Privacy-Preserving Dispute Resolution in the Improved Bingo Voting

Rosario Giustolisi(✉) and Alessandro Bruni

IT University of Copenhagen, Copenhagen, Denmark
{rosg,brun}@itu.dk

Abstract. Dispute resolution mechanisms are important components of voting schemes, deterring a voting authority to change the election outcome as any alteration can be *proved* by such mechanisms. However, these mechanisms are useless if not triggered by voters, who should not have to choose to either raise a dispute or keep their vote private. Hence, voting schemes should include privacy-preserving dispute resolution.

In this work, we advance the formal analysis in the symbolic model of an improved version of the Bingo Voting scheme, whose enhancements include privacy-preserving dispute resolution mechanisms. Most of our analysis of several verification, dispute resolution, and privacy properties is done automatically using ProVerif, which we complement with manual induction proofs as necessary. We find that the scheme meets some properties only if one makes additional trust assumptions to those stated in [6]. For example, we find that dispute resolution is met assuming an honest voting authority. Moreover, our work provides an understanding of privacy-preserving dispute resolution in general, which can be beneficial to similar analyses of other voting schemes.

1 Introduction

Consensus on the election outcome and vote privacy are two main pillars of voting schemes. On the one hand, voting schemes that fail in achieving consensus are worthless, hence a voting scheme should provide high confidence in the result of the election despite voters do not necessarily trust the voting authority. On the other hand, failing to provide vote privacy opens to effective manipulation of voters and to control the outcome of the election. Intuitively, consensus on the election outcome and vote privacy seem to be two contrasting properties: more evidence would increase confidence in the election outcome at the risk of fewer privacy guarantees. Recent work [12] has shown that vote privacy implies individual verifiability. However, individual verifiability only enables a voter to *check* that her ballot has been counted, but not to publicly *prove* it. This means that a dishonest voting authority may still change the election outcome and there is no public evidence that could prove so.

One can deter a voting authority from changing the election outcome by introducing dispute resolution mechanisms that enable a voter to prove to any observer that her vote was not included in the tally. This should be possible

© Springer Nature Switzerland AG 2020
R. Krimmer et al. (Eds.): E-Vote-ID 2020, LNCS 12455, pp. 67–83, 2020.
https://doi.org/10.1007/978-3-030-60347-2_5

for the voter without giving up vote privacy, hence dispute resolution should be privacy-preserving.

In this paper, we provide a formal analysis of an improved version of the Bingo Voting scheme [6,18], which aims at ensuring privacy-preserving dispute resolution mechanisms among other features. We check automatically several verification, dispute resolution, and privacy properties in ProVerif, and identify the additional trust assumptions required by the scheme respect to the ones stated in [18]. To the best of our knowledge, this work represents the first formal treatment of the improved version of Bingo Voting. We provide the precise algorithm that enables an observer to dispute the outcome of an election and details the aftermath of a privacy-preserving dispute resolution at the voting phase, considering different mitigation scenarios. The outcome of our analysis pinpoints the difficulties in designing privacy-preserving dispute resolution mechanisms and can be useful for other voting schemes.

Outline. This paper is organised as follows. Section 2 details the improved Bingo Voting scheme as well as its properties and trust assumptions. Section 3 presents the formal analysis of verification, dispute resolution, and privacy properties in the improved Bingo Voting. Then, it discusses the outcome of the analysis. Section 4 presents some related work. Finally, Sect. 5 concludes the paper.

2 Background

Bingo Voting was originally proposed by Bohli, Müller-Quade and Röhrich in 2007 [7]. The underlying idea of Bingo Voting is that each voter receipt assigns to each candidate either a *dummy* random number or a *fresh* random number. The voting authority generates the dummy random numbers before the voting phase starts. A trusted random generator (TRNG) creates the fresh random numbers during the voting phase. The voting machine then assigns the fresh random number to the candidate chosen by the voter and a different dummy random number to each of the remaining candidates.

In Bohli et al. [6] and later in Henrich [18], several improvements are proposed to the original Bingo Voting system, including extensions to use Bingo Voting for more complex elections and ways to address usability limitations. In this paper, we consider two key improvements, hence we will refer to the resulting system as the *improved Bingo Voting*. The first improvement that we consider consists of two privacy-preserving dispute resolution procedures, one at the voting and the other at tallying. The other improvement regards the optimisation of the proof of correct distribution of dummy votes, which in the improved version is done after the voting phase. Figure 1 presents a message sequence chart of the scheme. The details of the scheme are outlined below.

Before the voting phase, the voting authority generates and publishes a set of *dummy votes*. A dummy vote consists of a pair of Pedersen commitments that hide both the dummy random number and the assigned candidate. Each candidate receives the same number of dummy votes, that is, the number of

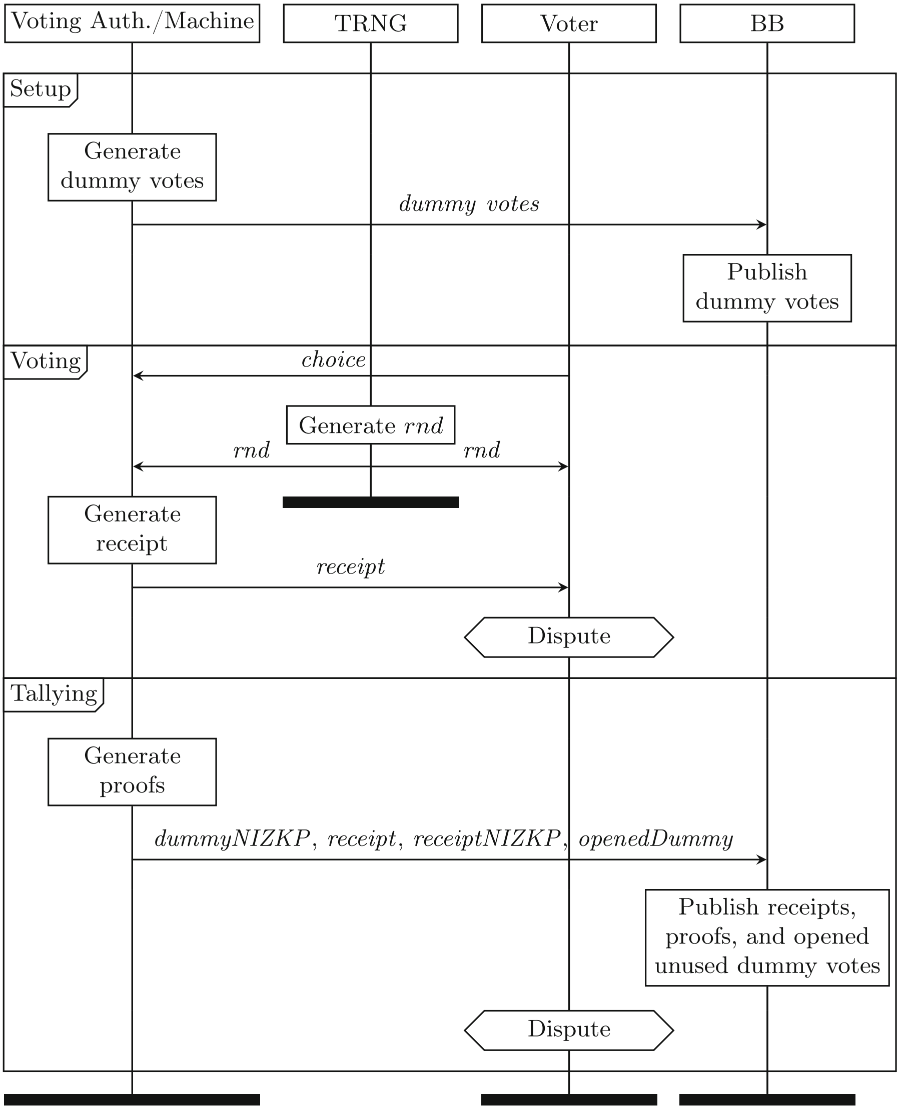
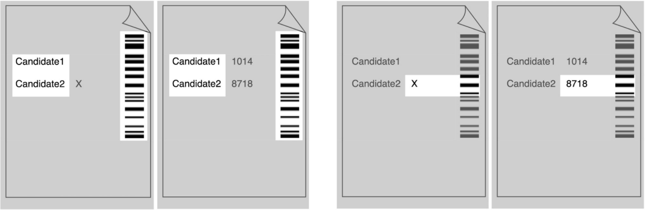


Fig. 1. Message sequence chart of the improved Bingo Voting

registered voters. Thus, the total number of generated dummy votes is equal to the product of the number of voters and the number of candidates.

Inside the voting booth, a display shows the fresh random number generated by the TRNG. The voter records her choice on a paper ballot and feeds it into the voting machine, which is equipped with a scanner-based interface. The voting machine scans the paper ballot and generates a receipt such that the fresh random number is printed next to the name of the candidate chosen by the voter.

Unused dummy random numbers, which the voting authority generated before the voting phase, are instead printed next to any other candidates. The voting machine also prints an identical barcode onto both paper ballot and receipt, and keeps the paper ballot inside a special compartment unless the voter decides to raise a dispute should she receive an incorrect receipt.



(a) Alignment sleeve (b) Encoding sleeve

Fig. 2. The privacy sleeves for the privacy-preserving dispute resolution at voting

In the case of a dispute, the voter can use two different pairs of privacy sleeves to prove that the printed receipt is incorrect, without revealing the way she voted. Each pair of privacy sleeves is to be used with both paper ballot and receipt. The first type of privacy sleeve leaves uncovered candidate names and the barcodes (see Fig. 2a) and enables a third party to check whether the candidates are not placed identically in respect to the barcode on the paper ballot and the receipt. The second type of privacy sleeve leaves uncovered the marking area for one candidate on the paper ballot and one row of random numbers on the receipt (see Fig. 2b). This enables a third party to check whether there is a discrepancy between the voter choice and the receipt as the printed random number differs from the one displayed on the TRNG.

At tallying, the voting authority publishes the final result of the election along with the following sets of data on an append-only bulletin board

- A non-interactive zero-knowledge proof of correct distribution of dummy votes showing that each candidate gets the same number of dummy votes.
- A non-interactive zero-knowledge proof for each receipt showing that it contains the correct amount of dummy random numbers and that each dummy random number is assigned to the right candidate.
- The list of all printed receipts.
- The list of opened unused dummy votes, which determines how many votes each candidate has received.

Since all the receipts are published, every voter can verify whether their vote is correctly counted. If not, they can raise a privacy-preserving dispute resolution at tallying proving that their receipt has not been published. Morever, any observer can check the correctness of the election outcome by verifying that the tally is indeed the sum of all votes cast.

Properties. The improved Bingo Voting aims at the following properties:

- *Individual verifiability*: a voter can check that the receipt encodes her vote.
- *Privacy-preserving dispute resolution at voting*[1]: a voter can prove that the receipt incorrectly encodes her vote, without revealing her vote.
- *Privacy-preserving dispute resolution at tallying*[2]: a voter can prove that her receipt is not in the bulletin board, without revealing her vote.
- *Global verification*: anyone can prove that the tally is incorrectly computed.
- *Vote privacy*: No one knows how the voter votes.
- *Receipt freeness*: The voter has no evidence proving how she voted.
- *Coercion resistance*: A voter deviating from the intended voting process receives no evidence that may be used to prove how she voted.

The improved Bingo Voting requires a number of trust assumptions to meet the security properties outlined above. The most important are that only eligible voters get access to a voting machine and that each voter casts a single ballot. Also, it is assumed that voters are unobserved as they cast their ballot, which is known as the *voting booth assumption*. Bulletin board (BB) and TRNG are always considered uncorrupted. For vote privacy, it can be assumed that both voting authority and the voting machine can be dishonest as soon as they do not communicate. For receipt freeness and coercion resistance, the voting authority should be uncorrupted, and the voting machine should not be able to communicate with an attacker. In the next section, we analyse the improved Bingo Voting in ProVerif to determine any necessary additional assumptions.

3 Formal Analysis

ProVerif [5] allows one to analyse reachability and equivalence-based properties in the symbolic attacker model. We chose ProVerif mainly because its input language fits well with our approach in modelling the verification and dispute resolution mechanisms. It is also one of the few tools that enable the automated analysis of privacy properties using observational equivalence. The input language of ProVerif is the applied π-calculus [1], which the tool automatically translates to Horn clauses. Cryptographic primitives can be modelled by means of equational theories. An equational theory E describes the equations that hold on terms built from the signature. Terms are related by an equivalence relation $=$ induced by E. For instance, the equation $dec(enc(m, pk(k)), k) = m$ models an asymmetric encryption scheme. The term m is the message, the term k is the secret key, the function $pk(k)$ models the public key, the term enc models the encryption function, and the term dec models the decryption function.

[1] In our formal analysis we separate this property into *dispute resolution at voting*, which checks the correctness of the test as a reachability property, and *vote privacy after a dispute*, which checks vote privacy in terms of observational equivalence.

[2] Since all the receipts are eventually published, vote privacy implies that dispute resolution at tallying is privacy-preserving.

The equational theory for the improved Bingo Voting is described in Table 1. It includes the equations for digital signature (in our case *checksign* returns the signed message only if one uses the correct verification key, and it fails otherwise), Pedersen commitment, dummy vote, and non-interactive zero-knowledge proofs (NIZKP) that prove the correctness of dummy votes and published receipts. To prove the correctness of the dummy votes, the voting authority uses the function *zkp1* showing that the content of the second commitment of each dummy vote is equal to the list of the two candidates cA and cB. The function *zkp2* allows the voting authority to prove that the content of a receipt $(cA, cB, Rtrg, rX)$ is identical to the content of the used dummy vote pair $dvp(com(rX, r1), com(cB, cr1))$ and to random number displayed on the TRNG $(Rtrg)$, which is hidden into the fresh dummy vote pair $dvp(com(Rtrg, tr), com(cA, cr0))$. An auditor can check both proofs against the dummy vote pairs and the receipts published on the BB.

We specify the processes modelling voting authority, voter, TRNG, and bulletin board into a ProVerif library and reuse it to check each property. This guarantees that all the properties are checked against the same model of the improved Bingo Voting.

Table 1. Equational theory modelling the improved Bingo Voting

Primitive	Equation
Digital signature	$checksign(sign(m, ssk), spk(ssk)) = m$
Commitment & Dummy vote	$openCommit(com(val, r)) = (val, r)$ $openDummyVote(dvp(com0, com1)) = (com0, com1)$
NIZKP dummy vote	$\mathit{checkzkp1}(cA, cB, dvp(com0A, com(cA, cr0)),$ $dvp(com0B, com(cB, cr1)),$ $\mathit{zkp1}(cA, cB, cr0, cr1, com(cA, cr0), com(cB, cr1) = OK$
NIZKP receipt (candidate A)	$\mathit{checkzkp2}(cA, cB, Rtrg, rX, dvp(com(Rtrg, tr), com(cA, cr0)),$ $dvp(com(rY, r0), com(cA, cr0)), dvp(com(rX, r1), com(cB, cr1)),$ $\mathit{zkp2}(cA, cB, rX, dvp(com(Rtrg, tr), com(cA, cr0)),$ $dvp(com(rX, r1), com(cB, cr1)),$ $cr1, cr0, r1, tr)) = OK$
NIZKP receipt (candidate B)	$\mathit{checkzkp2}(cA, cB, rY, Rtrg, dvp(com(Rtrg, tr), com(cB, cr1)),$ $dvp(com(rY, r1), com(cA, cr0)), dvp(com(rX, r1), com(cB, cr1)),$ $\mathit{zkp2}(cA, cB, rX, dvp(com(Rtrg, tr), com(cB, cr1)),$ $dvp(com(rY, r1), com(cA, cr0)),$ $cr0, cr1, r1, tr)) = OK$

3.1 Verification and Dispute Resolution

All the verification and dispute resolution properties of the improved Bingo Voting can be modelled as reachability properties. In line with the verification approach defined in [8,24], we identify the *tests* that decide whether a goal of the improved Bingo Voting fails. We then check that each of the tests meets soundness, completeness, and sufficiency conditions, as outlined in Table 2.

Table 2. $\mathcal{A}(\cdot)$: external attacker; $\mathcal{A}(\mathit{VA})$: attacker controlling the voting authority; V: voter instances; V_{test}: voter instance running the test; τ: a trace representing a run of the improved Bingo Voting; $\mathcal{T}$: the set of all traces. BB and TRNG are always honest according to the improved Bingo Voting assumptions.

	Strategy			Condition
	Individual verification	Dispute resolution	Global verification	
(Soundness)	$\mathcal{A}(\cdot)$	$\mathcal{A}(V)$	$\mathcal{A}(V)$	$\forall \tau \in \mathcal{T} \mid \mathit{goal}$ holds in $\tau \implies \mathit{test}(\tau) : \texttt{true}$
(Completeness)	$\mathcal{A}(\mathit{VA}, V \setminus V_{test})$	$\mathcal{A}(\mathit{VA}, V \setminus V_{test})$	$\mathcal{A}(\mathit{VA}, V)$	$\forall \tau \in \mathcal{T} \mid \mathit{test}(\tau) : \texttt{true} \implies \mathit{goal}$ holds in τ
(Sufficiency)	$\mathcal{A}(\mathit{VA}, V \setminus V_{test})$	$\mathcal{A}(\mathit{VA}, V \setminus V_{test})$	$\mathcal{A}(\mathit{VA}, V)$	$\exists \tau \in \mathcal{T} \mid \mathit{test}(\tau) : \texttt{false}$

Soundness guarantees that if the goal holds, then the test always succeeds. For dispute resolution and global verification, it means that an honest voting authority should never be blamed by any test. Note that individual verification requires a different verification strategy than dispute resolution, as the former considers no inside attacker since the verification is based on (the honest) voter's knowledge of the way she voted. In fact, individual verification does not give the voter a way to prove that the voting authority misbehaved. Conversely, in case of dispute resolution or global verification, in which tests are decided upon public information, we consider no honest voters, who may try to feed the tests with incorrect information. We prove that an honest voting authority cannot be unfairly blamed.

Completeness guarantees that whenever a test does not blame the voting authority, then the goal holds. Note that this is logically equivalent to saying that whenever a goal does not hold, then the test blames the voting authority. Thus, we check that a dishonest voting authority cannot feed the tests with incorrect information so that the test succeeds but the goal fails. The verification strategy for completenessregarding global verification is different from the one regarding individual verification and dispute resolution: in principle, global verification should hold even if all voters are dishonest as any election observer can run the test. However, as we shall see later, global verification can provide only guarantees up to dishonest voters.

While soundnessand completenessare conforming to [8], we introduce a third condition, *sufficiency*, which formalises that the misbehaviour of selected parties alone is sufficient to make the test fail. Without this condition, a protocol that does not permit any violation might still fulfil criteria to blame a party [23].

The conditions described in Table 2 show that the main difference between individual verification and dispute resolution boils down to be the verification strategy for checking soundness. Thus, a protocol that is dispute free for a specific goal is also individually verifiable for that goal. This is the case for individual verification and dispute resolution at voting for the improved Bingo Voting.

Due to space limitations, we only discuss the details of the dispute arising due to the global verification test in the improved Bingo Voting. The ProVerif code for all properties is available in [16]. Global verification enables any observers, including those who have not participated in the election at all, to verify the correctness of the election outcome. Global verification ensures that all candidates have received the same number of dummy votes and that for each receipt all but one candidate lose one dummy vote. This is the most complex test in improved Bingo Voting and requires the voting authority to release some information. The original paper presenting the improved Bingo Voting does not detail a specific algorithm for the test, thus we propose the test as defined in Algorithm 1. Our test considers two candidates, cA and cB. The input data of the test is published by the voting authority on the bulletin board.

We can define the goal for global verification $goal_{gv}$ as follows. Let us consider the set of all voters V of type $\mathcal{V}$, the set of voters' choices C of type $\mathcal{C}$, the set of candidates K of type $\mathcal{K}$, the set of honest voters $V_h \subseteq V$, and the set of choices of honest voters $C_h \subseteq C$. Let us now consider the relation $\texttt{Choice}$ as the votes accepted by the bulletin board according to the published receipts, linking voters to their choices such that $\texttt{Choice} \subseteq V \times C$. Similarly, consider the relation $\texttt{Choice\_h}$ that links honest voters to their choices such that $\texttt{Choice}_\texttt{h} \subseteq V_h \times C_h$. Let $\texttt{Count}$: $(\mathcal{V} \times \mathcal{C}) \rightarrow (\mathcal{K} \times \mathbb{N})$ be an ideal counting function that returns the number of votes for each candidate. We can say that the *$goal_{gv}$ holds in τ* if $\texttt{Choice}_\texttt{h} \subseteq \texttt{Choice}$ and the election result is equal to $\texttt{Count}(\texttt{Choice})$.

All our proofs consider an unbounded number of voters. While ProVerif can automatically prove sufficiency for global verification, it is not possible to prove soundnessand completenesssince, according to Algorithm 1, we need to iterate over all receipts, but ProVerif does not support loops. We thus prove the base case in ProVerif, in which we consider only one published receipt. Then, we provide a manual induction proof that generalises the ProVerif results to the general case with an arbitrary number of published receipts.

ProVerif proves soundnessand completenesswhen only one published receipt is considered. To prove the general case that considers an unbounded number of published receipt, it is necessary to show that

$$test(\tau) : \texttt{true} \Leftrightarrow \texttt{Choice}_\texttt{h} \subseteq \texttt{Choice} \wedge \text{ the election results is equal to } \texttt{Count}(\texttt{Choice})$$

It can be assumed that the number of published receipts is equal to the number of the published dummy votes and of the opened dummies. Any observer can check that these numbers coincide by looking at the bulleting board.

Theorem 1. *Let $test_k(\cdot)$ be the test applied to an execution that considers k receipts; let $test_k(\cdot) \rightarrow^* \texttt{true}$ denote the test that outputs $\texttt{true}$ after some steps; let τ be a trace that has n receipts; let τ_j be a version of τ that only considers the j^{th} receipt that is associated with a honest voter i_j and corresponding choice c_j. For soundness, we prove that*

$$\forall 1 \leqslant i \leqslant n : test_1(\tau_j) \rightarrow^* \texttt{true} \Rightarrow (i_j, c_j) \in \texttt{Choice} \wedge \textit{ the election results is equal to } \texttt{Count}(\texttt{Choice})$$

For completeness, we prove that

$$\forall 1 \leqslant i \leqslant n : (i_j, c_j) \in \mathtt{Choice} \wedge \ \textit{the election results is equal to}$$
$$\mathtt{Count(Choice)} \Rightarrow \mathit{test}_1(\tau_j) \rightarrow^* \mathtt{true}$$

Proof. $\mathit{test}_n(\tau)$ checks all the receipts, dummy votes, and proofs published in the bulletin board as defined in Algorithm 1. Similarly, the test $\forall 1 \leqslant j \leqslant n : \mathit{test}_{\mathit{1}}(\tau_j)$ does the same check for the j^{th} entry in the bulletin board. It follows that

$$\mathit{test}_n(\tau) \rightarrow^* \mathtt{true}$$
$$\Downarrow$$
$$\forall 1 \leqslant j \leqslant n : \mathit{test}_{\mathit{1}}(\tau_j) \rightarrow^* \mathtt{true}$$
$$\Downarrow_{(by\ ProVerif)}$$
$$\forall 1 \leqslant j \leqslant n : (i_j, c_j) \in \mathtt{Choice} \wedge \ \text{the election results is equal to } \mathtt{Count(Choice)}$$
$$\Downarrow$$
$$\mathtt{Choice}_{\mathrm{h}} \subseteq \mathtt{Choice} \wedge \ \text{the election results is equal to } \mathtt{Count(Choice)}$$

which proves soundnessalso for the general case.

$$\mathtt{Choice}_{\mathrm{h}} \subseteq \mathtt{Choice} \wedge \ \text{the election results is equal to } \mathtt{Count(Choice)}$$
$$\Downarrow$$
$$\forall 1 \leqslant j \leqslant n : (i_j, c_j) \in \mathtt{Choice} \wedge \ \text{the election results is equal to } \mathtt{Count(Choice)}$$
$$\Downarrow_{(by\ ProVerif)}$$
$$\forall 1 \leqslant j \leqslant n : \mathit{test}_{\mathit{1}}(\tau_j) \rightarrow^* \mathtt{true}$$
$$\Downarrow$$
$$\mathit{test}_n(\tau) \rightarrow^* \mathtt{true}$$

which proves completeness also for the general case.

3.2 Privacy

Like in the verification of the verifiability and dispute resolution properties, we prove privacy by encoding the protocol into one ProVerif library – with a few modifications compared to the previous one – and then check privacy of different setups. The main practical change required for proving privacy is to remove the channel that voter, voting authority, and bulletin board use to feed the test with the evidence, and let the attacker read all public data and impersonate misbehaving parties, including an unbounded number of dishonest voters. As the improved Bingo Voting requires that voters are unobserved as they cast their vote, all communications between honest voters, the voting machine, and the TRNG are done over private channels.

```
Algorithm 1: Global Verification
Data: $cA, cB, receipt : (cx, cy, rx, ry, barcode), zkp1, dummy\_vote, zkp2,$
$new\_dummy, opened\_dummy : (ca, ra)$
foreach $receipt$ in BB do
    if checkzkp1($cA, cB, dummy\_vote, zkp1$) = OK then
        if $cx = cA \wedge cy = cB \wedge$
         checkzkp2($cA, cB, rx, ry, new\_dummy, dummy\_vote, zkp2$) = OK $\wedge$
         $rx \neq ry \wedge rx \neq cx \wedge rx \neq cy \wedge ry \neq cx \wedge ry \neq cy \wedge$ then
            if dummy($ca, ra$) $\in dummy\_vote \wedge ra \neq rx \wedge ra \neq ry$ then
                return true
            else
                return false
        else
            return false
    else
        return false
```

In the privacy setting, we observe two voters in particular, hence the bulletin board needs to shuffle the votes specifically to avoid trivial attacks to privacy. We check vote privacy, receipt freeness, and coercion resistance considering an honest voting authority. We also check vote privacy, and vote privacy of disputed receipt at the voting phase consider a dishonest voting authority. First, we check whether vote privacy holds in the improved Bingo Voting. Specifically, we check that if two honest voters swap their votes in two different runs of the protocol then the attacker cannot distinguish the two resulting systems as in [22]:

$$S[V_A\{^a/_v\} \mid V_B\{^b/_v\}] \approx_l S[V_A\{^b/_v\} \mid V_B\{^a/_v\}]$$

Similarly, we check whether vote privacy holds after a dispute at the voting phase. We let the honest voters reveal the fresh random number obtained by the trusted random number generator and the dummy random number on the receipt that is revealed by the privacy sleeve.

To check receipt freeness, we additionally let the voters publish their receipts on the public channel, and verify that privacy still holds:

$$S[V_A\{^a/_v\} \mid V_B\{^b/_v\}] \approx_l S[V' \mid V_B\{^a/_v\}]$$

where V' is a process such that $V'^{\backslash \mathsf{out}(chc,\cdot)} \approx_l V_A\{^b/_v\}$, i.e. V' is the process that acts like V_A voting for candidate B, but pretends to cooperate with the attacker.

Finally, to check whether the scheme is coercion resistant, we set up the protocol so that one of the voters receives the instruction on how to vote from the attacker and then provides the receipt to the attacker. We check that

$$S[C[V_A\{^?/_v\}^{c_1,c_2}] \mid V_B\{^a/_v\}] \approx_l S[C[V'] \mid V_B\{^c/_v\}]$$

where $V_A\{^?/_v\}^{ch,a}$ is the coerced voter process that votes for candidate B, no matter their original intention, reveals all its private information to the attacker via channels c_1, c_2, while V_B is the other voter process intended to balance the resulting votes, that is, if V_A votes for candidate A, then V_B votes for candidate B and vice versa. Note that with the setup described here there is a trivial attack, which only appears in the model, as the bulletin board should not reveal whether the votes were swapped or not. In practice, this is done by shuffling. Thus, we let the bulletin board swap the order of published ballots if and only if the voters actually swap their choice following the attacker's instruction.

3.3 Findings

ProVerif proves individual verification and both dispute resolution at voting and at tallying automatically. It also proves global verification for one receipt, then we provide a manual inductive proof for the unbounded case. The outcome of our analysis shows that the improved Bingo Voting meets some properties only if one makes additional assumptions to the ones already defined in [6,18]. The additional assumptions are reported in Table 3. For dispute resolution at voting, we need to assume that the test does not blame the voting authority if the barcode printed on the paper ballot does not match with the one printed on the receipt. This avoids an attack due to a dishonest voter handing her receipt to another voter [8]. Without this assumption, the latter, isolated in the voting booth, may swap the receipt printed by the voting machine with the ones handed by the dishonest voter, leading to a successful blaming of the voting authority.

We also need to make additional assumptions for proving global verification. As already noted by in [24], it is only possible to have global verification up to the votes of dishonest voters since a dishonest voting authority can alter votes cast by such voters without being detected. Moreover, we found that honest voters should check that their receipts are well-formed at voting and at tallying, and raise disputes otherwise.

As regards privacy properties, we found that vote privacy, receipt freeness, and coercion resistance hold if the voting authority is honest and the voting machine cannot decide which dummy vote should be assigned to which receipt. This can be achieved by prearranging dummy votes in *clusters* [18], which limits the voting machine's choice on selecting the dummy votes. Considering two candidates, each cluster contains two dummy votes, one per candidate. The voting authority publishes the clusters in the same order in which the voting machine uses them for the receipts. The voting authority can prove in zero-knowledge that each receipt used the dummy votes from the expected cluster. However, the verification process of the correct order of clusters requires that the bulletin board publishes the receipts as they are issued. Revealing the order in which the receipts are issued may not be acceptable for many elections. In fact, ProVerif finds that if the bulletin board does not randomly shuffle the receipts before publishing them, the voting authority can easily break vote privacy by just looking at the order of voters, which is normally available in the voter registration record at the polling place. Thus, for vote privacy, it is not enough assuming that a dishonest voting authority does not communicate with a dishonest voting machine as suggested in [18]. We need to assume that at least either the voting authority or the voting machine is honest.

Table 3. The additional assumptions required in the improved Bingo Voting respect to the ones stated in [6,18], according to the outcome of our formal analysis

Property	Assumptions in [6,18]	**Additional assumptions**
Individual verification	Honest *TRNG* and *BB*	–
Dispute resolution at voting	Honest *TRNG* and *BB*	Do not blame the *VA* if barcodes are different
Dispute resolution at tallying	Honest *TRNG* and *BB*	–
Global verification	Honest *TRNG* and *BB*	Up to dishonest voters. Voters check and dispute incorrect receipts at voting and at tallying
Vote privacy if dispute at voting	Honest *TRNG* and *BB*. *VA* has no access to the voting machine	Honest *VA*
Vote privacy	Honest *TRNG* and *BB*. *VA* has no access to the voting machine	Honest *VA* or voting machine
Receipt freeness	Honest *TRNG*, *BB*, *VA*, and voting machine	–
Coercion resistance	Honest *TRNG*, *BB*, *VA*, and voting machine	–

ProVerif can prove that vote privacy holds after a dispute if the disputed receipt is not published on the bulletin board and the dummy vote corresponding to the dummy random numbers revealed by the privacy sleeve is not opened. In fact, if the receipt is published, vote privacy does not hold any more because the random number generated by the TRNG is revealed during the dispute. If the dummy vote is opened, vote privacy does not hold as well because this would reveal one of the candidates not chosen by the voter. However, we found that not revealing the receipt and not opening the dummy vote after a dispute might break vote privacy.

Privacy Attack Due to Dispute Resolution. Let us consider the scenario with two candidates in which a voter mistakenly disputes a valid receipt at voting. This vote should not be counted because the receipt is not published. Also, we require that a pair of dummy votes that are not in any receipts should not be opened

- The *disputed* dummy vote containing the disputed dummy random number associated with the candidate not chosen by the voter printed on the receipt.

– A dummy vote associated with the candidate chosen by the voter so that the disputed receipt is not counted at tallying.

Then, the voting authority should prove in zero-knowledge that the pair of dummy votes contain the list of the candidates. However, we observe that *any* pair of dummy votes containing the list of the candidates can serve for such proof since the corresponding receipt will not be published. Thus, a dishonest voting machine can signal a different dummy random number to the voting authority and print the disputed dummy random number again into another receipt, which will be published on the bulletin board. This would reveal how the disputing voter voted, breaking vote privacy. If one considers a dishonest voter, this attack is even more harmful. A dishonest voter can dispute a vote on purpose to learn how another voter voted since the dishonest voter knows the disputed dummy random number.

Note that the voting machine does not need to communicate with the dishonest voter to break vote privacy of another voter, and that this attack works even considering an honest voting authority. Of course, the attack is not possible if one considers an honest voting machine but there would not be need of dispute resolution at all in the first place if one makes such an assumption.

None of the papers presenting the improved Bingo Voting describes what happens after a dispute. Prearranging dummy votes may mitigate the attack at the cost of assuming an honest voting authority. Another possible mitigation to such an attack might be to allow voters who dispute their votes to revote. Revoting requires to generate additional dummy votes. The total amount of needed dummy votes should be the double of the original amount in order to avoid denial of voting attacks. However, this is a partial solution as it would not mitigate attacks due to dishonest voters.

4 Related Work

Several voting schemes have considered notions of dispute resolution or related properties. The FOO protocol [14] is one of the first voting schemes that enables voters to prove certain frauds due to a dishonest voting authority. Pret â Vòter [27] and vVote [13] provide some dispute resolution and accountability guarantees as a voter can use invalid proof and a ballot confirmation check as evidence. Remotegrity [31], Scantegrity II [9], and Scantegrity III [29] detail dispute resolution processes that allow voters to file disputes in case of incorrect designated ballots or *confirmation codes*, which are invisible random codes preprinted on the ballots. sElect [25] features a fully automated verification procedure that performs cryptographic checks without requiring any voter interaction. The procedure is capable to single out a specific misbehaving party and producing the necessary evidence of the misbehaviour. Schoenmarkers [28] and Kiayias and Yung [20] design dispute-free voting schemes, whose aim is to neutralise faults rather than providing mechanisms to address them. Some of the above protocols have been formally checked for accountability and/or privacy properties.

However, no formal analysis has been done to check whether disputes leak any information regarding how the voter voted.

Prior works on the formalisation of dispute resolution and related properties, such as accountability, include the seminal work by Küsters et al. [24], who advance accountability notions in the symbolic and computational models. Moreover, they provide an analysis of accountability and coercion resistance [26] of the original Bingo Voting scheme. Bruni et al. [8] propose formal definitions of accountability that are amenable to automated verification. One of their case studies is the improved Bingo Voting, which they analyse up to the voting phase, finding that it does not meet dispute resolution at voting. In contrast, we find that, if the dispute resolution test does not blame the voting authority when the barcodes are different between the paper ballot and the receipt, then the improved Bingo Voting achieves that property. Künneman et al. [23] give verification conditions that imply accountability based on *counterfactual relations*, capturing what actually happened to what could have happened. Basin et al. [2] proposed a definition of dispute resolution for voting requiring that voters get evidence that their ballot is incorrectly recorded before the end of the election.

The notions of individual verifiability and universal verifiability have been extensively studied in voting [3,4,10,11,19]. Kremer et al. [21] formalised both individual and universal verifiability in the applied pi-calculus, including the requirement of *eligibility verifiability*, which expresses that auditors can verify that each vote in the election result was cast by a registered voter, and there is at most one vote per voter. Smyth et al. [30] used ProVerif to check verifiability in three voting protocols expressing the requirements as reachability properties. Gallegos-Garcia et al. [15] studies how to achieve verifiability without any trust assumptions. Giustolisi et al. [17] observe that privacy-preserving verifiability can be achieved using non-interactive zero-knowledge proofs and functional encryption techniques. More recently, Cortier and Lallemand [12] have shown that a voting scheme that does not meet individual verifiability fails to achieve vote privacy, when one considers the same trust assumptions. This line of work opens up to interesting questions on how stronger properties such as dispute resolution and coercion resistance relate.

5 Conclusion

Dispute resolution mechanisms are essential components of a voting scheme, enabling the correctness of an election outcome. They can provably expose a misbehaving voting authority, hence deterring it by doing so. However, dispute resolution is useless if it is not triggered when it should be, and voters should not have to choose to either raise a dispute or keep their vote private. In this work, we have looked at the privacy-preserving dispute resolution mechanisms described in the improved Bingo Voting.

The formal analysis of the improved Bingo Voting allows us to identify precisely the necessary assumptions that enable the scheme to meet all the stated properties. It is found that global verification, which enables any observer to dispute the correctness on an election, cannot be achieved without dispute resolution both at voting and at tallying. To the best of our knowledge, it is an open question whether this is just for the improved Bingo Voting or it is a requirement for any voting scheme.

It is also found that assuming that the voting authority has not illegitimate access to the voting machine is not enough to guarantee vote privacy: either the voting authority or the voting machine must be honest at least. However, it is found that dispute resolution at voting can be achieved only assuming an honest voting authority as prearranging dummy votes would enable the voting authority to link votes to voters.

The results of this work also show that designing privacy-preserving dispute resolution mechanisms with minimal trust assumptions is not a trivial task in voting. The voting booth assumption should ideally be the sole assumption made in a voting scheme. Also, the details of the aftermath of a dispute resolution procedure in voting need to be described and thought with the same precision and care as are the *standard* voting procedures. For the improved Bingo Voting, we observe that, while cancelling an election due to a dispute is not an option, allowing voters who wrongly contest a receipt to revote mitigates an attack due to a dishonest voting machine. However, it does not help against a voting machine colluding with a dishonest voter.

Other voting schemes might achieve privacy-preserving dispute resolution with fewer assumptions than the improved Bingo Voting. With this work, we stress the importance of detailing the aftermath of disputes and aim at stimulating the voting community to make similar analyses to other voting schemes.

Acknowledgments. We are grateful to Rasmus Dilling Møller and Sean Wachs for helping out with the privacy analysis of an early model of Bingo Voting.

References

1. Abadi, M., Fournet, C.: Mobile values, new names, and secure communication. In: POPL, pp. 104–115. ACM, New York (2001)
2. Basin, D.A., Radomirovic, S., Schmid, L.: Dispute resolution in voting. CoRR abs/2005.03749 (2020). https://arxiv.org/abs/2005.03749
3. Benaloh, J.: verifiable secret-ballot elections. Ph.D. thesis, Yale University, December 1996
4. Benaloh, J., Tuinstra, D.: Receipt-free secret-ballot elections (extended abstract). In: STOC, pp. 544–553. ACM (1994)
5. Blanchet, B.: An efficient cryptographic protocol verifier based on prolog rules. In: CSFW, pp. 82–96. IEEE Computer Society (2001)
6. Bohli, J.M., Henrich, C., Kempka, C., Muller-Quade, J., Rohrich, S.: Enhancing electronic voting machines on the example of bingo voting. IEEE Trans. Inf. Forensics Secur. **4**, 745–750 (2009)
7. Bohli, J.-M., Müller-Quade, J., Röhrich, S.: Bingo voting: secure and coercion-free voting using a trusted random number generator. In: Alkassar, A., Volkamer, M. (eds.) Vote-ID 2007. LNCS, vol. 4896, pp. 111–124. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77493-8_10

8. Bruni, A., Giustolisi, R., Schuermann, C.: Automated analysis of accountability. In: Nguyen, P., Zhou, J. (eds.) ISC 2017. LNCS, vol. 10599, pp. 417–434. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69659-1_23
9. Carback, R., et al.: Scantegrity II municipal election at Takoma park: the first E2E binding governmental election with ballot privacy. In: USENIX Conference on Security. USENIX (2010)
10. Cohen, J., Fischer, M.: A robust and verifiable cryptographically secure election scheme (extended abstract). In: FOCS, pp. 372–382. IEEE (1985)
11. Cortier, V., Galindo, D., Küsters, R., Müller, J., Truderung, T.: SoK: verifiability notions for e-voting protocols. In: IEEE Symposium on Security and Privacy, pp. 779–798 (2016)
12. Cortier, V., Lallemand, J.: Voting: you can't have privacy without individual verifiability. In: CCS, pp. 53–66. ACM (2018)
13. Culnane, C., Ryan, P.Y.A., Schneider, S., Teague, V.: vVote: a verifiable voting system (DRAFT). CoRR abs/1404.6822 (2014)
14. Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: Seberry, J., Zheng, Y. (eds.) AUSCRYPT. LNCS, pp. 244–251. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-57220-1_66
15. Gallegos-García, G., Iovino, V., Rial, A., Rønne, P.B., Ryan, P.Y.A.: (universal) unconditional verifiability in e-voting without trusted parties. CoRR abs/1610.06343 (2016). http://arxiv.org/abs/1610.06343
16. Giustolisi, R., Bruni, A.: The ProVerif code used to verify the Improved Bingo Voting. https://itu.dk/people/rosg/code/evoteid20code.tar.gz
17. Giustolisi, R., Iovino, V., Lenzini, G.: Privacy-preserving verifiability - a case for an electronic exam protocol. In: SECRYPT, pp. 139–150. SciTePress (2017)
18. Henrich, C.: Improving and analysing bingo voting. Ph.D. thesis (2012). https://doi.org/10.5445/IR/1000030270
19. Hirt, M., Sako, K.: Efficient receipt-free voting based on homomorphic encryption. In: Preneel, B. (ed.) EUROCRYPT. LNCS, vol. 1807, pp. 539–556. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_38
20. Kiayias, A., Yung, M.: Self-tallying elections and perfect ballot secrecy. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 141–158. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45664-3_10
21. Kremer, S., Ryan, M., Smyth, B.: Election verifiability in electronic voting protocols. In: Gritzalis, D., Preneel, B., Theoharidou, M. (eds.) ESORICS 2010. LNCS, vol. 6345, pp. 389–404. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15497-3_24
22. Kremer, S., Ryan, M.: Analysis of an electronic voting protocol in the applied pi calculus. In: Gritzalis, D., Preneel, B., Theoharidou, M. (eds.) ESOP 2005. LNCS, vol. 3444, pp. 186–200. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31987-0_14
23. Künnemann, R., Esiyok, I., Backes, M.: Automated verification of accountability in security protocols. In: CSF, pp. 397–413. IEEE (2019)
24. Küsters, R., Truderung, T., Vogt, A.: Accountability: definition and relationship to verifiability. In: CCS, pp. 526–535. ACM (2010)
25. Küsters, R., Müller, J., Scapin, E., Truderung, T.: sElect: a lightweight verifiable remote voting system. In: CSF, pp. 341–354. IEEE (2016)
26. Küsters, R., Truderung, T., Vogt, A.: A game-based definition of coercion-resistance and its applications. In: CSF, pp. 122–136. IEEE (2010)
27. Ryan, P.Y.A., Bismark, D., Heather, J., Schneider, S., Xia, Z.: PrÊvoter: a voter-verifiable voting system. IEEE Trans. Inf. Forensics Secur. **4**, 662–673 (2009)

28. Schoenmakers, B.: A simple publicly verifiable secret sharing scheme and its application to electronic voting. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 148–164. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_10
29. Sherman, A.T., Fink, R.A., Carback, R., Chaum, D.: Scantegrity III: automatic trustworthy receipts, highlighting over/under votes, and full voter verifiability. In: Shacham, H., Teague, V. (eds.) EVT/WOTE. USENIX (2011)
30. Smyth, B., Ryan, M., Kremer, S., Mounira, K.: Towards automatic analysis of election verifiability properties. In: Armando, A., Lowe, G. (eds.) ARSPA-WITS 2010. LNCS, vol. 6186, pp. 146–163. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16074-5_11
31. Zagórski, F., Carback, R.T., Chaum, D., Clark, J., Essex, A., Vora, P.L.: Remotegrity: design and use of an end-to-end verifiable remote voting system. In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R. (eds.) ACNS 2013. LNCS, vol. 7954, pp. 441–457. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38980-1_28