The DROP Protocol: Dispute Resolution via Observation in Public for Verifiable, In-Person Voting

Josh Benaloh[1], Michael Naehrig[1], and Olivier Pereira[2]

[1] Microsoft Research, Redmond, WA, USA
[2] UCLouvain, B-1348 Louvain-la-Neuve, Belgium

Abstract. Dispute resolution has been a significant challenge in verifiable election protocols since such protocols were first proposed more than forty years ago. This work explores the problem from a new perspective and offers strong dispute resolution for in-person voting by depending on observers.

It proposes a simple definition of dispute resolution as a property of a voting protocol—a definition that is independent of any other security goal. It also presents the DROP protocol, a verifiable, in-person voting protocol that runs in the presence of observers who will always reach a correct conclusion in the case of a dispute without ever being able to compromise privacy or facilitate coercion.

1 Introduction

We describe a process for in-person voting with dispute resolution. We explore how human observers can be used to mitigate the trust assumptions that are typically made in previous protocols. The specific scenario that we address is for in-person voting using paper ballots and ballot scanners. Within this scenario, we describe how end-to-end verifiability can be achieved while allowing one or more honest observers to correctly resolve disputes between voters and election administrators (including equipment under the control of administrators). Although these observers will be able to correctly adjudicate disputes, they will not obtain any information about the selections made by voters on ballots that they cast; nor will observers have any means of coercing voters. Observers do not participate in the voting protocol; instead they merely serve as witnesses to some of the actions taken by voters and devices.

A common focus for dispute resolution is the context of recorded-as-cast verifiability. In this context, the goal is to address disputes that occur when a voter disagrees with the way in which a vote is recorded (or not recorded) in the election record published on a bulletin board.

We address this issue but additionally focus on dispute resolution in cast-as-intended scenarios where a voter follows the process of casting a vote for one option but has a device producing a vote for another option. Methods like "cast-or-challenge" can be used by voters to probe a vote collection device and

determine if it is recording votes correctly. A voter may request a vote for one option, receive a commitment to a vote, and then challenge the vote to see if the decommitted value is a vote on which the same option has been selected. The problem here is that there are scenarios that are very difficult to distinguish without compromising privacy. One such pair of scenarios is as follows.

1. A voter requests a vote for option A, receives a commitment, challenges the commitment, and receives a decommitment that shows a vote for option B. The voter then claims that a vote for option A was changed to option B.
2. A voter requests a vote for option B, receives a commitment, challenges the commitment, and receives a decommitment that shows a vote for option B. The voter then claims that a vote for option A was changed to option B.

Another pair of scenarios that is difficult to distinguish is the following.

1. A voter requests a vote, receives a commitment, challenges the commitment, and has the vote cast by the device.
   The voter then claims that the device failed to honor a challenge request.
2. A voter requests a vote, receives a commitment, asks that the vote be cast, and has the vote cast by the device.
   The voter then claims that the device failed to honor a challenge request.

This second scenario is of concern since a malicious device that has been *caught* attempting to change a vote for option A to a vote for option B might attempt to hide the malfeasance by refusing to answer a challenge and instead asserting that it was instructed by the voter to cast the vote.

The complication here is that if the interaction between the voter and device takes place within a private environment, there seems to be no way for an impartial judge to later decide between the two cases in either of the scenarios above absent assumptions about the hardware or processes used. This is true even if requests and responses are authorized by digital signatures or other mechanisms.

We will concentrate our attention on a voting process that offers dispute resolution in the following scenario.

- Ballots are hand-marked in person on paper and read by ballot scanners.
- Elections are end-to-end verifiable with the cast-as-intended property realized using the cast-or-challenge paradigm.
- Verifiability and dispute resolution should not compromise privacy or create additional risks of voter coercion.

This restricted scenario provides a clear, well-defined environment within which to explore options for dispute resolution.

2 Disputes in Elections

2.1 Verifiable Elections

Elections traditionally involve three sets of participants: a set of voters $\mathcal{V}$, the election authority $\mathcal{A}$, and a set of observers $\mathcal{O}$. Each of these participants may

or may not be honest and to keep our notation simple, we often use $\mathcal{V}$ or $\mathcal{O}$ to designate a single voter or an observer rather than the corresponding sets.

We are interested in end-to-end verifiable, in-person voting processes. Such processes provide voters with *individual verifiability*, meaning that voters can verify that the election authority has correctly included their votes in the election record. They also provide *universal verifiability*, meaning that the authority must publish information that makes it possible for anyone to verify that the tally accurately reflects the contents of the recorded ballots. Sometimes, *eligibility verifiability* is also enabled, meaning that the authority makes it possible for anyone to verify that the ballots appearing in the election record were cast by eligible voters only.[3]

Offering verifiability in any of the above forms means that one of the corresponding verification steps can fail. If this is the case, it must be possible for a verifying party to *complain* to observers. The role of observers here is demanding: they typically are not just auditors or judges that inspect evidence after the fact. They are present in voting stations during voting operations and are expected to observe the voters and voting equipment at crucial steps of the voting process. For instance, they are expected to be able to observe if a voter enters a piece of paper into a publicly visible ballot scanner, or whether a printer prints a take-home receipt for a voter when a ballot has been cast. They should also be able to receive complaints from voters. We emphasize, however, that observers are *not* able to see the selections made by voters, nor are they able to take any actions that may coerce voters into making particular selections.

Even within this scenario, it is challenging to design processes wherein honest observers will *always* reach the correct conclusion in any dispute.

2.2 When Do Interesting Disputes Arise?

Disputes are trivial to resolve for the verification steps that are part of universal and eligibility verification: any observer can inspect the records published by $\mathcal{A}$, perform the verification steps, and reach the correct conclusion depending on the outcome of the verification.

Non-trivial disputes actually arise in the context of individual verifiability: a voter may, for instance, claim any of the following.

- A vote has been incorrectly recorded.
- A vote was not recorded at all.
- A vote was recorded that should not have been.

Any such claims may be denied by the election authority.

A dispute can also arise in protocols that include a cast-or-challenge mechanism: a malicious machine that prepared an incorrect ballot and receives a `CHALLENGE` instruction may try to cast the ballot anyway in order to avoid detection, and a malicious authority willing to coerce a voter may try to enforce a challenge on a ballot that a voter has instructed the administrator to `CAST`.

[3] Eligibility often includes not having previously voted in a particular election.

In all of the above cases, voters should have means of convincing observers that they have been wronged.

In our voting protocol, disputes are only raised by voters. If a voter acts improperly, an honest election authority $\mathcal{A}$ will act according to the protocol—perhaps refusing to accept a ballot. It then becomes the sole responsibility of a voter to issue a complaint if the voter believes that $\mathcal{A}$ has acted improperly. This leads us to define a dispute as follows.

Definition 1. *A* dispute *arises in an election protocol if a voter $\mathcal{V}$ complains to an observer $\mathcal{O}$ asserting that the election authority $\mathcal{A}$ acted improperly.*

2.3 Dispute Resolution

We aim to provide *dispute resolution* whenever a dispute arises. Informally, a protocol offers dispute resolution when honest observers will always agree with an honest entity in any dispute.[4] Dispute resolution is solely concerned with identifying malfeasant parties. It does not ensure that an election can be completed in the case of malfeasant behavior. For instance, if $\mathcal{A}$ destroys all the ballots before they are tallied—which is something that $\mathcal{A}$ can often do—we should be able to identify that $\mathcal{A}$ misbehaved, but we may not be able to recover and tally the election. In general, it should not be possible for a malicious voter $\mathcal{V}$ or observer $\mathcal{O}$ to prevent an honest administrator $\mathcal{A}$ from completing an election; but in practice a malicious party may be able to prevent completion of an election with physical vandalism, for example by damaging a ballot scanner.

More precisely, we define what it means for a protocol to allow dispute resolution as follows.

Definition 2. *A voting protocol allows for* dispute resolution*, if an honest observer $\mathcal{O}$ always agrees with the voter $\mathcal{V}$ if $\mathcal{V}$ correctly followed the election protocol, and always agrees with the election authority $\mathcal{A}$ if $\mathcal{A}$ correctly followed the protocol.*

This definition only makes sense in the context of protocols that satisfy the following basic correctness requirement: if both $\mathcal{A}$ and $\mathcal{V}$ correctly follow the protocol, there will be no complaints and no disputes. Furthermore, our definition leaves the resolution of the dispute open if both $\mathcal{A}$ and $\mathcal{V}$ deviate from the protocol. Finally, our definition only applies to honest observers. No assertions are made about the conclusions reached by dishonest observers.

Our notion of dispute resolution is only a property that a protocol exhibits in the case of a complaint, and it does not say anything about the relevance or meaning of the complaint itself. A protocol in which nobody ever complains will offer dispute resolution, even though such a protocol would not be end-to-end

[4] Note that there is never a dispute between two honest entities, so at most one entity is honest in a dispute. If exactly one entity is honest, an honest observer should agree with the honest entity. If neither party to a dispute is honest, observers are free to reach any conclusion.

verifiable for instance. We keep it as a completely distinct objective to design protocols in such a way that complaints are raised when critical properties, like correctness of the election results, are violated—this is what should happen in an end-to-end verifiable protocol for example.

2.4 Related Work

The importance of dispute resolution in voting protocols has long been recognized in the design of verifiable voting systems [1,2,5,6,7,8,10,11,14,17].

In most of these works, the idea of dispute resolution is expressed informally, but it has also been formalized on different occasions. Küsters et al. [14] proposed a general notion of *accountability*, which they instantiate in various contexts. In their setting, a protocol comes with a set of goals, formally defined as a subset of possible executions of the protocol that are considered to be valid. For instance, in a voting protocol, valid executions could be those in which the tally that is announced matches the set of ballots that have been cast. The protocol is accountable if, whenever a goal is not met, a judge can blame at least one misbehaving party.

Basin et al. [2] focus on *timely* dispute resolution for a specific set of disputes: they require that protocols allow voters to always be able to hold evidence that makes it possible to resolve disputes before the end of the election process. They show that this can only be achieved by making strong assumptions on communication channels (that, for instance, may need to be reliable) and regarding some protocol participants (that may, for instance, never block any messages).

The main novelty of our notion of dispute resolution is its independence of any specific security goal or property (verifiability, etc.): it only claims that one can identify a party who deviated from the protocol when a voter complains. It is up to protocol designers to trigger complaints in useful situations, e.g., when a verification step fails. This makes it possible to express dispute resolution as a self-contained property.

3 The DROP Protocol

3.1 Overview of the DROP Protocol

Our *Dispute Resolution via Observation in Public* (DROP) protocol runs at a polling place that is operated under the authority of $\mathcal{A}$ encompassing of poll workers and using various pieces of equipment together with numerous back-end processes.

We outline the voting experience here, assuming that everyone behaves honestly, before offering a detailed description of our protocol below.

Voters begin by entering a polling place and identifying themselves to the poll workers. Voters who are deemed eligible to vote each receive an envelope and two paper slips marked `CAST` and `CHALLENGE` together with an envelope. The voter then enters a booth. Within the booth, the voter commits to a *declaration*

by placing one of the paper slips in the envelope and destroying or removing the other.[5] The voter then leaves the booth and places the sealed envelope in front of the poll workers—receiving a blank ballot in exchange.

The voter then returns to a voting booth and marks the ballot. If the declaration is `CAST`, the voter marks the ballot with true preferences. If the declaration is `CHALLENGE`, the voter may make any selections—including selections dictated by a coercer. The voter leaves the booth while keeping the ballot hidden (or folded), walks to a scanner that is publicly observable, and enters the ballot into the scanner, in such a way that all observers can see that a paper has been entered but without revealing the ballot contents (except to the scanner).

The scanner then interprets the ballot and prepares a verifiable encryption of its contents, that is, a public key encryption of the voter's choices together with a proof that demonstrates the ballot validity and makes the ciphertext non-malleable. There are various techniques to perform this kind of operation, and ElectionGuard is an example of an SDK that has been used for that purpose on several occasions [3,4]. When the encryption operation is complete, the scanner prints a signed confirmation code (typically a hash of the ciphertext) in a way that is again visible to the voter and all observers.

Next, the declaration envelope is publicly opened and its contents are displayed so that observers can see whether the declaration reads `CAST` or `CHALLENGE`. The voter presents the declaration slip to the scanner (using a separate reader), so that it learns the voter choice.[6] If the declaration indicates `CAST`, the scanner simply drops the ballot in an attached ballot box, and the voter can take the printed and signed confirmation code home. The voter can show this confirmation code to any observer who can confirm that the signature is valid.

If the declaration indicates `CHALLENGE`, the scanner prints `CANCELLED` onto the ballot (or takes some other action such as clipping a corner) to indicate that the ballot is ineligible for subsequent casting. The scanner then releases the paper ballot to the voter and additionally prints on the paper containing the confirmation code its interpretation of the voter choices and the randomness (nonce) it used in the encryption process, together with a signature of the whole content. The voter can now verify that the choices have been accurately encoded and present the document to observers who can verify that it has been signed properly. The voter keeps both the ballot and the signed confirmation code and then restarts the process, beginning with the preparation of a new declaration.

After voting operations are complete, tallying operations begin. This includes an electronic tally computed from the scanned ballots and a verifiable crypto-

[5] Under normal circumstances, the declaration is known only to the voter at this stage of the protocol. However, this declaration will be subsequently revealed, and a coerced voter may therefore be compelled to make a declaration as dictated by a coercer.

[6] The declaration slip may contain a bar code or other features to facilitate its reading by the scanner.

graphic tally computed from the ciphertexts created during the scanning process.[7]

The authority $\mathcal{A}$ signs and publishes all the cryptographic data on a bulletin board,[8] so that voters can verify (using their ballot confirmation codes) that their votes have been included in the tally. Anyone can check that the tally is consistent with the set of confirmation codes published for cast ballots. All data corresponding to challenged ballots is published as well.

As mentioned, we assume that observers $\mathcal{O}$ are also present at the polling station. Should a voter be unhappy with any aspect of the voting process, a complaint should be lodged with the observers to request dispute resolution.

3.2 Discussion in the Context of Related Work

As far as we know, DROP is the first end-to-end verifiable in-person voting protocol that offers full dispute resolution and receipt freeness without making assumptions on hardware or network channels. DROP makes use of various techniques that have been used elsewhere, while also exhibiting uncommon features for protocols that offer dispute resolution, including the absence of a voter signing key.

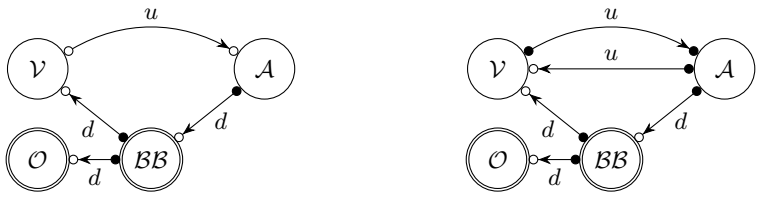
The DROP *topology.* The topology in which DROP executes can be placed within the framework of Basin, Radomirović, and Schmid [2], which is consistent with the fact that it offers (timely) dispute resolution. More precisely, DROP makes assumptions regarding parties and channels that are at least as strong as those that appear in Basin, Radomirović, and Schmid's topology T_6, depicted in Figure 1a (notation and the meaning of symbols is borrowed from [2] and is explained in the caption). The topology of DROP is shown in Figure 1b.

The main differences between the two topologies are as follows:

- DROP works in-person, so the channels between the voters and the authority are authentic.
- DROP assumes that voters can bring their paper ballots to the authority's scanner privately (the paper ballot content is not encrypted), so the channel from the voter to the authority is also private.
- DROP assumes that there is an undeniable channel from the authority to the voter: observers can see when the poll workers deliver declaration material and a blank ballot to a voter, and when the scanner delivers a ballot confirmation code to the voter.
- The topology shows trusted observers, which are not a strict requirement for the dispute resolution definition that we are proposing in this paper: our definition only says that, if there is an honest observer, then this observer will reach the correct conclusion in case of dispute.

[7] An RLA process such as VAULT [3] can optionally be added to limit the risk that a hand count of the paper ballots would lead to a different outcome.

[8] A bulletin board can be instantiated as a simple web page.



(a) Topology T_6 from [2]. (b) DROP's topology.

Fig. 1: Simple circles refer to untrusted parties and double circles refer to parties that are trusted for dispute resolution. Communication channels with a u label are undeniable—observers can see what is exchanged—and communication channels with a d are default channels, without any special properties. Channels with open circles at both ends are neither private nor authentic. Channels that start with a black circle are authentic, and channels with black circles at both ends are authentic and private.

The minimal topologies of [2] only show part of the picture, though: they are minimal for the resolution of three specific disputes that can occur in the context of recorded-as-cast verifiability, and these are also within DROP's scope. However, DROP also aims to address the disputes that may arise in the context of cast-as-intended verifiability. It would be interesting to explore minimal topologies for cast-as-intended verifiability, and to explore the extent to which the non-minimal assumptions made in DROP could somehow be relaxed in a meaningful way while maintaining the same security properties.

Dispute resolution in other protocols. A systematic survey of the voting protocols that have been designed with the goal of offering some form of dispute resolution, cast-as-intended verifiability, or receipt freeness is out of scope for this paper, but we would like to discuss DROP in the context of some related work.

Early deployments of end-to-end verifiable voting systems were already quite concerned about dispute resolution. Honest authorities may be worried that malicious voters could cast doubt on a correct election result by filing spurious claims, and observers may be concerned that dishonest authorities could simply dismiss complaints filed by honest voters by depicting them as malicious. For example, Scantegrity II [9], like DROP, relies on observers who see when ballots are challenged in order to solve disputes regarding ballot status. Large deployments of Helios 2.0 [1] had authorities producing various digitally signed pdf documents in order to address disputes regarding the election record, and also included in-person dispute resolution ceremonies. vVote [7], like DROP, has signed ballot confirmation codes produced in the polling station and expects that voters are able to check the validity of the signatures before leaving the polling station, either with the help of observers or using a dedicated mobile phone app.

Other protocols have been specially designed with dispute resolution in mind, but they typically do not address dispute resolution for cast-as-intended verifiability. For instance, the remote voting system sElect [13] offers dispute resolution

when the voting devices are trusted. Basin et al. [2] propose a mixnet-based voting protocol with dispute resolution in which voters use trusted personal devices to prepare their encrypted ballots, side-stepping challenges raised by disputes in cast-as-intended verification.

Receipt-free cast-or-challenge. It has long been recognized that many in-person voting protocols use the cast-or-challenge paradigm for cast-as-intended verifiability in a way that may damage receipt-freeness [12,16]. DROP has been expressly designed in a way that guarantees that the cast-or-challenge paradigm does not interfere with receipt-freeness.

4 Physical and Strctural Assumptions

In our DROP protocol outlined above, the voter authentication phase and the cryptographic parts related to universal verifiability are fairly standard and not specific to our system. We will therefore not develop these components further. Instead we focus on the steps taken by a voter to cast a ballot, and the current section describes the physical and structural assumptions we make in order to offer dispute resolution, cast-as-intended verifiability, and receipt-freeness.

Private declaration booth. The polling place has a private declaration booth that a voter must use to make a cast-or-challenge declaration. A voter enters the declaration booth with provided materials to prepare and commit to a cast-or-challenge decision. The voter emerges from the voting booth with a privately committed declaration, e.g., in the form of a sealed envelope that can later be verified to contain a single declaration slip.

Private voting booth. The polling place has a private voting booth for a voter to fill out a ballot and prepare it in a private manner such that it can be carried to a publicly observable voting device (scanner). The declaration booth may serve as voting booth as well, but this is not mandatory.

No recording device. Voters cannot carry image recording devices into the voting booth, making them unable to record evidence of the contents of their votes.

Public declaration table. The polling place has a public declaration table on which voters can place their declaration envelopes under the surveillance of poll workers and observers. These people keep track of who dropped what envelope—which may be marked by voters (perhaps with ink signatures) for identification purposes. When an envelope is opened, all observers can see that the envelope contained a single declaration slip and what declaration was made. If an envelope contains no declaration or more than one declaration, it is deemed illegitimate, and the voter must restart the process.[9]

[9] Aternatively, a *default* declaration can be substituted for any illegitimate declaration.

Voting device. The voting device (ballot scanner) is publicly visible to all observers but allows the completed ballot to be inserted without revealing the contents of the ballot (e.g., face-down scanning). Since a voting device has access to voter selections, it is assumed that the voting device is not compromised by or colluding with a coercer. Furthermore, we assume that the voting device cannot determine the contents of the cast-or-challenge declaration before issuing its ballot confirmation code.

Public-key infrastructure. The voting device has a signing key pair that enables it to print digital signatures of ballot confirmation codes and possibly of other data it produces. This means that there needs to be a public-key infrastructure that handles public keys for all voting devices. It is possible for voters to use personal devices or observers to verify that a signature is valid.

Voter knowledge. Voters know the election rules. In particular, they know the format of the blank ballots they are supposed to receive, and they know all the verification steps that they are supposed to perform.

Honest authority for privacy. It should also be noted that although end-to-end verifiability ensures that integrity of the election outcome does not depend on an honest election authority, voter privacy and protection from coercion do depend on an assumption of an honest authority. This is because a dishonest administrator has numerous ways to compromise voter privacy in any in-person election—including strategically-placed cameras, paper ballots which can be distinguished (e.g., with invisible ink or other subtle marks), and vote collection devices which collect and preserve more information than simple ballot contents.

5 Voting Process

We now describe in detail the steps taken by a voter who wishes to cast a ballot using the DROP protocol. Each eligible voter proceeds as follows after having been authenticated.

(1) **Receive cast-or-challenge declaration material:** The voter first receives material to make a cast-or-challenge decision. This can be human and machine readable `CAST` and `CHALLENGE` slips (or a single slip on which the voter marks a choice to cast or challenge) together with a sealable envelope.
(2) **Make cast-or-challenge declaration:** The voter enters the private declaration booth and commits to either cast or challenge the next ballot. When two slips are used, the voter inserts the slip reflecting the desired choice in the envelope and seals it, while destroying the second slip.[10] The voter may

[10] This process is similar to the voting process in various countries, where a voter enters the booth with pre-filled ballots supporting the various candidates, seals one ballot in an envelope, and drops the other ballots in the booth.

optionally sign by hand or otherwise mark the envelope for later identification. The voter leaves the private declaration booth and brings the envelope to the public declaration table.

(3) **Receive a ballot:** When it has been publicly observed that the voter left an envelope on the public declaration table, the voter receives from the poll workers a blank paper ballot for the election. The voter can inspect the paper ballot in order to see that it is valid.

(4) **Mark the ballot:** The voter enters the private voting booth and marks the paper ballot, then folds it so that it can be carried to the ballot scanner without making it possible for anyone else to see the ballot contents. If the voter does not fold the ballot, the ballot is taken and marked as cancelled by the authority; in this case, the voter may be given a fresh blank ballot and be sent back to the voting booth.

(5) **Scan the ballot:** The voter takes the paper ballot to a ballot scanner that can be publicly observed in a way that keeps the ballot private. The voter then inserts the ballot into the scanner. The scanner processes the ballot, encrypts the selections on the ballot, and produces a confirmation code committing to the vote encryptions. The confirmation code is printed together with a digital signature of it and made visible to the voter and all observers.

(6) **Open cast-or-challenge commitment:** The voter opens the cast-or-challenge declaration envelope publicly for the observers to see. It is also made visible to observers that the envelope did not contain more than one paper slip. If the envelope did not contain one and only one valid declaration slip, then $\mathcal{A}$ retrieves the paper ballot from the scanner and destroys it.[11] Otherwise, the voter brings the declaration slip to the scanner, which scans it and learns the cast-or-challenge choice made by the voter.

(7) **Cast decision:** If the declaration says `CAST`, the scanner pulls in the ballot and casts it by dropping it into a ballot box attached to it. Observers can see that the scanner is following this instruction properly. The voter takes the signed confirmation code and may verify or ask observers to verify whether the signature is valid. When the voter is convinced that the signed confirmation code has a valid signature, the voter can leave the polling place.

(8) **Challenge decision:** If the declaration says `CHALLENGE`, the scanner marks the ballot as challenged, returns it to the voter, and opens the commitment to the vote encryptions by printing the voter selections and encryption randomness next to the previously printed ballot confirmation code. The scanner also prints a signature of these opening values. The voter and the observers are now able to verify correct encryption of the selections on the ballot. Eventually, the voter is invited to restart the process from Step (1).

(9) **Bulletin board verification:** After leaving the polling place, voters can search for their ballot confirmation codes on the public bulletin board, both for their cast ballots and also for any ballots that they challenged.

[11] An alternative process would be to establish a default—either `CAST` or `CHALLENGE`—to be used for all instances when the envelope does not contain exactly one valid declaration slip.

6 Dispute Resolution in DROP

Following our description of the DROP voting flow, which focused on the cases where everything happens as expected, we now review the situations in which a voter may complain to observers and how each corresponding dispute can be resolved, proceeding step by step.

(1) **Receive cast-or-challenge declaration material:**
   - *Dispute 1:* The voter claims that the expected paper slips and envelope required to make a declaration were not received.
     Resolution 1: Observers $\mathcal{O}$ can inspect the slips and the envelope that were given to the voter. If they are valid, then $\mathcal{O}$ concludes that the voter $\mathcal{V}$ is dishonest. $\mathcal{O}$ concludes that the authority $\mathcal{A}$ is dishonest otherwise. It is possible to recover here by asking $\mathcal{A}$ to provide new declaration material to the voter.

(2) **Make cast-or-challenge declaration:**
   - $\mathcal{V}$ performs every step alone here, so there are no actions of $\mathcal{A}$ to dispute.

(3) **Receive a ballot:**
   - *Dispute 2:* $\mathcal{V}$ claims to have not received a blank paper ballot or to have received an incorrect blank paper from $\mathcal{A}$.
     Resolution 2: If $\mathcal{V}$ did not place a sealed envelope on the declaration table, observers $\mathcal{O}$ conclude $\mathcal{V}$ as dishonest. If $\mathcal{V}$ did place a sealed envelope on the declaration table and did not receive a valid blank paper ballot, then $\mathcal{O}$ concludes that $\mathcal{A}$ is dishonest. Observers $\mathcal{O}$ can insist that $\mathcal{V}$ should be given a valid blank ballot.

(4) **Mark the ballot:**
   - *Dispute 3:* $\mathcal{V}$ claims upon return of a declaration envelope that the envelope has been switched or altered.
     Resolution 3: Attentive observers $\mathcal{O}$ will reach a conclusion that supports or refutes this claim. But any action must be at the sole discretion of the poll workers as agents of the administrator $\mathcal{A}$.[12]
   - *Dispute 4:* $\mathcal{V}$ claims that $\mathcal{A}$ wants to cancel a ballot while the ballot had been folded and properly concealed.
     Resolution 4: Observers $\mathcal{O}$ concludee $\mathcal{V}$ is dishonest if the ballot was carried without having been properly folded and conclude that $\mathcal{A}$ is dishonest otherwise.

(5) **Scan the ballot:**
   - *Dispute 5:* $\mathcal{V}$ claims the scanner did not print a ballot confirmation code.
     Resolution 5: Observers $\mathcal{O}$ can see whether a ballot confirmation code was printed and side with $\mathcal{V}$ or $\mathcal{A}$ accordingly.

[12] The principal threat is that a dishonest administrator $\mathcal{A}$ can cause the contents of a ballot to be revealed by changing a `CAST` declaration to a `CHALLENGE`. However, a dishonest administrator already has numerous other ways to compromise ballot privacy; so this additional method does not materially change things. There is also a threat that a dishonest administrator $\mathcal{A}$ changes a `CHALLENGE` to a `CAST`—causing an unintended vote to be cast. In this case, honest observers $\mathcal{O}$ will recognize the misbeahvior of $\mathcal{A}$ and conclude that $\mathcal{A}$ is dishonest.

- *Dispute 6:* $\mathcal{V}$ claims that the ballot was deposited in the ballot box without a cast-or-challenge option being offered, or after a `CHALLENGE` instruction was issued.
  Resolution 6: Observers $\mathcal{O}$ can see whether the declaration envelope was opened and whether the declaration slip was scanned. If a `CAST` declaration slip was scanned, then $\mathcal{O}$ concludes that $\mathcal{V}$ is dishonest. Otherwise, $\mathcal{O}$ concludes that $\mathcal{A}$ is dishonest.
- *Dispute 7:* $\mathcal{V}$ claims that the scanner released the ballot and printed the opening information as if it received a `CHALLENGE` declaration, even though no declaration or a `CAST` declaration was given.
  Resolution 7: Observers $\mathcal{O}$ can see whether the declaration envelope was opened and whether the declaration slip was scanned. If a `CHALLENGE` declaration slip was scanned, then $\mathcal{O}$ concludes that $\mathcal{V}$ is dishonest. Otherwise, $\mathcal{O}$ concludes that $\mathcal{A}$ is dishonest.

(6) **Open cast-or-challenge commitment:**
- *Dispute 8:* $\mathcal{V}$ claims that $\mathcal{A}$ incorrectly asserted the cast-or-challenge declaration to be invalid.
  Resolution 8: Observers $\mathcal{O}$ can see the declaration envelope being opened by $\mathcal{V}$ and whether it contained one single declaration slip. $\mathcal{O}$ concludes $\mathcal{V}$ as dishonest if the envelope contains anything other than a single valid declaration slip and that $\mathcal{A}$ as dishonest otherwise.
- *Dispute 9:* $\mathcal{V}$ claims that $\mathcal{A}$ modified the content of the envelope containing the cast-or-challenge declaration.
  Resolution 9: Observers $\mathcal{O}$ can see if $\mathcal{A}$ opens an envelope containing a declaration. If $\mathcal{A}$ tampered with or switched an envelope, then $\mathcal{O}$ declares that $\mathcal{A}$ is dishonest. $\mathcal{O}$ declares that $\mathcal{V}$ is dishonest otherwise.

(7) **Cast decision:**
- *Dispute 10:* $\mathcal{V}$ claims that the ballot was not dropped in the ballot box even though a `CAST` declaration was scanned.
  Resolution 10: Observers $\mathcal{O}$ could see whether a `CAST` declaration was scanned and whether or not the ballot was dropped in the ballot box. $\mathcal{O}$ can side with $\mathcal{V}$ or $\mathcal{A}$ accordingly, as discussed above.
- *Dispute 11:* $\mathcal{V}$ claims that the ballot confirmation code that $\mathcal{V}$ received from the scanner does not contain a valid signature.
  Resolution 11: Observers $\mathcal{O}$ can verify the signature on the printed ballot confirmation code. If it is a valid signature, $\mathcal{O}$ concludes that $\mathcal{V}$ is dishonest. $\mathcal{O}$ concludes that $\mathcal{A}$ is dishonest otherwise.

(8) **Challenge decision:**
- *Dispute 12:* $\mathcal{V}$ claims that the ballot was not returned by the scanner, that the confirmation code opening does not reflect the selections made on the ballot, that the confirmation code is otherwise invalid, or that the signature is missing or invalid.
  Resolution 12: Observers $\mathcal{O}$ can see whether or not a `CHALLENGE` declaration was scanned and whether the printed confirmation code opening matches the paper ballot (if it is available) and bears a valid signature. If all of this is correct, $\mathcal{O}$ concludes that $\mathcal{V}$ is dishonest. Otherwise, $\mathcal{O}$ concludes that A is dishonest.

(9) **Bulletin board verification:**
   - *Dispute 13:* $\mathcal{V}$ claims a ballot is missing from the public bulletin board. *Resolution 13:* Observers $\mathcal{O}$ can ask $\mathcal{V}$ to show the associated ballot confirmation code slip signed by the scanner. If $\mathcal{V}$ can produce this signed ballot confirmation code for a missing ballot, $\mathcal{O}$ concludes that $\mathcal{A}$ his dishonest. Otherwise, $\mathcal{O}$ concludes that V is dishonest.

This list of disputes and solutions shows that DROP makes it possible for an honest observer to side with the right entity in any complaint raised by the voter. As discussed above, this however makes no guarantee regarding the verifiability, privacy, and receipt-freeness offered by DROP: this will be discussed below.

7 Verifiability and Receipt Freeness in DROP

7.1 Verifiability

We claim that DROP is an end-to-end verifiable voting system.

Individual verifiability. DROP offers cast-as-intended verifiability through the classical cast-or-challenge process in which the scanner must commit to the encrypted ballot before learning whether the ballot will be cast or challenged—forcing the scanner to provide proof of correct recording of the ballot. A central feature of DROP is that any honest complaint that a voter could make during this process can be interpreted as evidence that the authority is malicious. In addition, the paper ballots included in the ballot box can also be used to run an RLA and verify that the electronic record is consistent with the paper record.

DROP also offers recorded-as-cast verifiability by offering a bulletin board that voters can consult in order to confirm that their ballots have been properly recorded. The bulletin board displays both cast and challenged ballots, which enables the detection of ballot clashes that may be created by a malicious $\mathcal{A}$ [15].

Universal verifiability. Here, DROP relies on various standard techniques, typically based on homomorphic tallying or on mixnets and zero-knowledge proofs, in order to offer evidence that the reported tally is consistent with the electronic ballot record.

Eligibility verifiability. DROP relies on traditional authentication mechanisms provided at polling sites to make sure that only eligible voters can cast ballots and that they can do so only once. In some countries, lists of voters who cast ballots in an election are published. Such lists can be verified by observers and sometimes by voters as well. It is then possible to check that the number of ballots that appear in the electronic record matches the number of people who appear in the voter register, and voters who did not vote may also be able to check that no vote was recorded in their names.

7.2 Receipt Freeness

Two ingredients are used to make DROP receipt free.

First, we made two important assumptions. We stated that the voter does not have any recording device that could be used in a voting booth to offer evidence of vote selections to a third party. We also stated that the ballot scanner does not collude with a coercer. It would indeed be very easy for a corrupted ballot scanner to leak information about a voter's selections through a direct channel with a confederate or a side channel such as the randomness used for computing the ciphertexts that are posted on the bulletin board (e.g., if the scanner produces one ciphertext per choice, it could choose the randomness so that the last bit of each ciphertext matches the corresponding plaintext). More generally, we stated that voters are not considered to be under coercion by $\mathcal{A}$: a malicious $\mathcal{A}$ could coerce $\mathcal{V}$ into never issuing any challenges raising any complaints, and would then be able to arbitrarily modify/substitute ballots.

These assumptions are however insufficient to guarantee receipt freeness w.r.t. other entities, and various attacks have been demonstrated that build on the cast-or-challenge process to create receipts [12,16]. These attacks all proceed by somehow forcing the voter to declare whether to cast or challenge a ballot *after* making selections on the ballot. DROP prevents such attacks by forcing the voter to commit to a declaration *before* receiving a blank paper ballot. Coercers can make demands as to whether a particular ballot be cast or challenged, but these demands will be known to voters and executed before voters make their selections. This allows voters to make any selections they wish on cast ballots while displaying on challenged ballots whatever selections are demanded by coercers.

8 Conclusion

We have demonstrated the DROP protocol which allows honest observers of an end-to-end verifiable election to reach a correct conclusion to any dispute between voters and the election administrators—without any assumptions on the hardware or voting processes. The protocol is admittedly cumbersome, but not so much so as to make it unrealizeable. Nevertheless, this work should be regarded as an existence proof more than as a serious proposal for use in practice.

Until now, it was an open question whether dispute resolution was even possible without equipment or process assumptions. With that question settled, the door is now open for practical improvements towards making strong dispute resolution a seemless part of verifiable election protocols.

References

1. Adida, B., de Marneffe, O., Pereira, O., Quisquater, J.: Electing a university president using open-audit voting: Analysis of real-world use of Helios. In: 2009 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections, EVT/WOTE '09. USENIX Association (2009)

2. Basin, D.A., Radomirovic, S., Schmid, L.: Dispute resolution in voting. In: 33rd IEEE Computer Security Foundations Symposium, CSF 2020. pp. 1–16. IEEE (2020)
3. Benaloh, J., Foote, K., Stark, P.B., Teague, V., Wallach, D.S.: Vault-style risk-limiting audits and the Inyo County pilot. IEEE Secur. Priv. **19**(4), 8–18 (2021)
4. Benaloh, J., Naehrig, M., Pereira, O., Wallach, D.S.: Electionguard: a cryptographic toolkit to enable verifiable elections. In: 33rd USENIX Security Symposium, USENIX Security 2024. USENIX Association (2024)
5. Bernhard, M., Benaloh, J., Halderman, J.A., Rivest, R.L., Ryan, P.Y.A., Stark, P.B., Teague, V., Vora, P.L., Wallach, D.S.: Public evidence from secret ballots. In: Electronic Voting - Second International Joint Conference, E-Vote-ID 2017. Lecture Notes in Computer Science, vol. 10615, pp. 84–109. Springer (2017)
6. Bougon, M., Chabanne, H., Cortier, V., Debant, A., Dottax, E., Dreier, J., Gaudry, P., Turuani, M.: Themis: An on-site voting system with systematic cast-as-intended verification and partial accountability. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022. pp. 397–410. ACM (2022)
7. Burton, C., Culnane, C., Heather, J., Peacock, T., Ryan, P.Y.A., Schneider, S.A., Teague, V., Wen, R., Xia, Z., Srinivasan, S.: Using Prêt à Voter in Victoria State Elections. In: 2012 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections, EVT/WOTE '12. USENIX Association (2012)
8. Carback, R., Chaum, D., Clark, J., Conway, J., Essex, A., Herrnson, P.S., Mayberry, T., Popoveniuc, S., Rivest, R.L., Shen, E., Sherman, A.T., Vora, P.L.: Scantegrity II municipal election at Takoma Park: The first E2E binding governmental election with ballot privacy. In: 19th USENIX Security Symposium. pp. 291–306. USENIX Association (2010)
9. Chaum, D., Carback, R.T., Clark, J., Essex, A., Popoveniuc, S., Rivest, R.L., Ryan, P.Y.A., Shen, E., Sherman, A.T., Vora, P.L.: Scantegrity ii: End-to-end verifiability by voters of optical scan elections through confirmation codes. IEEE Transactions on Information Forensics and Security **4**(4), 611–627 (2009)
10. Chaum, D., Florescu, A., Nandi, M., Popoveniuc, S., Rubio, J., Vora, P.L., Zagórski, F.: Paperless independently-verifiable voting. In: E-Voting and Identity - Third International Conference, VoteID 2011, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7187, pp. 140–157. Springer (2011)
11. Kaczmarek, T., Wittrock, J., Carback, R., Florescu, A., Rubio, J., Runyan, N., Vora, P.L., Zagórski, F.: Dispute resolution in accessible voting systems: The design and use of Audiotegrity. In: E-Voting and Identify - 4th International Conference, VoteID 2013. Lecture Notes in Computer Science, vol. 7985, pp. 127–141. Springer (2013)
12. Kelsey, J., Regenscheid, A., Moran, T., Chaum, D.: Attacking paper-based E2E voting systems. In: Towards Trustworthy Elections, New Directions in Electronic Voting. Lecture Notes in Computer Science, vol. 6000, pp. 370–387. Springer (2010)
13. Küsters, R., Müller, J., Scapin, E., Truderung, T.: select: A lightweight verifiable remote voting system. In: IEEE 29th Computer Security Foundations Symposium, CSF 2016. pp. 341–354. IEEE Computer Society (2016)
14. Küsters, R., Truderung, T., Vogt, A.: Accountability: definition and relationship to verifiability. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010. pp. 526–535. ACM (2010)
15. Küsters, R., Truderung, T., Vogt, A.: Clash attacks on the verifiability of e-voting systems. In: IEEE Symposium on Security and Privacy, SP 2012. pp. 395–409. IEEE Computer Society (2012)

16. Rønne, P.B., Finogina, T., Herranz, J.: Expanding the toolbox: Coercion and vote-selling at vote-casting revisited. In: Electronic Voting - 9th International Joint Conference, E-Vote-ID 2024, Tarragona, Spain, October 2-4, 2024, Proceedings. Lecture Notes in Computer Science, vol. 15014, pp. 141–157. Springer (2024)
17. Zagórski, F., Carback, R., Chaum, D., Clark, J., Essex, A., Vora, P.L.: Remotegrity: Design and use of an end-to-end verifiable remote voting system. In: Applied Cryptography and Network Security - 11th International Conference, ACNS 2013. Lecture Notes in Computer Science, vol. 7954, pp. 441–457. Springer (2013)