Towards Secure E-Voting with Everlasting Privacy

Avitabile Gennaro[1], Sven Heiberg[2], Helger Lipmaa[3], Janno Siim[4], and Ivan Visconti[5]

[1] University of Salerno, ITALY
[2] Smartmatic-Cybernetica Centre of Excellence for Internet Voting, ESTONIA
[3] Simula UiB, NORWAY
[4] University of Tartu, ESTONIA
[5] University of Salerno, ITALY

Abstract

The digitalization of our society is increasingly transforming and improving traditional physical processes. While in some cases (e.g., e-commerce) the benefits of such transformations are well understood, there still are some important scenarios where the switch towards an electronic system raises severe concerns.

A main debate focuses on the switch from traditional to electronic voting. Indeed, while e-voting has been already implemented in some countries (e.g., Estonia) and can help in increasing voters' participation (e.g., democracy) and in improving the efficiency of the election process, it also brings challenges related to trust, privacy and coercion.

In this paper we discuss the main desirable properties in e-voting and present a preliminary version of a new e-voting system. We focus on end-to-end verifiability (i.e., each voter can verify the correctness of her vote and of the announced result through a bulletin board implemented with a decentralized ledger) and show the design of a practical scheme that aims at offering specific features towards receipt-freeness (i.e., limiting possibilities for a voter to convince others about her vote, therefore mitigating vote buying/selling/coercion issues) and everlasting privacy (i.e., under some circumstances, votes on the bulletin board remain undisclosed for ever, regardless of the future power of the adversary).

1 Introduction

In modern democratic societies democracy is implemented via the election of representative bodies chosen through voting. In such settings, electronic voting is gaining growing interest as a mean to increase voters' participation and the efficiency of the election process. Furthermore, it is also currently applied in many practical scenarios, including political elections (e.g. in Estonia). Moreover, e-voting could benefit every scenario in which multiple parties have to collectively take decisions or to designate representatives, like shareholders meetings or rector elections in universities. There are, of course, many concerns about e-voting systems, especially when they do not provide any way to verify that the election was run honestly.
Indeed, e-voting is a complex process with seemingly contradictory requirements: it should ensure voters' *privacy* while convincing everyone about the legitimacy of the results. One normally expects verifiability through transparency but this could in turn conflict with the privacy requirements. We are all familiar with how transparency and privacy are achieved in more traditional ballot paper elections. When electronic voting is used, the only way to achieve transparency is publishing data about the voting itself that can be accessed by public observers and used for auditing.

End-to-End verifiable voting. In particular, *End-to-End verifiable* voting systems make the election auditable by anyone publishing this audit data on a publicly accessible repository called bulletin board. It should be granted that once items are on the bulletin board then they

will not be removed and that the final information given at the end of the election is fixed and cannot be tampered with. Setting up a bulletin board is not trivial and they are usually implemented in a distributed manner, using Byzantine agreement algorithms [22, 11]. More recently, also blockchains have been investigated and proposed as voting bulletin boards since they are designed to fulfill very similar integrity requirements [20].
In End-to-End verifiable systems, ballot privacy is typically obtained thanks to public-key encryption, where the secret key is split between multiple trustees, who are part of the Election Authority (EA), to ensure that individual votes are never decrypted in isolation.

The importance of zero-knowledge proofs. Zero-knowledge (ZK) proofs demonstrate the truth of a certain statement without leaking any additional information [17]. So, for example, they can be used to prove that a certain vote is valid without exposing the vote itself. ZK proofs can be carried out through an interaction between a Prover and a Verifier, who, at the end of the interaction, will be convinced of the truth of the statement without learning anything else. Non-interactive zero-knowledge (NIZK) proofs, can be generated by the Prover on his own and can be verified many times by different Verifiers, without any interaction with the Prover. NIZK proofs are one of the enablers for universal verifiability.

Homomorphic and mixnet-based voting- Usually voting protocols follow two main paradigms, being based on homomorphic encryption [1] or mixnets [30]:

- When the underlying encryption scheme is homomorphic, the votes are homomorphically combined into a single ciphertext containing the final result, which is then jointly decrypted by the trustees. A ZK proof must be provided along with each vote to ensure that it is *well-formed*, i.e. it does not contain over-votes or negative votes.
- If the underlying encryption scheme is not homomorphic, then the votes are anonymized through a mixnet and subsequently decrypted and tabulated. ZK proofs ensure that the mixing is done correctly (i.e., without replacing any vote with some other vote).

Individual and Universal Verifiability. When the election ends, the EA publishes on the bulletin board the final result together with a proof ensuring that the tally was computed correctly. Each voter can check that his vote will be taken into account looking for a record related to him on the bulletin board. This property is termed *individual verifiability*. Furthermore, anyone can check the correctness of the overall election combining all the data and proofs on the bulletin board. This property is termed *universal verifiability*. *End-to-End verifiable* voting protocols must provide both individual and universal verifiability.

Everlasting Privacy. The introduction of End-to-End verifiability could result in a weakening of privacy. For instance, many End-to-End verifiable schemes publish encrypted votes to the public bulletin board, therefore, individual votes will be revealed if the related cryptosystem is broken for whatever reason. It should be noted that, since storage becomes cheaper everyday, it is reasonable to assume that the bulletin board could be stored for ever. This, however, opens privacy issues since the cryptographic tools used to protect the privacy of the votes could be broken in the future (e.g., thanks to improvements in computing speed, or to the advent of quantum computers, or to advancement in cryptanalysis).

[1] A homomorphic encryption scheme allows to perform a mathematical operation on two plaintexts manipulating only their encryptions, obtaining a ciphertext containing the result of the aforementioned mathematical operation.

Therefore, a voting protocol should provide End-to-End verifiability using audit data that ensure privacy while not relying on computational assumptions, achieving *everlasting privacy* even against unbounded adversaries.
More specifically, if this property is provided only with the respect to the audit data on the bulletin board, it is termed *practical* everlasting privacy [3]. Essentially, an adversary could recover a vote if she manages to intercept the communication between the voter and the EA at the time it takes place; for this reason an honest EA and a private channel between the two are assumed.
An intuitive way to obtain everlasting privacy, is detaching each encrypted ballot from the identity of the voter for example by only publishing pseudonyms. However, this approach would undermine the *eligibility verification*, since it would become impossible to determine if the ballots that are on the bulletin board were cast by legitimate voters or, dishonestly, by the EA.
In addition, votes must be cast through an *anonymous channel*, otherwise it would be possible to discover the identity of the sender of each vote even if it is not explicitly displayed on the bulletin board. If everlasting privacy is provided even against a malicious EA and with the respect to all the data sent by voters to cast a vote, then it is termed *perfect* everlasting privacy. Perfect everlasting privacy usually requires anonymous channels (e.g. TOR [14]).

Receipt Freeness. Another serious threat connected to the auditability of an election is the possibility of coercion: if a voter has a way to combine information in his possession with the audit data related to his vote in order to demonstrate how she voted, then vote buying becomes a serious threat. If a voting protocol does not allow this operation, it is said to be *receipt-free*. There are of course other ways to enable coercion, like selling private credentials or voting for someone else and these may be seen as a violation of the normal voting procedure and should have legal consequences. However, the voting scheme should not provide way to incentivize vote-buying by itself, allowing the voter to provide incontestable evidence that he voted in a particular way, increasing the scalability of vote-buying.

Efficiency. The efficiency of a voting protocol should be assessed both in terms of the voter and the EA computational load. Each of these entities has, of course, different times that would be considered acceptable for their own computations: a voter should be able to cast a vote fairly quickly (e.g. in few seconds), while the results could be computed by the EA even within few hours. Also the size of the ballots must be taken into account when evaluating a voting protocol.
Cramer, Gennaro and Schoenmakers define the optimality of a voting scheme in terms of the above factors seen as a function of the number of voters, authorities composing the EA, and candidates [10]. A voting protocol is said to be optimal in the sense of [10] if:

- The voter computational load and the ballot size do not depend on the number of voters nor authorities.
- The computational load of the tally increases linearly with the number of voters and candidates.

Of course, besides the dependencies between the performance and the number of the various entities involved in the protocol, one should ensure that the computations required to the parties are efficient enough.

About related work. In Appendix A we discuss related work on e-voting, focusing in particular on everlasting privacy and receipt freeness. Summing up, to the best of our knowledge, the

current literature leaves the following problem open: "Is it possible to realize an efficient voting scheme providing both receipt-freeness and everlasting privacy, together with all the properties illustrated above?".

Additionally, we point out the recent publication of Park et al. [28] that in the initial part of the paper seems to convey the message that blockchain-based e-voting exposes many vulnerabilities and should be discouraged. However, when giving a more careful look at the main body of their paper it becomes evident that their concerns apply to the specific examples of blockchain-based e-voting schemes that they propose and their specific definition of a blockchain. Similarly, Heiberg et al. [20] review the attempts to use blockchain in the context of e-voting and conclude that "in all of the proposals we considered, many of the shortcomings and trade-offs of blockchains were addressed insufficiently.".

As Park et al. admit, security depends a lot on the specific interplay of the underlying blockchain with the rest of the e-voting protocol including those steps that are not electronic at all and that clearly are part of any e-voting systems.

We intend to use blockchain technology to facilitate publicly verifiable tally based on the privacy preserving voting mechanism in our proposal, which is different from mainstream proposals that see blockchain as core ballot-box technology. We agree that more research is required to settle a good and complete design for a ready-to-use in real-world scenarios blockchain-based electronic voting system. Experiments in this direction should be encouraged.

1.1 Our Contribution

In this work, we study the above open problem proposing a preliminary design of the first e-voting protocol that provides both receipt-freeness and everlasting privacy without the use of anonymous channels. Our construction provides also eligibility verification and optimality in the sense of [10].

In Table 1, we compare, excluding clearly non-efficient constructions, our proposal with the related work in terms of the properties satisfied by the corresponding voting scheme.

Locher et al. [24] achieve everlasting privacy detaching the identity of the voter from the

Reference	Eligibility Verification	Everlasting Privacy	Receipt-freeness	Optimality in the sense of [10]	Not requiring an anonymous channel
[24]	✓	✓	✓	×	×
[19]	×	✓	✓	✓	×
[7]	✓	×	✓	✓	✓
[12, 29]	✓	✓	×	✓	✓
This Work	✓	✓	✓	✓	✓

Table 1: The symbol ✓ (resp. ×) indicates that the construction satisfies (resp. does not satisfy) the corresponding property.

corresponding ballot and using anonymous channels in combination with perfectly hiding commitments. However, in order to not impair eligibility verification, they use set membership proofs, resulting into a voter computational load that depends on the number of voters. Therefore, [24] is not optimal in the sense of [10].

Grontas et al. [19] use *Publicly Auditable Conditional Blind Signatures*, which are blind signature with some additional features which allow receipt-freeness. However, as in other protocols based on blind signatures, it is not possible to check eligibility. In fact, since the EA has the

signing key, it could cast a ballot on behalf of abstained voters without being detected. The authors argue that abstained voters should be seen as adversarial ones, since also a corrupted registration authority (provider of the voter credentials), without the Public-Key Infrastructure (PKI) assumption, could easily cast a ballot on behalf of them; and that, since the number of voters is known, the EA cannot cast more fraudulent ballots than the adversarial ones. However, this assumption is not reasonable in many practical settings, as political elections, where the abstention rate can be pretty high. For example, in the last European elections of 2019, the abstention rate in Italy was close to 50% [1].
Both [24] and [19] achieve perfect everlasting privacy requiring that ballots are cast through an anonymous channel, which are currently not practical.
In addition, as stated by Grontas et al. [18], the anonymous channel should either ensure anonymity information-theoretically or be run also by external non-governmental agency that would eventually deny the access to the entire anonymous network to the future computationally unbounded adversary willing to break privacy. Grontas et al. [19] provide practical everlasting privacy without assuming anonymous channels, but it still has the cons of not providing eligibility verification.
Chaidos et al. [7] propose a non-interactive voting protocol that, unlike [24] and [19], provide receipt-freeness without requiring re-voting or multiple ballots posted on the bulletin board on behalf of each single voter; however it does not provide everlasting privacy.
The protocol of Pereira et al. [12] is very straightforward and based on well-known primitives and it satisfies all the properties listed in table 1, except receipt-freeness.

High-level overview of our approach. In this work, we improve the work of Pereira et al. adding receipt-freeness. In particular, we refer to one concrete instantiation that uses homomorphic voting, as described in [29].
For the sake of simplicity, let's consider a simple election with just two candidates. In homomorphic voting, the voter submits his vote v encrypted under a homomorphic cryptosystem; a vote for the first candidate is cast encrypting $v = 1$, while a vote for the second candidate is cast encrypting $v = 0$. At the end of the election, the EA can compute the encrypted value of first candidate's votes homomorphically adding all the ciphertexts, and then decrypting a single ciphertext.
The votes for the second candidate are given by the difference between the total number of votes and the votes tallied for the first candidate. Along with his vote, the voter has to provide a NIZK proof in order to ensure that his vote is truly an encryption of 0 or 1, otherwise one could compromise the tally giving an extremely large number of votes to the first candidate.
In [12, 29], the enabler to everlasting privacy is the introduction of a Commitment Consistent Encryption (CCE) scheme, which is an encryption scheme with an additional feature: it is possible to extract from it a perfectly hiding commitment[2] which is consistent with the encrypted value, furthermore, the opening of the commitment can be extracted using the secret key. Both the underlying commitment and the encryption scheme are additively homomorphic.
To cast a ballot, the voter submits a CCE ciphertext which essentially contains a perfectly hiding commitment (Pedersen) to the vote and its encrypted opening, along with a NIZK proof

[2]Commitments are a primitive often used to prevent a player deviating from a protocol: to commit to a certain value v, he generates a commitment using an algorithm that takes v and a random coin r as input. Later on, the generated commitment can be opened using v and r, proving that the player did not change his mind. A commitment should be *hiding*, making impossible for one who does not hold the opening to discover the content of the commitment itself and binding, preventing the possibility to open the commitment with a different v. Both properties can be based on a computational assumption, or can be unconditional (perfect), however, if one property is unconditional, the other one must be based on a computational assumption.

that what is encrypted is actually the opening of the provided commitment (consistency). The voter also provides a NIZK proof that the commitment is well-formed according to the voting rules. After having verified its consistency, the EA extracts the commitment and posts it on the bulletin board along with its well-formedness proof.
Since the commitment is perfectly hiding, if the proofs are also perfect zero knowledge, the audit data on the bulletin board will never leak any information about the votes, even against an unbounded adversary, thus practical everlasting privacy is provided.
Finally, when the result is announced, the EA proves that the tally was computed correctly showing that it can open the homomorphic aggregation of the committed votes to the announced results. This will provide a guarantee of correctness based on a computational assumption, that is the binding of the commitment scheme [3]. The election result and its opening can be extracted by the EA in a similar way: the commitments (ciphertexts) are homomorphically aggregated and the result (opening) is then extracted using the secret key.
It is easy to observe that such protocol is not receipt-free. In fact, if the voter stores the random coins used to compute the CCE ciphertext, he can give them to a coercer along with his declared vote. The coercer can then re-compute the commitment to the vote and check if it matches the voter's record on the bulletin board.
Our starting idea to add receipt-freeness to [29] is to *re-randomize* the commitment to the vote before posting it to the bulletin board, so that the randomness contained in it is not under the total control of the voter, who has now no way to open the commitment and prove how he voted. However, finding a way to re-randomize the audit data without endangering the tally procedure is not sufficient. In fact, after the re-randomization of the audit data, two not trivial issues arise:

- The voter should be convinced that his intent was not changed during the re-randomization.
- [29] uses Σ-protocols with Fiat-Shamir (FS) [4] transform to create NIZK proofs of well-formedness. However, such proofs, that were valid on the original commitments, would not be valid anymore on the re-randomized ones.

We solve both issues formulating a voting protocol that achieves both receipt-freeness and everlasting privacy. The first issue is solved through a Designated Verifier NIZK (DV-NIZK) proof. Such a proof acts as a *personal* receipt: it can be checked by the voter to convince himself that the re-randomization was performed honestly, while it is useless in proving the same to a coercer, so that receipt-freeness is preserved. The second issue is solved going in a different direction compared to many recent literature works, turning the vote casting in an interactive protocol. Although making the vote casting non-interactive could provide an increase in efficiency, it is worth noticing that interaction is not negative per-se, especially if it can provide additional benefits without impacting substantially on the overall efficiency. For example, a lot of interaction goes on behind the scenes each time an HTTPS request is sent, however, this is pretty transparent to the end user. By introducing a three-message interaction between the voter and the EA during the vote casting phase, we allow the EA and the voter to jointly generate a NIZK proof proving the well-formedness of the commitment, while none of the two gains any knowledge about the randomization factor the other one knows, so that both voter

[3] The parameters of the commitment scheme can be selected by extracting them from the output of a cryptographic hash function modeled as random oracle on input publicly known values. In this way every player can derive parameters on its own and no player knows the trapdoor that would allow to violate the properties of the commitment scheme.

[4] Σ-protocols are three-round protocols providing zero knowledge assuming that the verifier is honest. Using a secure hash function assumed to model a random oracle, it is possible to turn a Σ-protocol into a NIZK proof. This transformation is termed Fiat-Shamir from the name of its authors. More detailed descriptions and definitions can be found in [6].

privacy and receipt-freeness are preserved. The proofs generated in this way can eventually be checked by anyone who wants to asses the correctness of the election outcome. This three-message interaction is, indeed, a general construction allowing two parties to perform a joint computation of NIZK proof of knowledge (PoK) of statements based on discrete logarithms (DL). We describe it in greater detail in Appendix C.

1.2 Practical Validation

In order to validate the proposed voting protocol we implemented all its necessary building blocks in Go [2]. The underlying base protocol [12, 29] uses pairings, thus, it requires particular pairing-friendly elliptic curves. Pereira [29], in order to target 128-bit security, suggests to use BN curves with 256-bit group and base field size order. However, their actual security level is now estimated around 110-bit [25]. So, we used BLS curves instead, with 256-bit group order and a base field of 384-bit order, to target 128-bit security level with just slightly worse performances of the former BN curves. We used the implementation of the BLS12-381 curve provided by the Apache Milagro Crypto Library [32]. SHA256 has been used as hash function for the FS transform.
As bulletin board we used the HyperLedger Fabric permissioned blockchain. A permissioned model seems to be more well-suited since it overcomes the uncertainty of confirmation times and geographical centralization of mining power that can be big issues in practical e-voting schemes [20]. We deployed a smart contract that uses our GO library to validate each vote in the moment its related audit data is posted on the bulletin board by the EA.
The integrity of the bulletin board is guaranteed if the majority of the parties maintaining it is honest. These parties could be either independent external auditors, like political parties, or the trustees that are part of the EA. Since this is a preliminary work, the implementation has not been made available yet.

2 Preliminaries, Assumptions and Primitives

For efficiency reasons, the construction of CCE scheme is based on cyclic prime order bilinear groups. For the security of the CCE we require the Decisional Diffie-Hellman (DDH) assumption to hold in both groups. We also require the random oracle model (ROM) assumption to hold, since we use Σ-protocols in combination with the FS transform. We also use a secure digital signature algorithm as ECDSA. A more detailed description of the assumptions and primitives used in our construction can be found in Appendix B.

3 End-to-End Verifiable Voting: Model and Security

In this section we describe the entities involved in an End-to-End verifiable voting protocol and its most common desirable properties. Several models defining the involved parties can be used, however, in the e-voting literature, there is not a particular established setting. We define the following parties who are involved in an election event or have some interests towards it:

- *Election Authority*: denoted by EA, it is responsible for the generation and the management of the election keypair (epk, esk) of the CCE.
  The secret key must be split between multiple trustees composing the EA, so that the decryption of votes can be done only if the majority of them cooperate.

- *Certification Authority*: we require the existence of a PKI. There is a Certification Authority, denoted by CA, and every eligible voter is in possession of a signing keypair (vpk, vsk) certified by the CA. We denote with *cert* the certificate related to vpk.
- *Voters*: they belong to a set of eligible people who have the right to express their preference during the election.
- *Coercer*: he wants to bribe voters to make them vote for a candidate of his choice. He is interested in gaining an incontestable proof that the coerced voters voted according to his instructions.
- *Observers*: they can be anyone who has the interest to verify the election outcome, they want assess the validity of individual votes and of the tally.
- *Bulletin Board*: a public bulletin board is made available to everyone to guarantee the verifiability of the election event.
  We also assume that the bulletin board has the capability to verify the ballots.

We define the following stages of an election event:

- **Preparation**: depending on the security parameter, groups are chosen and made public. The trustees composing the EA generate the CCE keys in a threshold manner using any suitable protocol that can be used for DDH-based cryptosystems (e.g. [16]).
  The bulletin board is created with the description of the election, including a public value called pv containing the description of the used groups and generators. The CCE public key is also published, along with every information that could be needed to verify its validity.
- **Vote Casting**: when the election event starts the voters can cast their vote using the appropriate protocol and can subsequently verify that their vote is included in the bulletin board.
- **Tally Computation**: when the election ends, the trustees composing the EA compute the tally and all the necessary audit data.
- **Announcement of the results and verification**: When the computation of the tally is completed, the EA publishes the results and the audit data on the bulletin board. The correctness of the outcome can be assessed using the universal verification mechanism of the protocol.

We assume that eligible voters are registered and authorized to vote thanks to a valid certified signing keypair already in their possession. In the last years, a considerable number of e-voting schemes have been published and a certain level of agreement has been reached on e-voting scheme desirable properties [5, 8, 7]. Here, due to page limitations, we informally list the most important ones, recalling that they also rely on some computational/trust assumptions:

- *Privacy*: no one can know who voted for whom.
- *Eligibility*: only authorized voters are allowed to vote.
- *Individual Verifiability*: every eligible voter can verify that her vote is counted.
- *Universal Verifiability*: any participant or passive observer can convince herself of the validity of individual votes and of the final tally of the election.
- *Eligibility Verification*: any participant or passive observer can convince himself that all the votes were cast only by eligible voters.
- *Receipt-freeness*: voters must neither be able to obtain nor construct a receipt which can prove the content of their vote.

- *(Practical) Everlasting Privacy*: the audit data published on the bulletin board will never leak any information about the votes, even to an adversary with unlimited computational resources.

A malicious EA could attempt to break privacy trying to learn the choice of single voters or to announce a fake election outcome, for example stuffing the bulletin board with dummy votes not cast by eligible voters or simply trying to announce an arbitrary result that does not come from the legitimate votes. However, the property of privacy, eligibility, individual and universal verifiability prevent a malicious EA from doing the aforementioned actions.
A voter could be rewarded by a coercer if she manages to incontestably prove that she voted accordingly to her instructions, however this is not possible if receipt-freeness holds. Privacy must be preserved also with the respect to any bounded adversary eavesdropping the communication. The EA is instead trusted in order to preserve everlasting privacy and receipt-freeness. In Appendix E, we give an informal security analysis of our protocol.

4 End-to-End Verifiable Voting: Our New Protocol

For the sake of simplicity, we describe the proposed voting protocol for an election with just two candidates. Appendix D shows the extension to n candidates.

4.1 Vote Casting

We define:

- *electionID*, a uniquely defined ID for the election.
- v as the selection made by the voter. It can be either 0 or 1.
- $E = Enc(v, epk)$, as the Commitment Consistent Encryption of the vote. v
- d, as the commitment extractable from E.
- d', as the re-randomization of the commitment d, so that d' opens to the same v of d, but with a different randomness.

The Voter casts the vote interacting with the EA: the vote casting process is based on the protocol of Appendix C, with the Voter proving the well-formedness of d to the EA. In the meanwhile, the EA re-randomizes d into d' and builds NIZK proofs for the well-formedness of d'. In addition, the EA has to prove back to the Voter that this re-randomization does not change his vote intentions. We use the letter a, k and σ to identify a Σ-protocol commitment, challenge and response, respectively. The Voter has to prove that d is a commitment to either 0 or 1, we denote the messages related to this proof with the subscript $0/1$. The EA has to prove that d' is an honest re-randomization of d, we denote the messages related to this proof with the subscript *rand*. Figure 1 shows the high-level functioning of the vote casting phase.
In this protocol the Voter interacts with a single trustee of the EA who is in charge of the re-randomization, therefore,in order to preserve receipt-freeness, this trustee must not collude with the coercer. Note that the Voter signs the $electionID$ and d' to state that d' is part of an approved ballot for the *current* election event.
However, the vote casting process is not completed yet, since the audit data must be posted on the Bulletin Board. The EA sends the just built proof $\sigma'_{0/1}$ and forwards $(cert, S, d', \sigma'_{0/1})$ to the Bulletin Board. Upon receiving, the Bulletin Board performs the following checks:

- Verify the eligibility of the voter through *cert*.

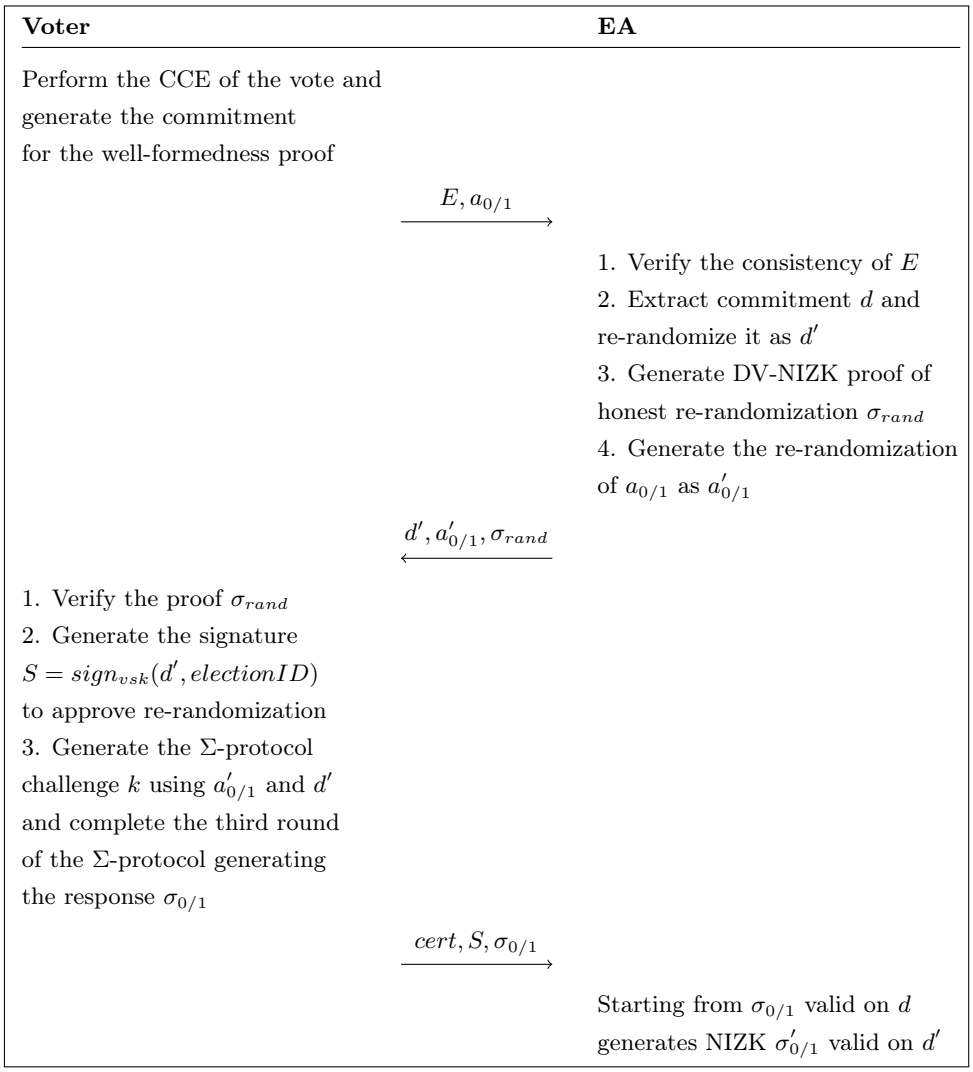

Figure 1: High-level function of the vote casting phase.

- Verify the signature S using the public key in *cert*.
- Check the well-formedness proof $\sigma'_{0/1}$.

If all the checks succeed, it publishes $(cert, S, d', \sigma'_{0/1})$.
The expanded sub-protocol to compute $\sigma'_{0/1}$ and the computation of the proof of honest re-randomization σ_{rand} can be found in Appendix C.

4.2 Tally and Verification

Tally. Note that, in order to prove the correctness of the tally, the EA should keep, for each submitted vote, a randomization factor as $b = g_1^s$ where s is the random coins used to

re-randomize the corresponding commitment. When the election ends, the EA performs the following operations:

- Multiply element-wise all the CCE ballots (excluding the consistency proofs), obtaining one single CCE ballot E^{result} encrypting the election result.
- Multiply the randomization factors b related at each submitted vote obtaining $b^{combined}$.
- Extract the commitment opening from E^{result} as $o^{combined}$.
- Decrypt the election results as $result$.
- Multiply $b^{combined}$ and $o^{combined}$ to obtain the opening to the election result as o^{result}.
- Submit $result$ and o^{result} to the Bulletin Board.

Individual Verification. To verify that his vote will be part of the final tally, the voter checks if a record $(S, cert, d', \sigma_{0/1})$ corresponding to his public signing key is on the Bulletin Board. Furthermore, the digital signature S certifies that the vote was not altered.

Universal Verification. Any observer, using the audit data present on the Bulletin Board, can verify the validity of the election as follows:

- Check that each vote was submitted by an eligible voter verifying each certificate $cert$ and signature S.
- Check the well-formedness proofs of all votes.
- Multiply all the commitments d' obtaining one single commitment d'^{result}.
- Check that $result$ and o^{result} provided by the Election Authority open the computed commitment d'^{result}.

Remark. Since the re-randomization is computed by a single trustee, he could make the universal verification fail by not providing the correct re-randomization factors, putting the EA in a very difficult situation. A simple solution to overcome this problem could be involving more trustees during the vote casting phase. In addition to the shown protocol, the voter sends the original commitment d to all the trustees. After having performed the re-randomization, the re-randomizing trustee sends the re-randomization exponent s and the re-randomized commitment d' to the other trustee who check that d' is a re-randomization of d and subsequently digitally sign it. A vote is admitted on the bulletin board only if it is signed by the majority of the trustees and by the correspondent voter. All this additional communication is performed via authenticated confidential channels.

5 Acknowledgments

Research supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 780477 (project PRIViLEDGE).

References

[1] European election results 2019. European Parliament, 2019. https://europarl.europa.eu/election-results-2019/en/turnout/.

[2] Go programming language. Go programming language, 2019. https://golang.org/.

[3] M. Arapinis, V. Cortier, S. Kremer, and M. Ryan. Practical everlasting privacy. In D. Basin and J. C. Mitchell, editors, *Principles of Security and Trust*, pages 21–40, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[4] J. C. Benaloh and D. Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *26th ACM STOC*, pages 544–553, Montréal, Québec, Canada, May 23–25, 1994. ACM Press.

[5] D. Bernhard, V. Cortier, D. Galindo, O. Pereira, and B. Warinschi. Sok: A comprehensive analysis of game-based ballot privacy definitions. In *2015 IEEE Symposium on Security and Privacy*, pages 499–516, May 2015.

[6] D. Bernhard, O. Pereira, and B. Warinschi. How not to prove yourself: Pitfalls of the Fiat-Shamir heuristic and applications to Helios. In X. Wang and K. Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 626–643, Beijing, China, Dec. 2–6, 2012. Springer, Heidelberg, Germany.

[7] P. Chaidos, V. Cortier, G. Fuchsbauer, and D. Galindo. BeleniosRF: A non-interactive receipt-free electronic voting scheme. In E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, editors, *ACM CCS 2016*, pages 1614–1625, Vienna, Austria, Oct. 24–28, 2016. ACM Press.

[8] V. Cortier, D. Galindo, S. Glondu, and M. Izabachène. Election verifiability for helios under weaker trust assumptions. In M. Kutylowski and J. Vaidya, editors, *ESORICS 2014, Part II*, volume 8713 of *LNCS*, pages 327–344, Wroclaw, Poland, Sept. 7–11, 2014. Springer, Heidelberg, Germany.

[9] 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, Santa Barbara, CA, USA, Aug. 21–25, 1994. Springer, Heidelberg, Germany.

[10] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In W. Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 103–118, Konstanz, Germany, May 11–15, 1997. Springer, Heidelberg, Germany.

[11] C. Culnane and S. Schneider. A peered bulletin board for robust use in verifiable voting systems. In *Proceedings of the 2014 IEEE 27th Computer Security Foundations Symposium*, CSF '14, pages 169–183, Washington, DC, USA, 2014. IEEE Computer Society.

[12] E. Cuvelier, O. Pereira, and T. Peters. Election verifiability or ballot privacy: Do we need to choose? In J. Crampton, S. Jajodia, and K. Mayes, editors, *ESORICS 2013*, volume 8134 of *LNCS*, pages 481–498, Egham, UK, Sept. 9–13, 2013. Springer, Heidelberg, Germany.

[13] D. Demirel, J. Van De Graaf, and R. Araújo. Improving helios with everlasting privacy towards the public. In *Proceedings of the 2012 International Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*, EVT/WOTE'12, pages 8–8, Berkeley, CA, USA, 2012. USENIX Association.

[14] R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: The second-generation onion router. In M. Blaze, editor, *USENIX Security 2004*, pages 303–320, San Diego, CA, USA, Aug. 9–13, 2004. USENIX Association.

[15] A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In J. Seberry and Y. Zheng, editors, *AUSCRYPT'92*, volume 718 of *LNCS*, pages 244–251, Gold Coast, Queensland, Australia, Dec. 13–16, 1993. Springer, Heidelberg, Germany.

[16] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure distributed key generation for discrete-log based cryptosystems. *Journal of Cryptology*, 20(1):51–83, Jan. 2007.

[17] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304, Providence, RI, USA, May 6–8, 1985. ACM Press.

[18] P. Grontas, A. Pagourtzis, and A. Zacharakis. Security models for everlasting privacy. Cryptology ePrint Archive, Report 2019/1193, 2019. https://eprint.iacr.org/2019/1193.

[19] P. Grontas, A. Pagourtzis, A. Zacharakis, and B. Zhang. Towards everlasting privacy and efficient coercion resistance in remote electronic voting. In A. Zohar, I. Eyal, V. Teague, J. Clark, A. Brac-

ciali, F. Pintore, and M. Sala, editors, *Financial Cryptography and Data Security*, pages 210–231, Berlin, Heidelberg, 2019. Springer Berlin Heidelberg.

[20] S. Heiberg, I. Kubjas, J. Siim, and J. Willemson. On trade-offs of applying block chains for electronic voting bulletin boards. Cryptology ePrint Archive, Report 2018/685, 2018. https://eprint.iacr.org/2018/685.

[21] M. Hirt and K. Sako. Efficient receipt-free voting based on homomorphic encryption. In B. Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 539–556, Bruges, Belgium, May 14–18, 2000. Springer, Heidelberg, Germany.

[22] A. Kiayias, A. Kuldmaa, H. Lipmaa, J. Siim, and T. Zacharias. On the security properties of e-voting bulletin boards. In D. Catalano and R. De Prisco, editors, *Security and Cryptography for Networks*, pages 505–523, Cham, 2018. Springer International Publishing.

[23] O. Kulyk, V. Teague, and M. Volkamer. Extending helios towards private eligibility verifiability. In R. Haenni, R. E. Koenig, and D. Wikström, editors, *E-Voting and Identity*, pages 57–73, Cham, 2015. Springer International Publishing.

[24] P. Locher and R. Haenni. Receipt-free remote electronic elections with everlasting privacy. *Annals of Telecommunications*, 71(7):323–336, Aug 2016.

[25] A. Menezes, P. Sarkar, and S. Singh. Challenges with assessing the impact of nfs advances on the security of pairing-based cryptography. In R. C.-W. Phan and M. Yung, editors, *Paradigms in Cryptology – Mycrypt 2016. Malicious and Exploratory Cryptology*, pages 83–108, Cham, 2017. Springer International Publishing.

[26] T. Moran and M. Naor. Receipt-free universally-verifiable voting with everlasting privacy. In C. Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 373–392, Santa Barbara, CA, USA, Aug. 20–24, 2006. Springer, Heidelberg, Germany.

[27] T. Moran and M. Naor. Split-ballot voting: everlasting privacy with distributed trust. In P. Ning, S. De Capitani di Vimercati, and P. F. Syverson, editors, *ACM CCS 2007*, pages 246–255, Alexandria, Virginia, USA, Oct. 28–31, 2007. ACM Press.

[28] S. Park, M. A. Specter, N. Narula, and R. L. Rivest. Going from bad to worse: from internet voting to blockchain voting. *J. Cybersecur.*, 7(1), 2021.

[29] O. Pereira. Verifiable elections with commitment consistent encryption – a primer, 2014.

[30] K. Sako and J. Kilian. Receipt-free mix-type voting scheme - a practical solution to the implementation of a voting booth. In L. C. Guillou and J.-J. Quisquater, editors, *EUROCRYPT'95*, volume 921 of *LNCS*, pages 393–403, Saint-Malo, France, May 21–25, 1995. Springer, Heidelberg, Germany.

[31] C.-P. Schnorr. Efficient identification and signatures for smart cards. In G. Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252, Santa Barbara, CA, USA, Aug. 20–24, 1990. Springer, Heidelberg, Germany.

[32] M. Scott. The apache milagro crypto library (version2.0). MIRACL, 2019. https://miracl.com/assets/pdf-downloads/amcl.pdf.

A Related Work

The term everlasting privacy was introduced by Moran and Naor [26, 27] in a setting in which ballots are cast in a private voting booth. A previous line of work, based on blind signatures [15], can provide everlasting privacy assuming the existence of an anonymous channel which is used to cast the signed unblinded ballots. However, in such protocols, ballot stuffing by authorities cannot be detected. More recently Demirel et al. proposed an approach to achieve everlasting privacy in remote electronic election combining Pedersen commitments and Paillier encryption [13], however, as they acknowledge, their solution is not practical in terms of efficiency, relying on cut-and-choose ZK proofs. The first practical solution providing everlasting privacy while

being based on the tallying of threshold encrypted ballot has been proposed by Pereira et al. [12], introducing a new primitive called Commitment Consistent Encryption (CCE), proposing both a voting protocol based on homomorphic tallying and one based on mixnets. In this construction ballots are cast through a private channel between the voter and the Election Authority, who eventually extracts from the ballot the audit data that can be posted on the bulletin board. In other words, everlasting privacy is guaranteed only with respect to the data contained in the bulletin board. It is assumed that an adversary would not be able to intercept and store all the communications between the voters and the EA at the moment they take place. Furthermore, all the private data of the EA should be destroyed right after the end of the election event. This flavor of everlasting privacy is named *practical everlasting privacy* [3]. The first practical solution providing everlasting privacy while being based on the tallying of threshold encrypted ballot has been proposed by Pereira et al. [12], introducing a new primitive called Commitment Consistent Encryption (CCE), proposing both a voting protocol based on homomorphic tallying and one based on mix-nets. In this construction ballots are cast through a private channel between the voter and the Election Authority, who eventually extracts from the ballot the audit data that can be posted on the bulletin board. In other words, everlasting privacy is guaranteed only with respect to the data contained in the bulletin board. It is assumed that an adversary would not be able to intercept and store all the communications between the voters and the EA at the moment they take place. Furthermore, all the private data of the EA should be destroyed right after the end of the election event. This flavor of everlasting privacy is named *practical everlasting privacy* [3]. Benaloh and Tuinstra [4] were the first to define receipt-freeness giving a rather informal description. After this, several proposals based on untappable channels have been made [30, 21]. Kulyk et al. [23] propose an extension of Helios using proxies that cast null votes on behalf of other voters. Real votes are hidden in a crowd of null votes that are cast by others but are indistinguishable from those of the eligible voters. This system is difficult to deploy in practice and strongly relies on revoting, which is currently not allowed in most countries. Chaidos et al. [7] propose BelenionsRF, a protocol providing both receipt-freeness and end-to-end verifiability using signatures on re-randomizable ciphertext and Groth-Sahai proofs. Furthermore they provide the first game-based definition of receipt-freeness. Although there is a considerable amount of proposals achieving everlasting privacy or receipt-freeness, to the best of our knowledge there are only two proposals achieving both. A proposal by Locher and Haenni essentially achieves receipt-freeness using the idea of [23]. Another proposal by Grontas et al. [19] achieves a stronger property than receipt-freeness which is called *coercion-resistance*: an honest voter can cast his desired vote even if he is, for some time, under the full control of the coercer. However, this property requires the voter to follow a strategy, voting with fake credentials when under coercion in order to submit a null vote. The first proposal [24] achieves everlasting privacy using perfectly hiding commitments while [19] uses a primitive named *Publicly Auditable Conditional Blind Signature*. Both proposals achieve perfect everlasting (i.e. even against the EA) privacy under the assumption that the ballots are cast through an anonymous channel.

B Preliminaries, Assumptions and Primitives

We work with asymmetric bilinear groups and assume the existence of a group generator `GrpGen`, which on input 1^λ outputs $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, h_1, h_2, e, p)$, where p is a prime of length λ and $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are cyclic groups of order p, g_1 is a generator of G_1, h_1 and h_2 are generators of G_2 and e is bilinear map $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ such that $e(g_1, h_1)$ generates $\mathbb{G}_T$. Furthermore, e is a so called type-3 pairing, meaning that $\mathbb{G}_1 \neq \mathbb{G}_2$ and it does not exist an efficient isomorphism

$\phi(\mathbb{G}_1) \to \mathbb{G}_2$. Elliptic-Curve Cryptography (ECC) provides the same security levels of non-EC constructions with much smaller keys. Furthermore, the usage of specific elliptic curves to efficiently implement the CCE is crucial. In order to have a full EC-based protocol, ECDSA is used for digital signatures.

B.1 Assumptions and Primitives

Let be λ a security parameter, g be a generator of a cyclic group $\mathbb{G}$ of prime order p of length λ. The Decisional Diffie-Hellman assumption (DDH) holds for group $\mathbb{G}$ if the tuples $(g^a, g^b, g^{ab}), (g^a, g^b, g^c)$ with $a, b, c \leftarrow_\$ \mathbb{Z}_p$ can be distinguished by a PPT adversary only with advantage negligible in λ. We assume the existence of a hash function H that acts as a Random Oracle. This assumption is required because FS transform is used to build NIZK proofs.

Commitment Consistent Encryption CCE. In the following, we describe the Commitment Consistent Encryption scheme [12]. Given a bilinear group $\mathcal{G}=(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, h_1, h_2, e, p)$ we define the following algorithms:

1. `KeyGen(`$\mathcal{G}$`)`: Generate an ElGamal encryption key $g_2 = g_1^x$, with $x \leftarrow_\$ \mathbb{Z}_p$. Return ($\mathsf{pk} = g_2, \mathsf{sk} = x$).
2. `Enc(v,pk)`: Return $E = (c_1, c_2, d, \sigma_{cc}) = (g_1^s, g_1^r g_2^s, h_1^r h_2^v, \sigma_{cc})$ with $(r, s) \leftarrow_\$ \mathbb{Z}_p$ and σ_{cc} as consistency proof.
3. `Dec(E,sk)`: Extract the discrete logarithm of $e(c_1^x/c_2, h_1)e(g_1, d)$ in basis $e(g_1, h_2)$.
4. `ExtractComm(E)`: Output d.
5. `ExtractOpening(E,sk)`: Return $a = c_2/c_1^x$.
6. `VerifyOpening(d,a,v)`: Check if $e(a, h_1) = e(g_1, d/h_2^v)$.

The use of pairings allows to open a commitment using the group element a instead of the scalar r. Extracting r from its lifted ElGamal encryption would require exponential time, which is not acceptable given the large size of r. However, this can be done rather efficiently for v, whose value lies in a relatively small range (i.e. bounded by the number of voters). In the following, we describe how the consistency proof σ_{cc} for the triple $(c_1, c_2, d) = (g_1^s, g_1^r g_2^s, h_1^r h_2^v)$ can be computed and verified.

Computation:

1. *Commitment:* compute commitment $c' = (c'_1, c'_2, d') = (g_1^{s'}, g_1^{r'} g_2^{s'}, h_1^{r'} h_2^{v'})$ with $r', s', v' \leftarrow_\$ \mathbb{Z}_p$.
2. *Challenge:* compute challenge $e = \mathsf{H}(c, c', pv)$.
3. *Response:* compute $f_r = r' + er$, $f_s = s' + es$, $f_v = v' + ev$.

The proof is defined as $\sigma_{cc} = (e, f_r, f_s, f_v)$.

Verification:

1. Compute $c' = (c'_1, c'_2, d')$ with $c'_1 = g_1^{f_s}/c_1^e, c'_2 = g_1^{f_r} g_2^{f_s}/c_2^e, c'_3 = h_1^{f_r} h_2^{f_v}/d^e$.
2. Check that $e = \mathsf{H}(c, c', pv)$.

We require the DDH to hold in $\mathbb{G}_1$ since (c_1, c_2) is the ElGamal Encryption of the commitment opening, and we require DDH to hold in $\mathbb{G}_2$ so that the commitment d is computationally binding, note that d is perfectly hiding.

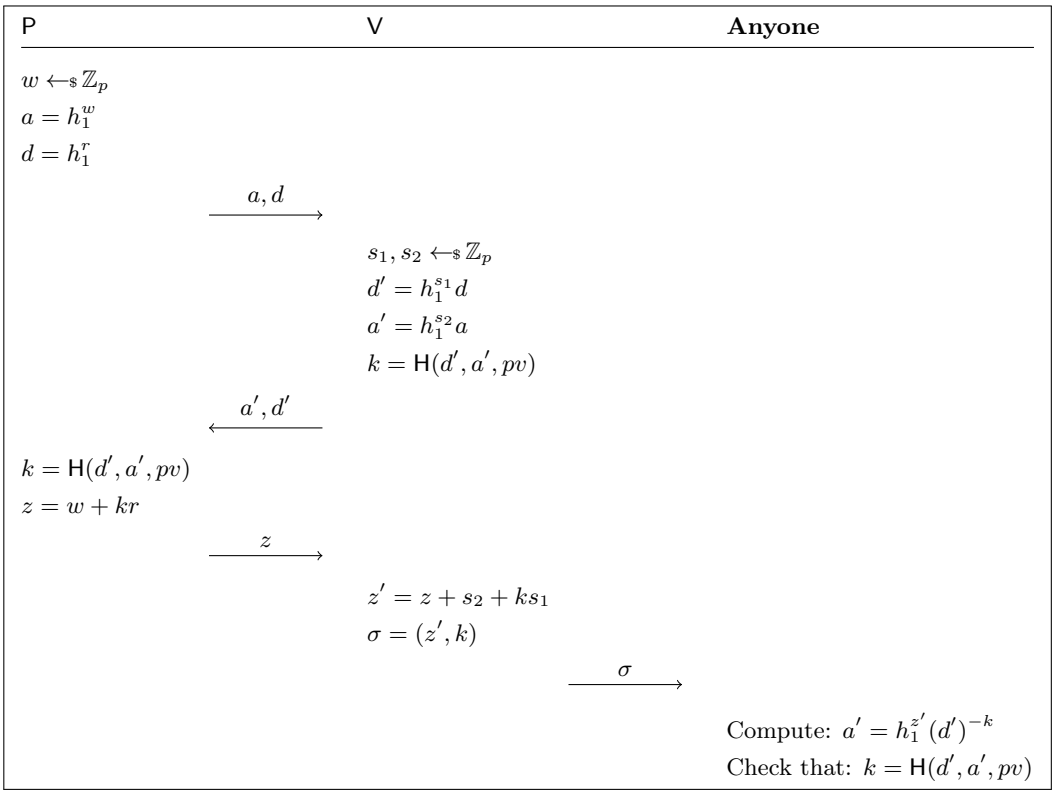

Figure 2: Joint Computation of NIZK DL PoK. P and V jointly generate a NIZK PoK of the DL of $d' = h_1^{r+s}$, with r and s respectively known by P and V. However, at the end of the protocol, none of the two knows the randomness controlled by the other party.

C Subprotocols

Joint computation of NIZK DL PoK. We propose a protocol allowing two parties, a prover P and a verifier V the first knowing r and the latter knowing s, scalars in $\mathbb{Z}_p$, to compute a NIZK proof of the knowledge of the discrete logarithm of $d' = h_1^{r+s}$ with h_1 generator of a group $\mathbb{G}$ of prime order p. The proof is computed using the Schnorr protocol [31], in which P wants to prove to V the knowledge of the discrete logarithm of $d = h_1^r$. However, in the meantime, V performs different operations with respect to the regular Schnorr protocol, so that he will eventually be able to build a NIZK proof of the knowledge of the discrete logarithm of $d' = h_1^s d = h_1^{r+s}$. The protocol to compute such proof is shown in figure 2. One can decompose the protocol in two parts: one is the regular Schnorr protocol, and another one which is another Schnorr protocol with the FS transform. Thus, V plays two roles at the same time: she is a verifier in the regular Schnorr protocol, in which P is proving the knowledge of DL of $d = h_1^r$; she is a prover in the Schnorr protocol with FS transform. He combines information he knows with data provided by P to prove the knowledge of $d' = h_1^{r+s_1}$. The junction point between the two protocols is given by the algebraic relation between the commitments d and d', and by the shared challenge k. This enables V to combine the transcripts of the two protocols so that the overall result for an external observer is exactly the same as if one single prover had carried

out a regular Schnorr protocol with the FS transform to prove the knowledge of the DL of d'. Note that P receives no information about the re-randomization value s_1, as well as V does not gain any knowledge about the concrete value of r. In fact, since we are operating under the ROM assumption, V is a honest verifier in the sense that the challenge k is genuinely random and independent from the commitment a. To convince P of this fact, V sends the randomized commitment a' along with d', instead of the challenge itself, letting P compute it using the hash function. Furthermore, a' and d' are perfectly hiding commitments, so they do not leak any information about the respective re-randomization factors.

Well-formedness: Proving that a Commitment is to either 0 or 1. Given $d = h_1^r, h_2^v$ and $d' = h_1^s d = h_1^{r+s} h_2^v$ the proof is a disjunctive ZK PoK of the discrete logarithm of d' or d'/h_2. The proof is carried out embedding an OR composition of Σ-protocols, as described in [9], in the base protocol proposed in Appendix C. The detailed functioning of the protocol is shown below, where the symbol $\oplus$ represents the bit-wise XOR operation.

- **Voter**: If $v = 1$ compute: $(z_1, k_1, w_2) \leftarrow_\$ \mathbb{Z}_p^3$, $a_{0/1}^1 = h_1^{z_1} d^{-k_1}, a_{0/1}^2 = h_1^{w_2}$. Else compute: $(z_2, k_2, w_1) \leftarrow_\$ \mathbb{Z}_p^3$, $a_{0/1}^1 = h_1^{w_1}$, $a_{0/1}^2 = h_1^{z_2}(d/h_2)^{-k_2}$. Send $(a_{0/1}^1, a_{0/1}^2)$ to the EA.
- **EA**: Compute $(s, s_1, s_2) \leftarrow_\$ \mathbb{Z}_p^3$, $d' = h_1^s d$, $a_{0/1}^{1'} = h_1^{s_1} a_{0/1}^1$, $a_{0/1}^{2'} = h_1^{s_2} a_{0/1}^2$. Send $(d', a_{0/1}^{1'}, a_{0/1}^{2'})$ to the Voter.
- **Voter**: Compute $k = \mathsf{H}(d', a_{0/1}^{1'}, a_{0/1}^{2'}, pv)$. If $v = 1$, $k_2 = k \oplus k_1, z_2 = w_2 + r k_2$, otherwise, $k_1 = k \oplus k_2$, $z_1 = w_1 + r k_1$. Send $\sigma_{0/1} = (k_1, k_2, z_1, z_2)$ to the EA.
- **EA**: Construct $\sigma'_{0/1} = (k_1, k_2, z'_1, z'_2)$ with $z'_1 = z_1 + s_1 + s k_1$ and $z'_2 = z_2 + s_2 + s k_2$.

Given d' and the proof $\sigma'_{0/1} = (k_1, k_2, z'_1, z'_2)$ can be verified as follows:

- Compute $a_{0/1}^{1'} = h_1^{z'_1} d'^{-k_1}$ and $a_{0/1}^{2'} = h_1^{z'_2}(d'/h_2)^{-k_2}$.
- Compute $k = \mathsf{H}(d', a_{0/1}^{1'}, a_{0/1}^{2'}, pv)$ and check that $k = k_1 \oplus k_2$.

DV-NIZK Proof of Honest Re-randomization. The EA can prove to the voter that commitment d' is an honest re-randomization of d, i.e. $d' = h_1^s d$, by computing a disjunctive proof of knowledge of either the secret signing key of the voter vsk or of the process to honestly re-randomize d into d'. This is achieved proving the knowledge of the discrete logarithm of $vpk = g^{vsk}$ OR of d'/d, where g is the generator of the group, of a certain prime order q, used by the ECDSA signature scheme. Since the secret key is not known by the EA, the Voter is sure that, if the proof verifies, only the other branch of the OR can be true. And in turn, the truth of this statement guarantees the correctness of the re-randomization process: in fact, if the re-randomization is performed honestly as $d' = h_1^s d$, the EA must know the DL of d'/d. This proof can only convince the voter but not the coercer, since the voter could use the knowledge of his secret key to compute verifying proofs for every arbitrary vote. The proof is computed by the EA using the OR Composition of two Σ-protocols, plus the FS transform. The branch related to vsk is of course simulated, since the EA does not know its value. The proof σ_{rand} is computed as described below, where, without loss of generality, we assume $p \leq q$.

- *Commitment:* with $l_1, w, z_1 \leftarrow_\$ \mathbb{Z}_p$, compute $a_1 = g^{z_1} vpk^{-l_1}, a_2 = h_1^w$.
- *Challenge:* compute $l = \mathsf{H}(d, d', vpk, a_1, a_2, pv)$, $l_2 = l \oplus l_1$.
- *Response:* compute $z_2 = w + s l_2$.

The proof is defined as $\sigma_{rand} = (l_1, l_2, z_1, z_2)$. For d and d' the proof is verified as follows:

- Compute $a_1 = g^{z_1} vpk^{-l_1}, a_2 = h^{z_2}(d'/d)^{-l_2}$.
- Compute $l = \mathsf{H}(d, d', vpk, a_1, a_2, pv)$ and verify that $l = l_1 \oplus l_2$.

D Multiple Candidates

We described the voting scheme for an election with only two candidates but it can be easily generalized to n-candidate elections applying the following modifications:

- Instead of a single ciphertext, the voter submits a vector containing n CCE ciphertexts, one for each candidate. To vote the i-th candidate he encrypts 1 in the i-th position of the vector and 0 in all the others.
- The voter proves that each element of the vector contains either 0 or 1. In addition, to ensure that he votes for $k < n$ candidates, where typically $k = 1$, he has to prove the knowledge of the DL of $\prod_{i=1}^{n} d_i h_2^{-k}$. Therefore, for each vote, $n + 1$ protocols using the construction of Appendix C are run in parallel.
  The voter still generates a single digital signature that will be used to approve n re-randomized commitments. These commitments will be posted by the EA on the bulletin board, along with the just generated $n + 1$ NIZK proofs.
- When tallying, the EA has to multiply all the CCE vectors, without the consistency proofs, element-wise. The only practical difference is that now the EA handles n CCE ciphertexts per-voter instead than one. Thus, the EA obtains a single CCE vector from which extracts the election results (i.e., the votes for each candidate, and the relative commitment openings). The verification procedure is the same as showed before, except the fact that it is carried out element-wise.

E Discussion on the Security of the Scheme

Here, we informally discuss the security of the voting scheme with respect to the properties we defined in section 3. We stress that this is just a preliminary high-level analysis and more effort is required to fully claim that those properties are satisfied.

- *Ballot secrecy*: It is satisfied under the assumption that votes are never decrypted in isolation, but only after the homomorphic addition, this translates into trusting the majority of the trustees of not cooperating to break ballot secrecy.
- *Eligibility*: It is satisfied since only an eligible voter in possession of a valid certificate can generate a signature that is accepted by the Bulletin Board.
- *Eligibility Verification*: Every observer can check that each vote on the Bulletin Board was cast by eligible voter verifying the related certificate and digital signature.
- *Individual Verifiability*: The voter can check that his ballot is recorded checking if his vote is on the bulletin board. The digital signature assures the voter that his vote has not been modified before being posted on the bulletin board.
- *Universal Verifiability*: The validity of individual votes is guaranteed by the well-formedness proofs. In addition, one can convince himself of the validity of the final tally using the verification procedure defined in section 4.2.
- *Practical Everlasting Privacy*: It is satisfied since the part of the audit data related to a vote only includes perfectly hiding commitments and perfect zero-knowledge proofs. The digital signatures and the certificates are instead independent from the voter's choice.

- *Receipt-freeness*: The voter is not able to prove to a coercer how he voted. In fact, he does not know how the EA transformed d into d'. However, this re-randomization process does not require trusting the EA since the DV-NIZK proof σ_{rand} acts as a personal receipt convincing the voter. However, this receipt is not convincing at all for a coercer since the voter could generate such proofs for any arbitrary commitment vector.