an agent who believes that the product should receive an overall rating of 7 may strategically give it a rating of 10, in order to "correct" earlier ratings by others that resulted in an average rating of 6 so far. Finally, in an election, an agent whose favorite alternative is A may strategically claim that B is her most-preferred alternative, because she believes that A has no realistic chance of winning, and she very much wants to keep C from winning.

A fundamental problem caused by such strategic reports is that they may result in the "wrong" outcome. For example, let us again consider the bidder who values the item at \$100 but chooses to bid \$70 instead because she believes that she will still win with this bid. It is possible that she is mistaken—in particular, it could happen that there is another bidder who values the item at \$90 but, being more cautious than the former bidder, bids \$80. In this case, the latter bidder wins, even though from the perspective of the bidders' *true* valuations, it would have been more economically efficient for the item to end up with the former bidder. Similar failures can occur with rating and voting.

Mechanism design, which is based on game theory, concerns the study of how to design mechanisms that result in good outcomes even when the agents act strategically. A fundamental result known as the *revelation principle* (Gibbard 1973; Green and Laffont 1977; Dasgupta, Hammond, and Maskin 1979; Myerson 1979) shows that without loss of generality, we can restrict our attention to the design of *incentive compatible* mechanisms—that is, mechanisms in which it is in each agent's best interest to report truthfully. A strong notion of incentive compatibility is *strategy-proofness*: a mechanism is strategy-proof if no agent ever benefits from misreporting, regardless of the others' reports.

However, in highly anonymous settings such as the Internet, declaring preferences insincerely is not the only way to manipulate the mechanism. Often, it is possible for an agent to pretend to be multiple agents, and participate in the mechanism multiple times. Many Web applications only require a valid e-mail address, and it is easy for one agent to create multiple e-mail accounts. In an online election, this allows a single agent to vote multiple times—a significant drawback of online elections. Similarly, in a rating system, a single agent can manipulate the average or median rating to be effectively anything by rating the product a sufficient number of times. (At some level, this is not fundamentally different from the situation in elections: rating can be thought of as a special case of voting.) It is perhaps less obvious how using multiple identities to bid can help in an auction, but we will see examples of this shortly. We will refer to this type of strategic behavior as *false-name manipulation*. It is closely related to the notion of a *Sybil attack* in the sys-

tems literature (Douceur 2002), where an attacker also uses pseudonymous identities to subvert a system. As in the case of strategic misreporting of preferences, the main downside of false-name manipulation for the system as a whole is that it may result in suboptimal outcomes.

There are several ways in which the problem of false-name manipulation can be addressed. One approach is to try to prevent it directly. For example, we can require users to submit information that would completely identify them in the real world, such as a social security number. However, such an approach would doom most Internet-based applications to failure, because users are extremely averse to giving out such information—for example due to concerns about identity theft, or simply because the user prefers to stay anonymous. Various alternative approaches to directly preventing false-name manipulation have been pursued.

- A Completely Automated Public Turing Test to Tell Computers and Humans Apart, or CAPTCHA (von Ahn et al. 2003; von Ahn, Blum, and Langford 2004), is an automated test that is easy to pass for humans, but difficult to pass for computers. While CAPTCHAs can prevent a manipulator from obtaining a very large number of identifiers by writing a program that automatically registers for them, they do not prevent the manipulator from obtaining multiple identifiers by hand.
- A recent approach consists of attempting to create a test that is easy for a person to pass once, but difficult for a single person to pass more than once (Conitzer 2008b). Early attempts to design such tests focused on memory tests that were set up in such a way that a user taking the test a second time would become confused with the first time that she took the test. Unfortunately, for the tests designed so far, results from studies with human subjects are nowhere close to robust enough for practical use.
- Another direction is to use social-network structure to prevent a user from obtaining too many identifiers. Here, the basic idea is that it is easy to create new nodes in the network, as well as edges among them, but it is difficult to get legitimate nodes to link to these new nodes—so that if a user creates many false nodes, they will be disconnected from the legitimate nodes by an unexpectedly small cut. This observation has been leveraged to limit the number of identifiers that a manipulating user can obtain (Yu et al. 2008; 2010).
- A simple approach is to limit the number of identifiers registered from one IP address. A downside of this approach is that there are often many users behind a single IP address, so that the limit must be set rather high.

Some of these approaches can successfully prevent a single agent from obtaining an extremely large number of identifiers. This may be sufficient if the agent's goal is, for example, to send spam e-mail. However, in the settings in which we are interested, this is generally not sufficient: an agent may still derive significant benefits from creating just a few false names.

In this article, we consider to what extent the issue of false-name manipulation can be addressed using techniques from mechanism design. Under this approach, we accept the fact that it is possible for an agent to participate multiple times, but we design the mechanism—the rules that map reported preferences to outcomes—in such a way that good outcomes result even when agents strategically decide whether to participate multiple times. The primary approach to doing this is to simply ensure that it is always optimal for an agent to participate only once (again, a revelation principle can be given to justify this approach). A mechanism is said to be *false-name-proof* if no agent ever benefits from using multiple identifiers. The typical formal definition also implies strategy-proofness. In this article, we do not give formal mathematical definitions of false-name-proofness; rather, we rely on examples to illustrate the concept.

Voting

We will first discuss *voting* settings. One should immediately be suspicious of the idea that an election in which a single agent can vote multiple times can lead to good results, and the technical result that we will discuss in this section will lend support to this suspicion. A natural reaction is that we should simply avoid such elections. However, examples of real-world online elections abound.

An intriguing recent example of this phenomenon is the "New Seven Wonders of the World" election, a global election to elect contemporary alternatives to the ancient wonders. Anyone could vote, either by phone or over the Internet; for the latter, an e-mail address was required. One could also buy additional votes (of course, simply using another e-mail address was a much cheaper alternative). In spite of various irregularities (including unreasonably large numbers of votes in some cases (Dwoskin 2007)) and UNESCO distancing itself from the election, the election nevertheless seems to have attained some legitimacy in the public's mind.

To illustrate the difficulties that such online elections face, let us first consider an election with two alternatives, say, A and B. In this case, each voter prefers one of the two, and will be asked to vote for the one she prefers. If false-name manipulation is not possible, the most natural approach is to run the simple *majority* rule: the alternative with more votes wins (with some way of breaking ties, for example, flipping a coin). It is easy to see that this rule is strategy-proof: there is nothing that can be gained from voting for one's less-preferred alternative. Also, if we suppose that an agent receives utility 1 if her preferred alternative is elected, and 0 otherwise, then the majority rule maximizes the sum of the agents' utilities.

Unfortunately, the majority rule is clearly not false-name-proof. For example, consider an election in which one agent prefers A and two agents prefer B. If the two agents that prefer B each use a single identifier and vote truthfully, then the agent that prefers A has an incentive to create two additional fake identifiers, and vote for A with all three of her identifiers, to make A win. More generally, holding the other agents' votes fixed, an agent can always make her preferred alternative win by casting sufficiently many votes for that alternative.

From this, the difficulty of designing a good false-name-proof voting rule should be apparent. One may conjecture that votes are necessarily entirely meaningless in this context, and that we might as well choose the winning alternative randomly (flipping a coin), without regard to the votes. Doing so is certainly false-name-proof: in this case, there is no incentive to vote multiple times, because there is no incentive to vote at all! Obviously, this is not very satisfactory.

Conitzer (2008a) studies false-name-proof voting rules more thoroughly, and it turns out that we can do just a little better than choosing the winning alternative completely at random. Consider the following *unanimity* rule for two alternatives. If *all* the voters vote for the same alternative (and at least one vote is cast), then we choose that alternative; otherwise, we flip a fair coin to decide the winner. Using a case-by-case analysis, we see that this rule leaves an agent (who prefers, say, A) no incentive for manipulation:

- If B does not receive any votes from the other agents, voting truthfully results in a win for A;
- If both A and B receive votes from the other agents, then it does not matter what the agent does;
- If B receives votes from the other agents and A does not, then the agent wants to (truthfully) cast a vote for A to force the coin flip, but casting additional votes will have no effect.

While this rule avoids the bizarre scenario where we flip a coin even though all agents agree on what the preferred alternative is, it is otherwise still not very desirable. For example, even if 100 agents prefer A and only 1 agent prefers B, the probability that A wins is only 50%. Thus, we may wonder whether there is another false-name-proof voting rule that is a closer approximation of the majority rule. It turns out that the answer is negative: in a sense, the unanimity rule is the best we can do under the constraint of false-name-proofness.

In settings with more than two alternatives, there is an even more negative result: in a sense, the best we can do under the constraint of false-name-proofness is to choose two alternatives uniformly at random (without regard to the votes), and then run the unanimity rule on these two alternatives. This is somewhat reminiscent of another fairly negative characterization by Gibbard (1977) for the case of strategy-proof randomized voting rules (when there are no restrictions on preferences and false-name manipulation is not possible). Gibbard's characterization allows for rules such as:

- choose two alternatives at random and run a majority election between these two, or
- randomly choose one of the agents as a dictator, whose most-preferred alternative is then chosen.

Unfortunately, Gibbard's characterization does not allow for much more than these rules. Still, it is much more permissive than the characterization for false-name-proof rules. For example, choosing a random dictator is not false-name-proof: an agent would have an incentive to use many identifiers, to increase the chances that one of these will be chosen as the dictator. Also, unlike in the case of false-name-proofness, Gibbard's characterization poses no problem in the two-alternative case, because there it allows for the majority rule, which is quite natural. Finally, the strategy-proofness (in fact, group-strategy-proofness—no coalition of agents has an incentive to deviate) of the majority rule can be extended to more alternatives if we restrict the agents' possible preferences to *single-peaked* preferences (Black 1948; Moulin 1980). In contrast, for false-name-proofness, there appears to be little hope of finding a positive result based on restricting the preferences, because we already get a negative result for two alternatives.

We will discuss what can be done about (or in spite of) this impossibility result later in this article. But, first, we turn to a discussion of combinatorial auctions, in which the concept of false-name-proofness was originally defined.

Combinatorial auctions

In a *combinatorial auction*, multiple items are simultaneously for sale. An agent (aka. bidder) is allowed to place complex bids on these items. For example, an agent may say, "If I receive both items A and B, that is worth \$100 to me, but if I only receive one of them, that is only worth \$10." This is a case of *complementarity*, where the items are worth more together than the sum of their parts. Complementarity often motivates the use of a combinatorial auction.

Generally, if I is the set of items, an agent i has a *valuation function* $v_i : 2^I \rightarrow \mathbb{R}$ that specifies how much she values each possible bundle of items, and her bid will be a reported valuation function $\hat{v}_i : 2^I \rightarrow \mathbb{R}$. (We consider only *sealed-bid* auctions here, where an agent only places a single bid; this is justified by the revelation principle.) Usually, the goal is to assign subsets of the items to the agents in a way that maximizes efficiency, that is, if agent i receives $S_i \subseteq I$ (where $S_i \cap S_j = \emptyset$ for $i \neq j$), the goal is to maximize $\sum_i v_i(S_i)$.

How can we incentivize truthful bidding in a combinatorial auction? To explain this, it is helpful to first consider a single-item auction, in which each agent i bids some amount $\hat{v}_i$ on the item. The standard solution here is the *Vickrey* or *second-price sealed-bid* auction (Vickrey 1961), where the highest bid wins and pays the price of the second-highest bid. This is strategy-proof, and the reason is that the winning bidder automatically pays *the smallest amount she could have bid* and still won the item. It turns out that this intuition generalizes to combinatorial auctions: in the *Generalized Vickrey Auction (GVA)*, an allocation is chosen that maximizes efficiency according to the reported valuation functions—that is, it maximizes $\sum_i \hat{v}_i(S_i)$ (how ties are broken is not essential); each bidder pays the smallest amount she could have bid to win her bundle of items. The GVA is a special case of the *Clarke mechanism* (Clarke 1971), and it is strategy-proof.

However, the GVA is not false-name-proof. For example, suppose we are allocating two items, A and B. Agent 1 bids (reports a valuation of) \$100 for the bundle $\{A, B\}$ of both items (and \$0 for any other bundle). Suppose agent 2's true valuation for the bundle $\{A, B\}$ of both items is \$80 (and it is \$0 for any other bundle). Thus, if agent 2 truthfully reports her valuation \$80, she does not win any item. Alternatively, in a highly anonymous environment, agent 2 can participate

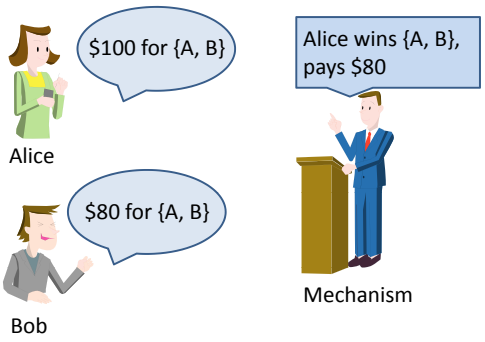

Figure 1: The Generalized Vickrey Auction in a standard setting where the true identities of the agents can be observed.

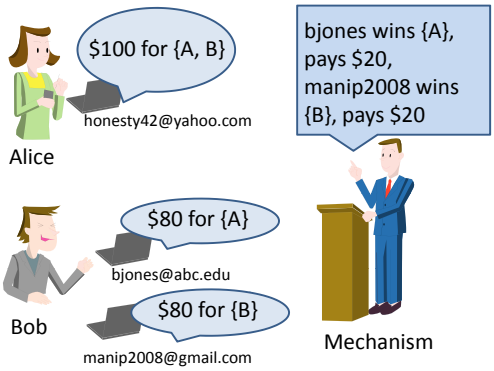

Figure 2: The Generalized Vickrey Auction in a highly anonymous (Internet) setting. The mechanism cannot observe the agents' true identities directly; all it can observe is the identifiers (e-mail addresses) and the bids that are submitted through those identifiers.

under two different identifiers, $2a$ and $2b$; if $2a$ bids \$80 on $\{A\}$, and $2b$ bids \$80 on $\{B\}$, then $2a$ and $2b$ will both win their item (so that 2 wins both items). Moreover, the GVA payments of $2a$ and $2b$ are 20 each, because each of them could have reduced the bid to \$20 and still won the item. Hence, using the false-name manipulation, agent 2 gets both items and pays \$40 in total. Thus, this manipulation is profitable for agent 2. This results in an inefficient outcome, because 1 values the items more. Figures 1 and 2 show how this example illustrates the difference between standard and highly anonymous mechanism design settings.

While the previous example already illustrates the potential for false-name manipulation in the GVA, a somewhat different type of false-name manipulation is also possible. Namely, the manipulating agent can bid under multiple identifiers, but then, once the outcome has been decided, fail to respond for some of them—that is, have these identifiers refuse to pay. While these refusing identifiers will presumably not obtain the items that they won, it is possible that their presence was beneficial to the agent's other identifiers. For example, suppose bidder 1 bids \$100 for the bundle $\{A, B\}$, and bidder 2 bids \$40 for the bundle $\{A\}$. Bidder 3—the false-name bidder—has true valuation \$20 for the bundle $\{B\}$ and any superset of it (and \$0 for any other bundle). Under the GVA, if bidder 3 bids truthfully (which is optimal if false-name bidding is impossible), she wins nothing and pays nothing. She also cannot benefit from the type of false-name bidding in the previous example: for example, she can win both items by bidding \$100 for $\{A\}$ under identifier $3a$ and \$60 for $\{B\}$ under identifier $3b$, in which case $3a$ pays \$40 and $3b$ pays \$0; but her valuation for $\{A, B\}$ is only \$20, so this would make her worse off. However, now suppose that she can disown identifier $3a$ (e.g., by never checking that e-mail account anymore), never making the payment and never collecting A. Then, she has obtained B with the other identifier at a price of \$0. This type of manipulation is not addressed by the standard definition of false-name-proofness, but recent work (Guo and Conitzer 2010) considers a modified definition that does also capture this type of manipulation. In any case, most (but not all) of the standard false-name-proof mechanisms also satisfy this stronger condition.

At this point, the obvious question is: can we fix the GVA mechanism, or develop a completely new mechanism, so that the obtained mechanism is false-name-proof and achieves efficient outcomes? Unfortunately, the answer is no. Yokoo, Sakurai, and Matsubara (2004) give a simple generic counter-example illustrating that there exists no false-name-proof combinatorial auction mechanism that always achieves an efficient outcome.[1] They also show that the revelation principle holds for false-name-proof mechanisms. This implies that there exists no efficient mechanism in general when false-name bids are possible.

Another question we might ask is: although the GVA is not false-name-proof in general, can we identify some (hopefully natural and general) special cases where the GVA is false-name-proof? Yokoo, Sakurai, and Matsubara (2004) show that a well-known condition called *submodularity* is sufficient to guarantee that the GVA is false-name-proof. Submodularity is defined as follows: for any subset of bidders N, for two sets of items S_1, S_2, the following condition

[1] Fairly weak conditions that preclude false-name-proofness were later given by Rastegari, Condon, and Leyton-Brown (2007).

holds:

$$V^*(N, S_1)+V^*(N, S_2) \geq V^*(N, S_1 \cup S_2)+V^*(N, S_1 \cap S_2),$$

where $V^*(N, S)$ represents the social surplus (sum of valuations) when allocating S optimally among N. The idea is that additional items become less useful as there are more items already. This condition does not hold for Alice in Figure 2. When Alice has nothing, adding A does not increase her valuation. When Alice has B already, adding A increases her valuation from 0 to 80. In other words, A and B are *complementary* for Alice, i.e., the bundle is worth more than the sum of its parts.

Theoretically, the submodularity condition is very useful, since it guarantees several other desirable properties of the GVA, for example: the outcome is in the *core*—i.e., the seller does not wish to sell items to some loser rather than the winners; collusion by the losers is useless; and the condition facilitates the computation of the winners and payments (Müller 2006). However, the submodularity condition is of limited use, because in practice often a major motivation for using a combinatorial auction (rather than multiple single-item auctions sequentially) is that there is complementarity among the items—though, of course, combinatorial auctions may be useful in settings without complementarity as well.

A series of mechanisms that are false-name-proof in various settings has been developed: combinatorial auction mechanisms (Yokoo, Sakurai, and Matsubara 2001a; Yokoo 2003), multi-unit auction mechanisms (Yokoo, Sakurai, and Matsubara 2001b; Terada and Yokoo 2003; Iwasaki, Yokoo, and Terada 2005), double auction mechanisms (Sakurai and Yokoo 2002; 2003; Yokoo, Sakurai, and Matsubara 2005), and combinatorial procurement auctions (Suyama and Yokoo 2005).

For the purpose of illustration, let us describe some false-name-proof combinatorial auction mechanisms. The simplest such mechanism is called the *Set mechanism*. It allocates all items I to a single bidder, namely, the bidder with the largest valuation for the grand bundle of all items. Effectively, it sells the grand bundle as a single good, in a Vickrey/second-price auction. It is not difficult to see that false-name bids are ineffective under the Set mechanism: there is only one winner and placing additional bids only increases the payment of the winner.

Of course, we would hope to find a mechanism that does better than this rather trivial Set mechanism. A non-trivial false-name-proof mechanism called the Minimal Bundle (MB) mechanism (Yokoo 2003) can be thought of as an improved version of the Set mechanism. (In the following description, we assume each agent is interested only in a single bundle (*single-minded*) for simplicity, but the general MB mechanism can also be applied to non-single-minded agents.) Let us assume bidder i is the winner under the Set mechanism. The grand bundle might contain some useless items for bidder i, i.e., it may be the case that for some $S \subsetneq I$, $v_i(S)$ is the same as $v_i(I)$. We call the minimal bundle S for which $v_i(S) = v_i(I)$ holds the *minimal bundle* for i. Instead of allocating all items I to bidder i, we first allocate $S_i \subseteq I$ to i, where S_i is the minimal bundle for i. Then, we consider the next highest bidder j; if her minimal bundle S_j does not overlap with S_i, then she wins S_j, and so on. The price for a bundle S is equal to the highest valuation of another bidder for a bundle that is minimal for that bidder and conflicting with S, i.e., it has an item in common with S.

Let us show a simple example. Assume there are four items, A, B, C and D, and five bidders. Their valuations are as follows.

bidder 1: \$100 for $\{A, B\}$
bidder 2: \$80 for $\{C, D\}$
bidder 3: \$70 for $\{B, D\}$
bidder 4: \$60 for $\{C\}$
bidder 5: \$50 for $\{A\}$

In this case, bidder 1 wins $\{A, B\}$. Since this bundle conflicts with bidder 3's bundle, the payment is \$70. Then, bidder 2 wins $\{C, D\}$. Since this bundle also conflicts with bidder 3's bundle, the payment is again \$70.

Again, under this mechanism, false-name bids are useless. If bidder 1 splits her bid and obtains $\{A\}$ and $\{B\}$ with separate identifiers, her payment would be \$50+\$70, which is more than her original payment of \$70. More generally, for disjoint bundles S_1 and S_2, the price for obtaining $S_1 \cup S_2$ is the maximum of the price of S_1 and the price of S_2. However, if the bidder obtains S_1 and S_2 with separate identifiers, then she must pay the sum of these prices. Also, placing additional bids only increases the payments of the winners.

An auction mechanism consists of an allocation rule and a payment rule. There have been several studies on characterizing allocation rules for which there exists a payment rule that makes the mechanism as a whole strategy-proof. Bikhchandani et al. (2006) propose *weak monotonicity* and show that it is a necessary and sufficient condition for strategy-proofness when several assumptions hold on the domain of valuation functions.

In a similar type of result, Todo et al. (2009) show that if (and only if) an allocation rule satisfies a condition called *sub-additivity* as well as weak monotonicity, then there exists an appropriate payment rule so that the mechanism becomes false-name-proof, i.e., sub-additivity and weak monotonicity fully characterize false-name-proof allocation rules. In other work, Iwasaki et al. (2010) derive a negative result showing that any false-name-proof combinatorial auction mechanism (satisfying certain conditions) must have a low worst-case efficiency ratio (not much better than that of the Set mechanism), and develop a mechanism whose worst-case efficiency ratio matches this theoretical bound.

Ways around the negative results

Many of the results so far are quite negative. This is especially the case in voting settings, where even when there are only two alternatives, the best possible rule is the unanimity rule, which will flip a fair coin unless all the voters agree on which alternative is better. Even in combinatorial auctions, we have a strong impossibility result about the worst-case efficiency ratio. Of course, the worst-case efficiency ratio may

not occur very often in practice—in particular, under some conditions on the valuations, even the regular GVA mechanism is false-name-proof. In any case, it is worthwhile investigating whether we can somehow circumvent these negative results, especially in voting settings.

A natural response is that we should just not run mechanisms, especially voting mechanisms, in highly anonymous settings! That is, we should run the mechanism in an environment where we can verify the identities of all of the agents. While this thought is not without its merit—it does not seem wise to conduct, for example, presidential elections over the Internet—it is apparent that many mechanisms *will* be run over the Internet, and objecting to this phenomenon will not make it go away. For example, numerous organizations stubbornly continue running polls over the Internet in spite of past troubles, and these polls can still have significant impact. The New Seven Wonders of the World event discussed earlier clearly illustrates this phenomenon: in spite of questionable methodology (and, eventually, questionable results), the election attracted an enormous amount of attention, as well as significant effort from various organizations that tried to get their preferred alternative elected. Moreover, a follow-up event, the New 7 Wonders of Nature, is already underway. Similarly, with the continued growth of e-commerce, the presence of product rating mechanisms and auctions on the Web seems more likely to increase than to decrease. It appears that when organizations decide whether to run a mechanism over the Internet, the convenience of doing so often far outweighs the potential trouble from false-name manipulations in their minds.

In the remainder of this section, we consider several ways around the impossibility results that do not require us to verify every identity.

Costly false names

The assumption that a manipulator can obtain an unlimited number of identifiers at no cost is not realistic. Setting up a free (say) e-mail account requires some effort, including, perhaps, solving a CAPTCHA. This effort comes at a (presumably small) economic cost that will make false-name manipulation somewhat less appealing. Can we design mechanisms that are false-name-proof *when these costs are taken into account*—that is, when the cost is taken into account false-name manipulation becomes strategically suboptimal—and that outperform mechanisms that are false-name-proof in the standard sense (i.e., when the cost of creating false names is not taken into account)?

It turns out that this is, in fact, possible (Wagman and Conitzer 2008). Of course, if the cost of creating an additional identifier is extremely high, then (with two alternatives) even the majority rule—choose the alternative preferred by more voters (breaking ties randomly)—becomes false-name-proof: even if the election is tied and casting one additional vote will make the difference, which is a case in which casting an additional vote has the greatest possible value to a manipulating agent, no agent will be willing to do this if the cost of creating an additional identifier is sufficiently high. Of course, it is unreasonable to expect the cost to be so high if it corresponds to something as trivial as solving a CAPTCHA. We may try to increase the cost—for example, by attempting to detect manipulating agents and severely punishing them in the real world, perhaps under some new law. Of course, this would be extremely difficult to do. Is there a mechanism that works even if the cost of creating another identifier is relatively small?

It turns out that this is possible, but we need to consider mechanisms that use randomization (and not just for tie-breaking). The problem with the majority rule is that when the election is currently tied, then a single additional vote for A will make the probability that alternative A wins jump from .5 to 1. For an agent that prefers A, this is an enormous incentive to cast another vote. To make this more concrete, let us suppose that the agent has a utility of 1 for A winning the election, and a utility of 0 for B winning the election. Then, the agent has an expected utility of .5 for the election being tied. Hence, the benefit of casting another vote is .5, which the agent will do if the cost of obtaining another identifier is less than .5.

However, now suppose that we use the following rule. If A and B are tied, then A (and hence also B) wins with probability .5. If A is ahead by one vote, then A wins with probability .51. If A is ahead by two votes, then A wins with probability .52, etc. If A is ahead by fifty or more votes, A wins with probability 1. Under this rule, the benefit of casting another vote is always at most .01, so as long as the cost of obtaining another identifier is greater than this, no agent will be incentivized to obtain additional identifiers.

The downside of this rule, of course, is that if A is ahead by (say) 25 votes, then with probability 25% we choose alternative B, which is suboptimal from a welfare perspective because A makes 25 more agents happy. However, one can make an argument that if the number of agents is large, then the probability that the alternatives are within fifty votes of each other is small—so that we almost always choose the alternative that would have won under the majority rule, which is the alternative that maximizes welfare.[2] The cost of obtaining a false identifier also plays a role. For example, if we are sure that the cost of obtaining a new identifier is always at least .05 for any agent, then we can increase the probability that A wins by .05 every time it receives another vote (and once A receives at least 10 more votes than B, A wins with probability 1). Thus, the larger the number of agents voting, and the larger the cost of obtaining an additional identifier, the closer the rule gets to the majority rule—while remaining false-name-proof.

Verifying only some of the identifiers

As pointed out above, a simple way of addressing the issue of false-name manipulation is to verify that all the identifiers correspond to real agents in the real world. If we do so, then it suffices to run a strategy-proof mechanism (assuming that we are not worried about collusion, etc.). Of course, this generally puts an unacceptable overhead on the system. On

[2]This may lead one to ask why we do not simply use the majority rule; the answer is that the majority rule is not false-name-proof with small costs, so that the votes can no longer be taken at face value.

the other hand, it is not clear that we must really verify *all* of the identifiers. For example, in an election between two alternatives, must we really verify identifiers who voted for the losing alternative? In a combinatorial auction, must we really verify the identifiers that placed a losing bid? One would think that this should not be necessary, because in both cases, these identifiers are losing anyway. Generally, we would like to verify as few identifiers as possible, but enough to make false-name manipulation suboptimal.

Conitzer (2007) pursues this approach in detail. The basic version of the model is as follows. The mechanism first collects the identifiers' reports of their preferences (for example, their votes or bids). Based on these reports, the verification protocol will ask a subset of the identifiers for real-world identifying information. If an agent participated under multiple identifiers, she will be able to respond for at most one of these identifiers. This is because if she responds for multiple identifiers with the same identifying information, then the manipulation is easily detected.[3] This poses no problem for the manipulating agent if the verification protocol asks for identifying information for at most one of her identifiers. However, if the verification protocol wishes to verify two of her identifiers, then the agent has a problem. She can choose to submit identifying information for either one, but must then stay silent for the other. If an identifier stays silent, the verification protocol knows that something fishy is going on: presumably, the reason that the identifier stays silent is that it is one of the identifiers used by a manipulating agent, who has chosen to respond for one of her other identifiers instead. However, the verification protocol cannot identify which identifier this is; nor can it, presumably, find the agent in the real world to punish her. Thus, it is assumed that all that the verification protocol can do is to remove the nonresponsive identifier(s) from the mechanism. If a nonempty set of identifiers is removed, then the verification protocol starts from scratch with the remaining reports (and can thus choose to verify additional identifiers).[4]

As a simple example, suppose that we wish to run a majority election between two alternatives. We can proceed as follows. First, let each identifier vote for either A or B. Suppose that A comes out ahead by l votes ($n_A = n_B + l$). Then, the verification protocol will ask for identifying information of $n_B + 1$ of the identifiers voting for A. If all of them respond with valid (and distinct) identifying information, we declare A the winner; otherwise, all the nonresponsive identifiers are removed, and the verification protocol starts anew with the remaining votes (note that the balance may have shifted to B now). In the end, we will have guaranteed that there were more responsive identifiers for the winning alternative than for the losing alternative. This removes any incentive for an agent to participate multiple times.

As another example, let us consider again a combinatorial auction in which we use the GVA. Agent 1 uses identifier 1 to bid \$100 for $\{A, B\}$, and agent 2 uses two identifiers, $2a$ and $2b$, and bids \$80 for $\{A\}$ with the former and \$80 for $\{B\}$ with the latter. Without verification, this is an effective false-name manipulation for agent 2. However, now let us suppose that the verification protocol decides to ask both $2a$ and $2b$ for identifying information. At this point, agent 2 has a problem. She can respond for neither, in which case both identifiers are removed and the manipulation was obviously ineffective. She can also respond for (say) $2a$, in which case $2b$ will be removed. After the removal of $2b$, $2a$ loses. Thus the manipulation becomes ineffective.

Unlike in the case of majority voting, for combinatorial auctions we have not made it clear how the verification protocol chooses which identifiers to ask for identifying information *in general*. Without a general specification of this, we cannot say whether the resulting overall mechanism is robust to false-name manipulation, or not. More ambitiously, can we give a general characterization of how much verification is needed in order to guarantee false-name-proofness? It turns out that we can.

First, let us say that a subset of at least two reports (votes or bids) *requires verification* if it is possible that this subset consists exactly of the identifiers used by a single agent, and that moreover, under the mechanism without verification, this agent is strictly benefiting from this manipulation (relative to just using a single identifier). For instance, in the previous example, the set of bids $\{2a, 2b\}$ requires verification, because we have already seen that an agent can benefit from using these two bids under the standard GVA. Similarly, in a majority election between two alternatives, if A is ahead of B by l votes, then any subset of $l + 1$ votes for A requires verification. (A subset of l or fewer votes does not require verification, because, if a single agent had submitted these l votes, then the agent would have succeeded just as well without manipulating, since her single true vote would have been enough to make A win.)

Now, it turns out that the necessary and sufficient condition for the verification protocol to guarantee false-name-proofness is as follows: for every subset that requires verification, the verification protocol must ask for identifying information from at least two of the identifiers in this subset. The intuition is simple: if the protocol asks at most one of the identifiers in the subset for identifying information, then we have found a situation where this subset would lead to a successful manipulation for an agent (because the agent can respond for this one identifier). On the other hand, a verification protocol that satisfies this condition will not leave any incentives for false-name manipulation, because in every situation where an agent engages in a false-name manipulation that might be beneficial, that agent will be asked to provide identifying information for at least two of her identifiers, and will hence fail to respond for at least one. It is

[3] It is possible that the agent can respond, for some identifier, with the identifying information of some other real-world agent. However, if the other real-world agent is a willing participant in this, then this is a case of collusion, not false-name manipulation. Otherwise, it is a case of identity theft, which would have to be prevented through other means.

[4] It may seem inefficient to start entirely from scratch; however, it facilitates the analysis. Moreover, because the protocol will remove any incentives to participate more than once, we may assume that, in fact, nobody will participate more than once, so that we do not expect any identifiers to not respond. (This, of course, does not mean that we do not need to do any verification at all, because then incentives to participate multiple times would reappear.)

important to recall here that when at least one identifier is removed, the verification protocol restarts, so that the identifiers that the agent has left when verification finally terminates completely cannot constitute a beneficial false-name manipulation.

Using social network structure to prevent false-name manipulation

Yet another way around the impossibility results for false-name-proofness is to use the social relationships among the agents. This is an idea that has been explored in the systems literature in the context of preventing Sybil attacks (Yu et al. 2008; 2010), but more recent work takes a mechanism design approach to this (Conitzer et al. 2010). Let us suppose that the entity running the mechanism (the center) has access to some social network structure on the identifiers. For example, in 2009, Facebook, Inc. conducted a poll among its users regarding its new terms of use. Facebook naturally knows the social network structure among the accounts. At the same time, it is easy for a user to create one or more fake accounts on Facebook. Moreover, it is easy for that user to connect some of her accounts (including her legitimate account) to each other, in arbitrary ways: the user simply logs in under one account, requests to connect to the other account, then logs in under that account and approves the request. However, it is more difficult for this user to connect her fake accounts to the accounts of other users: presumably, if the user sends a request from one of her fake accounts to another user's account, that user will not recognize the fake account, and reject the request.[5]

Let us assume that the manipulating user is unable to connect her fake accounts to the accounts of any other users. Of course, she can still connect her *true* account to the accounts of her real-life friends. This results in an odd-looking social network graph, where the manipulating user's true account provides the only connection between her fake accounts and the rest of the graph. Technically, her true account is a *vertex cut* of size 1 in the graph (where the vertices of the graph are the accounts). While the center cannot directly observe that the accounts on the other side of this vertex cut are indeed fake, she has reason to be suspicious of them. To remove incentives for false-name manipulation, the center can simply refuse to let such accounts participate. Of course, the downside of this is that in some cases, such accounts are actually legitimate accounts that just happen not to be very well connected to the rest of the graph. However, if this rarely occurs (for legitimate accounts), then preventing a few legitimate accounts from participating may be a reasonable price to pay to obtain a type of false-name-proofness guarantee.

There are several issues that need to be addressed to make this approach successful. The first is that, at least in principle, the manipulating user could build a structure of fake accounts that is incredibly large and complex, just as much so as the true social network. If she does so, then how does the center know which one is the true social network? To address this, we make the reasonable assumption that some accounts are *trusted* by the center, in the sense that these accounts are known to correspond to real agents. Thus, the accounts that the center should suspect are the ones that are separated from the trusted accounts by a vertex cut of size (at most) 1.

Another issue is that two legitimate users may conspire and create fake accounts together. In this case, they can connect the fake accounts to both of their legitimate accounts, so that there is no vertex cut of size 1. Of course, the two legitimate accounts now constitute a vertex cut of size 2. The general solution to this problem, unsurprisingly, is to refuse to let any account participate that is separated from the trusted accounts by a vertex cut of size at most k, where k is the largest number of users that can be conceived to conspire together.

In fact, this introduces another subtlety that needs to be addressed. It turns out that, if the only accounts that we prevent from participating are the ones that are separated from the trusted accounts by a vertex cut of size at most k, then there can still be incentives to create fake accounts. The reason is that, while these fake accounts will not be allowed to participate, they may nevertheless prevent *other* accounts from being separated from the trusted accounts by a vertex cut of size at most k, which can be strategically valuable. A solution is to apply the procedure iteratively: remove the accounts that are separated from the trusted accounts by a vertex cut of size at most k, then do the same on the remaining graph, etc., until convergence.

For the case where there are no trusted accounts, these techniques can still be applied if we have a method of *verifying* whether accounts are legitimate. Then, accounts that have passed the verification step take the role of trusted accounts. This naturally leads to the question of which accounts should be verified. One natural approach is to try to find a minimum-size set of accounts that, when verified, guarantees that *every* account in the graph is legitimate (i.e., no accounts are separated from the verified accounts by a vertex cut of size at most k). It turns out that this optimization problem can be solved in polynomial time, using a matroid property of this problem (Nagamochi, Ishii, and Ito 1993).

Coalitional games

In this final section before the conclusion, we consider one additional setting that is slightly different in nature from the mechanism design settings that we have considered so far. Here, we consider some elements of *cooperative game theory*, also known as *coalitional game theory*. Specifically, we consider settings in which agents can work together in a coalition to generate some type of value. For example, multiple companies may be able to increase their profit by working together. A key question is how to divide the gains that result from such cooperation among the members of the coalition.

Coalitional game theory provides several solution concepts that prescribe how much of the generated value each

[5] Of course, depending on the particular social network, some users may actually approve such requests. However, it seems that it would be easy to detect an account that illegitimately attempts to connect to many other users: it would have a noticeably low success rate, and these other users may report the account.

agent should receive. These solution concepts require us to know the characteristic function $w : 2^N \rightarrow \mathbb{R}$, where N is the set of all agents and $w(S)$ gives the value that would be generated by coalition S. A good example of a solution concept is that of the *Shapley value* (Shapley 1953). To understand the Shapley value, it helps to first consider a simpler way of dividing the value, which we will call the *marginal contribution* solution. Place the agents in some order $\pi : \{1, \ldots, n\} \rightarrow N$, where $\pi(i)$ gives the agent ordered ith. Let $S(\pi, k) = \{a \in N : \pi^{-1}(a) \leq k\}$ consist of the first k agents according to the order π. Then, we give each agent her *marginal contribution* to the coalition—that is, the agent $\pi(k)$, who is ordered kth, receives $w(S(\pi, k)) - w(S(\pi, k-1))$, the difference between the value that the first k agents can generate and the value that the first $k-1$ agents can generate.

A drawback of the marginal contribution solution concept is that it requires us to choose some order π, and this order can have a significant impact on the values received by individual agents. For example, suppose that two agents are substitutable, in the sense that having either one of them in a coalition generates a significant amount of value, but having both of them generates hardly more value than just having one. Then, each of these two agents would strongly prefer to be earlier in the order, where she can still make a difference. The Shapley value gives a fair solution to this problem: it simply averages the marginal contribution value *over all possible orders*. That is, agent a receives $\frac{1}{|\Pi|} \sum_\pi w(S(\pi, \pi^{-1}(a))) - w(S(\pi, \pi^{-1}(a) - 1))$ under the Shapley value, where $|\Pi|$ is the number of possible orders of the agents.

However, it turns out that the Shapley value is vulnerable to a type of false-name manipulation (Yokoo et al. 2005). To see why, we first consider the fact that the reason that an agent is useful to a coalition is that she brings certain resources to the coalition. Letting R be the set of all possible resources, we can define a characteristic function directly over subsets of these resources, $v : 2^R \rightarrow \mathbb{R}$. If we know that agent a owns resources R_a, then we can rederive the characteristic function over subsets of agents from this: the value of a coalition is simply the value of all the resources they possess, $w(S) = v(\bigcup_{a \in S} R_a)$.

Now, consider a situation where there are three resources, $\{A, B, C\}$, and all these resources are necessary to generate any value: $v(\{A, B, C\}) = 1$ and $v(S) = 0$ for $S \subsetneq \{A, B, C\}$. Also, suppose that there are two agents: agent 1 owns resource C and agent 2 owns resources A and B. It is straightforward to calculate that the Shapley value of each agent is $1/2$. However, now suppose that 2 pretends to be two agents instead: $2a$ who owns resource A, and $2b$ who owns resource B. Then, the Shapley value of each identifier is $1/3$. Because agent 2 controls two of these identifiers, she obtains a total value of $2/3$, greater than the $1/2$ that she would have obtained without false-name manipulation. (This example was given by Yokoo et al. (2005). This type of manipulation has also been studied in the context of weighted voting games (Bachrach and Elkind 2008).)

This leads to the question of whether there are good solution concepts in this context that are not vulnerable to this type of manipulation. Because we have a characteristic function v that is defined over subsets of the resources rather than the agents, a natural idea is to distribute payoffs to the resources, instead of to the agents. Then, an agent receives the payoffs of all of the resources she owns. For example, if we apply the idea of the Shapley value to the resources directly, then each of the resources A, B, and C receives $1/3$, regardless of who owns them. This immediately prevents the type of false-name manipulation discussed above: distributing one's resources over multiple identifiers does not affect how much these resources will receive.

Unfortunately, distributing to resources instead of to agents introduces another problem, namely that an agent may wish to *hide* some of her resources. To see why, consider a different function v, namely one for which $v(\{A, B, C\}) = v(\{A, C\}) = v(\{B, C\}) = 1$, and $v(S) = 0$ for all other S. A straightforward calculation shows that the Shapley value (applied to resources) gives $1/6$ to each of A and B. Now, consider a situation where agent 1 owns resource C, and agent 2 owns resources A and B. If agent 2 hides resource A (but reports resource B), then the restriction of v on the reported resources is $v(\{B, C\}) = 1$ and $v(S) = 0$ for all $S \subsetneq \{B, C\}$. Hence, the Shapley value distributes $1/2$ to B, which is more than the $1/6 + 1/6 = 1/3$ that agent 2 would have received without hiding A.

Based on these ideas, Yokoo et al. (2005) define the *anonymity-proof core*, which is robust to these manipulations. Ohta et al. (2006) give a compact representation of outcome functions in the anonymity-proof core, and also introduce a concept called the *anonymity-proof nucleolus*. Finally, Ohta et al. (2008) introduce the *anonymity-proof Shapley value*, based on the concept of the Shapley value discussed above.

Conclusion

As we have seen, the basic notion of false-name-proofness allows for useful mechanisms under certain circumstances, but in general there are impossibility results that show that false-name-proof mechanisms have severe limitations. One may react to these impossibility results by saying that, since false-name-proof mechanisms are unsatisfactory, we should not run any important mechanisms in highly anonymous settings—unless, perhaps, we can find some methodology that directly prevents false-name manipulation even in such settings, so that we are back in a more typical mechanism design context.

However, it seems unlikely that the phenomenon of false-name manipulation will disappear anytime soon. Because the Internet is so attractive as a platform for running certain types of mechanisms, it seems unlikely that the organizations running these mechanisms will take them offline. Moreover, because a goal of these organizations is often to get as many users to participate as possible, they will be reluctant to use high-overhead solutions that discourage users from participating. As a result, perhaps the most promising approaches at this point are those that combine techniques from mechanism design with other techniques, as discussed towards the end of this article. It appears that this is a rich

domain for new, creative approaches that can have significant practical impact.

Acknowledgments

We wish to thank our many collaborators on these topics, including Mingyu Guo, Nicole Immorlica, Atsushi Iwasaki, Joshua Letchford, Kohki Maruono, Shigeo Matsubara, Kamesh Munagala, Naoki Ohta, Yoshifusa Omori, Yuko Sakurai, Tuomas Sandholm, Yasufumi Satoh, Takayuki Suyama, Kenji Terada, Taiki Todo, Liad Wagman, and Xiaowei Yang. Conitzer was supported by NSF under award number IIS-0812113 and CAREER 0953756, and by an Alfred P. Sloan Research Fellowship. Yokoo was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (A), 20240015.

References

Bachrach, Y., and Elkind, E. 2008. Divide and conquer: False-name manipulations in weighted voting games. *AAMAS*, 975–982.

Bikhchandani, S.; Chatterji, S.; Lavi, R.; Mu'alem, A.; Nisan, N.; and Sen, A. 2006. Weak monotonicity characterizes deterministic dominant strategy implementation. *Econometrica* 74(4):1109–1132.

Black, D. 1948. On the rationale of group decision-making. *Journal of Political Economy* 56(1):23–34.

Clarke, E. H. 1971. Multipart pricing of public goods. *Public Choice* 11:17–33.

Conitzer, V.; Immorlica, N.; Letchford, J.; Munagala, K.; and Wagman, L. 2010. False-name-proofness in social networks. Draft.

Conitzer, V. 2007. Limited verification of identities to induce false-name-proofness. *TARK*, 102–111.

Conitzer, V. 2008a. Anonymity-proof voting rules. *WINE*, 295–306.

Conitzer, V. 2008b. Using a memory test to limit a user to one account. *AMEC*.

Conitzer, V. 2010. Making decisions based on the preferences of multiple agents. *CACM* 53(3):84–94.

Dasgupta, P.; Hammond, P.; and Maskin, E. 1979. The implementation of social choice rules: some general results on incentive compatibility. *Review of Economic Studies* 46(2):185–216.

Douceur, J. R. 2002. The Sybil attack. *First International Workshop on Peer-to-Peer Systems*, 251–260.

Dwoskin, E. 2007. Vote for Christ. *Newsweek*. http://www.newsweek.com/id/33187.

Gibbard, A. 1973. Manipulation of voting schemes: a general result. *Econometrica* 41:587–601.

Gibbard, A. 1977. Manipulation of schemes that mix voting with chance. *Econometrica* 45:665–681.

Green, J., and Laffont, J.-J. 1977. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica* 45:427–438.

Guo, M., and Conitzer, V. 2010. False-name-proofness with bid withdrawal. *AAMAS* (short paper).

Iwasaki, A.; Conitzer, V.; Guo, M.; Todo, T.; Omori, Y.; Sakurai, Y.; and Yokoo, M. 2010. Worst-case efficiency ratio in false-name-proof combinatorial auction mechanisms. *AAMAS*.

Iwasaki, A.; Yokoo, M.; and Terada, K. 2005. A robust open ascending-price multi-unit auction protocol against false-name bids. *Decision Support Systems* 39:23–39.

Moulin, H. 1980. On strategy-proofness and single peakedness. *Public Choice* 35(4):437–455.

Müller, R. 2006. Tractable cases of the winner determination problem. In Cramton, P.; Shoham, Y.; and Steinberg, R., eds., *Combinatorial Auctions*. MIT Press. 319–336.

Myerson, R. 1979. Incentive compatibility and the bargaining problem. *Econometrica* 41(1).

Nagamochi, H.; Ishii, T.; and Ito, H. 1993. Minimum cost source location problem with vertex-connectivity requirements in digraphs. *Information Processing Letters* 80(6):287–293.

Ohta, N.; Iwasaki, A.; Yokoo, M.; Maruono, K.; Conitzer, V.; and Sandholm, T. 2006. A compact representation scheme for coalitional games in open anonymous environments. *AAAI*, 697–702.

Ohta, N.; Conitzer, V.; Satoh, Y.; Iwasaki, A.; and Yokoo, M. 2008. Anonymity-proof Shapley value: Extending Shapley value for coalitional games in open environments. *AAMAS*, 927–934.

Rastegari, B.; Condon, A.; and Leyton-Brown, K. 2007. Revenue monotonicity in combinatorial auctions. *AAAI*, 122–127.

Sakurai, Y., and Yokoo, M. 2002. An average-case budget-non-negative double auction protocol. *AAMAS*, 104–111.

Sakurai, Y., and Yokoo, M. 2003. A false-name-proof double auction protocol for arbitrary evaluation values. *AAMAS*, 329–336.

Shapley, L. S. 1953. A value for n-person games. In Kuhn, H. W., and Tucker, A. W., eds., *Contributions to the Theory of Games*, volume 2 of *Annals of Mathematics Studies, 28*. Princeton University Press. 307–317.

Suyama, T., and Yokoo, M. 2005. Strategy/false-name proof protocols for combinatorial multi-attribute procurement auction. *JAAMAS* 11(1):7–21.

Terada, K., and Yokoo, M. 2003. False-name-proof multi-unit auction protocol utilizing greedy allocation based on approximate evaluation values. *AAMAS*, 337–344.

Todo, T.; Iwasaki, A.; Yokoo, M.; and Sakurai, Y. 2009. Characterizing false-name-proof allocation rules in combinatorial auctions. *AAMAS*, 265–272.

Vickrey, W. 1961. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance* 16:8–37.

von Ahn, L.; Blum, M.; Hopper, N.; and Langford, J. 2003. CAPTCHA: Using hard AI problems for security. *EUROCRYPT*, 294–311.

von Ahn, L.; Blum, M.; and Langford, J. 2004. Telling humans and computers apart automatically: How lazy cryptographers do AI. *CACM* 47(2):56–60.

Wagman, L., and Conitzer, V. 2008. Optimal false-name-proof voting rules with costly voting. *AAAI*, 190–195.

Yokoo, M.; Conitzer, V.; Sandholm, T.; Ohta, N.; and Iwasaki, A. 2005. Coalitional games in open anonymous environments. *AAAI*, 509–514.

Yokoo, M.; Sakurai, Y.; and Matsubara, S. 2001a. Robust combinatorial auction protocol against false-name bids. *AIJ* 130(2):167–181.

Yokoo, M.; Sakurai, Y.; and Matsubara, S. 2001b. Robust multi-unit auction protocol against false-name bids. *IJCAI*, 1089–1094.

Yokoo, M.; Sakurai, Y.; and Matsubara, S. 2004. The effect of false-name bids in combinatorial auctions: New fraud in Internet auctions. *Games and Economic Behavior* 46(1):174–188.

Yokoo, M.; Sakurai, Y.; and Matsubara, S. 2005. Robust double auction protocol against false-name bids. *Decision Support Systems* 39:23–39.

Yokoo, M. 2003. The characterization of strategy/false-name proof combinatorial auction protocols: Price-oriented, rationing-free protocol. *IJCAI*, 733–742.

Yu, H.; Kaminsky, M.; Gibbons, P. B.; and Flaxman, A. 2008. SybilGuard: Defending against sybil attacks via social networks. *IEEE/ACM Transactions on Networking (ToN)* 16(3):576–589.

Yu, H.; Gibbons, P. B.; Kaminsky, M.; and Xiao, F. 2010. SybilLimit: A near-optimal social network defense against sybil attacks. *IEEE/ACM Transactions on Networking (ToN)*. To appear.