Computational Criticisms of the Revelation Principle[*]

Vincent Conitzer[†] Tuomas Sandholm[†]

March 31, 2004

Abstract

The *revelation principle* is a cornerstone tool in mechanism design. It states that one can restrict attention, without loss in the designer's objective, to mechanisms in which A) the agents report their types completely in a single step up front, and B) the agents are motivated to be truthful. We show that reasonable constraints on computation and communication can invalidate the revelation principle. Regarding A, we show that by moving to multi-step mechanisms, one can reduce exponential communication and computation to linear—thereby answering a recognized important open question in mechanism design. Regarding B, we criticize the focus on truthful mechanisms—a dogma that has, to our knowledge, never been criticized before (besides on the basis of privacy reasons). First, we study settings where the optimal truthful mechanism is $\mathcal{NP}$-complete to execute for the center. In that setting we show that by moving to insincere mechanisms, one can shift the burden of having to solve the $\mathcal{NP}$-complete problem from the center to one of the agents. Second, we study a new oracle model that captures the setting where utility values can be hard to compute even when all the pertinent information is available—a situation that occurs in many practical applications. In this model we show that by moving to insincere mechanisms, one can shift the burden of having to ask the oracle an exponential number of costly queries from the center to one of the agents. In both cases the insincere mechanism is equally good as the optimal truthful mechanism in the presence of unlimited computation. More interestingly, whereas being unable to carry out either difficult task would have hurt the center in achieving his objective in the truthful setting, if the agent is unable to carry out either difficult task, the value of the center's objective *strictly improves*.

1 Introduction

Systems, especially on the Internet, are increasingly being used by multiple self-interested parties with different preferences. The coordination of these agents is of key importance, but one cannot assume that they will behave in a way that is desirable systemwide. Rather, they will act in their own interest. Mechanism design, a subfield of game theory, deals with designing the rules of the game (aka. a *mechanism*) so that a good systemwide outcome will be achieved despite the fact that the agents act based on self-interest.

The *revelation principle* is a cornerstone tool in mechanism design. It states that one can restrict attention, without loss in the designer's objective, to mechanisms in which A) the agents report their types completely in a single step up front, and B) the agents are motivated to be truthful. In settings where computation and communication are free and unlimited, the argument for the revelation principle is valid.

However, in this paper we show that reasonable constraints on computation and communication can invalidate the revelation principle. We separate the two prescriptions (A and B) of the revelation principle, and show how each of them can fail. We address A in Section 2, and B in Section 3.

1.1 Mechanism design: Definitions

In the framework of mechanism design, there is a *center* to whom the agents reveal information. The center makes sure that the rules of the game (e.g. rules of an auction) are followed, and in the end imposes an outcome based on how the agents played the game. We now define the setting formally.

[*]The material in this paper is based upon work supported by the National Science Foundation under CAREER Award IRI-9703122, Grant IIS-9800994, ITR IIS-0081246, and ITR IIS-0121678.

[†]Carnegie Mellon University, Computer Science Department, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA, {conitzer,sandholm}@cs.cmu.edu

Definition 1. *A* preference aggregation setting *consists of a set of outcomes O,*[1] *a set of agents A with $|A| = N$, and for each agent: A set of* types Θ_i*; A probability distribution p_i over Θ_i; A utility function $u_i : \Theta_i \times O \to \Re$.* [2]

Though this follows standard game theory notation, the fact that agents have both utility functions and types is perhaps confusing. The types encode the various possible preferences that agents may turn out to have, and the agents' types are not known by the center. The utility functions are common knowledge, but the agent's type is a parameter in the agent's utility function. So, the utility of agent i is $u_i(\theta_i, o)$, where $o \in O$ is the outcome and θ_i is the agent's type.

The *mechanism designer* now has the choice between various *game forms with consequences.* This choice will decide which actions are available to the agents in the game, and how their actions map to an outcome.

Definition 2. *A* strategic game form with consequences *consists of a set of actions A_i for each agent, and an outcome function $o : A_1 \times A_2 \times \ldots \times A_N \to O$.*[3] *Of particular interest are* direct-revelation games*, where for each agent i, $A_i = \Theta_i$.*

Once the game is chosen, each agent will adopt a *strategy.*

Definition 3. *A* strategy *for agent i is a function $s_i : \Theta_i \to A_i$.*[4]

A *solution concept* indicates which vectors of strategies are strategically stable. We discuss the two most common solution concepts: implementation in *dominant strategies*, and implementation in *Bayes-Nash equilibrium.* We use the following standard notation. We write (for example) a_{-i} for the vector of all players' actions besides i's; and we write (for example) (a, a_{-i}) for $(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_N)$. We also use the notation $E_{\theta \leftarrow p}[]$ to indicate that the expectation is taken over the probability distribution p for θ. That is, θ is *drawn from* p.

Definition 4. *The vector of strategies $(s_1, \ldots, s_N)$ is a* dominant strategy equilibrium *if for every agent i, for every type $\theta_i \in \Theta_i$, every alternative action $a_i \in A_i$, and every action vector $a_{-i} \in A_{-i}$ of the other agents, we have $u_i(\theta_i, o(s_i(\theta_i), a_{-i})) \geq u_i(\theta_i, o(a_i, a_{-i}))$. In this case, we say that the game form (or the mechanism)* implements the social choice rule *$c : \Theta_1 \times \Theta_2 \times \ldots \times \Theta_N \to O$ given by $c(\theta_1, \ldots, \theta_N) = o(s_1(\theta_1), \ldots, s_N(\theta_N))$* in dominant strategies.

Thus, in dominant strategy equilibrium, the action prescribed by one's strategy is optimal regardless of what the other agents do. If it is optimal only given the other agents' strategies, and given that one does not know the other agents' types, we have a *Bayes-Nash equilibrium.*

Definition 5. *The vector of strategies $(s_1, \ldots, s_N)$ is a* Bayes-Nash equilibrium *if for every agent i, for every type $\theta_i \in \Theta_i$, and every alternative action $a_i \in A_i$, we have $E_{\theta_{-i} \leftarrow p_{-i}}[u_i(\theta_i, o(s_i(\theta_i), s_{-i}(\theta_{-i})))] \geq$ $E_{\theta_{-i} \leftarrow p_{-i}}[u_i(\theta_i, o(a_i, s_{-i}(\theta_{-i})))]$. In this case, we say that the game form (or the mechanism) implements the social choice rule $c : \Theta_1 \times \Theta_2 \times \ldots \times \Theta_N \to O$ given by $c(\theta_1, \ldots, \theta_N) = o(s_1(\theta_1), \ldots, s_N(\theta_N))$* in Bayes-Nash equilibrium.

Given the preference aggregation setting, the space of possible game forms, and the desired solution concept, the mechanism designer attempts to choose the game form so as to maximize the expected value of some *objective.* The most studied objective is *social welfare*, which is simply $\sum_{i=1}^{N} u_i(\theta_i, o)$.

1.2 The revelation principle

The revelation principle states that in designing mechanisms, we only need to consider direct-revelation games, where each agent reports his type directly (completely and in a single step up front). Additionally, we only need to consider those games in which every agent reveals his type *truthfully* in equilibrium. Roughly, the argument

[1]Sometimes, it is possible for the agents to make side payments. If so, each outcome includes a specification of how much each agent pays or receives. The results of this paper do not rely on side payments.

[2]The revelation principle, discussed shortly, also applies in more general settings, such as when the types are correlated, as well as when the agents observe different signals and each agent's utility depends on the others' signals too. However, all of our results go through even in the simple setting defined above (aka. the independent private types model).

[3]If the function o instead produces a *probability distribution* over outcomes, we say the mechanism is *randomized.* In this paper we only discuss deterministic mechanisms.

[4]If an agent randomizes over strategies, he is said to use a *mixed strategy.* We present our subsequent definitions in terms of pure strategies, but they are easily generalized to mixed strategies.

is as follows. We show that given any mechanism, we can construct a truthful direct-revelation mechanism whose performance is identical. Given a mechanism, we can build an interface layer between the agents and this mechanism. The agents input (some report of) their types into the interface layer; subsequently, the interface layer inputs the actions *that the agents would have strategically played* if their types were as declared, into the original mechanism, and the resulting outcome is the outcome of the new mechanism. Since the interface layer acts "strategically on each agent's best behalf", there is never an incentive to report falsely to the interface layer. Hence, the actions played by the interface layer are the actions that would have been played without the interface layer, so the results are exactly as they would have been with the original mechanism. We now state the revelation principle formally for both of the solution concepts under discussion.

Revelation Principle, version 1 (Known). *Suppose there is a strategic game form that implements a social choice rule $c : \Theta_1 \times \Theta_2 \times \ldots \times \Theta_N \to O$ in dominant strategies. Then there exists a direct-revelation game with outcome function o that implements c in dominant strategies, where the dominant strategies equilibrium through which it is implemented is truthful. That is, for any i and $\theta_i \in \Theta_i$, $s_i(\theta_i) = \theta_i$, which is a dominant strategy, and $o(\theta_1, \ldots, \theta_N) = c(\theta_1, \ldots, \theta_N)$.*

Revelation Principle, version 2 (Known). *Suppose there is a strategic game form that implements a social choice rule $c : \Theta_1 \times \Theta_2 \times \ldots \times \Theta_N \to O$ in Bayes-Nash equilibrium. Then there exists a direct-revelation game with outcome function o that implements c in Bayes-Nash equilibrium, where the Bayes-Nash equilibrium through which it is implemented is truthful. That is, for any i and $\theta_i \in \Theta_i$, $s_i(\theta_i) = \theta_i$, where these strategies constitute a Bayes-Nash equilibrium, and $o(\theta_1, \ldots, \theta_N) = c(\theta_1, \ldots, \theta_N)$.*

For the interested reader, we present the (known) proofs of these two versions of the revelation principle in an appendix.

We are now ready to begin our computational critique of the revelation principle.

2 Single-step vs. multi-step mechanisms

The revelation principle suggests that the designer can, without loss, restrict attention to mechanisms where each agent reveals his preferences completely in a single step.[5] However, complete revelation of preferences is problematic in many applications in practice, due to several reasons.

First, the agents may not know their preferences *a priori*, but rather may have to spend effort in determining their preferences, for instance by computing or by gathering additional information. One important setting where this occurs in practice is an auction where a bidder needs to solve his own planning problem (what he would do with the items if he would win them in the auction) in order to determine his valuation for the items that are being auctioned [38, 39, 23, 34]. One real-world application where this occurs is reverse auctions for trucking services (e.g., [38]). In that setting, there is a buyer who wants to have a set of items (delivery tasks) handled. Each bidder (a trucking company) can bid on any combinations of tasks. Each bid states how cheaply the bidder is willing to handle that combination of tasks. If the auction is conducted using *Vickrey-Clarke-Groves (VCG)* pricing (aka. the *generalized Vickrey auction*), and the bidders have quasilinear utility functions, then each bidder is motivated to bid his true cost of handling the combination of tasks [43, 5, 17]. However, evaluating one's cost of handling a combination of tasks involves solving an $\mathcal{NP}$-hard vehicle routing and scheduling problem (and furthermore, the number of combinations of tasks to consider is exponential in the number of tasks).

Second, communicating complete preferences may waste bandwidth and is impossible in many practical cases. For example, in a combinatorial auction, a bidder may need to bid on every bundle of goods to express his preferences, and the number of bundles to bid on is exponential in the number of goods for sale.

Third, complete preference revelation is undesirable from a privacy perspective. It would be more desirable to only have the agents reveal those parts of their preferences that are relevant for choosing the outcome.[6]

[5]This assumption has also been criticized on the basis that a broader class of social choice functions is implementable in *subgame perfect equilibrium* [27], which is a more sophisticated type of equilibrium that takes into account the sequential structure of the game. In this paper, we focus on simpler notions of equilibrium that do not concern themselves with the sequential structure of the game. It is for these notions of equilibrium that the revelation principle holds (assuming free and unlimited computation and communication).

[6]In some settings, the privacy issue can be mitigated by other techniques as well, such as secure function evaluation.

To address these problems, it makes sense—unlike the revelation principle would suggest—to shift to using multi-step mechanisms where the agents reveal their preferences incrementally, only on an as-needed basis. There are several different types of mechanism for accomplishing this, such as

• incremental anytime mechanisms that have some feasible outcome ready at every step of the mechanism. *Quantity tatonnement* for (resource/task) allocation problems is an example of this. There, at each iteration, the center imposes a candidate allocation on the agents, and each agent responds with how much he would pay (or would have to be paid) for his allotment in the allocation [27]. This is closely related to iterative (resource/task) reallocation among agents (e.g., [38, 1]).

• *price tatonnement* mechanisms like ascending (or descending) auctions. In these types of mechanism, the center, at every iteration, posts a price vector (prices on items or combinations of items) to each bidder, and each bidder states which combination (potentially the empty combination) the bidder prefers at that price vector [33, 35, 36, 44, 4, 28, 2, 15, 42]. The mechanism terminates in a solution where the bidders' demands, together, are feasible.

• explicit preference elicitation. In these mechanisms, the center explicitly queries the agents about specific aspects of their preferences in light of what the center has learned about the agents' preferences so far. This has been applied to combinatorial auctions [7, 9, 8], and in practice only a vanishingly small fraction of the bidders' valuation information needs to be revealed before the optimal allocation can be determined for certain [18].[7] This approach has also been applied to preference elicitation in voting [11].

In multi-stage mechanisms, the agents get signals about each others' actions so far, so the agents can condition their actions on these signals. This potentially introduces additional opportunitiues for strategic manipulation of the mechanism by the agents [11]. However, if the mechanism is social welfare maximizing (assuming truthful revelation) and VCG pricing is used, then revealing one's preferences (the part of the preferences that the mechanism asks about) truthfully is an *ex post* equilibrium [7].[8] [9]

While it is known that multi-step mechanisms can save revelation compared to single-step mechanisms, it has not been clear whether these savings can be drastic. For one, Christos Papadimitriou recently (at the DIMACS Fall 2001 workshop on Computational Issues in Game Theory and Mechanism Design) posed the question of whether multi-step mechanisms can yield an *exponential* reduction in communication/computation.[10] The following theorem shows that this is the case even in very simple settings. This clearly demonstrates the impracticality of the single-step mechanisms advocated by the revelation principle.

Theorem 1. *There exist preference aggregation settings (even when the objective is social welfare maximization, there are only 2 agents, and the agents' types are private and drawn independently), where*

- *each optimal single-step mechanism requires the communication of an exponential number of bits (and thus exponential computation by the center to receive these bits) for every type vector, and*
- *there exists a multi-step (2-step) mechanism that implements the same social choice rule, requires only a linear number of bits to be communicated, and uses only a linear amount of computation at the center.*

This holds both for dominant strategy implementation and for Bayes-Nash implementation.

Proof. Consider a setting where the outcome is a string of $n+1$ bits, so $|O| = 2^{n+1}$. Let there be 2 agents. Agent 1's type θ_1 is a string of n bits, so $|\Theta_1| = 2^n$. The utility of agent 1 is $u_1(\theta_1, o) = 2$ if the first n bits of the outcome o agree with θ_1, and 0 otherwise. Agent 2's type θ_2 is a mapping from the set of all n-bit strings to a bit, that is $\theta_2 : \{0,1\}^n \to \{0,1\}$. The interpretation is that agent 2's type defines how agent 2 wants the last bit of the outcome set based on how the first n bits of the outcome are set. Define $prefix_n(s)$ to be the string

[7]This is despite the fact that in the worst case, to determine an (even approximately) optimal allocation in a combinatorial auction requires exponential communication [32].

[8]*Ex post* equilibrium is a game-theoretic solution concept that is strictly stronger than Nash equilibrium and strictly weaker than dominant strategy implementation. In short, a strategy profile is in *ex post* equilibrium if it is a Nash equilibrium *for any prior*. The claim that we have an *ex post* equilibrium here relies on the fact that in the direct-revelation VCG mechanism, truthful revelation is a dominant strategy, that is, an optimal strategy no matter what types the other agents reveal. So, if in the equilibrium of the multi-step mechanism the other agents' strategies are such that they always report the same type no matter what one does, then truthful revelation is one's optimal strategy. This holds for every agent. Therefore truthful revelation is in equilibrium (for any prior).

[9]In some games of this type, there may be additional equilibria.

[10]In the theory of communication complexity (in a non-game-theoretic setting), it has already been shown that an exponential communication gap can exist between the best single-step and the best multi-step communication protocol [20].

of the first n bits of bit string s, and define $last(s)$ to be the last bit of s. The utility of agent 2 is $u_2(\theta_2, o) = 1$ if $\theta(prefix_n(o)) = last(o)$, and 0 otherwise. As usual, the utility functions are known by the center as well, but the types are privately known by the agents.

Let the mechanism's objective be the maximization of social welfare, i.e., $\sum_{i \in \{1,2\}} u_i(\theta_i, o)$. The unique optimal direct-revelation mechanism for this game elicits θ_1 (that is, n bits) from agent 1 and θ_2 (that is, 2^n bits) from agent 2, and chooses the outcome $(\theta_1, \theta_2(\theta_1))$. The outcome always maximizes social welfare.

It is easy to see that in this mechanism, each agent's dominant strategy is to reveal his type truthfully. Agent 1 unilaterally determines all of the aspects of the outcome that he cares about (that is, the first n bits of the outcome), so his dominant strategy is to reveal θ_1 truthfully. Agent 2 cannot affect the first n bits in any way, and unilaterally determines the last bit of the outcome (conditional on what agent 1 does), so so his dominant strategy is to reveal θ_2 truthfully.

Now, any single-step mechanism that is guaranteed to find the social welfare maximizing outcome will have to elicit this much information (θ_1 and θ_2), in the following sense. If the mechanism cannot distinguish whether agent 1's type is θ_1 or θ_1', then the mechanism does not know how to set the first n bits of the outcome. On the other hand, suppose the mechanism cannot distinguish whether agent 2's type is θ_2 or θ_2'. If $\theta_2 \neq \theta_2'$, then there exists an n-bit string σ such that $\theta_2(\sigma) \neq \theta_2'(\sigma)$, and if it happens that $\theta_1 = \sigma$, then the mechanism does not know how to set the last bit of the outcome.

Now, consider a 2-step mechanism. In the first step the mechanism elicits θ_1 (i.e., n bits). In the second step, the mechanism elicits $\theta_2(\theta_1)$ (i.e., 1 bit). The outcome again is $(\theta_1, \theta_2(\theta_1))$. For the same reasons as in the single-step mechanism, each agent's dominant strategy is to reveal his type truthfully. So, this multi-step mechanism implements the same social choice rule as the single-step mechanism, but only elicits a linear number of bits $(n + 1)$ instead of the exponential number required by the single-step mechanism $(n + 2^n)$. $\square$

3 Truthful vs. insincere mechanisms

The fact that multi-step mechanisms have computational and communication advantages over single-step mechanisms has been observed before and explored to a certain extent already, as discussed above. However, the revelation principle has another questionable facet: it states that restricting attention to truthful mechanisms comes at no loss. In this section we show that, interestingly, under limited computational resources, this restriction *does* incur a loss.

3.1 Computational complexity in truthful vs. insincere mechanisms

In many real-world mechanism design settings, the center faces an intractable optimization problem in trying to execute the mechanism. For example, the problem of determining the winners of a combinatorial auction is $\mathcal{NP}$-complete [37] and inapproximable [40]. Recently there has been a surge of research in developing faster algorithms for optimally executing mechanisms, e.g., winner determination algorithms for combinatorial auctions (e.g., [37, 40, 16, 29, 25, 41]). There has also been considerable recent work (called *algorithmic mechanism design*) on designing mechanisms that 1) can be executed with polynomial effort, 2) yield an outcome that is provably within a bound from optimal, and 3) where the agents have incentive to act truthfully (see, e.g., [30, 31, 26]). A third interesting avenue of research in this area is *automated mechanism design*, where the mechanism is generated automatically (i.e., *designed* computationally) for the setting at hand [10]. Each of these three strands of research follows the revelation principle's prescription that attention should be restricted to truthful mechanisms.

In this subsection, we question the focus on truthful mechanisms when the setting requires the solution of computationally hard problems. In particular, we show that there are settings where by abandoning truthful mechanisms, we can shift a computationally hard problem from the center to one of the agents. Additionally, whereas not being able to cope with the issue of computational hardness would have hurt the center in achieving his objective, if the agent is unable to cope with it, this actually helps the designer in achieving his objective.

We first observe that dominant strategy implementation and Bayes-Nash implementation differ only on what agents can be expected to know about each others' types and actions. An interesting special case is games where only one agent needs to choose an action. In this case, the acting agent always knows everything there is to know about the other agent's actions (namely, nothing). So, both solution concepts coincide here. We prove the remaining two theorems for this types of game, so the results hold both for dominant strategy implementation and Bayes-Nash implementation.

Theorem 2. *Suppose that the center is trying to maximize social welfare, and neither payments nor randomization is allowed. Then, even with only two agents (one of whom does not even report a type, so dominant strategy implementation and Bayes-Nash implementation coincide), there exists a family of preference aggregation settings such that:*

- *the execution of any optimal truthful mechanism is $\mathcal{NP}$-complete for the center, and*
- *there exists an insincere mechanism which 1) requires the center to carry out only polynomial computation, and 2) makes finding any beneficial insincere revelation $\mathcal{NP}$-complete for the type-reporting agent. Additionally, if the type-reporting agent manages to find a beneficial insincere revelation, or no beneficial insincere revelation exists, the social welfare of the outcome is identical to the social welfare that would be produced by any optimal truthful mechanism. Finally, if the type-reporting agent does not manage to find a beneficial insincere revelation where one exists, the social welfare of the outcome is* strictly greater *than the social welfare that would be produced by any optimal truthful mechanism.*

Put in perspective, the mechanism designer would reap two benefits from using the second, insincere mechanism rather than a truthful mechanism:

1. Doing so shifts the computational hardness from the center to the agent. This can also be seen as a statement about how the social welfare that can be obtained by truthful mechanisms compares to the social welfare that can be obtained by insincere mechanisms, as follows. If it is computationally infeasible to execute the optimal truthful mechanism, the designer might resort to another truthful mechanism which merely approximates the social welfare obtained by the optimal truthful mechanism (this is exactly the approach taken in algorithmic mechanism design).
2. If the agent cannot consistently solve instances of an $\mathcal{NP}$-complete problem, then, even if the agent is trying to act strategically, using the second mechanism improves social welfare in some cases (and never decreases it).

Hence, (by the second argument) the insincere mechanism—which is computationally feasible to execute—outperforms the optimal truthful mechanism, which (by the first argument) in turn outperforms any computationally feasible truthful mechanism.

We now present the proof.

Proof. We present the family of settings in "story form" to improve readability; the presentation is easily transcribed into formal notation. The head of an organization has to decide on a team of k employees to send on a project. n employees are available. In deciding on a team, the head of organization is trying to aggregate the preferences of two parties: the job manager of the project (who is already known), and the head of recruiting. (Note that, in this example, we are not interested in the preferences of the employees who may actually be sent on the project.) Some of the employees are "old friends"; this relationship can be represented in a graph, where the vertices represent the employees, and an edge is drawn between two vertices if the employees they represent are old friends. (This graph is common knowledge.) The only interest that the head of recruiting has in the selection of the team is that she would like at least two members of the team to be old friends, because this will make for a nice story in the next recruiting brochure. She gets a utility of 2 if there is a pair of old friends in the team, and a utility of 0 otherwise. (This is common knowledge, so the head of recruiting need not report a type.) The job manager, on the other hand, prefers it if there are no old friends in the team, because he feels that this will lead to an unprofessional atmosphere. He gets a utility of 0 if there is a pair of old friends in the team, and a utility of 1 otherwise. Again, this is common knowledge. However, the job manager may also have detailed knowledge about which team of employees would collectively have a skill set appropriate for the project. If this is the case, and this exact team of employees is selected to go on the project (whether this team has old friends in it or not), the job manager gets an additional utility of 3. Whether the job manager believes a particular team is ideal, and if so, which team it is, is not common knowledge. Hence, this information needs to be reported by the job manager; and thus, the job manager has a type set of size $\binom{n}{k} + 1$ (one type for each possible team, indicating that the job manager believes that this team is ideal for the project, and one additional type indicating that the job manager does not believe any particular team is ideal). Let the probability distribution over this type set be uniform. Now let us consider the problem of creating a mechanism for such a setting that uses neither payments nor randomization.

First, we claim that all optimal *truthful* mechanisms are of the following form.

- If the job manager reports a type corresponding to a particular team, then choose this exact team;
- If the job manager reports the type indicating that he does not believe that any particular team is ideal, then 1) if a team without any old friends exists (corresponding to an independent set of size k in the graph), choose such a team; or 2) if no such team exists, choose any team (it does not matter which).

It is straightforward to verify that mechanisms of this form act in the job manager's best interest, i.e., they always choose one of the outcomes that are optimal for the job manager given his type. Hence, the job manager never has any incentive to misreport his type, so these mechanisms are truthful. All that remains to show is that all other truthful mechanisms have strictly less expected social welfare than these. We first observe that the only case in which we get less than the optimal social welfare with the mechanisms of the given form is when the job manager has the type that corresponds to no team, and an independent set of size k exists. In this case, the mechanisms of the given form choose an independent set as the team, leading to a social welfare of 1; whereas a social welfare of 2 could have been obtained by choosing a team with friends in it. It follows that the expected social welfare that we get from one of the mechanisms in the given form is at most $\frac{1}{\binom{n}{k}+1}$ below the maximal expected social welfare that we could have obtained if the agents did not play strategically. Now consider an alternative truthful mechanism that, for some team, does not choose this team when the job manager reports the type corresponding to that team. In this case, this mechanism can obtain a social welfare of at most 2, whereas the optimal social welfare in this case is at least 4. It follows that the expected social welfare we get from this mechanism is at least $\frac{2}{\binom{n}{k}+1}$ below the maximal expected social welfare that we could have obtained if the agents did not play strategically. Hence, all optimal truthful mechanisms always choose the team corresponding to the type that the job manager reports, when the type corresponds to a team. But then, if an independent set in the friendship graph exists, an optimal truthful mechanism must choose such an independent set in the case where the job manager reports the type corresponding to no team: because if it does not, then when the job manager has this type, he would benefit from misreporting his type as a type corresponding to the independent set—and the mechanism would no longer be truthful. Thus, we have established that all optimal truthful mechanisms are of the given form. We observe that executing such a mechanism requires solving an $\mathcal{NP}$-complete problem, because we have to construct an independent set if it exists.

Now consider the following mechanism:

- If the job manager reports a type corresponding to a particular team, then choose this exact team;
- If the job manager reports the no-team type, then choose some team with at least a pair of friends.

We observe that this mechanism is computationally easy to execute. Also, this mechanism is not truthful if there is an independent set, because in this case, if the job manager has the type corresponding to no team, the job manager would be better off reporting the type corresponding to the independent set. However, there are no other beneficial insincere revelations. Thus, it is straightforward to verify that if the job manager always reports the type that is strategically optimal for him, the outcome of this mechanism is always identical to that of one of the optimal truthful mechanisms. Of course, in order for the job manager to always report the type that is strategically optimal for him, he needs to construct an independent set (if possible) when he has the type corresponding to no team. Because this problem is $\mathcal{NP}$-complete, it is reasonable to suspect that the job manager will not always be able to construct such a set even when it exists. If the job manager indeed fails to construct an independent set in this case, the outcome will be a team with at least a pair of friends. This outcome actually has a social welfare of 2, as opposed to the social welfare of 1 that would have been obtained if the job manager had managed to construct an independent set. Hence the social welfare is strictly greater than in the case where the job manager has unlimited computational power; and hence it is also a greater than it would have been with an optimal truthful mechanism. $\square$

3.2 Valuation complexity in truthful vs. insincere mechanisms

In this subsection, we question the focus on truthful mechanisms when determining valuations is hard. We focus on the following abstract model of this. Suppose that there is a commonly accessible *oracle* which, when supplied with an agent, that agent's type, and an outcome, returns a utility value for that agent. This oracle is the only available means for determining agents' utilities for outcomes. Depending on the supplied type, the query may be costless, or carry a constant (computational) cost.[11]

We first provide two settings where this model is realistic.

Example 1. Suppose we are trying to allocate delivery tasks to various shipping companies. In this case, a shipping company's type (private information) consists of the resources available to this company. An outcome consists of an allocation of the delivery tasks to the companies. Even when we know both the company's type and the outcome selected by the center, in order to compute the company's cost for this allocation, we need to compute the optimal delivery schedule for the company given its type. This may or may not be difficult depending on the resources available to the company. For instance, if the company only has ground-based vehicles available, this may require solving a computationally hard routing problem, which is to be solved by some software package (an "oracle"). Running this software package may be expensive. On the other hand, the company may have a helicopter available to it whose flight time between any two locations is negligible, and the only cost it incurs is a constant cost per takeoff. In this case, the cost to the company of an allocation of tasks to it is simply proportional to the number of tasks in the allocation, which is trivial to compute. Hence there is no cost to this computation.

Example 2. Suppose we are trying to sell a piece of art. In this case, a bidder's private information may include whether she is an art trader, or an art collector. In the former case, to determine her valuation for an outcome (which indicates whether she won the piece of art or not), she needs to query an (expensive) expert about the authenticity of the painting. In the latter case, she simply has an intrinsic valuation for the piece of art, so that there is no cost to her in evaluating how much the piece of art would be worth to her.

We now show that there are settings where by abandoning truthful mechanisms, one can transfer the burden of having to make an exponential number of costly oracle queries from the center to the agent. Additionally, whereas not being able to cope with having to make exponentially many costly queries would have hurt the center in achieving his objective, if the agent is unable to cope with it, this *helps* the designer in achieving his objective.

Theorem 3. *Suppose that the center is trying to maximize social welfare, and neither payments nor randomization is allowed. Then, even with only two agents (one of whom does not even report a type, so dominant strategy implementation and Bayes-Nash implementation coincide), there exists a family of preference aggregation settings such that:*

- *the execution of any optimal truthful mechanism requires the center to make (on average) exponentially many costly queries to the oracle for some type reports, and*
- *there exists an insincere mechanism 1) which does not require the center to make any costly queries, and 2) where the agent needs to make (on average) exponentially many costly queries to the oracle to find a beneficial insincere revelation. Additionally, if the type-reporting agent manages to find a beneficial insincere revelation, or no beneficial insincere revelation exists, the social welfare of the outcome is identical to the social welfare that would be produced by any optimal truthful mechanism. Finally, if the type-reporting agent does not manage to find a beneficial insincere revelation where one exists, the social welfare of the outcome is* strictly greater *than the social welfare that would be produced by any optimal truthful mechanism.*

Proof. Let the outcome space be $X \cup \{d\}$, where $|X| = 2^{2n}$. (Here n is the length of a natural representation for the problem.)[12] The type-reporting agent (agent 1) has the following type set Θ. For each $x \in X$, there is a type θ_x which occurs with probability $\frac{1}{|X|+1}$. Any query to the oracle involving a type θ_x is costless. The utility function, to be obtained with these queries, is as follows: $u_1(\theta_x, x) = 4$; $u_1(\theta_x, y) = 0$ for all $y \neq x$. Additionally, for each subset $Y \subseteq X$, there is a type θ_Y which occurs with probability $\frac{1}{|X|+1}(\frac{1}{2^n})^{|Y|}(\frac{2^n-1}{2^n})^{|X|-|Y|}$. (That is, with probability $\frac{1}{|X|+1}$ one of these types θ_Y occurs, and given this, any given x is in Y with probability $\frac{1}{2^n}$, independently.) Any query to the oracle involving a type θ_Y has a nonzero query cost associated with it.[13] The

[11]Mechanism design with a cost of information acquisition (or equivalently, computation) to determine one's valuation has been studied before [13, 14, 12, 3, 6, 19, 24, 21, 23, 22, 39]. However, we do not assume that there is a method for equating the cost of a query with a cost in utility units; rather, we merely assume that an exponential query (computational) cost is unmanageable, as is typical in the computer science literature.

[12]For instance, a combinatorial auction with n items and 4 bidders has $4^n = 2^{2n}$ possible outcomes.

[13]We note that there is no comparison between this query cost and the utilities given by the utility functions; we are not assuming that agents have any way of trading query cost and utility off against each other.

utility function, to be obtained with these queries, is as follows: $u_1(\theta_Y, x) = 1$ for all $x \in Y$; $u_1(\theta_Y, x) = -1$ for all $x \notin Y, \in X$; $u_1(\theta_Y, d) = 0$. It is crucial to not be misled by the notation into thinking that the set Y is immediately obvious when observing that somebody's type is θ_Y. Rather, the set Y can only be determined by asking a costly query to the oracle with the type θ_Y for each $x \in X$. For instance, as in the shipping company's example, the type can be thought of as a set of resources, which implies which outcomes are favorable and which are not—but only through costly (oracle) computation. However, we do allow for immediate distinguishing between types of the form θ_x and those of the form θ_Y—this would require only a single query anyway.

Agent 2, who does not report a type, has the following utility function: $u_2(x) = 0$ for all $x \in X$; $u_2(d) = 2$. Now consider creating a mechanism for such a setting that uses neither payments nor randomization.

First, we claim that all optimal *truthful* mechanisms are of the following form.

- If agent 1 reports a type θ_x, then choose outcome x.
- If agent 1 reports a type θ_Y, then choose some outcome $x \in Y$ (unless Y is the empty set, in which case, choose d.)

It is straightforward to verify that mechanisms of this form act in agent 1's best interest, i.e., they always choose one of the outcomes that are optimal for agent 1 given his type. Hence, agent 1 never has any incentive to misreport his type, so these mechanisms are truthful. All that remains to show is that all other truthful mechanisms have strictly less expected social welfare than these. We first observe that the only case in which we get less than the optimal social welfare with the mechanisms of the given form is when agent 1 has a type θ_Y (with $Y \neq \{\}$). In this case, the mechanisms of the given form choose some $x \in Y$, leading to a social welfare of 1; whereas a social welfare of 2 could have been obtained by choosing d instead. Because such types occur with probability less than $\frac{1}{|X|+1}$, it follows that the expected social welfare that we get from one of the mechanisms in the given form is at most $\frac{1}{|X|+1}$ below the maximal expected social welfare that we could have obtained if the agents did not play strategically. Now consider an alternative truthful mechanism that for some type θ_x does not choose outcome x. In this case, this mechanism can obtain a social welfare of at most 2, whereas the optimal social welfare in this case is 4. Because this type occurs with probability $\frac{1}{|X|+1}$, it follows that the expected social welfare we get from the alternative mechanism is at least $\frac{2}{|X|+1}$ below the maximal expected social welfare that we could have obtained if the agents did not play strategically. Hence, all optimal truthful mechanisms always choose outcome x when the reported type is some θ_x. But then, if agent 1 reports a type θ_Y (with $Y \neq \{\}$), an optimal truthful mechanism must choose some $x \in Y$; because if it does not, agent 1 is better off reporting θ_x for some $x \in Y$ instead of truthfully reporting θ_Y—and the mechanism would no longer be truthful. Finally, any optimal truthful mechanism must choose d for the type $\theta_{\{\}}$, because this gives maximal social welfare in this case, and has no negative strategic effects. Thus, we have established that all optimal truthful mechanisms are of the given form. We observe that for the θ_Y types, executing such a mechanism requires on average an exponential number of (costly) queries: the only way to find some $x \in Y$ is by asking queries to the oracle. Because, given that the type is some θ_Y, each x is in the set Y with probability $\frac{1}{2^n}$ (independently), the expected number of queries necessary to find an $x \in Y$ is exponential.

Now consider the following mechanism:

- If agent 1 reports a type θ_x, then choose outcome x.
- If agent 1 reports a type θ_Y, then choose outcome d.

We observe that executing this mechanism does not require any costly queries at all, because queries with a type θ_x are costless, and we do not need to distinguish between the different θ_Y at all. The mechanism is not truthful, because if agent 1's type is some θ_Y (with $Y \neq \{\}$), then it is better off reporting some θ_x with $x \in Y$, to effect this outcome x. However, there are no other beneficial insincere revelations. Thus, it is straightforward to verify that if agent 1 always reports the type that is strategically optimal for it, the outcome of this mechanism is always identical to that of one of the optimal truthful mechanisms. Of course, in order for agent 1 to always report the type that is strategically optimal for it, in the cases where it has a type θ_Y, it needs to ask (on average) an exponential number of queries to find some $x \in Y$, so that it can report the corresponding θ_x. (Additionally, it may not be trivial to actually construct the type θ_x from x. If this is so, it only strengthens this argument. However, it is possible to argue that is should be cheap to find this θ_x because queries involving a θ_x are costless.) It is reaonable to suspect that agent 1 will not always be able or willing to ask this many queries. If agent 1 indeed does not manage to find some $x \in Y$, the best it can do is report θ_Y, leading to outcome d. This outcome actually has a social welfare of 2, as opposed to the social welfare of 1 that would have been obtained if agent 1 had managed to find some $x \in Y$. Hence the social welfare is strictly greater than in the case where agent 1 manages to play strategically optimallly; and hence it is also a greater than it would have been with an optimal truthful mechanism. □

4 Conclusions and future research

The *revelation principle* is a cornerstone tool in mechanism design. It states that one can restrict attention, without loss in the designer's objective, to mechanisms in which A) the agents report their types completely in a single step up front, and B) the agents are motivated to be truthful. In settings where computation and communication are free and unlimited, the argument for the revelation principle is valid.

However, in this paper we showed that reasonable constraints on computation and communication can invalidate the revelation principle. We cleanly separated the two prescriptions (A and B) of the revelation principle, and showed how each of them can fail.

Regarding A, we layed out the arguments that have been made in favor of moving to multi-step mechanisms, and presented different important families of multi-step mechanisms in a unified context. We then formally showed that by moving to multi-step mechanisms, one can reduce exponential communication and computation to linear—thereby answering a recognized important open question in mechanism design.

Regarding B, we criticized the focus on truthful mechanisms—a dogma that has, to our knowledge, never been criticized before. (Besides on the basis of privacy reasons First, we studied settings where the optimal truthful mechanism is $\mathcal{NP}$-complete to execute for the center. We showed that by moving to insincere mechanisms, one can shift the burden of having to solve the $\mathcal{NP}$-complete problem from the center to one of the agents. Second, we studied a new oracle model that captures the setting where utility values can be hard to compute even when all the pertinent information is available—a situation that occurs in many practical applications. In this model we showed that by moving to insincere mechanisms, one can shift the burden of having to ask the oracle an exponential number of costly queries from the center to one of the agents. In both cases the insincere mechanism is equally good as the optimal truthful mechanism in the presence of unlimited computation. More interestingly, whereas being unable to carry out either difficult task would have hurt the center in achieving his objective in the truthful setting (because the center would have had to opt for a suboptimal mechanism instead), if the agent is unable to carry out either difficult task, the value of the center's objective *strictly improves*.

In summary, our results suggest that there is a potentially fruitful new avenue of research on the boundary of mechanism design and computer science, where one removes the restriction to single-step mechanisms or to truthful mechanisms, or both. We have shown that in some settings, this approach can not only *reduce* computation and communication, but can also *use* such complexities to the mechanism designer's advantage.

References

[1] Martin Andersson and Tuomas Sandholm. Time-quality tradeoffs in reallocative negotiation with combinatorial contract types. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 3–10, Orlando, FL, 1999.

[2] Lawrence M Ausubel and Paul Milgrom. Ascending auctions with package bidding. *Frontiers of Theoretical Economics*, 1, 2002. No. 1, Article 1.

[3] Dirk Bergemann and Juuso Välimäki. Information acquisition and efficient mechanism design. *Econometrica*, 70:1007–1034, 2002.

[4] Sushil Bikhchandani, Sven de Vries, James Schummer, and Rakesh V. Vohra. Linear programming and Vickrey auctions, 2001. Draft.

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

[6] Olivier Compte and Philippe Jehiel. Auctions and information acquisition: Sealed-bid or dynamic formats?, 2001. Working Paper.

[7] Wolfram Conen and Tuomas Sandholm. Preference elicitation in combinatorial auctions: Extended abstract. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 256–259, Tampa, FL, October 2001.

[8] Wolfram Conen and Tuomas Sandholm. Differential-revelation VCG mechanisms for combinatorial auctions. In *AAMAS-02 workshop on Agent-Mediated Electronic Commerce (AMEC)*, Bologna, Italy, 2002.

[9] Wolfram Conen and Tuomas Sandholm. Partial-revelation VCG mechanism for combinatorial auctions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 367–372, Edmonton, Canada, 2002.

[10] Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 103–110, Edmonton, Canada, 2002.

[11] Vincent Conitzer and Tuomas Sandholm. Vote elicitation: Complexity and strategy-proofness. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 392–397, Edmonton, Canada, 2002.

[12] J Cremer and F Khalil. Gathering information before signing a contract. *American Economic Review*, v82, n3:566–578, 1992.

[13] J Cremer, F Khalil, and J Rochet. Contracts and productive information gathering. *Games and Economic Behavior*, v25, n2:174–193, 1998.

[14] J Cremer, F Khalil, and J Rochet. Strategic information gathering before a contract is offered. *Journal of Economic Theory*, v81, n1:163–200, 1998.

[15] C DeMartini, A Kwasnica, J Ledyard, and D Porter. A new and improved design for multi-object iterative auctions. Technical Report 1054, California Institute of Technology, Social Science, September 1999.

[16] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 548–553, Stockholm, Sweden, August 1999.

[17] Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[18] Benoit Hudson and Tuomas Sandholm. Effectiveness of preference elicitation in combinatorial auctions. In *AAMAS-02 workshop on Agent-Mediated Electronic Commerce (AMEC)*, Bologna, Italy, 2002.

[19] Matthew Jackson. Efficiency and information aggregation in auctions with costly information. Unpublished, April 1999.

[20] E Kushilevitz and N Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[21] Kate Larson and Tuomas Sandholm. Bargaining with limited computation: Deliberation equilibrium. *Artificial Intelligence*, 132(2):183–217, 2001.

[22] Kate Larson and Tuomas Sandholm. Computationally limited agents in auctions. In *AGENTS-01 Workshop of Agents for B2B*, pages 27–34, Montreal, Canada, May 2001.

[23] Kate Larson and Tuomas Sandholm. Costly valuation computation in auctions. In *Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 169–182, Sienna, Italy, July 2001.

[24] Kate Larson and Tuomas Sandholm. Bidders with hard valuation problems. In *International Conference on Autonomous Agents and Multi-Agent Systems*, Bologna, Italy, July 2002. (Poster presentation.) Early version: Computationally Limited Agents in Auctions. International Conference on Autonomous Agents 2001, Workshop on Agent-based Approaches to B2B, pp. 27–34, Montreal, Canada, May 28th.

[25] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 2003. Early version in ACMEC-01.

[26] Daniel Lehmann, Lidian Ita O'Callaghan, and Yoav Shoham. Truth revelation in rapid, approximately efficient combinatorial auctions. *Journal of the ACM*, 49(5):577–602, 2002. Early version appeared in ACMEC-99.

[27] Andreu Mas-Colell, Michael Whinston, and Jerry R. Green. *Microeconomic Theory.* Oxford University Press, 1995.

[28] Paul Milgrom. Putting auction theory to work: The simultaneous ascending auction. Technical report, Stanford University, Department of Economics, 1997. Revised 4/21/1999.

[29] Noam Nisan. Bidding and allocation in combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 1–12, Minneapolis, MN, 2000.

[30] Noam Nisan and Amir Ronen. Computationally feasible VCG mechanisms. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 242–252, Minneapolis, MN, 2000.

[31] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001. Early version in STOC-99.

[32] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting Lindahl prices, 2003. Working Paper (version: March 2003).

[33] David C Parkes. iBundle: An efficient ascending price bundle auction. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 148–157, Denver, CO, November 1999.

[34] David C Parkes. Optimal auction design for agents with hard valuation problems. In *Agent-Mediated Electronic Commerce Workshop at the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999.

[35] David C Parkes and Lyle Ungar. Iterative combinatorial auctions: Theory and practice. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 74–81, Austin, TX, August 2000.

[36] David C Parkes and Lyle Ungar. Preventing strategic manipulation in iterative auctions: Proxy-agents and price-adjustment. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 82–89, Austin, TX, August 2000.

[37] Michael H Rothkopf, Aleksandar Pekeč, and Ronald M Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.

[38] Tuomas Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 256–262, Washington, D.C., July 1993.

[39] Tuomas Sandholm. Issues in computational Vickrey auctions. *International Journal of Electronic Commerce*, 4(3):107–129, 2000. Special Issue on Applying Intelligent Agents for Electronic Commerce. A short, early version appeared at the Second International Conference on Multi–Agent Systems (ICMAS), pages 299–306, 1996.

[40] Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54, January 2002.

[41] Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. CABOB: A fast optimal algorithm for combinatorial auctions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1102–1108, Seattle, WA, 2001.

[42] Yoav Shoham and Moshe Tennenholtz. On rational computability and communication complexity. *Games and Economic Behavior*, 35:197–211, 2001.

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

[44] Peter R Wurman and Michael P Wellman. AkBA: A progressive, anonymous-price combinatorial auction. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 21–29, Minneapolis, MN, October 2000.

5 Appendix: Proofs of the revelation principle

Revelation Principle, version 1 (Known). *Suppose there is a strategic game form that implements a social choice rule $c : \Theta_1 \times \Theta_2 \times \ldots \times \Theta_N \rightarrow O$ in dominant strategies. Then there exists a direct-revelation game with outcome function o that implements c in dominant strategies, where the dominant strategies equilibrium through which it is implemented is truthful. That is, for any i and $\theta_i \in \Theta_i$, $s_i(\theta_i) = \theta_i$, which is a dominant strategy, and $o(\theta_1, \ldots, \theta_N) = c(\theta_1, \ldots, \theta_N)$.*

Proof. We show how to transform the given game form that implements c into a truthful direct-revelation game that implements c. For each i, let $s_i^{old} : \Theta_i \rightarrow A_i^{old}$ be the strategy played by agent i in the equilibrium that implements c in the given game, and let o^{old} be the given game's outcome function, so that $o^{old}(s_1^{old}(\theta_1), \ldots, s_N^{old}(\theta_N)) = c(\theta_1, \ldots, \theta_N)$, and the s_i^{old} constitute a dominant strategies equilibrium. Then let our new mechanism have the outcome function o given by $o(\theta_1, \ldots, \theta_N) = o^{old}(s_1^{old}(\theta_1), \ldots, s_N^{old}(\theta_N)) = c(\theta_1, \ldots, \theta_N)$. All we need to show is that truthtelling is a dominant strategies equilibrium. To show this, we observe that for any i and $\theta_i \in \Theta_i$, for any alternative type $\hat{\theta}_i \in \Theta_i$, and for any $\theta_{-i} \in \Theta_{-i}$, $u_i(\theta_i, o(\theta_i, \theta_{-i})) = u_i(\theta_i, o^{old}(s_i^{old}(\theta_i), s_{-i}^{old}(\theta_{-i}))) \geq u_i(\theta_i, o^{old}(s_i^{old}(\hat{\theta}_i), s_{-i}^{old}(\theta_{-i})))= u_i(\theta_i, o(\hat{\theta}_i, \theta_{-i}))$, where the inequality derives from the fact that the s_i^{old} constitute a dominant strategies equilibrium in the original game. □

Revelation Principle, version 2 (Known). *Suppose there is a strategic game form that implements a social choice rule $c : \Theta_1 \times \Theta_2 \times \ldots \times \Theta_N \rightarrow O$ in Bayes-Nash equilibrium. Then there exists a direct-revelation game with outcome function o that implements c in Bayes-Nash equilibrium, where the Bayes-Nash equilibrium through which it is implemented is truthful. That is, for any i and $\theta_i \in \Theta_i$, $s_i(\theta_i) = \theta_i$, where these strategies constitute a Bayes-Nash equilibrium, and $o(\theta_1, \ldots, \theta_N) = c(\theta_1, \ldots, \theta_N)$.*

Proof. We show how to transform the given game form that implements c into a truthful direct-revelation game that implements c. For each i, let $s_i^{old} : \Theta_i \rightarrow A_i^{old}$ be the strategy played by agent i in the equilibrium that implements c in the given game, and let o^{old} be the given game's outcome function, so that $o^{old}(s_1^{old}(\theta_1), \ldots, s_N^{old}(\theta_N)) = c(\theta_1, \ldots, \theta_N)$, and the s_i^{old} constitute a Bayes-Nash equilibrium. Then let our new mechanism have the outcome function o given by $o(\theta_1, \ldots, \theta_N) = o^{old}(s_1^{old}(\theta_1), \ldots, s_N^{old}(\theta_N)) = c(\theta_1, \ldots, \theta_N)$. All we need to show is that truthtelling is a Bayes-Nash equilibrium. To show this, we observe that for any i and $\theta_i \in \Theta_i$, for any alternative type $\hat{\theta}_i \in \Theta_i$, $E_{\theta_{-i} \leftarrow p_{-i}}[u_i(\theta_i, o(\theta_i, \theta_{-i}))] = E_{\theta_{-i} \leftarrow p_{-i}}[u_i(\theta_i, o^{old}(s_i^{old}(\theta_i), s_{-i}^{old}(\theta_{-i})))] \geq E_{\theta_{-i} \leftarrow p_{-i}}[u_i(\theta_i, o^{old}(s_i^{old}(\hat{\theta}_i), s_{-i}^{old}(\theta_{-i})))] = E_{\theta_{-i} \leftarrow p_{-i}}[u_i(\theta_i, o(\hat{\theta}_i, \theta_{-i}))]$, where the inequality derives from the fact that the s_i^{old} constitute a Bayes-Nash equilibrium in the original game. □