Privacy Desiderata in Mechanism Design*

Andreas Haupt        Zoë Hitzig

December 15, 2021

Abstract

In a direct mechanism, a communication protocol queries agents’ private information in order to determine the outcome. Protocols make a distinction between the information solicited by the mechanism designer and the information revealed to the designer, and thus allow for the formulation of privacy desiderata in mechanism design.

One such desideratum is need-to-know privacy, which formalizes a notion of data minimization. A protocol is need-to-know private if every piece of an agent’s private information that is revealed to the designer is needed to determine their outcome. A social choice rule is need-to-know implementable if there is a need-to-know protocol that implements it.

Need-to-know implementability depends on the commitment power of the designer. When the designer is able to commit to arbitrary (cryptographic) protocols, any non-bossy social choice rule is need-to-know implementable. When the designer can only commit to personalized queries that correspond to messages sent in an extensive-form game, random serial dictatorship is the unique need-to-know and efficient object assignment rule, and the first price auction is the unique need-to-know and efficient standard auction. When the designer can commit to making some anonymous queries, the second-price auction becomes need-to-know implementable.

1 Introduction

Standard mechanism design relies heavily on the revelation principle. The revelation principle implicitly assumes that when an agent communicates their private information to the designer, they communicate all of their private information to the designer all-at-once. In practice, private information is often communicated sequentially rather than all-at-once, and different “direct” mechanisms may result in different information revealed to the designer.

Consider the difference between a sealed-bid second-price auction and an ascending auction. A sealed-bid second price auction aligns well with the standard information revelation paradigm: Participants write their bids in sealed

*We thank Shengwu Li, Ben Golub, Eric Maskin, Yannai Gonczarowski, Matthew Rabin, Benji Niswonger and John Macke for helpful conversations, and seminar audiences at Harvard for their remarks. Errors are our own.
envelopes, and when the envelopes are opened their entire bids are revealed to the designer. Meanwhile, the ascending auction aligns less well with the standard mechanism design paradigm. The designer learns agents’ private information gradually. At the end of the auction, the designer knows the losing bidders’ exact bids while it knows only a lower bound on the winner’s bid.

In this paper, we study privacy desiderata in auctions and assignment problems. In order to do so, we define communication protocols that specify how the designer queries agents’ reports once the reports have been submitted. This formal tool allows us to depart from the standard mechanism design paradigm—we make a distinction between the information solicited by the designer and the information revealed to the designer.

Depending on the environment, the designer may have different types of protocols available to her. Technological constraints may prevent some designers from implementing high-tech cryptographic protocols which completely preserve agents’ privacy. For instance, it may be computationally infeasible for platforms to use advanced cryptography in their advertising auctions, which have to compute winners in a fraction of a second. In other settings, designers may simply not have the resources to implement such complex protocols. Even the largest auctions for cattle (e.g. Fort Worth) or flowers (e.g. Aalsmeer) are often highly local and run by auctioneers who do not have advanced training in—or access to—multi-party computation.

Beyond technological feasibility, trust, commitment and sophistication play a role in determining which kinds of protocols are available to a designer in a given setting. In auctions with sophisticated bidders and a trustworthy designer, the designer can communicate how participants’ privacy is preserved simply by describing the specific cryptographic protocol used. The first large-scale use of secure multiparty computation was in 2008—in this instance, the Danish sugar beet farmers’ association and Danisco jointly decided on a cryptographic protocol, creating a high degree of trust in the resulting cryptographic auction of sugar beets to Danisco [Bogetoft et al., 2009]. In 2018 case with a similar degree of sophistication and trust, gem traders participated in a high stakes auction for rough diamonds using a blockchain-based protocol administered through the highly regarded Antwerp diamond supply chain consultancy Gemdax.

But apart from some specialized settings with high trust and technological sophistication, cryptographic auctions are not widely used. For instance, bidders in fine art and design auctions run by Sotheby’s, Phillips and similar auction houses care about their privacy to some degree (most bidders call their bids in to auction house representatives, shielding their identity at least from other bidders if not from the auctioneer). And yet, despite the fact that these auction houses have the technological capacity and financial resources to implement highly sophisticated fully privacy-preserving protocols, they do not (to our knowledge).

How can designers commit to preserving participants’ privacy when high-tech cryptographic protocols are not available to them? In this paper, we study privacy desiderata that can guide design choices when—for whatever reason—the designer cannot implement secure multiparty computing protocols.

There are many imaginable privacy desiderata. We define and analyze need-
to-know private protocols, which are derived from principles of data minimization. A protocol is need-to-know if no more is learned by the designer than is needed to determine the outcome.

We study two variants of need-to-know privacy. Individually need-to-know private protocols require that each piece of information learned about each agent must be used in determining that agent’s outcome while collectively need-to-know private protocols require that each piece of information learned about the type profile must be used in determining the overall allocation.

A choice rule is need-to-know implementable if there is a need-to-know private protocol that implements it. So, need-to-know implementability necessarily depends on the types of protocols that are available to the designer. We study three different types of protocols that correspond to varying degrees of technological sophistication and/or trust of the designer in a given setting.

First, when the designer can use any arbitrary protocol, many choice rules are need-to-know implementable. Second, when the designer is limited to using only protocols in which participants are queried personally and sequentially—as if being called to play in an extensive form game—very few choice rules are need-to-know implementable. Finally, we discuss a class of protocols that fall in between the prior two extremes. When the designer has access to protocols in which agent reports can be anonymized up to a point, some but not all choice rules are need-to-know.

The design desideratum we define, need-to-know implementability, relates to other concepts in the literature on privacy and commitment in market design and mechanism design. Our privacy desideratum is distinct from prior work in that it focuses specifically on how the revealed information is used by the designer. In addition, relative to prior work, we place greater emphasis on how the implementability of privacy-preserving rules depends on the commitment power of the designer. While many other papers have looked at particular privacy desiderata in either auction or assignment domain, our privacy desideratum applies to both domains. We discuss the most closely related ideas in section 2.

The remainder of the paper proceeds as follows. After reviewing closely related concepts in section 2, we present the basic model in section 3. Then, we discuss need-to-know implementation when: arbitrary (cryptographic) protocols are available (section 4), only sequential elicitation protocols (section 5) and only partially anonymous protocols are available (section 6). Section 7 concludes.

2 Related Literature

Several authors have incorporated measures of “privacy loss” into mechanism design. For instance, [Eilat et al. (2021)] define privacy loss to be the Kullback-Leibler divergence between the designer’s prior and her posterior. They study optimal mechanisms subject to a constraint on the privacy loss. Our approach

---

1 Data minimization has been legally codified in Article 5.1 of the European Union’s General Data Protection Regulation.
differs from theirs in two ways: First, our approach is much more tractable when dealing with multi-agent mechanisms and does not need to make strong symmetry assumptions. Also, our privacy desideratum, need-to-know privacy, is not about how much or whether information is revealed, but rather it is about how the information revealed is used.

The degree of privacy preservation in auctions has been studied extensively in computer science. The literature on cryptographic protocols for auctions, going back to Nurmi and Salomaa (1993) and Franklin and Reiter (1996) is too vast to summarize here—the main point is that there are many cryptographic protocols that do not reveal any private information to a designer. Such protocols allow participants to jointly compute the outcome without relying on any trusted third party. Most relevant to our purposes is the concept of unconditional full privacy, which allows bidders to compute an auction outcome while relying neither on trusted third parties nor on computational intractability (Brandt and Sandholm, 2008). Unconditional full privacy requires that the only information revealed is the information contained in the outcome. Milgrom and Segal (2020) takes inspiration from Brandt and Sandholm (2008)’s concept in its proposal of unconditional winner privacy.

Differential privacy, originally proposed in Dwork (2006) as a tool for database management, has been studied widely in mechanism design, see survey in Pai and Roth (2013). Differential privacy, as adapted for mechanisms design contexts, says that the report of a single agent should have a negligible effect on the outcome. Differential privacy is thus very different in spirit to need-to-know privacy. Loosely speaking, differential privacy sees the justifiability of information use as inversely related to the degree to which it makes a difference to the outcome, while need-to-know privacy sees the justifiability of information use as positively related to the degree to which that information makes a difference to the outcome.

Though we interpret need-to-know implementability primarily through the lens of privacy, it also relates to fairness properties of choice rules. For example, nonbossiness introduced in Satterthwaite and Sonnenschein (1981) has been widely studied in economics and computer science. A social choice function is nonbossy if an agent cannot make a report that leaves her own outcome unchanged while changing someone else’s outcome. Non-bossiness is a necessary but not sufficient condition for need-to-know privacy with sequential elicitation protocols. Further, our work can be understood as offering a privacy-related argument in favor of nonbossy mechanisms, and, more broadly, connects non-bossiness to other concepts in the privacy literature.

Another fairness-related desideratum that relates to need-to-know implementation is credibility (Akbarpour and Li, 2020). An extensive-form mechanism is credible if it is incentive compatible for the auctioneer to follow the rules. The first-price auction is the unique credible and static auction while the ascending auction is the unique credible and strategyproof auction. Generally, credibility is orthogonal to need-to-know implementability. But, there may be special cases in which the designer has objectives such that credibility implies need-to-know implementability. This connection will be further discussed.
3 Model

There are $n \in \mathbb{N}$ agents $i = 1, 2, \ldots, n$ with types $\theta_i \in \Theta_i$. Agents interact with an allocation mechanism that assigns outcomes in $\mathcal{X}$. Agents have utilities over outcomes $u_i : \mathcal{X} \times \Theta \to \mathbb{R}_+$. We denote by $\theta = (\theta_1, \theta_2, \ldots, \theta_n)$ the full type profile. We denote the information that an agent observes on the outcome after the game by $g_i : \mathcal{X} \to \mathcal{O}$ for some set $\mathcal{O}$.

3.1 Protocols

For a choice rule $\phi : \Theta^n \to \mathcal{X}$, a protocol defines the process through which information is revealed about players.

**Definition 1 (Protocol).** A protocol $P$ is a directed rooted tree with vertices $V$ and edges $E$. Each vertex corresponds to a non-empty subset of types $\Theta^v = \times_{i=1}^n \Theta^v_i$. For all $w$ such that $(v, w) \in E$, $(\Theta^w_j)_{j \neq i}$ form a partition of $\Theta^v_i$.

We call the nodes with zero out-degree terminal. We say that a protocol $P$ certifies a social choice rule $\phi$ if $P$ yields sufficient information to compute the choice rule, i.e. for any terminal node $v$, and any $\theta, \theta' \in \Theta^v$, $\phi(\theta) = \phi(\theta')$. We then also say that $P$ is a protocol for $\phi$.

We also consider two classes of restricted protocols.

**Definition 2 (Sequential Elicitation Protocol).** A protocol $P = (V, E)$ is a sequential elicitation protocol if for any node $v \in V$ there is an agent $i \in N$ such that for all edges $e = (v, w) \in E$, $\Theta^v_j = \Theta^w_j$ for any $j \neq i$.

**Definition 3 (Partially Anonymous Mechanism).** A protocol $P = (V, E)$ is an anonymous protocol if for any node $v \in V$,

- Either there is an agent $i \in N$ such that for all edges $e = (v, w) \in E$, $\Theta^v_j = \Theta^w_j$ for any $j \neq i$, or
- there are sets $\tilde{\Theta}_v \subseteq \Theta$ for each $v$ a subset $A_v \subseteq \mathbb{N}$ such that $\Theta^v = \{\theta \in \Theta^n || \{i \in N : \theta_i \in \tilde{\Theta}_v\} \in A_v\}$.

A sequential elicitation protocol is asking participants one after another for parts of their private information, and can be realized as an extensive-form game. A partially anonymous mechanism can in addition check whether there are some users that have a certain property, without revealing the identity of the users. This can be, for example, realized by dissociating users from their types until the allocation has determined.

We now introduce the main desideratum of this article.

---

2The present note develops the most substantial results on protocols which sequentially elicit private information section 5.
Definition 4 (Need-to-Know Private Protocol). A protocol $P$ is (individually) need-to-know for a social choice function $\phi$ if for any distinct terminal nodes $v, v', \theta \in \Theta^n$ and $\theta_i' \in \Theta$, if $(\theta_i, \theta_{-i}) \in \Theta^v (\theta_i', \theta_{-i}) \in \Theta^{v'}$, we have

$$\phi_i(\theta_i, \theta_{-i}) \neq \phi_i'(\theta_i', \theta_{-i}).$$

If for all such types,

$$\phi(\theta_i, \theta_{-i}) \neq \phi(\theta_i', \theta_{-i})$$

holds, we call the protocol collectively need-to-know.

Roughly, a protocol is need-to-know if for all agents $i$ and all types, if $\theta_i$ is distinguishable from some $\theta_i'$ at the conclusion of the protocol, it must be the case that $\theta_i$ and $\theta_i'$ result in different outcomes under $\phi$. This criterion captures the idea that there must be a reason that the designer needed to know whether agent $i$ had type $\theta_i$ and not $\theta_i'$. That reason, in particular, is that without distinguishing $\theta_i$ from $\theta_i'$, the designer could not have determined $i$'s allocation.

Definition 5 (Need-to-Know Implementation). A choice rule $\phi$ is (individually or collectively) need-to-know implementable for a class of protocols $P$ if there exists a protocol $P \in P$ for $\phi$ that is need-to-know.

3.2 Properties of Social Choice Functions

We will relate individual need-to-know implementability to non-bossiness Satterthwaite and Sonnenschein [1981]. A choice rule is non-bossy if

$$\phi_j(\theta_i, \theta_{-i}) = \phi_j(\theta_i, \theta_{i}'_{-i}) \implies \phi_i(\theta_i, \theta_{-i}) = \phi_i(\theta_i, \theta_{i}'_{-i})$$

for all $\theta_i \in \Theta$.

A weaker notion than non-bossiness is group non-bossy if for any $\theta_{-i}, \theta_{i}'_{-i} \in \Theta_{-i}$ and $\theta_i \in \Theta$

$$(\forall j \in N \setminus \{i\} : \phi_j(\theta_i, \theta_{-i}) = \phi_j(\theta_i, \theta_{i}'_{-i})) \implies \phi_i(\theta_i, \theta_{-i}) = \phi_i(\theta_i, \theta_{i}'_{-i}).$$

3.3 Domains

We prove statements about the single-dimensional auction environment and a strict assignment domain.

In the auction domain, the type space is $\Theta_i = \mathbb{R}_+$ and there are at least two agents, i.e. $|N| \geq 2$. The set of outcomes $X = N \times \mathbb{R}_+$ has elements $x = (y, t)$ where $y \in N \cup \{\}$ is the winner and $t = (t_1, \ldots, t_n)$ is a vector of payments. Agents observe $g_i((y, t)) = (y, t_i)$. That is, agents observe whether they win the object and their own payment. Agents have utility

$$u_i((y, t), \theta) = 1_{i=y}v_i(\theta_i) - t_i$$

with $v_i : \Theta \to \mathbb{R}_+$.

In the assignment domain, we fix a set $C$ of objects. Then, the set of outcomes is $X = N \times C$. That is, outcomes are pairs between agents and objects—an element of the outcome set is $x = (i, c)$ which says that agent $i$
is assigned to object \( c \). Sometimes, we will assume that \( X \) needs to consist of matchings, i.e. injective and surjective relations. Types are permutations of the objects, i.e. bijective functions \( \theta_i : C \to C \).

### 3.4 Incentives

A social choice function is Dominant-Strategy incentive compatible or *strategy-proof* if

\[
u(\phi_i(\theta_i, \theta_{-i}); \theta_i) \geq u(\phi_i(\theta'_i, \theta_{-i}); \theta_i)
\]

for any \( \theta'_i \). Given i.i.d. types \( \theta_i \sim F \), \( \phi \) is Bayesian incentive-compatible if

\[
\mathbb{E}_{\theta_{-i} \sim F}[u(\phi_i(\theta_i, \theta_{-i}); \theta_i)] \geq \mathbb{E}_{\theta_{-i} \sim F}[u(\phi_i(\theta'_i, \theta_{-i}); \theta_i)].
\]

### 4 Full Commitment

In this section, we characterize the class of mechanisms that is (collectively or individually) need-to-know implementable if all protocols are admitted.

**Proposition 6.** If all protocols are admitted, every social choice function is collectively need-to-know implementable. A social choice function is individually need-to-know implementable if and only if it is non-bossy.

**Proof.** We consider a protocol with a root note \( \Theta^n \) and one child node per outcome \( x \in X \), with associated set \( \phi^{-1}(\{x\}) \). This is a protocol, and collectively need-to-know private. Indeed, if \( (\theta_i, \theta_{-i}) \in \Theta^v \) and \( (\theta'_i, \theta_{-i}) \in \Theta^w \), then by definition

\[
\phi(\theta_i, \theta_{-i}) = \phi(\theta'_i, \theta_{-i}).
\]

In addition, if \( \phi \) is non-bossy, assume that this protocol was not need-to-know. Assume that \( (\theta_i, \theta_{-i}) \in \Theta^v \) and \( (\theta'_i, \theta_{-i}) \in \Theta^{v'} \), and we \( \phi_i(\theta_i, \theta_{-i}) = \phi_i(\theta'_i, \theta_{-i}) \). Then by non-bossiness, \( \phi_j(\theta_i, \theta_{-i}) = \phi_j(\theta'_i, \theta_{-i}) \) for \( j \in N \setminus \{i\} \). This contradicts collective need-to-know privacy of this protocol which was proved above.

### 5 Sequential Elicitation Protocol

We characterize the choice rules with need-to-know sequential elicitation protocols in the auction and item assignment domains.

#### 5.1 Auctions

In the auction domain, we focus on a particular environment and set of auction rules which we call *standard* auctions.

**Definition 7.** A standard auction has the following features:

(i) there is a single indivisible item to be allocated \( (\phi(\theta) = (y, t)) \),

(ii) agents have independent private values for the item \( \theta \sim \text{id} F \), and
(iii) only the winner pays \((\phi(\theta) = (y, t) \text{ where } t_i = 0 \text{ for all } i \neq y)\).

An efficient auction is one in which the bidder who values the item most wins.

**Definition 8.** An auction \(\phi\) is efficient if the outcome \(\phi(\theta)\) satisfies

\[
\phi(\theta) \in \arg\max_{(y,t)} \sum_i u_i((y,t), \theta_i)
\]

for all \(\theta \in \Theta^n\).

Our first theorem characterizes the set of need-to-know standard auctions.

**Proposition 9.** The unique individual need-to-know Bayesian IC protocol that certifies an efficient standard auction is the Dutch protocol.

**Proof.** The proof proceeds in two steps: We first show that the certified choice function needs to be a First-Price Auction. Then, we show that the Dutch auction is the unique need-to-know protocol that certifies the First-Price Auction.

1. Given that the auction needs to be efficient, the allocation must be to the agent with the highest type. We prove that the payment can only depend on the winning agent’s type by contradiction. Assume that for some type profile \(\theta\), the payment of the winner \(t_i\) depended on the other agents’ types in the sense that there are \(\theta_j\) and \(\theta_j'\) such that

\[
t_i(\theta_j, \theta_{-j}) \neq t_i(\theta_j, \theta_{-j}'),
\]

while the winner is unchanged,

\[
y(\theta_j, \theta_{-j}) = y(\theta_j, \theta_{-j}').
\]

Since only the winner pays, the outcome for all agents but \(i\) are the same. (If there are changes in the payment from simultaneous changes of types \(\theta_{-i}\), then there must be single-agent type changes that change the payment as well.) This implies that the mechanism is bossy, which contradicts need-to-know privacy.

By revenue equivalence, Bayesian IC, and the fact that only the winner pays the payment needs to be the one in the First-Price Auction.

2. It remains to show that the unique protocol need-to-know implementing the first-price auction choice rule is the Dutch protocol. For this, it’s enough to show that the protocol’s queries are all binary, and for \((v, w), (v, w') \in E\) and \(\Theta^n_v \times \Theta^n_{-v}\), either \(\Theta^w = \{\max(\bigcup_{i=1}^n \Theta^n_i)\} \times \Theta^n_{-v}\)

or \(\Theta^{w'} = \{\max(\bigcup_{i=1}^n \Theta^n_i)\} \times \Theta^n_{-v}\). In English: The protocol always asks whether the type is the highest among the remaining possible types of all agents.

First observe that we can transform a protocol without affecting need-to-knownness to a binary protocol: If there is a query to several partition sets,
several questions can be asked to an agent one after the other. If there is only one child, the query is trivial, and the corresponding edge can be contracted. Therefore, it is without loss to assume binary queries.

Assume that a query to an agent $i$ happens when $\max v_i < \max(\bigcup_{i=1}^n v_i)$. With this set, it is compatible that $\theta_i < \max_{j \in [n]} \theta_j$. The query must elicit information on a loser’s type on where their type is. This contradicts need-to-knowness.

Now assume a binary query, which we write without loss as

$$\left\{ \max \left( \bigcup_{i=1}^n \Theta^v_i \right) \right\} \cup \tilde{\Theta}^w_i \times \Theta^v_{-i}$$

and

$$\Theta^w_i \setminus \left( \left\{ \max \left( \bigcup_{i=1}^n \Theta^v_i \right) \right\} \cup \tilde{\Theta}^w_i \right) \times \Theta^v_{-i}$$

Assume that $\theta_i \in \Theta^w_i \setminus \left( \left\{ \max(\bigcup_{i=1}^n \Theta^v_i) \right\} \cup \tilde{\Theta}^w_i \right)$ and hence $i$ being a loser. It would, however, only be necessary to certify that $i$ is a loser to know $\theta_i \in \Theta^w_i \setminus \left\{ \max(\bigcup_{i=1}^n \Theta^v_i) \right\}$, which contradicts need-to-knowness. Altogether, the protocol must be the Dutch protocol.

Proof. Let $\phi$ be group bossy and let $\theta_i$, $\theta_{-i}$ and $\theta'_{-i}$ such that

$$\forall j \in [n] \setminus \{i\} : \phi_j(\theta_i, \theta_{-i}) = \phi_j(\theta_i, \theta'_{-i}) \land \phi_i(\theta_i, \theta_{-i}) \neq \phi_i(\theta_i, \theta'_{-i}).$$

Assume there was a need-to-know protocol certifying $\phi$. By certification, there must be a query that separates $\phi_{-i}$ from $\phi'_{-i}$, and it must be to one of the agents $j \in [n] \setminus \{i\}$. The protocol distinguishes $\theta_j$ and $\theta'_j$, while the output for $j$ is the same. This is a contradiction to need-to-knowness.

Theorem 10. There is no efficient, strategyproof and need-to-know standard auction protocol.

Proof. The only need-to-know, efficient, standard auction (social choice function) is by the Dutch auction protocol. The Dutch auction protocol is not a dominant-strategy protocol. In particular, for types $\theta_j = \min \Theta$ for $j \in [n] \setminus \{i\}$ and $\theta_i > \theta_j$, agent $i$ has an incentive to misreport to $\theta_i > \theta_i' > \theta_j$. Hence, there is no efficient, strategyproof, need-to-know standard auction protocol.

5.2 Object Assignment

Proposition 11. The serial dictatorship protocol is the unique individually need-to-know, strategyproof and efficient assignment protocol.
Proof. Satterthwaite and Sonnenschein (1981) show that the unique nonbossy, strategyproof and efficient assignment rule is the serial dictatorship. By the lemma above, every need-to-know choice rule is nonbossy. So, it follows directly that the unique need-to-know, strategyproof and efficient assignment rule is the serial dictatorship. 

5.3 House Assignment

Need-to-know is a strong criterion. We discuss the compatibility of need-to-know with other desiderata such as efficiency, strategyproofness, stability, and individual rationality.

In the auction domain, we again focus on standard auctions.

Theorem 12. There is no non-trivial, individually rational and need-to-know protocol for the house assignment problem.

Proof. Consider an assignment problem where agent $i$ owns agent $i$ at the outset. Consider any preference profile $\succ$ in which agent $i$ prefers house $i$ over all other houses for any $i$ and there are two agents $i$ and $j$ whose top-2 choices are $i$ and $j$. Define preference profiles $\succ'$ and $\succ''$ such that they equal $\succ$ except that for two agents $i$ and $j$,

$$j \succ'_i k \iff i \succ_i k$$

for any $k \in \mathcal{N}$ and

$$i \succ''_j k \iff k \succ_j k$$

for any $k \in \mathcal{N}$. In particular, in both cases, agent $i$ and $j$ have the same most preferred house. Also, there is a unique individually rational allocation, which is $\phi(k) = k, i \in \mathcal{N}$. Consider any protocol $P$ and the profiles $\succ'$ and $\succ''$. We call the point of earliest departure the node in the tree $P$ such that $\succ$ and $\succ'$ go into different directions. As the preference profiles only consider agents $i$ and $j$ and also only the relative preference of $i$ and $j$, it must be that the protocol for $P$ makes a query that elicits the preference for $i$ and $j$. Without loss of generality, let’s assume the preference is elicited from agent $i$. Then, $j \succ' i$ is learned. This information, however, is unnecessary to determine the outcome $\phi(i) = i$, for which the fact that $k \succ_k l$ for $k, l \in \mathcal{N}$ is needed. For $j$, a similar argument can be made for $\succ''$. Hence, the protocol is not need-to-know.

5.4 School Choice

Theorem 13. There is no stable and need-to-know implementable choice rule for the two-sided matching problem.

Proof. Assume that the two sides of the market are $s_1, s_2, \ldots, s_n$ and $c_1, c_2, \ldots, c_n$. Consider a preference profile in which $c_i \succ_s i, c_j$ for $s_i \succ c_i, s_j$ for any $i, j \in \mathcal{N}$ and for which for some $i, j \in \mathcal{N}, c_i, c_j$ are student $i$ and $j$’s top-2 choices and
$s_i, s_j$ are college $j$’s top-2 choices. Consider four preference profiles $\succ^i, \succ''', \succ'''$ and $\succ'''$ that equal $\succ$ except for

$$
c_i \succ_{s_i} c_k \iff c_j \succ_{s_i} c_k
\quad
s_i \succ_{c_i} s_k \iff s_j \succ_{c_i} s_k
\quad
s_j \succ_{c_j} s_k \iff s_i \succ_{c_j} s_k
$$

for any $k \in N$. There is a unique stable allocation, which is $\phi(s_i) = c_i$.

Consider any protocol $P$. We call the point of earliest departure the node in the tree $P$ such that any of the four profiles can be distinguished. As the only types that differ are for students $i$ and $j$ and colleges $i$ and $j$, this must be when any of these is asked. We assume that student $i$ is asked. (The other cases are symmetric.) Consider this query for $\succ^i$. As the profiles only differ in the ranking of $c_i$ and $c_j$ for this agent, $c_i \succ_{s_i} c_j$ is learned in this query. Note, however, that this information is not needed to determine the outcome $\phi(s_i) = c_i$. Hence, the protocol is not need-to-know.

\section{Partially Anonymous Protocols}

This section presents a first result on partially anonymous mechanisms: The second-price auction is need-to-know implementable when partially anonymous protocols are admitted.

\subsection{Auctions}

\begin{proposition}
The second-price auction is partially anonymously need-to-know implementable.
\end{proposition}

\begin{proofidea}
The elicitation protocol resembles an ascending auction. For nodes $v \in \mathbb{R}$ and edges given by $\Theta = \{i \in N|\theta > v\}$ and the sets $\{1\}$ and $N \setminus \{1\}$, we can elicit at which point exactly one user has a higher type than $v$. We can sequentially ask each user whether they are willing to buy the good at $v$.
\end{proofidea}

\section{Conclusion}

This article proposed communication protocols to formally think about privacy desiderata of mechanisms. We characterize need-to-know private implementability in the presence of three different classes of protocols: arbitrary (cryptographic) protocols, sequential elicitation protocols, and partially anonymous protocols.
References

Akbarpour, Mohammad and Shengwu Li, “Credible auctions: A trilemma,” *Econometrica*, 2020, 88 (2), 425–467.

Bogetoft, Peter, Dan Lund Christensen, Ivan Damgård, Martin Geisler, Thomas Jakobsen, Mikkel Krøigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter et al., “Secure multiparty computation goes live,” in “International Conference on Financial Cryptography and Data Security” Springer 2009, pp. 325–343.

Brandt, Felix and Tuomas Sandholm, “On the existence of unconditionally privacy-preserving auction protocols,” *ACM Transactions on Information and System Security (TISSEC)*, 2008, 11 (2), 1–21.

Dwork, Cynthia, “Differential privacy,” in “International Colloquium on Automata, Languages, and Programming” Springer 2006, pp. 1–12.

Eilat, Ran, Kfir Eliaz, and Xiaosheng Mu, “Bayesian privacy,” *Theoretical Economics*, 2021, 16 (4), 1557–1603.

Franklin, Matthew K and Michael K Reiter, “The design and implementation of a secure auction service,” *IEEE Transactions on Software Engineering*, 1996, 22 (5), 302–312.

Milgrom, Paul and Ilya Segal, “Clock auctions and radio spectrum reallocation,” *Journal of Political Economy*, 2020, 128 (1), 1–31.

Nurmi, Hannu and Arto Salomaa, “Cryptographic protocols for Vickrey auctions,” *Group Decision and Negotiation*, 1993, 2 (4), 363–373.

Pai, Mallesh M and Aaron Roth, “Privacy and mechanism design,” *ACM SIGecom Exchanges*, 2013, 12 (1), 8–29.

Satterthwaite, Mark A and Hugo Sonnenschein, “Strategy-proof allocation mechanisms at differentiable points,” *The Review of Economic Studies*, 1981, 48 (4), 587–597.