

P0KERface: Coercion Resistant End-to-End Remote Voting

Jennifer Ramseyer, Ray Hua Wu, Victoria Xia, Dai Yang

May 12, 2016

Abstract

We analyze recent electoral systems, particularly focusing on how they address end-to-end verifiability, coercion resistance, and remote voting. We show that these systems are vulnerable to various types of electoral fraud if remote votes may be coerced or corrupted. We then propose amendments to these systems in the form of *P0KERface*¹, a new electoral system that admits remote voters while providing both end-to-end verifiability and coercion resistance with high probability, while keeping the electoral process as simple and secure as possible.

1 Introduction

Designing and implementing a voting system that guarantees confidentiality, integrity, and accessibility (including to remote and absentee voters) is a hard problem that has been closely considered for many years. Systems that partially solve these problems include ThreeBallot [Riv06], Scantegrity [CEC⁺08], Remotegrity [ZCC⁺13], sElect [KMST], and Civitas [CCM07]. The improvement of voting methodology is critical to protecting the validity of the central tenet of democracy.

ThreeBallot [Riv06] is the first of these systems proposed, designed to simultaneously provide coercion resistance and verifiability. In ThreeBallot, the voter casts three ballots among which choices they are voting for are marked on two of the three ballots and choices they are not voting for are marked on one of the three ballots; a copy of one of the ballots is their receipt. A coercer will not be able to tell what the actual votes were for by looking at the receipt, as marked votes could be either candidates voted for or not voted for, and the fact that the total number of votes across all ballots is fixed is central to the integrity of the system.

Scantegrity [CEC⁺08] is a voting system that allows any member of the public to verify not only that their ballot was counted but also that overall tallies of votes are correct. One implementation of Scantegrity is Scantegrity II [CCC⁺08], where ‘II’ stands for Invisible Ink. A special pen is given to voters to mark ballots with their votes of choice in a way that reveals codes originally on the ballot in invisible ink that can be used for verification later.

Remotegrity [ZCC⁺13] extends many of the benefits of Scantegrity to remote voters. It gives a means of accountability for which upon verification, an

¹Probabilistic, Zero-Knowledge, E2E verifiable, Remote voting interFACE

incorrect tally can be accurately blamed on either malicious voting software or corrupt election officials. It provides a mechanism of dispute resolution upon detection of incorrect tallies. Unfortunately, Remotegrity lacks coercion resistance to remote voters, an essential protection to voters that vote remotely and thus cannot benefit from the confidentiality protections of an uncompromised polling station.

sElect [KMST] is a very new system that provides privacy, verifiability, and accountability through a web voting system for low-risk elections. Unfortunately, it is the high-risk elections in which we more likely care further about election security.

Civitas [CCM07] is another very new system that addresses many problems. It provides remote voting, and all of verifiability by the public, verifiability by the voter, confidentiality, and coercion resistance. We think these attributes could be improved to a certain degree though.

Absentee voting is an important part of elections, and should be made secure. During elections, voters may be overseas because of travel or service in the military. They may also have a disability making it easier for them to vote remotely. The Uniformed and Overseas Citizens Absentee Voting Act in 2015 [oD15] illustrates the importance of secure absentee voting. To prevent election fraud, an E2E electoral system is highly desirable, and thus we seek to improve upon Remotegrity's [ZCC⁺13] absentee voting protections.

E2E voting remotely is a particularly thorny problem. Votes should be protected from corrupt officials, and also from client-side attacks. These client-side attacks include coercion: that is, the manipulation of a voter's behavior through force or threat. Coercion is especially problematic for absentee voters who cannot be ensured the safety of voting on-site.

We improve on previous voting schemes by providing verifiability for the public and individual voters, confidentiality for voters, and coercion resistance for all voters including remote voters. We reach a better guarantee of coercion resistance than Civitas provides.

2 Setup

We outline the basic setup of our voting system, were it to be implemented in an election.

Our security policy is as follows.

2.1 Parties Involved

Parties involved include:

- voters
- coercers; may be voters themselves
- election authority (EA); in charge of distributing and tallying ballots, may be corrupt
- registration office; a third party assumed to be a trusted which voters may only interact with in person after presenting photo ID.

We also assume the existence of a public append-only bulletin board (BB) to which the EA may append.

2.2 Desired Properties

The desired properties of our system are:

- voter secrecy: voters should not be able to prove how they voted to anyone else, even if they wish to, thereby preventing coercion and vote-selling
- voter verifiability: voters should have reassurance that their votes were recorded and counted correctly
- public verifiability: the public (i.e., voters) should be able to verify with reasonable confidence that the election results announced by the EA are correct.

3 The Scheme

We describe a coercion-resistant and E2E with high probability absentee voting scheme.

3.1 Voter Registration

When a voter registers to vote, she also creates a password, known only to her and the EA. To ensure secrecy of this password, voters are required to register to vote in person so that the registration office, assumed to be a trusted third-party, can ensure the voter chooses her password privately and cannot leave with proof of the password. Note that this is not an unreasonable burden on voters since registration need only be done once, and it is reasonable to expect that voters will be in the location of residence at some point.

3.1.1 Registration Procedure

The registration procedure is as follows:

1. Voter reports to registration office and shows photo ID.
2. Registration office checks voter eligibility, then asks the voter to choose a password in private.
3. Voter leaves without any proof (including photos, etc.) of the password chosen.

3.1.2 Password Strength

A minimum length will be imposed, perhaps around 10 characters. Special characters will not be mandatory because such requirements are usually fulfilled in predictable ways, such as changing ‘password’ to ‘Pa55w0rd!’. We do not believe this is an unreasonable burden on the voter, since the importance of retaining one’s password will be emphasized during the registration process. Furthermore, passwords will be screened against common dictionary attack candidates,

such as ‘1234567890’ and combinations of the voter’s birthdate and birthplace; voters with such passwords will be asked to recreate their password to be less predictable.

3.1.3 Password Recovery/Updates

To keep voters’ passwords as secure as possible, any changes or updates to a voter’s password must be handled in person, at the registration office (after the voter’s photo ID has been checked). Thus, the voter should remember and secure her password. Note that because there is no way to prove the correctness of a password, a voter cannot be forced into revealing a password to a coercer, since the coercer has no way of determining whether a response from the voter is correct or not. Likewise, there is no way for a vote buyer to ensure that their transaction is fulfilled.

3.2 Before the Election

Before the start of the election, the EA publishes to the BB a list (i.e.: makes a public commitment) of all registered voters and their unique voter IDs (chosen by the EA). As is the case of most voting schemes, we deem it unnecessary to keep the list of registered voters secret, since there is little harm in others knowing whether an individual is registered to vote or not. Because voter IDs are not sensitive in our scheme, there is no harm in making public the mapping between voter names and voter IDs.

If a registered voter recognizes that her name does not appear on this public list of registered voters, she may bring a dispute to the registration office, which may then verify whether she is registered to vote or not and potentially reveal a corrupt EA if the voter’s claim is accurate. In this way, a corrupt EA cannot successfully remove voters from the list of registered voters.

In order to impede a corrupt EA from adding fake voters to the list of registered voters and thereby forging their votes, after the public commitment is made, the registration office will randomly sample some proportion of listed registered voters to confirm that the names correspond to actual registered voters. This check ensures that with reasonable probability, a corrupt EA will not get away with adding names to the list of registered voters.

3.3 During the Election

Voters vote by providing their voter ID, password, and desired vote. If the provided password and voter ID do not match, the vote is discarded without informing the voter. In this way, a voter being coerced to vote a certain way by a coercer can simply enter an incorrect password to prevent the vote from being counted, and coercer will have no way of knowing that the password was incorrect. If a voter votes multiple times with her correct password and voter ID, only the last vote is counted.

These votes may either be provided via an online interface or via mail. See § 4.1 for tradeoffs between these two voting mediums. For now, we simply assume that voters have a way to provide votes to the EA for tallying.

3.4 After the Election

3.4.1 Tallying

As soon as the election is over, the EA publishes a set of recorded votes (i.e., how many voters there were for each candidate, along with how many registered voters chose not to vote). The sum of the number of votes (where we include "no vote" as a valid vote option) should match the number of registered voters committed to before the start of the election. With this public set of recorded votes, anyone can verify the announced winner of the election.

3.4.2 Voter Verification of Votes

For a period of two months after the end of the election, and long before the results of the election take effect), voters can verify their votes, in person, as follows:

1. A voter appears, in person, at the registration office and presents photo ID.
2. An official from the registration office, the voter, and an EA representative (possibly an automated software) will isolate themselves in a room. Their privacy will be ensured by the registration official.
3. The EA representative and voter will begin a zero-knowledge proof of correctness of the voter's vote, as follows:
 - (a) EA commits to a mapping from each of the published votes to a value $H(\text{voter ID} + 'v' + n)$, where $H(\cdot)$ is a public, collision-resistant hash function, $+$ represents concatenation, and n is a large random number that differs from line to line.
 - (b) The voter checks that this commitment agrees with the publicly published set of votes.
 - (c) The registration official provides the EA with the voter's voter ID (i.e., the challenge).
 - (d) The EA points to the line in the commitment that corresponds to the voter's vote, and reveals the random n associated with that hash.
 - (e) The voter verifies the hash matches the commitment, and that the line pointed to by the EA maps to the voter's correct vote.

This zero-knowledge proof for verifying honesty of the EA's recording and tallying of votes is complete, sound, and zero-knowledge. The protocol is complete because an honest EA will always be able to provide the random n that hashes to the voter's vote. It is also sound because the number of lines in the EA's commitment matches the number of registered voters, which means if some votes have been tampered with, then there is a nonzero chance that the provided challenge will be one of the ones tampered with, which would result in the EA being caught with high probability, since the hash function is collision-resistant, which means the EA cannot use the same hash to satisfy multiple challenges. Finally, the protocol is zero-knowledge since the inclusion of the random n values with each hash means voters do not learn anything about other voters' votes from

the EA's commitment (assuming the random n come from a sufficiently large space).

Should a voter uncover dishonesty on the part of the EA, we enter a dispute-resolution stage, as in § 3.4.3.

3.4.3 Dispute Resolution

If the EA fails the challenge of the above zero-knowledge proof, the vote is considered disputed. A dispute between a voter and the EA can be the result of foul play, human error, or both. Dispute resolution is handled as follows. Prior to each election, election officials will decide on a security parameter $0 < r < 1$ representing the level of tolerance. After the results are published, they must track the ratio of disputed votes to minimum margin of victory, and if that fraction ever exceeds r , then the election results are declared to be untrustworthy. At that point, the correct course of action may be a rerun, an investigation, or even an impeachment, though such details are best left to policy-makers. The reason we use a proportion r of the minimum margin of victory, and not the minimum margin of victory itself, is to accommodate for human error. For example, if a coercer demands a voter to reveal their password, that voter may be intimidated into revealing their actual password, even though giving a fake password would be safe.

More precisely, the minimum margin of victory is defined to be $m = \min_{i < j} |C_i - C_j|$, where C_i is the number of votes (possibly weighted depending on the election format) cast in favor of candidate i . Hence, if at least rm votes are disputed at any point, then the election can no longer be trusted. The value of r will be chosen for each individual election based on socio-political circumstances, such as ease of coercion, level of importance, and previous history of foul play. Clearly, the lower the value of r , the stricter the election is. Generally, we recommend setting r as low as possible while permitting occasional human errors on the part of the voters.

3.5 Assumptions

The assumptions made by our scheme are as follows:

- A sufficient proportion of voters will verify their vote after the election, in order to provide probabilistic security against a malicious EA.
- The EA cannot predict which voters will verify their vote. (Otherwise, a malicious EA could know which votes they may safely tamper with.)
- The existence of a public, append-only bulletin board.

4 Additional Design Decisions

4.1 Voting Medium

The voting medium can be either by mail or electronic, each with their own advantages and downsides. Voting by mail means that mail interception and rerouting are possible, and latency will be greater. On the other hand, voting

by Internet speeds the process up, but opens up numerous avenues of attack. This includes replay attacks, timing attacks, and even keylogs. An additional concern arises in the case where a voter's computer used for voting is silently compromised, for example through a phishing email, prior to registration. Although the decision ultimately depends on the state of each medium available to the voter population, we suggest that as of now, voting by paper is likely a safer option.

5 Security Analysis

The security of our voting scheme relies on a few key points: voter passwords, the vote verification protocol, and the vote-collection and tallying duties of the EA. In this section we discuss how each of these components affect the security of our scheme.

5.1 Compromise of Voter Password

Our scheme relies on voters keeping their passwords secure. If a coercer were to learn a voter's real voting password, the coercer could make the voter vote with the real password (or vote using the voter's voting ID and password himself), thus making the coerced vote the actual vote that is counted. Our hope is that the voter, when confronted by the coercer, will lie about her real password. But, in the event that the coercer learns her password, the voter's vote may be lost. In that election, the voter may have to vote as the coercer dictates. After the election, she can go to the registration office and change her password. Because the voter needs to go to the registration office in person and provide photo identification, the coercer cannot go in her stead.

It is mathematically implausible to guess a password's hash by brute-force. If each password is alphanumeric and contains at least 10 characters, then the space of possible passwords is $36^{10} > 2^{51}$. Even a supercomputer would not be able to find a hash collision efficiently, provided that the passwords are hard-to-guess. As aforementioned, this requires discouraging the voters from choosing predictable passwords, a practice which is sadly and remarkably common.

5.2 Critical Mass of Voter Verification

Because our scheme depends on voters verifying their votes in person, perhaps our largest failure mode lies in human laziness. If not enough voters verify their votes, we cannot guarantee that the votes were counted correctly. We hope that the voting public, filled with a sense of civic duty, will come to the verification office and check their vote. It seems reasonable to assume that a substantial proportion of voters will want to confirm their vote, whether from civic duty, curiosity, or mistrust in the voting system. If we notice in trials that not enough people are checking their vote, we will offer incentives. Potential incentives include coupons to local businesses, tax breaks, cookies, or free postage stamps. Lured by the promise of these treasures, we are confident that enough of the voting public will verify their vote.

Mathematically, suppose that the EA wants to corrupt c votes randomly. But a fraction v of the voters verify their vote. Hence, the probability that the

EA gets away with it is about $(1 - v)^c$, which decreases exponentially with c . This strongly discourages tampering with the votes, though it does require v to not be minuscule. As an example, with $v \approx 37\%$, if a corrupt EA tampers with just ten votes, they will be caught with a 99% chance.

Alas, incentivizing verification only works if a large enough proportion of the voting population will be local at some point during the specified time-window after the election closes, in order to verify their votes in person. If a particular voting population struggles with this, we will either have to extend the time-window for verification, or create a system in which one can verify her vote remotely. One possible method to achieve the latter would be to verify over the telephone, although our group has not yet explored this option.

5.3 Corrupt EA

Our scheme offers the following protections against a corrupt EA.

5.3.1 Adding Votes (Ballot-Box Stuffing)

If the election authority is corrupt, then he may try to add additional ballots to the official vote. Our system handles ballot-box stuffing by randomly sampling the ballots to check their legitimacy. Please see § 3.2 for more details.

5.3.2 Tampering with Votes

A corrupt EA may try to tamper with the votes in the verification step. Assume there is no remote verification method, and that the EA knows who is voting remotely. Since the remote voters cannot verify their votes, the EA can tamper with their votes without anyone noticing. The remote voters cannot check that their vote was counted correctly, and no one locally can check on their behalf. Therefore, the EA's vote modification will escape undetected. In order to block this attack, we prevent the EA from knowing who is voting remotely. As such, we have the trusted, third-party registration office maintain voters' addresses separate from the EA, so that the EA only knows who is registered, and not where they reside. The registration office also handles voting passwords, so that a corrupt EA cannot steal the passwords.

6 Conclusion

Designing protections for absentee voting, long a source of insecurity in elections, is an imperative need. Our solution not only secures remote voting, but also makes it coercion resistant. By providing both verifiability and coercion resistance, we provide election standards that are comparable or better than current governmental elections. Granted, we add some overhead compared to the current electoral system, but this overhead is an acceptable tradeoff given our increased security.

References

- [CCC⁺08] David Chaum, Richard Carback, Jeremy Clark, Aleksander Essex, Stefan Popoveniuc, Ronald L Rivest, Peter YA Ryan, Emily Shen, and Alan T Sherman. Scantegrity ii: end-to-end verifiability for optical scan election systems using invisible ink confirmation codes. *EVT*, 8:1–13, 2008.
- [CCM07] Michael R Clarkson, Stephen Chong, and Andrew C Myers. Civitas: A secure voting system. Technical report, Cornell University, 2007.
- [CEC⁺08] David Chaum, Aleks Essex, Richard Carback, Jeremy Clark, Stefan Popoveniuc, Alan Sherman, and Poorvi Vora. Scantegrity: End-to-end voter-verifiable optical-scan voting. *Security & Privacy, IEEE*, 6(3):40–46, 2008.
- [KMST] Ralf Küsters, Johannes Müller, Enrico Scapin, and Tomasz Truderung. select: A lightweight verifiable remote voting system. *Grand Region Security and Reliability Day*.
- [oD15] Federal Voting Assistance Program: Department of Defense. The uniformed and overseas citizens absentee voting act. *U.S. Department of Justice*, 2015.
- [Riv06] Ronald L Rivest. The threeballot voting system. 2006.
- [ZCC⁺13] Filip Zagórski, Richard T Carback, David Chaum, Jeremy Clark, Aleksander Essex, and Poorvi L Vora. Remotegrity: Design and use of an end-to-end verifiable remote voting system. In *Applied Cryptography and Network Security*, pages 441–457. Springer, 2013.