

Time, Privacy, Robustness, Accuracy: Trade Offs for the Open Vote Network Protocol

Gergei Bana*, Marco Biroli[†], Megi Dervishi[†], Fatima-Ezzahra El Orche^{†‡} Rémi Géraud-Stewart^{†§}
David Naccache[†], Peter B. Rønne[‡], Peter Y.A. Ryan[‡], Hugo Waltsburger[¶]

*University of Missouri at Columbia, Columbia MO, USA

Email: banag@missouri.edu

[†]Département d’informatique de l’ENS, ENS, CNRS, PSL University, Paris, France

Email: {given_name.family_name}@ens.fr

[‡]SNT, FSTC, University of Luxembourg, Luxembourg

Email: {fatima.elorche, peter.roenne, peter.ryan}@uni.lu

[§]QPSI, Qualcomm Inc., San Diego CA, USA

Email: rgerauds@qti.qualcomm.com

[¶]CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette, France

Email: hugo.waltsburger@centralesupelec.fr

Abstract—Open vote network (OV-Net, [1]) is a secure multi-party protocol allowing to compute a sum of integer votes without revealing their values. As such it has several applications in social choice and financial applications [2], [3].

An inherent limitation of OV-Net is its lack of robustness against denial-of-service attacks, which occur when at least one of the voters initiates the protocol but (maliciously or accidentally) does not complete it. Unfortunately such a situation is very likely to occur in any real-world implementation of the protocol. This will cost serious time delays from either waiting for the failing parties and perhaps having to perform extra protocol rounds with the remaining participants.

This paper provides a solution to this problem by extending OV-Net with mechanisms tolerating a number of unresponsive participants. The price to pay is a carefully controlled privacy loss, an increase in computation, and a statistical loss in the accuracy.

Index Terms—Multi-party computation, open vote network, denial of service

I. INTRODUCTION

Cryptographic voting protocols allow mutually-distrusting entities to reach a consensus on some value. This has applications in social choice (e.g., elections) as well as financial mechanisms [2], [3]. Most of these protocols involve a trusted authority responsible for running the election or tallying the results. However, there exist a number of so-called “boardroom” or “self-tallying” schemes that do away with the need for a central authority [4]. In such schemes, the election is an interactive protocol between the voters only. Whether a centralised or decentralised protocol is better-suited to a given situation depends on practical and context-specific concerns, such as the protocol’s scalability, or whether the trusted authority assumption makes sense. For instance, the self-tallying protocol of Groth [5] has $n + 1$ rounds for n voters, which makes it impractical to use for larger elections.

Open vote network (OV-Net) is a self-tallying voting scheme proposed by Hao, Ryan and Zieliński [1]. Improving upon Hao and Zieliński’s earlier AV-net [6], [7], it is a 2-round

protocol which makes it an appealing candidate for larger-scale elections. One of OV-Net’s limitations, according to Hao–Ryan–Zieliński, is that the protocol cannot handle denial-of-service (DoS) events:

“ (...) For example, if some voters refuse to send data in round 2, the tallying process will fail. This kind of attack is overt; everyone will know who the attackers are. To rectify this, voters need to expel the disrupters and restart the protocol; their privacy remains intact. However the voting process would be delayed, which may prove costly for large scale (countrywide) elections (...)” — [1, Sec 3.4]

While the protection of privacy and the identification of culprits are desirable properties, the need to restart the protocol every time a voter drops out is a very strong limitation. This weakness is what we set out to rectify in this paper, by extending OV-Net to handle DoS events gracefully. Our modifications come at a cost, which we investigate.

Some earlier works have already tried to improve the security and efficiency of OV-Net. In [8] fairness (i.e. preventing that voters get partial results before casting their vote) was guaranteed by committing to the vote in the first round. Further, the robustness against denial of service attacks was improved by introducing a recovery round: if some voters did not participate in the second round, the remaining voters perform a third round to achieve the partial tally for their cast votes. However this does not guarantee that there are no fall outs in the recovery round. In [9] it was shown that using a bilinear group setting and assuming a public key infrastructure, the voting protocol can be made non-interactive, i.e. 1-round. This increases the run time considerably, but does not in itself remove the robustness problem. Finally, in [2] the OV-Net was implemented via a smart contract that financially punishes voters who drops out of the election. This gives an economic incentive to participate in the second round, but does

not prevent dedicated DoS attacks, nor involuntary drop outs e.g. due to lack of network access, and it assumes that the participants are willing to risk the economic punishment in the first place.

II. PRELIMINARIES

A. Notations

Throughout this paper, we will use the following notations. If X is a finite set, $x \stackrel{\$}{\leftarrow} X$ means that x is sampled uniformly at random from X . When working in a cyclic group \mathbb{G} generated by g , we write $[x]$ to denote g^x . If $q > 1$ is an integer, we denote by $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$ the ring of integers modulo q . We write $\vec{1}$ is the vector whose coordinates are all 1. $\text{BD}(p, n)$ denotes the binomial distribution of mean p for a population n .

B. Open vote network (OV-Net)

We recall here the OV-Net protocol in the case of of a referendum: there are two choices $A = 1$ and $B = 0$ and n voters; each voter will cast a vote v_i for either A or B (i.e., $v_i \in \{0, 1\}$) and the final tally will reveal the sum of all votes. (Ultimately, we may set a threshold to adopt A or B as a final decision based on the tally, but this is beyond the scope of OV-Net.)

We assume that all participants have agreed ahead of time to use a given cyclic group \mathbb{G} of generator g in which the decisional Diffie–Hellman problem is intractable. Let q be the order of \mathbb{G} . Each voter $i \in \{1, \dots, n\}$ samples a random value $x_i \stackrel{\$}{\leftarrow} \mathbb{Z}_q$ as a secret.

- 1) **Round 1:** Each voter $i \in \{1, \dots, n\}$ publishes g^{x_i} along with a zero-knowledge proof of knowledge (ZKP) of x_i . In practice, ZKP can be a Schnorr signature [10]. When this round finishes, each voter $i \in \{1, \dots, n\}$ does the following:
 - checks the validity of the ZKP for all $x_j, j \in \{1, \dots, n\} \setminus \{i\}$,
 - computes:

$$g^{y_i} = \prod_{j=1}^{i-1} g^{x_j} / \prod_{j=i+1}^n g^{x_j}$$

- 2) **Round 2:** Each participant $i \in \{1, \dots, n\}$ publishes $g^{x_i y_i} g^{v_i}$ and a ZKP for v_i showing that $v_i \in \{0, 1\}$. In practice, this proof can be implemented using the technique of Cramer–Damgård–Schoenmakers [11].

At the end of this procedure, each voter checks the proof of knowledge of all others, and multiplies together all the $g^{x_i y_i} g^{v_i}$'s. The result is $g^{\sum_{i=1}^n v_i}$, from which the value $\sum_{i=1}^n v_i$ can be recovered by solving the discrete logarithm problem in G — this is tractable because n is small (by cryptographic standards), with the total world population being less than 2^{34} . Thus generic algorithms such as Pollard's ρ , with a complexity of $O(\sqrt{q})$, can be used here.

Remark 1. The OV-Net protocol can be extended to more than two candidates by an appropriate encoding of v_i [12], [13], with the final tally requiring a (superincreasing) knapsack resolution after a discrete logarithm computation [1, Sec. 2.2]. We focus on the simpler case of two candidates.

C. Denial of Service

In the description of OV-Net, we implicitly assume that all participants are honest, to the extent that the proofs of knowledge are valid and that they follow the protocol. If one or several voters publish an incorrect proof of knowledge, or do not follow through with the protocol, then it is impossible to reach a conclusion for this particular vote event. This is called a denial of service (DoS) event.

When a DoS event occurs, the non-compliant voters can be identified and removed from a subsequent vote. However the results for that particular vote must be discarded (or cannot be computed) and a fresh vote must take place. This is troublesome for several reasons. One reason is that as n becomes large, disconnection or time-out events become more common and therefore the protocol's failure probability increases. Another reason is that accounting for protocol errors and re-voting adds complexity to real-world OV-Net implementations.

III. PARALLEL OV-NET

We consider a modification of OV-Net where users participate in several voting sessions in parallel. However, not all voters take part to all votes, as we now explain. Let n be the number of voters and M the number of parallel vote sessions. Each voter will participate in k pseudo-randomly chosen sessions amongst M .

More precisely, voter i picks k sessions before the protocol is run which we call i 's selection. We assume that this selection is pseudo-random, i.e. that any given selection happens with the same probability $1/\binom{M}{k}$. As a result not all sessions have the same number of voters, a phenomenon that we will need to account for.

Remark 2. A natural question is whether we could impose some more clever rule, that would guarantee that there is always the same number of voting opportunities for each of them. Indeed, a solution is provided, in some cases, by Steiner systems [14]: a Steiner system with parameters t, k, n , written $S(t, k, n)$, is an n -element set S together with a set of k -element subsets of S (called blocks) with the property that each t -element subset of S is contained in exactly one block.

The existence of Steiner systems is deeply connected to number-theoretic properties, and in particular the existence of a $S(t, k, n+1)$ system precludes that of a $S(t, k, n)$. Thus, although we could initially form a balanced set of voters in some initial setting, it cannot be done if any of the voters bails out (or is disconnected).

However, it is not obvious how a decentralised pool of voters could agree on such a setting in a non mutually-trusting way and without leaking private information. It also remains an interesting question whether approximately balanced block

designs exist that are “stable” in the sense that they retain this property when elements are removed.

Should one voter drop out during one voting session, this particular session will be discarded, but all sessions in which this voter didn’t participate will go through. Unfortunately this also discards all the votes of honest voters in the dropped session. To overcome this exclusion we allow each voter to vote k times: in other words, each voter will cast k votes into k independent ballots amongst the M .

Our claim is that in this case, the final tally’s result reflects the choice of honest voters even after discarding all the sessions that were blocked by a dishonest voter. Furthermore, when several voters are dishonest, their cumulative effect on the final tally is weighed down by the fact that they shared many vote sessions. Concretely, for $k = M/2$, the first dishonest voter makes about $M/2$ sessions invalid; but amongst the remaining sessions only about $M/4$ can share a second dishonest voter, etc. Hence, this setting tolerates roughly $\log_2 M$ droppers, at the price of running M sessions.

In summary, by running several sessions, several competing phenomena occur:

- 1) The overall protocol’s resilience against DoS events is improved as we run more sessions — more sessions however bring an additional computational and communication cost;
- 2) Sessions have a varying number of voters in them, and not every voter partakes in every session, which introduces a bias — we can expect this bias to become small when many sessions are run;
- 3) The list of participants in each session is public, therefore some information about individual voters’ preferences is leaked — running more sessions results in a increased loss of privacy.

There is therefore a balance to be struck, and we must quantify these phenomena more precisely.

IV. PARALLEL OV-NET DOS RESILIENCE

Let ℓ be the number of voters causing a DoS event; they cause a (random) number X_ℓ of sessions to be discarded. The protocol fails when all sessions have been discarded, i.e., when $X_\ell \geq M$ — this cannot happen when $\ell < M/k$. If $\ell \geq M/k$ then it is possible to stop the protocol entirely when the selections of dropping voters cover all sessions. However, the likelihood of this happening when each selection is random and independent is low, as many of the dropping voters will have sessions in common.

This is a particular variant of the famous coupon collector’s problem, which has been extensively studied.

Lemma 1. *The average number of DoS events necessary to cause an overall failure, when we run M parallel sessions and each voter partakes in k of them is*

$$\mathbb{E}[\ell \mid \text{overall protocol failure}] = \binom{M}{k} \sum_{r=1}^M (-1)^{r-1} \frac{\binom{M}{r}}{\binom{M}{k} - \binom{M-r}{k}}.$$

Proof. Let ℓ be the number of DoS events causing a protocol failure. W.l.o.g. these events correspond to dropping voters $r = 1, \dots, \ell$. Let $\text{sel}(r)$ be the selection of r and let T_j be the index of the first dropping voter that has j in its selection. Thus $\ell = \max_j T_j$.

Let L be an integer, note that $\max_j T_j > L$ if and only if there is an index v such that $T_v > L$. Similarly, for all $1 \leq j_1 < j_2 < \dots < j_v \leq M$ we have $\min\{T_{j_1}, \dots, T_{j_v}\} > L$ if and only if T_{j_u} for all $u \in \{1, \dots, v\}$. The inclusion-exclusion formula then gives

$$\Pr[\ell > L] = \sum_{v=1}^M (-1)^{v-1} \sum_{1 \leq j_1 < \dots < j_v \leq M} \Pr[\min\{T_{j_1}, \dots, T_{j_v}\} > L].$$

The right-hand side happens only when none of the sessions j_1, \dots, j_v belongs to the first u selections; using independence, we have

$$\Pr[\min\{T_{j_1}, \dots, T_{j_v}\} > L] = \left(\frac{\binom{M-v}{k}}{\binom{M}{k}} \right)^L$$

hence

$$\begin{aligned} \Pr[\ell > L] &= \sum_{v=1}^n (-1)^{v-1} \sum_{1 \leq j_1 < \dots < j_v \leq M} \left(\frac{\binom{M-v}{k}}{\binom{M}{k}} \right)^L \\ &= \sum_{v=1}^n (-1)^{v-1} \binom{M}{v} \left(\frac{\binom{M-v}{k}}{\binom{M}{k}} \right)^L \end{aligned}$$

from which we obtain the claimed formula by computing the expected value

$$\mathbb{E}[\ell] = \sum_{L \geq 0} \Pr[\ell > L].$$

□

Figure 1 compares simulation results to the formula of Lemma 1, showing excellent agreement. The simulation is for $M = 50$ and k varying from 1 to 49, over 10^5 runs¹. Using this information, we can choose parameters M and k to accommodate a given number of potential drop-outs.

When we have fewer than the critical number of DoS events, the remaining sessions can be tallied. We can estimate the number of remaining valid sessions as $\mu = M - X_\ell$:

Lemma 2.

$$\mathbb{E}(\mu) = (M - k) \left(1 - \frac{k}{M} \right)^{\ell-1}$$

Proof. We compute the expected number of surviving sessions $\mathbb{E}(M - X_\ell)$ when ℓ voters have quit the voting process, and we deduce $\mathbb{E}(X_\ell)$ by induction:

- $\mathbb{E}(X_1) = k = M - (M - k)(M - k)^0$

¹The corresponding Python code is available from the authors upon request.

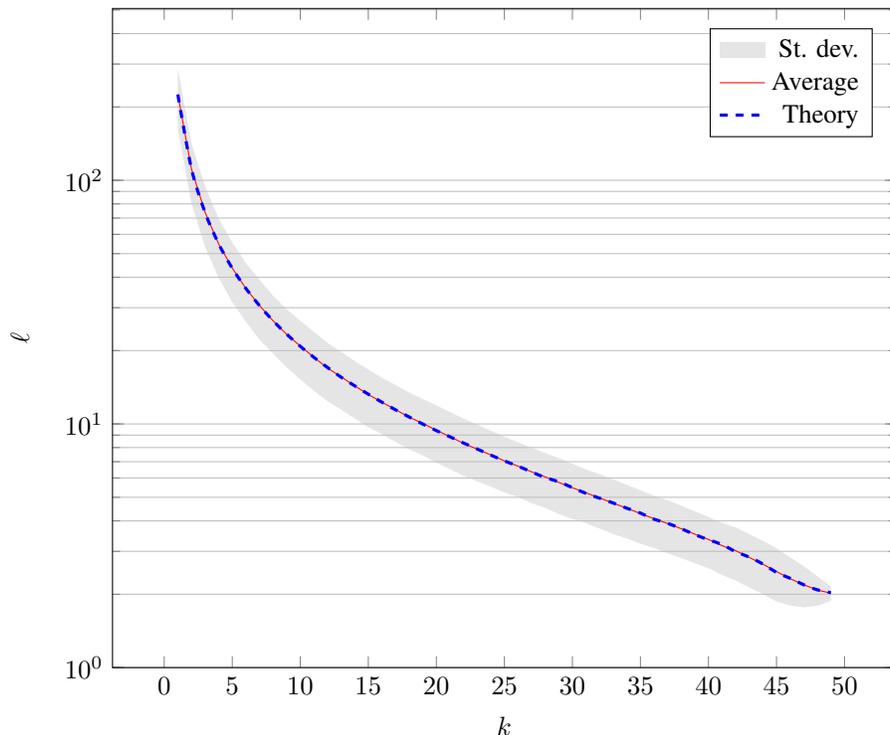


Fig. 1. Simulated and predicted minimum number of DoS events necessary to cause an overall protocol failure, for $M = 50$ and $k = 1, 2, \dots, 49$.

- Assume that the formula is correct until ℓ . When a new dropout happens, an average of $\frac{k}{M}\mathbb{E}(M - X_\ell)$ sessions will be discarded, thus:

$$\begin{aligned}\mathbb{E}(M - X_{\ell+1}) - \mathbb{E}(M - X_\ell) &= -\frac{k}{M}\mathbb{E}(M - X_\ell) \\ M - \mathbb{E}(X_{\ell+1}) - M + \mathbb{E}(X_\ell) &= -\frac{k}{M}(M - \mathbb{E}(X_\ell))\end{aligned}$$

Therefore

$$\begin{aligned}\mathbb{E}(X_{\ell+1}) &= \mathbb{E}(X_\ell) \left(1 - \frac{k}{M}\right) + k \\ &= \left(M - (M - k) \left(1 - \frac{k}{M}\right)^{\ell-1}\right) \left(1 - \frac{k}{M}\right) + k \\ &= M - (M - k) \left(1 - \frac{k}{M}\right)^\ell\end{aligned}$$

□

Finer results about the distribution X_ℓ are given in the following section.

Distribution of X_ℓ

We have the following recurrence relation:

Lemma 3.

$$\Pr[X_{\ell+1} = x | X_\ell = r] = \frac{\binom{M-r}{x-r} \cdot \binom{r}{k-x+r}}{\binom{M}{k}}.$$

Let us explain this formula term by term: for $X_{i+1} = j$, knowing that $X_i = \ell$ we know that a not-yet-accounted-for

quitter voted in $j - \ell$ sessions that did not overlap with the previous ones and in $k - j + \ell$ sessions that overlapped with the previous ones. There are $M - \ell$ live sessions and ℓ discarded ones, the choice of which sessions the quitter will vote in therefore gives us the first two terms at the numerator. The denominator counts all the possible votes. In matrix form,

$$A_{i,j} = \Pr(X_i = j) \quad \text{and} \quad B_{i,j} = \frac{\binom{M-j}{i-j} \binom{j}{k-i+j}}{\binom{M}{k}}$$

This enables us to simulate the probability distribution functions for any set of parameters. Figure 2 shows numerical computations for the distribution of X_q , with $q \in \llbracket 1, 8 \rrbracket$, $M = 500$ and $k = 25$, and Figure 3 shows the very good agreement between theory and experiment.

V. TALLY-COMBINING ALGORITHMS

In this section we formalise how a final result can be obtained from the parallel OV-Net protocol. It is practical at this point to use vector notations.

We make the assumptions that voters are consistent, i.e., that they make the same choice across all the voting sessions in which they participate². We denote v_i the choice of voter i , and collect this (unknown) information into a vector $\vec{v} =$

²This makes our analysis simpler, but in practice a voter casting inconsistent votes simply weakens his own position.

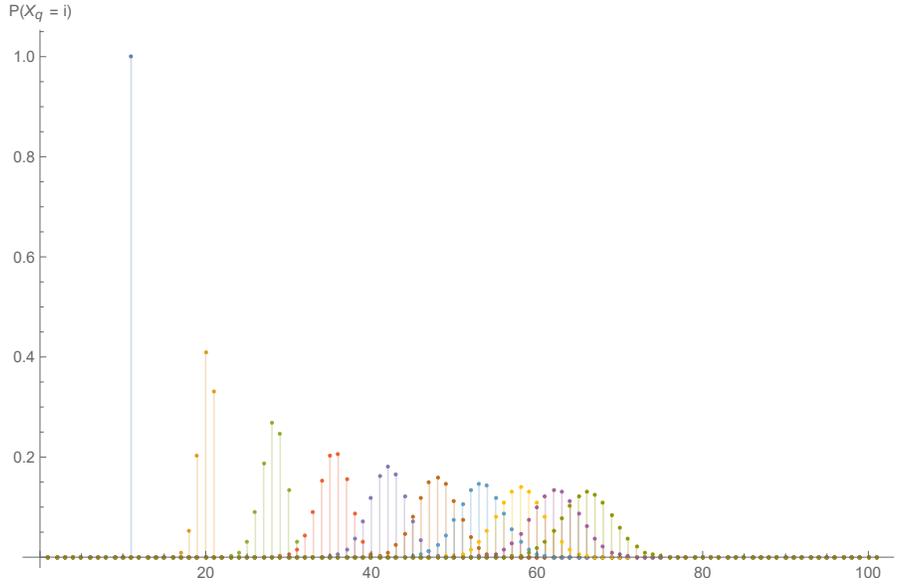


Fig. 2. PDF of X_q for $q \in \llbracket 1, 8 \rrbracket$

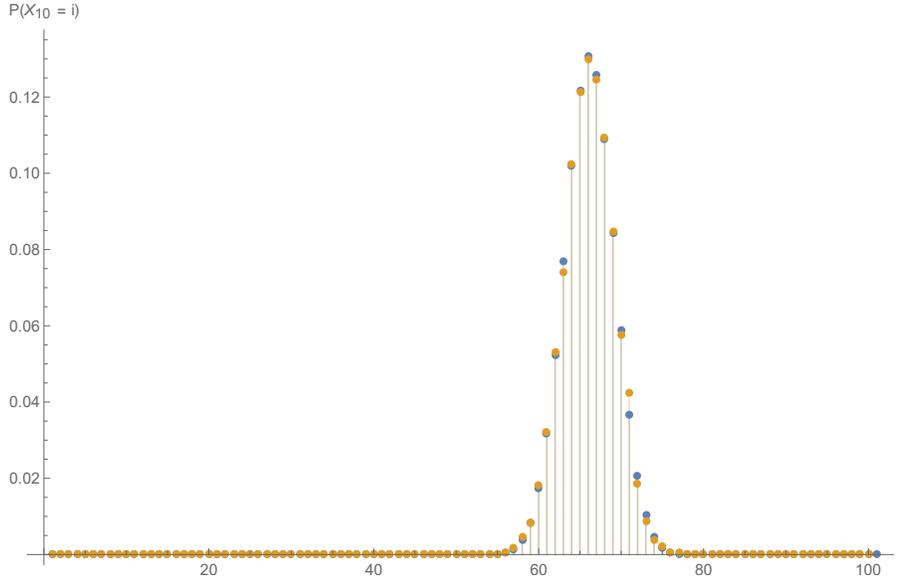


Fig. 3. Theoretical results (blue) vs experimental values in Python (orange).

(v_1, \dots, v_n) . If the vote went through with no incident, we would obtain the final tally

$$V = \sum_{i=1}^n v_i = \vec{v} \cdot \vec{1}.$$

When a voter drops out, all the sessions in which he participated are discarded. Let $0 < \mu \leq M$ be the number of remaining sessions and for each session $j \in \{1, \dots, \mu\}$ let $s_{j,i}$ be the number of times that voter i participated in session j ; hence $s_{j,i}$ can take values in $\{0, 1\}$ with the minimum value meaning that voter i did not partake in session j , and the maximum value indicating that they voted during session j .

The tally for session j is therefore

$$t_j := \sum_{i=1}^n s_{j,i} v_i = \vec{v} \cdot \vec{s}_j \quad \text{where } \vec{s}_j := (s_{j,1}, \dots, s_{j,n}).$$

By definition, $s_{j,i} = 0$ if voter i dropped out, and \vec{s}_j is non-zero (otherwise $\mu = 0$). At the end of the procedure, the following information is public knowledge:

$$\vec{T} := (t_1, \dots, t_\mu) \quad \vec{S} := (\vec{s}_1, \dots, \vec{s}_\mu)$$

The question is now: given (\vec{S}, \vec{T}) , and the parameters $\text{pp} = (n, k, M, \mu)$ how well can we approximate V ? To answer this question we need a precise definition of the error.

Definition 1 (Average- and worst-case error). Let \mathcal{A} be an algorithm taking as input \vec{S}, \vec{T} and (implicitly) pp, and returning a real number. We refer to \mathcal{A} as a tally-combining algorithm, and we write $\delta(\vec{v}, \vec{S}) := V - \mathcal{A}(\vec{S}, \vec{T})$ the tallying error.

Since δ depends on a choice of \vec{v} , which is not public information, and since \vec{S} is a collection of randomly chosen selections, it is more meaningful to consider the average error:

$$\pi_{\text{avg}}^{\mathcal{A}} := \mathbb{E}_{\vec{v}, \vec{S}}[\delta(\vec{v}, \vec{S})],$$

where \vec{v} and \vec{S} span all their possible values.

While \mathcal{A} may give results that are close to V on average, there may be corner cases in which the predicted value wanders substantially away from V ; this phenomenon is controlled by the worst-case error:

$$\pi_{\text{wc}}^{\mathcal{A}} := \max_{\vec{v}, \vec{S}} |\delta(\vec{v}, \vec{S})|,$$

where again \vec{v} and \vec{S} span all their possible values.

A simple tally-combining algorithm is given by averaging the tallies and rescaling to account for lost sessions, i.e.

$$\mathcal{A}_{\text{naive}}(-, T) = \frac{M}{\mu k} (\vec{1} \cdot \vec{T})$$

(we must divide by k since each voter casts k votes).

Lemma 4. The naïve tally-combining algorithm gives:

$$\pi_{\text{avg}}^{\text{naive}} = 0 \quad \pi_{\text{wc}}^{\text{naive}} = k \left(\frac{M+1}{\mu} - 1 \right)$$

Proof. Let V' be the result of the naïve tally-combining algorithm, i.e.

$$\begin{aligned} V' &= \frac{M}{\mu k} (\vec{1} \cdot \vec{T}) = \frac{M}{\mu k} \sum_{j=1}^{\mu} t_j = \frac{M}{\mu k} \sum_{j=1}^{\mu} \sum_{i=1}^n s_{j,i} v_i \\ &= \frac{M}{\mu k} \sum_{i=1}^n \sum_{j=1}^{\mu} s_{j,i} v_i = \sum_{i=1}^n v_i \underbrace{\left(\frac{M}{\mu k} \sum_{j=1}^{\mu} s_{j,i} \right)}_{\beta_i} = \vec{v} \cdot \vec{\beta} \end{aligned}$$

where $\vec{\beta} = (\beta_1, \dots, \beta_n)$. We now make use of the fact that \vec{v} and \vec{s} are independent and of the linearity of \mathbb{E} to compute the average error:

$$\begin{aligned} \pi_{\text{avg}}^{\text{naive}} &= \mathbb{E}_{\vec{v}, \vec{s}}[V' - V] = \mathbb{E}_{\vec{v}, \vec{s}}[\vec{v} \cdot (\vec{\beta} - \vec{1})] = \frac{1}{2} \vec{1} \cdot \mathbb{E}_{\vec{s}}[\vec{\beta} - \vec{1}] \\ &= -\frac{n}{2} + \frac{1}{2} \sum_{i=1}^n \mathbb{E}_{\vec{s}}[\beta_i] = -\frac{n}{2} + \frac{1}{2} \frac{M}{\mu k} \sum_{i=1}^n \mathbb{E}_{\vec{s}} \left[\sum_{j=1}^{\mu} s_{j,i} \right] \\ &= -\frac{n}{2} + \frac{1}{2} \frac{M}{\mu k} \sum_{i=1}^n \sum_{j=1}^{\mu} \mathbb{E}_{\vec{s}}[s_{j,i}] \\ &= -\frac{n}{2} + \frac{1}{2} \frac{M}{\mu k} \sum_{j=1}^{\mu} \mathbb{E}_{\vec{s}} \left[\sum_{i=1}^n s_{j,i} \right] \\ &= -\frac{n}{2} + \frac{1}{2} \frac{M}{\mu k} \sum_{j=1}^{\mu} \mathbb{E}_{\vec{s}} [\vec{1} \cdot \vec{s}_j] \end{aligned}$$

Since every voter participates in k sessions, on average each session has nk/M participants. Thus $\pi_{\text{avg}}^{\text{naive}} = -n/2 + n/2 = 0$.

We have $\pi_{\text{wc}}^{\text{naive}} = \max_{\vec{v}, \vec{s}} |V' - V| = \max_{\vec{v}, \vec{s}} |\vec{v} \cdot (\vec{\beta} - \vec{1})|$. Recall that we have

$$\beta_i - 1 = -1 + \frac{M}{\mu k} \sum_{j=1}^{\mu} s_{j,i} = \frac{M}{\mu k} \sum_{j=1}^{\mu} \left(s_{j,i} - \frac{k}{M} \right)$$

Fix i ; voter i casts a total of k votes which means that at most k values among $s_{1,i}, \dots, s_{\mu,i}$ are equal to 1. Since we are aiming at the maximum value we can assume that all k votes are present and without loss of generality that they happen during the first k sessions, hence

$$\begin{aligned} \beta_i - 1 &= \frac{M}{\mu k} \left(\sum_{j=1}^k \left(1 - \frac{k}{M} \right) - \sum_{j=k+1}^{\mu} \frac{k}{M} \right) \\ &= \frac{M}{\mu k} \left(k \left(1 - \frac{k}{M} \right) - (\mu - k - 1) \frac{k}{M} \right) \\ &= \frac{M+1}{\mu} - 1 \end{aligned}$$

which is positive since $\mu \leq M$. Therefore the vector \vec{v} achieving the maximum value has k coordinates (the maximum amount) equal to 1, all the other equal to 0, and we get $\pi_{\text{wc}}^{\text{naive}} = k \left(\frac{M+1}{\mu} - 1 \right)$. \square

Figure 4 shows how increasing either μ or k can reduce the worst-case error.

More generally, let $\vec{x} = (x_1, \dots, x_{\mu})$ be a vector of real coefficients, and define the *weighed tally-combining algorithm* $\mathcal{A}_{\vec{x}}(T) = \vec{x} \cdot \vec{T}$, which gives the result

$$V_{\vec{x}} = \vec{x} \cdot \vec{T} = \vec{v} \cdot \left(\sum_{j=1}^{\mu} x_j \vec{s}_j \right) = \vec{v} \cdot \vec{\beta}_{\vec{x}}$$

How do we choose \vec{x} ? The following result partially answers this question

Theorem 1. A sufficient condition for the bias of $\mathcal{A}_{\vec{x}}$ to be zero in average is $\vec{1} \cdot (\vec{1} - \vec{w}) = 0$ where $\vec{w} = x_1 \vec{s}_1 + \dots + x_{\mu} \vec{s}_{\mu}$. Furthermore, under these conditions, standard deviation is proportional to $\|\vec{1} - \vec{w}\|_2^2$.

Proof. Suppose $V = V_0$ for some fixed V_0 . Because V does not change by permuting the votes of the voters, adding all the \vec{v} 's that give V_0 , we get a multiple of $\vec{1}$. Therefore, $E[\vec{v}|V = V_0] = \frac{V_0}{n} \vec{1}$. Therefore

$$\begin{aligned} E[V - V_{\vec{x}}|V = V_0] &= E[\vec{v} \cdot (\vec{1} - \vec{w})|V = V_0] \\ &= E[\vec{v}|V = V_0] \cdot (\vec{1} - \vec{w}) \\ &= \frac{V_0}{n} \vec{1} \cdot (\vec{1} - \vec{w}). \end{aligned}$$

Thus, if $\vec{1} \cdot (\vec{1} - \vec{w}) = 0$ then the bias is zero.

We now look at the variance: for such \vec{w} , we have $E[(V - V_{\vec{x}})^2|V = V_0] = c_{V_0} \|\vec{1} - \vec{w}\|^2$ for some c_{V_0} integer.

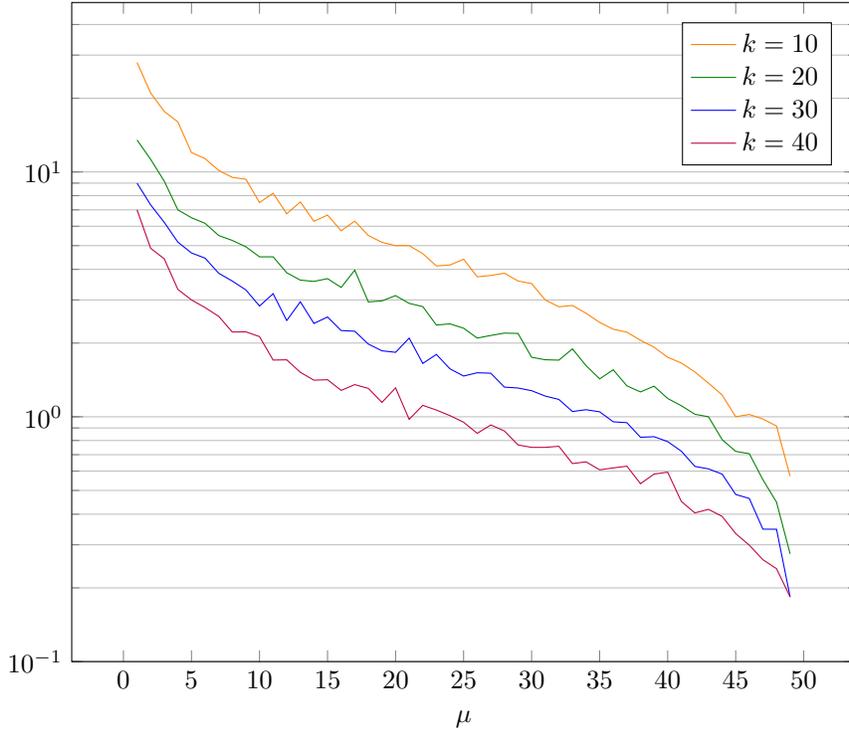


Fig. 4. Naïve tally-combining algorithm’s worst-case deviation from intended vote. Simulation results for $M = 50$, $k \in \{1, 10, 20, 30, 40\}$, $n = 10$ over 10^5 runs per value of μ . An error smaller than 0.5 means that we recover the exact result by rounding to the nearest integer.

- When $V_0 = 0$, the statement is trivial as there is only one such \vec{v} , so there is no variance.
- Suppose that $V = 1$, that is \vec{v} has the form $\{0, \dots, 0, 1, 0, \dots, 0\}$. Hence,

$$E[(V - V_{\bar{x}})^2 | V = 1] = \sum_{j=1}^n (1 - w_j)^2 = |1 - \vec{w}|^2.$$

- Now suppose that $V = 2$, that is \vec{v} has the form $\{0, \dots, 0, 1, 0, \dots, 0, 1, 0, \dots, 0\}$. In this case,

$$\begin{aligned} E[(V - V_{\bar{x}})^2 | V = 2] &= \sum_{i,j \in \{1, \dots, n\}, i \neq j} ((1 - w_i) + (1 - w_j))^2 \\ &= \sum_{i,j \in \{1, \dots, n\}, i \neq j} ((1 - w_i)^2 + 2(1 - w_i)(1 - w_j) + (1 - w_j)^2). \end{aligned}$$

Note that in the total sum, the kind of terms $2(1 - w_i)(1 - w_j)$ for some fixed i differing and j appear only once. On the other hand, the kind of terms of the form $(1 - w_i)^2$ for some i also appear the same number of times, namely as many times as 2 elements can be chosen from the n such that one of the 2 is i , which is $n - 1$. This means that the terms of the kind $(1 - w_i)^2$ appear $n - 2$ times more than those of the kind $2(1 - w_i)(1 - w_j)$. Let $c_2 := n - 2$. Then

$$\begin{aligned} E[(V - V_{\bar{x}})^2 | V = 2] &- (n - 2) \sum_i (1 - w_i)^2 \\ &= \sum_{i,j \in \{1, \dots, n\}} (1 - w_i)(1 - w_j) \end{aligned}$$

This however is 0, because

$$\sum_{i \in \{1, \dots, n\}} (1 - w_i) = \vec{1} \cdot (\vec{1} - \vec{w}) = 0$$

So we obtain

$$E[(V - V_{\bar{x}})^2 | V = 2] = (n - 2) \sum_i (1 - w_i)^2 = (n - 2) |1 - \vec{w}|^2$$

The same idea works for any $V_0 = m$: In this case,

$$\begin{aligned} E[(V - V_{\bar{x}})^2 | V = m] &= \sum_{i_1, \dots, i_m \in \{1, \dots, n\}, i_j \neq i_i} \left(\sum_{j=1}^m (1 - w_{i_j}) \right)^2. \end{aligned}$$

Again, once we carry out the squaring, in the total sum, the kind of terms $2(1 - w_i)(1 - w_j)$ for any fixed i differing and j appear exactly the same times, namely the number of times m elements can be chosen from n such that i and j are included. Let c_m'' denote this number. On the other hand, the kind of terms of the form $(1 - w_i)^2$ for some i also appear the same number of times, namely as many times as m elements can be chosen from the n such that i is among them. Let c_m' denote this number. Let $c_m := c_m' - c_m''$. Then the same as before,

$$E[(V - V_{\bar{x}})^2 | V = m] - c_m \sum_i (1 - w_i)^2 = 0$$

because $\sum_{i \in \{1, \dots, n\}} (1 - w_i) = \vec{1} \cdot (\vec{1} - \vec{w}) = 0$. So we obtain

$$E[(V - V_{\vec{x}})^2 | V = m] = c_m \sum_i^n (1 - w_i)^2 = c_m |1 - \vec{w}|^2.$$

□

If \vec{S} spans \mathbb{R}^n , then by definition of a generating family we can find $\{x_1, \dots, x_\mu\}$ such that $\vec{w} = \vec{1}$.³ Concretely, we can construct an orthonormal basis of \mathbb{R}^n from vectors of \vec{S} and project $\vec{1}$ onto each coordinate. We dub this method of computing \vec{x} the *minimum variance tally-combining algorithm* (MV, Table I). When \vec{S} span \mathbb{R}^n , the MV algorithm gives an exact result (zero bias and variance).

Input: $\vec{S} = \{\vec{s}_j\}, \vec{T}, \mu, n$

Output: $V_{\vec{x}}, \vec{x}, \vec{w}$

- 1) $Z \leftarrow \emptyset$
- 2) For each $\vec{s}_j \in \vec{S}$, if \vec{s}_j is linearly independent from Z , $Z \leftarrow Z \cup \vec{s}_j$
- 3) $\tilde{Z} \leftarrow \text{GramSchmidtOrthogonalisation}(Z)$
- 4) For each \tilde{z}_j , let $\hat{x}_j \leftarrow \vec{1} \cdot \tilde{z}_j$
- 5) $\vec{w} \leftarrow \sum_j \hat{x}_j \cdot \tilde{z}_j$
- 6) $M \leftarrow (\tilde{z}_j \cdot \tilde{z}_\ell)_{j, \ell}$
- 7) $\vec{x} \leftarrow (M^\top)^{-1} \cdot \vec{w}$
- 8) $V_x \leftarrow \vec{x} \cdot \vec{T}$
- 9) Return V_x, \vec{x}, \vec{w}

TABLE I

ALGORITHM FOR MINIMUM VARIANCE TALLY COMBINING (MV).

However, when \vec{S} does not span \mathbb{R}^n , the MV algorithm can only find a vector \vec{w} close to $\vec{1}$, namely the closest such vector in terms of Euclidean distance that can be expressed in terms of vectors in \vec{S} . This is still the solution resulting in the smallest variance, but no longer the solution with the least bias!

This leads us to consider the following approach: we can construct tally-combining algorithms that guarantee zero bias, and select amongst these an algorithm that minimizes variance. Indeed, the constraint $\vec{1} \cdot (\vec{1} - \vec{w}) = 0$ can be guaranteed by determining x_1 as a linear function of other variables⁴. It remains to minimize $\|\vec{1} - \vec{w}\|_2^2$ which is simply a quadratic form in $\mu - 1$ variables. Therefore its minimum is easy to find as it amounts to solving a linear system in $\mu - 1$ rational variables. We call the corresponding algorithm the *zero-bias minimum variance tally-combining algorithm* (ZBMV, Table II). In table II, “symbolic expression” refers to the notion that x_1, \dots, x_μ are not evaluated but are symbols to be manipulated formally.

A. Comparing tally-combining algorithms

Let’s consider a toy example to illustrate how the three discussed tally-combining algorithms compare. Throughout

³The average value of μ such that \vec{S} spans \mathbb{R}^n is $\sum_{k=1}^n \frac{2^k}{2^k - 1}$. See [15] for more precise results.

⁴There is nothing special about \vec{s}_1 , any other vector of \vec{S} can be used. Note that $\vec{1} \cdot \vec{s}_1 \neq 0$.

Input: $\vec{S} = \{\vec{s}_j\}, \vec{T}, \mu, n$

Output: $V_{\vec{x}}, \vec{x}$

- 1) Let x_1 be the symbolic expression $\frac{1}{\vec{1} \cdot \vec{s}_1} \left(n - \sum_{j=2}^\mu x_j (\vec{1} \cdot \vec{s}_j) \right)$
- 2) Let D be the symbolic expression $\|\vec{1} - \sum_{j=1}^\mu x_j \vec{s}_j\|_2^2$
- 3) $(x_2^*, \dots, x_\mu^*) \leftarrow$ solutions of the linear system $\nabla D = 0$
- 4) $x_1^* \leftarrow \frac{1}{\vec{1} \cdot \vec{s}_1} \left(n - \sum_{j=2}^\mu x_j^* (\vec{1} \cdot \vec{s}_j) \right)$
- 5) $\vec{x} \leftarrow (x_1^*, \dots, x_\mu^*)$
- 6) $V_x \leftarrow \vec{x} \cdot \vec{T}$
- 7) Return V_x, \vec{x}

TABLE II

ALGORITHM FOR ZERO-BIAS MINIMUM VARIANCE TALLY COMBINING.

this section, we take $n = 4, M = 6, \mu = 3, k = 3$ and $\vec{s}_1 = (1, 1, 1, 0), \vec{s}_2 = (1, 1, 0, 0), \vec{s}_3 = (0, 1, 0, 1)$ and $\vec{T} = (1, 0, 0)$.⁵ The results are summarized in Table III.

Tally-combining algorithm	Bias	Variance	Tally
	$\vec{1} \cdot (\vec{1} - \vec{w})$	$\ \vec{1} - \vec{w}\ _2^2$	$\vec{x} \cdot \vec{T}$
Naïve algorithm	-2/3	4/3	2/3
ZBMV	0	5/7	6/7
MV	1/3	1/3	1

TABLE III

COMPARISON BETWEEN TALLY-COMBINING ALGORITHMS ON THE TOY EXAMPLE.

Algorithm 1 (Zero-bias minimum variance). *We can express x_1 in terms of x_2 and x_3 to ensure zero bias:*

$$x_1 = \frac{1}{\vec{1} \cdot \vec{s}_1} (n - x_2 (\vec{1} \cdot \vec{s}_2) - x_3 (\vec{1} \cdot \vec{s}_3)) = \frac{1}{3} (4 - 2x_2 - 2x_3).$$

We are left to determine x_2 and x_3 , which we choose to minimize the distance of $\vec{w} = x_1 \vec{s}_1 + \dots + x_3 \vec{s}_3$ to $\vec{1}$, i.e. the quantity

$$\begin{aligned} \|\vec{1} - \vec{w}\|_2^2 &= \sum_{i=1}^n (1 - w_i)^2 \\ &= (1 - x_1 - x_2)^2 + (1 - x_1 - x_2 - x_3)^2 \\ &\quad + (1 - x_1)^2 + (1 - x_3)^2 \\ &= \frac{1}{3} (4 + 5x_2^2 + 2x_2(x_3 - 3) + 3x_3^2 - 2x_3) \end{aligned}$$

This achieves its global minimum value of 5/7 at $x_2^ = 4/7$ and $x_3^* = 1/7$. Therefore, we have:*

$$\vec{x} = \frac{1}{7} (6, 4, 1).$$

In particular, $\vec{w} = x_1^ \vec{s}_1 + \dots + x_3^* \vec{s}_3 = \frac{1}{7} (10, 11, 6, 1)$ (note that computing this vector is not necessary for the algorithm).*

⁵Note that in this example, knowing the tallies t_1 and t_2 reveals one participant’s vote. This privacy issue is addressed later in the paper.

Algorithm 2 (Minimum variance). We begin by computing an orthonormal basis \vec{Z} from \vec{S} :

$$\vec{z}_1 = \frac{1}{\sqrt{3}}(1, 1, 0, 0) \quad \vec{z}_2 = \left(\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, -\sqrt{\frac{2}{3}}, 0 \right)$$

$$\vec{z}_3 = \left(-\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, 0, \sqrt{\frac{2}{3}} \right)$$

which gives $\hat{x}_1 = \sqrt{3}$, $\hat{x}_2 = 0$, $\hat{x}_3 = \sqrt{2/3}$, from which we get $\vec{w} = \frac{1}{3}(2, 4, 3, 2)$ and finally

$$\vec{x} = \left(1, -\frac{1}{3}, \frac{2}{3} \right).$$

As expected this tally-combining algorithm has smaller variance (since $\|\vec{1} - \vec{w}\|_2^2 = 1/3$), compared with the ZBMV algorithm in of Algorithm 1, but its bias is not guaranteed to be zero (since $\vec{1} \cdot (\vec{1} - \vec{w}) = 1/3$).

Algorithm 3 (Naïve tally combining). Let's use the naïve tally-combining algorithm, i.e.,

$$\vec{x} = \frac{M}{\mu k} \vec{1}.$$

We assume here that $M = 6$, $\mu = 3$ and $k = 3$ so that $\vec{x} = \frac{2}{3}\vec{1}$, yielding $\vec{w} = (\frac{4}{3}, 2, \frac{2}{3}, \frac{2}{3})$. The bias for this algorithm is $-2/3$, however this algorithm has larger variance than the other two, since $\|1 - \vec{w}\|_2^2 = 4/3$.

VI. PRIVACY OF PARALLEL OV-NET

In this section we investigate the decrease in privacy which we can expect due to the multiple parallel elections which can be tallied individually, thus giving the adversary extra information. As an example, let us consider a simple yes/no referendum. If all voters happen to vote the same, we have of course lost privacy. However, the probability of this might be small. However, if we split the voters into two elections, the probability is roughly the square root of the old probability, i.e. much higher.

Recall that M is the number of the parallel and independent elections, n is the total number of voters and k is the number of elections that each voter has randomly chosen to participate in. We denote by M_i the set of voters who participated in election i and we consider that the elections are enumerated from 1 to M . Let $\text{Res}(M_i)$ be the random variable that gives the number of 'Yes' votes in the set M_i . We recall also that Y_i is the random variable that gives the number of voters in the set M_i .

A. Definitions and Assumptions

To quantify privacy, we use the δ -privacy definition for voting from [16] which assumes that, besides the voting elements of a voting protocol, there exists an additional party called an observer O , who can observe publicly available information. Moreover, we assume that among the n honest voters, there exists a voter V_{obs} who is under observation. For the sake of clarity, V_{obs} will refer at the same time to the voter under observation and to its vote.

Definition 2. Let P be a voting protocol and V_{obs} be the voter under observation. We say that P achieves δ -privacy if the difference between the probabilities

$$\mathbb{P}[(\pi_O || \pi_{V_{\text{obs}}}(\text{Yes}) || \pi_v)^{(\ell)} \rightarrow 1]$$

and

$$\mathbb{P}[(\pi_O || \pi_{V_{\text{obs}}}(\text{No}) || \pi_v)^{(\ell)} \rightarrow 1]$$

is δ -bounded as a function of the security parameter ℓ , where π_O , $\pi_{V_{\text{obs}}}$ and π_v are respectively the programs run by the observer O , the voter under observation V_{obs} and all the honest voters v (clearly without V_{obs}).

To calculate the privacy we use the following result from [16]

$$\delta(n) = \sum_{r \in M_{\text{Yes, No}}^*} (A_r^{\text{No}} - A_r^{\text{Yes}}) \quad (1)$$

where $M_{\text{Yes, No}}^* = \{r \in \mathbb{R} : A_r^{\text{Yes}} \leq A_r^{\text{No}}\}$, \mathbb{R} is the set of all possible election results and A_r^j denotes the probability that the choices of the honest voters yield the result r of the election given that V_{obs} 's choice is j .

We consider a referendum with n honest voters with a uniform distribution between yes and no votes. For simplicity, we will assume that nobody abstains. We also assume that no voters are corrupted. This is reasonable, since instructing corrupted voters to vote in a special way does not give further advantage compared to simply knowing the corrupted voters' votes. Moreover, we assume that at least one of the elections in which V_{obs} participated is surviving.

B. Basic Cases: $M = k = 1$ and $M \geq 1, k = 1$

The δ for a single referendum is :

$$\begin{aligned} \delta(n) &= \left(\frac{1}{2}\right)^n \frac{1}{n} \sum_{a=0}^n \binom{n}{a} |2a - n| \\ &= \begin{cases} 2^{-n} \binom{n}{\frac{n}{2}} & \text{if } n \text{ is even} \\ \frac{2^{1-n}}{n} \binom{n}{1 + \lfloor \frac{n}{2} \rfloor} \left(1 + \lfloor \frac{n}{2} \rfloor\right) & \text{Otherwise} \end{cases} \end{aligned}$$

where the first equality holds using the result from (1) and the second one using the binomial theorem.

The formula above refers to the case $M = k = 1$ where all voters had chosen to vote in the same and unique election 1. For the case $M > 1$ and $k = 1$, δ becomes a random variable and the expected value of δ of the election in which V_{obs} is participating can be defined as follows:

$$\delta_{\text{expected}}(n, M) = \sum_{n'=1}^n \mathbb{P}(Y_i' = n') \delta(n') \quad (2)$$

where Y_i' is the random variable that gives the number of voters who participated in the election i , including V_{obs} ; and

$Y'_i \sim 1 + \text{BD}(n-1, \frac{k}{M})$. Equation (2) for $k=1$ and $M > 1$ becomes:

$$\delta_{\text{expected}}(n, M) = \sum_{n'=1}^n \binom{n-1}{n'-1} \left(\frac{1}{M}\right)^{n'-1} \left(1 - \frac{1}{M}\right)^{n-n'} \delta(n')$$

Figure 5 shows that privacy is almost lost when $M \gg n$.

C. General Case

In this part we give a general formula of δ . To this end, we consider the following. Let $y = (y_1, \dots, y_M)$ be an assignment of voters such that $\text{Card}(M_i) = y_i$ for $i \in [1, M]$. We can obtain all the possible assignments of voters by respecting the condition $\sum_{i=1}^M y_i = nk$. Let $r = (r_1, \dots, r_M)$ be a possible result corresponding to the assignment y with $r_i = \text{Res}(M_i)$ for $i \in [1, M]$. r verifies the conditions $(\sum_{i=1}^M r_i) \bmod k = 0$ and $r_i \leq y_i$ for $i \in [1, M]$. Remember that $\text{Res}(M_i)$ gives the number of "Yes" votes in M_i . We have $\text{Res}(M_i) \sim \text{BD}(y_i, \frac{1}{2})$ for $i \in [1, M]$. Intuitively, δ can be expressed as the following:

$$\delta(n, M, k) = \sum_{y_1 + \dots + y_M = nk} \mathbb{P}(Y_1 = y_1, \dots, Y_M = y_M) \cdot \sum_{r \in M_{\text{Yes, No}}^*} (A_r^{\text{No}} - A_r^{\text{Yes}})$$

By definition of A_r^j we have

$$A_r^j = \mathbb{P}(\text{Res}(M_1) = r_1, \dots, \text{Res}(M_M) = r_M / V_{\text{obs}} = j)$$

with $j \in \{\text{Yes}, \text{No}\}$.

To proceed we will introduce an additional notation. Remember that M_i denotes the voters in election i . Define Σ_k as the subsets of $\{1, \dots, M\}$ of cardinality k . For $\sigma \in \Sigma_k$ we define $M'_\sigma = \bigcap_{i \in \sigma} M_i$, i.e. the voters participating in the elections in the set σ . Note that the assignment of voters to elections is uniformly random, i.e. each voter is assigned uniformly and uniquely to a M'_σ . Also Z_σ is the random variable determining the number of voters in M'_σ .

There are $c = \binom{M}{k}$ possible M'_σ s. Suppose that σ s are enumerated from 1 to c . Let $z = (z_{\sigma_1}, \dots, z_{\sigma_c})$ be an assignment of voters such that $z_{\sigma_i} = \text{Card}(M'_{\sigma_i})$, for $(\sigma_i, i) \in \Sigma_k \times [1, c]$. All the possible assignments of voters z are obtained by respecting the condition $\sum_{\sigma_i \in \Sigma_k} z_{\sigma_i} = n$.

The variables Z_σ , $\sigma \in \Sigma_k$ correspond to the problem of putting n indistinguishable balls into c distinguishable boxes, i.e. the vector $Z = (Z_{\sigma_1}, \dots, Z_{\sigma_c})$ follows a multinomial distribution with equal parameters $p_i = 1/c$, and $\sum_{\sigma \in \Sigma} z_\sigma = n$ including V_{obs} . We can now calculate the probability for the assignment of the voters, and rewrite our formula as:

$$\delta(n, M, k) = \sum_{z_1 + \dots + z_c = n} \mathbb{P}(Z_{\sigma_1} = z_{\sigma_1}, \dots, Z_{\sigma_c} = z_{\sigma_c}) \cdot \sum_{r \in M_{\text{Yes, No}}^*} (A_r^{\text{No}} - A_r^{\text{Yes}})$$

Let $r' = (r'_{\sigma_1}, \dots, r'_{\sigma_c})$ such that $r'_{\sigma_i} = \text{Res}(M'_{\sigma_i})$ for $(\sigma_i, i) \in \Sigma_k \times [1, c]$. The variables $\text{Res}(M'_{\sigma_i})$, $\sigma \in \Sigma_k$, are independent and follow the binomial distribution of parameters z_σ and $1/2$.

In the case $M = c$, which means $k = M - 1$ or $k = 1$, there is a one-to-one correspondence between the sets $(M_i)_{i \in [1, M]}$ and $(M'_\sigma)_{\sigma \in \Sigma_k}$. However this is not true in general and we have a relation between r and r' defined by the function f as follows:

$$\begin{pmatrix} r_1 \\ \vdots \\ r_M \end{pmatrix} = B \cdot \begin{pmatrix} r'_{\sigma_1} \\ \vdots \\ r'_{\sigma_c} \end{pmatrix} = f(r'_{\sigma_1}, \dots, r'_{\sigma_c})$$

where $B = (b_{i\sigma})_{\substack{1 \leq i \leq M \\ \sigma \in \Sigma_k}}$ and $b_{i\sigma} = \mathbf{1}_{i \in \sigma}$.

We can now calculate the probability A_r^v as:

$$A_r^v = \sum_{r' | r = f(r')} A_{r'}^v$$

and we have

$$A_{r'}^v = \mathbb{P}(\text{Res}(M'_{\sigma_1}) = r'_{\sigma_1}, \dots, \text{Res}(M'_{\sigma_c}) = r'_{\sigma_c} / V_{\text{obs}} = v)$$

Suppose that V_{obs} is in the subset M'_{σ_1} . It is symmetric to choose any other subset. We have:

$$A_{r'}^v = \left(\frac{1}{2}\right)^{z_{\sigma_1} - 1} \cdot h(z_{\sigma_1}, r'_{\sigma_1}) \cdot \prod_{i=2}^c \left(\frac{1}{2}\right)^{z_{\sigma_i}} \cdot \binom{z_{\sigma_i}}{r'_{\sigma_i}}$$

where

$$h(x, y) = \begin{cases} \binom{x-1}{y-1} & \text{if } v = \text{"Yes"} \\ \binom{x-1}{x-y} & \text{if } v = \text{"No"} \end{cases}$$

Remember that

$$M_{\text{Yes, No}}^* = \{r' : A_{r'}^{\text{Yes}} \leq A_{r'}^{\text{No}}\},$$

and $A_{r'}^{\text{No}} \geq A_{r'}^{\text{Yes}}$ is true when $r'_{\sigma_1} \in [0, \lfloor \frac{z_{\sigma_1}}{2} \rfloor]$. We have

$$\sum_{r'_1=0}^{\lfloor \frac{z_{\sigma_1}}{2} \rfloor} (A_{r'}^{\text{No}} - A_{r'}^{\text{Yes}}) = \frac{1}{2} \sum_{r'_1=0}^{z_{\sigma_1}} |A_{r'}^{\text{No}} - A_{r'}^{\text{Yes}}|.$$

Since V_{obs} is in M'_{σ_1} , the vector to consider is

$$Z' = (Z_{\sigma_1} - 1, Z_{\sigma_2}, \dots, Z_{\sigma_c}).$$

The formula of δ becomes

$$\begin{aligned} \delta(n, M, k) &= a_n \cdot \sum_{z_{\sigma_1}=1}^n \frac{E(z_{\sigma_1})}{z_{\sigma_1}!} \sum_{z_{\sigma_2}=0}^n \dots \sum_{z_{\sigma_c}=0}^n \frac{\delta_{\sum_{\sigma \in \Sigma} z_\sigma, n}}{z_{\sigma_2}! \dots z_{\sigma_c}!} \\ &= a_n \cdot \sum_{z=1}^n \frac{E(z)}{z!} \cdot \frac{(c-1)^{n-z}}{(n-z)!} \end{aligned}$$

with

$$a_n = \frac{(n-1)!}{c^{n-1}} \cdot \left(\frac{1}{2}\right)^n, \quad E(z) = 2^{n-z+1} \binom{z}{\lfloor \frac{z}{2} \rfloor} \cdot \left[\frac{z}{2}\right]$$

and $\delta_{i,j}$ is the Kronecker delta function.

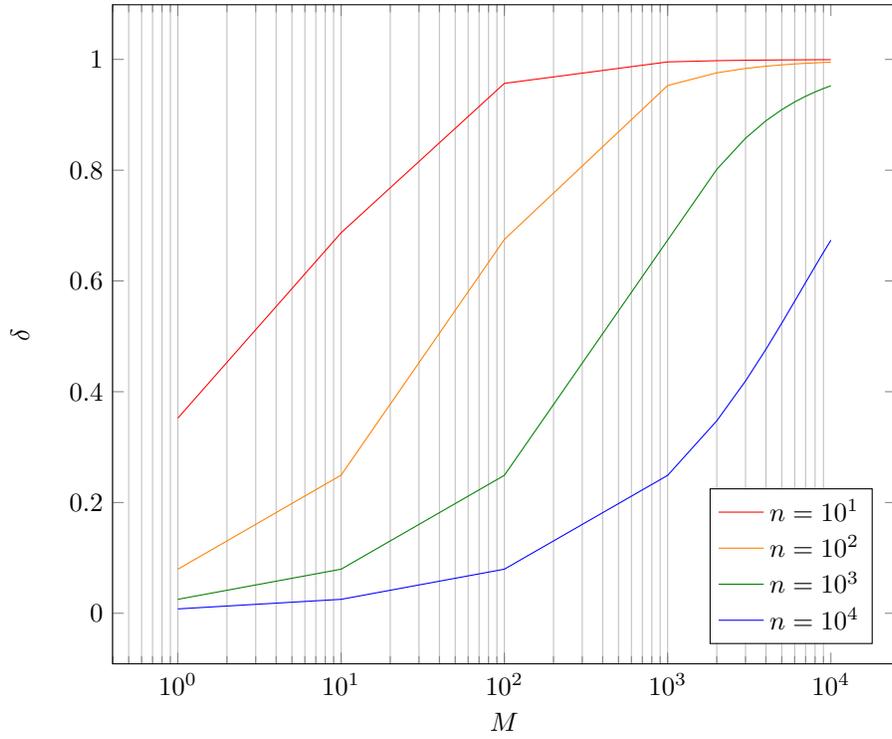


Fig. 5. The relationship between M and δ_{expected} for different values of $n = 10, 10^2, 10^3, 10^4$.

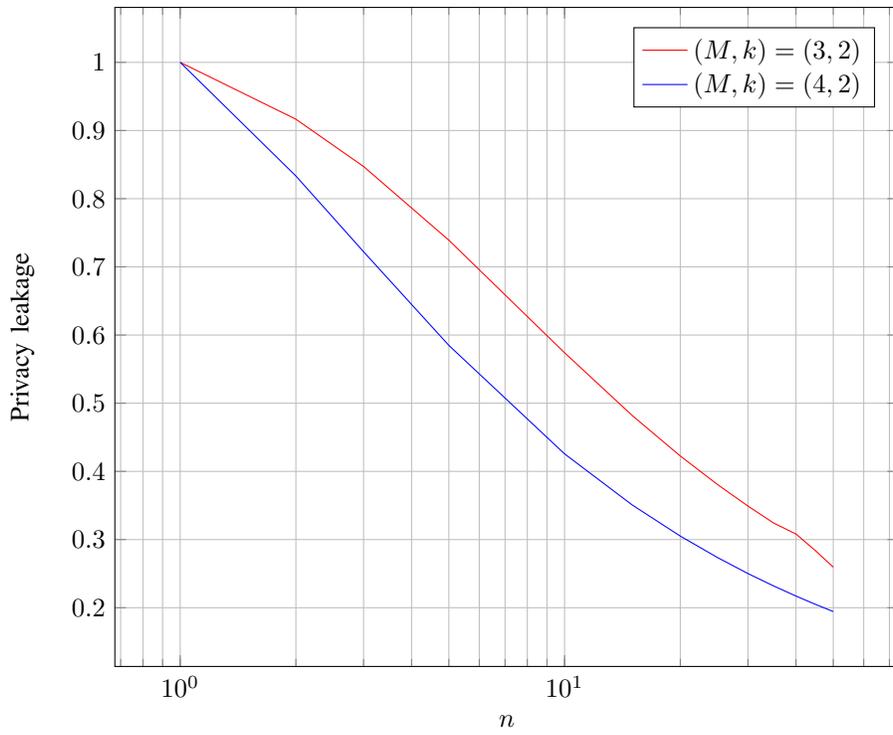


Fig. 6. Privacy leakage as function of n for the cases $(M, k) = (3, 2), (4, 2)$.

VII. CONCLUSION AND FURTHER RESEARCH

a) *Conclusion:* In this paper, we presented a new version of the protocol OV-Net which run several elections in parallel to achieve robustness against DoS failures without having to resort to time-consuming extra rounds. We computed quantitatively the increase in robustness from having M parallel elections with each voter participating in k of these, and demonstrated that robustness can be significantly improved. The improvement in time and robustness, come at a cost in terms of accuracy and privacy. We presented three different algorithms on how to optimally compute the tally using this new OV-Net version and we quantitatively measured the privacy decrease that is expected due to the multiple partial election results. The results allow the protocol initiator to choose parameters to carefully balance the wanted robustness with a controlled privacy loss and statistical loss in accuracy, as well as increased computation.

b) *Further work:* Redistribution is the process by which elections are conducted in several electoral districts. Unlike general elections, where the final result is known for the entire country only, in redistributed elections, each voter votes in their own district, results are consolidated per district and only then added up. Perhaps it is possible to confine problematic voters to a district of their own, as follows: partition the n voters into d districts of $n' = n/d$ voters, then run a vote in each of them. Then recompose the result by adding up the final tally.

This strategy confines the DoS problem to districts that do not influence each other. However, DoS tolerance is not exactly multiplied by d because each district is not allowed to exceed k unresponsive voters. In other words, tolerance is multiplied by d as long as the constraint that there are no more than k unresponsive voters per district is respected.

REFERENCES

- [1] F. Hao, P. Y. A. Ryan, and P. Zielinski, "Anonymous voting by two-round public discussion," *IET Information Security*, vol. 4, no. 2, pp. 62–67, 2010. [Online]. Available: <https://doi.org/10.1049/iet-ifs.2008.0127>
- [2] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in *Financial Cryptography and Data Security - 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers*, ser. Lecture Notes in Computer Science, A. Kiayias, Ed., vol. 10322. Springer, 2017, pp. 357–375. [Online]. Available: https://doi.org/10.1007/978-3-319-70972-7_20
- [3] M. Seifelnasr, H. S. Galal, and A. M. Youssef, "Scalable open-vote network on ethereum," in *Financial Cryptography and Data Security - FC 2020 International Workshops, AsiaUSEC, CoDeFi, VOTING, and WTSC, Kota Kinabalu, Malaysia, February 14, 2020, Revised Selected Papers*, ser. Lecture Notes in Computer Science, M. Bernhard, A. Bracciali, L. J. Camp, S. Matsuo, A. Maurushat, P. B. Rønne, and M. Sala, Eds., vol. 12063. Springer, 2020, pp. 436–450. [Online]. Available: https://doi.org/10.1007/978-3-030-54455-3_31
- [4] A. Kiayias and M. Yung, "Self-tallying elections and perfect ballot secrecy," in *Public Key Cryptography, 5th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2002, Paris, France, February 12-14, 2002, Proceedings*, ser. Lecture Notes in Computer Science, D. Naccache and P. Paillier, Eds., vol. 2274. Springer, 2002, pp. 141–158. [Online]. Available: https://doi.org/10.1007/3-540-45664-3_10
- [5] J. Groth, "Efficient maximal privacy in boardroom voting and anonymous broadcast," in *Financial Cryptography, 8th International Conference, FC 2004, Key West, FL, USA, February 9-12, 2004, Revised Papers*, ser. Lecture Notes in Computer Science, A. Juels, Ed., vol. 3110. Springer, 2004, pp. 90–104. [Online]. Available: https://doi.org/10.1007/978-3-540-27809-2_10
- [6] F. Hao and P. Zielinski, "A 2-round anonymous veto protocol," in *Security Protocols, 14th International Workshop, Cambridge, UK, March 27-29, 2006, Revised Selected Papers*, ser. Lecture Notes in Computer Science, B. Christianson, B. Crispo, J. A. Malcolm, and M. Roe, Eds., vol. 5087. Springer, 2006, pp. 202–211. [Online]. Available: https://doi.org/10.1007/978-3-642-04904-0_28
- [7] F. Hao, "A 2-round anonymous veto protocol," in *Security Protocols, 14th International Workshop, Cambridge, UK, March 27-29, 2006, Revised Selected Papers*, ser. Lecture Notes in Computer Science, B. Christianson, B. Crispo, J. A. Malcolm, and M. Roe, Eds., vol. 5087. Springer, 2006, pp. 212–214. [Online]. Available: https://doi.org/10.1007/978-3-642-04904-0_29
- [8] D. Khader, B. Smyth, P. Ryan, and F. Hao, "A fair and robust voting system by broadcast," *Lecture Notes in Informatics (LNI), Proceedings-Series of the Gesellschaft für Informatik (GI)*, pp. 285–299, 2012.
- [9] R. Giustolisi, V. Iovino, and P. B. Rønne, "On the possibility of non-interactive e-voting in the public-key setting," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 193–208.
- [10] C. Schnorr, "Efficient signature generation by smart cards," *J. Cryptol.*, vol. 4, no. 3, pp. 161–174, 1991. [Online]. Available: <https://doi.org/10.1007/BF00196725>
- [11] R. Cramer, I. Damgård, and B. Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols," in *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, ser. Lecture Notes in Computer Science, Y. Desmedt, Ed., vol. 839. Springer, 1994, pp. 174–187. [Online]. Available: https://doi.org/10.1007/3-540-48658-5_19
- [12] R. Cramer, M. K. Franklin, B. Schoenmakers, and M. Yung, "Multi-authority secret-ballot elections with linear work," in *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, ser. Lecture Notes in Computer Science, U. M. Maurer, Ed., vol. 1070. Springer, 1996, pp. 72–83. [Online]. Available: https://doi.org/10.1007/3-540-68339-9_7
- [13] O. Baudron, P. Fouque, D. Pointcheval, J. Stern, and G. Poupard, "Practical multi-candidate election system," in *Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing, PODC 2001, Newport, Rhode Island, USA, August 26-29, 2001*, A. D. Kshemkalyani and N. Shavit, Eds. ACM, 2001, pp. 274–283. [Online]. Available: <https://doi.org/10.1145/383962.384044>
- [14] C. J. Colbourn and J. H. Dinitz, *Handbook of combinatorial designs*. CRC press, 2006.
- [15] C. Cooper, A. M. Frieze, and W. Pegden, "On the rank of a random binary matrix," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, T. M. Chan, Ed. SIAM, 2019, pp. 946–955. [Online]. Available: <https://doi.org/10.1137/1.9781611975482.58>
- [16] R. Küsters, J. Liedtke, J. Müller, D. Rausch, and A. Vogt, "Ordinos: A verifiable tally-hiding remote e-voting system," *Cryptology ePrint Archive*, Tech. Rep., 2020.