Contextually Private Mechanisms

Andreas Haupt and Zoë Hitzig*

March 21, 2025

Abstract

The rise of powerful AI tools has transformed the information landscape, making it inexpensive to harvest, aggregate, and repurpose data. Preserving privacy in such an environment demands market mechanisms that elicit nothing beyond the information strictly required for the economic objective at hand. We study such privacy-preservation in sequential elicitation mechanisms for market rules. A mechanism produces a *contextual privacy violation* if it elicits superfluous information. A mechanism is *maximally contextually private* if there is no protocol that produces a subset of the violations it produces, while still implementing the market rule. We show that selecting a maximally contextually private protocol involves a deliberate decision about whose privacy is most important to protect, and these protocols delay questions to those they aim to protect. Taking the second-price auction rule as an instructive example, we derive two novel designs that are maximally contextually private: the *ascending-join* and *overdescending-join* protocols.

1 Introduction

In standard mechanism design, a designer elicits reports of agents' private information in order to determine the outcome of a social choice rule. Ex-post, in an incentive compatible direct revelation mechanism, the designer learns *all* of agents' private information—typically more than is necessary for computing the rule. For example, in a sealed-bid second price auction, the designer learns all losing bids exactly, even though it is only necessary to know that all losing bids fall below the second highest. The designer also learns the winner's bid exactly, even though it is only necessary to know that the winning bid is above the second highest.

*Haupt: MIT, haupt@mit.edu. Hitzig: Harvard Society of Fellows, zhitzig@g.harvard.edu. We thank especially Eric Maskin, Shengwu Li and Ben Golub. We also thank Yannai Gonczarowski, Jerry Green, Chang Liu, Dirk Bergemann, Stephen Morris, Mohammad Akbarpour, Ilya Segal, Paul Milgrom, Kevin He, Navin Kartik, Ludvig Sinander, Clayton Thomas, Oghuzan Celebi, David Laibson, Modibo Camara, Katrina Ligett, Benji Niswonger and Max Simchowitz, as well as audiences in seminars, workshops and conferences (Harvard and MIT Theory Lunch workshops; seminars at Oxford, Stanford GSB, Boston College and Boston University; Penn State Economic Theory Conference, Penn/Wharton Mini Conference in Economic Theory; Stony Brook Game Theory Festival, Berkeley SLMath Workshop on Mechanism and Market Design). An extended abstract of an earlier version of this paper appeared in Proceedings of the 23rd ACM Conference on Economics and Computation (EC 2022). This work was supported by a Microsoft Research PhD Fellowship (Hitzig).

At the same time, there are several reasons why it may be desirable that the designer not learn "too much" about agents. First, gaining excess knowledge of agents' private information could expose the designer to legal liabilities or political risk. For example, as recounted in McMillan (1994), a second-price auction for spectrum licenses in New Zealand had a "political defect": "by revealing the high bidder's willingness to pay, the auction exposed the government to criticism" (McMillan, 1994). The government would have benefited from a design that ensured it only learned what was strictly needed—with the idea that, as the adage goes, what they don't know can't hurt them. Second, if agents have privacy concerns, they may be reluctant to reveal their private information even if the appropriate allocative incentives are in place. Agents may worry that their private information could be used against them in subsequent interactions with the designer or third parties (Rothkopf et al., 1990; Ausubel, 2004). Indeed, a 2022 lawsuit alleged that Google stored losing bids in their second-price auctions for advertising, and used this information against advertisers in future interactions.[1]

In this paper, we study how mechanism designers can limit the superfluous information they learn. In our set up, when a designer commits to a social choice rule, they also choose a dynamic protocol for eliciting agents' information (or "types"). These dynamic protocols allow the designer to learn agents' private information gradually, ruling out possible type profiles until they know enough to compute the outcome of the rule. The key idea we introduce is a *contextual privacy violation*. A protocol produces a contextual privacy violation for a particular agent if the designer learns a piece of their private information that may be superfluous, i.e. it might not be necessary for computing the outcome. We study protocols that are on the frontier of contextual privacy and implementation—that is, we study a setwise order based on contextual privacy violations, and look for maximal elements in this order. For some choice rules, it will be possible to find protocols that produce no violations for any agent at any type profile—we call these protocols *contextually private*.

We define protocols in a way that allows us to accommodate a broad range of environments. A protocol is composed of queries. A query can be directed to one agent—for example, a designer may ask one agent "Is your type above x?" Or, a query can be directed to multiple agents—e.g. the designer could ask "How many agents have a type above x?" A query can also be more complex than a simple threshold, for example, the designer could ask "What is the average type among agents with types between x and x'?" The details of the environment may dictate the format of queries the designer is able to ask—we call the set of available queries the *elicitation technology*.

An elicitation technology represents *how* the designer can learn about agents' messages, and by inverting their strategies, their private information. One possible elicitation technology is a "trusted third party." If there is a trusted third party, the designer can delegate information retrieval to this third party, and ask the third party to report back only what is needed to compute the outcome.

[1] In particuarly, the lawsuit *State of Texas v. Google* alleged that "Google induced advertisers to bid their true value, only to override pre-set AdX floors and use advertisers' true value bids against them... generat[ing] unique and custom per-buyer floors depending on what a buyer had bid in the past."

So, under this trusted third party elicitation technology, all choice rules can trivially eliminate all contextual privacy violations. Cryptographic techniques like secure multi-party computation and zero-knowledge proofs are elicitation technologies that similarly trivialize contextual privacy.[2]

But there are many environments in which the social or technological environment will neither permit the use of a trusted third party nor of trusted cryptographic tools. Advanced cryptography may be excessively costly in terms of time, money or computational power.[3] In addition, sophisticated cryptographic mechanisms require sophisticated participants—if participants do not understand how their information is kept private, their privacy concerns may not be alleviated.[4]

We focus on protocols founded on minimal assumptions regarding trust and comprehension. Specifically, for the most design-relevant portions of the paper, we restrict attention to *individual elicitation* technologies. In an individual elicitation protocol, the designer is limited to queries directed at individual agents, sequentially learning about this agent's type. Individual elicitation protocols neatly expose the privacy properties of a given protocol: for an agent to understand what the designer has learned about her, she merely has to recall her responses to the designer's questions. In other words, such protocols are unmediated.[5] They have the convenient property that the designer knows a piece of an agents' private information if and only if the agent said it.

We now present a brief overview of the paper. In Section 2, we articulate our formal framework and present the key definitions of the paper. We first must formally define the dynamic messaging game played by agents, and the elicitation technologies that limit what the designer learns about agents' messages. Then, we introduce the central definitions: contextual privacy violations, the contextual privacy order (an inclusion order based on violations), and maximally contextually private protocols (maximal elements in the inclusion order).

In Section 3, we study how specific properties of choice rules tend to lead to contextual privacy violations. In our first results, Proposition 1 and Proposition 2, we provide characterizations of choice rules that fully avoid contextual privacy violations under an arbitrary fixed elicitation technology, and under individual elicitation technologies, respectively. These abstract characterizations lead us to a more intuitive insight, Theorem 1, which says that under the restriction to individual elicitation protocols, any time there is some group of agents who are *collectively pivotal* but there is

[2]For a survey of cryptographic protocols for sealed-bid auctions, see Alvarez and Nojoumian (2020).

[3]Even if possible, some sophisticated solutions may be wasteful—in one of the earliest large-scale uses of secure multi-party computation, a double auction with sugar beet farmers in Denmark, designers wondered "if the full power of multiparty computation was actually needed," or if a simpler implementation guided by a weaker privacy criterion may have sufficed (Bogetoft et al., 2009).

[4]A recent survey shows that only 61% of WhatsApp's users believe the company's claim that their messages are end-to-end encrypted (Alawadhi, 2021).

[5]This minimal assumption stands in contrast to more substantive assumptions we could make. For example, if the designer's elicitation technology allows them to count the number of agents whose type satisfies a certain property without learning *whose* type satisfies that property, the agents must trust the designer's use of anonymization techniques. If the designer's elicitation technology enables the secret sharing behind secure multiparty computation, agents must grasp the idiosyncratic guarantees and computational assumptions of the particular secure multiparty computation protocol in use.

no agent who is *individually pivotal,* there must be a contextual privacy violation and the designer must choose whose privacy to protect.[6] Propositions 3-7 show how this conflict between collective and individual pivotality arises in common choice rules in environments with and without transfers.

In Section 4, we take the perspective of a privacy-conscious designer who wants to implement a choice rule through a maximally contextually private protocol. Our discussion yields two central insights. First, maximally contextually private protocols involve a deliberate decision about the *protection set*, i.e. a set of agents whose privacy ought to be protected if possible. Second, given that protection set, maximally contextually private protocols delay asking questions to protected agents as much as possible.

To get to these insights, we first present a key theoretical result of the paper that helps us reason about the contextual privacy-implementation frontier. Theorem 2 shows that for choice rules defined on ordered type spaces, it is without loss for such a designer to consider only *bimonotonic* protocols. Formally, the result is a representation theorem, showing that every protocol is contextual privacy equivalent to a bimonotonic protocol. A bimonotonic protocol consists of *threshold queries* which, for each agent, are monotonically increasing or decreasing in the threshold.

We then turn to the special case of the second-price auction rule as an instructive and practically relevant example—we see how maximally contextually private protocols for the second-price auction rule choose a set of agents to protect, and delay asking questions to the protected agents. In Theorem 3, we use the representation theorem (Theorem 2) to derive two maximally contextually private protocols: the *ascending-join* and the *overdescending-join* protocols. In the ascending-join protocol, which is a variant of the familiar ascending or "English" auction protocol, the designer begins by conducting an ascending protocol with just two agents. Whenever one agent drops out, another agent "joins" the protocol at the going threshold. The overdescending-join protocol is analogous, and related to the overdescending protocol introduced in Harstad (2018). The ascending-join protocol protects the winner, the overdescending-join protocol protects the losers, and both protocols avoid violations by delaying questions to agents in the protected set as much as possible.

Under the restriction to individual elicitation, protocols induce a well-defined extensive-form game. In Section 5, we "check" the incentive properties of the maximally contextually private protocols for the second-price auction rule described in Section 4. The ascending-join and overdescending-join protocols have implementations in dominant strategies, with the former satisfying the stronger requirement of obvious strategyproofness. This section relies on prior results regarding the strategic properties of personal clock-auctions (Li, 2017).

Finally, we discuss related literature in Section 6 and conclude in Section 7.

Proofs omitted from the main text are in Appendix A.

[6] One can derive as a special case of Theorem 1 a result from the cryptography literature known as the Corners Lemma (Chor and Kushilevitz, 1989; Chor et al., 1994). This result has been used to show that the second-price auction does not permit a decentralized computation protocol (Brandt and Sandholm, 2005) that satisfies *unconditional privacy*, compare Chor and Kushilevitz (1989) and Milgrom and Segal (2020).

2 Model

There is a set $N = \{1, 2, \ldots, n\}$ of agents with private information (or "types") θ_i distributed according to $F_i \in \Delta(\Theta)$, where Θ is a finite type space. We denote by $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n) \in \Theta^n = \boldsymbol{\Theta}$ a profile of agents' types.[7] Agents have utility functions u over outcomes in X which may depend on their private types, $u_i : \Theta \times X \to \mathbb{R}$. All primitives of the model besides the realized type profile $\boldsymbol{\theta}$ are common knowledge.

A designer implements a deterministic social choice function $\phi\colon \boldsymbol{\Theta} \to X$ through a dynamic protocol in which agents repeatedly send messages m from a message space M.

2.1 Protocols.

The protocol consists of rounds. In each round, all agents submit a message $m \in M$ from a finite message space, with message profile $\mathbf{m} \in \mathbf{M} \coloneqq M^n$. The designer chooses a partition $\mathcal{S} \subset 2^{\mathbf{M}}$ and observes which partition cell $S \subset \mathcal{S}$ the message profile lies in.

Not all partitions are admissible. The designer must choose partitions from an *elicitation technology*, a set $\mathfrak{S} \subseteq 2^{2^{\mathbf{M}}}$ of admissible partitions. Most of this paper considers a particular elicitation technology that mimics communication with one agent at a time. The *individual elicitation technology,* denoted $\mathfrak{S}_{\text{IE}}$, contains n partitions:

$$\mathcal{S}_i = \{\{\mathbf{m} \in \mathbf{M} : m_i = m\} : m \in M\}. \tag{1}$$

The partition cells of $\mathcal{S}_i$ differ only in one agent i's message. So, we can interpret protocols that exclusively use the individual elicitation technology as protocols in which the designer queries one agent at a time. These protocols are unmediated in the sense that the designer need not *commit to forget* anything about agent i's submitted report.

To represent salient features of the designer's social and technological environment one could consider a variety of elicitation technologies. In Appendix D, we consider the *count elicitation technology,* under which a designer can observe the number of agents whose messages satisfy a given property, without learning the identities of agents who send a message with the given property. This elicitation technology corresponds to the common practice of anonymized elicitation (for example, many political elections use a secret ballot and online auctions permit anonymous bidding). Or, more formally, these technologies represent the designer's commitment to forget the labels of submitted messages.

Each protocol—a series of queries that may adaptively depend on prior queries—can be represented as a directed rooted tree. The tree's nodes correspond to histories of chosen partitions and

[7] We follow the usual convention of referring to partial type profiles excluding an agent i by $\boldsymbol{\theta}_{-i} = (\theta_j)_{j \in N \setminus \{i\}}$. We also sometimes extend this notation to refer to partial type profiles excluding sets of agents $A \subseteq N$, i.e. $\boldsymbol{\theta}_{-A} = (\theta_j)_{j \in N \setminus A}$.

observed partition cells. We denote the root node by r and terminal nodes by $z \in Z$. The set of all nodes of the tree is V and the set of edges is E. We will also write $P = (V, E)$ for a protocol. We encode the choice of partition at a node v through a function $s_v \colon \mathbf{M} \to \text{children}(v) \subset V$. Each child of v corresponds to one of the partition cells. It is then equivalent to using an elicitation technology $\mathfrak{S}$ such that,

$$\{s_v^{-1}(w) \subseteq \mathbf{M} : w \in \text{children}(v)\} \in \mathfrak{S}.$$

In this formalism, nodes v encode the full history of the chosen partitions and observed cells. We refer to a protocol with elicitation technology $\mathfrak{S}$ as an $\mathfrak{S}$-protocol, and denote by $\mathcal{P}_{\mathfrak{S}}$ the set of all protocols with elicitation technology $\mathfrak{S}$. Table 1 compiles notation.

2.2 Strategies.

At non-terminal nodes, agents submit messages according to deterministic strategies $\sigma_i \colon (V \setminus Z) \times \Theta \to M$, with strategy profiles denoted $\boldsymbol{\sigma} \coloneqq (\sigma_1, \sigma_2, \ldots, \sigma_n)$. That is, depending on their type θ_i, an agent chooses a message to send at each non-terminal node $v \in V$. We will write $\boldsymbol{\sigma}_v \colon \boldsymbol{\Theta} \to \mathbf{M}$ to refer to the strategy profile evaluated at node v.

Nodes $v \in V$ may be identified with the *information available to the designer at node*, which is the set of all type profiles that are consistent with the observed partition cells so far,

$$\boldsymbol{\Theta}_v \coloneqq \bigcap_{(u,w) \in \text{path}(v)} \boldsymbol{\sigma}_u^{-1}(s_u^{-1}(w)),$$

where, path(v) is the set of edges from the root node r to the node v. Protocols can be equivalently represented as $((\boldsymbol{\Theta}_v)_{v \in V \setminus \{r\}}, \phi)$ and $((s_v)_{v \in V \setminus Z}, \phi)$.[8]

We inductively define the outcome $P(\boldsymbol{\sigma}(\boldsymbol{\theta}))$ of protocol P under type profile $\boldsymbol{\theta}$ and strategies $\boldsymbol{\sigma}$. The induction starts with the root node r. For any non-terminal node $v \in V \setminus Z$, the successor node is

$$s_v(\sigma_1(\theta_1, v), \sigma_2(\theta_2, v), \ldots, \sigma_n(\theta_n, v)) \in V.$$

At a terminal node $z \in Z$, we choose the outcome x associated with z. We say that P is a protocol *for* choice rule ϕ with strategies $\boldsymbol{\sigma}$ if

$$P(\boldsymbol{\sigma}(\boldsymbol{\theta})) = \phi(\boldsymbol{\theta}).$$

If there is an $\mathfrak{S}$-protocol for a choice rule ϕ with some strategies $\boldsymbol{\sigma}$, we say that the choice rule is $\mathfrak{S}$-*computable*. We will also sometimes call $(P, \boldsymbol{\sigma})$ a protocol.

[8] That is, the set of protocols $\mathcal{P}_{\mathfrak{S}}$ represented as trees with node labels s_v for non-terminal nodes is isomorphic to the set $\mathcal{P}_{\mathfrak{S}}$ with node labels $\boldsymbol{\Theta}_v$.

Name	Sets	Representative Element
Agents	$\{1, \ldots, n\}$	i
Type profiles	$\boldsymbol{\Theta} = \Theta^n$	$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$
Message profiles	$\mathbf{M} = M^n$	$\mathbf{m} = (m_1, \ldots, m_n)$
Queries	$\mathfrak{S}$	$\mathcal{S},\ s_v$
Message partition cells	$\mathcal{S}$	S
Protocols	$\mathcal{P}_{\mathfrak{S}}$	$P = (V, E)$
Nodes	V	v, w
Edges	E	$e = (v, w)$
Terminal nodes of protocol P	Z	z
Information at node v	$\boldsymbol{\Theta}_v$	
Relations		
Agent i prefers outcome x to x'		$x \succ_i x'$
Protocol P is more contextually private than P' for ϕ		$P \succ_\phi P'$
Type θ is succeeded by θ' in the type space		$\theta' = \text{succ}(\theta)$ or $\theta = \text{pred}(\theta')$

Table 1: Notation for Protocols and Relations

2.3 Contextual Privacy Violations.

Type profiles $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \boldsymbol{\Theta}$ lead to different terminal nodes if and only if there is a non-terminal node that *distinguishes* them, i.e. there are sibling nodes $v, v' \in V$ such that $\boldsymbol{\theta} \in \boldsymbol{\Theta}_v$ and $\boldsymbol{\theta}' \in \boldsymbol{\Theta}_{v'}$. We then also say that $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are distinguished *at* $v \in V$.

The main definition of this article is the *contextual privacy violation*. A contextual privacy violation is a piece of agent i's private information learned by the designer that did not play a role in determining the outcome at a given type profile $\boldsymbol{\theta}$. All of the key concepts of the paper employ this definition.

Definition (Contextual Privacy Violation). Let P be a protocol for ϕ with strategies $\boldsymbol{\sigma}$. We say that P and $\boldsymbol{\sigma}$ produce a *ϕ-contextual privacy violation* for agent i at $\boldsymbol{\theta} = (\theta_i, \boldsymbol{\theta}_{-i}) \in \boldsymbol{\Theta}$ if there is a type $\theta_i' \in \Theta$ such that

$$(\theta_i, \boldsymbol{\theta}_{-i}) \text{ and } (\theta_i', \boldsymbol{\theta}_{-i}) \text{ are distinguished, yet } \phi(\theta_i, \boldsymbol{\theta}_{-i}) = \phi(\theta_i', \boldsymbol{\theta}_{-i}).$$

Otherwise, we say agent i's contextual privacy is *preserved* at $\boldsymbol{\theta}$. We denote by $\Gamma(P, \boldsymbol{\sigma}, \phi) \subseteq N \times \boldsymbol{\Theta}$ the set of contextual privacy violations produced by $P, \boldsymbol{\sigma}$ and ϕ. If the strategy is clear from the context, we will also write $\Gamma(P, \phi)$ for brevity.

In other words, there is a contextual privacy violation for agent i under protocol P and choice rule ϕ at type profile $\boldsymbol{\theta}$ if the designer can tell apart types θ_i, θ_i' but θ_i and θ_i' lead to the same outcome under ϕ, holding fixed $\boldsymbol{\theta}_{-i}$.

An agent's privacy is preserved at a realized type profile $\boldsymbol{\theta}$ if the private information she reveals is necessary for computing the outcome. This notion of preservation aligns with the principle of *purpose limitation* laid out in the European Union's General Data Protection Regulation (GDPR),

implemented in 2018.[9] More broadly, contextual privacy preservation aligns with the influential theory of privacy as *contextual integrity* (Nissenbaum, 2004), which states that an information flow (e.g. from the agent to the designer) is private if it respects contextual expectations and norms (e.g. that the designer learns agent's private information *only* for the purposes of computing the choice rule).

Contextual privacy can also be used to formulate other meaningful desiderata—for example, the idea that queries should have only *legitimate explanations*.[10] Imagine that the designer is asked to explain why they must learn something about agent i, i.e. they are asked why they need to know that agent i's type is θ_i and not θ_i'. A *legitimate explanation* $\boldsymbol{\theta}_{-i} \in \boldsymbol{\Theta}^{n-1}$ is a profile of other agents' types that explains why they need this information. That is, an *explanation* is a profile of other agents' types $\boldsymbol{\theta}_{-i}$ under which the designer claims they will need to distinguish between types θ_i and θ_i'—an explanation $\boldsymbol{\theta}_{-i}$ is *legitimate* if the outcome of the choice rule would be different depending on the information distinguished about agent i. There is no contextual privacy violation for agent i, that is $\Gamma(P,\phi) \cap (\{i\} \times \boldsymbol{\Theta}) = \emptyset$, when all possible explanations are legitimate. Or, put in the contrapositive, there is a contextual privacy violation when there is some explanation that is not legitimate.

In Appendix E, we explore two modifications of contextual privacy violations that may track different design goals: individual contextual privacy and group contextual privacy. These variations highlight connections to other desiderata such as non-bossiness (Satterthwaite and Sonnenschein, 1981; Pycia and Raghavan, 2022) and (strong) obvious strategyproofness (Li, 2017; Pycia and Troyan, 2023).

The goal of the privacy-conscious market designer is to choose a protocol that is, in some sense, "better," from a privacy perspective, than other protocols that achieve the same outcome. In order to study this goal, we define the *contextual privacy order*.

Definition (Contextual Privacy Order). A protocol P for ϕ with strategies $\boldsymbol{\sigma}$ is *more contextually private* than another protocol P' for ϕ under strategies $\boldsymbol{\sigma}$, denoted $(P,\boldsymbol{\sigma}) \preceq_\phi (P',\boldsymbol{\sigma}')$ if $\Gamma(P,\boldsymbol{\sigma},\phi) \subseteq \Gamma(P',\boldsymbol{\sigma}',\phi)$. A protocol $(P,\boldsymbol{\sigma})$ for ϕ, $P \in \mathcal{P}_{\mathfrak{S}}$, is $\mathfrak{S}$-*maximally contextually private* if there is no $P' \in \mathcal{P}_{\mathfrak{S}}$ and no strategy $\boldsymbol{\sigma}'$ such that $\Gamma(P',\boldsymbol{\sigma}',\phi) \subsetneq \Gamma(P,\boldsymbol{\sigma},\phi)$.

The contextual privacy order is an inclusion order on the set of contextual privacy violations that allows us to compare two protocols P and P' for a given choice rule ϕ.[11] We call $\preceq_\phi$-maximal

[9] The principle of *purpose limitation* appears in Article 5(1) of the GDPR, and reads: "Personal data shall be collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes."

[10] The idea of a legitimate explanation parallels Akbarpour and Li (2020)'s notion of an *innocent explanation* which is used to define credibility: in their setting, an innocent explanation is a profile of other agent's types $\boldsymbol{\theta}_{-i}$ that can explain the observed outcome.

[11] Segal (2007) and Mackenzie and Zhou (2022) consider a related order, the *relative informativeness*, on the information revealed by different extensive-form implementations of choice rules. In Mackenzie and Zhou (2022), the relative informativeness order is invoked to illustrate that menu mechanisms can be less informative than direct

objects within $\mathcal{P}_{\mathfrak{S}}$ also *maximally contextually private* if the elicitation technology, strategies, and social choice function are clear from the context.

The contextual privacy order captures another principle of the EU's GDPR law, known as *data minimisation*.[12] This order takes the pragmatic approach of locating protocols that violate contextual privacy as little as possible while still computing the choice rule. Maximally contextually private protocols are those that trade off contextual privacy and computability in a Pareto optimal way, and thus can be seen as lying on the *privacy-implementation frontier*.

The contextual privacy order is robust in the following sense: if a protocol P is more contextually private than a protocol P', it elicits superfluous information from fewer agents at *any* type profile $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. That is, it is more private for any prior over the type profile F.

For the extreme case where a maximally contextually private protocol leads to no contextual privacy violation at all, i.e. there exists a P such that $\Gamma(P, \phi)$, we also call the choice rule ϕ *contextually private*.

2.4 Incentives.

As the agents' strategies $\sigma_i \colon \Theta \times V \to M$ condition on the full state of the protocol at every node, dynamic incentives may be complex under arbitrary elicitation technologies. As our main goal in this article is to introduce and study the demands of contextual privacy, we for the most part isolate contextual privacy concerns from incentive compatibility concerns. We give necessary conditions for contextual privacy that consider any (incentive compatible or not) strategies $\boldsymbol{\sigma}$ and we verify dynamic incentive compatibility for the protocols we construct in Section 5 and Appendix B. We defer our definitions of dynamic incentive compatibility to Section 5.

3 How Contextual Privacy Violations Arise

We first study how contextual privacy violations depend on the social choice rule a designer wishes to implement. We do this in two ways. First, we provide a general characterization of choice rules that admit a contextually private protocol—i.e. choice rules for which it is possible to find a protocol that produces *no* contextual privacy violations at any type profile. We give such characterizations in Section 3.1—one characterization is fully general, for an arbitrary fixed class of elicitation technologies, and the other characterization is for the restriction to individual elicitation technologies. Second, we use these characterizations to derive a simple illustration of how and where contextual privacy violations tend to arise under individual elicitation in common choice rules. That is, in

revelation mechanisms, while in Segal (2007), the order is employed to identify the communication costs of choice rules.

[12] Article 5(1) of the EU's GDPR law lays out the principle of data minimization as follows: "Personal data shall be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed".

Section 3.2, we see that whenever there is *collective pivotality but no individual pivotality*, there will be a violation. This principle helps us to understand not just whether contextual privacy will be violated but *where* contextual privacy fails in common choice rules.

3.1 Two Characterizations of Contextually Private Choice Rules.

We first characterize $\mathfrak{S}$-contextually private choice rules, for an arbitrary fixed class of elicitation technologies $\mathfrak{S}$ in Proposition 1. Then we offer in Proposition 2 a characterization that is specialized to the individual elicitation technology $\mathfrak{S}_{\text{IE}}$.

Proposition 1. *An $\mathfrak{S}$-computable choice rule ϕ is $\mathfrak{S}$-contextually private if and only if there does not exist a subset of type profiles $\hat{\boldsymbol{\Theta}} \subseteq \boldsymbol{\Theta}$ such that*

(i) $\phi|_{\hat{\boldsymbol{\Theta}}}$ is non-constant, and

(ii) for every partition $\mathcal{S} \in \mathfrak{S}^*$ such that none of the partition cells $S \in \mathcal{S}$ contains $\hat{\boldsymbol{\Theta}}$, there are type profiles $(\theta_i, \boldsymbol{\theta}_{-i}), (\theta_i', \boldsymbol{\theta}_{-i}) \in \hat{\boldsymbol{\Theta}}$ that lie in different partition cells of $\mathcal{S}$.

Here, $\mathfrak{S}^* \subseteq 2^{2^{\boldsymbol{\Theta}}}$ is the set of revealed partitions. *Define $\mathcal{S}^* \in \mathfrak{S}^*$ if there is a partition $\mathcal{S} \in \mathfrak{S}$ and functions $f_1, f_2, \ldots, f_n \colon \Theta \to M$ such that every $S^* \in \mathcal{S}^*$ is the preimage under $(f_1, f_2, \ldots, f_n) \colon \boldsymbol{\Theta} \to \mathbf{M}$ of some $S \in \mathcal{S}$, $S^* = (f_1, f_2, \ldots, f_n)^{-1}(S)$.*

In addition, if there is such a $\hat{\boldsymbol{\Theta}}$ then any protocol for ϕ must produce a contextual privacy violation for some type profile in $\hat{\boldsymbol{\Theta}}$, that is, it must be that for any $P \in \mathcal{P}_{\mathfrak{S}}$, $\Gamma(P, \boldsymbol{\sigma}, \phi) \cap (\hat{\boldsymbol{\Theta}} \times N) \neq \emptyset$.

Note that this characterization is purely about the properties of a choice rule ϕ, and suggests that whether there is an admissible protocol P that does not produce a contextual privacy violation can be understood in terms of how the choice rule acts on subsets of the space of type profiles. This result is valuable because it tells us that for an arbitrary elicitation technology, small "counterexamples" can entirely characterize contextual privacy. For example, in the count elicitation technology discussed in the model section and in detail in Appendix D, a counterexample of four type profiles forces a contextual privacy violation for some agent.

That said, at this level of generality, this theorem gives little insight into *which* subsets of the space of type profiles $\hat{\boldsymbol{\Theta}}$ are likely to produce a violation. Checking whether conditions (i) and (ii) hold might require the consideration of all subsets of the space of type profiles $\hat{\boldsymbol{\Theta}}$ as well as any partition $\mathcal{S} \in \mathfrak{S}^*$. We now show that under a restriction to individual elicitation protocols, we can derive a more insightful characterization. This characterization relies on the notion of *type inseparability*.[13] Roughly, two types for an agent i are *inseparable* if the designer cannot distinguish between them without violating contextual privacy.

[13] Our concept of *inseparability* parallels the concept of *forbidden matrices* used in the work on 2-agent decentralized computation Chor and Kushilevitz (1989); Chor et al. (1994).

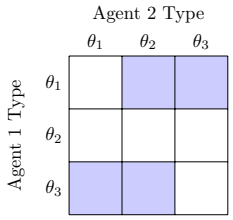


Figure 1: Illustration of Inseparable Types with $n = 2, \hat{\mathbf{\Theta}} = \{\theta_1, \theta_2, \theta_3\}^2$. Shaded regions represent outcome x under ϕ. For agent 1, $\theta_3 \sim_{1,\phi,\hat{\mathbf{\Theta}}} \theta_1$. For agent 2, $\theta_1 \sim_{2,\phi,\hat{\mathbf{\Theta}}} \theta_2 \sim_{2,\phi,\hat{\mathbf{\Theta}}} \theta_3$.

Definition (Inseparable Types). For a social choice function ϕ, call two types θ_i, θ_i' for an agent i *directly inseparable* on $\hat{\mathbf{\Theta}}$ under ϕ, denoted $\theta_i \sim'_{i,\phi,\hat{\mathbf{\Theta}}} \theta_i'$ if there exists $\boldsymbol{\theta}_{-i} \in \mathbf{\Theta}_{-i}$ such that $(\theta_i, \boldsymbol{\theta}_{-i}), (\theta_i', \boldsymbol{\theta}_{-i}) \in \hat{\mathbf{\Theta}}$, and

$$\phi(\theta_i, \boldsymbol{\theta}_{-i}) = \phi(\theta_i', \boldsymbol{\theta}_{-i}).$$

Denote the transitive closure of $\sim'_{i,\phi,\hat{\mathbf{\Theta}}}$ by $\sim_{i,\phi,\hat{\mathbf{\Theta}}}$. If $\theta_i \sim_{i,\phi,\hat{\mathbf{\Theta}}} \theta_i'$, call θ_i and θ_i' *inseparable* for i. We denote equivalence classes under $\sim_{i,\phi,\hat{\mathbf{\Theta}}}$ by $[\theta]_{i,\phi,\hat{\mathbf{\Theta}}}$.

To build further intuition for this definition, Figure 1 gives an illustration of inseparable types. The 3×3 grid represents a subset of the type space in a setting where there are two agents ($n = 2$). The shaded regions of the grid represent type profiles for which the outcome under ϕ is a particular outcome $x \in X$. Regions of the grid that are not shaded lead to arbitrary (other than x) outcomes under ϕ. On $\hat{\mathbf{\Theta}}$, all of agent 2's types are inseparable. To see this, note that θ_1 and θ_2 are directly inseparable—they lead to the same outcome x when agent 1's type is fixed at θ_3. Furthermore, for agent 2, θ_2 and θ_3 are directly inseparable, since they lead to the same outcome x when agent 1's type is fixed at θ_1. So, since inseparability is transitive, θ_1, θ_2 and θ_3 are all inseparable for agent 2.

In short, when a choice rule ϕ requires separating inseparable types there is a contextual privacy violation under $\mathfrak{S}_{\text{IE}}$.

Proposition 2. *A choice function ϕ fails to be $\mathfrak{S}_{\text{IE}}$-contextually private if and only if there exists some cylinder set $\hat{\mathbf{\Theta}} = \bigtimes_{i=1}^{n} \hat{\Theta}_i$ such that (i) $\phi|_{\hat{\mathbf{\Theta}}}$ is non-constant and (ii) for all agents i and all $\theta_i, \theta_i' \in \hat{\mathbf{\Theta}}_i$, θ_i and θ_i' are inseparable with respect to $\hat{\mathbf{\Theta}}$ and ϕ.*

In addition, it must be that for any protocol $(P, \boldsymbol{\sigma}) \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$, one of the agents i that has at least two inseparable types must incur a privacy violation, that is, $\Gamma(P, \boldsymbol{\sigma}, \phi) \cap \{(i, \boldsymbol{\theta}) \in N \times \mathbf{\Theta} \mid \exists \theta_i' \neq \theta_i : (\theta_i, \boldsymbol{\theta}_{-i}), (\theta_i', \boldsymbol{\theta}_{-i}) \in \hat{\mathbf{\Theta}}\} \neq \emptyset$.

The proofs of both characterizations Proposition 1 and Proposition 2 are similar. For necessity, we consider the first query that distinguishes type profiles from $\hat{\mathbf{\Theta}}$ which must exist due to the assumed non-constancy of ϕ restricted to $\hat{\mathbf{\Theta}}$. The conditions on $\hat{\mathbf{\Theta}}$ in (ii) require that this query

must lead to a contextual privacy violation. The proof of sufficiency is constructive.

These characterizations allow us to analyze the contextual privacy of choice rules, albeit in a way that does not directly offer insight into why contextual privacy fails. In the next subsection, we will leverage Proposition 2 to derive a more intuitive result that shows how contextual privacy violations arise in many common choice rules.

3.2 Collective and Individual Pivotality.

The main result in this section allows us to identify the type profiles at which contextual privacy violations arise. This result shows if there is a type profile where agents are collectively pivotal, but no agent is individually pivotal, then there will be a violation for some agent at that type profile.

Theorem 1. *Let ϕ be a social choice function, and consider a type profile $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. If for any subset $A \subseteq N$ of agents, and types $\theta_i' \in \Theta$ for agents $i \in A$,*

$$\phi(\boldsymbol{\theta}_A, \boldsymbol{\theta}_{-A}) \neq \phi(\boldsymbol{\theta}'_A, \boldsymbol{\theta}_{-A}) \qquad \text{(collective pivotality)}$$

and for all $i \in A$

$$\phi(\theta_i, \boldsymbol{\theta}_{-i}) = \phi(\theta_i', \boldsymbol{\theta}_{-i}) \qquad \text{(no individual pivotality)}$$

then for any individual elicitation protocol $P \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$, there exists an agent $i \in A$ whose contextual privacy is violated at $\boldsymbol{\theta}$, i.e. $A \times \{\boldsymbol{\theta}\} \cap \Gamma(P, \phi) \neq \emptyset$.

This theorem tells us that a social choice function will produce a contextual privacy violation under any $\mathfrak{S}_{\text{IE}}$-protocol at a type profile $\boldsymbol{\theta}$ if there is a set of agents that *collectively* are pivotal for the outcome, but none of them is individually pivotal. This is intuitive, as the individual elicitation technology needs to query one agent at a time. So, a situation of collective pivotality without individual pivotality suggests that the designer must ask some agent a question that may not matter for the outcome, but had to ask that question in case it did matter.

In the remainder of this section, we discuss instances in which collective pivotality vs. individual pivotality arise in common choice rules. We consider specific choice rules in three domains: assignment, auctions and voting.

3.2.1 Application to Assignment.

In the assignment domain, we fix a set C of objects. The set of outcomes is $X = 2^{N \times C}$.

In the standard object assignment setting, agents may receive at most one object, and agents have ordinal preferences over objects, which are private information. So agents' types $\boldsymbol{\theta} \in \Theta$ are preference orders on C where $\succeq_i$ refers to agent i's preference ordering.

Table 2: Contextual Privacy Violations of Choice Rules Under Individual Elicitation Technology

Choice Rule ϕ	$\inf_{P \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}} \Gamma(P, \phi)$	
Assignment (Section 3.2.1)		
Serial Dictatorship Rule	$= \emptyset$	Appendix B.1
Efficient and IR Rules (House Assignment)	$\neq \emptyset$	Proposition 3
Stable Rules (Matching with Priorities)	$\neq \emptyset$	Proposition 4
Auctions (Section 3.2.2)		
First-price Auction Rule	$= \emptyset$	Appendix B.2
Second-price Auction Rule	$\neq \emptyset$	Proposition 5
Efficient Double Auction	$\neq \emptyset$	Proposition 6
Voting (Section 3.2.3)		
Generalized Median Voting Rule	$\neq \emptyset$	Proposition 7

In the classic *serial dictatorship* social choice function, individual and collective pivotality coincide, and allow for a protocol without any contextual privacy violation. We show in Appendix B.1 that the serial dictatorship (with deterministic lexicographic tie-breaking) is contextually private.[14] It is intuitive that the serial dictatorship does not produce a conflict between collective and individual pivotality—it is impossible to be collectively pivotal without being individually pivotal, since one agent must choose a different object to lead to different allocation.

Other object assignment choice rules fail to be contextually private because of a conflict between individual and collective pivotality. Consider first the house assignment problem Shapley and Scarf (1974). All agents are initially endowed with an object from C. Denote the initial assignment by an injective function $e\colon N \to C$, where $e(i) \in C$ refers to agent i's initial endowment. For our result it will be irrelevant whether the endowments are private information or known to the designer. We call a choice rule ϕ *individually rational* if for all $i \in N$ and all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$,

$$\phi_i(\boldsymbol{\theta}) \succeq_i e(i).$$

Proposition 3. *Assume $|N| \geq 2$. Every protocol $P \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$ for an individually rational and efficient housing assignment choice rule ϕ produces contextual privacy violations at any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ in which two agents prefer their endowment over the other agent's endowment.*

Proof. The proof shows that there is collective but not individual pivotality. Consider two agents i and j and two possible preference profiles for each agent. For agent i, consider a type θ_i which contains $e(i) \succ_i e(j)$, and a type θ_i' which contains $e(j) \succ_i e(i)$. For agent j, consider θ_j which

[14] In fact, serial dictatorships satisfy two even stronger notions of privacy protection than contextual privacy. They satisfy *individual* and *group* contextually privacy, as discussed in Appendix E.

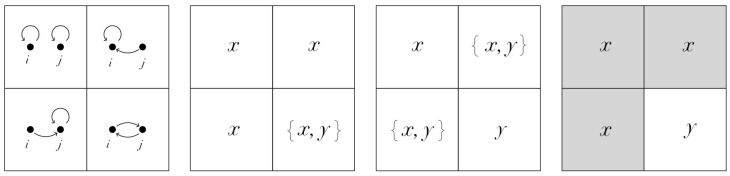


Figure 2: Constructing collective but not individual pivotality for house assignment. Type profiles $\{\theta_i, \theta_i'\} \times \{\theta_j, \theta_j'\}$ used in the proof (left, arrows denote whether agent prefers own or other's endowed object); required outcomes for each type profile under individual rationality (mid-left), under efficiency (mid-right), and under both efficiency and individual rationality (right).

contains $e(j) \succ_j e(i)$, and a type θ_j' which contains $e(i) \succ_j e(i)$. Hold fixed all other types $\boldsymbol{\theta}_{-ij}$, to be such that they prefer their own endowment to all other objects, i.e. $\boldsymbol{\theta}_{-ij} = (e(k) \succeq_k c \text{ for all } c \in C \setminus \{e(k)\})_{k \in N \setminus \{i,j\}}$.

When the type profile is $(\theta_i, \theta_j, \boldsymbol{\theta}_{-ij})$, the agents both prefer their own endowment to the other's. When the profile is $(\theta_i', \theta_j, \boldsymbol{\theta}_{-ij})$, or $(\theta_i, \theta_j', \boldsymbol{\theta}_{-ij})$, they both prefer i's endowment and j's endowment, respectively. When $(\theta_i', \theta_j', \boldsymbol{\theta}_{-ij})$, they each prefer the other's endowment to their own. Let x be the outcome in which both agents retain their endowment, i.e. $x = (i, e(i)), (j, e(j))$. Let y be the outcome in which each agent gets each other's endowment $y = (i, e(j)), (j, e(i))$. Then, individual rationality makes the requirements shown on the mid-left in Figure 2: $\phi(\theta_i, \theta_j, \boldsymbol{\theta}_{-ij}) = (\theta_i, \theta_j', \boldsymbol{\theta}_{-ij}) = (\theta_i', \theta_j, \boldsymbol{\theta}_{-ij}) = x$. Meanwhile, efficiency requires $\phi(\theta_i, \theta_j, \boldsymbol{\theta}_{-ij}) = y$ (shown on the mid-right in Figure 2). Hence, by Theorem 1, every individual elicitation protocol for an individually rational and efficient choice rule produces contextual privacy violations. □

The failure of contextual privacy in the house assignment problem is illuminating. There will be a violation at any type profile $\boldsymbol{\theta}$ in which there is a pair of agents who each prefer their endowment to the other's. Note that in a setting with many agents and many objects, there are many such type profiles $\boldsymbol{\theta}$ in $\boldsymbol{\Theta}$. In such cases, the conjunction of efficiency and individual rationality produce an instance where the agents are collectively but not individually pivotal.

In two-sided matching, we see a similar failure mode for contextual privacy and stable outcomes under individual elicitation, but here it is the requirement of stability that produces a conflict between collective and individual pivotality. In two-sided matching, every agent (also called "student") is matched to at most one object (also called "school"), and at most $\kappa(c)$ agents are matched to an object c, for every $c \in C$, for some *capacities* $\kappa(c)$. That is, the set of outcomes is

$$X = \{\mu \subseteq N \times C \colon \ \forall i \in N \colon |\{c \in C \mid (i,c) \in \mu\}| \leq 1 \text{ and } \forall c \in C |\{i \in N \colon (i,c) \in \mu\}| \leq \kappa(c)\}.$$

We say there is *no oversupply* if the aggregate capacity equals the number of agents, $\sum_{c \in C} \kappa(c) = n$.

We assume that object's preferences over agents are determined by agents' *priority scores*. We

assume the scores for different objects are private information of the agents. This matches the *college assignment problem* with standardized test scores (Balinski and Sönmez, 1999). We assume that each agent has a vector of *scores* score_c, representing their score at each object $c \in C$. Objects prefer agents with higher scores. Agent i has private information $\theta_i = (\preceq_i, \text{score}_i)$, where $\preceq_i$ is i's preference ranking over schools, and $\text{score}_i : C \to \mathbb{R}$ maps objects to scores.

In such school choice settings, a desirable property of choice rules is *stability* (or does not induce *justified envy*). A choice rule ϕ is *stable* or *induces no justified envy* if there is no blocking pair (i, c), $i \in N$, $c \in C$ such that $c \succ_i \phi_i(\boldsymbol{\theta})$ and $\text{score}_i(c) > \text{score}_i(\phi_i(\boldsymbol{\theta}))$.

Proposition 4. *Assume $|C| \geq 2$ and $|N| \geq 2$ and that there is no oversupply. Then any protocol $P \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$ for a stable choice rule produces contextual privacy violations.*

This proposition highlights one way in which stability produces contextual privacy violations. Whether an agent has justified envy of another student is a collective feature of both agents' types. A single student's change in score may lead to a new justified claim of another student to their seat in this school, yet no envy. A change in this other student's preference may lead to envy which is not justified. Yet both changes together lead to justified envy, and force collective pivotality in these changes in type.

We can also see, for the school choice problem, how common protocols will produce contextual privacy violations. Consider for example the deferred acceptance protocol, which produces a stable outcome. The designer must learn information about tentative assignments that may not be final, and in so doing violates contextual privacy.

3.2.2 Application to Auctions.

We next consider how single-item auctions and double auctions produce contextual privacy violations.

Consider a standard private values auction environment in which a single indivisible item is to be allocated to one agent. Agent types are real numbers $\Theta \subseteq \mathbb{R}_+$. The outcomes are given by $x = (q_i, t_i) \in X$, with $q_i \in \{0, 1\}$, $t_i \in \mathbb{R}$, where q_i is agent i's allocation (with $q_i = 1$ meaning that the agent receives the object), and t_i is their payment. Agents have quasilinear preferences defined by

$$u_i((q, t); \theta_i) = \theta_i q_i - t_i, \tag{2}$$

We call a single-item auction choice rule *standard* if there is at most one agent $i \in N$ such that $t_i \neq 0$, and $q_i = 1$. We call an auction *efficient* if it maximizes agent welfare, i.e.

$$\phi(\boldsymbol{\theta}) \in \text{argmax}_{(q(\boldsymbol{\theta}),\, (\boldsymbol{\theta}))} \sum_{i \in N} u_i((q(\boldsymbol{\theta}), t(\boldsymbol{\theta})), \theta_i)$$

for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. That is, the highest valuation agent wins.

The two most widely studied standard auction rules are the first-price and the second-price auction. As this article considers deterministic mechanisms, we consider these rules with deterministic lexicographic tiebreaking. The *first-price auction* is a choice rule $\phi^{\mathrm{FP}}(\boldsymbol{\theta}) = (\phi_1^{\mathrm{FP}}(\boldsymbol{\theta}), \ldots, \phi_n^{\mathrm{FP}}(\boldsymbol{\theta})) = ((q_1, t_1), \ldots, (q_n, t_n))(\boldsymbol{\theta})$, where

$$\phi_i^{\mathrm{FP}}(\boldsymbol{\theta}) = \begin{cases} (1, b_i(\theta_i))) & \text{if } b(\theta_i) = \min \operatorname{argmax}_{j \in N} b(\theta_j) \\ (0,0) & \text{otherwise.} \end{cases}$$

for some monotonic bid function $b_i \colon \boldsymbol{\theta}_i \to \boldsymbol{\theta}$. The *second-price auction* is a choice rule $\phi^{\mathrm{SP}}(\boldsymbol{\theta})$, where

$$\phi_i^{\mathrm{SP}}(\boldsymbol{\theta}) = \begin{cases} (1, \theta_{[2]}) & \text{if } i = \min \operatorname{argmax}_{j \in N} \theta_j \\ (0,0) & \text{otherwise,} \end{cases}$$

where $\theta_{[2]}$ is the second-highest type in the type profile $\boldsymbol{\theta}$. Both of these auction rules can be computed via a number of different protocols. Commonly studied protocols for the first-price and second-price rules, respectively, include the descending ("Dutch") protocol and the ascending ("English") protocol.

The first-price auction does not at any point lead to an instance of collective pivotality without individual pivotality—only the winner determines the outcome. Indeed, there is a protocol for the first-price auction rule that produces no contextual privacy violations. In particular, the familiar descending or "Dutch" protocol is fully contextually private. We present and discuss this result in Appendix B.2. The intuition for this result is very similar to the intuition behind the result that the serial dictatorship is contextually private. The designer begins at the top of the type space and asks questions of the form "Is your type above $\tilde{\theta}$?" As soon as one agent answers in the affirmative at some $\tilde{\theta}$, the protocol ends and assigns the object to the agent who responded affirmatively and the price is set at $t = \tilde{\theta}$. The designer thus only elicits information that directly changes the outcome.

Proposition 5. *Assume $|\Theta| \geq 3$ and $|N| \geq 3$. Any protocol $P \in \mathcal{P}_{\mathfrak{S}_{\mathrm{IE}}}$ for the second-price auction choice rule produces contextual privacy violations for $\boldsymbol{\theta}$ at which exactly two agents hold the second-highest type.*

Proof. The proof constructs an instance of collective but not individual pivotality. Consider a type profile with $\theta_{[1]} > \theta_{[4]}$ (or no constraint if $n = 3$), and consider agents i, j that have types in $[\underline{\theta}, \bar{\theta}]$ such that $\theta_{[4]} < \underline{\theta} < \bar{\theta} < \theta_{[1]}$. Consider the product set $\{\underline{\theta}, \bar{\theta}\} \times \{\underline{\theta}, \bar{\theta}\} \times \bigtimes_{k \in N \setminus \{i,j\}} \theta_k$. This corresponds to a square depicted in Figure 3.

Let x be the outcome in which the highest type wins ($q_i = 1$ for $\theta_i = \theta_{[1]}$, $q_i = 0$ otherwise) and pays the price $t_i = \bar{\theta}$. Let x' be the outcome under which the highest type wins and pays the price $t_i = \underline{\theta}$. Then, $\phi^{\mathrm{SPA}}(\bar{\theta}, \bar{\theta}, \boldsymbol{\theta}_{-ij}) = \phi^{\mathrm{SPA}}(\underline{\theta}, \bar{\theta}, \boldsymbol{\theta}_{-ij}) = \phi^{\mathrm{SPA}}(\bar{\theta}, \underline{\theta}, \boldsymbol{\theta}_{-ij}) = x$. But, under ϕ^{SPA}, it must

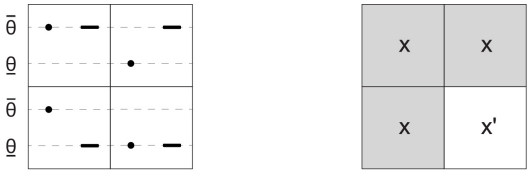


Figure 3: Showing collective but not individual pivotality to the second-price auction. Type profiles for agent i and agent j (left, agent j's type is represented by a dot, and agent i's type is represented by a dash); required outcome under the second-price auction rule ϕ^{SPA}.

be the case that $\phi(\underline{\theta}, \underline{\theta}, \boldsymbol{\theta}_{-ij}) = x'$. Since $x \neq x'$, the is violated, and thus the second-price choice rule is not contextually private under individual elicitation. □

This result is intuitive: when two agents redundantly determine the price, they are collectively pivotal but not individually pivotal. We show in Appendix A.5 that there are contextual privacy violations even if ties are ruled out (for $|N| \geq 3$ and $|\Theta| \geq 9$).

A similar result holds for standard double auction rules. Suppose m agents are buyers and m agents are sellers, and $n = 2m$. The m sellers are each endowed with one homogeneous, indivisible object. The buyers have unit demand for objects. Formally, agents have initial endowments $e_i \in \{0, 1\}$ where $e_i = 0$ for buyers and $e_i = 1$ for sellers. The preferences are

$$u_i((q, t), \theta) = -e_i \theta_i q_i + (1 - e_i \theta_i) q_i + t_i.$$

A double auction price rule seeks to find a price t that maximizes $\sum_{i \in N} u_i((q, t); \theta_i)$ if buyers with types $\theta_i \geq t$ buy a good at price t, and sellers with value $\theta_i \leq t$ sell their good at price t. Agents with $\theta_i = t$ sell or buy in order to match supply to demand.

Proposition 6. *Assume $|\Theta| \geq 4$ and $|N| \geq 4$. Any protocol $P \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$ for an efficient, uniform-price double auction price rule produces contextual privacy violations for $\boldsymbol{\theta}$ where two agents determine the clearing price.*

The intuition for this result is similar to the result for the second-price auction rule. The agents that, together, determine the clearing price are jointly pivotal but not individually so. The proof of this statement requires additional care because the median of an even number of types is not uniquely determined.

3.2.3 Application to Voting and Information Aggregation.

We next turn to a voting environment. The outcome space is ordered, $(X, \leq)$. The preference domain for all agents are the single-peaked preferences with respect to $\leq$.[15]

[15] A preference $\preceq$ on $(X, \leq)$ is single-peaked if $x < x' \leq \theta_i \implies x' \succ_i x$ and $x > x' \geq \theta_i \implies x' \succ_i x$.

We consider a commonly studied class of voting rules. Namely, we study *generalized median voting rules*. As shown in Moulin (1980), this class is the class of all anonymous, strategy-proof and Pareto-efficient voting rules, where anonymity means that the outcome cannot depend on the identity of any agent. A generalized median voting rule takes as input submitted peaks of agents' preferences $\theta_1, \theta_2, \ldots \theta_n$ as well as *phantom ballots* $k_1, k_2, \ldots, k_{n-1} \in X \cup \{-\infty, \infty\}$, where for all $x \in X$, $-\infty < x < \infty$. The output is the median of the submitted votes and phantom votes, i.e.

$$\phi_{(k_1, k_2, \ldots, k_{n-1})}(\boldsymbol{\theta}) = \text{median}(\theta_1, \theta_2, \ldots, \theta_n, k_1, k_2, \ldots, k_{n-1}).$$

Proposition 7. *Assume $|N| \geq 2$ and $|X| \geq 2$ and the preference domain of single-peaked preferences. Any protocol $P \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$ for a generalized median voting rule that is neither the maximum nor the minimum rule produces contextual privacy violations.*

In such voting rules, classical examples of collective but not individual pivotality arise from voting thresholds, e.g., a qualified majority. Consider a voting rule with at least two phantom ballots k_i on an alternative x that would win under a type profile $\boldsymbol{\theta}$. If there are two agents on the right of this alternative, and *both* of them having a type on the left of x, it would change the outcome—they are *collectively pivotal*—and yet it is still possible that neither of them alone would change the outcome.

4 Maximal Contextual Privacy

In the previous section, we focused on how specific properties of the social choice rule lead to contextual privacy violations under individual elicitation technologies. In this section, we take the perspective of a privacy-conscious designer who needs to choose some protocol for a given choice rule. Such a designer must make choices about whose privacy to protect, and at what type profiles. They would like to find protocols that are maximally contextually private—i.e. protocols that implement the choice rule, and are maximal elements in the contextual privacy order.

The results of the previous section illuminate a key dimension of the designer's choice—when there is a conflict between collective pivotality and individual pivotality, the designer has a choice of whom to query first, and whose privacy to protect. Those agents who are queried first have their privacy violated. We will see that maximally contextually private protocols protect the privacy of a set of agents i by delaying questions to them as long as possible, so that the designer learns as much as possible about $\boldsymbol{\theta}_{-i}$ before getting to agent i. Note that if everything relevant to the choice rule is revealed about $\boldsymbol{\theta}_{-i}$ before asking anything to agent i, agent i will be asked only what is needed, and will not have a contextual privacy violation.

So, maximally contextually private protocols delay asking questions to protected agents. But how does the designer choose which agents to protect? Maximally contextually private protocols protect different sets of agents and highlight the normative tradeoffs that the designer must consider in choosing a protocol. For example, the two maximally contextually private protocols for the second-price auction rule we derive in Theorem 3, the ascending-join and the overdescending join protocols, protect the winner and the losers, respectively. In some cases, such as the 2017 incentive auction described in Milgrom and Segal (2020), it will be more important to protect the winner, while in other settings, such as in repeated interactions like Google's second-price auctions for advertising, it may be more important to protect the losers.

Theorem 3 is the most design-relevant result of the paper, and illustrates the power of the contextual privacy perspective we have developed (the maximally contextually private protocols we derive have not, to our knowledge, been described before). As the theorem's proof illuminates, reasoning about maximally contextually private protocols can be very difficult due to the vast combinatorial space of protocols. In the proof of Theorem 3, we rely on a representation theorem, Theorem 2, that puts structure on the space of candidate maximally contextually private protocols. The result shows that for choice rules defined on ordered type spaces, it is without loss, from a contextual privacy perspective, to consider only *bimonotonic* protocols. In a *bimonotonic* protocol, every query is a threshold query, and every agent is asked a sequence of queries that is either monotonically ascending or monotonically descending in the thresholds. This tells us that the two key levers of design, from a contextual privacy perspective, are (i) the initial threshold at which to query agents, and (ii) the order in which to query agents.

We begin in Section 4.1 with the representation theorem (Theorem 2), and then leverage the representation theorem to derive maximally contextually private protocols for the second-price auction rule (Theorem 3) in Section 4.2.

4.1 A Representation Theorem.

The goal of this subsection is to present our representation theorem (Theorem 2) for contextual privacy equivalence classes under the restriction to individual elicitation protocols. The theorem shows that a privacy-concerned designer can, without loss, restrict attention to *bimonotonic* protocols under the restriction to individual elicitation protocols $\mathfrak{S}_{\text{IE}}$. This reduces the designer's levers of design: for each agent they choose an initial threshold, and the order in which to query agents.

First, we formally define bimonotonicity. We will assume that message spaces M have at least two elements, and hence protocols can compute some non-constant social choice functions. In every query of a bimonotonic protocol, an agent sends one of two messages m, m' depending on whether

their type is larger than a *threshold* $\tilde{\theta}$,

$$\sigma_i(v, \theta_i) = \begin{cases} m & \text{if } \theta_i > \tilde{\theta} \\ m' & \text{if } \theta_i \leq \tilde{\theta}. \end{cases}$$

That is, a query in a *threshold protocol* asks an agent for whether their type falls strictly above some value $\tilde{\theta} \in \Theta$. A bimonotonic protocol is a protocol in which all queries are threshold queries, and the thresholds of threshold queries to a single agent i form an increasing or decreasing interval in the type space Θ.

Definition (Bimonotonic Protocol). A threshold protocol $(P, \boldsymbol{\sigma})$ is *bimonotonic* if for any path in P and for all agents $i \in N$, the sequence of thresholds $\tilde{\theta}$ presented to agent i form an increasing or decreasing interval in the type space Θ.

The representation theorem will show that for many choice rules of interest, any $\mathfrak{S}_{\text{IE}}$-protocol is contextual privacy equivalent to a bimonotonic protocol. The condition for this reduction is that the choice rule of interest must be injective over exactly one interval inside of the type space, a property which we call *interval pivotality.*

Definition (Interval Pivotality). A social choice function ϕ defined on an ordered type space Θ exhibits *interval pivotality* if for all $\boldsymbol{\theta}_{-i} \in \boldsymbol{\Theta}_{-i}$ there are elements $\underline{\theta} \in \Theta$ and $\overline{\theta} \in \Theta$ such that

$$\phi(\theta_i, \boldsymbol{\theta}_{-i}) = \phi(\theta_i', \boldsymbol{\theta}_{-i}) \iff (\theta_i, \theta_i' \leq \underline{\theta} \text{ or } \theta_i, \theta_i' \geq \overline{\theta}).$$

This property of choice rules states that, holding fixed some profile of other agents' types $\boldsymbol{\theta}_{-i}$, the choice rule $\phi(\theta_i, \boldsymbol{\theta}_{-i})$ is constant in θ_i if and only if θ_i is outside some interval $[\underline{\theta}, \overline{\theta}] \in \Theta$. So, inside the interval, i is "pivotal" in that her report changes the outcome. Note in this definition that the interval defined by $[\underline{\theta}, \overline{\theta}]$ may depend on the profile of other agents' types, $\boldsymbol{\theta}_{-i}$.[16]

Many social choice rules of interest are interval pivotal. For example, all auction and voting social choice functions considered in Section 3.2 are interval pivotal. Note, however, that interval pivotality only applies to choice rules defined on ordered type spaces, so assignment and matching rules cannot be interval pivotal. And not all choice rules on ordered type spaces are interval pivotal. For example, the third price auction is not interval pivotal—there are two disjoint intervals of θ_i over which the choice rule $\phi(\theta_i, \boldsymbol{\theta}_{-i})$ is changing (when θ_i is setting the price, and when θ_i is determining the winner).

Now that we have defined bimonotonicity and interval pivotality, we can present the main result of this subsection, and indeed a central theoretical insight of the paper. This representation theorem simplifies the designer's search for privacy-preserving protocols when choosing among individual

[16] We suppress subscripts to simplify notation. Formally, the interval is $[\underline{\theta}_{(i,\boldsymbol{\theta}_{-i})}, \overline{\theta}_{(i,\boldsymbol{\theta}_{-i})}]$.

elicitation protocols for an interval pivotal choice rule: the designer need only consider bimonotonic protocols, because every $\mathfrak{S}_{\text{IE}}$-protocol is contextual privacy equivalent to a bimonotonic protocol.

Theorem 2. *Let $(P, \boldsymbol{\sigma})$ be a $\mathfrak{S}_{\text{IE}}$-protocol for ϕ, where ϕ exhibits interval pivotality. There is a bimonotonic protocol $(P', \boldsymbol{\sigma}')$ that has the same set of contextual privacy violations as $(P, \boldsymbol{\sigma})$, i.e. $\Gamma(P, \boldsymbol{\sigma}, \phi) = \Gamma(P', \boldsymbol{\sigma}', \phi)$.*

In other words, every contextual privacy equivalence class has a bimonotonic representative. The proof of Theorem 2 proceeds by considering an arbitrary protocol and applying two modifications to it. These modifications transform the arbitrary protocol into a bimonotonic one, while preserving the set of contextual privacy violations. The first transformation "fills in" threshold queries for any threshold that lies between any types that are distinguished. Second, a "scrubbing" operation keeps only a monotonic sequence of threshold queries. It is not hard to see that the result of these operations will be a bimonotonic protocol. What is less obvious is why contextual privacy violations do not change. Interval pivotality is crucial here—it implies that when filling in gaps between threshold queries, the protocol is either filing in inside of the "pivotal interval" $[\underline{\theta}, \overline{\theta}]$ (in which case no violations are introduced because the agent is changing the outcome) or outside the pivotal interval (in which case there was already a violation).

Several common protocols are bimonotonic: The ascending protocol for the second-price auction, the descending protocol for the first-price auction, and the overdescending protocol for the second-price auction. For voting rules, one-sided protocols are bimonotonic: First, query all agents whether they are below the right-most alternative, then query the next-to-rightmost alternative, and so forth.

The fact that every equivalence class has a bimonotonic representative emphasizes that the decisions a designer makes about contextual privacy amount to: (i) the threshold of the initial query to each agent, and (ii) the order in which the designer asks agents these threshold queries. So, in searching for maximally contextually private protocols in particular, it is without loss to focus on these two dimensions. We leverage this theorem in the next subsection.

4.2 Two Maximally Contextually Private Protocols for the Second-Price Auction Rule.

Next, we use the representation theorem to guide design: we find maximally contextual private protocols for the second-price auction rule. We will derive two novel protocols, *ascending-join* and *overdescending-join* protocols, that are maximally contextually private.

We begin by formalizing the idea of a protection set, i.e. some set of agents and type profiles $A \subseteq N \times \boldsymbol{\Theta}$ that do not have privacy violations under P.

Definition. We say that a protocol P *protects* $A \subseteq N \times \boldsymbol{\Theta}$ if

$$\Gamma(P, \phi) \cap A = \emptyset.$$

Denote the set of $\mathfrak{S}$-protocols that protect A by $\mathcal{P}^A_{\mathfrak{S}}$.

Protection sets are helpful constructs for communicating the contextual privacy properties of particular protocols. For example, Milgrom and Segal (2020) study protocols that preserve *unconditional winner privacy* which is analogous, in our formalism, to protocols that protect only the contextual privacy of winners. We will use protection sets that also include losers.

The following observation helps us in characterizing maximally contextually private protocols. It is direct from the definition of the contextual privacy order defined by inclusion of the set of contextual privacy violations. It says simply that if a protocol is maximally contextually private among a subset of $\mathfrak{S}$-protocols that protect a subset A, i.e. among $(\mathcal{P}^A_{\mathfrak{S}}, \preceq_\phi)$, then that protocol is also maximally contextually private among all $\mathfrak{S}$-protocols.

Lemma 1. *Assume that P is maximal in $(\mathcal{P}^A_{\mathfrak{S}}, \preceq_\phi)$. Then, it is maximal in $(\mathcal{P}_{\mathfrak{S}}, \preceq_\phi)$.*

We will use Lemma 1 to show maximality of protocols for ϕ^{SPA}. We define specific protection sets A and characterize maximally contextually private protocols within $\mathcal{P}^A_{\mathfrak{S}}$, as it is easier to prove maximal contextual privacy within $\mathcal{P}^A_{\mathfrak{S}}$ than within $\mathcal{P}_{\mathfrak{S}}$ directly.

The protocols we show to be maximally contextually private protect specific agent-type profile pairs $(i, \boldsymbol{\theta})$. In the definition of these sets, tiebreaking requires extra care. We call an agent a *winner* if she is the highest (in lexocographic order) priority, highest-type agent (as usual). We say that an agent *determines the price* if she is the highest priority, second-highest type agent (or the second-highest priority, highest type agent in the case of a tie). If the agent neither determines the price nor is a winner, then she is a *loser*. These definitions give rise to the winner and loser protection sets, W and L .

$$\begin{aligned} \text{W} &= \{(i, \boldsymbol{\theta}) \mid i \text{ is the winner at } \boldsymbol{\theta}\} \\ \text{L} &= \{(i, \boldsymbol{\theta}) \mid i \text{ is neither the winner nor determines the price at } \boldsymbol{\theta}\}. \end{aligned}$$

We also define subsets of the winner and loser protection sets. Let $d(\boldsymbol{\theta})$ be the index of the agent who determines the price at type profile $\boldsymbol{\theta}$. We define the protection sets LPL and HPW as follows:

$$\begin{aligned} \text{LPL} &= \text{L} \cap \{(i, \boldsymbol{\theta}) \mid i > d(\boldsymbol{\theta})\} \\ \text{HPW} &= \text{W} \cap \{(i, \boldsymbol{\theta}) \mid i < d(\boldsymbol{\theta})\}. \end{aligned}$$

Low-priority losers are losers whose index i is greater than the index of the agent with the second-highest type. Note that agents with high indices are "low priority" because of lexicographic tiebreaking. High-priority winners are winners whose index is lower than the index of (or have higher priority than) the agent with the second-highest type.

The next theorem states that the maximally contextually private protocols are simple. that

there is a unique protocol for the second-price auction rule that protects the winner and low-priority losers, and a unique protocol for the second-price auction rule that protects the losers and high-priority winners. These two protocols are maximally contextually private.

Theorem 3. *For the second-price auction rule ϕ^{SPA}, there is:*

(a) a unique minimal set of contextual privacy violations for W- *and* LPL*-protecting $\mathfrak{S}_{\text{IE}}$-protocols,*
(b) a unique minimal set of contextual privacy violations for L- *and* HPW*-protecting $\mathfrak{S}_{\text{IE}}$-protocols.*

Two bimonotonic protocols attain these sets of contextual privacy violations, which we call the ascending-join protocol *and the* overdescending-join protocol*, respectively. Both of these protocols are maximally contextually private among $\mathfrak{S}_{\text{IE}}$-protocols for ϕ^{SPA}.*

The basic idea of the result is that the designer delays asking questions to the agents in the protection set as much as possible, so that on type profiles where it is possible to avoid violating their privacy, the designer will avoid violating their privacy. The two protocols we derive are variants of more common protocols. The ascending-join protocol is related to the standard ascending or "English" auction protocol that begins with queries at the bottom of the type space and stops when only one agent remains, and the overdescending-join protocol is related to the overdescending protocol (see Harstad (2018)) which begins at the top of the type profile, and descends until the second-highest value is found.

The proof begins with the restriction to bimonotonic protocols via Theorem 2. Recall that the restriction to bimonotonic protocols reduces the designer's problem to choices around (i) the order in which to query agents and (ii) the initial threshold query to pose to each agent. We show that at every point in the protocol, there is exactly one option for (i) a next agent to ask, and (ii) the threshold strategy for this agent that maintains the privacy of the protection sets in the hypothesis.

We give further intuition for the ascending-join protocol here, and refer readers to the appendix for further understanding of the overdescending-join protocol. In the proof of Theorem 3, we show that any W $\cup$ LPL protecting protocol can be understood in "rounds". We show that after each such round but the last, the designer's knowledge is a *protective information state* $\boldsymbol{\Theta}_v$. We go on to show that from protective information states, W $\cup$ LPL protection demands a unique sequence of "non-redundant" queries, which we show to imply the result.

Here we give further intuition for the information state reached at the end of a round. Any W- and LPL-protective information state has a running price $\tilde{\theta}$ and an index of the most recently "joined" agent l such that: (i) nothing is known about agents with index higher than l (*idle agents*); (ii) there are exactly two agents for whom it is exactly known that their type is above $\tilde{\theta}$ (*active agents*); and (iii) all other agents are known to have a type below $\tilde{\theta}$ (*dropped-out agents*). These requirements are illustrated in Figure 4. With this definition of a protective information state, we then show that at every possible stage of the protocol, there is exactly one query that the designer can make that leads into either another protective information state or termination.

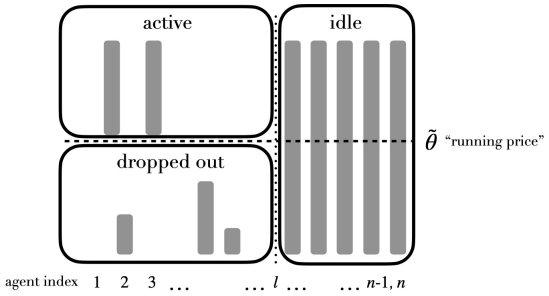


Figure 4: W- and LPL-Protective Information State.

The ascending-join protocol can roughly be understood as follows. The designer wants to maintain exactly two active agents at any point in time, until termination. The protocol begins with asking the second-highest priority agent (agent with index 2) the lowest threshold query. When the designer gets an affirmative answer for some threshold query, they next ask an agent with higher priority than the queried agent the same threshold query. When the designer gets a negative answer, they ask an agent of lower priority next.

To illustrate what the ascending-join protocol actually looks like in practice, we turn to a simple example with four agents.

Example (Ascending-Join Protocol for the Second-Price Auction Rule). Consider a set of four agents $|N| = 4$ and a type space of the integers from 1 and 10, i.e. $\Theta = \{1, 2, \ldots, 10\}$. Suppose the true type profile is $\boldsymbol{\theta} = (4, 3, 8, 2)$.

For this type profile, the winner is agent 3, who has value $\theta_3 = 8$. There is a single low-priority loser, agent 4 who has value $\theta_4 = 2$. The ascending-join protocol for this type profile is spelled out in Table 3, and illustrated (with a comparison to the ascending protocol) in Figure 5.

Note in the ascending-join protocol that at every stage except the terminal stage, the designer is in or leading into a *protective information state*: there is a running price $\tilde{\theta}$, there are two active agents who are known to have types above $\tilde{\theta}$ (the agents who say yes for each threshold), any agents who have dropped out have indices to the right of an active agent (e.g. when agent 2 drops out at running price $\tilde{\theta} = 3$, she is to the right of agent 1, who is active) and there is a set of idle agents about whom nothing is known (agent 4 is idle until the final stage). The winner (agent 2) and the lowest-priority loser (agent 4) do not have contextual privacy violations. Compare this to the ascending protocol, which additionally violates the contextual privacy of agent 4, as shown in Figure 5.

The constructive proof strategy of Theorem 3 can be understood through this example. The basic idea is that all queries must lead into a protective information state or termination, and there is a unique sequence of queries that do so. In this sequence, we show in a lemma that it must

$\tilde{\theta} = 1$	Query agent 2, "Is your type above 1?" Agent 2 answers **yes**.
	Query agent 1, "Is your type above 1?" Agent 1 answers **yes**.
$\tilde{\theta} = 2$	Query agent 2, "Is your type above 2?" Agent 2 answers **yes**.
	Query agent 1, "Is your type above 2?" Agent 1 answers **yes**.
$\tilde{\theta} = 3$	Query agent 2, "Is your type above 3?" Agent 2 answers no.
	Query agent 3, "Is your type above 3?" Agent 3 answers **yes**.
	Query agent 1, "Is your type above 3?" Agent 1 answers **yes**.
$\tilde{\theta} = 4$	Query agent 3, "Is your type above 4?" Agent 3 answers **yes**.
	Query agent 1, "Is your type above 4?" Agent 1 answers no.
	Query agent 4, "Is your type above 4?" Agent 4 answers no.
The protocol terminates, allocates the object to agent 2 at price $\tilde{\theta} = 4$.	

Table 3: Example: Ascending-Join Protocol for $\boldsymbol{\theta} = (4, 3, 8, 2)$.

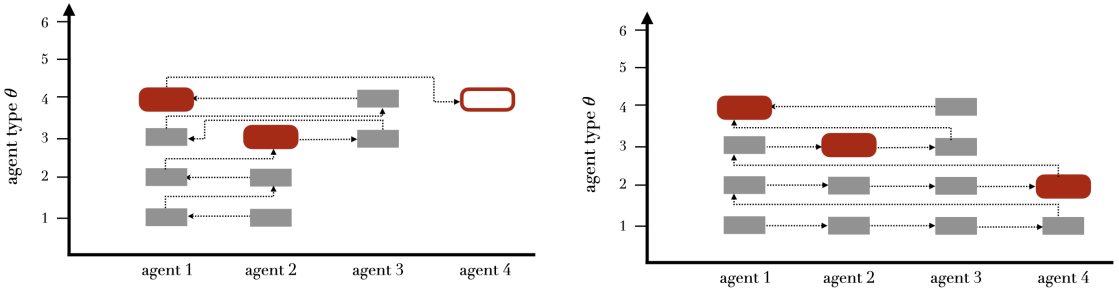


Figure 5: Example—Ascending-Join (left) vs. Ascending (right) Protocol for $\boldsymbol{\theta} = (4, 3, 8, 2)$. Queries to an agent are represented as rectangles, positioned at the corresponding threshold $\tilde{\theta}$ (gray solid rectangles represent affirmative responses, red rounded rectangles represent negative responses and a contextual privacy violation, red unfilled rectangles represent negative responses without a contextual privacy violation). The path of the dotted line represents the order in which queries are asked. In both protocols, the winner (agent 3) does not have their privacy violated. In the ascending-join protocol, the low-priority loser (agent 4) also does not have their privacy violated. Theorem 3 shows that the ascending-join protocol is $\mathfrak{S}_{\text{IE}}$-maximally contextually private.

never be the case that a query to a particular agent could land them in the protection set *regardless of their answer*. That is, no agent i can never be asked a query at node v that could, for some $\boldsymbol{\theta}_{-i} \in \text{children}(\boldsymbol{\Theta}_v)$, lead to $i \in W \cup \text{LPL}$ regardless of i's answer to the query at v.

This lemma is very powerful. To see how we use it to construct the protocol, consider the initial query in the example. We know from bimonotonicity that we must determine the initial query's threshold, and the agent to ask it to. First off, we know that the initial query must have a threshold $\tilde{\theta} = 1$, asking about the lowest type, because otherwise the winner's privacy could be violated. The more interesting part is: to whom does the designer ask the first query? The initial query must go

to agent 2. To see this, note that if the designer asked agent 1 whether her type is above $\tilde{\theta} = 1$, there are cases in which she ends up a winner regardless of her answer. If she answers "no" (i.e. she has the lowest possible value) and it turns out that all other agents have the lowest possible value, she wins the object, and at the same time, if she answers "yes" (i.e. she has a value above the lowest), and everyone else has the lowest value, then again agent 1 wins the object. So, regardless of agent 1's answer, she could end up in the protection set, and so we cannot ask the initial query to agent 1. If the initial query with threshold $\tilde{\theta} = 1$ went to agents 3 or 4, note also that they could end up in the protection set regardless of their answers. If agent 3 or 4 answered "yes", they could be a low-priority loser, and if agent 3 or 4 answered "no" they could be a low-priority loser. So, the only remaining option for the initial query is to query agent 2. For agent 2, a positive answer could lead them into the protection set (they could end up the winner) but a negative answer could not lead them into the protection set (they could be a loser, but they could not be a low-priority loser).

The rest of the constructive proof proceeds in a similar manner. In general, the designer wants to keep two active agents at a time. Starting with the initial query to agent 2, whenever the designer gets an affirmative response, they "move left" asking an agent to the left the same threshold. When the designer hits a negative response, they "move right", i.e. another agent *joins* the protocol from the right. For instance, in the example, when agent 2 drops out, the designer "moves right" to agent 3 at the same threshold.

The overdescending-join protocol is constructed in an analogous manner, beginning at the top of the type space, i.e. with an initial threshold query of $\tilde{\theta} = \max \Theta$, and asks a descending sequence of threshold queries. As the principles of the construction are similar, we relegate its discussion to the appendix.

We highlight that different maximally contextually private protocols protect different agents. In particular, the ascending-join protocol is not preferable, from a privacy standpoint alone, to the overdescending-join protocol. The two protocols violate the privacy of disjoint sets of agent-type profiles.

Proposition 8. *The ascending-join protocol and the overdescending-join protocol are incomparable in the contextual privacy order.*

This result is a consequence of the fact that the ascending-join protocol, while protecting *low priority* losers, does violate the privacy of some losers (e.g. agent 3 in the example). The overdescending-join protocol, while protecting *high priority* winners, does violate the privacy of some winners (e.g. agent 2 in the example). The designer must make a normative decision about whose privacy is more important to protect—for example, in the incentive auction design described in Milgrom and Segal (2020), it is more important to protect winners.

Are the ascending-join and overdescending-join protocols the uniquely maximally contextually private protocols for the second-price auction rule? No, they are not. To show the existence of

other maximally contextually private protocols, we need only note that there are protocols that are incomparable in the contextual privacy order to these two. In Appendix C, we discuss a particular protocol which we call the *guessing protocol* that is incomparable to both the ascending-join and overdescending-join protocols.

5 Contextual Privacy and Incentives

In this section, we briefly discuss the strategic aspects of the messaging game induced by an individual elicitation protocols. We formally define two dynamic implementation notions in Section 5.1: implementation in dominant and obviously dominant strategies. Although there is no conceptual innovation in these definitions, it is useful to explicitly write them in our formalism.

In Section 5.2 we consider the incentive properties of the maximally contextually private protocols discussed in Section 4. We first observe that the ascending-join protocol is implementable in obviously dominant strategies. This result is a direct consequence of Li (2017)'s characterization of obvious dominance. Then, we show that the overdescending-join protocol is implementable in dominant strategies. This result strengthens and formalizes an observation in Harstad (2018) that the overdescending protocol is strategyproof.

5.1 Implementation in (Obviously) Dominant Strategies.

Recall that agents submit messages from a set M according to strategies $\sigma_i \colon \Theta \times V \to M$. The state of the protocol is *public*, so that their strategy takes into account information known at node $v \in V$, i.e. $\boldsymbol{\Theta}_v$.

Implementation in dominant strategies requires that at every node v, for every agent i and any opponent message profile $\mathbf{m}_{-i}$, sending a message according to the strategy $\sigma_i(\theta_i, v) \in M$ yields a higher utility than any other message $m_{iv} \in M$. To distinguish actions of agent i at node v from actions of agent i at other nodes, we write $\sigma_{i-v} \colon \Theta \to (V \to M)$ for the partial strategy of agent i at nodes other than v.

Definition. A strategy σ_i in protocol P is *dominant* for agent i if for all θ_i, at all nodes v and possible opponent message profiles $\mathbf{m}_{-i}$,

$$u_i\left(P(\sigma_i(\theta_i, v), \sigma_{i-v}(\theta_i), \mathbf{m}_{-i}); \theta_i\right) \geq u_i\left(P(m_{iv}, \sigma_{i-v}(\theta_i), \mathbf{m}_{-i}); \theta_i\right). \tag{3}$$

for any deviation $m_{iv} \in M$. We say that a protocol P for ϕ *implements* ϕ in *dominant strategies* if there exists a dominant strategy σ_i for each agent i such that $P(\boldsymbol{\sigma}(\boldsymbol{\theta})) = \phi(\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.

Implementation in dominant strategies demands that at any node and under any play by other agents or at other nodes, it must be optimal for an agent to follow their strategy $\sigma_i(\theta_i, v)$. Note

that our notion of dominance assumes that the agent continues to follow her strategies at other nodes σ_{-iv}.

Implementation in obviously dominant strategies requires that the worst outcome following the messaging strategy is better than the best outcome from a deviation, assuming that the deviating agent follows her strategy in future nodes.

Definition. A strategy σ_i in protocol P is *obviously dominant* for agent i if for all θ_i, at all nodes v,

$$\min_{\mathbf{m}_{-i}} u_i\left(P(\sigma_i(\theta_i, v), \sigma_{i-v}(\theta_i), \mathbf{m}_{-i}); \theta_i\right) \geq \max_{\mathbf{m}_{-i}} u_i\left(P(m_{iv}, \sigma_{i-v}(\theta_i), \mathbf{m}_{-i}); \theta_i\right),$$

for all potential deviations $m_{iv} \in M$. We say that a protocol P for ϕ *implements* ϕ in *obviously dominant strategies* if there exists an obviously dominant strategy σ_i for each agent i such that $P(\boldsymbol{\sigma}(\boldsymbol{\theta})) = \phi(\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.

In an obviously dominant strategy profile, it has to be the case that for all agents i, the worst outcome that can result from following the strategy σ_i, is better than the best outcome that can result from some other strategy σ_i', regardless of what other agents do in the future.

5.2 Dynamic Incentives of Maximally Contextually Private Protocols.

We now show that the two maximally contextually private designs highlighted in Section 4 also have good incentive properties. We begin with the ascending-join protocol.

Proposition 9. *The ascending-join protocol implements the second-price auction choice rule ϕ^{SPA} in obviously dominant strategies.*

To show that the second part of this statement holds, we observe that the ascending-join protocol we have defined is equivalent to a *personal-clock auction*. Therefore, we can rely on the characterization in Li (2017, Theorem 3), which states that every personal-clock auction protocol has an equilibrium in obviously dominant strategies.

We also show that the second protocol shown to be maximally contextually private in Theorem 3 has a dynamic implementation, albeit with a weaker incentive guarantee.

Proposition 10. *The overdescending-join protocol implements the choice rule ϕ^{SPA} in dominant strategies.*

To see that this statement holds, observe that there are two phases of the overdescending-join protocol: a phase before the winner is found, and a phase after the winner is found. Before the winner is found, for any fixed opponent message profile $\mathbf{m}_{-i}$, the price of the good is not influenced by the action of the agent i. The agent i only influences whether they win the good or not—they have an incentive to send a message that they are above a threshold as long as they prefer winning

over losing, which is the case if and only if the current price is at or below their type. In the second phase, once the winner has been found, all agents are indifferent between all messages. So, this set of strategies is clearly weakly dominant.

In addition to the ascending-join and overdescending-join protocols, the serial dictatorship also has good privacy properties and incentive guarantees. In particular, we show in Appendix B that the serial dictatorship protocol is contextually private (i.e. it produces no privacy violations) and that the messaging strategies are obviously dominant.

We view further study of incentives and contextual privacy as a direction for future work. The intention of this section was primarily to show that our formalism is capacious enough to discuss incentives, and to begin to show how our results connect to the literature on dynamic implementation.

6 Related Literature

This paper brings privacy considerations into extensive-form mechanism design. We discuss here our connection to extensive-form mechanism design, as well as other literature on designing for privacy in computer science and cryptography.

Our restriction to individual elicitation protocols coincides with the extensive-form messaging game used to define and study obvious strategyproofness (Li, 2017) and credibility (Akbarpour and Li, 2020). Credibility shares a motivation with contextual privacy—both criteria have to do with the potential for the designer to somehow abuse their communication channel with the agent. Credibility, which requires incentive compatibility for the auctioneer, sometimes coincides with the diagnoses of contextual privacy and sometimes not.[17] Under individual elicitation protocols, contextual privacy can be understood as a form of privacy that is easier for participants to understand, just as obvious strategyproofness is a form of strategyproofness that is easier for participants to understand. Many papers study variants of obvious strategyproofness and their compatibility with other axioms and computational properties (Bade and Gonczarowski, 2016; Ashlagi and Gonczarowski, 2018; Mackenzie, 2020; Golowich and Li, 2021; Mackenzie, 2020; Chen and Westkamp, 2022; Pycia and Troyan, 2023).

Other considerations related to privacy and trust have been incorporated into mechanism design and market design, in both static and dynamic models. Though it is not the focus of their paper, Mackenzie and Zhou (2022) discuss how the dynamic *menu mechanisms* they define (in which at each history the agent chooses from a menu of possible outcomes) protect privacy compared to direct mechanisms. Grigoryan and Möller (2023) and Woodward (2020) define two different but related notions of the *auditability* of different mechanisms, based on the amount of information that

[17] The descending or *Dutch* protocol of the first-price auction is both contextually private and credible, but the ascending protocol of the second-price auction is credible but not contextually private.

would be required to determine whether the outcome of a mechanism had been correctly computed, and Hakimov and Raghavan (2022) shows how providing feedback to participants can help to verify the mechanism. Others study how the disclosure of past trades affects future trades (Dworczak, 2020; Ollar et al., 2021) and how the desire for privacy may stem from a desire to avoid price discrimination (Ichihashi, 2020; Ali et al., 2023). Canetti et al. (2023) considers the privacy of the designer as opposed to our focus on the privacy of agents, and investigates the use of zero-knowledge proofs to prove properties of the mechanism without revealing the designer's objectives. Several papers have incorporated measures of "privacy loss" as constraints on mechanism design (Eilat et al., 2021; Liu and Bagh, 2020), where privacy loss is defined as some measure (e.g. Shannon entropy, Kullback-Leibler divergence) of information revelation. These measure-based criteria treat all datum as equal. Contextual privacy, unlike these measure-based criteria, is not about *how much* information is revealed, and is also not just about *whether* information is revealed, but rather it is about *how* the information that is revealed is *used*.

There are two important precursors to contextual privacy in the theory of decentralized computation: unconditional full privacy and perfect implementation. Contextual privacy under the individual elicitation technology parallels the concept of *unconditional full privacy* for decentralized protocols (Chor and Kushilevitz, 1989; Brandt and Sandholm, 2005, 2008). Unconditional full privacy requires that the only information revealed through a decentralized protocol is the information contained in the outcome—this notion of privacy is *unconditional* in that it does not condition on the presence of a mediator or on computational hardness assumptions. It has been applied to an auction domain (Brandt and Sandholm, 2008), and a voting domain (Brandt and Sandholm, 2005), stressing impossibility results. Our definition of contextual privacy brings unconditional full privacy into a framework amenable to economic design and extends it in several ways, highlighting that design for privacy involves tradeoffs regarding whose privacy to protect. Milgrom and Segal (2020)'s concept of *unconditional winner privacy* is similar to contextual privacy in that it brings unconditional full privacy into centralized mechanism design: unconditional winner privacy is unconditional full privacy in a centralized mechanism, for the winner only. We generalize the Milgrom and Segal (2020)'s concept to objectives beyond protection of the winner's privacy. The notion of *perfect implementation* (Izmalkov et al., 2005, 2011) seeks implementations that do not rely on trusted mediators, but rather rely on simple technologies that enable verification of what was learned—like sealed envelopes. The construction allows for particular elicitation technologies that allow, e.g., envelopes-inside-envelopes. Although our general set up can accommodate such technologies, we focus on more minimal assumptions about the technology available to the designer (e.g. the individual elicitation technology), and offer a privacy order rather than an implementation concept.

Beyond unconditional full privacy and perfect implementation lies an extensive literature on privacy preserving protocols for auctions and allocation. 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 while usually relying on computational hardness assumptions. Compared to this literature, contextual privacy makes explicit the social and technological environments in which many designers operate: when arbitrary cryptographic protocols are not available, we need some other privacy desideratum to guide design.

An influential privacy desideratum is differential privacy (Dwork et al., 2006). Contextual privacy sharply diverges from interpretations of differential privacy in mechanism design contexts. Differential privacy was originally defined for database management. For a survey of its incorporation into mechanism design, including works that incorporate privacy concerns in agent utility functions (Nissim et al., 2012; Roth and Schoenebeck, 2012; Ligett and Roth, 2012; Ghosh and Roth, 2015), see Pai and Roth (2013). Differential privacy, as adapted for mechanism design contexts, says that the information revealed about a single agent should have a *negligible* effect on the outcome. Contextual privacy, however, says that subsets of the agent's information should be revealed if it has *some* effect on the outcome. Where contextual privacy justifies information revelation through its relevance to a social outcome, differential privacy explicitly restricts the sensitivity of an outcome to revealed information.

More broadly, our paper connects to a growing theoretical and empirical literature on digital privacy in economics. While we cannot do justice to this literature here, Goldfarb and Que (2023) and Acquisti et al. (2016) offer valuable surveys of this literature. One preoccupation of this literature is the appearance of a digital "privacy paradox", wherein consumers' stated preferences about privacy (that it matters) do not align with their revealed preferences (which suggest that it doesn't matter). Some explanations of the privacy paradox point out that data externalities may complicate individual incentives to pursue privacy-preserving actions (Ichihashi, 2021; Acemoglu et al., 2022; Bergemann et al., 2022). The data externalities argument helps to motivate why a social planner would want to design for privacy (using a criterion like contextual privacy) even when operating in a setting where agents don't appear to have privacy concerns. Recent empirical work quantifies preferences for privacy and emphasizes that they are context-dependent and changing over time, suggesting that what appears to be a privacy paradox may simply be a reflection of the highly context-sensitive nature of preferences for privacy (Goldfarb and Tucker, 2012; Lin, 2022; Tang, 2019). This evidence suggests that it is important to have a normative notion of privacy that is attentive to not just *how much* information is revealed but also *how the information revealed is used*—this is what contextual privacy captures.

7 Conclusion

An agent's contextual privacy is violated if the designer learns superfluous information about her. It's not always possible for the designer to protect every agent's contextual privacy while still eliciting enough information to compute the outcome of a choice rule. So, the designer needs to make a deliberate decision about whose privacy is most important to protect. In auctions, for example, this means prioritizing either winner or loser privacy. It might be, as is the case for the English protocol for the second-price auction rule, that a commonly-used protocol's privacy can be strictly improved. These improvements come from delaying the involvement of some agents.

The contextual privacy order we defined leads to robust comparisons independent of the designer's prior knowledge. An alternative approach, not captured by our analysis, is to specify a cardinal objective for the designer that incorporates a loss from contextual privacy violations. Such a designer can compute an optimal protocol taking into account their privacy objectives. Consider, for example, a case in which all privacy violations incur the same loss in social welfare. In such a case, the designer could use the prior on type profiles to find an optimal protocol from a privacy perspective, weighing the losses from (many) losers against (few) winners.

A remaining concern, for the privacy-conscious designer, is about how contextual privacy interacts with dynamic incentives. While the serial dictatorship and the ascending-join and overdescending-join protocols have good dynamic incentives, it is not clear which properties of these privacy-preserving protocols lead to such incentive guarantees. Absent more structure on the choice rule—such as a strengthening of interval pivotality—incentive properties of privacy-preserving protocols will depend on many details of the environment.

References

Acemoglu, Daron, Ali Makhdoumi, Azarakhsh Malekian, and Asu Ozdaglar, "Too much data: Prices and inefficiencies in data markets," *American Economic Journal: Microeconomics*, 2022, *14* (4), 218–256. 31

Acquisti, Alessandro, Curtis Taylor, and Liad Wagman, "The economics of privacy," *Journal of economic Literature*, 2016, *54* (2), 442–492. 31

Akbarpour, Mohammad and Shengwu Li, "Credible auctions: A trilemma," *Econometrica*, 2020, *88* (2), 425–467. 8, 29

Alawadhi, Neha, "Only 1 in 250 understand role of encryption in securing messaging: Study," *Business Standard*, 2021, *March 21*. 3

Ali, S Nageeb, Greg Lewis, and Shoshana Vasserman, "Voluntary disclosure and personalized pricing," *Review of Economic Studies*, 2023, pp. 537–538. 30

Alvarez, Ramiro and Mehrdad Nojoumian, "Comprehensive survey on privacy-preserving protocols for sealed-bid auctions," *Computers & Security*, 2020, *88*, 101502. 3

Ashlagi, Itai and Yannai A Gonczarowski, "Stable matching mechanisms are not obviously strategy-proof," *Journal of Economic Theory*, 2018, *177*, 405–425. 29

Ausubel, Lawrence M, "An efficient ascending-bid auction for multiple objects," *American Economic Review*, 2004, *94* (5), 1452–1475. 2

Bade, Sophie, "Random serial dictatorship: The one and only," *Mathematics of Operations Research*, 2020, *45* (1), 353–368. 66

_ **and Yannai A Gonczarowski**, "Gibbard-Satterthwaite success stories and obvious strategyproofness," *arXiv preprint arXiv:1610.04873*, 2016. 29

Balinski, Michel and Tayfun Sönmez, "A tale of two mechanisms: student placement," *Journal of Economic theory*, 1999, *84* (1), 73–94. 15

Bergemann, Dirk, Alessandro Bonatti, and Tan Gan, "The economics of social data," *The RAND Journal of Economics*, 2022, *53* (2), 263–296. 31

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. 3

Brandt, Felix and Tuomas Sandholm, "Unconditional privacy in social choice," in "Theoretical Aspects of Rationality and Knowledge" 2005, pp. 207–218. 4, 30, 59

_ **and** _ , "On the existence of unconditionally privacy-preserving auction protocols," *ACM Transactions on Information and System Security (TISSEC)*, 2008, *11* (2), 1–21. 30

Canetti, Ran, Amos Fiat, and Yannai A Gonczarowski, "Zero-Knowledge Mechanisms," *arXiv preprint arXiv:2302.05590*, 2023. 30

Chen, Yiqiu and Alexander Westkamp, "Optimal Sequential Implementation," 2022. working paper. 29

Chor, Benny and E. Kushilevitz, "A Zero-One Law for Boolean Privacy," in "Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing" STOC '89 Association for Computing Machinery New York, NY, USA 1989, p. 62–72. 4, 10, 30

_ **, Mihály Geréb-Graus, and Eyal Kushilevitz**, "On the structure of the privacy hierarchy," *Journal of Cryptology*, 1994, *7* (1), 53–60. 4, 10

Dworczak, Piotr, "Mechanism design with aftermarkets: Cutoff mechanisms," *Econometrica*, 2020, *88* (6), 2629–2661. 30

Dwork, Cynthia, Frank McSherry, Kobbi Nissim, and Adam Smith, "Calibrating noise to sensitivity in private data analysis," in "Theory of cryptography conference" Springer 2006, pp. 265–284. 31

Eilat, Ran, Kfir Eliaz, and Xiaosheng Mu, "Bayesian privacy," *Theoretical Economics*, 2021, *16* (4), 1557–1603. 30

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. 31

Ghosh, Arpita and Aaron Roth, "Selling privacy at auction," *Games and Economic Behavior*, 2015, *91*, 334–346. 31

Goldfarb, Avi and Catherine Tucker, "Shifts in privacy concerns," *American Economic Review*, 2012, *102* (3), 349–353. 31

_ **and Verina F Que**, "The economics of digital privacy," *Annual Review of Economics*, 2023, *15*, 267–286. 31

Golowich, Louis and Shengwu Li, "On the computational properties of obviously strategy-proof mechanisms," *arXiv preprint arXiv:2101.05149*, 2021. 29

Grigoryan, Aram and Markus Möller, "A theory of auditability for allocation and social choice mechanisms," *arXiv preprint arXiv:2305.09314*, 2023. 29

Hakimov, Rustamdjan and Madhav Raghavan, "Improving transparency in school admissions: Theory and experiment," *Available at SSRN 3572020*, 2022. 30

Harstad, Ronald, "Systems and methods for advanced auction management," *US Patent No. 10,726,476 B2*, 2018. 4, 23, 27

Ichihashi, Shota, "Online privacy and information disclosure by consumers," *American Economic Review*, 2020, *110* (2), 569–595. 30

_ , "The economics of data externalities," *Journal of Economic Theory*, 2021, *196*, 105316. 31

Izmalkov, Sergei, Matt Lepinski, and Silvio Micali, "Perfect implementation," *Games and Economic Behavior*, 2011, *71* (1), 121–140. 30

_ **, Silvio Micali, and Matt Lepinski**, "Rational secure computation and ideal mechanism design," in "46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)" IEEE 2005, pp. 585–594. 30

Li, Shengwu, "Obviously strategy-proof mechanisms," *American Economic Review*, 2017, *107* (11), 3257–87. 4, 8, 27, 28, 29, 56, 67

Ligett, Katrina and Aaron Roth, "Take it or leave it: Running a survey when privacy comes at a cost," in "International workshop on internet and network economics" Springer 2012, pp. 378–391. 31

Lin, Tesary, "Valuing intrinsic and instrumental preferences for privacy," *Marketing Science*, 2022, *41* (4), 663–681. 31

Liu, De and Adib Bagh, "Preserving bidder privacy in assignment auctions: design and measurement," *Management Science*, 2020, *66* (7), 3162–3182. 30

Mackenzie, Andrew, "A revelation principle for obviously strategy-proof implementation," *Games and Economic Behavior*, 2020, *124*, 512–533. 29

_ **and Yu Zhou**, "Menu mechanisms," *Journal of Economic Theory*, 2022, *204*, 105511. 8, 29

McMillan, John, "Selling spectrum rights," *Journal of Economic Perspectives*, 1994, *8* (3), 145–162. 2

Milgrom, Paul and Ilya Segal, "Clock auctions and radio spectrum reallocation," *Journal of Political Economy*, 2020, *128* (1), 1–31. 4, 19, 22, 26, 30

Moulin, Hervé, "On strategy-proofness and single peakedness," *Public Choice*, 1980, *35* (4), 437–455. 18

Nissenbaum, Helen, "Privacy as contextual integrity," *Washington Law Review*, 2004, *79*, 119. 8

Nissim, Kobbi, Claudio Orlandi, and Rann Smorodinsky, "Privacy-aware mechanism design," in "Proceedings of the 13th ACM Conference on Electronic Commerce" 2012, pp. 774–789. 31

Nurmi, Hannu and Arto Salomaa, "Cryptographic protocols for Vickrey auctions," *Group Decision and Negotiation*, 1993, *2* (4), 363–373. 31

Ollar, Mariann, Marzena J Rostek, and Ji Hee Yoon, "Privacy in markets," *Available at SSRN 3071374*, 2021. 30

Pai, Mallesh M and Aaron Roth, "Privacy and mechanism design," *ACM SIGecom Exchanges*, 2013, *12* (1), 8–29. 31

Pycia, Marek and M Utku Ünver, "Incentive compatible allocation and exchange of discrete resources," *Theoretical Economics*, 2017, *12* (1), 287–329. 66

_ **and Madhav Raghavan**, "Non-bossiness and first-price auctions," Technical Report, CEPR Discussion Paper No. DP16874 2022. 8, 66

_ **and Peter Troyan**, "A theory of simplicity in games and mechanism design," *Econometrica*, 2023, *91* (4), 1495–1526. 8, 29, 67

Roth, Aaron and Grant Schoenebeck, "Conducting truthful surveys, cheaply," in "Proceedings of the 13th ACM Conference on Electronic Commerce" 2012, pp. 826–843. 31

Rothkopf, Michael H, Thomas J Teisberg, and Edward P Kahn, "Why are Vickrey auctions rare?," *Journal of Political Economy*, 1990, *98* (1), 94–109. 2

Satterthwaite, Mark A and Hugo Sonnenschein, "Strategy-proof allocation mechanisms at differentiable points," *The Review of Economic Studies*, 1981, *48* (4), 587–597. 8, 65

Segal, Ilya, "The communication requirements of social choice rules and supporting budget sets," *Journal of Economic Theory*, 2007, *136* (1), 341–378. 8, 9

Shapley, Lloyd and Herbert Scarf, "On cores and indivisibility," *Journal of Mathematical Economics*, 1974, *1* (1), 23–37. 13

Svensson, Lars-Gunnar, "Strategy-proof allocation of indivisible goods," *Social Choice and Welfare*, 1999, *16*, 557–567. 66

Tang, Huan, "The value of privacy: Evidence from online borrowers," *Available at SSRN*, 2019. 31

Woodward, Kyle, "Self-auditable auctions," Technical Report, Working paper 2020. 29

Appendix Table of Contents

A Proofs **38**

A.1 Proof of Proposition 1. . . . 38
A.2 Proof of Proposition 2. . . . 39
A.3 Proof of Theorem 1 40
A.4 Proof of Proposition 4. . . . 41
A.5 Strengthening of Proposition 5 Under No Ties. . . . 41
A.6 Proof of Proposition 6. . . . 43
A.7 Proof of Proposition 7. . . . 44
A.8 Proof of Theorem 2. . . . 44
A.9 Proof of Theorem 3. . . . 49
A.10 Proof of Proposition 8. . . . 56
A.11 Proof of Proposition 9. . . . 56
A.12 Proof of Proposition 10. . . . 57

B Two Contextually Private Protocols **57**

B.1 Serial Dictatorships. . . . 57
B.2 First-Price Auction. . . . 59

C Other Maximally Contextually Private Protocols **60**

D Count Elicitation Technology **62**

E Variations of Contextual Privacy **64**

E.1 Individual Contextual Privacy . . . 64
E.2 Group Contextual Privacy . . . 66

A Proofs

A.1 Proof of Proposition 1.

Proposition 1. *An $\mathfrak{S}$-computable choice rule ϕ is $\mathfrak{S}$-contextually private if and only if there does not exist a subset of type profiles $\hat{\boldsymbol{\Theta}} \subseteq \boldsymbol{\Theta}$ such that*

(i) $\phi|_{\hat{\boldsymbol{\Theta}}}$ is non-constant, and

(ii) for every partition $\mathcal{S} \in \mathfrak{S}^*$ such that none of the partition cells $S \in \mathcal{S}$ contains $\hat{\boldsymbol{\Theta}}$, there are type profiles $(\theta_i, \boldsymbol{\theta}_{-i}), (\theta_i', \boldsymbol{\theta}_{-i}) \in \hat{\boldsymbol{\Theta}}$ that lie in different partition cells of $\mathcal{S}$.

Here, $\mathfrak{S}^* \subseteq 2^{2^{\boldsymbol{\Theta}}}$ *is the set of* revealed partitions. *Define* $\mathcal{S}^* \in \mathfrak{S}^*$ *if there is a partition* $\mathcal{S} \in \mathfrak{S}$ *and functions* $f_1, f_2, \ldots, f_n \colon \Theta \to M$ *such that every* $S^* \in \mathcal{S}^*$ *is the preimage under* $(f_1, f_2, \ldots, f_n) \colon \boldsymbol{\Theta} \to \mathbf{M}$ *of some* $S \in \mathcal{S}$, $S^* = (f_1, f_2, \ldots, f_n)^{-1}(S)$.

In addition, if there is such a $\hat{\boldsymbol{\Theta}}$ *then any protocol for* ϕ *must produce a contextual privacy violation for some type profile in* $\hat{\boldsymbol{\Theta}}$*, that is, it must be that for any* $P \in \mathcal{P}_{\mathfrak{S}}$, $\Gamma(P, \boldsymbol{\sigma}, \phi) \cap (\hat{\boldsymbol{\Theta}} \times N) \neq \emptyset$.

Proof. We first show that contextual privacy implies that no such set $\hat{\boldsymbol{\Theta}}$ exists. We prove this statement in the contrapositive. Hence, assume that $\hat{\boldsymbol{\Theta}}$ exists. As $\phi|_{\hat{\boldsymbol{\Theta}}}$ is non-constant (by hypothesis (i)) and P is a protocol for ϕ, the set of nodes that distinguish two type profiles in $\hat{\boldsymbol{\Theta}}$ is non-empty. Let v be any earliest (i.e. minimal in precedence order) node that distinguishes two type profiles in $\hat{\boldsymbol{\Theta}}$. The query at node v, s_v, must have $|s_v(\hat{\boldsymbol{\Theta}})| \geq 2$ as it distinguishes types in $\hat{\boldsymbol{\Theta}}$. Hence, by the hypothesis (ii) in the statement, there exist $\boldsymbol{\theta} = (\theta_i, \boldsymbol{\theta}_{-i})$ and $\boldsymbol{\theta}' = (\theta_i', \boldsymbol{\theta}_{-i})$ with $s_v(\boldsymbol{\sigma}(\boldsymbol{\theta}, v)) \neq s_v(\boldsymbol{\sigma}(\boldsymbol{\theta}', v))$ and $\phi(\boldsymbol{\theta}) = \phi(\boldsymbol{\theta}')$. These constitute a contextual privacy violation.

For the converse direction, assume that no such $\hat{\boldsymbol{\Theta}}$ exists. We construct a contextually private protocol inductively. For the base case, we consider $\boldsymbol{\Theta}_r = \boldsymbol{\Theta}$. For the inductive step, we consider an arbitrary node v associated with $\boldsymbol{\Theta}_v = \hat{\boldsymbol{\Theta}}$. Either, $\phi|_{\hat{\boldsymbol{\Theta}}}$ is constant, and the protocol can terminate and compute ϕ, or not. If $\phi|_{\hat{\boldsymbol{\Theta}}}$ is not constant then there must be a query s_v' and a strategy profile $\boldsymbol{\sigma}$ such that $|s_v'(\boldsymbol{\sigma}(\hat{\boldsymbol{\Theta}}, v))| \geq 2$ because, by assumption, there is an $\mathfrak{S}^*$-protocol for ϕ. Since hypothesis (i) holds, hypothesis (ii) cannot: there must be a query s_v such that $|s_v(\boldsymbol{\sigma}(\hat{\boldsymbol{\Theta}}, v))| \geq 2$ and there are no types $(\theta_i, \boldsymbol{\theta}_{-i}), (\theta_i', \boldsymbol{\theta}_{-i}) \in \hat{\boldsymbol{\Theta}}$ with $s_v(\boldsymbol{\sigma}((\theta_i, \boldsymbol{\theta}_{-i}), v)) \neq s_v(\boldsymbol{\sigma}((\theta_i', \boldsymbol{\theta}_{-i}), v))$ and $\phi(\theta_i, \boldsymbol{\theta}_{-i}) = \phi(\theta_i', \boldsymbol{\theta}_{-i})$. Hence, the query s_v does not introduce any contextual privacy violations. The induction terminates after finitely many rounds (because $|\boldsymbol{\Theta}| < \infty$) and the cardinality of $\boldsymbol{\Theta}_v$ strictly decreases along paths on the tree P. □

A.2 Proof of Proposition 2.

Proposition 2. *A choice function* ϕ *fails to be* $\mathfrak{S}_{\text{IE}}$*-contextually private if and only if there exists some cylinder set* $\hat{\boldsymbol{\Theta}} = \bigtimes_{i=1}^{n} \hat{\Theta}_i$ *such that (i)* $\phi|_{\hat{\boldsymbol{\Theta}}}$ *is non-constant and (ii) for all agents* i *and all* $\theta_i, \theta_i' \in \hat{\boldsymbol{\Theta}}_i$, θ_i *and* θ_i' *are inseparable with respect to* $\hat{\boldsymbol{\Theta}}$ *and* ϕ.

In addition, it must be that for any protocol $(P, \boldsymbol{\sigma}) \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$*, one of the agents* i *that has at least two inseparable types must incur a privacy violation, that is,* $\Gamma(P, \boldsymbol{\sigma}, \phi) \cap \{(i, \boldsymbol{\theta}) \in N \times \boldsymbol{\Theta} \mid \exists \theta_i' \neq \theta_i : (\theta_i, \boldsymbol{\theta}_{-i}), (\theta_i', \boldsymbol{\theta}_{-i}) \in \hat{\boldsymbol{\Theta}}\} \neq \emptyset$.

Proof. We first consider necessity. Assume for contradiction that there is a contextually private protocol P for the choice function ϕ and that there is a product set $\hat{\boldsymbol{\Theta}}$ such that all types are inseparable under $\hat{\boldsymbol{\Theta}}$ and ϕ is non-constant on this set.

As ϕ is non-constant on $\hat{\boldsymbol{\Theta}}$, the protocol must make a query separating type profiles $(\theta_i, \boldsymbol{\theta}_{-i})$ and $(\theta_i', \boldsymbol{\theta}_{-i})$ for some agent i. Consider the earliest such query in the precedence order on P.

By the choice of v and $\theta_i \sim_{i,\phi,\hat{\mathbf{\Theta}}} \theta_i'$, there must be a chain $\theta_i^1, \theta_i^2, \ldots, \theta_i^k$ such that $\theta_i^1 = \theta_i$ and $\theta_i^k = \theta_i'$ and

$$\theta_i^1 \sim'_{i,\phi,\hat{\mathbf{\Theta}}} \theta_i^2 \sim'_{i,\phi,\hat{\mathbf{\Theta}}} \cdots \sim'_{i,\phi,\hat{\mathbf{\Theta}}} \theta_i^k.$$

That is, there is a chain of direct inseparability from θ_i to θ_i'. As θ_i and θ_i' are separated at v, there must be $l = 1, 2, \ldots, k-1$ such that v distinguishes θ_i^l and θ_i^{l+1}. By direct inseparability, there is $\boldsymbol{\theta}_{-i}$ such that $(\theta_i^l, \boldsymbol{\theta}_{-i}), (\theta_i^{l+1}, \boldsymbol{\theta}_{-i}) \in \hat{\mathbf{\Theta}}$ and $\phi(\theta_i^l, \boldsymbol{\theta}_{-i}) = \phi(\theta_i^{l+1}, \boldsymbol{\theta}_{-i})$. Together, these two observations yield a contradiction to contextual privacy of P.

Now consider sufficiency. We define a contextually private protocol inductively on nodes v. Throughout the induction, the following holds:

$$\text{For any terminal nodes } w, w' \text{ whose latest common ancestor in } P \text{ is } v, \text{ there are no } (\theta_i, \boldsymbol{\theta}_{-i}) \in \mathbf{\Theta}_w \text{ and } (\theta_i', \boldsymbol{\theta}_{-i}) \in \mathbf{\Theta}_{w'} \text{ such that } \phi(\theta_i, \boldsymbol{\theta}_{-i}) = \phi(\theta_i', \boldsymbol{\theta}_{-i}). \tag{4}$$

Note that a protocol that satisfies (4) at all internal nodes is contextually private. We start the induction in the root node r of P.

Assume a protocol has been constructed until query v associated to type set $\mathbf{\Theta}_v \subseteq \mathbf{\Theta}$. If ϕ is constant on the remaining set, the node is terminal, and the outcome of the social choice function can be determined.

Otherwise, because there is no restriction to a product set $\hat{\mathbf{\Theta}}$ under which all types are inseparable, there are types θ_i, θ_i' that are separable under $\mathbf{\Theta}_v$ for agent i. Consider the binary query that distinguishes the equivalence class of θ_i, $[\theta_i]_{i,\phi,\mathbf{\Theta}_v}$, from its complement $\Theta_{v,i} \setminus [\theta_i]_{i,\phi,\mathbf{\Theta}_v}$, which is non-empty as it contains at least θ_i'. By definition of $\sim$, for any $\boldsymbol{\theta}_{-i}$, and any $\tilde{\theta}_i, \tilde{\theta}_i'$ such that $(\tilde{\theta}_i, \boldsymbol{\theta}_{-i}), (\tilde{\theta}_i', \boldsymbol{\theta}_{-i}) \in \hat{\mathbf{\Theta}}$,

$$\phi(\tilde{\theta}_i, \boldsymbol{\theta}_{-i}) \neq \phi(\tilde{\theta}_i', \boldsymbol{\theta}_{-i}),$$

completing the induction step and the proof (as (4) continues to hold). □

A.3 Proof of Theorem 1

Theorem 1. *Let ϕ be a social choice function, and consider a type profile $\boldsymbol{\theta} \in \mathbf{\Theta}$. If for any subset $A \subseteq N$ of agents, and types $\theta_i' \in \Theta$ for agents $i \in A$,*

$$\phi(\boldsymbol{\theta}_A, \boldsymbol{\theta}_{-A}) \neq \phi(\boldsymbol{\theta}'_A, \boldsymbol{\theta}_{-A}) \qquad \text{(collective pivotality)}$$

and for all $i \in A$

$$\phi(\theta_i, \boldsymbol{\theta}_{-i}) = \phi(\theta_i', \boldsymbol{\theta}_{-i}) \qquad \text{(no individual pivotality)}$$

then for any individual elicitation protocol $P \in \mathcal{P}_{\mathfrak{S}_{\mathrm{IE}}}$, there exists an agent $i \in A$ whose contextual privacy is violated at $\boldsymbol{\theta}$, i.e. $A \times \{\boldsymbol{\theta}\} \cap \Gamma(P, \phi) \neq \emptyset$.

Proof. Assume that $\phi(\boldsymbol{\theta}_A, \boldsymbol{\theta}_{-A}) \neq \phi(\boldsymbol{\theta}'_A, \boldsymbol{\theta}_{-A})$ and $\phi(\theta_i, \boldsymbol{\theta}_{-i}) = \phi(\theta'_i, \boldsymbol{\theta}_{-i})$ for all $i \in A$. Note that the cylinder set

$$\hat{\boldsymbol{\Theta}} := \bigtimes_{i \in A} \{\theta_i, \theta'_i\} \times \{\boldsymbol{\theta}_{-A}\}$$

satisfies the conditions of Proposition 2. Hence, we know that there is a contextual privacy violation for some type profile $\boldsymbol{\theta}' \in \hat{\boldsymbol{\Theta}}$.

To show that there is a violation at $\boldsymbol{\theta}$ in particular, consider a first query separating types from $\hat{\boldsymbol{\Theta}}$. This query must distinguish θ_i and θ'_i for some agent $i \in A$. By no individual pivotality, this violates contextual privacy for this agent at $\boldsymbol{\theta}$. □

A.4 Proof of Proposition 4.

Proposition 4. *Assume $|C| \geq 2$ and $|N| \geq 2$ and that there is no oversupply. Then any protocol $P \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$ for a stable choice rule produces contextual privacy violations.*

Proof. The proof constructs collective but not individual pivotality.

Let $s_1 > s_2 > s_3 > s_4$. Fix the type profile of all $n-2$ agents that are not i or j to be $\boldsymbol{\theta}_{-ij}$ where each agent has a score greater than s_1 for their top choice object, and their top choice object has capacity to accommodate them. Assume further that the remaining spots are for different objects. Label these objects with remaining spots a and b.

Consider the final two agents $i, j \in N$. Choose their type profiles to be:

$$\theta_i = (a \succ_i b, s_i(a) = s_1, s_j(b) = s_4), \qquad \theta'_i = (b \succ_i a, s_i(a) = s_3, s_j(b) = s_2)$$
$$\theta_j = (b \succ_j a, s_j(a) = s_4, s_j(b) = s_1), \qquad \theta'_j = (a \succ_j b, s_j(a) = s_2, s_j(b) = s_3).$$

The preferences of agents i and j for objects a and b are represented in Figure 6. Agent i and j's scores and preferences for other schools are arbitrary.

Let x be the outcome in which agent i is matched to school a and j is matched to b. Let y be the outcome in which agent i is matched to b and j is matched to a. In both x and y, all agents not i or j are assigned to their top choice object at which they have a high score.

Stability requires that $\phi(\theta_i, \theta_j, \boldsymbol{\theta}_{-ij}) = (\theta_i, \theta'_j, \boldsymbol{\theta}_{-ij}) = (\theta'_i, \theta_j, \boldsymbol{\theta}_{-ij}) = x$ while $\phi(\theta_i, \theta_j, \boldsymbol{\theta}_{-ij}) = y$. We have chosen a particular $\boldsymbol{\theta}_{-ij} \in \boldsymbol{\Theta}_{-ij}$, and particular $\theta_i, \theta'_i, \theta_j, \theta'_j \in \Theta$ such that the condition of Theorem 1 holds. □

A.5 Strengthening of Proposition 5 Under No Ties.

Proposition 2 allows us to strengthen Proposition 5 for cases without ties in the type profile (i.e. where no two agents have the same type).

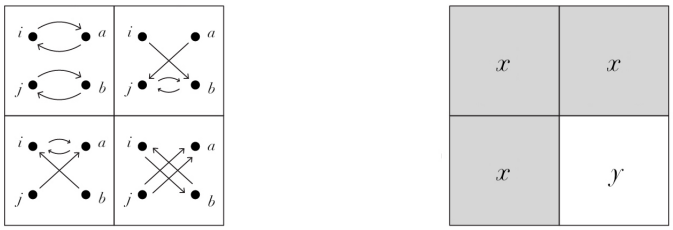


Figure 6: Constructing collective but not individual pivotality for college assignment. Agent types $\theta_i, \theta_i', \theta_j, \theta_j'$ (left, arrows from agents denote favored object, arrows from objects denote high score); outcomes under any stable choice rule (right, where $x = ((i,a),(j,b))$ and $y = ((j,a),(i,b))$).

Proposition 11. *Assume $|\Theta| \geq 9$. If there are $n \geq 3$ agents, any protocol $P \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$ for the second-price auction rule produces contextual privacy violations.*

Proof. Consider the three highest priority (in lexicographic order) agents 1, 2 and 3, and 9 types

$$\theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7, \theta_8,$$

that are increasing in the index.

Consider the product set $\hat{\boldsymbol{\Theta}} = \{\theta_5, \theta_0, \theta_2\} \times \{\theta_8, \theta_7, \theta_3\} \times \{\theta_6, \theta_4, \theta_1\} \times \bigtimes_{i=4}^{n} \{\theta_0\}$. In this product set, the first factor represents types of agent 1, the second represents possible types of agent 2, the third represents possible types of agent 3, and all other agents have type θ_0. We will show that when ϕ^{SPA} is evaluated on this restricted product set, it is non-constant and all types in the product set are inseparable.

To see this, we construct the tensor of outcomes for the product set. This tensor is represented in Figure 7. We represent agent 1's type on the up-down axis, agent 2's type on the left-right axis, and agent 3's type is constant for each box. The outcomes under ϕ^{SPA} are represented by letters and colors. For example, the upper left corner in the left-most box signifies $\phi^{\text{SPA}}(\theta_2, \theta_8, \theta_6) = a$, where a is the outcome under which agent 2 wins the object and pays a price θ_6.

To see that this constitutes a violation of contextual privacy, we show that: (i) ϕ is non-constant on $\hat{\boldsymbol{\Theta}}$, and (ii) for all agents i, and all $\theta_i, \theta_i' \in \hat{\boldsymbol{\Theta}}$, θ_i and θ_i' are inseparable. As for (i), we can observe immediately that $\phi|_{\hat{\boldsymbol{\Theta}}}$ is non-constant. To see (ii) that all types are inseparable, we go through each agent in turn.

- Agent 1: Outcome a is the same for $\boldsymbol{\theta}_{-1} = (\theta_7, \theta_6)$, hence all agent 1 types are inseparable.
- Agent 2: Outcome i for $\boldsymbol{\theta}_{-2} = (\theta_5, \theta_1)$ show that all agent 2 types are inseparable.
- Agent 3: θ_6 and θ_4 are inseparable because they both yield outcome b for $\boldsymbol{\theta}_{-3} = (\theta_0, \theta_3)$. θ_1 and θ_4 are inseparable because they both yield outcome d for $\boldsymbol{\theta}_{-3} = (\theta_2, \theta_7)$.

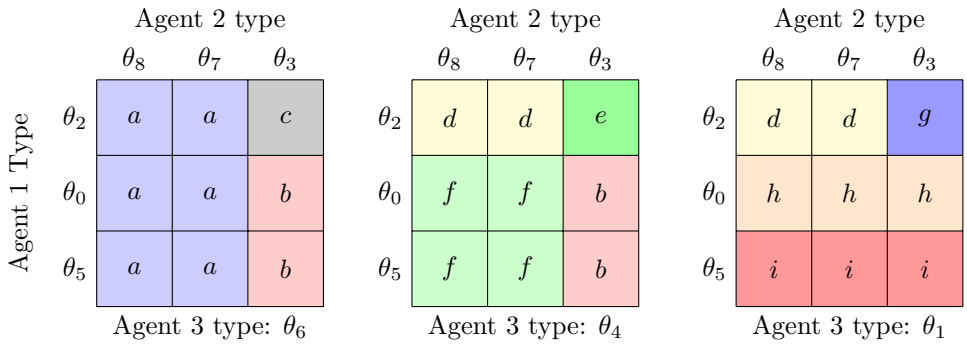


Figure 7: Counterexample ϕ^{SPA} with $n = 3$ and $|\Theta| = 9$.

□

A.6 Proof of Proposition 6.

Proposition 6. *Assume $|\Theta| \geq 4$ and $|N| \geq 4$. Any protocol $P \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$ for an efficient, uniform-price double auction price rule produces contextual privacy violations for $\boldsymbol{\theta}$ where two agents determine the clearing price.*

Proof. We construct collective but not individual pivotality. Consider four agents i, j, k, and ℓ, two types $\underline{\theta}, \bar{\theta} \in \Theta$ and a partial type profile $\boldsymbol{\theta}_{-ijk\ell}$ such that

$$\text{median}(\boldsymbol{\theta}_{-ijk\ell}) = \{\underline{\theta}, \bar{\theta}\}.$$

Endowments are arbitrary.

Consider outcomes in type profiles $(\theta_i, \theta_j, \theta_k, \theta_\ell, \boldsymbol{\theta}_{-ijk\ell})$ as represented on the left side in Figure 8.

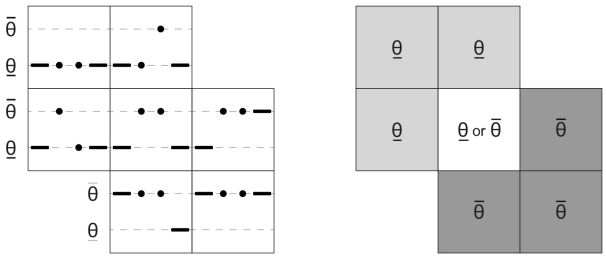


Figure 8: Constructing collective but not individual pivotality for the double auction. Combinations of types for agents i, j, k, ℓ (left); outcomes in an efficient uniform-price choice rule (right).

Denote $\underline{\theta}$ the outcome in which the market clearing price is $\underline{\theta}$ and let $\bar{\theta}$ be the outcome in

which the market clearing price is $\overline{\theta}$. Consider first the top square which holds the types of agents i and ℓ fixed and varies the types of agents j and k. Efficiency requires $\phi(\underline{\theta},\underline{\theta},\underline{\theta},\underline{\theta},\boldsymbol{\theta}_{-ijk\ell}) = \phi(\underline{\theta},\underline{\theta},\overline{\theta},\underline{\theta},\boldsymbol{\theta}_{-ijk\ell}) = \phi(\underline{\theta},\overline{\theta},\underline{\theta},\underline{\theta},\boldsymbol{\theta}_{-ijk\ell}) = \underline{\theta}$.

Now consider the bottom square which holds the types of agents j and k fixed and varies the types of agents i and ℓ. Efficiency requires that $\phi(\overline{\theta},\overline{\theta},\overline{\theta},\underline{\theta},\boldsymbol{\theta}_{-ijk\ell}) = \phi(\underline{\theta},\overline{\theta},\overline{\theta},\overline{\theta},\boldsymbol{\theta}_{-ijk\ell}) = \phi(\overline{\theta},\overline{\theta},\overline{\theta},\overline{\theta},\boldsymbol{\theta}_{-ijk\ell}) = \overline{\theta}$.

The outcome under the type profile in the box that conjoins the two squares, $(\underline{\theta},\overline{\theta},\overline{\theta},\underline{\theta},\boldsymbol{\theta}_{-ijk\ell})$ must be *either* $\underline{\theta}$ or $\overline{\theta}$. If it is $\underline{\theta}$, then we can apply Theorem 1 for the bottom k-ℓ-square. If it is $\overline{\theta}$, then we can apply this statement in the top i-j-square. □

A.7 Proof of Proposition 7.

Proposition 7. *Assume $|N| \geq 2$ and $|X| \geq 2$ and the preference domain of single-peaked preferences. Any protocol $P \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$ for a generalized median voting rule that is neither the maximum nor the minimum rule produces contextual privacy violations.*

Proof. The proof is related to the one for Proposition 6, however simplified as the median is unique for an odd number of inputs, as is the case for a generalized median voting rules. Consider two adjacent types $\theta, \theta' \in \Theta$ and $\boldsymbol{\theta}_{-ij} \in \boldsymbol{\Theta}_{-ij}$ such that

$$\theta = \text{median}(\boldsymbol{\theta}_{-ij}, k_1, k_2, \ldots, k_{n-1})$$

and exactly one type in $\boldsymbol{\theta}_{-ij}, k_1, k_2, \ldots, k_{n-1}$ is θ. The type profiles $(\theta,\theta,\boldsymbol{\theta}_{-ij})$, $(\theta',\theta,\boldsymbol{\theta}_{-ij})$ and $(\theta,\theta',\boldsymbol{\theta}_{-ij})$ will all result in outcome θ. However, $(\theta',\theta',\boldsymbol{\theta}_{-ij})$ will result in θ'. This hence produces an instance of collective pivotality without individual pivotality and shows that there must be a contextual privacy violation for at least one of the agents i, j. □

A.8 Proof of Theorem 2.

Theorem 2. *Let $(P,\boldsymbol{\sigma})$ be a $\mathfrak{S}_{\text{IE}}$-protocol for ϕ, where ϕ exhibits interval pivotality. There is a bimonotonic protocol $(P',\boldsymbol{\sigma}')$ that has the same set of contextual privacy violations as $(P,\boldsymbol{\sigma})$, i.e. $\Gamma(P,\boldsymbol{\sigma},\phi) = \Gamma(P',\boldsymbol{\sigma}',\phi)$.*

Proof. We begin with an arbitrary protocol and transform it with two operations into a bimonotonic protocol that has the same set of contextual privacy violations: *filling-in* and *scrubbing*. By the definition of individual elicitation, we can identify queries with partitions of an agent's type space.

We begin with an auxiliary definition that will be used in the anchoring and filling-in steps.

Definition (v-before-v' injected protocol). Let $P = (V,E)$ be a protocol. Let $s_v \colon \mathbf{M} \to \text{children}(v)$ be a query and $\sigma_{iv} \colon \Theta \to M$ be a strategy for the queried agent. Let $v' \in V$ be a non-root node

with parent u. Denote subtree(v') to be the subtree from v' in P. Define the v-before-v'-injected protocol $P_{v,v'}$ to be the protocol in which u has a single child v (children(u) = $\{v\}$) and all children of v are followed by subtree(v').

Note that subtree(v') refers to the sequence of queries s_w for nodes w in the subtree following v' in the original protocol P. When we inject v before v' in the protocol $P_{v,v'}$, all children of v are followed by the sequence of queries s_w from the original protocol P. See Figure 9 for an illustration of injection.

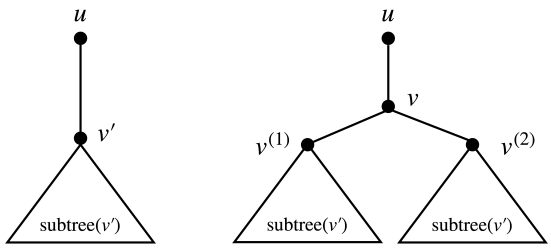


Figure 9: Protocol P (left); v-before-v' injected protocol $P_{v',v}$ (right). Here, in $P_{v,v'}$, the nodes $\{v^{(1)}, v^{(2)}\}$ = children(v) are "clones" of v' in original protocol P.

We denote $i(v)$ as the agent who receives a query at node v.

Definition. Let Θ be a finite ordered type space and denote by succ(θ) the adjacent type higher than θ in Θ (if it exists) and by pred(θ) the adjacent type lower than θ in Θ (if it exists). Also, let s_v be a query for node v. We call $\underline{\theta}_{v,i}$ the *lowest separator* if

$$\underline{\theta}_{v,i} = \min\{\theta_i \in \Theta \mid \exists \boldsymbol{\theta}_{-i} : (\theta_i, \boldsymbol{\theta}_{-i}) \in \boldsymbol{\Theta}_v \text{ and } s_v(\sigma_i(\text{succ}(\theta_i), v), \boldsymbol{\sigma}_{-i}(\boldsymbol{\theta}_{-i}, v)) \neq s_v(\sigma_i(\theta_i, v), \boldsymbol{\sigma}_{-i}(\boldsymbol{\theta}_{-i}, v))\}$$

and we call $\overline{\theta}_{v,i}$ the *highest separator* if

$$\overline{\theta}_{v,i} = \max\{\theta_i \in \Theta \mid \exists \boldsymbol{\theta}_{-i} : (\theta_i, \boldsymbol{\theta}_{-i}) \in \boldsymbol{\Theta}_v \text{ and } s_v(\sigma_i(\text{pred}(\theta_i), v), \boldsymbol{\sigma}_{-i}(\boldsymbol{\theta}_{-i}, v)) \neq s_v(\sigma_i(\theta_i, v), \boldsymbol{\sigma}_{-i}(\boldsymbol{\theta}_{-i}, v))\}.$$

That is, the lowest separator at node v is the lowest element in the type space that is distinguished from its next lowest type at v. The highest separator at node v is the highest element in the type space that is distinguished from its next highest type at node v. If these sets are empty, the query is trivial in that it does not affect the knowledge of the designer. It is then without loss to drop this query from the protocol.

Lemma 2. *Let ϕ be interval pivotal, $i(v) = i(v')$, and $v' \in \text{subtree}(v)$. Define*

$$\begin{aligned} \underline{\theta}_{v,v'} &= \min\{\underline{\theta}_{v,i}, \underline{\theta}_{v',i}\} \\ \overline{\theta}_{v,v'} &= \max\{\overline{\theta}_{v,i}, \overline{\theta}_{v',i}\} \end{aligned} \tag{5}$$

where $\underline{\theta}_{v,i}, \overline{\theta}_{v,i}$ are the highest and lowest separators at v as defined above (and similarly for v'). Then, for any $\tilde{\theta} \in [\underline{\theta}_{v,v'}, \overline{\theta}_{v,v'}]$, a threshold query *$v''$ with threshold $\tilde{\theta}$ satisfies*

$$\Gamma(P, \boldsymbol{\sigma}, \phi) = \Gamma(P_{v'',v'}, \boldsymbol{\sigma}', \phi),$$

where $\boldsymbol{\sigma}'$ is a threshold strategy with threshold $\tilde{\theta}$ at v'' and the same as $\boldsymbol{\sigma}$ on all other nodes and all other agents.

In other words, consider an agent that is asked two queries v and v' one after the other on one path through the protocol. Inserting a threshold query *in between* v and v', where the threshold $\tilde{\theta}$ lies *between* the highest and lowest types that v and v' can distinguish, this insertion will result in no change to the set of contextual privacy violations.

Proof. We show first that $\Gamma(P_{v'',v'}, \boldsymbol{\sigma}', \phi) \subseteq \Gamma(P, \boldsymbol{\sigma}, \phi)$. Let $(i, \boldsymbol{\theta}) \in \Gamma(P_{v'',v'}, \boldsymbol{\sigma}', \phi) \setminus \Gamma(P, \boldsymbol{\sigma}, \phi)$ for contradiction. Then there is a θ_i' such that $(\theta_i, \boldsymbol{\theta}_{-i})$ and $(\theta_i', \boldsymbol{\theta}_{-i})$ are distinguished at v'' and

$$\phi(\theta_i, \boldsymbol{\theta}_{-i}) = \phi(\theta_i', \boldsymbol{\theta}_{-i}).$$

By symmetry, it is without loss to assume $\theta_i \leq \theta_i'$. As θ_i and θ_i' are distinguished, and by the definition of threshold queries, it must be that $\theta_i \leq \tilde{\theta} < \theta_i'$.

We will now show that $\boldsymbol{\theta}$ has a contextual privacy violation for agent i at node v or node v' in P, leading to a contradiction. As $\phi(\theta_i, \boldsymbol{\theta}_{-i}) = \phi(\theta_i', \boldsymbol{\theta}_{-i})$ and because of interval pivotality, there are two cases, corresponding to the "upper" or "lower" interval over which $\theta_i \mapsto \phi(\theta_i, \boldsymbol{\theta}_{-i})$ is constant:

(a) $\phi(\hat{\theta}_i, \boldsymbol{\theta}_{-i}) = \phi(\theta_i', \boldsymbol{\theta}_{-i})$ for all types $\hat{\theta}_i \leq \theta_i'$ (in particular this holds for $\hat{\theta}_i = \underline{\theta}_{v,v'}$ by the definition of the lowest separator), or

(b) $\phi(\hat{\theta}_i, \boldsymbol{\theta}_{-i}) = \phi(\theta_i, \boldsymbol{\theta}_{-i})$ for all types $\hat{\theta}_i > \theta_i$ (in particular this holds for $\hat{\theta}_i = \overline{\theta}_{v,v'}$ by the definition of the highest separator).

For the first case (a), for notational simplicity call v the node that attains the minimum in (5). By definition of $\underline{\theta}_{v,v'}$, the types $\underline{\theta}_v, \text{succ}(\underline{\theta}_v)$ are distinguished at v. There are two further cases within case (a):

- $(\theta_i, \boldsymbol{\theta}_{-i})$ and $(\theta_i', \boldsymbol{\theta}_{-i})$ are distinguished at v. In this case, both type profiles produce contextual privacy violations for agent i at v.

- $(\theta_i, \boldsymbol{\theta}_{-i})$ and $(\theta_i', \boldsymbol{\theta}_{-i})$ are *not* distinguished at v, but this means that θ_i is distinguished from either of $\underline{\theta}_v, \text{succ}(\underline{\theta}_v)$, or both. So, $(i, \boldsymbol{\theta})$ produces a contextual privacy violation with either $(\underline{\theta}_v, \boldsymbol{\theta}_{-i})$ or $(\text{succ}(\underline{\theta}_v), \boldsymbol{\theta}_{-i})$.

For the second case (b), we can follow similar reasoning, but with flipped inequality signs. In this case, one proceeds by showing that a contextual privacy violation is produced at the node that attains the maximum in (5).

For the converse direction, observe that $P_{v'',v'}$ reveals weakly more information to the designer than P. Hence, also $\Gamma(P, \boldsymbol{\sigma}, \phi) \subseteq \Gamma(P_{v'',v'}, \boldsymbol{\sigma}', \phi)$. □

Using Lemma 2, we will "fill-in" all queries between the highest and lowest separator. This results in a *filled-in* protocol.

Definition. We call a protocol P *filled-in* if (a) all queries are threshold queries and (b) all thresholds of adjacent queries (with respect to protocol P) are adjacent in Θ.

That is, in a filled-in protocol, every query is a threshold query and the threshold for every query adjacent in the protocol is also adjacent in the type space.

Lemma 3. *Let ϕ be an interval pivotal choice rule. Then, for any protocol $(P, \boldsymbol{\sigma})$, there is a filled-in protocol $(P', \boldsymbol{\sigma})$ such that $\Gamma(P, \boldsymbol{\sigma}, \phi) = \Gamma(P', \boldsymbol{\sigma}', \phi)$.*

Proof. We prove this lemma in three steps: anchoring, inserting and deleting.

Step 0: Grounding. We first add trivial queries $s_i(\boldsymbol{\theta}) = r_i$ to all agents in the beginning, where r_i can be thought of as a copy of the root node. These allow us to perform protocol injection on initial queries to agents and resolves issues for protocols where an agent only gets a single query. Inserting trivial queries affects neither computability of the choice rule nor contextual privacy violations.

Step 1: Anchoring. We then introduce threshold queries at the highest $(\overline{\theta}_{v,v'})$ and lowest $(\underline{\theta}_{v,v'})$ separators of adjacent queries v, v' to the same agent, that is, before the later in P of v, v', $i(v) = i(v')$. This leads to the introduction of at most $2|V|$ many new queries and does not affect computability of the choice rule or contextual privacy violations.

Step 2: Inserting. For any pair of threshold queries (with respect to the protocol P), v, v' to the same agent, whose thresholds are not adjacent in θ, insert threshold queries for all thresholds between $\text{thresh}(v)$ and $\text{thresh}(v')$ (in the type space) before the later of v, v'. This leads to at most $2|V||^2\Theta|$ many queries.

Step 3: Deleting. For any non-threshold queries v, all thresholds between $\underline{\theta}_v$ and $\overline{\theta}_v$ were added in Steps 1 and 2. This implies that all non-threshold queries can be deleted without affecting computability of the choice rule. Additionally, their deletion cannot affect contextual privacy violations because any distinguished types under the queries are distinguished at at least one of the added threshold queries.

The resulting protocol is filled-in. □

A filled-in protocol is almost bimonotonic. But, it may have sequences of threshold queries to a single agent that increase and then decrease (or vice versa) in the thresholds. So, we use a *scrubbing* operation to dispose of non-monotonicity in the threshold queries.

Let $(P, \boldsymbol{\sigma})$ be a protocol. A query v that distinguishes no type profiles in $\boldsymbol{\Theta}_v$ is *trivial*. (An example of trivial queries are duplicates, queries for which there is an earlier, identical query).

To create a protocol without trivial queries, we delete such queries "from the bottom up", i.e. from the leaves to the root. We call this operation *scrubbing*. For any trivial query v, there is a unique $w \in \text{children}(v)$ such that $\boldsymbol{\Theta}_w \neq \emptyset$. To delete v, attach subtree(w) to parent(v) and delete subtree(w') for all $w' \in \text{children}(v) \setminus \{w\}$. This operation terminates after at most $|V(P)|$ rounds, and neither affects computability of the choice rule nor contextual privacy violations. The scrubbing operation is illustrated in Figure 10.

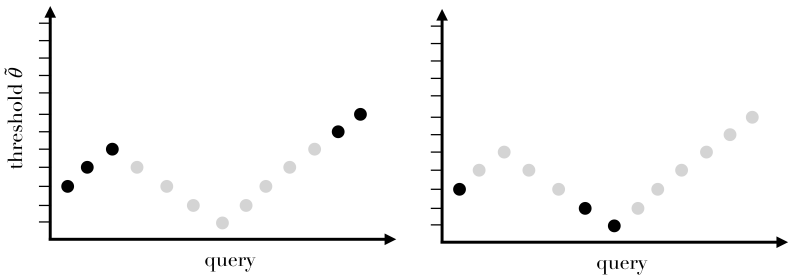


Figure 10: Queries to agent i in protocol P, scrubbed after affirmative answer to initial query (left); Queries to agent i in protocol P scrubbed after negative answer to initial query (right). The x-axis represents the order of the queries in protocol P. The y-axis is the threshold of threshold queries in protocol P. Each query in the original protocol is a dot, queries deleted during scrubbing are gray dots.

This results in a bimonotonic protocol. An agent's response to the initial query tells us that either all queries with higher thresholds are trivial, or that all queries with lower thresholds are trivial. Consider the former case, the latter is analogous. The agent's initial response indicates that they are below the initial threshold. Because the protocol is filled-in, the next non-trivial query must have a threshold below and adjacent to the initial threshold. The response to this query either fully reveals the agent's type (in which case all future queries are trivial) or renders any threshold above this one trivial. This continues until the protocol terminates, leading to a sequence of queries with monotonically decreasing thresholds.

□

A.9 Proof of Theorem 3.

Theorem 3. *For the second-price auction rule ϕ^{SPA}, there is:*

(a) a unique minimal set of contextual privacy violations for W- *and* LPL*-protecting $\mathfrak{S}_{\text{IE}}$-protocols,*
(b) a unique minimal set of contextual privacy violations for L- *and* HPW*-protecting $\mathfrak{S}_{\text{IE}}$-protocols.*

Two bimonotonic protocols attain these sets of contextual privacy violations, which we call the ascending-join protocol *and the* overdescending-join protocol*, respectively. Both of these protocols are maximally contextually private among $\mathfrak{S}_{\text{IE}}$-protocols for ϕ^{SPA}.*

We prove a more general version of Theorem 3 that clarifies the structure of the argument, and shows that our proof technique applies more broadly. In particular, we show that for any k-item uniform price auction where the price is set by the $(k+1)^{\text{st}}$ agent's value, there is a unique maximal protocol for contextual privacy.

We will denote k-item uniform $(k+1)^{\text{st}}$ price auction rules by $\phi^{k\text{-PA}}$. Before stating our more general result, we must define protection sets for $\phi^{k\text{-PA}}$ that are analogous to the protection sets defined for ϕ^{SPA} from the main text. We call an agent a *winner* if she is allocated a good. If she also does not determine the price, then she is a *loser*. For the $(k+1)^{\text{st}}$-price auction, we define the sets of winners and losers as follows:

$$\begin{aligned}\text{W} &= \{(i, \boldsymbol{\theta}) \mid i \text{ is a winner at } \boldsymbol{\theta}\}\\ \text{L} &= \{(i, \boldsymbol{\theta}) \mid i \text{ is neither a winner nor determines the price at } \boldsymbol{\theta}\}.\end{aligned}$$

We also have subsets of the winner and loser protection sets, which are exactly the same as the main text—we reproduce the definitions here for convenience. Let $d(\boldsymbol{\theta})$ be the index of the agent who determines the price at type profile $\boldsymbol{\theta}$. We define the protection sets LPL and HPW as follows:

$$\begin{aligned}\text{LPL} &= \text{L} \cap \{(i, \boldsymbol{\theta}) \mid i > d(\boldsymbol{\theta})\}\\ \text{HPW} &= \text{W} \cap \{(i, \boldsymbol{\theta}) \mid i < d(\boldsymbol{\theta})\}.\end{aligned}$$

The generalization of Theorem 3 is as follows.

Theorem 4. *For the uniform $(k+1)^{st}$-price k-item auction rule $\phi^{k\text{-PA}}$, there is:*

(a) a unique minimal set of contextual privacy violations for W- *and* LPL*-protecting $\mathfrak{S}_{\text{IE}}$-protocols,*
(b) a unique minimal set of contextual privacy violations for L- *and* HPW*-protecting $\mathfrak{S}_{\text{IE}}$-protocols.*

Two bimonotonic protocols attain these sets of contextual privacy violations, which we call the ascending-join protocol *and the* overdescending-join protocol*, respectively. Both of these protocols are maximally contextually private among $\mathfrak{S}_{\text{IE}}$-protocols for $\phi^{k\text{-PA}}$.*

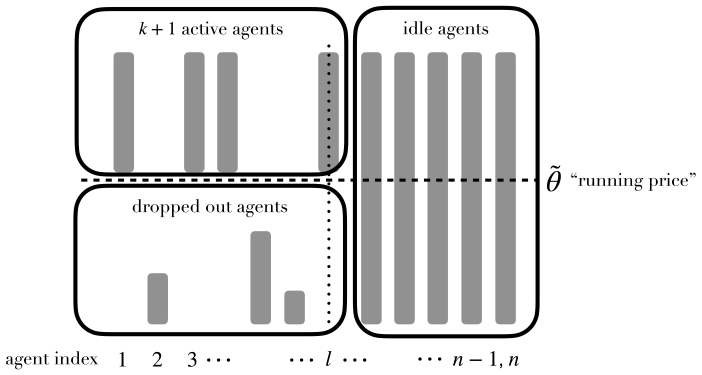


Figure 11: W- and LPL- Protective Information State

Proof. The proofs for both (a) and (b) are constructive. We show that at each point in the protocol, the principal has a unique choice for which agent to ask next, and for which threshold. This relies on our characterization theorem Theorem 2.

Our argument will roughly structure the protocol into "rounds," after which the designer's information has a particular structure. Whenever such a structure is reached, a new round begins.

We begin with part (a). The relevant designer information structure for the ascending-join protocol: W- *and* LPL-*protective information states.*

Definition. We call a set of type profiles $\mathbf{\Theta}_v \subseteq \mathbf{\Theta}$ a W- and LPL-*protective information state* if there is a type $\tilde{\theta} \in \Theta$ such that (recall that $\Theta_{v,i}$ is the projection of $\mathbf{\Theta}_v$ onto the i^{th} component):

1. There are exactly $k+1$ agents i such that $\Theta_{v,i} = \{\theta_i \mid \theta_i > \tilde{\theta}\}$.
2. For agents $i = 1, 2, \ldots, l$ with l being the highest index among agents who ever got a question (i.e. $l := \max\{l' \colon \Theta_{v,l'} \neq \Theta\}$), it must be that, either:
   (i) $\Theta_{v,i} = \{\theta_i \mid \theta_i > \tilde{\theta}\}$ ("*dropped-out agent*"), or
   (ii) $\Theta_{v,i} \subseteq \{\theta_i \mid \theta_i \leq \tilde{\theta}\}$ ("*active agent*").

We call all other agents *idle*.

For a protective information state $\mathbf{\Theta}_v$, we call $\tilde{\theta}$ the *running price*. Sets that are protective have $k+1$ agents about whom it is known only that their type is strictly above the running price (active agents). Nothing is known about agents with indices "to the right" of the agent with the highest index among agents who have ever been asked a question (idle agents). All other agents have a type weakly below the running price (dropped-out agents). We illustrate this definition in Figure 11.

Definition. We say that a query v is ϕ-redundant if for any θ_i, θ_i', $\boldsymbol{\theta}_{-i}$ such that $(\theta_i, \boldsymbol{\theta}_{-i})$ and $(\theta_i', \boldsymbol{\theta}_{-i})$ are distinguished at v, we have

$$\phi(\theta_i, \boldsymbol{\theta}_{-i}) = \phi(\theta_i', \boldsymbol{\theta}_{-i}).$$

We make the following observation, which is direct from the definitions.

Lemma 4. *Let P be a protocol for ϕ and v be a ϕ-redundant query in P. Any protocol P' that removes v and continues with* any *of the children of v from P is still a protocol for ϕ. P' is weakly more contextually private than P.*

We next make an additional observation about the structure of protection sets to more easily reason about which queries are forbidden under a given protection set. To this end, the structure of a *closed* protection set is helpful.

Definition (Closed protection sets). We call a protection set A *downward closed* if we have that for $(i, (\theta_i, \boldsymbol{\theta}_{-i})) \in A$ and $\theta_i' \leq \theta_i$,

$$(i, (\theta_i', \boldsymbol{\theta}_{-i})) \in A.$$

A protection set is A *upward closed* if we have that for $(i, (\theta_i, \boldsymbol{\theta}_{-i})) \in A$ and $\theta_i' \geq \theta_i$,

$$(i, (\theta_i', \boldsymbol{\theta}_{-i})) \in A.$$

If a model is downward closed or upward closed, we call it *closed*.

Definition (ϕ-closed protection sets). If A is a closed protection set, and for any $i \in N$, $\theta_i, \theta_i' \in \Theta$ and $\boldsymbol{\theta}_{-i} \in \boldsymbol{\Theta}_{-i}$, such that $(i, (\theta_i, \boldsymbol{\theta}_{-i})), (i, (\theta_i', \boldsymbol{\theta}_{-i})) \in A$, it holds

$$\phi(\theta_i, \boldsymbol{\theta}_{-i}) = \phi(\theta_i', \boldsymbol{\theta}_{-i}),$$

we call it ϕ-closed.

The property ϕ-closedness may be seen as a minimality property for $\mathfrak{S}_{\text{IE}}$ protocols: Any distinction of type profiles in a ϕ-closed protection set would lead to a contextual privacy violation for the queried agent.

Lemma 5. *Let A be a ϕ-closed protection set. A protocol P has a contextual privacy violation in A if it contains a query v and two type profiles $(\theta_i, \boldsymbol{\theta}_{-i}), (\theta_i', \boldsymbol{\theta}_{-i})$ that P distinguishes such that $(i, (\theta_i, \boldsymbol{\theta}_{-i})), (i, (\theta_i', \boldsymbol{\theta}_{-i})) \in A$.*

Proof. We show this lemma for upward closedness. The proof for downward closedness is analogous. We say that "i is in A under $\boldsymbol{\theta}$" if $(i, \boldsymbol{\theta}) \in A$.

Consider a query v in protocol P and two type profiles $\boldsymbol{\theta}, \boldsymbol{\theta}'$ that are distinguished at v such that i is in A for both $(\theta_i, \boldsymbol{\theta}_{-i})$ and $(\theta'_i, \boldsymbol{\theta}_{-i})$. Without loss we may assume that $\theta_i < \theta'_i$. As A is upward closed, it must be that i is in A also under $(\theta'_i, \boldsymbol{\theta}_{-i})$, and does not change the outcome, by ϕ-closedness. This means that there is a contextual privacy violation in the protection set A. □

Note that the protection sets we defined in this section (that is, W, L, HPW, LPL) are $\phi^{k\text{-PA}}$-closed.

Lemma 6. *There is a unique sequence of non-redundant,* W- *and* LPL*-protecting queries on* $\boldsymbol{\Theta}$ *that leads to a* W- *and* LPL*-protective information state or termination.*

Proof. We may restrict to bimonotonic protocols by Theorem 2, in particular to protocols that only use threshold queries. By Lemma 5, any query with a threshold higher than the lowest type violates winner privacy because both a negative answer and an affirmative answer may lead to the queried agent winning. This continues to hold until k higher-priority agents have given an affirmative answer to a threshold query about the lowest type. We will check that the $(k+1)^{\text{st}}$ query will still have the lowest type as a threshold, and leads to a W- and LPL-protective information state. For the first k queries, we only need to determine the order in which agents are asked.

Denote $h \in \{1, 2, \ldots, n\}$ the number affirmative answers given so far. Call agents for which a type higher than the running price is not ruled out "remaining". We show by induction over h the following statement:

$$\text{The only non-redundant query that does not violate winner or low-priority loser privacy is to the } (k+1-h)^{\text{th}} \text{ remaining agent in priority order.} \tag{6}$$

Note that all queries to agents who dropped out are redundant and it is without loss to abstract from them by Lemma 4. Also note that all queries to agents $1, 2, \ldots, k-h$ make it possible that the queried agent is a winner both in the case of an affirmative and a negative answer, hence violating W privacy by Lemma 5. All queries to agents $k-h+2, k-h+3, \ldots, n$ make it possible that the queried agent is a low-priority loser in the case of an affirmative or a negative answer, hence violating LPL privacy. Asking the $(k+1-h)^{\text{th}}$ remaining agent does not lead to a W or LPL contextual privacy violation.

(As a concrete example, consider the very first query. We may not ask agent $1, 2, \ldots, k$ as they might win both with an affirmative and a negative answer. We may also not ask agents $k+2, k+3, \ldots, n$, as they might be low-priority losers in case of a positive and a negative answer. Hence, agent $k+1$ must be queried. What's important is that a negative answer from the $(k+1)^{\text{st}}$ unambiguously means that they are not a winner. An affirmative answer from this agent means that they are not a low-priority loser.)

This determines the order of the first k queries. Observe that agents $2, \dots, m$ for some $m \leq n$ have been asked by the property (6). If we show that agent 1 is the only agent that can be queried, and must be queried for the lowest type, then the resulting information of the designer is a W- and LPL-protective information state (unless the protocol can terminate before this is the case).

We distinguish three cases. First, asking agents $1, \dots, m$ for a higher threshold might lead to them being a winner for both an affirmative and a negative answer rendering these queries forbidden by Lemma 5. Second, asking agents $m+1, \dots, n$ for any threshold might lead them to being a low-priority loser for both an affirmative or a negative answer, rendering these forbidden by Lemma 5. Third, asking agent 1 for any threshold higher than the lowest type might lead to them being a winner for both an affirmative and a negative answer, making this query forbidden, also by Lemma 5. Asking agent 1 a threshold query with the lowest type does not lead to W and LPL violations. Note that asking agent 1 complies with the agent order in (6). Hence, (6) can be seen as the definition of the ascending-join protocol.

Either the designer reaches information state $\mathbf{\Theta}_v$ in which $k+1$ agents have given an affirmative answer. In this case, the designer's information state $\mathbf{\Theta}_v$ is W- and LPL-protective. Or, all agents except for k have given negative answers. In this case, the protocol may terminate (allocating to the remaining k agents at the current running price). □

Lemma 7. *On a* W- *and* LPL-*protective set* $\mathbf{\Theta}_v$*, there is a unique sequence of non-redundant,* W- *and* LPL-*protecting queries, that leads to a* W- *and* LPL-*protective set or termination.*

Proof. We reduce this to Lemma 6. Observe that any query to a dropped-out agent is redundant. We may therefore drop them from consideration. Also observe that any query with a threshold lower than the running price is either trivial (if it is to an active agent), redundant (if it is to a dropped-out agent) or could lead to the queried agent being a high-priority winner for both an affirmative and a negative answer (if it is to an idle agent). Hence, we may reduce to a situation in which the designer faces only active agents and needs to compute the choice rule $\phi^{k\text{-PA}}$ on a restricted type space $\Theta' \coloneqq \{\theta' \mid \theta' > \tilde{\theta}\}$, where $\tilde{\theta}$ is the current running price, and apply Lemma 6. □

This concludes the proof of part (a). We now turn to part (b), using a similar proof strategy.

We first need to define an analogous version of a protective information state.

Definition. We call a set of type profiles $\mathbf{\Theta}_v \subseteq \mathbf{\Theta}$ a k-item L- and HPW-*protective information state* if there is a type $\tilde{\theta} \in \Theta$ such that (recall that $\Theta_{v,i}$ is the projection of $\mathbf{\Theta}_v$ onto the i^{th} component):

1. There are exactly $n - k + 1$ agents i such that $\Theta_{v,i} = \{\theta_i \mid \theta_i \leq \tilde{\theta}\}$.

2. For agents $i = l, \dots, n$ with l being the smallest index among agents who ever got a question (i.e. $l \coloneqq \min\{l \colon \Theta_{v,l} \neq \Theta\}$), it must be that, either:

   (i) $\Theta_{v,i} \subseteq \{\theta_i \mid \theta_i > \tilde{\theta}\}$ ("*winners*"), or

(ii) $\Theta_{v,i} = \{\theta_i \mid \theta_i \leq \tilde{\theta}\}$ ("*active agent*").

We call all other agents *idle*.

Because of Theorem 2, we may restrict to bimonotonic protocols. As in part (a), we will show that there is a unique sequence of non-redundant queries within each round. The following two lemmas conclude this part of the proof. The first lemma shows that after a "round," the designer's information state is L- and HPW-protective information state if the protocol does not terminate. The proof will use the reverse priority order which greatly simplifies notation (in particular, the management of indices).

Lemma 8. *There is a unique sequence of non-redundant,* L- *and* HPW*-protecting queries on* $\boldsymbol{\Theta}$ *that leads to a* L- *and* HPW*-protective information state or termination.*

Proof. By Lemma 5, any query with a threshold lower than the highest type violates loser privacy because both a negative answer and an affirmative answer may lead to the queried agent losing. This continues to hold until $n-k-1$ lower-priority agents have given a negative answer to a threshold query about the highest type. We will check that the $(n-k)^{\text{th}}$ query will still have the highest type as a threshold, and leads to a L- and HPW-protective information state. For the first $n-k$ queries, we only need to determine the order in which agents are queried.

Denote $h \in \{1, 2, \ldots, n\}$ the number negative answers given so far. Call agents for which a type lower than the running price is not ruled out "remaining". We show by induction over h the following statement:

$$\text{The only non-redundant query that does not violate loser or high-priority winner privacy is to the } (n-k-h)^{\text{th}} \text{ remaining agent in reverse priority order.} \tag{7}$$

For the rest of this proof, we will index agents in reverse priority order, meaning that agent 1 is the lowest-priority agents, and agent n is the highest-priority agent.

Note that all queries to agents who have been identified to be winners are redundant and it is without loss to abstract from them by Lemma 4. Also note that all queries to any remaining agent $i \leq n-k-h-1$ make it possible that the queried agent is a loser both in the case of an affirmative and a negative answer, hence violating L privacy by Lemma 5. All queries to remaining agents $i \geq n-k-h+1$ make it possible that the queried agent is a high-priority winner in the case of an affirmative or a negative answer, hence violating HPW privacy. Asking the $(n-k-h)^{\text{th}}$ remaining agent does not lead to a W or LPL contextual privacy violation.[18]

[18] It may help to translate temporarily to the non-reversed priority order and consider the concrete example of the first query. We may not ask agent $n, n-1, \ldots, k+2$ as they might lose both with an affirmative and a negative answer. We may also not ask agents $k, k-1, \ldots, 1$, as they might be high-priority winner in case of a positive and a negative answer. Hence, agent $k+1$ must be queried. What's important is that an affirmative answer from the $(k+1)^{\text{st}}$ unambiguously means that they are not a loser. A negative answer from this agent means unambiguously

This determines the order of the first $n-k-1$ queries. Observe that agents $2, 3, \ldots, m$ for some $m \leq n$ have been asked by the property (7). If we show that agent 1 is the only agent that can be queried, and must be queried with a threshold that is the highest type, then the resulting information of the designer is a L- and HPW-protective information state (unless the protocol can terminate before this is the case).

We distinguish three cases. First, asking agents $1, \ldots, m$ for a lower threshold might lead to them being a loser for both an affirmative and a negative answer rendering these queries forbidden by Lemma 5. Second, asking agents $2, \ldots, m$ for any threshold might lead them to being a high-priority winner for both an affirmative or a negative answer, rendering these forbidden by Lemma 5. Third, asking agent 1 for any threshold lower than the highest type might lead to them being a loser for both an affirmative and a negative answer, making this query forbidden, also by Lemma 5. Asking agent 1 a threshold query with the highest type does not lead to L and HPW violations. Note that asking agent 1 complies with the agent order in (7). Hence, (7) can be seen as the definition of the overdescending-join protocol. The overdescending join protocol for the example in the main text (with type profile $\boldsymbol{\theta} = (4, 3, 8, 2)$ and for $k = 1$ i.e. a second-price auction rule ϕ^{SPA}) is shown in Figure 12.

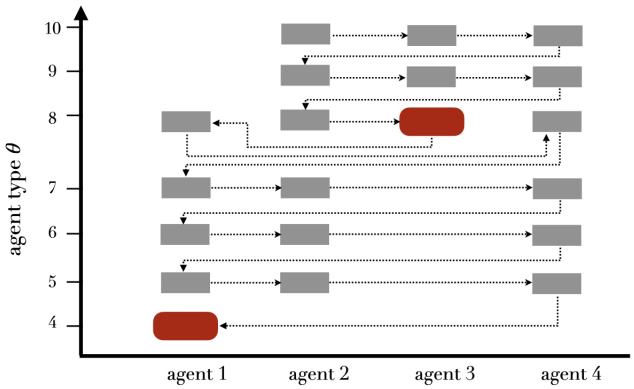


Figure 12: Example—Overdescending-join protocol for $\boldsymbol{\theta} = (4, 3, 8, 2)$ and ϕ^{SPA}. Queries to an agent are represented as rectangles, positioned at the corresponding threshold $\tilde{\theta}$ (gray solid rectangles represent negative responses, red rounded rectangles represent affirmative responses). The path of the dotted line represents the order in which queries are asked. Theorem 3 shows that the overdescending-join protocol is $\mathfrak{S}_{\text{IE}}$-maximally contextually private

There are two possible cases. The first case is one in which the designer reaches information state $\boldsymbol{\Theta}_v$ in which $n-k$ agents have given a negative answer. In this case, the designer's information state $\boldsymbol{\Theta}_v$ is L- and HPW-protective. The second case is one in which at least $k+1$ agents have

that they are not a high-priority winner.

given affirmative answers. In this case, the protocol may terminate (allocating to the k highest priority such agents at the $(k+1)^{\text{st}}$ price).

□

Lemma 9. *On a* L- *and* HPW*-protective set* $\boldsymbol{\Theta}_v$*, there is a unique sequence of non-redundant,* L- *and* HPW*-protecting queries, that leads to a* L- *and* HPW*-protective set or termination.*

Proof. We reduce this to Lemma 8. Observe that any query to a winning agent is redundant. We may therefore drop them from consideration. Also observe that any query with a threshold higher than the running price is either trivial (if it is to a active agent), redundant (if it is to a winner) or could lead to the queried agent being a high-priority winner for both an affirmative and a negative answer (if it is to an idle agent). Hence, we may reduce to a situation in which the designer faces only active agents and needs to compute the choice rule $\phi^{(k-b)\text{-PA}}$ where b is the number of newly identified winners in the last round, on a restricted type space $\Theta' \coloneqq \{\theta' \mid \theta' \leq \tilde{\theta}\}$, where $\tilde{\theta}$ is the current running price. We apply Lemma 8, given this choice rule and restricted type space. □

This concludes the proof. □

A.10 Proof of Proposition 8.

Proposition 8. *The ascending-join protocol and the overdescending-join protocol are incomparable in the contextual privacy order.*

Proof. Consider a type profile $\boldsymbol{\theta}$ where $\theta_i = \theta$ for some type $\theta \in \Theta$. In this case, by the properties of the protocols identified in (6) and (7),

$$(1, \boldsymbol{\theta}) \in \Gamma(P^{\text{AJ}}, \phi^{\text{SPA}}) \setminus \Gamma(P^{\text{OJ}}, \phi^{\text{SPA}})$$

and

$$(n, \boldsymbol{\theta}) \in \Gamma(P^{\text{OJ}}, \phi^{\text{SPA}}) \setminus \Gamma(P^{\text{AJ}}, \phi^{\text{SPA}}),$$

where P^{AJ} and P^{OJ} are the ascending and the overdescending-join protocols. □

A.11 Proof of Proposition 9.

Proposition 9. *The ascending-join protocol implements the second-price auction choice rule* ϕ^{SPA} *in obviously dominant strategies.*

Proof. The protocol presented in Proposition 9 is a *personal-clock auction* in the sense of Li (2017). The "personal clocks" in the ascending-join protocol raise running prices for agents that are active. Given this, Li (2017, Theorem 3), implies that there is an equilibrium in obviously dominant strategies. □

A.12 Proof of Proposition 10.

Proposition 10. *The overdescending-join protocol implements the choice rule ϕ^{SPA} in dominant strategies.*

Proposition. The overdescending-join protocol implements the second price auction choice rule ϕ^{SPA} in dominant strategies.

Proof. There are two types of nodes. Those in which the winner has not yet been determined, and those in which she has been determined. In the latter, all agents are indifferent between all messages, implying the incentive guarantee. In the former, before the winner has been determined, the agent does not affect the price they will be paying for any other fixed set of messages the other agents will send, leading to no profitable deviation from dropping out at an agent's type. □

B Two Contextually Private Protocols

We present two examples of individual elicitation protocols for common social choice functions that do not produce any contextual privacy violations. These protocols use message space $M = \Theta$, hence queries can be identified with partitions of an agent's type space.

B.1 Serial Dictatorships.

Recall the standard object assignment setting, in which agents may receive at most one object $c \in C$. Agents have ordinal preferences over objects, which are private information. So agents' types $\theta \in \Theta$ are preference orders of C where $\succeq_i$ reference to agent i's preference ordering. Outcomes are assignments of objects to agents $\mu \in X = 2^{N \times C}$.

Let $\mu \subseteq N \times C$ be an outcome. A *partial assignment* $N(\mu)$ is the set of agents who have an assigned object in μ, i.e. $N(\mu) = \{i \in N \colon \exists c \in C : (i, c) \in \mu\} \subseteq N$. If $N(\mu) = N$, we call A *complete*. For a partial assignment μ, denote $\mu(i)$ the (at most one) object assigned to agent i.

The *remaining objects* $R(\mu)$ are the objects that do not have an assigned agent in μ, i.e. $R(\mu) \coloneqq \{c \in C \colon \nexists i \in N : (i, c) \in \mu\}$.

The designer wishes to implement the serial dictatorship assignment rule ϕ^{SD}.

To define the serial dictatorship protocol P^{SD} we characterize the nodes and edges of the rooted tree. Fix the permutation $\pi \colon N \to N$ of agents that defines the priority order of the serial dictatorship. The nodes are all partial assignments to agents in $N_i^\pi \coloneqq \{\pi(i') \colon 1 \leq i' \leq i\}$ for any $i \in N$. Edges are between partial assignments μ, μ' such that exactly agents N_i^π resp. N_{i+1}^π are assigned an object, $N(\mu) = N_i^\pi$ and $N(\mu') = N_{i+1}^\pi$, and $\pi(1)$, $\pi(2) \ldots, \pi(i)$ are assigned the same objects.

We define sets of type profiles associated to each node recursively. For an edge (μ, μ'),

$$\boldsymbol{\Theta}_{\mu'} = \boldsymbol{\Theta}_{\mu} \cap \left\{ \boldsymbol{\theta} \in \boldsymbol{\Theta} : \max_{\theta_{\pi(i)}} R(\mu) = \mu'(\pi(i+1)) \right\}.$$

Here, $R(\mu)$ is the set of remaining objects when it is agent $\pi(i)$'s turn in the partial order; $\max_{\theta_{\pi(i)}} R(\mu)$ is the most preferred element of $R(\mu)$ with respect to the preference order $\theta_{\pi(i)}$. If a node is reached that is a complete assignment, the protocol ends, and the complete assignment is computed.

Proposition 12. *The serial dictatorship protocol $P^{SD} \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$ produces no contextual privacy violations for ϕ^{SD}.*[19]

Proposition 12. Consider $\theta_i, \theta_i' \in \Theta$ and a partial type profile for other agents $\boldsymbol{\theta}_{-i} \in \boldsymbol{\Theta}_{-i}$ such that $(\theta_i, \boldsymbol{\theta}_{-i})$ is distinguished from $(\theta_i', \boldsymbol{\theta}_{-i})$. We will show that $\phi(\theta_i, \boldsymbol{\theta}_{-i}) \neq \phi(\theta_i', \boldsymbol{\theta}_{-i})$.

Denote μ the node of distinction. By definition of individual elicitation, this must be a query to agent i. By definition of serial dictatorship, the designer's knowledge at children of μ is given by $\{\boldsymbol{\theta} \in \boldsymbol{\Theta} \mid \max_{\theta_{\pi(i)}} R(\mu) = c\}$ for some $c \in R(\mu)$. Hence, if $\phi(\theta_i, \boldsymbol{\theta}_{-i})$ and $\phi(\theta_i, \boldsymbol{\theta}_{-i})$ are distinguished, agent i must get a different assignment under θ_i and θ_i', hence $\phi(\theta_i, \boldsymbol{\theta}_{-i}) \neq \phi(\theta_i, \boldsymbol{\theta}_{-i})$.

□

In the case of only two agents, serial dictatorship is the unique contextually private and efficient mechanism, as the following example shows.

Example (Contextual Privacy and Serial Dictatorships, $n = 2$). Consider an example of two agents $N = \{1, 2\}$, each of which is allocated an object A or B. The two possible outcomes are $x = \{(1, A), (2, B)\}$ and $x' = \{(1, B), (2, A)\}$.

Table 4 shows possible assignments under efficiency. In the upper right table cell, efficiency requires that the outcome is x. In the lower left cell, efficiency requires that the outcome is x'. In the top left and bottom right cell, where both agents have the same type, efficiency allows either x or x'.

Four different assignments remain. The first two assignments have collective pivotality but not individual pivotality and hence produce contextual privacy violations by Theorem 1. The other two assignments correspond to serial dictatorships with agent orderings $\pi(1) = 1, \pi(2) = 2$ resp. $\pi(1) = 2, \pi(2) = 1$.

[19] The serial dictatorship also does not produce group or individual contextual privacy violations, as discussed in Appendix E.

	A	**B**
A	x or x'	x
B	x'	x or x'

	A	**B**
A	x	x
B	x'	x

	A	**B**
A	x	x'
B	x	x

	A	**B**
A	x	x
B	x'	x'

	A	**B**
A	x'	x
B	x'	x

Table 4: Outcomes for arbitrary efficient 2-agent social choice functions (left); under an efficient choice rule which breaks ties lexicographically (ϕ^{fair}) (mid-left, middle); under a serial dictatorship ϕ^{sd} (mid-right, right)

B.2 First-Price Auction.

While second-price auctions lead to contextual privacy violations, the "descending" or Dutch protocol for the first-price auction rule produces no contextual privacy violations.[20] A descending protocol queries, for each element of the type space $\tilde{\theta}$, in decreasing order, each agent 1 to n on whether their type θ_i is above θ. (This assumes lexicographic tiebreaking. Similar protocols do not produce contextual privacy violations for other deterministic tiebreaking rules). Formalized as a protocol, this leads to a set of nodes $N \times \Theta \times \{0,1\}$ and edges from $((i,\theta,0)$ to $(i+1,\theta,0))$, for $i \in N \setminus \{n\}$ and $\theta \in \Theta$. There are edges from $(n,\theta,0)$ to $(1, \max_{\theta' < \theta} \theta', 0)$. Furthermore, there are edges $((i,\theta,0),(i,\theta,1))$ for all $i \in N$ and $\theta \in \Theta$ corresponding to an agent stating that they have a type at least θ, which leads to them being allocated the good. Hence, the set of terminal nodes is $\Theta \times N \times \{1\}$. The associated set of type profiles is recursively defined as

$$\Theta_{(i+1,\theta,0),i} \coloneqq \Theta_{(i,\theta,0),i} \setminus \{\theta\} \tag{8}$$

$$\Theta_{(i,\theta,1),i} \coloneqq \{\theta\} \tag{9}$$

$$\Theta_{(1,\theta,0),i} \coloneqq \Theta_{(n,\max_{\theta' < \theta} \theta',0),i} \setminus \{\theta\}, \tag{10}$$

where $\Theta_{v,i}$ denotes the projection of $\mathbf{\Theta}_v$ onto agent i's type. For all non-queried agents at node v, and a child w, $\Theta_{w,i} = \Theta_{v,i}$. (8) rules out the type θ for type i when they respond that they are not type θ. (9) identifies an agent's type exactly when they respond they are type θ. (10) rules out the type θ for agent n when they respond they are not θ and leads to the protocol considering the next-lowest type $\max\{\theta' \mid \theta' < \theta\}$. This defines a protocol $P^{\text{desc}} \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$.

Proposition 13. *The descending protocol $P^{desc} \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$ for the first-price rule ϕ^{FP} does not produce any contextual privacy violations.*

Proof of Proposition 13. First note that by construction of the protocol, the first query leading to a singleton possible type space must be a type θ_i attaining $\max_{i \in N} \theta_i$. This implies that the descending protocol is a protocol for the first-price choice rule (with tie-breaking according to the order $1, 2, \ldots, n$).

[20] An analogous observation was made in the computer science literature on decentralized computation, compare Brandt and Sandholm (2005).

To show that P^{desc} does not lead to contextual privacy violations, let v be a node (querying agent i) that distinguishes $(\theta_i, \boldsymbol{\theta}_{-i})$ and $(\theta_i', \boldsymbol{\theta}_{-i})$. Note that terminal nodes cannot distinguish type profiles. Hence, $(\theta_i, \boldsymbol{\theta}_{-i})$ and $(\theta_i', \boldsymbol{\theta}_{-i})$ are distinguished at a node of the form $(i, \tilde{\theta}, 0)$. By definition of the descending protocol, the children of the node $(i, \tilde{\theta}, 0)$ are associated to sets

$$\{\tilde{\theta}\} \text{ and } \{\tilde{\theta}' \in \Theta \colon \tilde{\theta}' < \tilde{\theta}\}.$$

Let, without loss, $\theta_i = \tilde{\theta}$ and $\theta_i' < \tilde{\theta}$. In the former case, the outcome is that agent i gets the good at price $\tilde{\theta}$. By definition of the descending protocol, in the latter case, it is that either agent i does not get the good, or they get it at a price $\tilde{\theta}' < \tilde{\theta}$. In particular, $\phi(\theta_i, \boldsymbol{\theta}_{-i}) \neq \phi(\theta_i', \boldsymbol{\theta}_{-i})$.

□

C Other Maximally Contextually Private Protocols

In Section 4, we derived two maximally contextual private choice rules for the second-price auction rule. In particular, Theorem 3 showed that the ascending-join and overdescending-join protocols are maximally contextually private for the second-price auction rule. We also argued that these protocols give general principles for understanding maximally contextually private protocols: they choose a specific set of agents to protect, and then delay asking questions to those agents.

In general, there will be other ways to get maximally contextually private protocols, through "guessing." That is, the designer may eliminate *all* contextual privacy violations at some type profile by simply verifying a correct "guess" of the relevant statistics of the type profile. For example, the "guessing" protocol for the first-price auction rule is essentially a posted price, and, when correct, it avoids contextual privacy violations altogether.

We illustrate this principle again through the second-price auction rule. Here, we note that the ascending-join and overdescending-join protocols are not the unique maximally contextually private protocols for the second-price auction rule. To show the existence of other maximally contextually private protocols, we need only note that there are protocols that are incomparable in the contextual privacy order to these two. The following example gives on such protocol, the *guessing protocol*.

The *guessing protocol* for the second-price choice rule "guesses" a price $\tilde{\theta} \in \Theta$ and an agent who has exactly this value, and then aims to verify the guess. The protocol $P^{\tilde{\theta}\text{-guess}}$ starts with a threshold query for type $\tilde{\theta}$ for all agents. If all but one agent answer negatively, the protocol asks the agent who the designer guesses has a value of exactly $\tilde{\theta}$ whether this is indeed the case.[21]

Consider again the type profile from the example in Section 4.2. There are 4 agents, and the type profile is $\boldsymbol{\theta}^* = (4, 3, 8, 2)$. Let's consider first the contextual privacy violations of a guessing

[21] Otherwise, if the guess is not correct, the protocol proceeds in any manner compatible with bimonotonicity—or the point we make here, it will not matter exactly how the protocol proceeds when the guess is wrong.

protocol with a guess of $\tilde{\theta} = 4$, and $\{j \mid \theta_j = \boldsymbol{\theta}^*_{[2]}\} = \{1\}$. That is, the designer guesses that the second-highest bid is agent 1's, and that it is 4. The first threshold query in the protocol $P^{4\text{-guess}}$ is for the threshold 5. This protocol begins by asking every agent if their type is strictly greater than 5, beginning with agent 1. All agents except agent 3 answer negatively. Then, the protocol asks only agent 1 if her type is strictly above 4. Agent 1 answers affirmatively. The designer gets enough information to compute the rule, and there are no contextual privacy violations for any agent. That is,

$$N \times \{\boldsymbol{\theta}^*\} \cap \Gamma(P^{\text{AJ}}, \phi^{\text{SPA}}) = \emptyset \tag{11}$$

Now compare (11) to the privacy violations of the ascending-join protocol P^{AJ} at $\boldsymbol{\theta}^*$. As we know from the main text, the ascending-join protocol leads to a violation for agent 2 at $\boldsymbol{\theta}^*$, i.e.

$$(2, \boldsymbol{\theta}^*) \in \Gamma(P^{\text{AJ}}, \phi^{\text{SPA}}). \tag{12}$$

Similarly, the overdescending-join protocol P^{OJ} leads to a violation for the winner, agent 3. That is,

$$(3, \boldsymbol{\theta}^*) \in \Gamma(P^{\text{OJ}}, \phi^{\text{SPA}}). \tag{13}$$

So, we see that *at this particular type profile* $\boldsymbol{\theta}^*$, the violations from the guessing protocol form a subset of the violations of both the ascending-join and overdescending-join protocol.

As P^{AJ} and P^{OJ} are maximally contextually private, it must be that $P^{4-\text{guess}}$ has contextual privacy violations that neither of these protocols have. In particular, there is a maximally contextually private protocol with weakly fewer violations than $P^{4-\text{guess}}$ whose contextual privacy violations are incomparable to those of the ascending-join and overdescending-join protocols.[22]

The guessing protocol suggests an interesting direction for future work: If the designer cares about the *expected* contextual privacy violations, then the guessing protocol may be preferable to the ascending-join or overdescending-join protocols. In other words, if the designer uses their prior $f(\theta)$ to choose a "good guess", and if the designer cares about privacy violations *in expectation*,

[22] To see this in the example considered in the main text, consider another particular type profile $\boldsymbol{\theta}' = (6, 3, 8, 2)$. At this type profile, the violations for the ascending-join and over-descending join protocol remain the same as in (12) and (13), respectively (replacing $\boldsymbol{\theta}^*$ with $\boldsymbol{\theta}'$). However the 4-guessing protocol $P^{4\text{-guess}}$ must produce at least a violation for agent 1, i.e.

$$(1, \boldsymbol{\theta}') \in \Gamma(P^{4\text{-guess}}, \phi^{\text{SPA}}). \tag{14}$$

Given this, we know that it is the case that

$$\Gamma(P^{\text{AJ}}, \phi^{\text{SPA}}) \not\subset \Gamma(P^{4\text{-guess}}, \phi^{\text{SPA}}) \text{ and } \Gamma(P^{4\text{-guess}}, \phi^{\text{SPA}}) \not\subset \Gamma(P^{\text{AJ}}, \phi^{\text{SPA}}),$$

and that

$$\Gamma(P^{\text{OJ}}, \phi^{\text{SPA}}) \not\subset \Gamma(P^{4\text{-guess}}, \phi^{\text{SPA}}) \text{ and } \Gamma(P^{4\text{-guess}}, \phi^{\text{SPA}}) \not\subset \Gamma(P^{\text{OJ}}, \phi^{\text{SPA}}).$$

So, we have shown that there is a protocol that is incomparable to both the ascending-join and the overdescending-join protocols. This implies that there must be some maximally contextually private protocol that is neither the ascending-join nor the overdescending-join protocol, as there must be a protocol that includes the violation $(1, \boldsymbol{\theta}')$ that is also maximal.

then the guessing protocol might be preferable, from a privacy standpoint, to the ascending or overdescending protocols, because the designer can leverage their prior to minimize contextual privacy violations in expectation. But, if the designer can't use their prior to choose a "good guess", because e.g. the prior is diffuse, then the guessing protocol may produce more contextual privacy violations in expectation than the ascending or overdescending protocols.

D Count Elicitation Technology

The framework laid out in Section 2 captures a wide range of possible environments via the elicitation technology. Most of the paper focuses on a particular minimal trust elicitation technology, the individual elicitation technology. Here we define another elicitation technology that corresponds to a similarly simple assumption—it corresponds to an ability for the designer to *anonymize* messages, or otherwise commit to forget the labels of the messages received.

The *count elicitation technology* $\mathfrak{S}_{\text{Count}}$ is able to provide counts of all messages sent, and consists of a single partition

$$\mathfrak{S}_{\text{Count}} = \{\{\mathbf{m} \in \mathbf{M} \colon |\{m_i = m\}| = k\} : m \in M, k \in N\}$$

One example for a binary message space $M = \{0, 1\}$ is a ballot box: Agents that are sending message 1 are consider to vote *for* an issue, those sending message 0 *against*.

The count elicitation technology requires more trust than the individual elicitation technology in the sense that it requires "mediation". To see this, suppose the designer uses the count elicitation technology to ask "How many agents have a message above x?" Then, in order for it to be the case that the designer *only* learns the number, and not the identity of the agents, there must be a technology that anonymizes agents messages when they are sent. In practice, count elicitation technologies could be a ballot box, a third-party mediator, or another trusted anonymization technique.

Our techniques help us to understand that the count elicitation technology has similar properties as the individual elicitation technology. A first statement shows that for any $\mathfrak{S}_{\text{IE}}$-protocol P there is a $\mathfrak{S}_{\text{Count}}$-protocol P' with the same contextual privacy violations.

Proposition 14. *Let $(P, \boldsymbol{\sigma}) \in \mathcal{P}_{\mathfrak{S}_{IE}}$ be a protocol for ϕ. Then, there is a protocol $(P', \boldsymbol{\sigma'}) \in \mathcal{P}_{\mathfrak{S}_{\text{Count}}}$ for ϕ such that $\Gamma(P', \boldsymbol{\sigma'}, \phi) \subseteq \Gamma(P, \boldsymbol{\sigma}, \phi)$.*

Proof. If the message space contains only one element, then only constant social choice functions can be computed, in which case, the result is true.

Otherwise, consider any query in a protocol $P \in \mathcal{P}_{\mathfrak{S}_{\text{IE}}}$. We construct P' by induction on the tree P. By our induction hypothesis, we may assume that the designer's knowledge at node v

in protocol P' is the same as it is at node v in protocol P. For the induction step, we choose a $\mathfrak{S}_{\text{Count}}$-query s'_v and a strategy profile $\boldsymbol{\sigma}$ such that the designer also has the same knowledge at every child of v. By definition of $\mathfrak{S}_{\text{IE}}$, there must be an agent i such that the child reached only depends on the message sent by agent i. Choose any $m \in M$. i keeps their messaging strategy, all other agents send message m. The protocol determines the outcome based on the observed counts: $n-1$ agents are sending message m in any outcome, so the message of agent i can be determined, and the child can be computed.

The so constructed protocol has the same set of distinguished types. Hence, it is still a protocol for ϕ and has the same contextual privacy violations. □

This result may appear somewhat unnatural. The construction of the protocol P' strongly relies on the designer's ability to choose agents' strategies $\boldsymbol{\sigma}'$. Any deviation from these strategies $\boldsymbol{\sigma}'$ makes the "decoding" of a single agent's action impossible. Hence, in the presence of strategic incentives (where $\boldsymbol{\sigma}$ is not determined by the designer), this reduction will be of little use.

One might wonder whether count is then strictly more powerful for the purposes of contextual privacy. Interestingly, the count elicitation technology also fails in cases of collective and individual pivotality.

Proposition 15. *Let ϕ be a social choice function, and consider a particular type profile $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. If for any subset $A \subseteq N$ of agents, and types $\theta'_i \in \Theta$ for agents $i \in A$, if*

$$\phi(\boldsymbol{\theta}) \neq \phi(\boldsymbol{\theta}'_A, \boldsymbol{\theta}_{-A}) \qquad \text{(collective pivotality)}$$

and for all $i \in A$

$$\phi(\boldsymbol{\theta}) = \phi(\theta'_i, \boldsymbol{\theta}_{-i}) \qquad \text{(no individual pivotality)}$$

then for any count elicitation protocol $P \in \mathcal{P}_{\mathfrak{S}_{\text{Count}}}$, there exists an agent $i \in A$ whose contextual privacy is violated at $\boldsymbol{\theta}$, i.e. $(i, \boldsymbol{\theta}) \in \Gamma(P, \phi)$.

Proof. Consider the first query v that distinguishes types in the set

$$\tilde{\boldsymbol{\Theta}} = \{\boldsymbol{\theta}_{-A}\} \times \bigtimes_{i \in A} \{\theta_i, \theta'_i\}.$$

This query must exist because of collective pivotality. By the definition of individual elicitation, there must be an agent i that sends different messages for θ_i and θ'_i, $\sigma_i(\theta, v) \neq \sigma_i(\theta', v)$. By the lack of individual pivotality, observe that $\phi(\theta_i, \boldsymbol{\theta}_{-i}) = \phi(\theta'_i, \boldsymbol{\theta}_{-i})$. By the definition of the count elicitation technology, $(\theta_i, \boldsymbol{\theta}_{-i})$ and $(\theta'_i, \boldsymbol{\theta}_{-i})$ are distinguished at v. □

As a consequence of Proposition 15, all of the results on collective and individual pivotality obtained in Section 3.2 also hold for the count technology, which can guide intuitions even for this

richer elicitation technology.

E Variations of Contextual Privacy

In this section, we consider two ways to strengthen our definition of contextual privacy violations. The two key pieces of the definition of contextual privacy violations are whether two types for an agent are distinguished and whether there is some difference to the outcome that would justify the distinction. The following two variations of contextual privacy violations consider stronger notions of distinction and justification, respectively.

We explore these stronger concepts for both theoretical and practical reasons. On the practical side, these extensions may map onto design goals in some settings. On the theoretical side, these criteria help to illuminate connections to other desiderata in mechanism design, and illustrate which of our results are robust to alternative formulations of contextual privacy.

E.1 Individual Contextual Privacy

The first extension, the *individual contextual privacy violation*, requires that if two types are distinguishable for agent i, these types must lead to different outcomes under ϕ *for agent* i. This definition thus only applies in domains where the outcome space X, specifies an allocation for each agent $i \in N$. Let $\phi_i(\boldsymbol{\theta})$ denote the outcome under ϕ for agent i.

Definition (Individual Contextual Privacy Violations). Let $(P, \boldsymbol{\sigma}) \in \mathcal{P}_{\mathfrak{S}}$ be a protocol for ϕ. Then the set of *individual contextual privacy violations* $\Gamma_{\text{ind.}}(P, \boldsymbol{\sigma}, \phi) \subseteq N \times \boldsymbol{\Theta}$ contains tuples $(i, \boldsymbol{\theta})$ for which there exists $\theta_i' \in \Theta$ such that

$$P \text{ distinguishes } (\theta_i, \boldsymbol{\theta}_{-i}) \text{ and } (\theta_i', \boldsymbol{\theta}_{-i}) \text{ yet } \phi_i(\theta_i, \boldsymbol{\theta}_{-i}) \neq \phi_i(\theta_i', \boldsymbol{\theta}_{-i}).$$

We call a choice rule *individually contextually private* for ϕ if there is a protocol $(P, \boldsymbol{\sigma})$ such that $\Gamma_{\text{ind.}}(P, \boldsymbol{\sigma}, \phi) = \emptyset$.

As for $\Gamma(P, \boldsymbol{\sigma}, \phi)$, we may also omit $\boldsymbol{\sigma}$ from our notation if the strategies are clear from context. Notice that individual contextual privacy is stronger than contextual privacy—any choice rule that is individually contextually private is also contextually private. If there were an agent i for whom contextual privacy were violated, then individually contextual privacy would automatically be violated.

As a normative criterion, individual contextual privacy requires that if the designer can distinguish between two types for agent i, then it *should* be the case that agent i's outcome is changed. This criterion captures a notion of legitimacy—agent i may view participation in the mechanism as involving an inherent tradeoff between information revelation and allocation. We can imagine a

speech from agent i along the following lines: "The designer can learn that I have type θ_i and not θ_i' as long as the designer's knowledge of this makes a difference to my allocation."

Individual contextual privacy is closely related to *non-bossiness*, introduced by Satterthwaite and Sonnenschein (1981). A choice rule ϕ is *non-bossy* if for all $\theta_i, \theta_i' \in \Theta, \boldsymbol{\theta}_{-i} \in \boldsymbol{\Theta}_{-i}$,

$$\phi_i(\theta_i, \boldsymbol{\theta}_{-i}) = \phi_i(\theta_i', \boldsymbol{\theta}_{-i}) \implies \phi(\theta_i, \boldsymbol{\theta}_{-i}) = \phi(\theta_i', \boldsymbol{\theta}_{-i}). \tag{15}$$

We define the set of *non-bossiness violations* $B(P, \phi) \subseteq N \times \boldsymbol{\Theta}$ consisting of $(i, \boldsymbol{\theta})$ where (15) fails to hold.

Non-bossiness says that if agent i changes her report from θ_i to θ_i' and her allocation is unchanged, then no other agent j's allocation changes either. The idea is that if agent i could unilaterally change her report and affect a change in some agent j's allocation without changing her own allocation, agent i would be "bossy."

We can characterize the set of individual contextual privacy violations $\Gamma_{\text{ind.}}$ in terms of contextual privacy violations and bossiness violations.

Proposition 16. *The set of individual contextual privacy violations is the intersection of contextual privacy violations and non-bossiness violations,* $\Gamma_{ind.}(P, \phi) = \Gamma(P, \phi) \cup B(P, \phi)$.

That is, the set of individual contextual privacy violations is the union of contextual privacy and non-bossiness violations.

Proof. We first show that $B(P, \phi) \subseteq \Gamma_{\text{ind.}}(P, \phi)$. Let $(i, \boldsymbol{\theta})$ be a non-bossiness violation. Hence, there exists a $j \in N \setminus \{i\}$ and type profiles $(\theta_i, \boldsymbol{\theta}_{-i})$, $(\theta_i', \boldsymbol{\theta}_{-i})$ such that $\phi_i(\theta_i, \boldsymbol{\theta}_{-i}) = \phi_i(\theta_i', \boldsymbol{\theta}_{-i})$ but $\phi_j(\theta_i, \boldsymbol{\theta}_{-i}) \neq \phi_j(\theta_i', \boldsymbol{\theta}_{-i})$. Because of the latter, the protocol P must distinguish $(\theta_i, \boldsymbol{\theta}_{-i})$ and $(\theta_i', \boldsymbol{\theta}_{-i})$. By the former property, this leads to an individual contextual privacy violation for agent i. By definition, $\Gamma(P, \phi) \subseteq \Gamma_{\text{ind.}}(P, \phi)$. Hence, $B(P, \phi) \cup \Gamma(P, \phi) \subseteq \Gamma_{\text{ind.}}(P, \phi)$.

To show the converse, we show that $(N \times \boldsymbol{\Theta}) \setminus (\Gamma(P, \phi) \cup B(P, \phi)) = ((N \times \boldsymbol{\Theta}) \setminus \Gamma(P, \phi)) \cap ((N \times \boldsymbol{\Theta}) \setminus B(P, \phi)) \subseteq (N \times \boldsymbol{\Theta}) \setminus \Gamma_{\text{ind.}}(P, \phi)$, which finishes the proof. In other words, we show that if at $(i, \boldsymbol{\theta})$ there is neither a contextual privacy nor a non-bossiness violation, there can't be an individual contextual privacy violation. Consider any $(\theta_i, \boldsymbol{\theta}_{-i})$ and $(\theta_i', \boldsymbol{\theta}_{-i})$ that P distinguishes. By contextual privacy, $\phi(\theta_i, \boldsymbol{\theta}_{-i}) \neq \phi(\theta_i', \boldsymbol{\theta}_{-i})$. By non-bossiness, $\phi_i(\theta_i, \boldsymbol{\theta}_{-i}) \neq \phi_i(\theta_i', \boldsymbol{\theta}_{-i})$ follows. Thus, P does not produce an individual contextual privacy violation at $(i, \boldsymbol{\theta})$. □

Given this relationship of individual contextual privacy and non-bossiness, we are able to leverage existing results to yield unique characterizations for the first-price auction and the serial dictatorship.

We say a mechanism is *neutral* if for all type profiles $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, and permutations π, $\phi(\pi\boldsymbol{\theta}) = \pi\phi(\boldsymbol{\theta})$.

Proposition 17. *An object assignment choice rule ϕ is individually contextually private, neutral and strategyproof if and only if it is a serial dictatorship.*

Proof. The serial dictatorship is contextually private and non-bossy, hence individually contextually private by Proposition 16. It also is strategyproof.

Conversely, if ϕ is individually contextually private, then it is also non-bossy by Proposition 16. By Svensson (1999), a mechanism is neutral, strategyproof and non-bossy only if it is a serial dictatorship. So, if ϕ is individually contextually private, neutral and strategyproof, then it is a serial dictatorship. □

Note here that efficiency is implied by neutrality. A characterization of efficient, strategyproof and non-bossy mechanisms is supplied by Pycia and Ünver (2017) and Bade (2020)—note that the class characterized in those works is larger than just serial dictatorships.

Similarly, we can draw on existing literature on non-bossiness to uniquely characterize the first-price auction.

Proposition 18. *The first-price auction is the unique individually contextually private, efficient, and individually rational auction rule.*

Proof. The protocol for the first-price auction rule presented in Appendix B.2 is individually contextually private. It is well known that the first-price auction is efficient and individually rational.

Let ϕ be any individually contextually private, efficient and individually rational auction rule. By the characterization of efficient, non-bossy and individually rational auctions from Pycia and Raghavan (2022, Theorem 1), this means that it is a protocol for the first-price auction. □

E.2 Group Contextual Privacy

Another extension is *group* contextual privacy violations, which strengthens the notion of distinction. Group contextual privacy requires that if a protocol distinguishes two type *profiles*, they must lead to different outcomes. This notion strengthens contextual privacy, which requires only that if a single agent's types are distinguishable, then they lead to different outcomes.

Definition (Group Contextual Privacy Violations). A protocol $P = (V, E)$ for a social choice function ϕ with strategies $\boldsymbol{\sigma}$ produces a *group contextual privacy violation* at $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ if there is a type $\boldsymbol{\theta}' \in \boldsymbol{\Theta}$ such that

$$P \text{ distinguishes } \boldsymbol{\theta} \text{ and } \boldsymbol{\theta}' \text{ yet } \phi(\boldsymbol{\theta}) = \phi(\boldsymbol{\theta}').$$

Notice again that this definition strengthens contextual privacy. Here, it is because it strengthens the underlying notion of distinguishability—two type profiles $\boldsymbol{\theta}, \boldsymbol{\theta}'$ are distinguishable if they belong to different terminal nodes. Regular contextual privacy's notion of distinguishability is on the agent-level—two types θ_i, θ_i' are distinguishable if they belong to different terminal nodes, *holding all other agent's types fixed* at $\boldsymbol{\theta}_{-i}$.

We characterize the set of group contextually private protocols next. Denote the set of outcomes reachable from node v by X_v.

Theorem 5. *A protocol $P = (V, E)$ is $\mathfrak{S}$-group contextually private if and only if for any query, $\bigcup_{w \in \text{children}(v)} X_w = X_v$ is a disjoint union.*

Proof. First assume that P is group contextually private, and assume for contradiction that v is a query such that $(v, w), (v, w') \in E$ and $X_w \cap X_{w'} \neq \emptyset$. Hence, there are $\boldsymbol{\theta} \in \boldsymbol{\Theta}_w$ and $\boldsymbol{\theta}' \in \boldsymbol{\Theta}_{w'}$ such that $\phi(\boldsymbol{\theta}) = \phi(\boldsymbol{\theta}')$, which contradicts group contextual privacy.

Next assume that reachable outcomes are disjoint at each query. Let $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ be distinguished at v. As outcomes are disjoint, it must be that $\phi(\boldsymbol{\theta}) \neq \phi(\boldsymbol{\theta}')$. Hence the protocol is group contextually private. □

This general characterization lends more practical insight when specialized to group contextual privacy under only individual elicitation protocols.

Corollary 1. *A social choice function is group contextually private under individual elicitation protocols if and only if it can be represented by a protocol in which, at every node, the agent's choice rules out a subset of the outcomes.*

This characterization, in particular, implies that the serial dictatorship and the first price auction are group contextually private. In the serial dictatorship protocol, whenever an agent is called to play, they obtain their favorite object among those that remain. So, their choice rules out the outcomes in which a different agent gets their favorite object that remains. In the first price auction, the agent agent renders a particular outcome impossible, namely the outcome under which they win the good at a particular price.

Under individual elicitation protocols, group contextual privacy is thus reminiscent of other extensive-form properties related to simplicity that the serial dictatorship satisfies. In particular, the serial dictatorship is obviously strategyproof Li (2017) and strongly obviously strategyproof Pycia and Troyan (2023).

Is there are containment relationship between rules that admit group contextually private protocols and those that admit an implementation in obviously dominant strategies? It turns out that the answer is no: there are mechanisms that are group contextually private and not obviously strategyproof, and vice versa. The ascending auction is obviously strategyproof Li (2017), but not group contextually private with respect to individual elicitation (it is a protocol for the second-price auction, which is not contextually private). A class of "non-clinching rules", on the other hand, are strategyproof and group contextually private, but fail to have an obviously strategyproof implementation. We next offer an example of a non-clinching rule, and show that it is group contextually private but not obviously strategyproof.

Example (Non-Clinching Rule). In particular, there are strategyproof choice rules that are not obviously strategyprouf but group-contextually private. As an example, consider $n = 2$, $\Theta = \{\underline{\theta}, \overline{\theta}\}$ and $X = \{x_1, x_2, x_3, x_4\}$. Assume that for agent 1,

$$x_1 \succ_{\underline{\theta}} x_3 \succ_{\underline{\theta}} x_2 \succ_{\underline{\theta}} x_4$$
$$x_1 \prec_{\underline{\theta}} x_3 \prec_{\underline{\theta}} x_2 \prec_{\underline{\theta}} x_4$$

and for agent 2

$$x_1 \succ_{\underline{\theta}} x_2 \succ_{\underline{\theta}} x_3 \succ_{\underline{\theta}} x_4$$
$$x_1 \prec_{\underline{\theta}} x_2 \prec_{\underline{\theta}} x_3 \prec_{\underline{\theta}} x_4.$$

Consider the social choice function

$$\phi(\underline{\theta}, \underline{\theta}) = x_1 \qquad \phi(\underline{\theta}, \overline{\theta}) = x_2 \qquad \phi(\overline{\theta}, \underline{\theta}) = x_3 \qquad \phi(\overline{\theta}, \overline{\theta}) = x_4.$$

As ϕ is injective, any protocol for ϕ is group contextually private. It is also tedious but straightforward to check that this rule is strategyproof. There is no obviously strategyproof implementation, however. Assume that agent 1 is asked to play first. They face a choice between outcomes $\{x_1, x_3\}$ and $\{x_2, x_4\}$, which, for both $\underline{\theta}$ and $\overline{\theta}$ types are are not ordered in the set order, and hence make no action obviously dominated. A similar observation for agent 2 shows that neither first action can be obviously dominant.