The peasants are revoting, sire, and at random times

Enka Blanchard, Peter y A Ryan

The peasants are revoting, sire, and at random times

Enka Blanchard[1,2][0000−0003−1511−8864] and Peter Y. A. Ryan[3][0000−0002−1677−9034]

[1] Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines – UMR CNRS 8201, Université Polytechnique Hauts-de-France
[2] CNRS Center for Internet and Society, Paris, France
enka.blanchard@cnrs.fr, www.koliaza.com
[3] University of Luxembourg
peter.ryan@uni.lu

Abstract. Inspired by how coercion happens in practice in complex social networks where weak ties are more important than unilateral force, we propose a new mitigation mechanism based on revoting: adding an extra revoting period of random duration. Not only does this add opportunities to resist coercion, it also adds friction to the system, increasing the costs for both coerced voter and coercer, and disincentivising coercion. We investigate generalisations and variants of this mechanism in different frameworks (such as with and without shareable credentials), considering the optimal strategies for the electoral authority, voters and coercer. We also propose an implementation of the mechanism in a simple setting.

Keywords: Coercion mitigation · Revoting · E2E verifiable voting

1 Introduction

Freedom from coercion is one of the fundamental requirements for free and fair elections. It is a stronger property than vote privacy[1] and comes with a stronger attacker model: rather than passively observe the unfolding of the election, the attacker may actively interfere with the voters and the system. As such, both coercion and counter-measures against it have long been studied in the election systems literature [9]. For in-person voting, coercion is typically countered by ensuring that voters cannot be observed as they make their selection and cast their ballot — which is sometimes opposed to freedom of expression [12]. Securing remote voting, whether by postal vote or through the internet, remains an open

[0] Authors are listed in alphabetical order as they made equivalent contributions to the research.

[1] Although both are often discussed in similar ways, we will focus on coercion rather than vote buying, as the latter often assumes a weaker adversary with different strategies: only the coercer can use negative as well as positive incentives [6].

problem as there is no practical way to enforce such isolation at the time of casting, making coercion a serious threat.

As with most security properties, resistance to coercion is subtle and comes in many flavours. In its strongest form it requires that the voter always be able to cast their intended vote without the coercer detecting this, whatever interactions, observations or instructions the coercer performs (before, during or after the execution of the voting protocol). This is of course extremely difficult to achieve, especially if E2E verifiability is also required. Furthermore, to be possible at all, the voter must have some time interval and channel over which they can interact with the voting system unobserved by the coercer. These hypotheses have been relaxed in different ways, such as allowing the voter to nullify their vote instead, as in VoteXX [2], potentially even implementing *coercion evidence* in which this nullification leaves observable traces, as in Caveat Coercitor [8]. Others have also proposed forms of *coercion mitigation*, as in Selene and Hyperion tracker-based schemes, in which a coerced voter can disobey instructions, with a small but non-zero chance of getting caught — by a tracker collision attack [15].

One option that has been used in practice is to implement revoting, in which voters can cast a new ballot replacing their previous vote — especially when a vote cast in a polling place cancels any online vote cast, as in Estonia [14]. More cryptographically involved counter-measures have also been proposed, with JCJ presenting a seminal approach in which each voter can vote either with real credentials or fake credentials which are indistinguishable to the coercer [13], although full coercion resistance wasn't achieved in the original iteration [3]. Tallying is performed such that, after anonymisation, votes with valid credentials are counted and invalid credentials are discarded in a way that is undetectable.

A common element among many of the schemes proposed is the underlying image of the coercer: an adversary with immense power over the voter, such as someone threatening them with physical violence if they do not vote for a given candidate [1]. This is sometimes reasonable, but does not always correspond to the reality of coercion in anocratic countries. Instead of a voter casting a vote at gunpoint, one can picture an employer taking their workers to vote as a group and treating them to drinks afterwards, making it a social event with low costs for all participants and limited social friction [5, 10, 6]. The voters cannot be sure of whether they are watched when they cast their vote and might take the easy way out and obey orders — especially if they believe that the system is corrupt enough that their vote would not have an impact. Having all the voters cast their vote and stay with the coercer until the polls close — under the guise of a social event — is also a good way to prevent them from revoting at the last minute.

This reliance not on unilateral force but on complex social ties opens new ways to think about coercion mitigation. One option is to increase this social friction, making the coercion costlier for all participants. Although an all-powerful coercer could just shunt this cost entirely onto the voters, the same cannot be said of a social coercer. This is all the more applicable if said coercer is just one element in a larger patronage system, in which case some of the costs would be

spread onto the entire coercion machine, reducing its efficiency. This leads us to propose two contributions in this paper:

- A simple mechanism to mitigate coercion that should be adaptable to many voting systems: having an extra voting period of random duration.
- An investigation of this mechanism in different frameworks (with and without shareable credentials) and variants, considering the optimal strategies for the electoral authority (EA), voters and coercer.

In all the different cases we investigate, we show that the mechanism makes the coercer's task harder by forcing them to invest more time and resources — sometimes in an unbounded fashion if they want to guarantee getting the coerced vote. Moreover, we show that it adds opportunities to detect coercion through network analysis, depending on the coercer's behaviour. Beyond the immediate impact, a second benefit is that it serves to disincentivise would-be coercers.

This paper starts by discussing the central idea before looking at its impact on coercer, voter, and EA strategies, first with non-shareable credentials and then with shareable credentials. We then discuss how it could be implemented in practice before discussing political considerations — especially whether such a mechanism could be anti-democratic — as well as potential extensions.

2 Central idea

The central mechanism behind the different variants we introduce is the blurring of the cutoff point for casting votes in different ways (e.g., with secret individual deadlines or a limit to the number of revotes). In its simplest variant, instead of having the possibility to cast a new vote (replacing the previous one) until the end of the election, each voter is given a secret additional time during which they can continue revoting at will.

More formally, we consider a system where all voters can cast votes between the start of the original voting period, denoted T_0, and its end, denoted T_1. Between T_1 and the end of the extension period T_2, each voter will have a secret random cutoff point $t_i < T_2$, which is not known by the voter. They can continue casting votes at will but any vote cast after t_i will be discarded. Uncoerced voters should of course cast their vote before T_1. A coerced voter will however have more opportunities to vote, which creates a game of strategy between the voter, the coercer and the EA. We will say that the voter (respectively, the coercer) "wins" if they cast the vote that is eventually the one recorded and counted. Of course, if the coercer can afford to use unilateral force without caring for the cost or consequences, they can win despite any EA strategy, so we will consider coercers with limited capabilities[2].

[2] Even in the non-shareable credential setting detailed further down where a coercer can't guarantee a probability 1 of winning, this assumes that the coercer does not have the power to fully prevent the voter from casting any vote, which has a huge cost as it would require confiscating all the voter's devices and have them under continuous surveillance, among other elements.

To analyse these games, we will consider two main settings. The first and simplest one corresponds to the case where voting requires an uncloneable voting token, for example a physical device (such as the e-ID card used in Estonia, a banking card, a computer or a smartphone). This token should ideally be multi-purpose, making voters reluctant to give up their tokens for a protracted period[3]. The advantage of this approach is that we can assign a cost to the coercer for taking possession of the token, and this cost can be assumed to increase with the length of time they hold it. Importantly, neither voter nor coercer should be aware of when the cutoff point is. In this setting, the person holding the token at the cutoff point presumably wins, so the game becomes one of optimising one's probability of holding the token at the right moment while minimising the total cost.

Despite their advantages, uncloneable tokens introduce complexity and can come at a high cost if they are not already in use; thus they are not the only relevant model. The second setting will then focus on digital voting credentials that can be copied. This of course changes the dynamics of the revoting game as voter and coercer will be able to cast votes concurrently — thereby having an interest to cast many votes[4]. Although this removes the cost of holding the token, it introduces a new cost, corresponding to the energy and effort it takes to cast a vote. Moreover, it introduces a risk for the coercer, which is that a pattern of massive revoting would be observable and could lead to an investigation — or potentially a new election if widespread malfeasance is suspected. We will not consider this option here as integrating it formally opens the door to complex denial of service attacks — but the question of conflict-resolution mechanisms should be addressed in any implementation.

3 Non-shareable credentials

In the first model that we consider, we make the following assumptions on the voting system:

- Casting a vote — whether online or in person — requires an *uncloneable physical token*.
- Each voter can vote as many times as they want, with each subsequent vote replacing the previous one (so that only the last vote before the deadline counts).

[3] To compare this with a practical contemporary scenario, a voter might be willing to vote in front of their employer then part with their phone for an evening of partying on company budget — while being watched by said employer. The same voter might be much more reluctant to part with their phone for a week. Although this cost is on the end-user, it partially flows up the chain as an employer contemplating the prospect of spending a week surrounded by a team of disgruntled workers would need stronger incentives to participate in a coercion machine.

[4] In real elections with revoting, although few voters cast multiple ballots, the maximum revotes can be upwards of 500 [11].

– The logs of who voted and when are either non-existent or considered not accessible by would-be coercers[5].

Our proposal then consists in the following:

– There is a common voting period in which the system above works as announced, with a public starting date T_0 and a public deadline T_1.
– A secret time t_i is chosen at random and independently for each voter V_i, between T_1 and a second public deadline T_2.
– Any vote cast by V_i between T_1 and t_i is recorded (and erases previous votes), and any vote cast after t_i is discarded.
– Votes are counted after T_2.

Before looking at costs and optimal strategies, we need a preliminary remark. If the voter is not being coerced, we can assume that they will vote before the start of the extended voting period T_1 (they could forget about the deadline and still manage to vote afterwards but this is no different from existing systems, aside from giving them a second chance). This is why having a first public deadline T_1 is important, as it guarantees an opportunity to vote on par with existing systems. Thus, the only voters we are concerned with below are the ones who are targeted by coercers.

As stated in Section 2, the proposal works by increasing costs for users (who can forward it onto coercers). To model the coercer's costs, we make the simple assumption that the cost to keep the token increases linearly over time. This means that any coercer ready to pay a high cost will keep the token for the whole duration, but others will try to optimise the probability of successfully forcing their vote to be recorded while limiting costs. The following subsection considers optimal strategies for both the coercer and the EA (which seeks to maximise coercer costs).

Strategies

Following the model above, at any point, either the voter or the coercer holds the token which allows (re)voting. It is then in the coercer's best interest to vote as soon as they acquire the token and then keep it secure without voting again. The voter has similar interests (locally) in that they should vote as soon as they get the token and are not being watched by the coercer — we henceforth assume that, for the voter, having the token implies being able to vote privately.

Let us consider the system as a game played by the coercer and the EA[6]. The coercer's actions correspond to a partition of the $[T_1; T_2]$ time period into time

[5] This is still compatible with systems based on bulletin boards as long as the information revealed is sufficiently limited. One option is also to reveal the information either at a fixed time or after a delay. Not keeping such detailed logs would of course be a stronger guarantee for voters' privacy but could go against legislations/regulations. A coercer can have limited power in practice, such as a coercer with power over local elections but unable to access a nationally centralised election administration.

[6] The voter is not considered a player here since they make no strategic choice: their only action is to vote when they have the token.

slots during which they hold the token and ones during which the voter does. Their objective is then to maximise the probability that they cast the last vote, or equivalently, that t_i falls into a time slot where they hold the token. They try to maximise this probability while minimising the total cost, which corresponds to the sum of the lengths of their time slots. The EA's actions correspond to setting a probability distribution over when the t_i happen, a distribution which we assume to be public[7]. Its objective is to maximise the coercer's costs — potentially arbitrating to optimise different segments of the cost functions if required (which does not happen with linear costs).

We will use a discrete formulation of the problem for simplicity — an equivalent measure-theoretic proof in the continuous case is available in the Appendix, for comparison with the results of Section 4. We then set the deadline to happen at midnight[8] on a random day among the n days of the extra voting period, with the probability of it being on day j being p_j. The optimal strategy for a coercer willing to pay for k days will be to take the k days with highest total probability. This is minimised when all the p_j are equal.

If it is not the case, we can sort the probabilities and set p'_k as the k-th greatest p_j. If $p'_k \geq \frac{1}{n}$, then $\sum_{j \leq k} p'_j \geq \frac{k}{n}$, and taking the most probable k days gives us a winning probability greater than $\frac{k}{n}$. If $p'_k \leq \frac{1}{n}$, we similarly get $\sum_{j>k} p'_j \leq 1-\frac{k}{n}$, thus $\sum_{j \leq k} p'_j \geq 1-\sum_{j>k} p'_j \geq \frac{k}{n}$. Thus, in both cases, the total probability will be greater if the distribution is not uniform. This shows that for any coercer budget, the EA's optimal choice is to have a uniform distribution.

If both coercer and EA act optimally, the coercer can then obtain a probability p of obtaining the final vote only by paying a cost equal to $p \times T$, where T would be the total cost for keeping the token over the whole period. However, this is only true in the case where the cost is constant per day, an assumption we discuss in section 6.

4 Shareable credentials

We can now consider a second model where the voting credentials are shareable and do not require a physical token — for example, a password could suffice. Both voter and coercer can now vote as many times as they want between T_1 and T_2.

We still follow most of the previous assumptions: each t_i is randomly assigned and only the last vote cast before t_i counts, while voting logs are not accessible to the coercer. However the costs are of a different nature. Indeed, there is no additional cost incurred by the voter being deprived of their phone or ID card. However, we can consider that each vote costs a limited amount of time to

[7] We assume this as hiding the distribution would negatively impact not only transparency but also the understandability of the system, at least for laypeople. Moreover, the assumption only reinforces the coercer's power (hence having a secret distribution would not negatively impact the results).

[8] Equivalently, we could set it at the start of every hour or at any given regular frequency. The only goal here is to discretise time.

the party casting it. Moreover, having a large amount of votes cast (especially in a given locality) would increase the chance that an anomaly gets detected, potentially triggering an investigation which the coercer should seek to avoid. Thus, the coercer would either try to minimise the total cost or try to find the best strategy that keeps costs below a constant number of votes (what they assume the detection threshold to be).

To fully model the system's behaviour, we would have to consider three actors: the voter, the coercer and the EA. We then make two simplifying assumptions, both of which are generally to the coercer's advantage. First, we assume that the voters will use a simple strategy that is not directly dependent on the coercer's actual strategy. This assumption comes from the fact that, as each coercer presumably coerces many voters, they have a stronger incentive to understand voter strategies than the other way around. Moreover, what matters is not so much whether the voter strategies are simple but whether the coercer's model of the voter follows a simple strategy. Indeed, that voter model is what governs the coercer's strategy, with a more complex one generally being disadvantageous to the coercer. One can then remark that the simplest strategy of voting at regular times can potentially be guessed by the coercer and countered by voting right afterwards, giving equal cost but a probability of winning asymptotically equal to 1. A wary voter might then introduce some randomness (even if only by not making a conscious effort to vote each day at a regular time). We then consider one of the simplest available random strategies, setting the voter's votes to be equivalent to a Poisson point process, with each vote occurring independently of all others while following a constant expected rate λ_v.

The second assumption we make is that the deadline t_i is distributed uniformly at random between T_1 and T_2 and that this information is public. This is based on two intuitions. First, having an unbalanced distribution which is publicly known gives more information to the coercer on when to focus their efforts[9]. Second, if the distribution is sufficiently lopsided, it can behave similarly to a strict deadline, with the coercer having the option of exerting control over the voters during the limited time slot.

Before looking at the strategies, we can make an observation. Unlike in the non-shareable token model, assuming t_i is uniformly distributed between T_1 and T_2, the coercer cannot guarantee that the final vote will be cast in the way they desire, and their total cost is unbounded. Indeed, even if the voter casts a single vote in the period, there will always be a non-zero probability that the voter casts a vote right between the coercer and the cutoff. Supposing the voter casts votes with parameter λ_v, one can show (following the same methods as below) that a coercer with a budget for k votes will have a probability of casting the last vote equal to at most $e^{-\lambda_v \frac{T_2-T_1}{k}}$ by casting their votes at regular intervals.

[9] Supposing that the distribution is kept secret would probably push the coercer to assume uniformity in any case, unless they manage to obtain the secret distribution in which case it would be to their advantage compared to the voters.

4.1 Unlimited votes

Let us now consider the simplest case where both voter and coercer can cast as many votes as they want until T_2, with only the last vote cast before t_i being counted. Let's suppose temporarily that the coercer also follows a Poisson point process with parameter λ_c. Using the basic properties of Poisson point processes [7], the combination of both voter and coercer can be merged into a single Poisson point process of parameter $\lambda_c + \lambda_v$. Each event in this combined process corresponds to a vote cast by either the coercer or the voter, with respective probabilities $\frac{\lambda_c}{\lambda_c+\lambda_v}$ and $\frac{\lambda_v}{\lambda_c+\lambda_v}$.

We start by observing that this strategy is not optimal for the coercer and is in fact worse than voting at regular intervals. Indeed, discretising for simplicity, if we assume that the coercer votes every $\frac{1}{\lambda_c}$, the probability that the coercer wins is equal to the probability that the voter does not cast a vote between the last vote by the coercer (corresponding to the start of the final time slot) and t_i. Integrating over the distribution of t_i in the final time slot, this becomes:

$$\lambda_c \int_0^{1/\lambda_c} e^{-\lambda_v t} dt = \frac{\lambda_c}{\lambda_v}(1 - e^{-\lambda_v/\lambda_c})$$

Using the fact that $e^x \geq 1 + x$, a simple derivation shows that this is always greater than $\frac{\lambda_c}{\lambda_c+\lambda_v}$. In fact, we will show that, given the assumptions on the voter and EA (whose strategies do not depend on the coercer's voting strategy), voting on a regular schedule is optimal for the coercer.

Although we have considered the coercer's strategy as a probabilistic distribution of voting times, we can also see it as a set of points in time fixed in advance. Indeed, even when using a probabilistic process, there should be no outside information available to the coercer (assuming the voter is careful not to get caught revoting). The coercer can then simulate the process in advance to obtain the point set and then vote at the corresponding times. Let's first show that, for a fixed λ_c number of cast votes, having them set at regular intervals is optimal.

To do so, we suppose that we have an optimal set of points which maximises the coercer's probability of casting the last vote, still assuming that the EA has a uniform distribution for the deadline t_i and that the voter follows a Poisson distribution of parameter λ_v. We can then take two adjacent time slots of potentially different lengths (determined by the timing of three consecutive votes v_1,v_2, and v_3 cast by the coercer). We can renormalise the timeframe to simplify the computation, setting $v_3 - v_1 \triangleq 1$ and $v_2 - v_1 \triangleq x$, assuming that t_i belongs to one of these time slots. The coercer then wins if the voter does not vote before t_i in the corresponding time slot, hence with a probability

$$\int_0^x e^{-\lambda_v t} dt + \int_x^1 e^{-\lambda_v (t-x)} dt = \frac{1}{\lambda_v}(2 - e^{-\lambda_v x} - e^{-\lambda_v (1-x)})$$

Differentiating the right side of the equation, we can see that its maximum is reached in $x = 0.5$, that is, when the two time slots are of equal length. As this

applies for any two adjacent time slots in the optimal distribution, it corresponds to voting at regular intervals.

This shows that, with the given assumptions, the coercer's optimal strategy manages to win with probability $\frac{\lambda_c}{\lambda_v}(1 - e^{-\lambda_v/\lambda_c})$, assuming that the coercer's fixed budget is λ_c. By concavity of $x \mapsto x - e^{-1/x}$, we can similarly show that if the coercer's expected budget is λ_c, the optimal choice is to have exactly λ_c votes rather than a random distribution of budgets with expectation λ_c.

We can now replace the assumption that the voter follows a Poisson point process, and suppose instead that they also vote according to a regular pattern of period $1/\lambda_v$. To simplify computations, let's also assume that both coercer and voter have an initial offset (smaller than their period) chosen uniformly at random. Let's additionally assume that the coercer has $\lambda_c \geq \lambda_v$; otherwise we can invert the roles of coercer and voter for the computation. Going back from the deadline t_i chosen uniformly at random, the winner is the one whose regular slot started most recently. Assuming that both periods start with a random offset, the probability of the coercer winning becomes

$$\int_0^{1/\lambda_v} \lambda_v \min(1, t \times \lambda_c) dt = 1 - \frac{\lambda_v}{2\lambda_c}$$

Conversely, the probability that the voter wins is $\frac{\lambda_v}{2\lambda_c}$, which has a natural interpretation: the voter wins with probability $1/2$ for the proportion of time corresponding to the coercer's interval, and loses automatically beyond that point.

We can finally provide a partial generalisation of the results above. Intuitively, any strategy that favours one part of the time period over the others seems suboptimal. Moreover, as neither voter nor coercer know when the voting period ends and it is uniformly distributed, a form of temporal invariance seems natural. Assuming one such kind of temporal invariance, we can prove stronger results for both voter and coercer. More precisely, we assume that the strategies of both coercer and voter are left invariant by the transformation which associates a set of voting times with the same set offset by a constant and looping back to the start[10]. This type of invariance is respected by both Poisson point processes and regular voting with a uniform offset.

Let us define $P(t)$ as the probability that the voter has voted in the last interval of length t. If the voter casts n votes, we can look at the n corresponding time slots (with the last slot circling back until the first vote), of length a_j (for $1 \leq j \leq n$). Then $P(t)$ is equal to the sum of the probabilities that t_i was in the j-th interval, multiplied by the probability that enough time has passed in said interval. Hence

$$P(t) = \sum_j \frac{a_j}{\sum_k a_k} \min\left(1, \frac{t}{a_j}\right)$$

[10] This can be formalised by looking at the actions, for all values of t', of the bijections $t+T_1 \mapsto (t+t' \mod (T_2-T_1))+T_1$ over the strategies, seen as probability measures over sets of points.

This function is piecewise linear and non-decreasing. We can then use the same proof sketch as before, assuming that the coercer has a set of points fixed in advance and supposing that two adjacent time slots are of different lengths. We then obtain that the probability of the coercer winning is:

$$\int_0^x (1 - P(\frac{t}{a_i + a_{i+1}}))dt + \int_x^1 (1 - P(\frac{t-x}{a_i + a_{i+1}}))dt$$

As $t \mapsto 1 - P(t)$ is non-increasing, its integral is concave, meaning the above sum of integrals reaches it maximum at $x = 0.5$. Finishing the proof as previously, one shows that, under the assumption of temporal invariance for the voter's strategy, the optimal strategy for the coercer is to have regular time slots. By symmetry, the same proof applies to show that the voter's optimal strategy is also regular (under symmetric assumptions on the coercer's strategy).

Before going over further considerations on the impact of such strategies (which is the subject of section 4.3), we will first complete our study of these behaviours by establishing similar results on a natural evolution of this model.

4.2 Limited votes per time period

Instead of having a single time period with unlimited votes, one can imagine multiple natural alternatives where the time is split into different time periods, with up to k votes allowed in each time period (with all votes cast after the k-th being discarded). Having a limited number of votes (even if unknown) is not compatible with having a single time period because the coercer could then cast a large number of votes right after T_1 while preventing the voters from doing similarly, and the system would not make any meaningful difference — unless the expected number of votes is so high that the system behaves as if they are unlimited.

Let us then partition the extra voting period into a discrete set of time slots (with the last one being the most important). To simplify (and because it is entirely the choice of the EA), we consider this to be done following a Poisson point process — although most of the properties below can be shown to be true no matter the distribution. We will initially consider that each time slot allows a single vote before discarding the rest.

In this case, the only vote that matters is the first vote cast in the last period that includes t_i. As long as there is at least one vote cast in that period, the system behaves exactly like in the previous model of unlimited votes (albeit with a reversed time as it is the first vote and not the last one). Using the same arguments as before, one can show that the previous equalities stay true. Moreover, if no votes are cast, then what matters is the first vote cast in the last non-empty period, which follows an equivalent behaviour. Thus, one can show that, if both coercer and voter follow a Poisson process, the probability of casting the successful vote remains $\frac{\lambda_c}{\lambda_c+\lambda_v}$ and $\frac{\lambda_v}{\lambda_c+\lambda_v}$. One can similarly show that, if both coercer and voter vote following regular patterns (with $\lambda_c > \lambda_v$), the probability of the coercer winning becomes $1 - \frac{\lambda_v}{2\lambda_c} \geq \frac{\lambda_c}{\lambda_c+\lambda_v}$.

Let us now finish by considering the case where the votes cast are counted only up to the k-th, after which any subsequent vote is discarded. In its generality, this case affects the strategies above and introduces more complexity than could be analysed here, although it remains a potentially interesting option. However, we can still make a few observations. First, if the number of votes in the last period is less than k, having that limit makes no difference. Second, if it is greater than k, and supposing the same temporal invariance as before, then it becomes equivalent to taking a vote cast at random, which means that the probability of winning becomes $\frac{\lambda_c}{\lambda_c+\lambda_v}$ and $\frac{\lambda_v}{\lambda_c+\lambda_v}$, not only in the Poisson case but also if both coercer and voter vote in a regular fashion. To convince oneself of this, it is sufficient to consider two sets of points which can get arbitrarily cycled around the single time slot. One can then see that the probability of the k-th being the coercer's corresponds to the coercer's share of votes cast.

4.3 Considerations on coercion detectability

The previous subsections had the coercer maximise the probability of casting the winning vote while operating on a limited budget (either strictly or in expectation). However, there is a second important aspect to the process, which is the detectability of coercion. This relies on the EA keeping some logs, although those can be very limited depending on how the EA chooses to protect the voters' privacy. One option is to record only the number (and potentially a fuzzied timestamp) of votes cast in a given municipality without attributing them to any voter[11]. Looking at the voting patterns could give reasonable evidence of malfeasance, with potential false positives if some voters test the system or attempt a DOS attack.

This partially mitigates the presumed stronger computational and logistical capabilities of the coercer. Moreover, the kind of coercer we consider presumably targets more than one voter, which gives a second mitigation, as it splits their budget. On the other hand, the coerced voters have an incentive to have an anomaly appear as long as they cannot be linked to it. Thus, even if only some coerced voters cast extra votes, a coercer willing to guarantee a high number of coerced votes will have to add a massive number of votes to the system, putting a second limit to the scalability of coercion — even more so if the logs are more detailed, such as by including network details. If the logs show the number of votes with a given set of credentials, the coercer could perform a counting attack where they cast k_i votes with the $i-th$ credentials and check that this k_i appears in the logs. That forces them to have distinct k_i for different voters and to have them high enough to avoid collisions with random voters (thus a cost that evolves quadratically with the number of coerced voters). However, this in turn would create a detectable pattern of credentials with high use but presumably little to no collisions.

[11] If following this idea, one has to be extra careful as splitting constituencies exposes many systems to more attacks in practice [4].

Looking at timestamps could also reveal people voting in a regular pattern, which would prevent the naive use of the strategy above — although adding small perturbations could make it less noticeable. This also means that the coercer (or in practice the person coordinating coercion attempts) would have a choice between having a centralised system that casts votes using coerced credentials — which would be more detectable — or staying decentralised which might require more technical savvy from local coercers.

All the considerations above are not strictly limited to the case of shareable credentials. Indeed, if the voting token is on the user's smartphone, confiscating a large number of them for a week might appear as an anomaly in phone/mobile internet network use.

5 Example implementation

Stochastic revoting as described here can be added to essentially any internet voting system that does not already have some form of coercion resistance of mitigation. As far as uncoerced voters are concerned nothing changes and the process is unaffected for the conventional voting period going from T_0 to T_1. However, ensuring that such coercion suppression mechanisms interact smoothly with whatever E2E verification is in place is delicate.

If we are prepared to trust an entity to apply the vote selection policy then a Trusted Third Party will give us a trivial implementation. A slightly less trivial solution is to have a set of trustees who must cooperate to select the correct votes for the tally according to the policy, thus spreading the trust. As it is preferable to avoid reliance on trusted parties, we sketch below an approach that makes the tabulation verifiable using a bulletin board (BB). We confine our discussion here to the simple case of a random, secret deadline for each voter.

To facilitate verification, the EA reveals the secret deadlines after voting has finished. Thus, we require that the EA make a cryptographic commitment to the deadlines. To avoid counting style attacks, we do not want the fact that revotes have been cast against a particular credential to be visible on the BB. In particular, we want to avoid revealing the number of votes cast against a particular credential. To this end, we assume there that the underlying scheme employs JCJ-style credentials [12]. In the token based approach, the token could have the credential embedded in it.

In the setup phase, the list of assigned credentials is posted encrypted under EA public key to the BB in the usual JCJ manner. The commitments to the deadlines will be posted alongside, thus the encryption of the i-th credential will appear next to the commitment to the i-th deadline:

$$(\{Cred_1\}_{PK_{EA}}, Commit(t_1)), (\{Cred_2\}_{PK_{EA}}, Commit(t_2)), \\ (\{Cred_n\}_{PK_{EA}}, Commit(t_n)) \quad (1)$$

[12] We are not using the full JCJ mechanism in that here the voter might not be able to lie about their credential.

Votes will be cast in the usual JCJ fashion, as a pair of encrypted vote and encrypted credential: $(\{Cred_i\}_{PK_{EA}}, \{Vote_i\}_{PK_{EA}})$. These will be posted to the BB, timestamped with an encryption of the time at which they were cast.

To avoid revealing the number of ballots cast with a given credential we adopt the following approach to identifying the ballots that satisfy the revoting policy, in this case that the last ballot cast before the deadline is counted. After re-encryption mixing, the timestamps are decrypted and the ballots are arranged in time order of casting. Now the authority takes each of the encrypted credentials as posted during the setup phase and the corresponding deadline commitment is opened. The authority now performs Plaintext Equivalence Tests (PET) against the encrypted credentials in the ballots, working back in time from the deadline for this credential. The first ballot for which the PET succeeds will be the last ballot cast with this credential before the deadline and so it is selected for the input to the tally. We now move to the next credential and repeat the process. For each assigned credential we identify at most one ballot and thus the number cast with this credential remains hidden. This can of course be parallelised across the credentials.

6 Discussion

6.1 Variants

We have shown the expected coercer behaviours and success rate in multiple models based on the central idea. Many other variations are imaginable, with some of them being counter-productive. For example, one could take one of the cast votes at random, or only record the most frequent vote cast (restricted to the extra voting period to avoid affecting non-coerced voters) instead of just the last vote. However, the latter could allow spamming of votes at the beginning of the voting period and affects the strategies and winning probabilities in both directions. One could then implement some rate-limiting or add a small random delay between T_1 and the start of the extra voting period. There are also many questions of information whose answers depend on the precise model used. Depending on the secure channel assumption, would it be possible to let the voter know their own deadline — in a way that they can't share with the coercer? How about revealing that the deadline is passed as soon as it happens — whether on an individual or general basis?

With non-shareable credentials, we could add other actors, such as an accomplice of the voter which also votes following their choices (or more reasonably, a service that does that). We could also consider other cost models: for example, the cost of holding the token might be non-linear in the time spent holding it: it is easier to plan ahead and spend multiple separate days without a phone than to be without for a long period. Moreover, one could consider the logistical aspects of having to retrieve and give back the token (rather than keeping it somewhere), thus having an additional cost for each temporally disjoint component.

All of these considerations bring us to a central trade-off between the complexity of the system, its understandability and its perceived legitimacy. For

example, it would be surprising for there to be no psychological difference between having unlimited votes and having 1 vote per voting period, despite both models behaving similarly strategy-wise. Simply having the extra voting period could be stressful to some voters if they do not understand the principle and are afraid that their vote could be discarded. One option would then be to restrict the voting times in some ways. For example, only allowing votes during daytime or stating in advance that the end of voting period would only be during business hours — reflecting real voter habits which are not uniform — would affect the coercer's strategies by discouraging voting until early morning.

A second issue is that some voters could also believe that they need to vote frequently for their vote to be recorded. Not only would this be costly to them, it would increase the strain on the system as it would be equivalent to a DDOS. The EA should be very careful with such risks to prevent a shutdown (which could affect the legitimacy), as the strain would increase if coercion is prevalent.

Although it depends on the precise variant we look at, the system closest to what we described is probably Caveat Coercitor [8]. However, a few essential differences exist between the two schemes. In Caveat Coercitor, little strategy is needed by either player and the costs are set in advance: both coercer and voter can nullify the vote at limited cost. The coerced voter's strategy is very simple, and indeed identical to the normal, uncoerced voting procedure: cast the intended vote with the valid credential. This guarantees that at worst the vote is nullified and at best is counted as intended. Equally, for the coercer, the best strategy is to cast their vote with what they believe to be the valid credential. Casting multiple votes will not change anything. Thus, even with unbounded resources, the coercer cannot bias the odds in their favour.

Here, nullification is not available: either the coercer wins or the voter does. Moreover, in the shareable credentials setting, the cost is not bounded, giving rise to a different set of large-scale strategies for the coercer (who also suffers from having less information than the one in Caveat Coercitor, who can accurately guess how many votes are nullified). The system also has an interesting property if we assume that the voters keep casting their votes following a somewhat uniform distribution — that is, if the ratio between the densest voting period and the sparsest one is bounded by k. One can then show that the "return" the coercer gets from their budget is continuous (and is in fact k-lipschitzian), which prevents the coercer from adopting all-or-nothing min-maxing strategies.

6.2 Is it democratic?

One potential criticism of this kind of scheme is that it could infringe on some principles of democratic equality. Having different voting periods for various sets of voters could be seen as unfair as it gives some voters more opportunities to change their vote, especially if more information comes out. This is a politically non-trivial question, which is affected by multiple elements.

First, one should insist that all voters have an equal opportunity to vote in the initial period, anyone willing to cast a vote should then be able to get it counted — the only ones affected by this system are the potentially coerced voters. There

remains a small issue in that having a fuzzy deadline is psychologically different from having a strict one, and could lead to more people forgetting about it and missing their opportunity to vote.

Second, this is mostly a question of degree and not nature: in systems without revoting, voters who go to the polling office early renounce their rights to change their minds later. Although doing so remains the voter's choice, delaying runs the risk of missing the deadline at the polls. Some systems already have non-overlapping voting periods for different populations (for example, to let people vote from outside the country). Moreover, although some electoral systems (like France) enforce a news shutdown on the day of the polls to prevent early polling data from affecting later voters, online media from neighbouring countries has partially rendered obsolete this type of policy.

The final element is that the question of whether voting early or late is an advantage is context-dependent. Although late voters have more information, if early voters can commit to their choices (as in a Stackelberg game), they can force the other voters to change their voting patterns. An old example of this can be found in the Roman Republic as the choices made by the first set of voters called to openly vote generally determined the result of the election [16].

One potential way to assuage those fears in both frameworks would be to have the same cutoff date for all voters, while it it still taken uniformly at random over the extra voting period. However, this would facilitate min-maxing strategies for the coercer — as the proof that their return is k-lipschitzian depends on the independence of events.

6.3 Verifiability and future work

Getting verifiability and coercion resistance to work together is always challenging due to the tension between transparency on the one hand and privacy on the other. Here it is particularly difficult as we are dealing with potentially quite complex policies, involving randomisation, for selecting the votes to be included in the tally. To implement such policies in a verifiable way will involve zero-knowledge protocols, verifiable sources of randomness, distributed computation etc. Implementation of such variants is left to future work, as is the formalisation of the required properties and associated proofs.

7 Acknowledgements

This paper was supported by the Luxembourg National Research Fund (FNR) under the CORE project EquiVox (C19/IS/13643617/EquiVox/Ryan). The authors wish to thank Lê Thành Dũng (Tito) Nguyễn and Édouard Thomas for productive discussions on measure theory and some corrections in section 4.

References

1. Bojinov, H., Sanchez, D., Reber, P., Boneh, D., Lincoln, P.: Neuroscience meets cryptography: Designing crypto primitives secure against rubber hose attacks. In: 21st USENIX Security Symposium. pp. 129–141. USENIX, Bellevue, WA (2012)
2. Chaum, D., Carback, R.T., Clark, J., Liu, C., Nejadgholi, M., Preneel, B., Sherman, A.T., Yaksetig, M., Yin, Z., Zagórski, F., Zhang, B.: Votexx: A solution to improper influence in voter-verifiable elections. Cryptology ePrint Archive, Paper 2022/1212 (2022), https://eprint.iacr.org/2022/1212, https://eprint.iacr.org/2022/1212
3. Cortier, V., Gaudry, P., Yang, Q.: Is the JCJ voting system really coercion-resistant? In: 37th IEEE Computer Security Foundations Symposium (CSF). CSF 2024, IEEE, Enschede, Netherlands (2024), https://inria.hal.science/hal-03629587, this is the long version of the paper published at CSF 2024.
4. Debant, A., Hirschi, L.: Reversing, breaking, and fixing the french legislative election e-voting protocol. Real World Crypto (2023)
5. Enikolopov, R., Korovkin, V., Petrova, M., Sonin, K., Zakharov, A.: Field experiment estimate of electoral fraud in russian parliamentary elections. Proceedings of the National Academy of Sciences **110**(2), 448–452 (2013)
6. Frye, T., Reuter, O.J., Szakonyi, D.: Hitting them with carrots: Voter intimidation and vote buying in Russia. British Journal of Political Science pp. 1–25 (2018)
7. Gallager, R.: Discrete stochastic processes textbook. Tech. rep., MIT (2011), https://ocw.mit.edu/courses/6-262-discrete-stochastic-processes-spring-2011/
8. Grewal, G.S., Ryan, M.D., Bursuc, S., Ryan, P.Y.: Caveat coercitor: Coercion-evidence in electronic voting. In: 2013 IEEE Symposium on Security and Privacy. pp. 367–381. IEEE (2013)
9. Haines, T., Smyth, B.: Surveying definitions of coercion resistance. Cryptology ePrint Archive, Paper 2019/822 (2019), https://eprint.iacr.org/2019/822, https://eprint.iacr.org/2019/822
10. Harvey, C.J.: Changes in the menu of manipulation: Electoral fraud, ballot stuffing, and voter pressure in the 2011 russian election. Electoral Studies **41**, 105–117 (2016). https://doi.org/https://doi.org/10.1016/j.electstud.2015.11.004
11. Heiberg, S., Parsovs, A., Willemson, J.: Log analysis of estonian internet voting 2013–2014. In: International Conference on E-Voting and Identity. pp. 19–34. Springer (2015)
12. Horwitz, D.A.: A picture's worth a thousand words: Why ballot selfies are protected by the first amendment. SMU Science and Technology Law Review **18**, 247 (2015)
13. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society. pp. 61–70 (2005)
14. Pereira, O.: Individual verifiability and revoting in the estonian internet voting system. In: Financial Cryptography and Data Security. FC 2022 International Workshops: CoDecFin, DeFi, Voting, WTSC, Grenada, May 6, 2022, Revised Selected Papers. p. 315–324. Springer-Verlag, Berlin, Heidelberg (2023)
15. Ryan, P.Y.A., Rønne, P.B., Iovino, V.: Selene: Voting with transparent verifiability and coercion-mitigation. In: Financial Cryptography and Data Security – FC, Revised Selected Papers. pp. 176–192 (2016)
16. Vanderbroeck, P.J.: Popular leadership and collective behavior in the late Roman Republic (ca. 80-50 BC), vol. 3. Brill (2023)

A Appendix : additional proofs

A.1 Up to a set of measure 0, the coercer's optimal choices must be of the form $f^{-1}(]x;+\infty[) \cup E$ where $E \subseteq f^{-1}(\{x\})$,

We now consider a subset I of $[T_1;T_2]$ corresponding to when the coercer holds the token, with $\mu(I)$ its Lebesgue measure. The function $g\colon x \mapsto \mu(f^{-1}(]x;+\infty[))$ is non-increasing, has value $\geq \mu(I)$ at 0 and tends to 0 at $+\infty$. There must then exist some x such that $g(y) \leq \mu(I)$ for $y \leq x$ and $g(y) > \mu(I)$ for $y > x$. Therefore, we have $\mu(f^{-1}([x;+\infty[)) = g(x) \leq \mu(I) \leq \lim_{y\to x^+} g(y) = \mu(f^{-1}(]x;+\infty[))$.

Thus, there must exist a set J of measure equal to $\mu(I)$ in-between $f^{-1}([x;+\infty[)$ and $f^{-1}(]x;+\infty[)$; this set has the desired shape $J = f^{-1}(]x;+\infty[) \cup E$ where $E \subseteq f^{-1}(\{x\})$.

Since $\mu(I) = \mu(J)$, we have

$$\mu(I \setminus J) = \mu(I) - \mu(I \cap J) = \mu(J) - \mu(I \cap J) = \mu(J \setminus I)$$

Furthermore, since $f(J) \subseteq [x, +\infty[$ and $f(\mathbf{R} \setminus J) \subseteq [0, x]$ by definition, we have

$$\begin{aligned}\int_I f = \int_{I\cap J} f + \int_{I\setminus J} f &\leq \int_{I\cap J} f + x \times \mu(I \setminus J) \\ &= \int_{I\cap J} f + x \times \mu(J \setminus I) \leq \int_{I\cap J} f + \int_{J\setminus I} f = \int_J f\end{aligned}$$

As the inequalities above are in fact equalities due to I being optimal, if we let $K = \{y \in I \setminus J : f(y) < x\}$, then $\mu(K) = 0$.

A.2 The uniform distribution is optimal for the EA.

Let f be the non-uniform distribution and let $f'(y) = c$, with $\int_\Omega f = \int_\Omega f'$. Let us show that for any cost k that the coercer is ready to pay, the total probability achieved with this cost is at most equal in the uniform distribution.

Let us suppose that the coercer chose an optimal set I of measure $\mu(I) = k$. We can then find x such that $I = f^{-1}(]x;+\infty[) \cup E$.

If $x \geq c$, then $\int_I f \geq c \times \mu(I)$. Let us then suppose that $x \leq c$. For any $z \in \Omega \setminus I, z \leq x$. Then $\int_{\Omega\setminus I} f \leq c \times (1 - \mu(I))$. As $\int_I = \int_\Omega - \int_{\Omega - I}$, we obtain that $\int_I f \geq c - c(1 - \mu(I)) \geq c \times \mu(I)$.

Thus the coercer always achieves an equal or higher probability with a non-uniform distribution.