# Decentralized Voting with Unconditional Privacy

Felix Brandt
Computer Science Department
Stanford University
Stanford CA 94305
brandtf@cs.stanford.edu

Tuomas Sandholm
Computer Science Department
Carnegie Mellon University
Pittsburgh PA 15213
sandholm@cs.cmu.edu

## ABSTRACT

The aggregation of conflicting preferences is a key issue in multiagent systems. Due to its universality, voting has a central role among preference aggregation mechanisms. Voting among a set of alternatives can be used for such diverse tasks as choosing a joint plan in a multiagent system, determining a leader in a group of humans or agents, or voting among different resource or task allocations. Maintaining privacy of individuals' votes is crucial in order to guarantee freedom of choice (*e.g.*, lack of vote coercing and reputation effects), and not facilitate strategic voting. We investigate whether *unconditional full privacy* can be achieved in voting, that is, privacy that relies neither on trusted third parties (or on a certain fraction of the voters being trusted), nor on computational intractability assumptions (such as the hardness of factoring). In particular, we study the existence of distributed protocols that allow voters to jointly determine the outcome of an election without revealing any information but the election outcome. We show the impossibility of reaching unconditional full privacy for a variety of the most common voting schemes ranging from simple veto voting to the single transferable vote scheme. On the positive side, we propose several distributed protocols that privately compute the outcome of common voting schemes while only revealing a limited amount of information.

## Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*; E.4 [**Data**]: Coding and Information Theory

## General Terms

Economics, Security, Theory

## Keywords

Voting Protocols, Multiparty Computation


*AAMAS'05,* July 25-29, 2005, Utrecht, Netherlands.


## 1. INTRODUCTION

AI and voting theory are research fields that mutually benefit each other. On the one hand, results from voting theory have been used in AI, for example in multiagent planning (*e.g.*, [15]) and collaborative filtering (*e.g.*, [25]). On the other hand, AI research has made contributions to voting theory (*e.g.*, [11, 10]). Voting among a set of alternatives can be used for such diverse tasks as choosing a joint plan in a multiagent system, determining a leader in a group of humans or agents, or voting among different resource or task allocations. Two seminal impossibility results in voting theory [1] [16, 27] show that there is no voting scheme that satisfies even a modest set of desiderata in general.[1] This explains why a wide variety of voting schemes with differing advantages and disadvantages have evolved.

Maintaining privacy of individuals' votes is crucial in voting. For one, this is required to achieve freedom of choice: avoiding vote coercing, allowing a voter to vote for a casino over a school without fear of adverse reputation effects, *etc.* Second, learning about others' votes opens the possibility for a voter to benefit from voting insincerely—and according to one of the seminal impossibility theorems [16, 27], *all* voting schemes (except dictatorial ones) are manipulable in this sense, as long as there are more than two candidates. In other words, uncertainty about other agents' preferences is a critical requirement for voting schemes to operate as desired and therefore needs to be protected appropriately. Consulting a trusted third party is a straightforward but very weak way of obtaining privacy. It is virtually impossible to prevent the third party from revealing sensitive information to voters, candidates, or other agents.

This paper investigates whether *unconditional full privacy* can be achieved in voting, that is, privacy that relies neither on trusted third parties (or on a certain fraction of the voters being trusted), nor on computational intractability assumptions (such as the hardness of factoring). We study the existence of distributed protocols that allow voters to jointly determine the outcome of an election by exchanging messages without revealing any information but the outcome. In the rest of this paper, this is called *emulation* of a voting scheme. Our setting consists of $n$ agents that vote among $m$ candidates using common voting schemes (for convenience we assume that $n > 2$ and $m \leq n$). We derive several impossibility and possibility results in this setting.

[1] In certain special cases, the desiderata can be obtained. For example, if there are only two candidates, majority voting works.

One dimension along which privacy guarantees differ is how many of the agents need to collude before privacy can be breached. In this paper, we will require the strongest variant, *full privacy* or so-called $(n-1)$-privacy, which means that no information (beyond what can be inferred from the outcome) can be uncovered by a coalition that does not include *all* of the agents.

Another criterion along which privacy guarantees differ is if and how the computational power of the adversary is limited. In fact, using computational intractability as a barrier against undesirable behavior has a long tradition in modern cryptography since Diffie and Hellman's seminal paper [14]. When relying on the existence of so-called "trapdoor one-way permutations", it has been shown that arbitrary functions can be jointly computed so that no private input can be revealed by a polynomially-bounded adversary [18]. Unfortunately, computational intractability not only relies on the unproven assumption $\mathcal{P} \neq \mathcal{NP}$ but also on the widely unknown field of average-case complexity and further, more specific assumptions. Moreover, even when these conjectures are true, it may be possible to breach privacy in the future when sufficient computational power becomes available. In this paper, we will study the strongest privacy variant along this dimension, *unconditional privacy* (aka. *non-cryptographic* or *information-theoretic* privacy), where the adversary's computational power is unlimited and a complete network of private channels between agents is given. It is known that only a restricted class of functions can be computed fully privately in this model.[2] Section 4 contains some known results about this class of functions. In the rest of the paper, when we say *privacy*, we mean unconditional full privacy.

In order to simplify the presentation, we assume that the adversary is *passive*, that is, agents do not deviate from the prescribed protocol. There are standard cryptographic techniques (for example, perfect zero-knowledge arguments) that force active adversaries to act according to a protocol (see *e.g.*, [17]).[3] However, using these techniques will incur massive overhead. Less secure but potentially more efficient methods have recently been suggested, for example, redundancy, policing, and careful partitioning of the problem across agents [24, 23]. After all, *negative* results in the passive adversary model also hold in a model that allows active adversaries.

The remainder of this paper is structured as follows. In Section 2, we review related research. Descriptions of the voting schemes to be studied are given in Section 3. Section 4 contains fundamental theoretical results that we will leverage in our proofs. In Section 5, we propose impossibility results whereas in Section 6 we propose constructive possibility results. The paper concludes with an overview of the obtained results in Section 7.

[2]When assuming that a majority of the agents is trustworthy (recall that this is not *full* privacy), *all* functions can be jointly computed in the unconditional model [2, 7] (assuming passive adversaries).

[3]Using perfect zero-knowledge arguments to prevent manipulation requires a careful definition of the adversary. Loosely speaking, one would assume that the adversary is incapable of performing super-polynomial computations *during* the protocol (which is somewhat reasonable given the typically short execution time). Once the protocol is finished, the adversary might take as much time (and computational power) as he wants in order to try to breach privacy.

## 2. RELATED RESEARCH

There is a large body of cryptographic voting protocol research (*e.g.*, [6, 26, 13, 19]), out of which some proposed techniques provide unconditional privacy (*e.g.*, [6, 26]). However, with the notable exception of Kiayias et al's recent contribution [19], all of these approaches rely on a number of trusted third parties where privacy is based on the assumption that these third parties do not collude.

There is a conceptual difference between this paper and existing cryptographic work. Whereas the latter focuses on technical aspects of how to obtain *anonymity* in the *plurality* scheme, we investigate the existence of voter-distributed protocols that reveal as little information as possible (preferably *only* the election winner). Furthermore, we not only consider plurality but also a variety of other common voting schemes.

We recently generalized the impossibility result of Theorem 3 to arbitrary social choice functions (and social welfare functionals) that are non-dictatorial, Pareto-optimal, and monotonic [5]. Regarding sealed-bid auctions, it has been shown that the outcome of first-price auctions can be computed unconditionally fully privately by bidders whereas this is impossible for second-price auctions [4]. In a remotely related paper, the communication complexity of the common voting schemes was investigated (without privacy considerations) [12].

## 3. COMMON VOTING SCHEMES

In this section, we review the most common voting schemes. These are the schemes we will study in the remainder of the paper. The first scheme we consider differs from the following ones in that it only decides on the acceptance of a single candidate (or issue) over the *status quo*.

**Veto voting (aka. unanimity voting)** Each voter is only allowed to express his agreement or refusal. If at least one voter disagrees, the candidate/issue at hand is rejected. Otherwise, it is accepted.

The following schemes select one out of $m$ candidates. The candidates are not necessarily agents. A candidate can be any abstract object, *e.g.*, a plan, parameter, task assignment, or schedule. The first four schemes will be called *score-based* because the decision is based on the accumulated scores that voters assign to the candidates.

**Plurality** Each voter votes for his most preferred candidate. The candidate with the highest score wins.

**Rejection**[4] Each voter states his least preferred candidate. The candidate with the lowest number of votes wins.

**Borda** Each voter gives $m-1$ points to his most preferred candidate, $m-2$ to his second choice, ..., and 0 to his last. The candidate with the highest score wins.

**Approval** Each voter gives each candidate he likes a single point (everything from giving 0 to $m$ points in total is feasible). The candidate with the highest score wins.

In the following voting schemes, each voter submits a complete list of candidates in the order of his preference. We call these schemes *order-based*.

[4]This scheme is sometimes also called veto voting.

**Copeland** For each candidate $j$, candidate $i$ gets a point if there are more voters who prefer $i$ over $j$. He is deducted a point if more voters prefer $j$ over $i$. The candidate with the highest score wins.

**Maximin (aka. Kramer-Simpson)** A candidate's score is the lowest number of voters that prefer him over any other candidate, *i.e.*, his worst performance in any pairwise comparison. The candidate with the highest score wins.

**Cup** The cup is defined by a balanced binary tree with one leaf per candidate. Each non-leaf node is assigned to the candidate that more voters prefer among the node's children. The candidate assigned to the root wins.

**Single transferable vote (STV, aka. instant runoff)** Winner determination proceeds in rounds. In each round, a candidate's score is the number of voters that rank him highest among the remaining candidates, and the candidate with the lowest score drops out. The last remaining candidate wins. (A vote "transfers" from its top remaining candidate to the next highest remaining candidate when the former drops out.)

# 4. PRELIMINARIES

In this section we describe the underlying communication model and review some key results which we will use as building blocks in our proofs.

The outcome function $f(\cdot)$ of the election is jointly computed by agents using a distributed, randomized[5] protocol consisting of several rounds. In order to enable secure message exchange, we make the standard assumption of a complete synchronous network of private channels between agents. In each round, each agent may send a message to any other agent. Each message an agent sends is a function of his preferences, his independent random input $r_i$, the messages he received so far, and the recipient. When the protocol is finished, all agents know the value of $f(\cdot)$. As usual, full privacy in the context of information-theoretic function evaluation is defined as follows: A distributed protocol for computing $f(x_1, x_2, \ldots, x_n) = a$ is unconditionally fully private if any coalition of agents is incapable of uncovering any information besides what can be inferred from $a$ and the coalition's preferences.[6] More formally:

DEFINITION 1 (PRIVACY). *For any $T \subseteq \{1, 2, \ldots, n\}$ and every two input vectors $\vec{x}, \vec{y} \in X^n$ satisfying $\forall i \in T : x_i = y_i$ and $f(\vec{x}) = f(\vec{y})$, and for every choice of random inputs $\{r_i\}_{i \in T}$, the messages seen by agents belonging to $T$ in both cases are identically distributed. Let* VIEW$_T$ *be a function that, given the vector of individual inputs and random values, yields the concatenation of all (prefix-free) messages exchanged between members of $T$ and $\bar{T} = \{1, 2, \ldots, n\} \setminus T$. A protocol for computing $f(\cdot)$ is* private *if*

$$\langle \text{VIEW}_T(\vec{x}, \{r_i\}_{i \in T}) \rangle = \langle \text{VIEW}_T(\vec{y}, \{r_i\}_{i \in T}) \rangle$$

*where $\langle \ldots \rangle$ denotes the probability distribution of the inner term with the probability taken over $\{r_i\}_{i \in \bar{T}}$.*

A complete characterization of all privately computable *Boolean* functions has been given by Chor and Kushilevitz [9].

THEOREM 1. *A* Boolean *function is privately computable if and only if it is of the form $f(x_1, x_2, \ldots, x_n) = B_1(x_1) \oplus B_2(x_2) \oplus \cdots \oplus B_n(x_n)$, where $B_i(x_i)$ are Boolean predicates and $\oplus$ is the Boolean exclusive-or operator.*

Such a complete characterization for general (non-Boolean) functions is not yet known (except for only two agents [21]). However, there are necessary conditions for the private computability of a function [9].

LEMMA 1 (CORNERS LEMMA). *Let $f : X \times Y \to Z$ be a privately computable 2-ary function. For every $x_1, x_2 \in X$ and $y_1, y_2 \in Y$, if $f(x_1, y_1) = f(x_1, y_2) = f(x_2, y_1) = a$, then $f(x_2, y_2) = a$.*

LEMMA 2 (PARTITION LEMMA). *Let $f : X_1 \times X_2 \times \cdots \times X_n \to Z$ be a privately computable $n$-ary function. Then, for each $i \in \{1, 2, \ldots, n\}$ the 2-ary function $f_2(x_i, (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, x_{i+2}, \ldots, x_n)) \stackrel{def}{=} f(x_1, x_2, \ldots, x_n)$ is privately computable.*[7]

By combining Lemma 1 and Lemma 2, we can obtain a necessary condition for the possibility of privately computing an $n$-ary function. This can be used to prove that an $n$-ary function is *not* privately computable (as in Theorem 3). Due to the lack of a more detailed characterization of $n$-ary privately computable functions, the only way to show that a function *is* privately computable is to give a concrete protocol that fulfills this task (as in Theorems 4, 5, 6, and 7). As first observed by Benaloh, there is a simple protocol to privately compute modular sums [3].

LEMMA 3. *$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} x_i \mod p$ is privately computable.*

PROOF. Each agent $i$ chooses $n$ random values $x_{ij} \in \mathbb{Z}_p$ so that the modular sum $\sum_{j=1}^{n} x_{ij} \mod p = x_i$. He then sends each addend $x_{ij}$ to agent $j$ and keeps $x_{ii}$. After all agents have done this, each agent $i$ publishes $s_i = \sum_{j=1}^{n} x_{ji} \mod p$, *i.e.*, the modular sum of his remaining $x_{ii}$ and the $n-1$ addends he received. $f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} s_i \mod p$ can be computed by each participant. □

# 5. IMPOSSIBILITY RESULTS

We are now ready to present our results. In this section we present negative results whereas in the next section we present positive ones (by allowing the revelation of additional information besides the election outcome). We assume that each voter $i$ possesses a complete ranking of the

[5]Indeed, randomization is *necessary* in order to privately compute any non-degenerate function [17].

[6]Excluding the information that follows from $a$ and the coalition's preferences is essential for the definition to be non-trivial. For example, when privately computing the sum of input values, a coalition of $n-1$ agents can always infer the remaining agent's input $x_n = a - \sum_{i=1}^{n-1} x_i$ no matter which protocol is used. However, this does *not* rule out the existence of a private protocol for computing $f(\cdot)$ according to Definition 1. As a matter of fact, such a protocol exists (see Lemma 3).

[7]This is a special case of the Partition Lemma for $t = n-1$ according to the definition by Chor et al [8].

candidates in the order of his preference. Based on this ranking and the specific voting scheme, he enters some private input value $x_i$ (*e.g.*, his most preferred candidate) into the joint computation of the outcome function. The protocols in this section must not reveal more information than the winning candidate's identity. In other words, $f(\cdot)$'s range is $Z = \{1, 2, \ldots, m\}$. This is justified by the fact that whenever something beyond the election winner is revealed, that information could be used by a coalition of less than all the agents to breach a noncolluder's privacy. For example, if a veto protocol reveals the total number of vetoes, a coalition of all vetoers can easily derive the votes of all remaining agents. In the case of ties, we deliberately leave the outcome undefined. As a consequence, the impossibility results of this paper hold regardless of what is done in the case of a tie: picking the winner at random, using priorities, or even revealing the identities of tied candidates.

It might seem unlikely that *any* relevant function can be computed at all in our extremely rigorous privacy setting. This is not entirely true as for example the outcome of first-price sealed-bid auctions [4] or the arithmetic mean[8] can be computed privately. Nevertheless, when it comes to voting, it turns out that all schemes listed in Section 3 can *not* be emulated by private protocols.

THEOREM 2. *There is no private protocol for veto voting.*

PROOF. Because the veto protocol just yields a Boolean outcome (veto or not), we can apply Theorem 1. The veto outcome function is $f_v(x_1, x_2, \ldots, x_n) = x_1 \vee x_2 \vee \cdots \vee x_n$ and because this (inclusive) disjunction cannot be expressed as an exclusive disjunction of Booleans of its parts, it immediately follows from Theorem 1 that it cannot be computed privately. □

THEOREM 3. *There are no private protocols for plurality, rejection, Borda, approval, Copeland, maximin, cup, and STV voting.*

PROOF. The outcome function, $f_p$, of the plurality voting scheme is

$$f_p(x_1, x_2, \ldots, x_n) = \arg\max_{j=1}^{m}\Big\{\big|\{i \mid (1 \le i \le n) \wedge (x_i = j)\}\big|\Big\}.$$

Essentially, the impossibility of fully privately emulating a voting scheme can be shown by finding an "embedded OR" of combinations of votes for any number of voters $n$ and candidates $m$: Let $\vec{x}$ and $\vec{y}$ be vectors of $n-1$ votes and $x$ and $y$ single votes. If $(\vec{x}, x)$, $(\vec{x}, y)$, and $(\vec{y}, x)$ all yield candidate $a$ as the winner, then $(\vec{y}, y)$ has to yield $a$ as well. In order to obtain the most general impossibility, the following counter-examples are designed for a setting with just two candidates. They can trivially be extended to any number of candidates by assuming that nobody votes for the additional candidates.

[8]In multiagent systems, this can be used to decide on the setting of a global parameter via "average voting". In the average voting scheme, voters' preferences are numbers in a given interval and the outcome is the arithmetic mean of these numbers. Private computability follows from Lemma 3.

CASE 1 ($n \mod 2 = 1$): Let

$$\begin{aligned}\vec{x} &= (\underbrace{1, \ldots, 1}_{n-1}),\\ \vec{y} &= (\underbrace{1, 2, 1, 2, \ldots, 1, 2}_{n-1}),\\ x &= 1, \text{ and } y = 2.\end{aligned}$$

Then

$$f_p(\vec{x}, x) = f_p(\vec{x}, y) = f_p(\vec{y}, x) = 1,$$

but $f_p(\vec{y}, y) = 2$.

CASE 2 ($n \mod 2 = 0$): Let

$$\begin{aligned}\vec{x} &= (\underbrace{1, 2, 1, 2, \ldots, 1, 2}_{n-4}, 1, 1, 1),\\ \vec{y} &= (\underbrace{1, 2, 1, 2, \ldots, 1, 2}_{n-4}, 1, 1, 2),\\ x &= 1, \text{ and } y = 2.\end{aligned}$$

Then

$$f_p(\vec{x}, x) = f_p(\vec{x}, y) = f_p(\vec{y}, x) = 1.$$

$f_p(\vec{y}, y)$ results in a tie and thus is undefined. However, any other function value than 1 (including special "tie output symbols") will yield an embedded OR. For this reason, $f_p(\vec{y}, y)$ has to be set to 1. This yields an embedded OR at a different position. Let $\vec{p} = (\underbrace{1, 2, 1, 2, \ldots, 1, 2}_{n-4})$, then

$$f_p(\vec{p}, 2, 1, 2, 2) = f_p(\vec{p}, 2, 2, 2, 2) = f_p(\vec{p}, 1, 2, 2, 2) = 2,$$

but $f_p(\vec{p}, 1, 1, 2, 2) = f_p(\vec{y}, y) = 1$.

This proves the impossibility of computing $f_p$ privately.

Instead of constructing similar vote configurations that yield embedded ORs in all remaining schemes, we employ May's Theorem [22]. May's Theorem says that, if there are just two candidates, plurality voting is the *only* voting scheme that is neutral (no candidate is favored by the scheme), symmetric (all votes are treated equally), and monotonic (voting for a candidate cannot make him lose). Since all suggested schemes (except veto voting which is not neutral) satisfy these criteria, this implies that for two candidates all schemes collapse to plurality voting (which is then called "majority rule"). If we furthermore observe that all suggested schemes are Pareto-optimal (*i.e.*, if *all* voters prefer candidate $a$ over $b$, than $b$ cannot win the election), we can easily find configurations that yield embedded ORs: Any configuration of votes where each agent votes according to a preference ranking where candidates 1 and 2 are ranked as in the counter-examples for plurality voting and all remaining candidates are ranked in the same fixed order below 1 and 2, *e.g.*, $3, 4, \ldots, m$, will yield an embedded OR.

The only scheme for which this argumentation is flawed is rejection voting. Due to the very restricted expressiveness of votes in rejection voting (voters may only state their least preferred candidate), any candidate who is not ranked *last* in the configuration given above may be chosen. In other words, which candidate will be chosen entirely depends on the tie-breaking policy. Nevertheless, the impossibility of

privately emulating rejection voting for any number of candidates, regardless of tie-breaking, can be shown by the following construction. The outcome function, $f_r$, of the rejection voting scheme is

$$f_r(x_1, x_2, \ldots, x_n) = \arg\min_{j=1}^{m}\Big\{\big|\{i \mid (1 \le i \le n) \wedge (x_i = j)\}\big|\Big\}.$$

Since rejection voting is equivalent to plurality voting for two candidates, it suffices to consider the case $m > 2$: Let

$$\begin{aligned}
\vec{x} &= (\underbrace{1, \ldots, 1}_{n-m+1}, 2, 4, 5, \ldots, m),\\
\vec{y} &= (\underbrace{1, \ldots, 1}_{n-m+1}, 1, 4, 5, \ldots, m),\\
\vec{z} &= (\underbrace{1, \ldots, 1}_{n-m+1}, 3, 4, 5, \ldots, m),\\
x &= 2,\ y = 1, \text{ and } z = 3.
\end{aligned}$$

If $m > 3$, vectors $\vec{x}$, $\vec{y}$, and $\vec{z}$ are filled up with candidates 4, 5, etc. to ensure that always candidate 2 or 3 is chosen. This is unnecessary when there are just three candidates.

It turns out that

$$f_r(\vec{x}, x) = f_r(\vec{x}, y) = f_r(\vec{y}, x) = 3,$$

and $(\vec{y}, y)$ results in a tie, but has to be set to 3 due to the Corners Lemma: $f_r(\vec{y}, y) = 3$. However,

$$f_r(\vec{z}, z) = f_r(\vec{z}, y) = f_r(\vec{y}, z) = 2$$

requires $f_r(\vec{y}, y) = 2$ (again due to the Corners Lemma) which is a contradiction (see Table 1).

| $f_r$ | 1 | 2 | 3 | ... | m |
|---|---|---|---|---|---|
| $1, \ldots, 1, 1, 4, 5, \ldots m$ | ? | 3 | 2 | | |
| $1, \ldots, 1, 2, 4, 5, \ldots m$ | 3 | 3 | | | |
| $1, \ldots, 1, 3, 4, 5, \ldots m$ | 2 | | 2 | | |

**Table 1: Rejection voting ($m > 2$)**

□

We have recently extended this impossibility to any social choice function (and social welfare functional) that is non-dictatorial, Pareto-optimal, and monotonic (regardless of neutrality and symmetry) [5].

# 6. POSSIBILITY RESULTS

It is natural to ask which kind of privacy relaxations enable the private distributed emulation of a voting scheme. Modifying the outcome function to reveal more than just the minimum information makes it more likely that a scheme can be privately evaluated. The veto scheme outcome, for example, can be computed while only revealing the *number* of vetoers or without revealing any information to *non-vetoers*:

THEOREM 4.

*(i) There is a private veto protocol that only reveals the total number of vetoes.*

*(ii) There is a private probabilistic veto protocol that only reveals to a vetoer whether he is the only vetoer (and no information to non-vetoers).*

PROOF.

(i) Such a protocol can be designed easily by computing the sum (see Lemma 3) of individual inputs that are either 0 (no veto) or 1 (veto). Of course, $p$ (the finite group's size) must be greater than $n$ (the maximum number of vetoes) in order to avoid "overflows".

(ii) A similar protocol in which agents also compute the sum $f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} x_i$ of their inputs, but where $x_i$ is an arbitrarily chosen random number in $\mathbb{Z}_p$ when agent $i$ vetoes and 0 otherwise, has some interesting properties. For appropriately large $p$, $f(\cdot) = 0$ if and only if nobody vetoed, with exponentially small error probability. Agents that did not veto and outsiders do not learn any additional information besides the outcome (veto or not). Vetoers can see if somebody else vetoed by comparing $f(\cdot)$ and $x_i$. If $f(\cdot) \neq x_i \neq 0$, then there must have been another vetoer. However, the total number of vetoers remains unknown.

□

## 6.1 Anonymity

One of the lowest levels of privacy is that the outcome function is anonymous (we only deal with symmetric voting in this paper). Loosely speaking, a voting protocol is *anonymous* if the exchange of ballots from any pair of agents does not lead to different information to be revealed during the protocol. For a formal definition, we restrict the equality of distributions in Definition 1 to the case of permuted input vectors.

DEFINITION 2 (ANONYMITY). *Let $T$, $\vec{x}$, $\vec{y}$, and* $\text{VIEW}_T$ *be defined as in Definition 1 and furthermore assume that $\vec{x}$ is a permutation of $\vec{y}$ (in addition to $\forall i \in T : x_i = y_i$ and $f(\vec{x}) = f(\vec{y})$). A symmetric function $f(x_1, x_2, \ldots, x_n)$ can be computed* anonymously *if*

$$\langle \text{VIEW}_T(\vec{x}, \{r_i\}_{i \in T}) \rangle = \langle \text{VIEW}_T(\vec{y}, \{r_i\}_{i \in T}) \rangle.$$

Anonymity can be achieved in all voting schemes under consideration by using the protocol proposed in Lemma 3 to privately add numbers (including veto, see Theorem 4).

THEOREM 5. *There are anonymous protocols for all voting schemes considered in this paper. All protocols, except STV, require a constant number of rounds and polynomial communication resources.*

PROOF. Let us first consider score-based voting schemes: plurality, rejection, Borda, and approval voting. Every bidder constructs a vote-vector $\vec{v}_i = (v_{i1}, v_{i2}, \ldots, v_{im})$ where $v_{ij}$ denotes the number of points voter $i$ is willing to give candidate $j$. Each voter must distribute his points according to the rules of the underlying voting scheme, *i.e.*,

- Plurality/Rejection: $(\forall j : v_{ij} \in \{0,1\}) \wedge \left( \sum_{j=1}^{m} v_{ij} = 1 \right)$
- Approval: $\forall j : v_{ij} \in \{0,1\}$
- Borda: $\exists j : v_{ij} = m-1$, $\exists j : v_{ij} = m-2$, ..., $\exists j : v_{ij} = 0$

Voters can prove the correctness of their vote vectors using perfect zero-knowledge arguments (see Footnote 3).

In the following, voters jointly compute $\vec{s} = \sum_{i=1}^{n} \vec{v}_i$ using the protocol specified in Lemma 3 (with $p$ set to a sufficiently large number to avoid "overflows"). The outcome function $f(\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n) = \vec{s}$ is anonymous due to the commutativity of addition and the winner can easily be determined by looking for the greatest component.

In order-based schemes Copeland, maximin, and cup each bidder submits a $m \times m$ matrix[9] $\mathsf{V}_i$ in which element $\mathsf{V}_{ijk}$ is 1 if voter $i$ prefers candidate $j$ over $k$, $-1$ if he prefers $k$ over $j$, and 0 if he is indifferent. As mentioned before, in this paper, we assume that all agents honestly follow a prescribed protocol. In practical applications, voters not only should prove the correctness of the matrix in zero-knowledge, but also that it corresponds to a *non-cyclic* preference ordering. This may be a costly task. Voters then jointly compute $\mathsf{S} = \sum_{i=1}^{n} \mathsf{V}_i$. $f(\mathsf{V}_1, \mathsf{V}_2, \ldots, \mathsf{V}_n) = \mathsf{S}$ is an anonymous outcome function according to Definition 2. It can be easily verified that the outcome of all three voting schemes can be inferred from $\mathsf{S}$. An anonymous *multi-round* STV protocol consists of consecutive "rejection" executions that yield direct scores with more and more candidates removed. A *constant-round* STV protocol could be enabled by making each voter submit a vector of length $m!$ (the number of possible preference orderings) which results in exponential communication complexity. □

## 6.2 Partial Privacy

Now, can privacy beyond anonymity be obtained in the common voting schemes? To answer this, we define the notion of partial privacy. A voting protocol is partially private if it is anonymous and there is a configuration of votes in which the modification of a *single* vote does not affect the outcome.[10]

DEFINITION 3 (PARTIAL PRIVACY). *Let $T$ and $\text{VIEW}_T$ be defined as in Definition 1. A symmetric function $f(x_1, x_2, \ldots, x_n)$ can be computed* partially privately *if it is anonymous and there are two input vectors $\vec{x}, \vec{y} \in X^n$ so that $\exists j \in \bar{T} : x_j \neq y_j$, $\forall i \neq j : x_i = y_i$, $f(\vec{x}) = f(\vec{y})$, and*

$$\langle \text{VIEW}_T(\vec{x}, \{r_i\}_{i \in T}) \rangle = \langle \text{VIEW}_T(\vec{y}, \{r_i\}_{i \in T}) \rangle.$$

It turns out that in most schemes (rejection and cup being the exceptions) the incremental revelation of information enables partial privacy. As a consequence, a higher degree of privacy can be obtained at the cost of round complexity (see Table 2).

[9]Technically, a half-matrix suffices as all $\mathsf{V}_i$ are antisymmetric.

[10]Is is essential for the definition to only allow *one* vote to change. Otherwise, most protocols presented in Theorem 5 would be partially private already because the sums of scores (or pairwise comparisons) can be identical for different vote configurations.

THEOREM 6. *There are partially private protocols for plurality, Borda, approval, Copeland, maximin, and STV.*

PROOF. In some voting schemes, the components of the resulting vector $\vec{s}$ of candidates' scores can be revealed one after another in random order so that it it is unnecessary to reveal *all* scores (in expectation). *E.g.*, for plurality voting, scores can be revealed in random order until $\exists a \in I : \; s_a > n - \sum_{i \in I} s_i$ where $I$ is the set of candidates whose score has been revealed so far. Such a protocol is partially private because it reveals exactly the same information no matter who the remaining voters voted for. Similar criteria can be found for the Borda and approval scheme. Perhaps surprisingly, in the rejection scheme, the scores of *all* candidates have to be revealed in order to determine the outcome. The reason is that, even if there are just two scores left to be opened, it is always possible that one of them is zero.

It can easily be seen that the Copeland protocol also allows for criteria that make it unnecessary to open all matrix components (*e.g.*, when a candidate accumulated $m-1$ points). In the maximin protocol, a technique similar to *alpha-beta-pruning* [20] can be applied: Candidate $i$'s score $s_i$ is obtained by incrementally revealing components $\mathsf{S}_{i1}$, $\mathsf{S}_{i2}, \ldots, \mathsf{S}_{im}$ and then computing $s_i = \min_j\{\mathsf{S}_{ij}\}$. Whenever $\mathsf{S}_{ij} < s_k$ for some other candidate $k$, candidate $i$ can be discarded. Furthermore, some structural properties of $\mathsf{S}$ enable more pruning in both the maximin and the Copeland protocol.[11] Interestingly, there seems to be no pruning method for the cup protocol. All matrix elements (relevant to the pre-determined) cup need to be revealed which suggests that the cup protocol can only provide anonymity but not partial privacy (although we do not have enough evidence to claim that there *exists* no partially private protocol for the cup or the rejection scheme). Even though the STV protocol consists of consecutive iterations of the rejection protocol (which does not provide partial privacy), not all information has to be revealed. Since each remaining candidate's score is increasing from round to round (due to rejected candidates), protocol execution can terminate early whenever a candidate accumulated more points than the sum of the other candidates' scores. □

In many cases, the candidates are agents themselves, possibly even the voters themselves (*e.g.*, when determining a leader among a set of agents). In this case, a technique reminiscent of the Dutch (*i.e.*, descending) auction can be used to obtain a high degree of privacy in score-based schemes.

THEOREM 7. *Consider a voting setting where the candidates are agents who can take part in the distributed protocol (they need not be the voters, although they can be). There are private protocols for plurality, rejection, Borda, and approval voting in which the winner's score is revealed to everyone, each candidate learns his own score, and no other information is revealed.*

PROOF. The following protocols consist of two phases. In the first phase voters compute the score of each candidate, and in the second phase candidates compute the maximum score and corresponding candidate.

[11]*E.g.*, transitivity: if all voters prefer candidate $i$ over $j$, and half of them prefer $j$ over $k$, then at least half of the voters prefer $i$ over $k$.

| Protocol | Partial Privacy | Round Complexity |
|---|---|---|
| Plurality | ✓ | $\mathcal{O}(m)$ |
| Rejection | – | $\mathcal{O}(1)$ |
| Borda | ✓ | $\mathcal{O}(m)$ |
| Approval | ✓ | $\mathcal{O}(m)$ |
| Copeland | ✓ | $\mathcal{O}(m^2)$ |
| Maximin | ✓ | $\mathcal{O}(m^2)$ |
| Cup | – | $\mathcal{O}(1)$ |
| STV | ✓ | $\mathcal{O}(m^2)$ |
| Plurality | ✓ | $\mathcal{O}(n)$ |
| Rejection | ✓ | $\mathcal{O}(n)$ |
| Borda | ✓ | $\mathcal{O}(m \cdot n)$ |
| Approval | ✓ | $\mathcal{O}(n)$ |

**Table 2: Proposed anonymous voting protocols**

First, all voters compute the sum of their vote vectors (see Theorem 5) according to Lemma 3. However, the intermediate sums $s_i$ (see Lemma 3) are not published but privately sent to the corresponding candidate, so that only each candidate learns his score. In the following, the candidates engage in a protocol that determines the maximum score similar to the Dutch auction protocol.

1. $j = n$ (plurality and approval),
   $j = 0$ (rejection), or
   $j = (m-1) \cdot n$ (Borda), respectively

2. Each candidate $i$ broadcasts 1 if his score $s_i = j$, or 0 otherwise.

3. If all agents broadcasted 0,
   set $j = j - 1$ (plurality, approval, and Borda),
   or $j = j + 1$ (rejection), respectively,
   and proceed to step 2. Otherwise, the candidate who submitted 1 is the election winner.

This only reveals the winning candidate's score to the public. As mentioned before, we do not consider ties here. The round complexity of the proposed protocols is shown in the lower part of Table 2. As mentioned in Footnote 3, perfect zero-knowledge arguments can be used to ensure that voters follow the protocol truthfully and do not manipulate, for example by wrongfully broadcasting their identity in step 2. □

A different recent idea for achieving partial (unconditional) privacy is that of using an elicitor that incrementally asks questions from the voters about their preferences on an as-needed basis until the elicitor has enough information to determine the winning candidate (*e.g.*, [10]). However, unlike the protocols proposed in this paper, that method cannot guarantee anonymity because the elicitor knows from which agent each answer comes.

# 7. CONCLUSIONS

Voting among a set of alternatives can be used for such diverse tasks as choosing a joint plan in a multiagent system, determining a leader in a group of humans or agents, or voting among differing resource or task allocations.

This paper investigated whether unconditional full privacy can be achieved in voting, that is, privacy that relies neither on trusted third parties (or on a certain fraction of the voters being trusted), nor on computational intractability assumptions (such as the hardness of factoring). In particular, we studied the existence of distributed protocols that allow a group of voters to jointly determine the outcome of an election while revealing as little information as possible. We derived several impossibility and possibility results in this scenario for the most common voting schemes:

- None of the voting schemes under study can be emulated privately without revealing more information than just the winning candidate, even when there are just two candidates.[12]
- The veto scheme can be emulated privately without revealing information to non-vetoers.
- All voting schemes can be emulated anonymously (in a constant number of rounds and with polynomial communication resources, except STV).
- There are partially private protocols for all schemes except rejection and cup.
- When candidates are agents, there are partially private protocols for plurality, rejection, Borda, and approval in which the only publicly revealed information is the winner's score.

## Acknowledgements

We thank the anonymous referees for helpful comments.

This material is based upon work supported by the Deutsche Forschungsgemeinschaft under grant BR 2312/1-1, by the National Science Foundation under grants IIS-9800994, ITR IIS-0081246, and ITR IIS-0121678, and a Sloan Fellowship.

# 8. REFERENCES

[1] K. Arrow. *Social choice and individual values*. New Haven: Cowles Foundation, 2nd edition, 1963. 1st edition 1951.

[2] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the 20th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 1–10. ACM Press, 1988.

[3] J. Benaloh. Secret sharing homomorphisms: Keeping shares of a secret secret. In *Advances in Cryptology - Proceedings of the 13th Annual International Cryptology Conference (CRYPTO)*, volume 263 of *Lecture Notes in Computer Science (LNCS)*, pages 251–260. Springer, 1987.

[12] Using a more general version of the Partition Lemma, our impossibility results can be easily extended to a setting in which curious coalitions may contain only up to $\lceil \frac{n}{2} \rceil$ agents rather than $n - 1$ agents.

[4] F. Brandt and T. Sandholm. (Im)possibility of unconditionally privacy-preserving auctions. In C. Sierra and L. Sonenberg, editors, *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 810–817. ACM Press, 2004.

[5] F. Brandt and T. Sandholm. Unconditional privacy in social choice. In R. van der Meyden, editor, *Proceedings of the 10th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*. ACM Press, 2005. To Appear.

[6] D. Chaum. Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA. In *Proceedings of the 5th Eurocrypt Conference*, volume 330 of *Lecture Notes in Computer Science (LNCS)*, pages 177–182. Springer, 1988.

[7] D. Chaum, C. Crépeau, and I. Damgård. Multi-party unconditionally secure protocols. In *Proceedings of the 20th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 11–19. ACM Press, 1988.

[8] B. Chor, M. Geréb-Graus, and E. Kushilevitz. On the structure of the privacy hierarchy. *Journal of Cryptology*, 7(1):53–60, 1994.

[9] B. Chor and E. Kushilevitz. A zero-one law for Boolean privacy. In *Proceedings of the 21st Annual ACM Symposium on the Theory of Computing (STOC)*, pages 62–72. ACM Press, 1989.

[10] V. Conitzer and T. Sandholm. Vote elicitation: Complexity and strategy-proofness. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI)*, pages 392–397. AAAI Press, 2002.

[11] V. Conitzer and T. Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 781–788, 2003.

[12] V. Conitzer and T. Sandholm. Communication complexity of common voting rules. In *Proceedings of the 6th ACM Conference on Electronic Commerce (ACM-EC)*. ACM Press, 2005.

[13] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *Advances in Cryptology - Proceedings of the 14th Eurocrypt Conference*, volume 1233 of *Lecture Notes in Computer Science (LNCS)*, pages 103–118. Springer, 1997.

[14] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[15] E. Ephrati and J. S. Rosenschein. Deriving consensus in multi-agent systems. *Artificial Intelligence*, 87(1–2):21–74, 1996.

[16] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[17] O. Goldreich. *Foundations of Cryptography*. Cambridge University Press, 2001.

[18] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 218–229. ACM Press, 1987.

[19] A. Kiayias and M. Yung. Self-tallying elections and perfect ballot secrecy. In *Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptography (PKC)*, number 2274 in Lecture Notes in Computer Science (LNCS), pages 141–158. Springer, 2002.

[20] D. E. Knuth and R. W. Moore. An analysis of alpha-beta pruning. *Artificial Intelligence*, 6(4):293–326, 1975.

[21] E. Kushilevitz. Privacy and communication complexity. In *Proceedings of the 30th Symposium on Foundations of Computer Science (FOCS)*, pages 416–421. IEEE Computer Society Press, 1989.

[22] K. May. A set of independent, necessary and sufficient conditions for simple majority decisions. *Econometrica*, 20:680–684, 1952.

[23] D. Parkes and J. Shneidman. Using redundancy to improve robustness of distributed mechanism implementations. In *Proceedings of the 4th ACM Conference on Electronic Commerce (ACM-EC)*, pages 276–277. ACM Press, 2003.

[24] D. Parkes and J. Shneidman. Distributed implementations of Vickrey-Clarke-Groves mechanisms. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 261–268. IEEE Press, 2004.

[25] D. M. Pennock, E. Horvitz, and C. L. Giles. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, pages 729–734, 2000.

[26] B. Pfitzmann and M. Waidner. Unconditionally untraceable and fault-tolerant broadcast and secret ballot election. Hildesheimer Informatik-Berichte, Institut für Informatik, Universität Hildesheim, 1992.

[27] M. A. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.