

This is a repository copy of *DRE-ip : A Verifiable E-Voting Scheme without Tallying Authorities*.

White Rose Research Online URL for this paper:  
<http://eprints.whiterose.ac.uk/117997/>

Version: Accepted Version

---

**Proceedings Paper:**

Shahandashti, Siamak F. [orcid.org/0000-0002-5284-6847](https://orcid.org/0000-0002-5284-6847) and Hao, Feng (2016) DRE-ip : A Verifiable E-Voting Scheme without Tallying Authorities. In: ESORICS 2016::Computer Security – ESORICS 2016. European Symposium on Research in Computer Security, 26-30 Sep 2016 , GRC , pp. 223-240.

[https://doi.org/10.1007/978-3-319-45741-3\\_12](https://doi.org/10.1007/978-3-319-45741-3_12)

---

**Reuse**

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

**Takedown**

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing [eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk) including the URL of the record and the reason for the withdrawal request.

# DRE-ip: A Verifiable E-Voting Scheme without Tallying Authorities\*

Siamak F. Shahandashti and Feng Hao

School of Computing Science, Newcastle University, UK  
{siamak.shahandashti,feng.hao}@ncl.ac.uk

**Abstract.** Nearly all verifiable e-voting schemes require trustworthy authorities to perform the tallying operations. An exception is the DRE-i system which removes this requirement by pre-computing all encrypted ballots before the election using random factors that will later cancel out and allow the public to verify the tally after the election. While the removal of tallying authorities significantly simplifies election management, the pre-computation of ballots necessitates secure ballot storage, as leakage of precomputed ballots endangers voter privacy. In this paper, we address this problem and propose DRE-ip (DRE-i with enhanced privacy). Adopting a different design strategy, DRE-ip is able to encrypt ballots in real time in such a way that the election tally can be publicly verified without decrypting the cast ballots. As a result, DRE-ip achieves end-to-end verifiability without tallying authorities, similar to DRE-i, but with a significantly stronger guarantee on voter privacy. In the event that the voting machine is fully compromised, the assurance on tallying integrity remains intact and the information leakage is limited to the minimum: only the partial tally at the time of compromise is leaked.

## 1 Introduction

Direct-recording electronic (DRE) machines have been extensively used for voting at polling stations around the world. In a typical process, a registered voter obtains a token after being authenticated at the polling station. She then enters a private booth and presents the token to a DRE machine. The token is for one-time use and allows the voter to cast only one vote. Usually, the DRE machine has a touch screen to record the vote directly from the voter (hence the name DRE). The machine may tally the votes in real time, or store the votes and tally later. In either case, the machine works like a black box: if an attacker maliciously changes the votes (or the tally thereof), this is likely to go unnoticed.

Lack of assurance on tallying integrity is commonly regarded as a critical weakness of such DRE machines. To address this problem, several cryptographic protocols are proposed in the literature. The seminal work by Chaum in 2004 [16] involves using visual cryptography to allow voters to verify the integrity of an election. The assurance on the integrity includes guarantees that the votes are

---

\* This is the full version of a paper by the same title to appear in ESORICS 2016.

cast as intended, recorded as cast, and tallied as recorded. The fulfilment of all three constitutes the widely-accepted notion of end-to-end (E2E) verifiability.

Chaum’s solution inspired a class of voting systems providing E2E verifiability. Prominent examples include MarkPledge [28], Prêt à Voter [29], Scantegrity [14] (and its predecessor PunchScan [21]), Helios [1], and STAR-Vote [4]. These systems are based on different voting media including physical ballots, optical scanners, DREs and web browsers. They use different tallying techniques, based on mix-nets or homomorphic encryption. But all these schemes allow individual voters to verify if their votes have been cast as intended and recorded as cast, and any observer to verify if all votes have been tallied as recorded.

In this paper we limit our attention to DRE-based elections. We focus on DRE as it has already been widely deployed for national elections worldwide. Today, nearly all of the deployed DRE systems work like a black box and offer no guarantee on integrity; consequently, their use has been abandoned in several countries such as the Netherlands, Germany and Ireland. However, in many other countries, these (unverifiable) DRE machines continue to be extensively used. We believe there is an urgent need to address this real-world problem.

Apart from Chaum’s system called Voteegrity, other existing E2E verifiable schemes for DRE-based elections include MarkPledge [28], VoteBox [31], STAR-Vote [4], and vVote [18]. These systems may differ significantly in details, but they share some common features. They all offer integrity assurance by introducing a set of trustworthy tallying authorities (TAs). Instead of the DRE directly recording the vote, the machine encrypts the vote on the fly under the joint public key of the TAs. Each TA is responsible for safeguarding a share of the decryption key. When voting is closed, a quorum of TAs jointly perform the tallying process which involves decryption of the ballots (or tally thereof) in a publicly-verifiable manner.

The addition of external TAs however introduces difficulties in the implementation. In theory, the TAs should be selected from parties with conflicting interests. They should have the expertise to independently manage their own key shares and perform cryptographic operations, and if they delegate their key management tasks, the delegates need to be trusted as well. A comparatively high level of cryptographic and computing skills is expected from the TAs. Furthermore, the quorum should be set sufficiently large such that collusion among the TAs is infeasible, but at the same time, sufficiently small such that the process is error-tolerant, since non-availability of TA keys will render the election result non-computable. Reconciling the two is not an easy task. As reported by real-world experience of building E2E verifiable voting based on Helios, the implementation of the TAs proved to be “one particularly difficult issue” [2].

Hao et al. investigated if it was possible to achieve E2E verifiability for a DRE-based election without involving any TAs [24]. They proposed a TA-free E2E verifiable voting system, called DRE-i (DRE with integrity). In DRE-i, the machine directly records the voter’s choice as in the existing practice of current DRE-based elections. However, the machine is required to publish additional audit data on a public bulletin board, to enable every voter to verify the integrity

of the voting process. In DRE-i, the encryption of votes is based on a variant of the ElGamal encryption scheme: instead of using a fixed public key for encryption as in standard ElGamal, DRE-i uses a dynamically constructed public key for encrypting ballots. The system removes the need for TAs by pre-computing encrypted ballots in a structured manner such that after the election, multiplication of all the published ciphertexts cancels out the random factors that were introduced during the encryption process, and permits anyone to verify the tally.

DRE-i demonstrates that the role of the TAs is not indispensable in achieving E2E verifiability in a DRE-based election. However, its pre-computation strategy inevitably introduces the requirement of ensuring that the pre-computed data is securely stored and accessed during the voting phase. Furthermore, it means that it is possible for an adversary that breaks into the secure storage module to potentially compromise the privacy of all ballots. The authors of DRE-i [24] suggest to use tamper-resistant hardware to protect the pre-computed data in sensitive elections. However, the use of tamper-resistant hardware may significantly drive up the cost of each DRE machine. Furthermore, designing secure API for tamper-resistant hardware is a challenging problem on its own.

It remains an open problem as whether it is possible to achieve the best of both worlds, i.e. strong assurance on the integrity of a DRE-based election without involving any TAs, and simultaneously, a strong guarantee on the privacy of votes without depending on tamper-resistant hardware.

In this paper, we provide a positive answer to this question and present a new E2E verifiable voting system, which we call *DRE-ip* (DRE-i with enhanced privacy). Instead of pre-computing ciphertexts, DRE-ip adopts a more conventional approach, as in other existing DRE-based verifiable systems (see e.g. [31, 4]), to encrypt the vote on the fly during voting. DRE-ip achieves E2E verifiability without TAs, but at the same time provides a significantly stronger privacy guarantee than DRE-i.

*Our Contributions.* We present DRE-ip, an end-to-end verifiable DRE-based voting system that encrypts ballots in real-time, but requires no TAs to decrypt ballots in the tallying phase. We consider intrusive attacks in which the adversary is able to control an arbitrary number of voters and gets read access to the DRE machine for an arbitrary period during the voting phase. We prove that under such attacks, DRE-ip guarantees that elections with the same non-adversarial tally (i.e. tally of the votes neither controlled nor observed by the adversary) remain indistinguishable based on the decision Diffie-Hellman assumption. This shows that in the event of an intrusive attack, only the privacy of the ballots cast during the attack period is lost – a loss which is inevitable – and the ballots cast outside the attack period are guaranteed to remain private. DRE-ip constitutes the first verifiable DRE-based system that removes the need for tallying authorities without introducing new assumptions.

*Related Work.* In his seminal work on anonymous communications, Chaum put forward e-voting as an application of his technique [15]. This prompted considerable research on e-voting, among which is the work of Benaloh [10] that

proposed a formal definition of *ballot secrecy*. Later, Benaloh and Tuinstra argued for *receipt-freeness* [9], and Juels, Catalano, and Jakobsson put forward *coercion-resistance* [25] as progressively stronger notions of privacy. On the other hand, verifiability has evolved as a property guaranteeing the integrity of e-voting systems. Earlier works considered *individual verifiability*. The notion of *universal verifiability* emerged in later works and Sako and Kilian explicitly formalized it [30]. Finally, through the works of Chaum [16] and Neff [28], notions of verifiability were refined into that of *end-to-end verifiability*, which includes guarantees that the votes are cast as intended, recorded as cast, and tallied as recorded. End-to-end verifiability has now become a widely-accepted security requirement for e-voting schemes. Accordingly, in this paper, we limit our attention to end-to-end verifiable voting schemes.

There has been a renewed interest in academic research on e-voting in the past fifteen years and a number of end-to-end verifiable schemes have been designed and used in practice. Among the more influential schemes are Voteegrity, proposed by Chaum [16], and MarkPledge, proposed by Neff [28], which are the first end-to-end verifiable schemes. Many other schemes follow similar approaches, including Prêt à Voter [29], a tailored variant of which, vVote, has been used in state elections in Victoria, Australia [18], Scantegrity [14], which was trialed in local elections in Takoma Park, Maryland, USA [13], and STAR-Vote [4], which is scheduled for deployment in elections in Travis County, Texas, USA [26]. Other schemes that have been used in internal university or party elections include PunchScan [21], Bingo Voting [11], Helios [1], Wombat [7], and DRE-i [24].

## 2 Preliminaries

In this section, we review the preliminaries required for description of DRE-ip, including the notation and cryptographic setting we use.

*Notation.* Following the notation introduced by Camenisch and Stadler [12], we use  $\mathbb{R}_K\{\lambda : \Gamma = \gamma^\lambda\}$  to denote a non-interactive *proof of knowledge* of (a secret)  $\lambda$  such that (for publicly-known  $\Gamma$  and  $\gamma$ ):  $\Gamma = \gamma^\lambda$ . Where the context is clear, we shorten the notation to  $\mathbb{R}_K\{\lambda\}$ . We use  $\mathbb{R}_{WF}\{A : X, Y, Z\}$  to denote a *proof of well-formedness* of  $A$  with respect to  $X$ ,  $Y$ , and  $Z$ . Where the context is clear, we shorten the notation to  $\mathbb{R}_{WF}\{A\}$ .

### 2.1 Cryptographic Setting

We assume a DSA-like multiplicative cyclic group setting, where  $p$  and  $q$  are large primes that satisfy  $q \mid p - 1$ . We work in the subgroup  $\mathbb{G}_q$  of order  $q$  of the group  $\mathbb{Z}_p^*$  and assume that  $g$  is a generator of  $\mathbb{G}_q$ . Alternatively, our proposed system can be implemented over an elliptic curve in an ECDSA-like group setting.

The decision Diffie-Hellman (DDH) assumption [19] is defined as follows:

**Assumption 1. (DDH)** For randomly chosen  $a, b \in \mathbb{Z}_q^*$  and  $R \in \mathbb{G}_q$ , given  $(g, g^a, g^b, \Omega)$  where  $\Omega \in \{g^{ab}, R\}$ , it is hard to decide whether  $\Omega = g^{ab}$  or  $\Omega = R$ .

Zero knowledge proofs, first proposed by Goldwasser, Micali, and Rackoff [22], prove the truth of a statement without conveying any other information, i.e. they guarantee that whatever the verifier can feasibly compute after seeing a proof, they could have computed on their own. Subsequent work by Bellare and Goldreich [5] refined the definition of zero knowledge proofs to distinguish them from proofs of knowledge. Intuitively speaking, proofs of knowledge are guaranteed to be generated by a prover with explicit knowledge of a quantity. In our protocol, the Fiat-Shamir heuristic is employed to construct non-interactive proofs [20]. Consequently, our security proofs are in the Random Oracle Model [6].

### 3 Our Proposed Solution: DRE-ip

DRE-ip requires a secure and publicly-accessible bulletin board (BB) and incorporates voter-initiated auditing to achieve end-to-end verifiability. We assume the DRE has append-only write access to the BB over an authenticated channel. We assume voting is conducted in supervised polling stations and there are procedures in place to ensure the “one person, one vote” principle, including secure voter registration and authentication. At the time of voting, a voter is authenticated first and issued a token, unlinked to her identity. She then enters a private voting booth and authenticates herself to the DRE using the token. Up to here, the assumptions and mechanisms are similar to those of DRE-i.

We describe DRE-ip for the case where there are only two candidates, i.e. for  $v_i$  representing the vote of the  $i$ -th ballot, we have  $v_i \in \{0, 1\}$ . In DRE-ip the setup establishes two generators  $g_1$  and  $g_2$ , whose logarithmic relationship is unknown. The DRE keeps track of the running tally  $t = \sum v_i$  for the cast votes  $v_i$ , and the sum  $s = \sum r_i$  for random  $r_i$  generated on the fly.

To achieve individual verifiability, DRE-ip incorporates Benaloh-style voter-initiated auditing [8], i.e. the voter gets the option to audit the ballot composed by the DRE to gain confidence in that the DRE is preparing the ballots according to her choice. If a ballot is audited, it cannot be used to cast a vote. Therefore, the set of all ballots  $\mathbb{B}$  at the closing of the voting phase will be comprised of the audited ballots  $\mathbb{A}$  and the cast ballots  $\mathbb{C}$ , i.e.  $\mathbb{B} = \mathbb{A} \cup \mathbb{C}$ .

*Voting Phase.* This phase involves the voter, the DRE, and the BB:

1. The voter enters the booth, initiates voting, and keys in her vote  $v_i \in \{0, 1\}$ .
2. The DRE generates random  $r_i \in \mathbb{Z}_q^*$ , calculates

$$R_i = g_2^{r_i}, \quad Z_i = g_1^{r_i} g_1^{v_i}, \quad \text{R}_{\text{WF}}\{Z_i : g_1, g_2, R_i\},$$

and provides a signed receipt including the unique ballot index  $i$  and the ballot content  $R_i, Z_i$ , and  $\text{R}_{\text{WF}}\{Z_i\}$  to the voter.

3. The voter observes that the first part of the receipt is provided, and chooses to either audit the ballot or confirm her vote.

In case of audit:

4. The DRE adds  $i$  to  $\mathbb{A}$ , provides a signed receipt of audit, clearly marked **audited**, including  $r_i$  and  $v_i$  to the voter.
5. The voter takes and keeps the receipt, and verifies that  $v_i$  reflects her choice. If the verification succeeds, voting continues to Step 1; otherwise, the voter should raise a dispute immediately.

In case of confirmation:

4. The DRE adds  $i$  to  $\mathbb{C}$ , updates the tally and the sum:

$$t = \sum_{j \in \mathbb{C}} v_j \quad \text{and} \quad s = \sum_{j \in \mathbb{C}} r_j ,$$

and provides a signed receipt of confirmation, clearly marked **confirmed**, to the voter, and securely deletes  $r_i$  and  $v_i$ .

5. The voter leaves the booth with her receipts.
6. The DRE posts on the BB all the receipts provided to the voter.
7. The voter verifies that her receipts match those on the BB.

*Tallying Phase.* This phase involves the DRE, the BB, and the public:

1. The DRE posts on the BB the final tally  $t$  and the final sum  $s$ .
2. The public:
  - verify all the well-formedness proofs on the BB (*well-formedness verification*);
  - verify that for all the audited ballots on the BB:  $R_i$  and  $Z_i$  included in the first part of the receipt are consistent with  $r_i$  and  $v_i$  included in the second part (and with the system parameters  $g_1$  and  $g_2$ ) (*audit consistency verification*); and
  - verify that the following equations hold (*tally verification*):

$$\prod_{j \in \mathbb{C}} R_j \stackrel{?}{=} g_2^s \quad \text{and} \quad \prod_{j \in \mathbb{C}} Z_j \stackrel{?}{=} g_1^s g_1^t . \quad (1)$$

If at any point during the voting or tallying phases, any of the verifications carried out by the voter or the public does not succeed, the election staff should be notified and we assume that there are procedures in place dealing with such verification failures. These include voter verifications in Steps 5 (in case of audit) and 7 of the voting phase and public verifications in Step 2 of the tallying phase.

Figure 1 shows the DRE-ip bulletin board. An audited receipt (with index  $i$ ) and a confirmed receipt (with index  $j$ ) are shown. Each receipt has two parts: the first part is provided to the voter before she decides to either audit or confirm her ballot and includes similar information for all receipts; the second part is provided after the voter makes her decision and includes different information based on her choice. Both parts of the receipt are signed by the DRE.

The proof of well-formedness  $R_{\text{WF}}\{Z_i : g_1, g_2, R_i\}$  can be implemented as a non-interactive proof of knowledge

$$R_{\text{WF}}\{Z_i\} = P_K\{ r_i : (R_i = g_2^{r_i} \wedge Z_i = g_1^{r_i}) \vee (R_i = g_2^{r_i} \wedge Z_i/g_1 = g_1^{r_i}) \}.$$

Initial: $g_1, g_2$	
Receipts:	⋮
$i : R_i, Z_i, P_{WF}\{Z_i\}$	audited, $r_i, v_i$
	⋮
$j : R_j, Z_j, P_{WF}\{Z_j\}$	confirmed
	⋮
Final: $t, s$	

**Fig. 1.** DRE-ip bulletin board

This proof guarantees that  $Z_i \in \{g_1^{r_i}, g_1^{v_i} g_1\}$ , or equivalently  $v_i \in \{0, 1\}$ .

Such a proof can be realized based on Schnorr proofs of knowledge of discrete logarithm [32]. Starting with a Schnorr proof, one can apply techniques proposed by Cramer, Damgård, and Schoenmakers [17] to construct proofs of disjunctive knowledge, conjunctive knowledge, and combinations of both. The Fiat-Shamir heuristic [20] is then applied to make the constructed proofs non-interactive. The index  $i$  of the ballot is embedded in the proof (as an input to the hash function) to bind the proof to the ballot.

In practice, truncated hash functions may be used to calculate a short digest, e.g. 4 alphanumeric characters long, of each part of the receipt, so that the voter can easily compare the digests on their receipts with those on the bulletin board. In this case, voters are expected to verify the receipts before leaving the polling station and we assume facilities are provided for them to do so in the station.

## 4 Security of DRE-ip

In this section we provide proofs to show that DRE-ip is end-to-end verifiable and ensures ballot secrecy under both non-intrusive and intrusive attacks.

### 4.1 End-to-End Verifiability

We discuss the integrity (i.e. correctness) of the election tally in DRE-ip and show how DRE-ip achieves end-to-end verifiability: we prove that, assuming all proofs of well-formedness are proofs of knowledge, votes are tallied as recorded if public verification succeeds; furthermore, we demonstrate how voter-initiated auditing guarantees that votes are recorded as cast, and cast as intended.

We assume the bulletin board is secure, in particular it is append-only and publicly accessible. Besides, there should be a mechanism to establish an authenticated channel between authorized DRE(s) and the bulletin board, to ensure that only an authorized DRE can append new values to the BB, and also that such values are not modified in transit. This can be achieved using standard techniques such as digital signatures. Furthermore, we assume that the number of voters is less than the size of the group  $q$ .



Recall that public verification in DRE-ip, i.e. Step 2 of the tallying phase, includes three types of verification: well-formedness verification, audit consistency verification, and tally verification. The following theorem shows that if well-formedness and tally verifications succeed, DRE-ip achieves the tallied-as-recorded property, that is, DRE-ip guarantees that the tally on the bulletin board is the correct tally of all the confirmed ballots on the bulletin board.

**Theorem 1.** *In DRE-ip, assuming that all proofs of well-formedness are proofs of knowledge, if the public well-formedness and tally verifications succeed, then the reported tally  $t$  is the correct tally of all the confirmed votes on the BB.*

The proof is rather straightforward and hence omitted here. In short, one can demonstrate how the proofs of well-formedness and the first tally verification check (i.e. the first of the two in Equation 1) collectively guarantee that the second tally verification equation (i.e. the second of the two in Equation 1) holds if and only if  $t = \sum_{i \in \mathbb{C}} v_i$ , where  $\mathbb{C}$  denotes the set of confirmed votes. Hence, if well-formedness and tally verifications are carried out successfully, the reported tally  $t$  is guaranteed to be the correct tally of all the confirmed votes on the BB.

Voter initiated auditing includes the following checks: first, by observing the first part of the receipt is provided before deciding to either audit or confirm a ballot, the voter makes sure that the DRE commits to the first part of the ballot; second, by checking that the receipts match what is published on the BB, the voter makes sure that her interaction with the machine is captured faithfully on the bulletin board. The public verification of the consistency of the audited ballots, i.e. the audit consistency verification, guarantees that DRE has been successful in responding to the challenges made by voter initiated auditing. Hence, the individual verification and the public audit consistency verification collectively ensure that the votes are cast as intended and recorded as cast. Theorem 1 ensures that votes are tallied as recorded.

## 4.2 Ballot Secrecy

Ballot secrecy corresponds to the natural expectation from a voting system to protect the secrecy of cast ballots. We consider a definition of ballot secrecy which requires that an adversary controlling the voting behaviour of a group of dishonest voters should not be able to distinguish between any two elections, regardless of how honest voters vote, as long as the two elections have the same partial tally of honest votes. This definition originates from Benaloh [10, p. 74].

We assume a secure setup phase; that is, we assume that the discrete logarithm of  $g_2$  in base  $g_1$  is either not known to any party or securely deleted after the two generators are computed. We also assume secure deletion of values  $x_i$ ,  $y_i$ , and  $v_i$  after each vote is cast<sup>1</sup>.

<sup>1</sup> See, for instance, [23] and the references within for an overview of available solutions to secure data deletion.

We consider an intrusive adversary that apart from the ability to determine an arbitrary number of votes, gets read access to the DRE storage for a period during the voting phase. The adversary is able to read the publicly available information on the bulletin board, which includes the total tally. Besides, we assume that the adversary can control an arbitrary number of voters, hence in effect cast an arbitrary number of votes. The adversary is able to observe the votes cast during the access period and also read the running (partial) tally  $t$  and (partial) sum  $s$ .

Let us call the votes cast or observed by the adversary the *adversarial* votes. Knowledge of the adversarial votes along with the total and partial tallies enables the adversary to find out the tally of the non-adversarial votes cast before and after the adversarial access period. We prove that under the DDH assumption, this is the only information the adversary gains about the non-adversarial votes. In particular, we show that any two elections in which the non-adversarial votes cast before and after the adversarial access period have the same partial tallies are indistinguishable to the adversary. Note that in DRE-i, in case of an adversarial access to the voting machine storage, the privacy of the ballots cast outside the adversarial access period is also lost. Therefore, while DRE-i falls victim to such intrusive attacks, DRE-ip guarantees vote privacy under such attacks.

We first consider two elections in which all votes are the same except for two votes that are swapped. We show that the bulletin boards of these two elections remain indistinguishable to the adversary as long as these two votes are non-adversarial votes both cast either before or after the adversarial access period. More formally, we have:

**Lemma 1.** *In DRE-ip, assuming that all proofs of well-formedness are zero knowledge, if the DDH assumption holds, then an adversary that determines an arbitrary number of votes and gets temporary read access to the DRE storage cannot distinguish between two bulletin boards in which two votes both cast either before or after the adversarial access period are swapped.*

The proof of the lemma comes in Appendix A. The proof considers an adversary that not only can determine an arbitrary number of votes except two votes  $v_i$  and  $v_j$ , but gets access to DRE storage for an arbitrary period. Assuming that such an adversary is able to distinguish the bulletin boards in which  $v_i$  and  $v_j$  are swapped, we show how it can be used to break the DDH assumption. Basically, the proof shows that the sum  $s$  does not leak any extra information other than what the tally  $t$  does.

Given Lemma 1, we expand it to prove that any two elections with the same non-adversarial partial tallies of the votes cast before and after the adversarial access period remain indistinguishable to an adversary who controls an arbitrary number of votes. This shows that the only knowledge the adversary can gain about the non-adversarial votes cast before and after the adversarial access period is that disclosed by the partial and total tallies.

**Theorem 2.** *In DRE-ip, assuming that all proofs of well-formedness are zero knowledge, if the DDH assumption holds, then an adversary that determines an arbitrary number of votes and gets temporary read access to the DRE storage cannot gain any knowledge about the non-adversarial votes cast before and after the adversarial access period other than their partial tallies.*

*Proof.* To prove this theorem, we show that under the DDH assumption, given any two sets of non-adversarial votes cast before and after the adversarial access period with the same partial tallies, one can simulate two corresponding bulletin boards that are indistinguishable to an adversary that chooses an arbitrary number of adversarial votes.

First, note that any two given sets of non-adversarial votes with the same partial tally differ on an even number of votes, say  $2d$ . This means that with  $d$  “swaps” one set of these votes can be converted to the other, where in each swap, for some  $i$  and  $j$ , the  $i$ -th vote is replaced with the  $j$ -th one, and vice versa. In Lemma 1 we proved that the bulletin boards before and after each swap remain indistinguishable to the adversary under DDH. Consequently, the bulletin boards corresponding to the two given sets of non-adversarial votes remain indistinguishable to the adversary and the proof is complete.  $\square$

We discussed the case for a single adversarial access period, but the above theorem guaranteeing ballot secrecy can be easily extended to cover attacks involving multiple adversarial access periods.

## 5 Comparison

In this section we look at how DRE-ip compares with other DRE-based verifiable e-voting systems. In particular, we consider Chaum’s Voteegrity [16], Neff’s MarkPledge [28], VoteBox [31], STAR-Vote [4], DRE-i [24], and vVote [18].

Voteegrity is based on visual cryptography and uses onion encryption. MarkPledge employs a purpose-designed encryption scheme that allows challenge-response-style individual verifiability. VoteBox and STAR-Vote are both based on exponential ElGamal encryption which allows homomorphic tallying. In vVote, ballots are encrypted using elliptic curve ElGamal and later decrypted individually after mixing. DRE-i on the other hand uses encryption that does not admit to a fixed decryption key. DRE-ip basically uses the exponential ElGamal encryption in which no party knows the decryption key. All these systems consider voter registration and voter authentication outside their scope and assume they are carried out correctly and securely.

In general, systems that require tallying authorities, i.e. Voteegrity, MarkPledge, VoteBox, STAR-Vote, and vVote, assume a minimum number of them are available at the tallying phase to compute the election tally. DRE-i and DRE-ip do not require such an assumption to guarantee availability.

To guarantee integrity, all systems we consider rely on a secure bulletin board and on a sufficient number of voters carrying out individual verification. Systems

**Table 1.** Selected security assumptions for DRE-based verifiable e-voting systems. TA: tallying authority, VIA: voter-initiated auditing, BB: bulletin board, RNG: random number generation, ■: assumption is required, □: assumption is not required.

System	Availability	Integrity		Privacy				
	Reliable TA(s)	Sufficient VIA	Secure BB	Secure setup	Secure RNG	Secure deletion	Secure ballot storage	Trust-worthy TA(s)
Votegrity	■	■	■	■	■	■	□	■
MarkPledge	■	■	■	■	■	■	□	■
VoteBox	■	■	■	■	■	■	□	■
STAR-Vote	■	■	■	■	■	■	□	■
DRE-i	□	■	■	■	■	■	■	□
vVote	■	■	■	■	■	■	□	■
DRE-ip	□	■	■	■	■	■	□	□

that require tallying authorities, i.e. *Votegrity*, *MarkPledge*, *VoteBox*, *STAR-Vote*, and *vVote*, also require that the tallying authorities perform the decryption of the tally correctly. In a verifiable system, this is enforced by requiring the tallying authorities to produce universally verifiable proofs of correct decryption. Hence, we consider assumptions underlying all the systems to guarantee integrity to be comparable, whether the system requires tallying authorities or not.

To guarantee privacy, all systems we consider assume a secure setup phase to generate and distribute system parameters and keys, as well as secure random number generators to produce the randomness required for probabilistic encryption. Furthermore, all systems assume that the captured votes and any ephemeral secrets generated for the cryptographic operations during the voting phase are securely erased. *Votegrity* is based on decryption mix-nets and requires that the tallying authorities do not collude to compromise voter privacy. *MarkPledge* and *vVote* employ re-encryption mix-nets to shuffle encrypted ballots before decryption, and assume that the tallying authorities do not decrypt ballots before mixing although they are available on the bulletin board. *VoteBox* and *STAR-Vote* require that the tallying authorities do not collude to decrypt individual ballots. *DRE-i* does not require this assumption, but instead relies on a secure ballot storage mechanism to keep the pre-computed ballots safe after the setup phase. *DRE-ip* does not require trust assumptions on tallying authorities or ballot storage.

Table 1 summarizes the main similarities and differences in terms of their underlying security assumptions between the voting systems we consider.

Let us now compare the computation complexity of *DRE-ip* with that of the other DRE-based verifiable e-voting systems. We do not consider *Votegrity*, *MarkPledge*, and *vVote* since they use mix-nets and their computation complexity depend on how these verifiable mix-nets are implemented. All calculations are based on a two-candidate election, encryption implemented based on exponential ElGamal, and one TA if present. Note that having multiple TAs increases the complexity of tally calculation and verification for all the schemes requiring

**Table 2.** Computation complexity of selected DRE-based verifiable e-voting systems.  $\mathbb{B}$ ,  $\mathbb{A}$ ,  $\mathbb{C}$ : all, audited, confirmed ballots,  $\epsilon$ : exponentiation,  $m$ : multiplication.

System	Ballot calculation	Well-formedness and consistency verification	Tally calculation	Tally verification
VoteBox	$6.4 \mathbb{B}  \epsilon$	$(6.8 \mathbb{A}  + 4.8 \mathbb{C} ) \epsilon$	$ \mathbb{C}  m + 3 \epsilon$	$ \mathbb{C}  m + 2.4 \epsilon$
STAR-Vote	$6.4 \mathbb{B}  \epsilon$	$(6.8 \mathbb{A}  + 4.8 \mathbb{C} ) \epsilon$	$ \mathbb{C}  m + 3 \epsilon$	$ \mathbb{C}  m + 2.4 \epsilon$
DRE-i	$10.8 \mathbb{B}  \epsilon$	$(9.6 \mathbb{A}  + 4.8 \mathbb{C} ) \epsilon$		$ \mathbb{B}  m + 1 \epsilon$
DRE-ip	$6.4 \mathbb{B}  \epsilon$	$(6.8 \mathbb{A}  + 4.8 \mathbb{C} ) \epsilon$		$2 \mathbb{C}  m + 2 \epsilon$

tallying authorities. We assume in all systems that the TA, if present, provides proofs of correct decryption as required by end-to-end verifiability. We also assume that the simultaneous multiple exponentiation (SME) technique [27] is used to optimize computations. Using SME, a term of the form  $g^x h^y$  costs equivalent to around 1.2 exponentiations to calculate.

The systems considered here use two types of well-formedness proof in general. The first type consists of proofs of (knowledge and) equality of two discrete logarithms and are of the general form

$$\mathbb{R}_K\{\lambda : \Gamma_1 = \gamma_1^\lambda \wedge \Gamma_2 = \gamma_2^\lambda\}. \quad (2)$$

Consider an exponential ElGamal encryption scheme with key pair  $(k, K = g^k)$  in which a message  $m$  is encrypted to the ciphertext  $(R = g^r, C = K^r g^m)$ . The proof

$$\mathbb{R}_{WF}\{m : g, K, (R, C)\} = \mathbb{R}_K\{k : K = g^k \wedge C/g^m = R^k\}$$

which is of the form of Equation 2 can be used as a proof of correct decryption, e.g. in systems like VoteBox and STAR-Vote. Such a proof, when realized as a Fiat-Shamir non-interactive Schnorr proof and optimized using the SME technique, requires 2 exponentiations to generate, and (equivalent to) around 2.4 exponentiations to verify. Algorithms for generation and verification of such proofs are transcribed in Appendix B.

The second type consists of disjunctive proofs of equality (and knowledge) of either one pair of discrete logarithms or the other, and are of the general form

$$\mathbb{R}_K\{\lambda : (\Gamma_1 = \gamma_1^\lambda \wedge \Gamma_2 = \gamma_2^\lambda) \vee (\Gamma_3 = \gamma_3^\lambda \wedge \Gamma_4 = \gamma_4^\lambda)\} \quad (3)$$

Such proof can be constructed as a disjunction of two conjunctive proofs of the form of Equation 2. These proofs can be used to prove well-formedness of the ballots in all the systems we consider. In DRE-ip, the ballot well-formedness proof  $\mathbb{R}_{WF}\{Z_i : g_1, g_2, R_i\}$  is of this form. This proof, when realized as a Fiat-Shamir non-interactive Schnorr proof and optimized using the SME technique, requires (equivalent to) around 4.4 exponentiations to generate, and (equivalent to) around 4.8 exponentiations to verify. Algorithms for generation and verification of such proofs are transcribed in Appendix B.

VoteBox and STAR-Vote both encrypt the vote under exponential ElGamal, which involves similar computation as that of DRE-ip. In DRE-ip, calculating

$R_i$  and  $Z_i$  take 1 exponentiation each, and calculating  $R_{WF}\{Z_i\}$  takes around 4.4 exponentiations. Hence, ballot calculation takes around 6.4 exponentiations per ballot in VoteBox, STAR-Vote, and DRE-ip. In DRE-i, two proofs of well-formedness are (pre-)calculated for each ballot and hence ballot calculation requires 10.8 exponentiations per ballot.

In all four systems, checking well-formedness of a confirmed ballot consists of verifying a proof of the second type discussed above, so it takes around 4.8 exponentiations per confirmed ballot. Consistency verification of an audited ballot consists of checking well-formedness of the ballot plus verifying whether the revealed audit information is consistent with the ballot. In VoteBox, STAR-Vote, and DRE-ip, the computation involved is similar. In DRE-ip for example,  $R_i$  and  $Z_i$  are recalculated based on the revealed values of  $r_i$  and  $v_i$  and the result is compared against reported values of  $R_i$  and  $Z_i$  on the BB. This takes 2 exponentiations, and hence consistency verification takes around 6.8 exponentiations per audited ballot. In DRE-i, there is an extra proof of the second type discussed above to verify for each audited ballot and hence consistency verification takes around 9.6 exponentiations per audited ballot.

In VoteBox and STAR-Vote, tally calculation requires all confirmed vote encryptions to be multiplied, the result decrypted, and finally a proof of correct decryption generated. Decryption and generating the proof of correct decryption require 1 and 2 exponentiations, respectively. These calculations are obviously carried out by the TAs. In DRE-i and DRE-ip, tallies are kept track of and reported by the DRE, so no extra calculation is needed.

Tally verification in VoteBox and STAR-Vote consists of multiplying confirmed vote encryptions and verifying the proof of correct decryption. The latter costs around 2.4 exponentiations as discussed above. In DRE-i, a tally verification equation is checked which requires multiplication of all vote encryptions and 1 exponentiation. In DRE-ip, two tally verification equations are checked which require multiplication of all  $R_i$  and also all  $Z_i$  for confirmed ballots and an exponentiation per equation.

Table 2 summarizes the computation complexity of different operations in the systems we discussed above. Note that our calculations above and figures listed in the table do not include the cost of validating the inputs to the verification algorithms to ensure that they belong to the right cryptographic groups. In elliptic curve based implementations of the systems discussed above, such validations incur negligible cost.

## 6 Extension to Multiple Candidates

Although we have described DRE-ip for two candidates only, there are two rather standard ways to extend it to support multiple candidates (see e.g. [24, 3]). Here we discuss voting for 1 out of  $n$  candidates for  $n \geq 3$ .

A straightforward method is to essentially run a separate parallel DRE-ip system for each candidate. Let  $v_{ij}$  represent the vote in ballot  $i$  and candidate  $j$ . 1 out of  $n$  votes include a  $v_{ij} = 1$  vote for one candidate and  $v_{ij} = 0$  votes

**Table 3.** Computation complexity of DRE-ip supporting voting for 1 out of  $n \geq 3$  candidates.  $\mathbb{B}$ ,  $\mathbb{A}$ ,  $\mathbb{C}$ : all, audited, confirmed ballots,  $\epsilon$ : exponentiation,  $\mathfrak{m}$ : multiplication.

DRE-ip extension	Ballot calculation	Well-formedness and consistency verification	Tally verification
Parallel	$(6.4n + 2)  \mathbb{B}  \epsilon$	$((6.8n + 2.4)  \mathbb{A}  + (4.8n + 2.4)  \mathbb{C} ) \epsilon$	$2n  \mathbb{C}  \mathfrak{m} + 2n \epsilon$
Encoded	$(2.4n + 1.6)  \mathbb{B}  \epsilon$	$((2.4n + 2)  \mathbb{A}  + 2.4n  \mathbb{C} ) \epsilon$	$2  \mathbb{C}  \mathfrak{m} + 2 \epsilon$

for all other candidates. Hence, an extra proof of well-formedness is required to guarantee that only one of the votes  $v_{ij}$  over all values of  $j$  is 1. The  $i$ -th ballot in this case will be in the form of a  $(3n + 1)$ -tuple:  $((R_{ij}, Z_{ij}, \mathbb{P}_{\text{WF}}\{Z_{ij}\})_{j=1}^n, \pi)$ , where  $\pi$  represents the extra proof. Since for each  $j$  the well-formedness proof  $\mathbb{P}_{\text{WF}}\{Z_{ij}\}$  already guarantees that  $v_{ij} \in \{0, 1\}$ , it would be sufficient for the extra proof to only show that  $\sum_{j=1}^n v_{ij} = 1$ . Interestingly, given the values  $R_{ij} = g_2^{r_{ij}}$ , this proof can be easily constructed as the proof of knowledge

$$\mathbb{P}_{\text{K}}\{ \sigma_i : (\prod_{j=1}^n Z_{ij})/g_1 = g_1^{\sigma_i} \wedge \prod_{j=1}^n R_{ij} = g_2^{\sigma_i} \}, \quad \text{where } \sigma_i = \sum_{j=1}^n r_{ij}.$$

This is a proof of the first type discussed above (i.e. of the form of Equation 2). Ballot generation for such a parallel DRE-ip systems costs  $n$  times that of a two-candidate DRE-ip plus 2 extra exponentiations to generate the extra proof, i.e.  $6.4n + 2$  exponentiations per ballot in total. Verifying the extra proof takes 2.4 exponentiations, thus well-formedness and consistency verification cost  $4.8n + 2.4$  exponentiations per confirmed ballot and  $6.8n + 2.4$  exponentiations per audited ballot. Tally verification costs  $n$  times that of a two-candidate DRE-ip.

Another method is to extend DRE-ip and encode a vote for candidate  $j$  as  $v_i = M^{j-1}$ , where  $M$  is an upper bound on the number of voters. The  $i$ -th ballot in this case will be in the form of a triple  $(R_i, Z_i, \mathbb{P}_{\text{WF}}\{Z_i\})$ , where  $R_i = g_2^{r_i}$  and  $Z_i = g_1^{r_i} g_1^{M^{j-1}}$ . The ballot well-formedness proof  $\mathbb{P}_{\text{WF}}\{Z_i\}$  will be a 1-out-of- $n$  disjunctive proof, rather than 1-out-of-2, and it can be realized as follows:

$$\mathbb{P}_{\text{K}}\{ r_i : \bigvee_{j=1}^n (R_i = g_2^{r_i} \wedge Z_i/g_1^{M^{j-1}} = g_1^{r_i}) \}.$$

This is an extended version of a proof of the second type discussed above (i.e. of the form of Equation 3). Generation of such a proof costs  $2 + 2.4(n - 1) = 2.4n - 0.4$  exponentiations and verifying it  $2.4n$  exponentiations. Ballot calculation in such an “encoded” DRE-ip system costs  $2.4n + 1.6$  exponentiations per ballot. Well-formedness and consistency verification for the system cost  $2.4n$  exponentiations per confirmed ballot and  $2.4n + 2$  exponentiations per audited ballot. Tally verification cost is similar to that of a two-candidate DRE-ip.

Table 3 summarizes the computation complexity for the two extensions. Overall, while parallel DRE-ip is more modular and hence more straightforward to implement, encoded DRE-ip is more efficient. A similar observation seems to hold for extended versions of VoteBox, STAR-Vote, and DRE-i.

## 7 Concluding Remarks

In this paper we revisited the design of the DRE-i voting system and proposed a new system: DRE-ip. On the theoretical level, we have shown that it is possible to have verifiable DRE-based voting systems in which the privacy of the ballots does not rely on trustworthy tallying authorities or trusted hardware. On the practical level, we have shown that DRE-ip provides an efficient and practical verifiable DRE-based voting solution able to preserve the privacy of the ballots even if the adversary gets temporary read access to the voting machine during the voting phase. Designing a system without tallying authorities that can efficiently support more complex electoral systems such as single transferable vote (STV) or write-in candidates remains an open problem.

**Acknowledgement.** The authors wish to thank Changyu Dong and the anonymous reviewers of ESORICS 2016 for their valuable comments. The authors are supported by the ERC Starting Grant No. 306994.

## References

- [1] B. Adida. Helios: Web-based open-audit voting. In *USENIX Security Symp.*, volume 17, pages 335–348, 2008.
- [2] B. Adida, O. de Marneffe, O. Pereira, and J.-J. Quisquater. Electing a university president using open-audit voting: Analysis of real-world use of Helios. In *EVT/WOTE'09*, page 10. USENIX, 2009.
- [3] O. Baudron, P.-A. Fouque, D. Pointcheval, J. Stern, and G. Poupard. Practical multi-candidate election system. In *ACM Symp. on Principles of Distributed Computing*, PODC '01, pages 274–283. ACM, 2001.
- [4] S. Bell, J. Benaloh, M. D. Byrne, D. DeBeauvoir, B. Eakin, G. Fisher, P. Kortum, N. McBurnett, J. Montoya, M. Parker, O. Pereira, P. B. Stark, D. S. Wallach, and M. Winn. STAR-Vote: A secure, transparent, auditable, and reliable voting system. *USENIX Journal of Election Technology & Systems*, 1(1):18–37, 2013.
- [5] M. Bellare and O. Goldreich. On defining proofs of knowledge. In E. F. Brickell, editor, *Crypto'92*, volume 740 of *LNCS*, pages 390–420. Springer, 1993.
- [6] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS'93*, pages 62–73. ACM, 1993.
- [7] J. Ben-Nun, M. Llewellyn, B. Riva, A. Rosen, A. Ta-Shma, and D. Wikström. A new implementation of a dual (paper and cryptographic) voting system. In *EVOTE2012: 5th Int'l Conf. on Electronic Voting*, pages 315–329, 2012.
- [8] J. Benaloh. Ballot casting assurance via voter-initiated poll station auditing. In *USENIX Workshop on Accurate E-Voting Technology (EVT)*, page 14, 2007.
- [9] J. Benaloh and D. Tuinstra. Receipt-free secret-ballot elections. In *ACM Symp. on Theory of Computing*, STOC '94, pages 544–553. ACM, 1994.
- [10] J. D. C. Benaloh. *Verifiable Secret-Ballot Elections*. PhD thesis, Department of Computer Science, Yale University, 1987.
- [11] J.-M. Bohli, J. Müller-Quade, and S. Röhrich. Bingo voting: Secure and coercion-free voting using a trusted random number generator. In *E-Voting and Identity*, pages 111–124. Springer, 2007.



- [12] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In *Crypto'97*, volume 1294 of *LNCS*, pages 410–424. Springer, 1997.
- [13] R. Carback, D. Chaum, J. Clark, J. Conway, A. Essex, P. Herrnsen, T. Mayberry, S. Popoveniuc, R. Rivest, E. Shen, A. Sherman, and P. Vora. Scantegrity II municipal election at Takoma Park: The first E2E binding governmental election with ballot privacy. In *USENIX Security Symp.*, pages 291–306, 2010.
- [14] D. Chaum, R. Carback, J. Clark, A. Essex, S. Popoveniuc, R. Rivest, P. Ryan, E. Shen, A. Sherman, and P. Vora. Scantegrity II: End-to-end verifiability by voters of optical scan elections through confirmation codes. *Information Forensics and Security, IEEE Transactions on*, 4(4):611–627, Dec 2009.
- [15] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.
- [16] D. L. Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE security & privacy*, 2(1):38–47, 2004.
- [17] R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Y. Desmedt, editor, *Crypto'94*, volume 839 of *LNCS*, pages 174–187. Springer, 1994.
- [18] C. Culnane, P. Y. A. Ryan, S. Schneider, and V. Teague. vVote: A verifiable voting system. *ACM Trans. Inf. Syst. Secur.*, 18(1):3:1–3:30, June 2015.
- [19] W. Diffie and M. E. Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, Nov 1976.
- [20] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Crypto'86*, volume 263 of *LNCS*, pages 186–194. Springer, 1987.
- [21] K. Fisher, R. Carback, and A. T. Sherman. Punchscan: Introduction and system definition of a high-integrity election system. In *Workshop on Trustworthy Elections (WOTE)*, 2006.
- [22] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.
- [23] F. Hao, D. Clarke, and A. Zorzo. Deleting secret data with public verifiability. *IEEE Transactions on Dependable and Secure Computing*, PP(99):1, 2015.
- [24] F. Hao, M. N. Kreeger, B. Randell, D. Clarke, S. F. Shahandashti, and P. H.-J. Lee. Every vote counts: Ensuring integrity in large-scale electronic voting. *USENIX Journal of Election Technology & Systems*, 2(3):1–25, 2014.
- [25] A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. In *Privacy in Electronic Society, WPES'05*, pages 61–70. ACM, 2005.
- [26] A. Lim. Travis County, TX developing electronic voting system with a paper trail. *Government Technology*, July 2014. [www.govtech.com](http://www.govtech.com) (accessed Oct. 2015).
- [27] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC press, 1996.
- [28] C. A. Neff. Practical high certainty intent verification for encrypted votes, 2004. Available from <http://citeseer.ist.psu.edu>.
- [29] P. Ryan, D. Bismark, J. Heather, S. Schneider, and Z. Xia. Prêt à Voter: a voter-verifiable voting system. *IEEE T. Inf. Foren. Sec.*, 4(4):662–673, Dec 2009.
- [30] K. Sako and J. Kilian. Receipt-free mix-type voting scheme. In *EuroCrypt'95*, volume 921 of *LNCS*, pages 393–403. Springer, 1995.
- [31] D. Sandler, K. Derr, and D. S. Wallach. VoteBox: A tamper-evident, verifiable electronic voting system. In *USENIX Security Symp.*, volume 4, page 87, 2008.
- [32] C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174, 1991.

## A Proof of Lemma 1

We first consider the following assumption and prove that it is implied by DDH:

**Assumption 2.** For randomly chosen  $a, b \in \mathbb{Z}_q^*$ , given  $(g, g^b, g^{ab}, \Omega)$  where  $\Omega \in \{g^a, g^{a+1}\}$ , it is hard to decide whether  $\Omega = g^a$  or  $\Omega = g^{a+1}$ .

**Lemma 2.** *The DDH assumption implies Assumption 2.*

*Proof.* Taking  $h = g^b$  as the new generator, and assuming  $x = a$  and  $y = b^{-1}$ , we have  $g = h^y$ ,  $g^b = h$ ,  $g^{ab} = h^x$ , and  $g^a = h^{xy}$ . Therefore, the assumption can be rewritten as follows for generator  $h$ : for randomly chosen  $x, y \in \mathbb{Z}_q^*$ , given  $(h, h^x, h^y, \Omega)$ , where  $\Omega \in \{h^{xy}, h^{xy+1}\}$ , it is hard to decide whether  $\Omega = h^{xy}$  or  $\Omega = h^{xy+1}$ . This assumption is proven to be implied by DDH by Hao et al. [24] and hence the proof is complete.  $\square$

Now we show that Lemma 1 holds under Assumption 2.

*Proof (of Lemma 1).* Let A be an adversary that, after determining a number of votes and obtaining temporary access to the voting machine, distinguishes the two bulletin boards. We construct an algorithm D that given  $g, g^b, g^{ab}$ , and a challenge  $\Omega \in \{g^a, g^{a+1}\}$  distinguishes which  $\Omega$  is given.

Consider an abridged bulletin board resulting from removing the well-formedness proofs. Let us call this the *bare* bulletin board. Let the adversary determine any subset of votes other than the swapped votes  $v_i$  and  $v_j$ . A has access to the bulletin board. Furthermore, A has temporary access to the voting machine which means it can observe some votes  $v_k$  and their respective secret values  $r_k$ , and also the value of  $s = \sum_{\ell=1}^k r_\ell$  for the duration of its access. Therefore, apart from simulating the values on the bulletin board, D ought to provide the adversary with the values of  $r_k$  and  $s = \sum_{\ell=1}^k r_\ell$  for a subset of the votes cast or audited during the adversarial access period.

D simulates the bare bulletin board as follows. We describe how confirmed ballots are constructed. Audited ballots can be easily calculated since  $r_k$  and  $v_k$  are known to D for all  $k \notin \{i, j\}$ . Recall that ballots  $i$  and  $j$  are confirmed ballot, both cast either before or after the adversarial access period.

D posts  $g_1 = g$  and  $g_2 = g^b$  as the initial parameters on the bulletin board. For all  $k \notin \{i, j\}$ , D simply chooses  $r_k$  randomly and generates the ballot according to the protocol. D generates random  $\alpha_i$  and  $\alpha_j$  and calculates the  $i$ -th and  $j$ -th ballots as follows. First, D sets

$$R_i = (g^b)^{\alpha_i} g^{ab}, \quad Z_i = g^{\alpha_i} \Omega, \quad R_j = (g^b)^{\alpha_j} / g^{ab}, \quad Z_j = g^{\alpha_j+1} / \Omega.$$

Assuming implicitly that  $r_i = \alpha_i + a$  and  $r_j = \alpha_j - a$ , we can see that  $R_i$  and  $R_j$  are well-formed since:

$$R_i = (g^b)^{\alpha_i} g^{ab} = (g^b)^{\alpha_i+a} = g_2^{r_i}, \quad R_j = (g^b)^{\alpha_j} / g^{ab} = (g^b)^{\alpha_j-a} = g_2^{r_j}.$$

Now if  $\Omega = g^a$ , then we have

$$Z_i = g^{\alpha_i} \Omega = g^{\alpha_i+a} = g_1^{r_i}, \quad Z_j = g^{\alpha_j+1} / \Omega = g^{\alpha_j-a} g = g_1^{r_j} g_1.$$

On the other hand, if  $\Omega = g^{a+1}$ , then we have

$$Z_i = g^{\alpha_i} \Omega = g^{\alpha_i+a} g = g_1^{r_i} g_1, \quad Z_j = g^{\alpha_j+1} / \Omega = g^{\alpha_j-a} = g_1^{r_j}.$$

In other words,  $\Omega = g^a$  corresponds to a bulletin board with  $v_i = 0$  and  $v_j = 1$ , and  $\Omega = g^{a+1}$  corresponds to a bulletin board with  $v_i = 1$  and  $v_j = 0$ , with all other votes being identical in the two bulletin boards.

Since all the votes other than  $v_i$  and  $v_j$  are known to D, it can calculate the partial tallies of the votes other than  $v_i$  and  $v_j$  cast *before*, *during*, and *after* the adversarial access period. In addition, we have either  $v_i = 0$  and  $v_j = 1$ , or  $v_i = 1$  and  $v_j = 0$ , hence  $v_i + v_j = 1$ . So whether both  $v_i$  and  $v_j$  are cast before or after the adversarial access period, the partial tallies of all votes (including  $v_i$  and  $v_j$ ) cast before, during, and after the the adversarial access period can be easily calculated by D.

A similar argument holds for the random values: all random values except for  $r_i$  and  $r_j$  are known to D, and for  $r_i$  and  $r_j$  we implicitly have:

$$r_i + r_j = (\alpha_i + a) + (\alpha_j - a) = \alpha_i + \alpha_j$$

which means that  $r_i + r_j$  is known to D. Hence following a similar reasoning, whether both  $v_i$  and  $v_j$  are cast before or after the adversarial access period, the partial sums of all random values (including  $r_i$  and  $r_j$ ) for votes cast before, during, and after the the adversarial access period can be easily simulated by D.

Thus, D is able to simulate all the elements of a bare bulletin board and the internal DRE information revealed to the A during the adversarial access period. Since the well-formedness proofs are assumed to be zero knowledge, they can be simulated in the Random Oracle Model for ballots  $i$  and  $j$ , and the simulated proofs remain indistinguishable from real proofs. Consequently, D is able to simulate a full bulletin board corresponding to one of the two cases, with  $\Omega = g^a$  corresponding to the case where  $v_i = 0$  and  $v_j = 1$ , and  $\Omega = g^{a+1}$  corresponding to  $v_i = 1$  and  $v_j = 0$ , with all other votes being identical in the two bulletin boards. Now if A is able to distinguish the two cases, D will be able to distinguish whether  $\Omega = g^a$  or  $\Omega = g^{a+1}$ , and hence the proof is complete.  $\square$

## B Well-Formedness Proofs

The first type of proofs are proofs of equality and knowledge of two discrete logarithms. The proof generation and verification procedures are shown in Algorithms 1 and 2 on page 19, respectively.

The second type of proofs are disjunctive proofs of equality and knowledge of either a first pair of discrete logarithms or a second pair. The proof generation and verification procedures are shown in Algorithms 3 and 4 on page 19, respectively. Algorithm 3 is written for the case where the prover knows the *first* pair of discrete logarithms. The algorithm for the case where the prover knows the *second* pair can be obtained by straightforward modifications.

---

**Algorithm 1:** A prover with identifier  $ID$  generates a proof of knowledge of a secret  $\lambda$  s.t.  $F_1 = \gamma_1^\lambda$  and  $F_2 = \gamma_2^\lambda$  for known  $ID, \gamma_1, F_1, \gamma_2, F_2$

---

**Input:**  $ID, \gamma_1, F_1, \gamma_2, F_2, \lambda$  s.t.  $F_1 = \gamma_1^\lambda$  and  $F_2 = \gamma_2^\lambda$   
**Output:**  $\zeta = \mathbb{R}_k\{\lambda : F_1 = \gamma_1^\lambda \wedge F_2 = \gamma_2^\lambda\}$   
**begin**  
  **choose** random  $w \in \mathbb{Z}_q$   
  **calculate**  $t_1 = \gamma_1^w$  and  $t_2 = \gamma_2^w$   
  **calculate**  $c = H(ID, \gamma_1, F_1, \gamma_2, F_2, t_1, t_2)$   
  **calculate**  $r = w - c\lambda$   
  **return**  $\zeta = (c, r)$

---

**Algorithm 2:** Verification of a proof  $\zeta$  generated by Algorithm 1 against  $ID, \gamma_1, F_1, \gamma_2, F_2$

---

**Input:**  $ID, \gamma_1, F_1, \gamma_2, F_2, \zeta = (c, r)$   
**Output:** valid or invalid  
**begin**  
  **calculate**  $t_1 = \gamma_1^r F_1^c$  and  $t_2 = \gamma_2^r F_2^c$   
  **calculate**  $c' = H(ID, \gamma_1, F_1, \gamma_2, F_2, t_1, t_2)$   
  **if**  $c = c'$  **then return** valid  
  **else return** invalid

---

**Algorithm 3:** A prover with identifier  $ID$  generates a proof of knowledge of a secret  $\lambda$  s.t. either  $F_1 = \gamma_1^\lambda$  and  $F_2 = \gamma_2^\lambda$  or  $F_3 = \gamma_3^\lambda$  and  $F_4 = \gamma_4^\lambda$  for known  $ID, \gamma_1, F_1, \dots, \gamma_4, F_4$

---

**Input:**  $ID, (\gamma_i, F_i)_{i=1}^4, \lambda$  s.t.  $F_1 = \gamma_1^\lambda$  and  $F_2 = \gamma_2^\lambda$   
**Output:**  $\zeta = \mathbb{R}_k\{\lambda : (F_1 = \gamma_1^\lambda \wedge F_2 = \gamma_2^\lambda) \vee (F_3 = \gamma_3^\lambda \wedge F_4 = \gamma_4^\lambda)\}$   
**begin**  
  **choose** random  $w, r_2, c_2 \in \mathbb{Z}_q$   
  **calculate**  $t_1 = \gamma_1^w, t_2 = \gamma_2^w, t_3 = \gamma_3^{r_2} F_3^{c_2}, t_4 = \gamma_4^{r_2} F_4^{c_2}$   
  **calculate**  $c = H(ID, (\gamma_i, F_i)_{i=1}^4, (t_i)_{i=1}^4), c_1 = c - c_2$   
  **calculate**  $r_1 = w - c_1\lambda$   
  **return**  $\zeta = (c_1, c_2, r_1, r_2)$

---

**Algorithm 4:** Verification of a proof  $\zeta$  generated by Algorithm 3 against  $ID, \gamma_1, F_1, \dots, \gamma_4, F_4$

---

**Input:**  $ID, (\gamma_i, F_i)_{i=1}^4, \zeta = (c_1, c_2, r_1, r_2)$   
**Output:** valid or invalid  
**begin**  
  **calculate**  $t_1 = \gamma_1^{r_1} F_1^{c_1}, t_2 = \gamma_2^{r_1} F_2^{c_1}, t_3 = \gamma_3^{r_2} F_3^{c_2}, t_4 = \gamma_4^{r_2} F_4^{c_2}$   
  **calculate**  $c' = H(ID, (\gamma_i, F_i)_{i=1}^4, (t_i)_{i=1}^4)$   
  **if**  $c_1 + c_2 = c'$  **then return** valid  
  **else return** invalid

---