A strongly software independent voting system

Prashant Agrawal[*†] Kabir Tomer[*] Subodh Sharma[‡] Subhashis Banerjee[‡]

Email: {*prashant, cs5170410, svs, suban*}*@cse.iitd.ac.in*

Computer Science and Engineering
Indian Institute of Technology Delhi
New Delhi 110016

1 Introduction

As recent events have demonstrated, conducting large scale public elections in a dispute free manner is not an easy task. End-to-end verifiable (E2E-V) cryptographic voting systems can theoretically guarantee correctness of elections, but they are not yet very popular. Not only do they shift significant parts of the responsibilities for audit from the authorities to individual voters, but also the following observation made by the German Constitutional Court while banning electronic elections [NDI, 2019]

> "The use of voting machines which electronically record the voters' votes and electronically ascertain the election result only meets the constitutional requirements if the essential steps of the voting and of the ascertainment of the result can be examined reliably and without any specialist knowledge of the subject . . .
>
> The legislature is not prevented from using electronic voting machines in elections if the possibility of a reliable examination of correctness, which is constitutionally prescribed, is safeguarded. A complementary examination by the voter, by the electoral bodies or the general public is possible for example with electronic voting machines in which the votes are recorded in another way beside electronic storage."

makes depending entirely on cryptographic guarantees somewhat untenable. Moreover, in case audits fail and elections do not verify, the E2E-V systems do not usually provide easy methods of recovery without necessitating complete re-election [Bernhard et al., 2017].

The other approach that is often advocated for verifiability of electronic voting in a software-independent manner is to also maintain voter-verified paper records (VVPR). These may be voter-marked paper ballots or marked using ballot-marking machines but verified by voters, which can be preserved for statistical audit of electronic counts using rigorous *risk-limiting audit* (RLA) techniques [Stark and Wagner, 2012, Bernhard et al., 2017]. In fact, in a recent report, the national academies in the USA have recommended conducting elections only with such human readable paper ballots [National Academies of Sciences, Engineering and Medicine, 2018], with or without electronic counting. However, such an approach requires ensuring that the VVPRs, which may be trustworthy at the time of voting, also remain trustworthy at the time of counting or auditing. Verification of the trustworthiness of the custody chain of VVPRs - to the satisfaction of all interested parties including candidates and voters - requires strict *compliance audit* [Stark and Wagner, 2012, Bernhard et al., 2017] and corresponding guarantees using traditional non-computer science based methods. Assuming that the VVPRs are trustworthy, RLA can provide a reliable method for recovery from incorrect electronic counting, if necessary by complete manual counting of paper ballots.

In this paper we propose two novel E2E-V polling booth voting protocols that support VVPRs and also provide for verifiability of the compliance audit process. They are thus *strongly software independent* [Rivest, 2008], i.e., they support recovery from errors - if possible at all - in case elections do not verify. In fact, in such a case, the protocols enable a diagnosis at polling booth or even voter level granularity without compromising individual or community vote secrecy.

[*] Prashant Agrawal and Kabir Tomer contributed equally to this work.

[†] Prashant Agrawal is supported by the Pankaj Jalote doctoral grant.

[‡] Subodh Sharma and Subhashis Banerjee are also associated with the School of Public Policy, IIT Delhi

Most E2E-V systems encrypt the cast votes and display only the encrypted votes on public bulletin boards to enable voters to verify that their cast votes are recorded in the tally. Usually the guarantees provided for the correctness of the decryption are universal and in case of an error it is not possible to narrow down the problem. As such, it is not possible in these systems to demonstrate a one-to-one correspondence between the VVPRs - which are necessarily in clear text to facilitate manual counting - and the displayed items on the bulletin boards since that will compromise voter secrecy. In contrast, we maintain a publicly verifiable one-to-one correspondence between the VVPRs and the votes displayed in clear text on public bulletin boards where the electronic tallies are computed, and allow voters to verify their receipts against the bulletin board using a zero knowledge protocol. The verifiable one-to-one correspondence also resolves the question as to which is the valid definition of a vote, the one on the VVPR or the electronic one?

We present two strongly software-independent E2E-V voting protocols. The first uses optical scanning of paper ballots that are hand marked by the voter, and is hence *dispute-free* [Bernhard et al., 2017]. However, hand-marked paper ballots often have usability issues and result in a large number of 'invalid' ballots. In view of this, in our second proposal we consider a direct-recording electronic (DRE) option. But it is well known that when a person interacts with a machine in private, there is no way to establish that the machine has recorded the voter's intent correctly [Kaczmarek et al., 2013]. We mitigate this problem by requiring a pattern-matching by the voter, printing commitments on paper and requiring a direct channel for a one bit communication between the voter and the polling officer and polling agents (representatives of the candidates) outside the private polling booth independent of the voting machine. We show that the DRE protocol provides *cast-as-intended* guarantees and hence also *dispute resolution* [Bernhard et al., 2017] in case the voter accepts the vote in a clear text printout. The DRE protocol also prevents a type of coercion attack where the coercer can force the voter to cast randomly.

In the first protocol the ballots are completely self contained, and a random sample can be publicly audited on site at the polling booths before the elections. However, in the second protocol, the ballots collected at random from the poll sites can only be audited afterwards. In either case, auditing a statistically significant random sample of ballots guarantees with high probability (using standard techniques) that all ballots are well formed.

2 Our E2E-V requirements and the threat model

E2E-V voting protocols do not rely only on electronic voting machines (EVM) for correctness but try to provide provable guarantees that votes are recorded and tallied correctly [Chaum, 2004, Chaum et al., 2008, Dzieduszycka-Suinat et al., 2015].

In what follows we outline the design requirements for our bare-handed [Chaum, 2004] E2E-V polling booth protocol. We consider single race, and first past the post voting.

2.1 Trust requirement for correctness

Democracy principles demand that it should not be necessary to trust any authorities, individually or collectively, for the correctness of the election process. Moreover, every component of the election process should be publicly auditable.

Eligibility verification is outside the scope of this paper, hence trust on the polling officers for offline identity verification and eligibility checks is unavoidable. However, this trust must be publicly recorded, and we require the polling officers to certify each valid vote.

2.2 Trust requirement for voter secrecy

In any polling system *voter secrecy* must be preserved at all times. Hence, voting systems must never issue a receipt for the cast vote to a voter to ensure that a voter is never able to prove to a coercer or a potential vote buyer who they voted for [Benaloh and Tuinstra, 1994]. *Secrecy* and *receipt-freeness* are necessary conditions for *coercion-free voting*. *Receipt-freeness* however does not prevent from issuing a token receipt to a voter from which no information about who they voted for can be gleaned.

All electronic voting systems need to trust the security and privacy implementations to protect the recorded vote and the cryptography secrets, both at the front-end and the back-end, and also the custody chain of authorities for not compromising voter secrecy. The protocol itself must guarantee not to leak information.

2.3 Correctness guarantees

The overall correctness of voting is established by the correctness of three steps: *cast-as-intended* indicating that the voting machine has registered the vote correctly, *recorded-as-cast* indicating the cast vote is correctly included in the final tally, and *counted-as recorded* indicating that final tally is correctly computed. *Recorded-as-intended* is a composition of the first two.

A voting system must also be *free of spurious vote injection*, at all times before, during or after polling. There must be guarantee that no votes are recorded and tallied other than those approved and certified by the polling officer.

2.4 Universal verifiability

A voting system is *universally verifiable* if it can provide provable recorded-as-cast and counted-as-recorded guarantees for every vote, either deterministically or with a high probability. It must also guarantee that there are no spurious vote injection. Universal verifiability implies that a system is auditable.

2.5 Individual verifiability

Individually verifiable usually implies [Cortier and Lallemand, 2018, Castelló, 2016] that every voter can verify that their vote is cast-as-intended and is recorded in the final list to be counted.

We *extend* the traditional definition of individual verifiability to further include that voters can proactively seek sound and complete individual proofs that their votes are also recorded-as-intended and counted-as-recorded. The proof of individual verifiability should be available on demand, and if it depends on a global universally verifiable component, then that component should be publicly auditable. Every voter should be able to trace their vote to the tally for their chosen candidate and verify the tally.

Such individual verifiability is at the very root of voter confidence in electoral democracy.

2.6 Strong software independence

A voting system is *software-independent* if an undetected change or error in its software cannot cause an undetectable change or error in an election outcome [Rivest, 2008]. Software independence is a necessary condition for universal verifiability, because hardware-software verifiability of a system such as an EVM is almost surely an intractable (at least NP-Hard) problem [Mercuri, 1992].

A voting system is *strongly software-independent* [Rivest, 2008] if a detected change or error in an election outcome (due to a change or error in the software) can be corrected without re-running the election. This will almost certainly require depending entirely on the trustworthiness of the VVPRs during RLA. We modify the definition to imply that the verification failure can be narrowed down to polling booth or even individual level granularity, so that, if absolutely required, the problem can be rectified by re-running the election locally before announcing results.

2.7 VVPR verifiability

We require the VVPRs to be in one-to-one correspondence with the electronically recorded votes at the time of counting either of them. Note that this still requires defining a use-case protocol for recovery from error in case an election does not verify. However, this provides more options for *strong software independence* and does not need to repose entire trust on *compliance audit* of VVPRs [Stark and Wagner, 2012, Bernhard et al., 2017]. In particular, such one-to-one correspondence enables identifying exactly which votes do not verify.

2.8 Dispute resolution

Effective dispute resolution requires a process for clear determination in favour of either the voter or the election authority in case of a challenge, without compromising voter secrecy. Individual verifiability is necessary to establish that every cast vote is *non-repudiable*, i.e., a voter cannot later claim that their vote was not recorded or counted correctly. Non-repudiability of a cast vote cannot be established without the election authority being able to provide a sound proof of recorded-as-intended, or compromising on voter secrecy. In pure DRE systems, dispute resolution is additionally difficult because a vote cast by the press of a button in private is not non-repudiable [Kaczmarek et al., 2013].

Dispute resolution also requires non-repudiability of the verification receipts issued to voters by the election and polling authorities. This, in turn, requires all receipts to be duly signed.

2.9 Large aggregation

Finally, making the vote tally of an EVM or a polling booth - typically of a few thousand voters - public may enable profiling of a locality or a community. Hence, it is essential to aggregate the votes over several polling booths and EVMs leading upto perhaps even an entire constituency before making the tally public. *Large aggregations* are essential for community privacy, and this necessitates preventing identification of polling booth level voting patterns during tallying and RLA. This also requires aggregating the VVPR slips at a central place before RLA.

2.10 Threat model

In view of the trust requirements highlighted in Sections 2.1 and 2.2, we consider the following threat model. We assume a polynomial-time and polynomial-space bounded adversary that can

1. alter or delete cast votes in the EVM or the VVPR box during polling, during the collection process, or while publishing on public bulletin boards;

2. introduce fake votes in the system, i.e., those not approved by polling officers;

3. try to determine a given voter's vote or polling booth level voting patterns from the issued receipts, VVPR slips or public bulletin boards;

4. try to determine a voter's identity from the VVPR slips or the public bulletin board where votes are tallied;

5. try to coerce the voter into voting in a way different from their true preference;

6. be a malicious voter and complain that her vote was not correctly recorded even when it was.

We also assume that

1. The people managing the election cannot access the secrets of the global cryptographic setup (e.g., discrete log of the generators).

2. The EVM and the backend machines of the election authorities maintain cast votes and other secrets securely. While some other protocols depend on multiple mix servers held by independent trustees to distribute these secrets, in practice there is only one election authority that controls all backend machines and trust on this authority is unavoidable. However, we give some mitigation strategies to reduce this trust in the beginning of Section 6. Further, we necessarily need the EVM to read the cast vote to be able to print the VVPR in clear text. To mitigate the risk of a tampered EVM leaking the vote, hardware-based sandboxed environments McKeen et al. [2013] may be deployed at the EVM.

3. The polling officer in any polling booth cannot authorise fake votes by certifying them with digital signatures. This is reasonable because polling officers operate under public scrutiny (especially of the polling agents who are representatives of the candidates).

4. The custody chain of ballots from printing to the polling booths is secure, and no adversary can peek into any to-be-used ballots. This can be ensured by using covered and sealed ballots or scratch surfaces, etc.

We require that the properties highlighted in Sections 2.4-2.9 should hold under the above threat model.

3 Existing E2E protocols and their limitations

In this section, we review some E2E-V voting techniques and protocols with respect to the design objectives outlined above.

3.1 Hand marked paper ballot based frontends

There are several E2E-V voting systems that use optical scanning of hand marked paper ballots. Some popular examples are Scratch & Vote [Adida and Rivest, 2006], Punchscan [Essex and Clark, 2007], Prêt à voter [Ryan et al., 2009], Scantegrity I [Chaum et al., 2008], Scantegrity II [Chaum et al., 2009] and Scantegrity III [Sherman et al., 2011]. The voters are required to mark their choices in the ballots, get them scanned and take home an encrypted part as receipt. Since the receipts are encrypted, the votes cannot be determined from them. Post polling, the encrypted receipts are displayed on public bulletin boards from where voters can verify that their votes have been recorded correctly. The ballot encryptions can be publicly audited by collecting a statistically significant random sample - which are not to be used for voting - either on the spot in case the ballot encryption is self contained as in the case of Scratch & Vote [Adida and Rivest, 2006], or later in case the encryptions are maintained in backend cryptographic commitments as in the case of most others.

Since the ballots are hand marked, they are, by construction, *dispute free* [Bernhard et al., 2017]. However, in some of these schemes like Scratch & Vote, Punchscan and Prêt à voter, the vote is encrypted in the scanned part of the ballots, and hence the voting machine cannot know the clear text votes. While that is excellent for privacy, these frontends cannot generate VVPRs in clear text and no RLA is possible. These methods need to rely entirely on end-to-end cryptographic verifiability. If VVPRs need to be generated it will need to be through a completely decoupled independent system. Some of these, where the candidate orders are random, are also susceptible to a coercion attack where a coercer may force a voter to cast her vote randomly instead of for a candidate of her choice.

3.2 DRE fontends

The DRE systems where the voters directly enter their choice by pressing a button can print VVPRs in clear text. However, dispute resolution is problematic in such systems [Kaczmarek et al., 2013]. When a voter transacts with a machine in the privacy of a polling booth without any witness, and raises a dispute that the machine has not recorded her choice correctly, there is no way to determine whether the voter lied or not. In such a system, if voters can detect at the polling booth that the machine did not cast their vote as intended, they may be allowed to revote [Kaczmarek et al., 2013].

Markpledge [Adida and Neff, 2009] and Bingo voting [Bohli et al., 2007] are DRE protocols that provide sound cast-as-intended guarantees. Markpledge achieves this through generating a printed transcript for a human verifiable interactive zero-knowledge proof (ZKP) [Goldwasser et al., 1985] where a voter sends her own random challenge to the voting machine. Simulated transcripts are generated for all other vote choices to make the receipt coercion free. Bingo voting relies on the availability of a trusted random number generator (RNG) in the polling booth. The voter needs to verify that the random number displayed on the RNG is indeed the one printed on her receipt against her choice. The challenge/response protocol in Markpledge and the requirement of a trusted RNG and matching random numbers in Bingo voting may have usability and practicality issues.

In STAR-Vote [Bell et al., 2013] there is an optional cast-or-audit challenge step after the printouts containing the encrypted cryptographic commitments to the votes are generated where a voter can choose to challenge the commitment on the receipt, in which case the commitment would be opened to demonstrate that it was indeed correct. Although this does not allow a given voter to detect if her *cast* vote was as intended or not, a statistically significant number of voters opting to challenge can ensure that the probability of detecting any malicious incorrect commitment is sufficiently high. Even then the protocol is vulnerable to selective targeting of individual voters who can be guessed to be unlikely to challenge.

3.3 Backends

Most E2E-V voting systems either use homomorphic encryption and tallying or mixnet [Chaum, 1981] based decryption.

In homomorphic encryption based systems like Scratch & Vote [Adida and Rivest, 2006] and STAR-Vote [Bell et al., 2013], the encrypted votes are displayed on public bulletin boards which the voters can match with their receipts. A homomorphic computation can be publicly carried out to obtain the encrypted tally of all candidates. The authorities publicly provide a zero-knowledge proof of the correct decryption of the tally.

In mixnet-based systems like Punchscan [Essex and Clark, 2007], Prêt à voter [Ryan et al., 2009], Scantegrity I [Chaum et al., 2008], Scantegrity II [Chaum et al., 2009] and Markpledge [Adida and Neff, 2009], the encrypted votes available on the bulletin board are decrypted by a robust universally-verifiable backend mixnet that shuffles and decrypts cast ballots, then proves it did so correctly.

There are systems that are not based on either homomorphic encryption or mixnet. For example, Scantegrity III [Sherman et al., 2011] provides individual verifiability of backend decryption using the receipts, and Bingo voting [Bohli et al., 2007] provides a non-interactive zero-knowledge proof (NIZK) of the backend decryption and tallying.

In case of homomorphic tallying, only a *universal* guarantee for the correctness of decryption of the final tally is provided from which it cannot be identified which encrypted votes may be incorrect. Note that the community voter secrecy requirement prevents tagging polling booth information to the decrypted votes, or computing the tally at a smaller polling booth level granularity.

In addition, homomorphic tallying cannot directly support VVPRs since the clear text votes corresponding to the encrypted votes are never revealed. Even for DRE systems using mixnet or mixnet-like backends, supporting VVPRs requires the plaintext votes decrypted by the mixnet to also contain corresponding ballot identifiers against which VVPRs could be matched. STAR-Vote, which is based on homomorphic tallying, supports VVPRs through a special mechanism which we describe in Section 3.4.

3.4 Bringing VVPRs in one-to-one correspondence with electronic records

STAR-Vote [Bell et al., 2013] is a DRE protocol that actually generates VVPRs for RLA. Voters enter their votes in a voting terminal which prints two items. The first is the VVPR which contains the voter's choice in clear text and a random ballot id sequence number, bid. The second is a take home receipt that identifies the voting terminal, the time of voting and a cryptographic commitment to the vote that does not reveal it.

A hash of the bid, $H(bid)$, is appended to the electronic encrypted vote recorded in the voting machine. The voter needs to deposit the VVPR in a box to complete casting of the vote. Post polling, the voter can match the take home receipt on a public bulletin board, where all encrypted votes are displayed, to ensure that the vote is recorded correctly.

For RLA, the encrypted votes along with the $H(bid)$ hashes are passed through a shuffler - which can be made universally verifiable - and decrypted to generate a list of the $H(bid)$ hashes and the corresponding clear text votes. The auditors pick a random sample of the VVPR slips, compute $H(bid)$ from the bid printed on each of them, lookup the corresponding entry from the shuffled list using the $H(bid)$, and compare the clear text votes.

The voters are in one-to-one correspondence with the items on the bulletin board through the receipts, the items of the bulletin boards are in one-to-one correspondence with the decrypted list at the output of the shuffle, which in turn are in one-to-one correspondence with the VVPR slips through the $H(bid)$. The bulletin board is publicly displayed and some coercers may have access to both the voters' receipts and the public audit process of the VVPRs. This necessitates that the shuffle must be kept secret, and verification, if any, can only be universal. Hence, in case of problems, the diagnosis at the granularity of individuals and polling booths is not possible, and strong software independence can only be established by assuming the integrity of the VVPR slips.

In contrast, we propose a system where

1. the published votes in the tally are always in clear text
2. each individual voter can obtain a direct and independent zero-knowledge proof with their receipts that their vote is recorded correctly in the tally
3. the RLA audit of the VVPR slips can be carried out against the clear text votes using which the electronic tally is computed
4. if the VVPR correspondence with the published tally fails for a VVPR slip, the polling booth information can be decrypted from the VVPR slips in a special audit
5. even without the RLA audit, anyone can verify which encrypted votes are correctly decrypted in the final tally and which are not.

Hence, if necessary, re-elections may be conducted only for those polling booths from where the votes did not verify. We believe that such transparency and individual verifiability are central to voters' confidence in the voting and counting processes.

4 Cryptography basics

Before we describe our protocol we describe some basic cryptographic primitives that we will use in this paper.

Throughout this paper p and q denote large primes such that q divides $p-1$, G_q is a unique cyclic subgroup of $\mathbb{Z}_p^*$ of order q, and g and h are generators of G_q. There exist standard and efficient procedures to generate such prime pairs, and $G_q = \langle g \rangle = \langle h \rangle$. We assume that g and h are system initialized once before the election commences and are publicly known, but the discrete logarithm $\log_g h$ is not known to anybody and is hard to compute.

4.1 Pedersen commitment

Given a message $\rho \in \mathbb{Z}_q$ we use the Pedersen commitment scheme [Pedersen, 1992], $C = g^{\rho}h^{r}$, where $r \in \mathbb{Z}_q$ is a secret randomness, to compute a value C that hides ρ.

Pedersen commitment is *perfectly hiding* because C gives no information about ρ. The *binding* property, which demands that given a commitment C, it is hard to compute a different pair of message and randomness (ρ', r') with the same commitment, is derived from the hardness of the *discrete logarithm problem*, because distinct openings (ρ, r) and (ρ', r') of a given commitment $g^{\rho}h^{r} = g^{\rho'}h^{r'}$ reveal that $\log_g(h) = (\rho - \rho')/(r' - r) \mod q$.

Moreover, Pedersen commitment is additively homomorphic, i.e., if $C_1 = g^{\rho_1}h^{r_1}$ and $C_2 = g^{\rho_2}h^{r_2}$ are commitments of ρ_1 and ρ_2 respectively, then $C_1 * C_2 = g^{\rho_1+\rho_2}h^{r_1+r_2}$ is a commitment of $\rho_1 + \rho_2$.

4.2 ZKP of set membership: C is a commitment of some $\rho_i \in \Phi$

We use the scheme proposed by [Camenisch et al., 2008] to provide a zero knowledge proof that a given commitment C corresponds to a message $\rho \in \Phi$, where Φ is a publicly available set, without revealing the message. If Φ is stored indexed by C, then the ZKP of set membership is computationally efficient and requires only $O(1)$ sized proofs [Camenisch et al., 2008]. See supplementary Section 8 for a description of the procedure and the associated security properties.

5 The protocol

We present two protocols. The first one is based on hand marked paper ballots, and the second one is a DRE protocol that uses pre-printed paper ballots. We assume that a vote is an element from the set $\{0, 1, \ldots, m-1\}$ where m is a small integer.

5.1 Protocol I: using hand marked paper ballots

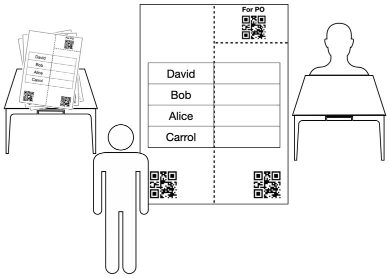

Figure 1: Ballot selection

The voter picks up a random ballot before entering the polling booth (Figure 1). The ballots are self contained and a random selection can be publicly audited - by candidates, their agents and any interested voter - before polling starts. An audited ballot cannot be used for polling, and auditing a statistically significant sample ensures that the probability of a malformed or tampered ballot is negligible.

We propose a Prêt à voter [Ryan et al., 2009] style hand marked ballot which can be optically scanned. However, unlike in Prêt à voter, the scanning machine can determine the clear text vote from the ballot parameters and the voter's choice. We illustrate the ballot design in Figure 2.

$rid \in_R \mathbb{Z}_q$ is a ballot id selected at random from $\mathbb{Z}_q$. $brid$ is a blinded [Chaum, 1983] version of rid printed on the top right part of the ballot. The election authority retains the blinding key at the time of ballot generation. The left half of the ballot contains the candidate list in a random order and a QR code at the bottom containing the code for the random permutation and secrets for the commitments. The remaining detachable right half of the ballot is for the voter to mark her choice, and it also contains in a QR code at the bottom the Pedersen commitment C_{rid} of rid, and the Pedersen commitments of the votes for all the candidates in their given order on the ballot. See Section 7 for a discussion on the required sizes of the QR codes.

All ballots are cryptographically committed after generation. The election authority can provide a zero knowledge proof that each commitment corresponds to a unique rid (see supplementary Section 9). In addition, each component of the ballot is digitally signed by the election authority.

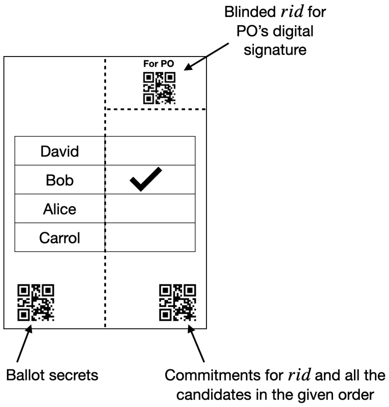
Blinded rid for PO's digital signature
For PO
David
Bob
✓
Alice
Carrol
Ballot secrets
Commitments for rid and all the candidates in the given order

Figure 2: Ballot design for Protocol I

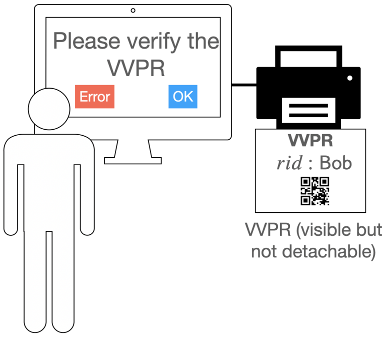

Figure 3: Acknowledgment

A voter enters the polling booth after identity and eligibility verification at a polling officer's desk. The polling officer checks that the voter has picked up a fresh ballot and scans the $brid$. The voter marks her choice on the ballot paper and gets the QR codes on both the left and right sides scanned at an electronic voting machine (EVM). The EVM stores the record $(brid, rid, vote, C_{rid}, C_{vote}, r_v, r_{rid})$, encrypted using a public key of the election authority and indexed by $brid$. Here r_v and r_{rid} are the secrets for the Pedersen commitments C_{rid} and C_{vote} corresponding to the rid and the recorded vote respectively.

The EVM also prints $(rid, vote)$ in clear text on a partial printout of a VVPR slip (Figure 3). If the voter is satisfied with the VVPR slip, she presses the *OK* button to accept, in which case the digitally signed VVPR slip automatically falls into the designated box (Figure 4). If not satisfied, she presses the *Error* button, then a "CANCELLED" symbol is printed on the VVPR and the electronic record is deleted. The VVPR slip may contain a QR code signed by the EVM, which may be electronically checked for validity during the RLA audit. The QR code may also contain the polling booth identifier in an encrypted form.

The voter's *OK/Error* decision should be communicated to the polling officer and the polling agents (representatives of the candidates) in the outside room through a direct channel independent of the EVM. The one bit communication channel should be simple - like a light switch or a bell or even walking out and informing verbally - that can be verified by everybody at the polling booth.

The voter detaches the left part and destroys it in a shredder before leaving the polling booth. If the vote is not cancelled, the voter takes the marked bottom of the right part as a take home receipt and gives the top of the right part to the polling officer. The polling officer and the polling agents must make sure that the $brid$ and the receipts do not get signed in case the voter selects the *Error* button. They may put a bound on the number of times a voter is allowed to cancel.

For all valid votes, the polling officer re-scans the $brid$ to ensure that the ballot that was picked up was used to cast and stores a copy after digitally signing, thereby certifying that the voter has cast her vote after proper identity verification and following the protocol. The polling officer also non-digitally stamps the take home receipt certifying that it was indeed used to cast a vote. It is assumed that the polling officers actions are always in public and are verified by the polling agents.

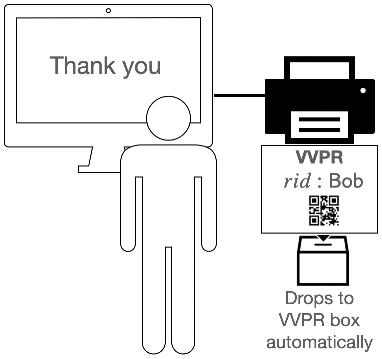

Figure 4: VVPR deposit

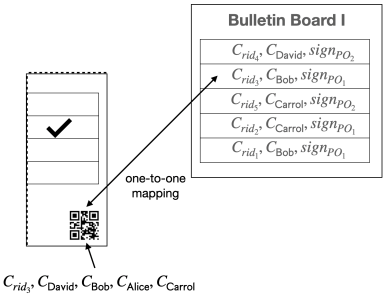
Bulletin Board I
$C_{rid_4}, C_{\text{David}}, sign_{PO_2}$
$C_{rid_3}, C_{\text{Bob}}, sign_{PO_1}$
$C_{rid_5}, C_{\text{Carrol}}, sign_{PO_2}$
$C_{rid_2}, C_{\text{Carrol}}, sign_{PO_1}$
$C_{rid_1}, C_{\text{Bob}}, sign_{PO_1}$
one-to-one
mapping
$C_{rid_3}, C_{\text{David}}, C_{\text{Bob}}, C_{\text{Alice}}, C_{\text{Carrol}}$

Figure 5: Public bulletin board (BB I). The voters can match their receipts.

Post election, the polling officer uploads all the signed $brid$s to the EVM which appends these to the corresponding records. The vote records are collected from all the EVMs and the commitments corresponding to the voter receipts are displayed on a public bulletin board (BB I) [Heather and Lundin, 2009] (see Figure 5), indexed by C_{rid}, where the voter - or their representatives - can match their receipts to satisfy themselves that their votes are recorded correctly.

Finally, the clear text votes are displayed on another public bulletin board (BB II), sorted by the rids, where it can be publicly tallied. Anybody can verify that

1. All items in BB I are certified by a polling officer
2. for each row in BB I, C_{rid} corresponds to a rid for some row in BB II (using the interactive zero-knowledge proof of set membership procedure described in Section 4.2). They can also verify that each C_{rid} corresponds to a unique rid (see supplementary Section 9).
3. for each row in BB I, $C_{rid} * C_{vote}$ corresponds to a $rid + vote$ for some row in BB II (using the interactive zero-knowledge proof of set membership procedure described in Section 4.2).
4. That the $rid + vote$ are unique for each row in BB II.
5. That the sizes of BB I and BB II are equal.

The items on BB II carry no information about which polling booth they came from. Also, the cleartext VVPRs are in one-to-one correspondence with the items on BB II using the $rids$, which can be ascertained during RLA. In case a zero-knowledge proof in steps 2 or 3 fail, the corresponding voter and the polling booth can easily be determined. In case the one-to-one correspondence for a VVPR slip fails, then the polling booth identifier in the slip may be decrypted in a special audit step. We provide correctness arguments against our threat model in Section 6.

Although it is impossible to determine a vote from the receipt, the scheme is vulnerable to a coercion attack where a coercer can force a voter to cast for a random candidate. For example, if there are four candidates on the ballot and the coercer and the voter have different preferences, then, by asking the voter to mark on a pre-decided fixed position, the coercer can ensure that the voter votes for the coercer's candidate choice with a probability $1/4$, with the voter's own choice with a probability $1/4$, and for someone else with a probability $1/2$. Also, hand marking can result in a large

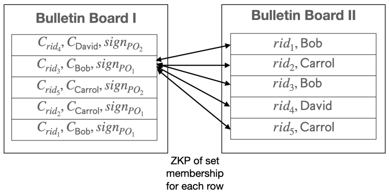
Bulletin Board I
C_{rid_4}, C_{David}, $sign_{PO_2}$
C_{rid_3}, C_{Bob}, $sign_{PO_1}$
C_{rid_5}, C_{Carrol}, $sign_{PO_2}$
C_{rid_2}, C_{Carrol}, $sign_{PO_1}$
C_{rid_1}, C_{Bob}, $sign_{PO_1}$
Bulletin Board II
rid_1, Bob
rid_2, Carrol
rid_3, Bob
rid_4, David
rid_5, Carrol
ZKP of set membership for each row

Figure 6: Public bulletin board (BB II). ZKP that the votes corresponding to the commitments in bulletin board I are in the tally,

number of invalid ballots, especially when the electorate is unfamiliar with hand marking or the voter literacy levels are low.

In view of the above we also present a DRE protocol which is not vulnerable to the above coercion attack.

5.2 Protocol II: DRE

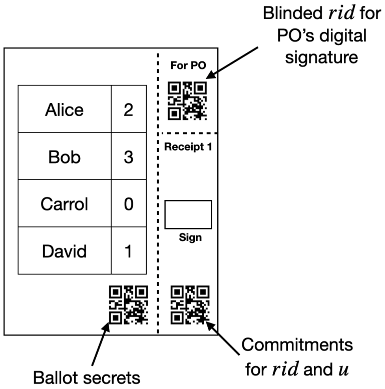
Blinded *rid* for PO's digital signature
For PO
Alice
2
Bob
3
Carrol
0
David
1
Receipt 1
Sign
Ballot secrets
Commitments for *rid* and *u*

Figure 7: Ballot design for Protocol II

Our DRE protocol also uses a paper ballot (Figure 7) without the space for marking votes and without the cryptographic commitments for the votes. The top right QR code contains the $brid$ as before. In the left secret part of the ballot, the candidate order is no longer random but alphabetic. The bottom right QR code contains a random obfuscation token $u \in_R \mathbb{Z}_q$ along with other ballot secrets, and against each candidate is listed a number $w_{(m)} = w \mod m$, where $w = u + vote$ and $vote$ is the integer corresponding to the candidate. The bottom right QR code contains C_{rid} and C_u, the Pedersen commitments for $rid \in_R \mathbb{Z}_q$ and u respectively. The secret keys $r_{rid}, r_u \in \mathbb{Z}_q$ for the commitments are placed in the left bottom QR code along with other ballot secrets. As before, the ballots are pre-committed and the list of C_{rid}s is published.

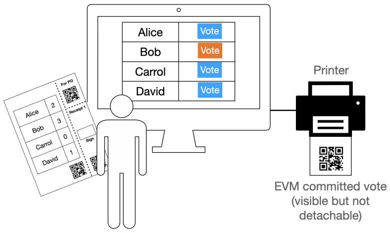

Figure 8: Vote selection at the EVM

After identity and eligibility verification a voter proceeds to the polling booth where the EVM displays an electronic ballot with the candidate order identical to that in the ballot paper, and the voter selects a candidate by press of a button (Figure 8). The EVM prints a Pedersen commitment C_{vote} corresponding to the vote, using the secret $r_{vote} \in \mathbb{Z}_q$, in a partial receipt printout.

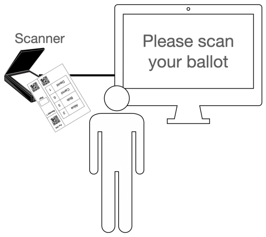

Figure 9: Ballot scan at the EVM

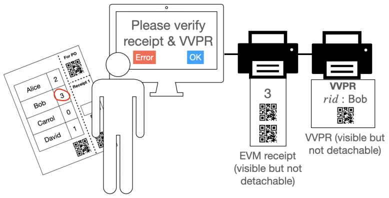
Please verify receipt & VVPR
Error
OK
Alice
2
Bob
3
Carrol
0
David
1
For PO
Receipt 1
3
VVPR
rid : Bob
EVM receipt (visible but not detachable)
VVPR (visible but not detachable)

Figure 10: Receipt verification at the EVM

At this stage the voter is asked to put up the ballot for scanning (Figure 9). All QR codes are scanned. The EVM computes $w_{(m)} = (u + vote) \mod m$ and adds it on to the partial printout of the receipt along with a QR code containing $P = (w_{(m)}, w, r_w = r_u + r_{vote} \mod q)$ (Figure 10). It also prints $(rid, vote)$ in clear text on the partial printout of a VVPR slip.

The voter matches the $w_{(m)}$ on the EVM printout receipt against the one next to her candidate in her ballot. She also checks the clear text vote on the VVPR slip (Figure 10). If the voter is satisfied with the EVM printout receipt as well as the VVPR slip, she presses *OK* button; otherwise she presses *Error*, following which similar steps as Protocol I are taken.

The voter collects her receipt from the EVM (Figure 11) and puts the secret part of the ballot in the shredder. The voter leaves with two receipts - one from the right hand side of the ballot, and the other one from the EVM.

If the voter accepts $w_{(m)}$, then P provides a proof that $C_u * C_{vote}$ is a commitment for w. If the ballot is not malformed and C_u is correct, then P implies that C_{vote} must be correct. With that, the protocol has a dispute resolution strategy built into it.

The remaining part of the protocol - including the voter shredding the left half of the ballot before leaving, detaching the $brid$ on the way out and handing over to the polling officer and getting her receipts (one from the RHS of her ballot containing C_{rid} and C_u, and the other from the EVM containing C_{vote} and P) stamped - remain same as in the previous protocol. The action points for the polling officer and the EVM, and the verification procedures at BB I and BB II and RLA also remain unaltered. The voter's take-home receipts containing $(C_{rid}, C_u, C_{vote}, w_m, w, r_w)$ do not leak the vote because of the hiding property of commitments and because w encrypts $vote$ under (an almost) one-time pad u.

This protocol too, like the previous one, is vulnerable to the coercion attack where a coercer may force a randomization on a voter's choice by asking her to choose a candidate corresponding to a fixed $w_{(m)} \in \{0, \ldots, m-1\}$ from her ballot. Since $w_{(m)}$ is printed on the receipt, the coercer will be able to verify whether the voter has followed the instruction or not. We present a mitigation strategy below.

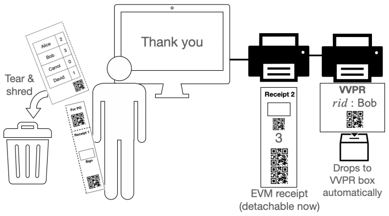

Figure 11: Receipt issue at EVM

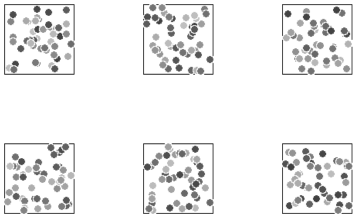

Figure 12: Random patterns. Fixed number of circles of constant sizes are drawn in each square at random positions chosen from a uniform distribution. The shades of the circles are also chosen from a uniform distribution.

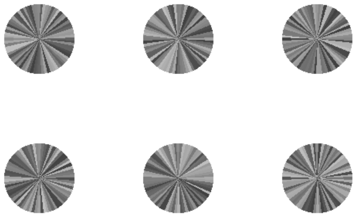

Figure 13: Random patterns. Sectors of circles shaded randomly. The equal sized sectors start from random angles and the shades are drawn from a uniform distribution.

5.3 Random symbols

We propose to replace the integers corresponding to $w_{(m)} \in \{0, \dots, m-1\}$ in the left hand side of the ballots with random patterns of the type depicted in Figure 12 or Figure 13. The random seed for generating the random patterns in the ballot may be added to the bottom left QR code containing the ballot secrets. Correspondingly, the $w_{(m)}$ in the EVM receipt printout in Figure 10 needs to be replaced with the equivalent random pattern. For every ballot, the mapping from the random patterns to the $w_{(m)}$s may be cryptographically committed during ballot generation, and the commitments may be revealed for the chosen pattern for every entry in BB I. Since $w_{(m)}$ does not reveal the vote, this does not compromise voter secrecy.

The only requirements for the patterns are that

1. they are easy to distinguish and match visually
2. the patterns should be random for all candidates
3. there should be a significant probability of any identifiable feature that can be described to an average voter not appearing on any ballot, enabling the voter to avoid any coercion.

6 Verifiability, secrecy, strong software independence and dispute resolution

In this section, we give informal arguments to show that our protocols satisfy the properties identified in Sections 2.4-2.9.

To minimise the trust on the election authority for secrecy (see our threat model in Section 2.10), we propose the following strategy. For distribution of ballots from the election authority to the polling booths, and between bulletin boards BB I and BB II, a verifiable secret shuffle can be introduced to prevent tracking. For the zero-knowledge proofs, their non-interactive versions may be generated once, using ballot secrets kept inside hardware-based sandboxed environments such as McKeen et al. [2013], and then these non-interactive zero knowledge proofs can be published and the ballot secrets can be destroyed.

6.1 Protocol I

6.1.1 Universal verifiability

As per Section 2.4, we need to show that the protocol provides recorded-as-cast and counted-as-recorded guarantees. Counted-as-recorded guarantee is trivial because the votes recorded in BB II are in clear text. For recorded-as-cast guarantee, we show that if C_{rid} and C_{vote} are the commitments in the receipts for rid and $vote$ respectively, and they exist in BB I, then a row $(rid, vote)$ must be recorded in BB II.

Note that anybody can run the ZKP of set membership procedures to verify that commitments $C_{rid} * C_{vote}$ and C_{rid} correspond to some $rid + v$ for some row in BB II, and rid for some (possibly different) row in BB II, respectively. Since anybody can verify whether the C_{rid}'s in BB I are distinct and the rid's in BB II are at distance greater than m of each other, and since $vote \in \{0, \ldots, m-1\}$, this implies that the row containing rid is also the row containing $rid + vote$.

We also show that each commitment C_{rid} in BB I commits a distinct rid. This guarantees that the BB II row corresponding to each $C_{rid} * C_{vote}$ in BB I in the above ZKP is a distinct one.

The computational hardness of the discrete log problem ensures that no attacker can fake the verifiability of the above steps.

In addition, universal verifiability requires that there are no spurious votes. Anybody can verify that all items in BB I are signed by a polling officer. Thus, there are no spurious cast votes in BB I. It is also verified that the sizes of BB I and BB II are equal, which implies that to accommodate a spurious vote in BB II there must be a cast vote in BB I that is not recorded in BB II. But that is not possible as shown above.

We show below that if cast votes do not appear on BB I, then they can be detected through individual verifiability and VVPR verifiability.

6.1.2 Individual verifiability

As discussed in Section 2.5, each voter should individually be able to verify that their vote was recorded as intended and counted as recorded.

The recorded-as-cast and counted-as-recorded parts of this proof are identical to the ZKPs for each row in BB I as described above, but using the commitments in the voter's take-home receipt. For cast-as-intended guarantee it is essential that the ballots are not malformed, the probability of which can be made negligible by a public audit of sufficiently many random ballots in the polling booth verifying the correctness of C_{rid} and C_{vote}s printed next to each candidate in each ballot.

6.1.3 VVPR verifiability

VVPR verifiability does not require a cryptographic proof. Clearly, the VVPRs are in one-to-one correspondence with the clear text votes in BB II that are used to compute the final tally. In case the corresponding entry for a VVPR slip is not found on BB I, a special audit can decrypt the polling booth information.

6.1.4 Strong software independence

Strong software independence requires that for any error in the election processes we should have a way to narrow down the error to avoid re-running the entire election.

RLA depending on the margin between the winner and the second highest vote getter in the electronic count determines the number of VVPR slips that need to be audited and hand counted. In case of a mismatch between the VVPR slip and the clear text vote on BB II, the polling booths information may be decrypted from the VVPR slip. Problematic polling booths may also be identified from voter complaints of failure of individual verifiability, and from cross checking each entry in BB I against BB II to find the entries that do not verify, if any. Depending on the RLA, it may be necessary to rerun the election for those particular polling booths before announcing the result.

6.1.5 Dispute resolution

Protocol I requires hand marking of paper ballots. It is therefore dispute free as per the definition of [Bernhard et al., 2017]. Further, the voter's marked receipts are non-repudiable as they enable a third party to decide if the voter's vote was counted in the final tally or not (see Section 6.1.2).

6.1.6 Vote secrecy

The voter's receipt and her corresponding entry in BB I only contain perfectly hiding Pedersen commitments and thus they do not leak any information about the vote. The tick mark in the receipt (see Figure 2) is against a randomly permuted candidate which does not leak any information about the vote, though the randomization coercion mentioned in Section 5.1 is possible. BB II contains no identifying information that could be related to the voter. BB I and BB II are completely unlinkable to each other since they are sorted in different orders. None of the verifiability steps leak any information because of the zero-knowledgeness of ZKPs of set membership.

6.1.7 Community vote secrecy

BB II - which displays votes in clear text - does not contain any polling booth level identifiers and BB I which contains polling booth information does not reveal any information about any vote. Also, entries in BB I and BB II are completely unlinkable to each other except through ZKP. Hence the protocols preserve community privacy.

6.2 Protocol II

The arguments for Protocol II are exactly the same as Protocol I for *universal verifiability*, *VVPR verifiability*, *strong software independence*, and *community vote secrecy*. We argue the rest of the properties below.

6.2.1 Individual verifiability

The cast-as-intended proof for Protocol II depends on the correctness of C_{vote}, which may be established through verification of proof P in the take-home receipt and a statistical audit for the correctness of C_u printed in the ballot.

Assuming C_u is the correct commitment of the ballot secret u, correctness of C_{vote} is established as follows. Intuitively, this follows from the public audit ensuring the correctness of the ballot commitments, verification of $w_{(m)}$ by each candidate, the receipt verification of Figure 10, and the binding property of Pedersen commitments.

Suppose $C_{vote} = g^{vote'}h^{r_v}$ where $vote' \in \mathbb{Z}_m \backslash \{vote\}$ (each member of the group generated by generators g and h can be expressed in this form). Since $C_u * C_{vote} = g^w h^{r_w}$, we have $g^{u+vote'}h^{r_u+r_v} = g^w h^{r_w}$. Note that it is computationally hard for the adversary to force $(u + vote') \neq w \mod q$ because it would break the binding property of Pedersen commitments.

We also note that as long as $u \in \{0, \dots, q-m-1\}$ and $q \gg m$, the above implies that $(u + vote') = w$ with very high probability.

We show that in such a case, $vote' = vote$, i.e., the voter's intended vote. Let $u_{(m)} = u$ mod m and $u^{(m)} = u$ div m. It is given that $w_{(m)} = w$ mod m. Let $w^{(m)} = w$ div m. We thus have $u^{(m)}.m + u_{(m)} + vote' = w^{(m)}.m + w_{(m)}$. Note that $w_{(m)} = (u_{(m)} + vote) \mod m$ is verified by the voter in Figure 10, and the correctness of the ballot is ensured through public audits.

Case 1: $u_{(m)} + vote < m$. In this case, $w_{(m)} = u_{(m)} + vote$. Thus, we have

$$\begin{aligned} u^{(m)}.m + u_{(m)} + vote' &= w^{(m)}.m + u_{(m)} + vote \\ \therefore \quad u^{(m)}.m + vote' &= w^{(m)}.m + vote \end{aligned} \tag{1}$$

Since $vote < m$ and $vote' < m$, we must have $u^{(m)} = w^{(m)}$ and $vote' = vote$.

Case 2: $m \leq u_{(m)} + vote < 2m - 1$. In this case, $w_{(m)} = u_{(m)} + vote - m$. Thus, we have

$$\begin{aligned} u^{(m)}.m + u_{(m)} + vote' &= w^{(m)}.m + u_{(m)} + vote - m \\ \therefore \quad u^{(m)}.m + vote' &= (w^{(m)} - 1).m + vote \end{aligned} \tag{2}$$

Again, since $vote < m$ and $vote' < m$, we must have $u^{(m)} = w^{(m)} - 1$ and $vote' = vote$.

6.2.2 Dispute resolution

We consider the following possible outcomes of the receipt and VVPR verification steps (Figure 10).

1. If the voter accepts the VVPR and the receipt printouts in the polling booth, then (by Section 6.2.1) the vote is non-repudiable by the voter. Also, if a voter can present valid receipts duly stamped and physically signed by a polling officer that is not present on BB I, then individual verifiability will fail and that would indicate malfeasance on part of the authorities.

2. Suppose in the polling booth the VVPR does not show voter's intended vote or the EVM receipt does not contain $w_{(m)}$ corresponding to the voter's preferred candidate, or both. In such eventualities, we allow the voter to cancel the vote and ask for a revote. The voter may do this only for a bounded number of times. Note that in all such cases,

a cancelled vote is not counted because the PO does not sign the $brid$ present in the voter's ballot. All such records should be ignored from the tally. We assume that the polling officer acts under the oversight of the polling agents and hence cannot cheat, hence the polling officer's signature is not disputable.

3. If the ZKP verifications fail for some voters in BB I, or if some VVPRs do not match, then depending on the RLA it may be necessary to re-run election in some booths.

6.2.3 Vote secrecy

The receipt in Protocol II consists of $(C_{rid}, C_u, C_{vote}, w_{(m)}, w, r_w)$. The commitments do not leak any information about the committed value because of the hiding property of Pedersen commitments, r_w does not reveal anything about secrets r_u and r_v corresponding to commitments C_u and C_v, and w (by extension, $w_{(m)} = w \mod m$) does not leak any information about the vote. Intuitively this is because unless w falls in the interval $[0, m) \cup [q - m, q)$, it acts as an encryption of vote $vote$ using one-time pad u in the group $\mathbb{Z}_q$, and the probability of w falling in this interval is negligibly small for u chosen uniformly from $\mathbb{Z}_q$, since $q \gg m$. We give a more formal reduction argument in Appendix **??**.

The remaining aspects of voter and community secrecy is similar to that of Protocol I.

6.3 Random symbols

The introduction of random symbols does not change the proofs for Protocol II except the following. Cryptographic commitments store the mapping between the symbol to the corresponding $w_{(m)}$. Thus, during polling, voters do not know the $w_{(m)}$ corresponding to the shown symbol and cannot be coerced. Individual and universal verification is only possible after polls close when the election authority opens the commitments. Thereafter the verifiability steps proceed exactly as in Protocol II using the committed $w_{(m)}$.

7 Practicalities of implementation

The ZKP set-membership protocol is based on bilinear maps (see supplementary Section 8), which are usually realized only for elliptic curves. All operations in Section 4 are equally valid for cyclic groups of elliptic curves over a finite field instead of modular subgroups of $\mathbb{Z}_p$. Hence we present our analysis on groups over elliptic curves.

We consider a prototypical high security elliptic curve, a *Type a1* elliptic curve of the PBC library [Lynn, 2013] with a base field size of 1024 bits (p such that $\log p = 1024$), a group order of 160 bits (q such that $\log q = 160$) and an embedding size of 2. The discrete log problem is believed to be hard for such groups. The security of the set-membership protocol depends on the N-SDH assumption [Boneh and Boyen, 2004, Camenisch et al., 2008], where $N = |\Phi|$ is the number of voters in a constituency. For $N \approx 10^6$ the chosen curve is secure against the best known attacks on this assumption [Brown and Gallant, 2004, Cheon, 2006].

7.1 Ballots

The election authority needs to print the signed commitments, random secrets and the blinded rid on each ballot. C_{rid} and C_u are group elements, i.e., points on the elliptic curve, and each requires $2 * \log p = 2048$ bits for representing its two coordinates. A typical high-security RSA modulus is of 2048 bits and RSA signatures in this modulus take 2048 bits. Thus the two commitments and an RSA signature on them take 6144 bits. Ballot secrets $r_{rid}, r_v, r_u, rid, u$ are of $\log q = 160$ bits each. $brid$ for PO's signature takes $2 \log q = 320$ bits (using the blind signature scheme of Mohammed et al. [2000]). Each of the three QR codes of the ballot (see Figures 2 and 7) thus each take less than 1 KB and can easily fit into adjacent QR codes in a piece of paper (the largest QR codes can fit around 3 KB of binary data).

7.2 Computations

During polling, the polling officer machine and the EVM need to be connected to QR code scanners. Using PBC library, the time required to generate the group generators g and h for the elliptic curve we selected were 6.7 *ms* on the average and generating a Pedersen commitment took 28 *ms* on average on commodity hardware. There are no operations, either at the backend for ballot generation, or at the EVM that are more complex than this.

7.3 Receipts

The polling officer needs to blind sign an acknowledgment, which is very efficient [Mohammed et al., 2000]. The EVM needs to print signatures on the voter's receipt. Voter's take-home receipt contains $(C_{rid}, C_u, C_{vote}, w_{(m)}, w, r_w)$ for the voter in succession on a single piece of paper. All of these items can easily fit into modern QR codes ($w_{(m)}$ needs to be printed in clear text). In addition, we need the EVM and the VVPR printers to be able to print partially without allowing the receipt to be detached. The polling booth also needs a shredder.

7.4 ZKPs of set membership

The two ZKPs of set membership (one for C_{rid} and the other for $C_{rid} * C_{vote}$) incur a one-time cost of downloading BB II (of total size $N(2\log q)$ bits - roughly 38 MB for $N = 10^6$) and uploading $2N$ Boneh-Boyen signatures per verifier - N for each ZKP (of total size $2N * (2\log p)$ - roughly 488 MB for $N = 10^6$). Computation of N Boneh-Boyen signatures requires N inverse calculations in group $\mathbb{Z}_q$ and N group exponentiations (scalar multiplications) in the elliptic curve group. Given a particular voter's commitments C_{rid} and C_{vote} though, the set-membership protocol requires an $O(1)$ lookup and a small number of bilinear map evaluations and group exponentiations. Thus, the universal verifiability steps of comparing BB I and BB II are $O(N)$. In our experiments using the PBC library, on average one bilinear map evalutation took 22 *ms*. Standard fast algorithms exist for inverse calculation and group exponentiation too.

Acknowledgements

We thank Partha Dhar and Rajbir Malik for their help with the survey and the initial discussions. We thank Narayanan Kurur for his suggestions on improving the manuscript.

References

Ben Adida and C. Andrew Neff. Efficient receipt-free ballot casting resistant to covert channels. In *Proceedings of the 2009 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*, EVT/WOTE'09, pages 11–11, Berkeley, CA, USA, 2009. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1855491.1855502.

Ben Adida and Ronald L. Rivest. Scratch & vote: Self-contained paper-based cryptographic voting. In *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, WPES '06, pages 29–40, New York, NY, USA, 2006. ACM. ISBN 1-59593-556-8. doi: 10.1145/1179601.1179607. URL http://doi.acm.org/10.1145/1179601.1179607.

Susan Bell, Josh Benaloh, Michael D. Byrne, Dana Debeauvoir, Bryce Eakin, Philip Kortum, Neal McBurnett, Olivier Pereira, Philip B. Stark, Dan S. Wallach, Gail Fisher, Julian Montoya, Michelle Parker, and Michael Winn. Star-vote: A secure, transparent, auditable, and reliable voting system. In *2013 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 13)*, Washington, D.C., 2013. USENIX Association. URL https://www.usenix.org/conference/evtwote13/workshop-program/presentation/bell.

Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 544–553, New York, NY, USA, 1994. ACM. ISBN 0-89791-663-8. doi: 10.1145/195058.195407. URL http://doi.acm.org/10.1145/195058.195407.

Matthew Bernhard, Josh Benaloh, J. Alex Halderman, Ronald L. Rivest, Peter Y. A. Ryan, Philip B. Stark, Vanessa Teague, Poorvi L. Vora, and Dan S. Wallach. Public evidence from secret ballots. In *Electronic Voting - Second International Joint Conference, E-Vote-ID 2017, Bregenz, Austria, October 24-27, 2017, Proceedings*, pages 84–109, 2017. doi: 10.1007/978-3-319-68687-5_6. URL https://doi.org/10.1007/978-3-319-68687-5_6.

Jens-Matthias Bohli, Jörn Müller-Quade, and Stefan Röhrich. Bingo voting: Secure and coercion-free voting using a trusted random number generator. In *Proceedings of the 1st International Conference on E-voting and Identity*, VOTE-ID'07, pages 111–124, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-77492-0, 978-3-540-77492-1. URL http://dl.acm.org/citation.cfm?id=1787456.1787470.

Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan L. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, pages 56–73, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-24676-3.

Daniel R. L. Brown and Robert P. Gallant. The Static Diffie-Hellman Problem. Cryptology ePrint Archive, Report 2004/306, 2004. https://eprint.iacr.org/2004/306.

Jan Camenisch, Rafik Chaabouni, and Abhi Shelat. Efficient protocols for set membership and range proofs. In *Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '08, pages 234–252, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-89254-0. doi: 10.1007/978-3-540-89255-7_15. URL http://dx.doi.org/10.1007/978-3-540-89255-7_15.

Sandra Guasch Castelló. *Individual Verifiability in Electronic Voting*. PhD thesis, Universitat Politècnica de Catalunya, Barcelona, 2016. URL https://upcommons.upc.edu/bitstream/handle/2117/96245/TSGC1de1.pdf.

D. Chaum, A. Essex, R. Carback, J. Clark, S. Popoveniuc, A. Sherman, and P. Vora. Scantegrity: End-to-end voter-verifiable optical-scan voting. *IEEE Security and Privacy*, 6(3):40–46, May 2008. ISSN 1540-7993. doi: 10.1109/MSP.2008.70.

David Chaum. Blind signatures for untraceable payments. In D. Chaum, R.L. Rivest, and A.T. Sherman, editors, *Advances in Cryptology Proceedings of Crypto 82*, pages 199–203, 1983.

David Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security and Privacy*, 2(1):38–47, January 2004. ISSN 1540-7993. doi: 10.1109/MSECP.2004.1264852. URL http://dx.doi.org/10.1109/MSECP.2004.1264852.

David Chaum, Richard T. Carback, Jeremy Clark, Aleksander Essex, Stefan Popoveniuc, Ronald L. Rivest, Peter Y. A. Ryan, Emily Shen, Alan T. Sherman, and Poorvi L. Vora. Scantegrity ii: End-to-end verifiability by voters of optical scan elections through confirmation codes. *Trans. Info. For. Sec.*, 4(4):611–627, December 2009. ISSN 1556-6013. doi: 10.1109/TIFS.2009.2034919. URL https://doi.org/10.1109/TIFS.2009.2034919.

David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, February 1981. ISSN 0001-0782. doi: 10.1145/358549.358563. URL http://doi.acm.org/10.1145/358549.358563.

Jung Hee Cheon. Security Analysis of the Strong Diffie-Hellman Problem. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, pages 1–11, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

Véronique Cortier and Joseph Lallemand. Voting: You can't have privacy without individual verifiability. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 53–66, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356930. doi: 10.1145/3243734.3243762. URL https://doi.org/10.1145/3243734.3243762.

Ronald Cramer, Ivan Damgård, and Philip MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography*, pages 354–372, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. ISBN 978-3-540-46588-1.

Susan Dzieduszycka-Suinat, Judy Murray, Joseph R. Kiniry, Daniel M. Zimmerman, Daniel Wagner, Philip Robinson, Adam Foltzer, and Shpatar Morina. The Future of Voting. End-to-end verifiable internet voting. Specification and feasibility assessment study. Technical report, U.S. VOTE FOUNDATION, July 2015.

Aleks Essex and Jeremy Clark. Punchscan in practice: an e2e election case study. In *Proceedings of IAVoSS Workshop on Trustworthy Elections (WOTE)*, Ottowa, Canada, 2007.

S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, pages 291–304, New York, NY, USA, 1985. ACM. ISBN 0-89791-151-2. doi: 10.1145/22145.22178. URL http://doi.acm.org/10.1145/22145.22178.

James Heather and David Lundin. The append-only web bulletin board. In Pierpaolo Degano, Joshua Guttman, and Fabio Martinelli, editors, *Formal Aspects in Security and Trust*, pages 242–256, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-01465-9.

Tyler Kaczmarek, John Wittrock, Richard Carback, Alex Florescu, Jan Rubio, Noel Runyan, Poorvi L. Vora, and Filip Zagórski. Dispute resolution in accessible voting systems: The design and use of audiotegrity. In James Heather, Steve A. Schneider, and Vanessa Teague, editors, *E-Voting and Identify - 4th International Conference, VoteID 2013, Guildford, UK, July 17-19, 2013. Proceedings*, volume 7985 of *Lecture Notes in Computer Science*, pages 127–141. Springer, 2013. doi: 10.1007/978-3-642-39185-9_8. URL https://doi.org/10.1007/978-3-642-39185-9_8.

Ben Lynn. PBC Library (pbc-0.5.14). https://crypto.stanford.edu/pbc/, 2013. [Accessed June 10, 2019].

Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V. Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R. Savagaonkar. Innovative Instructions and Software Model for Isolated Execution. In *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '13, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450321181. doi: 10.1145/2487726.2488368. URL https://doi.org/10.1145/2487726.2488368.

Rebecca T. Mercuri. Physical verifiability of computer systems. In *In International Computer Virus and Security Conference*, 1992. URL http://www.notablesoftware.com/PENN2008/PhysVerify.pdf.

E. Mohammed, A. E. Emarah, and K. El-Shennawy. A blind signature scheme based on elgamal signature. In *IEEE/AFCEA EUROCOMM 2000. Information Systems for Enhanced Public Safety and Security (Cat. No.00EX405)*, pages 51–53, 2000.

National Academies of Sciences, Engineering and Medicine. *Securing the Vote: Protecting American Democracy*. The National Academies Press, Washington, DC, 2018. ISBN 978-0-309-47647-8. doi: 10.17226/25120. URL https://www.nap.edu/catalog/25120/securing-the-vote-protecting-american-democracy.

NDI. The Constitutionality of Electronic Voting in Germany. https://www.ndi.org/e-voting-guide/examples/constitutionality-of-electronic-voting-germany, 2019. [Accessed June 8, 2019].

Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '91, pages 129–140, London, UK, UK, 1992. Springer-Verlag. ISBN 3-540-55188-3. URL http://dl.acm.org/citation.cfm?id=646756.705507.

Ronald L. Rivest. On the notion of software independence in voting systems. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881):3759–3767, 2008. doi: 10.1098/rsta.2008.0149. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2008.0149.

Peter Y. A. Ryan, David Bismark, James Heather, Steve Schneider, and Zhe Xia. Prêt à voter: A voter-verifiable voting system. *Trans. Info. For. Sec.*, 4(4):662–673, December 2009. ISSN 1556-6013. doi: 10.1109/TIFS.2009.2033233. URL http://dx.doi.org/10.1109/TIFS.2009.2033233.

Alan T. Sherman, Russell A. Fink, Richard Carback, and David Chaum. Scantegrity III: automatic trustworthy receipts, highlighting over/under votes, and full voter verifiability. In Hovav Shacham and Vanessa Teague, editors, *2011 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections, EVT/WOTE '11, San Francisco, CA, USA, August 8-9, 2011*. USENIX Association, 2011. URL https://www.usenix.org/conference/evtwote-11/scantegrity-iii-automatic-trustworthy-receipts-highlighting-overunder-votes.

Philip B. Stark and David A. Wagner. Evidence-based elections. *IEEE Secur. Priv.*, 10(5):33–41, 2012. doi: 10.1109/MSP.2012.62. URL https://doi.org/10.1109/MSP.2012.62.

Common Input:	A Group $G_q = \langle g \rangle = \langle h \rangle$, a commitment C, and a set Φ
Prover Input:	ρ, r such that $C = g^{\rho} h^{r}$ and $\rho \in \Phi$
$P \xleftarrow{y,\{A_i\}} V$	Verifier picks random $x \in \mathbb{Z}_q$ and sends $y \leftarrow g^x$ and $A_i \leftarrow g^{\frac{1}{x+i}}$ for every $i \in \Phi$.
$P \xrightarrow{V} V$	Prover picks random $v \in \mathbb{Z}_p$ and sends $V \leftarrow A_{\rho}^{v}$.
	Prover and verifier run PK$\{(\rho, r, v) : C = g^{\rho} h^{r} \wedge V = g^{\frac{v}{x+\rho}}\}$
$P \xrightarrow{a,D} V$	Prover picks random $s, t, m \in \mathbb{Z}_q$ and sends $a \leftarrow e(V,g)^{-s} e(g,g)^{t}$ and $D \leftarrow g^{s} h^{m}$.
$P \xleftarrow{c} V$	Verifier sends a random challenge $c \in \mathbb{Z}_q$.
$P \xrightarrow{z_\rho, z_v, z_r} V$	Prover sends $z_\rho = s - \rho c$, $z_v = t - vc$, and $z_r = m - rc$,
	Verifier checks that $D \overset{?}{=} C^{c} h^{z_r} g^{z_\rho}$ and that $a \overset{?}{=} e(V,y)^{c} \cdot e(V,g)^{-z_\rho} \cdot e(g,g)^{z_v}$

Figure 14: Set membership protocol for set Φ [Camenisch et al., 2008]

8 ZKP of set membership

The treatment follows [Camenisch et al., 2008]. We include it here for completion.

Definition 8.1. *For an instance of commitment C, a proof of set membership with respect to a set Φ is a* zero knowledge proof of knowledge *of (ρ, r) such that $C = g^{\rho} h^{r} \wedge \rho \in \Phi$.*

The ZKP protocol requires bilinear groups and associated hardness assumptions. Let $\mathcal{G}$ take a security parameter k written in unary as input and output a description of a bilinear group $(p, G_q, G_T, e) \leftarrow \mathcal{G}(1^k)$ such that

1. p is a k-bit prime.
2. G_q, G_T are cyclic groups of order q. Let $G_q^* = G_q \setminus \{1\}$ and let $g \in G_q^*$.
3. $e : G_q \times G_q \rightarrow G_T$ is a bilinear map (pairing) such that $\forall a, b : e(g^a, g^b) = e(g,g)^{ab}$.
4. If g generates G_q then $e(g,g)$ generates G_T.
5. Membership in G_q,G_T can be efficiently decided, group operations and the pairing e are efficiently computable, generators are efficiently sample-able, and the descriptions of the groups and group elements each have size $O(k)$ bits.

The ZKP protocol relies on the Boneh-Boyen short signature scheme [Boneh and Boyen, 2004]. The secret key of the signer is $x \leftarrow \mathbb{Z}_q$, the corresponding public key is $y = g^x$. The signature on a message ρ is $m \leftarrow g^{1/(x+\rho)}$; verification is done by checking that $e(m, y \cdot g^{\rho}) = e(g,g)$. Suppose the $|\Phi|$-Strong Diffie Hellman assumption ($|\Phi|$-SDH) holds in (G_q, G_T), then the basic Boneh-Boyen signature scheme is $|\Phi|$-secure against an existential forgery under a weak chosen message attack [Boneh and Boyen, 2004].

A *honest-verifier zero knowledge proof* under a strong Diffie-Hellman assumption associated with the above pairing generator ($\mathcal{G}$) is given by the set membership protocol in Figure 14 [Camenisch et al., 2008]. Standard techniques exist to efficiently convert an honest-verifier zero knowledge proof to a general zero-knowledge proof [Cramer et al., 2000].

Note that if Φ is stored indexed by C, then the ZKP of set membership is computationally efficient and requires only $O(1)$ sized proofs [Camenisch et al., 2008]. Also, the first communication in Figure 14 from each verifier to the prover needs to happen only once.

9 Each C_{rid} corresponds to a unique rid

In addition to the set membership proof above, we provide ZKPs for each pair of C_{rid_i} and C_{rid_j} that C_{rid_i}/C_{rid_j} is not a commitment for $rid_i - rid_j = 0$. That is, let $C = C_{rid_i}/C_{rid_j} = g^{\rho} h^{r}$ where ρ, r are private knowledge of the prover. The prover needs to demonstrate in ZKP that $\rho \neq 0$. This can be done in the following steps:

1. The prover chooses $s \in \mathbb{Z}_q$ at random and sends $D = h^s \mod p$. The prover also provides a ZKP of knowledge of discrete log of D with base h, i.e., $\log_h D = s$.

2. The verifier sends a random challenge $e \neq 0 \in \mathbb{Z}_q$
3. The prover sends $u = e\rho \mod q$ and $v = s + er \mod q$
4. The verifier accepts if $g^u h^v = DC^e \mod p$ and $h^v \neq DC^e \mod p$

NIZKs for the above can be generated in the standard way after the items are displayed on BB I and BB II respectively.

Note that the more efficient proof of sorting the rids and demonstrating that $C_{rid_{i+1}}/C_{rid_i}$ is not a commitment of 0 leaks information about the ordering since rids are displayed in BB II.