Absolute Privacy in Voting*

Dmitri Asonov[1], Markus Schaal[2], and Johann-Christoph Freytag[1]

[1] Humboldt-Universität zu Berlin,
10099 Berlin, Germany
{asonov, freytag}@dbis.informatik.hu-berlin.de
[2] Technische Universität Berlin
D-10587 Berlin, Germany
schaal@cs.tu-berlin.de

Abstract. If nobody can prove (even in an all-against-one cooperation) that one did not vote with a particular cast, then one can claim anything about his cast even under oath, and has no fear of being caught. We consider the question of constructing a voting scheme that provides all participants with this "absolute" privacy.
We assume that half of the problem is already solved: The votes are evaluated so that only the result is revealed. Latest achievements of secure coprocessors are supposedly a justification for such a presumption.
We prove that even under the presumption that the voting reveals nothing but a result, the privacy of an individual input can withstand an "all-against-one" attack under certain conditions only.
First condition: The function that maps a set of casts to the result of voting must be non-deterministic. Second condition (paradoxically): for any set of casts any result must be possible.

1 Introduction

The problem of privacy in voting (e.g. [1]) is presumed to be a specific case of secure multiparty computations [2, 3]. That is, the privacy of a voter is presumed to be preserved if the computation of votes is performed in a way that nothing but a result is revealed. This goal is trivially achievable by assuming a third trusted party and private channels between voters and the trusted party. Most of the work on improving privacy in voting is concentrated on achieving the same goal with more and more relaxed cryptographic assumptions [4–8, 1].

We claim, that achieving the above goal is not sufficient to guarantee a voter that his vote cannot be revealed, even if underlying cryptographic assumptions hold. Namely, there is a second, independent problem: Voters, by cooperating against another voter, may reveal his vote by deducing it from their own votes and the result.[1]

* Published at *Information Security Conference 2001.*

[1] This problem was passed over in all the previous work by assuming that the majority would never cooperate to break someone's privacy. The problem is also present in the voting schemes with so-called unconditionally-secret ballots [7] or with information-theoretic privacy [8].

So, only if both problems are solved, a voter can be sure in his privacy *absolutely*. That is, a voter then can be sure, that even if everybody colludes against him, his vote stays private.

Clearly, if simply a sum function is used to calculate the result of voting, then the all-but-one cooperation resolves (and can prove) the vote of a victim by subtraction the sum of their votes from the result. We are interested in finding and investigating functions, that "smooth" the result in a way, that all-but-one cooperation cannot prove how the victim voted (or how the victim did not vote - more on this later). Jumping ahead, we call such functions "absolutely private voting functions" or "private voting functions" for short.

If we find such functions, we say that the voters are provided with *absolute privacy*, assuming of course, that the first problem (of calculating a result in a way, that nothing but a result is revealed) is solved too.

Any constant or, alternatively, some function unknown to the participants would be a private voting function. So we require, that any voting function we consider can not be a constant and must be officially known.

One motivation for this problem setting is to find out whether *absolute* privacy exists at all or not. Although such a privacy is recognized to be important[2], no work is found that deals with absolute privacy in voting.

Along with its academic interest, the "absolutely private voting" setting might be practically applicable in cases where the number of voters is small enough to consider the possibility that all might cooperate against one to break his privacy.

1.1 Preliminaries and Assumptions

Normally, the result of a voting is determined by a well-known function, that maps a set of casts to a voting result. We prove our theorems only for particular kind of functions, that we call voting functions. A voting function is defined in Sect. 1.3. A voting function can be deterministic (one set of casts refers to only one result) or probabilistic (one set of casts refers to several results with some probabilities).

Informally, we say that a voter has an absolute privacy in voting, if no cooperation can break his privacy. His privacy is assumed to be broken if some cooperation of participants may prove how the voter voted. The privacy is also assumed to be broken, if some cooperation may prove how (with what vote) the voter did not vote[3]. This is a privacy violation too because the voter cannot argue anymore that he voted with some arbitrary vote. We give formal definition and motivation for this kind of privacy in Sect. 1.4.

We assume that for any cooperation against any voter no information about his cast is known except the voting result. This assumption is shown to be

[2] It is said in [9]: "Voter privacy must be fail-safe - i.e., it must be assured even if everything fails, everyone colludes and there is a court order to reveal all election data."

[3] These two cases are the same, if two casts are possible only (like yes/no).

impossible if no special hardware is used [3]. However, this assumption may be taken seriously due to the commercially available secure coprocessors that passed FIPS 140-1 Level 4 test (the highest possible rank for a physical and logical protection) [10]. Generally speaking, such a device provides the calculation of any function in such a way that nothing but a result is revealed to several independent parties that provide inputs for this function. From the theoretical point of view, we assume a third trusted party (a secure coprocessor), that processes privately the result of a voting out of given votes.

We also assume, that the result of a voting is made public after the voting. Figure 1 demonstrates the basic architecture of the voting system that we

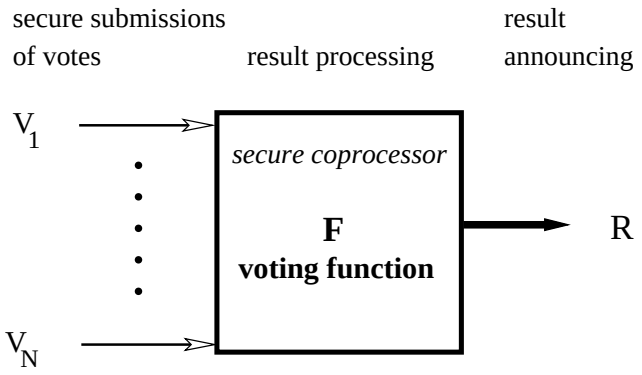

Fig. 1. Private voting using a secure coprocessor.

consider.

1.2 Our Results

We prove that:

1. Absolute privacy is not possible for deterministic voting functions (Sect. 2, Theorem 2). That is, if one wants to have only one result possible for the given casts, one never gains absolute privacy in voting.
2. Absolute privacy is possible for probabilistic voting functions (Sect. 3, Theorem 3). We give an example. Simply speaking, we show how to conduct voting such that every voter has absolute privacy.
3. All (probabilistic) voting functions that preserve absolute privacy have a well-recognized drawback (Sect. 3, Theorem 4). Namely, all results must be assigned to every set of casts with non-zero probabilities, i.e., for any set of casts any result is possible.

1.3 Voting Function

The voting function is formally defined as a function, that satisfies the properties listed below. These three properties (we call them influence, commutativity and openness) are used later to prove our results.

By N we denote the number of voters. R denotes the voting result. The (values of) casts of participants are denoted by $v_1, ..., v_N$. The arguments of the function represent the casts of voters and are the values from the set V, the number of elements in the set is $|V|$. The number of possible results is denoted by $|R|$. For short, below we write $\forall\, v_x$ instead of $\forall\, v_x \in V$.

Property 1: Influence. Voters have an *influence* on the result of voting, i.e.

$$\forall\, v_1, ..., v_N \quad \exists\, j \leqslant N,\, v'_1, ..., v'_j \qquad : \\ F(v_1, ..., v_N) \neq F(v'_1, ..., v'_j, v_{j+1}, ..., v_N) \quad (1)$$

This property basically means, that a constant is not a good function for voting. On the other hand, it does not state that one vote changes result. Instead, it only states that some group of votes can change the result.

Property 2: Commutativity. By this property we require F to be commutative, i.e.

$$\forall\, v_1, ..., v_N,\, 1 \leqslant i < j \leqslant N \qquad : \\ F(v_1, ..., v_i, ..., v_j, ..., v_N) = F(v_1, ..., v_j, ..., v_i, ..., v_N) \quad (2)$$

One might think that this does not reflect voting models, where some voters have a privilege of a stronger impact on the result. The commutative property does not reject such models. It only states that the result does not differ if inputs are processed in different order.

Property 3: Function Openness. By this property we state, that a function definition is known[4]. This property allows us to presume in the proofs, that an all-but-one cooperation (that, of course, may include organizers) knows how to calculate the result of the function for any given input.

$$\text{One knows } F(v_1, ...v_N) \text{ for any } v_1, ..., v_N \quad . \quad (3)$$

Evidently, if nobody knows how the result of a voting is processed, the voting result does not carry any information. And so this is not a voting at all.

Definition 1 (Voting Function). *A function is a voting function, iff it satisfies the properties 1, 2, 3.*

[4] This is a voting scheme property which corresponds to a function used for the result processing. Therefore this property cannot be expressed as a formal mathematical property of a function. This property might be defined as a voting scheme property, but this would change neither the results nor the proofs. So we call it "a voting function property" for the sake of a convenient presentation.

1.4 Private Voting Function

In our definition of absolute privacy, any cooperation against one voter cannot prove that the voter did not vote with a particular value.

Imagine you have participated in a voting where you are interested in keeping your cast *absolutely* private. And one can prove that you did not vote C given that only three casts were allowed (A, B, C). It would be natural for you to consider this as your privacy violation. There is a more specific example of why our definition of absolute privacy is appropriate. It allows a voter to claim that he voted with any arbitrary cast (even under oath), while having no fear of being caught.

Herein "absolute privacy" and "absolutely private" are often reduced to the words "privacy" and "private". In this paper, we do not consider any privacy, except absolute one.

Definition 2 (Private Voting Function). *A voting function F is private, iff for any inputs $v_1, ..., v_N$, given the first $N-1$ inputs and the result $R = F(v_1, ..., v_N)$, for any $A \in V$ it is impossible to prove that $v_N \neq A$.*

Example 1. This example shows what a private voting function might look like. Imagine 100 voters, each votes "1" or "0". A function F summarizes the votes and maps the sum to a number from 0 to 9: If the sum is less or equal 10, then the result of voting is 0; if the sum is more than 10 but less or equal 20, then the result of voting is 1 and so on. The function F satisfies all properties of a deterministic voting function.

Suppose, only 15 of the 100 voters vote with "1". Therefore the sum of the votes is 15, and the function F, by definition, produces result $R = 1$. Can any 99 voters in cooperation say something about the cast of the 100-th voter? The answer is no. Because, no matter, what was the vote of the 100-th voter ("0" or "1"), the result would not change. And, by our presumption, nothing but a result is revealed.

Still, the voting function F is not private, because it does not preserve privacy for arbitrary input. Let the number of those who voted with "1" be 10. Then the result of voting is 0. Now, consider any voter who voted with "0". The cooperation of the rest 99 voters can tell something about his vote. More precise, in this example, they can prove that he did not vote with "1", because if his cast were "1", then the result would change to 1. □

2 Deterministic Voting Functions

First, we prove a theorem about a property any private deterministic voting function must have. Second, we prove that there are no deterministic voting functions that have that property.

But first, let us formally mention, that we consider deterministic voting functions:

$$\nexists \quad v_1, ..., v_N, \qquad : \qquad F(v_1, ..., v_N) \neq F(v_1, ..., v_N) \tag{4}$$

Theorem 1 (The Necessary Condition for a Private Deterministic Voting Function). *A voting function F is private only if*

$$\nexists \quad v_1, ..., v_N,\ v'_N \qquad : \qquad F(v_1, ..., v_N) \neq F(v_1, ..., v'_N) \tag{5}$$

This is equivalent to

$$\forall \quad v_1, ..., v_N,\ v'_N \qquad : \qquad F(v_1, ..., v_N) = F(v_1, ..., v'_N)$$

Proof. Suppose, from contradiction, that some voting function F is private, and (5) does not hold:

$$\exists \quad v_1, ..., v_N,\ v'_N \qquad : \qquad F(v_1, ..., v_N) \neq F(v_1, ..., v'_N)$$

Let the casts be exactly $v_1, ..., v_N$. So, the last equation is true for this situation:

$$\exists \quad v'_N \qquad : \qquad F(v_1, ..., v_N) \neq F(v_1, ..., v'_N)$$

This means that $N-1$ cooperating voters, who know their own votes $v_1, ..., v_{N-1}$ and the result of the voting R, can (due to (3)) compute $F(v_1, ..., v_{N-1}, x_N)$ for all the range of the values of a vote x_N. This way they find such an $x_N = v'_N$, that

$$F(v_1, ..., v'_N) \neq R$$

From the property 4 of a deterministic function they conclude, that $v'_N \neq v_N$. That is, they can prove that the N-th voter did not vote with v'_N value. (If there are only two possible values for votes, like "yes" or "no", they can even tell exactly what his vote was.)

Formally, the coalition of $N-1$ voters has the right to conclude, that

$$v_N \neq v'_N$$

Then, by definition 2, F is not a private function. It is a contradiction. □

Theorem 2 (Nonexistance of Private Deterministic Voting Function). *A deterministic voting function cannot be private.*

Proof. Assume that a private deterministic voting function F exists. Then the properties of a deterministic voting function (statements 1, 2, 3, 4) are true for F, and F satisfies the necessary condition of a private deterministic voting function (Theorem 1). Starting with this, we lead to a contradiction.

Let the casts of the voters be $v_1, ..., v_N$ and let the result be R:

$$F(v_1, ..., v_N) = R$$

STEP1. We can find the minimal number of voters j, that would change result by changing their casts. Due to the voting function property 1, this number exists and it is less then or equal to N. Due to Theorem 1, this number must be more than 1:

$$\exists \quad 1 \underset{Theorem\ 1}{<} j \underset{Property\ 1}{\leqslant} N \quad :$$

$$\forall \quad v_1^\star, ..., v_{j-1}^\star \quad : \quad F(v_1^\star, ..., v_{j-1}^\star, v_j, v_{j+1} ..., v_N) = R \tag{6}$$

$$\text{And} \quad \exists \quad v_1', ..., v_{j-1}', v_j' \quad : \quad F(v_1', ..., v_{j-1}', v_j', v_{j+1}, ..., v_N) \neq R \tag{7}$$

STEP2. In (6), let us take $v_1^\star = v_1', ..., v_{j-1}^\star = v_{j-1}'$:

$$F(v_1', ..., v_{j-1}', v_j, v_{j+1}, ..., v_N) = R \tag{8}$$

By combining (7) and (8), we get:

$$F(v_1', ..., v_{j-1}', v_j, v_{j+1}, ..., v_N) \neq F(v_1', ..., v_{j-1}', v_j', v_{j+1}, ..., v_N)$$

In the last expression, if we take into account property 2 of a voting function, then we get a contradiction to Theorem 1. □

3 Probabilistic Voting Functions

We considered deterministic functions. Probabilistic functions are more general functions in the sense, that for unique input they make possible different results with different probability.

In our notation, a probabilistic function is defined by assigning a finite set of result–probability pairs for each possible set of casts (see Fig. 2). Considering the example where the result is calculated inside a secure coprocessor, a secure coprocessor simply outputs the result in accordance to given probability distributions.

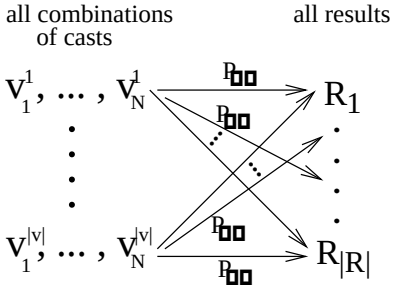

Fig. 2. Visual representation of a probabilistic function.

In this section we prove, although private voting is possible with probabilistic functions, any result is possible too. This might be viewed as a significant drawback for those who "make some use" of a result of an absolutely private voting.

From now on, whenever we refer to a voting function, or a private voting function, we undermine a probabilistic voting function or a private probabilistic voting function, if not specified exactly.

3.1 Voting Function

First, we rewrite shortly the definitions of voting function properties because the notations are slightly changed to carry the notion of probability.

Influence. Voters have an *influence* on the result of voting, i.e.

$$\forall\, v_1, ..., v_N, R \quad \exists\, j \leqslant N,\ v'_1, ..., v'_j \qquad : \\ P(R|v_1, ..., v_N) \neq P(R|v'_1, ..., v'_j, v_{j+1}, ..., v_N) \quad (9)$$

This property means, that if the distribution is the same for any casts, this probabilistic function is not good for voting.

Commutativity. Voters have an *equal* influence on the result of voting, i.e.

$$\forall\, v_1, ..., v_N, R, 1 \leqslant i < j \leqslant N \qquad : \\ P(R|v_1, ..., v_i, ..., v_j, ..., v_N) = P(R|v_1, ..., v_j, ..., v_i, ..., v_N) \quad (10)$$

Function Openness. Voters (or at least organizers) know how the votes are counted, i.e.

$$\text{One knows } P(R|v_1, ..., v_N) \text{ for any } R, v_1, ..., v_N \ . \quad (11)$$

3.2 Private Voting Function

We continue using the same definition of a private voting function as in the deterministic case (Definition 2).

Example 2. Let us start with the deterministic example given in Sect. 1. We might try to fix the problem in that example in the following way:

1. If the sum is not critical for privacy (like 1,2,3,4,5,6,7,8,9,12,13,14,15,...), we output the result as before: No privacy is revealed as we discussed. (Because, for these sums, changing one vote does not change the result.)
2. If the sum is critical (like 10 or 11), we run some probabilistic function, that outputs results (like 0 or 1) with equal probability.

Suppose 11 voters voted with "1" and the result announced is 1, and 99 voters cooperate against the one (who voted with "1"). They cannot argue, that his vote was "1" (as they do in the deterministic example) because he could voted with "0" (with the same probability), then the sum would be 10, but then it would be flipped to the result 1 due to our new voting function for critical sums.

Still, the described probabilistic voting function does not preserve privacy. Counterexample: Again, consider 11 voters who voted with "1" and the result announced is 0, and 99 voters cooperate against the one (who voted with "0"). They can argue, that his vote was not "1", because if it were "1", then the sum would be 12 and the result 0 would be impossible.

Below we prove formally, that the private voting is still possible, but (considering this example) only if sometimes (very rarely) for the all 100 votes "1" the result 0 appears, and (also very rarely) for the all 100 votes "0" the result 9 appears. □

Theorem 3 (Existance of a Private Probabilistic Voting Function). *There exists a private probabilistic voting function.*

Proof. To prove the theorem, we give an example of a private probabilistic voting function. In other words, we give an example of a function, that satisfies all three voting function properties listed before (statements 9, 10, 11) and that preserves absolute privacy of any voter (Definition 2).

There are 3 voters, two possible casts (0/1 or yes/no) and 4 different results (Fig. 3).

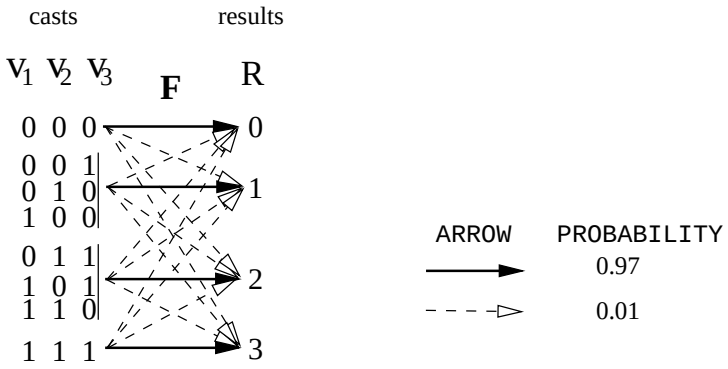

Fig. 3. An example of a private probabilistic voting function.

Clearly, the scheme might be generalized for any N, $|V|$ and $|R|$.

It also may be slightly adjusted so, that the less the result corresponds to the set of casts, the smaller probability it has (In our case, for simplicity, all such probabilities are 0.01). □

It is remarkable, that removing any single arrow makes the function not private. This is the subject of the next theorem.

The next theorem might be seen as a necessary condition for a function to be a private probabilistic voting function. It also shows how "poor" the voting functions must represent the result of a voting in order to provide absolute privacy. Roughly speaking, it proves, that nobody can give an example of an

absolutely private voting function, where for all 100 "yes" votes the result "no" is impossible.

Theorem 4 (The Flaw of a Private Probabilistic Voting Function). *Any private probabilistic voting function F has the following property:*

$$\forall R, v_1, ..., v_N \quad : \quad P_{F(v_1,...,v_N)=R} \neq 0 \tag{12}$$

Proof. We give a very short proof, although it might be extended for better understanding.

Let us consider possible results (the probabilities are not zero) for the set of votes $v_1, ..., v_N$. We take one of this (possible) results - R. Starting with this, we prove, that R has a non-zero probability for any other set of votes. This proves the theorem.

So, we have

$$P(R|v_1, ..., v_N) \neq 0$$

Then let us change one of the votes in the set to the arbitrary value. Let it be $v_N \longrightarrow v'_N$. Then we can write

$$P(R|v_1, ..., v'_N) \neq 0$$

If we cannot write the last equation, then, considering the equation before the last, we have absolute privacy definition violation, if the first $N-1$ voters cooperate against the N-th.

Using the same technique, by changing votes one by one, we achieve

$$P(R|v'_1, v'_2..., v'_N) \neq 0$$

□

4 Related Work

Nothing in the related work tackles the problem setting we consider.

4.1 Secure Multi-Party Computation

Secure multi-party computation (SMPC) deals with computing *securely* any probabilistic function in a distributed network where each participant holds one of the inputs. "Securely" means that no more information is revealed to a participant than can be computed from that participant's input and a result [3, 2]. Thus SMPC does not consider the question of how much information about a single input can be revealed in the result given the other inputs. Our work might be seen as a partial answer to this question.

4.2 Electronic Elections

In the research work on security and privacy in electronic elections (see [6, 7, 1] for example), an enormous number of voting schemes have been proposed. The similarity between all those schemes is that in none of them the voter's privacy withstands an all-against-one attack. There is only one exception: Stochastic anonymous voting is discussed later in this section.

It seems that the hypothetical question of what it costs to protect the privacy of a single voter (or can it be protected at all) if all-but-one voters cooperate is not pondered even once.

Receipt-Free Voting. To stop vote buying, receipt free voting [11, 12] schemes are proposed that prevent (under such assumptions as "private booth" and "private channel" only) a voter from proving his cast to somebody else. So the all-against-one conspiracy of voters still may *know* the victim's vote, but they cannot *prove* it to somebody else. Because in order to prove it they should prove their own votes, that is made impossible by the receipt-freeness. Again, the question is not considered how a single vote might be protected if all other votes are somehow made known.

Stochastic Anonymous Voting. In [13] a stochastic voting protocol is proposed with the main idea that the voters have to randomize their votes before submitting them. A technique is also proposed of how do organizers force users to randomize their votes. Then, even if all votes are made known, the voters' privacy is preserved.

In that protocol, for any set of votes, any result is possible. However the question is not considered whether there exists such an absolutely private voting scheme, that not all results are possible for any given set of votes.[5] Instead, some statistical properties of the protocol are investigated. And the primary result of that work is that the accuracy of the voting result improves as the number of voters increases.

Real Systems. There are a lot of voting systems implemented and offered by commercial companies. Some of them build their advertising campaigns on terms like "absolute privacy"[6]. What they probably mean is privacy under the assumption that all-but-one conspiracy of voters is impossible.

[5] And the main achievement of our work is that we give and prove the *negative* answer to this question.

[6] "Our technology provides for absolute privacy of an individual's ballot..." is officially claimed by one electronic voting company. Yet another company claims to provide "fail-safe" voter privacy, where fail-safe means that one cannot link a voter to a vote even if everyone colludes etc.

4.3 Statistical Disclosure Control

The problem for a statistical database is how to preserve the privacy of individual data from the anyone, who wishes to get some statistics calculated on the set of several individual data [14]. The absolutely private voting problem is different, since not only the result of the processing is known, but also all except one individual data are known too.

4.4 One-Way Hash Functions

One-way hash functions produce the result in a way that, given the result and not given some (even relatively small) part x_i of the input X, it is computationally difficult to say what is this hidden input x_i equal to [15]. From the first point of view, such functions might be considered as private voting functions. They are not, because for one-way functions, although it is difficult to find $A: \quad x_i = A$, it costs nothing to find $A: \quad x_i \neq A$. And the latter is a privacy violation in case of voting.

4.5 Theories of Voting

The mathematics of voting (or, the theory of voting), in spite of being rather developed [16], has nothing relevant to the problem we consider. One of the corner goals of the theory is to calculate votes so, that the winner (one of more than two candidates) is "whom the voters really want" [17]. In this context, an impossibility for some particular electoral systems [18] is just one of the paradoxes known to the voting theory [19, 20].

A probabilistic voting theory [21] has a goal different from the voting theory. Probabilistic voting theory is the mathematical prediction of candidate behaviour in, or in anticipation of, elections in which candidates are unsure of voters' preferences. This theory, as well as similar ones – spatial theory of voting [22, 23] and a unified theory of voting [24], does not consider privacy questions.

4.6 No Voting – No Problem

Some work was dedicated to show that an election, as a main instrument of democracy, has many disadvantages. The basic one is that after the election occurs, a winner holds the promises no more. So, the election is just an illusion.

These disadvantages of an election might be removed just by avoiding elections. Interestingly, one insist that democracy is still well possible without elections [25]. Instead of elections, other techniques are proposed to make decisions in the society.

Some of those techniques, like referendums[7], have the same mathematical model as elections, so the privacy problem remains. Other alternatives are com-

[7] Instead of electing politicians who then make policy decisions, these decisions are made directly by the public voting [25].

pletely different from normal elections, so that the notion of privacy is meaningless for them. One of the most radical alternatives (the sortition) is to take randomly one of all allowed outcomes as a result [26].

5 Discussion

In this section we clarify some questionable points of the paper.

5.1 The Absolutely Private Voting Scheme Proposed

From some prospective, using the absolutely private voting function we proposed in Theorem 3 is not very different from picking candidates at random. And so, one might conclude the proposed voting scheme is useless.

The proposed voting scheme is a side-effect of our work. So we do not discuss much about whether this system is good for voting or not - this is done very well by Kikuchi et al. [13] using the probability theory. The main point of our paper is a proof of the paradoxical statement: If one wants to build an absolutely private voting system, he can do nothing better than (accidentally very similar) our or Kikuchi et al. [13] scheme.

5.2 The Definition of Voting Function

Our definition of a voting function is very general. Basically, it is any non-constant function. The question is whether we should define it more strictly. We think that the more general the functions we consider to prove the theorems, the more universal these theorems are.

For example, we could improve the definition of a voting function by saying that the voting function should indicate undoubtedly if the majority (or generally, a given amount) of voters voted for a proposition. But it follows directly from Theorem 4, that such improved voting functions can not be private. It is also evident that if we substitute "undoubtedly" for "with high probability" then such private functions exist.

6 Future Work

An open issue in this work is what one can say about the privacy of a single vote, if at most N-2 (or N-k) participants are allowed to cooperate.

The work describing applications of our results for some practical Private Information Retrieval schemes (like one described in [27]) is in progress.

7 Conclusion

Voting is absolutely private if a voter can insist that he voted with an arbitrary cast (to preserve his privacy) and nobody (in any conspiracy) can prove that he lies.

Our goal was to investigate whether or not absolute privacy is possible. And if it is possible, then under which conditions.

We considered special types of functions, that are likely to reflect the most important properties of any voting.

For these functions, we proved that absolute privacy holds only if the voting function is probabilistic and for any set of casts any result is possible.

In summary, absolute privacy has a price. It is up to the participants to decide whether they want the "real" result or "real" privacy. It is impossible to have both simultaneously[8]. We have shown that there is a tradeoff between the absolute privacy of a vote and the precision of a result of the voting.

Acknowledgements. Many thanks go to Myra Spiliopoulou for the valuable suggestions on the terminology and the context, and to Bernhard Thalheim for an interesting discussion about related work, and to anonymous referees for valuable notes and recommendations. Additional comments were kindly given by Hans-Joachim Lenz, Oliver Günther, Johannes Köbler, and Rainer Conrad.

This research was supported by the German Research Society, Berlin-Brandenburg Graduate School in Distributed Information Systems (DFG grant no. GRK 316).

References

1. Cramer, R., Gennaro, R., Schoenmakers, B.: A secure and optimally efficient multi-authority election scheme. In: Theory and Application of Cryptographic Techniques. (1997) 103–118
2. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Proceedings of STOC'87. (1987)
3. Goldreich, O.: Preface to special issue on general secure multi-party computation. `http://www.wisdom.weizmann.ac.il/~oded/PS/preSI.ps` (1999)
4. Chaum, D.: Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM **24** (1981) 84–88
5. Cohen, J., Fischer, M.: A robust and verifiable cryptographically secure election scheme. In: Proceedings of 26th FOCS. (1985)
6. Cohen, J.: Improving privacy in cryptographic elections. Technical Report 454, Yale University, Department of Computer Science (1986)
7. Chaum, D.: Elections with unconditionally-secret ballots and disruption equivalent to breaking RSA. In: Advances in Cryptology: Proc. of EuroCrypt'88, LNCS 330, SpringerVerlag. (1988) 177–182
8. Cramer, R., Franklin, M., Schoenmakers, B., Yung, M.: Multi-authority secret-ballot elections with linear work. In: Proceedings of EUROCRYPT'96, LNCS 1070. (1996)
9. Gerck, E.: Internet voting requirements. The Bell **1** (2000) 3–5,11–13

[8] This might be seen as a self-contradiction of democracy: Privacy and voting are two important aspects of democracy. But we have shown that they cannot perfectly coexist, i.e., one or the other must inherently be flawed.

10. Smith, S.W., Palmer, E.R., Weingart, S.H.: Using a high-performance, programmable secure coprocessor. In: Proceedings of the 2nd International Conference on Financial Cryptography, Springer-Verlag LNCS. (1998)
11. Benaloh, J., Tuinstra, D.: Receipt-free secret-ballot elections. In: Proceedings of the 26th ACM Symposium on Theory of Computing. (1994) 544–553
12. Hirt, M., Sako, K.: Efficient receipt-free voting based on homomorphic encryption. In Preneel, B., ed.: Advances in Cryptology - EUROCRYPT'00. Volume 1807 of Lecture Notes in Computer Science., Springer-Verlag (2000) 539–556
13. Kikuchi, H., Akiyama, J., Gobioff, H., Nakamura, G.: Stochastic anonymous voting. Technical Report CMU-CS-98-112, Carnegie Mellon University (1998)
14. Willenborg, L., de Waal, T.: Statistical Disclosure Control in Practice. Volume 111 of Lecture Notes in Statistics. Springer-Verlag (1996)
15. Schneier, B.: Applied Cryptography. 2nd edn. Wiley, New York (1996)
16. Saari, D.G.: Basic Geometry of Voting. Springer-Verlag (1995)
17. Saari, D.G.: Geometry, voting, and paradoxes. Mathematics Magazine (1998) 243–259
18. Woodall, D.R.: An impossibility theorem for electoral systems. Discrete Mathematics (1987) 209–211
19. Saari, D.G.: A dictionary for voting paradoxes. Journal of Economic Theory (1989) 443–475
20. Nurmi, H.: Voting Paradoxes and How to Deal with Them. Springer-Verlag (1999)
21. Coughlin, P.J.: Probabilistic Voting Theory. Cambridge University Press (1993)
22. Enelow, J.M., ed.: Spatial Theory of Voting. Cambridge University Press (1984)
23. Enelow, J.M., Hinich, M.J., eds.: Advances in the Spatial Theory of Voting. Cambridge University Press (1990)
24. Samuel Merrill, I., Grofman, B.: A Unified Theory of Voting. Cambridge University Press (1999)
25. Martin, B.: Democracy without elections. Social Anarchism (1995-96) 18–51
26. Carson, L., Martin, B.: Random selection in politics. Praeger (1999)
27. Smith, S.W., Safford, D.: Practical private information retrieval with secure coprocessors. Technical report, IBM Research Division, T.J. Watson Research Center (2000)