Implementation of an E-Voting Scheme Using Hyperledger Fabric Permissioned Blockchain

Denis Kirillov(✉), Vladimir Korkhov, Vadim Petrunin, Mikhail Makarov, Ildar M. Khamitov, and Victor Dostov

Saint-Petersburg State University, St. Petersburg, Russia
kirillovdenand@gmail.com, {v.korkhov,v.petrunin}@spbu.ru, makarovmma@gmail.com, CryptoVoter@gmail.com, greygato@gmail.com

Abstract. Since the issue of using e-voting in both corporate and government voting has not yet been fully resolved, there remains a wide scope for improving existing approaches and proposing new protocols enabling the voting system to be resistant to various kinds of attacks. Due to the rapid development of distributed ledger technologies and their potential for solving existing problems we propose a modified protocol of the published earlier voting scheme which is complemented by blockchain technology to increase trust between participants. This approach allows carrying out combined voting of both traditional paper voting and e-voting. In this paper we describe the architecture of our solution, discuss its implementation based on Hyperledger Fabric platform and demonstrate its functionality.

Keywords: E-voting · Blockchain · Hyperledger fabric · Distributed ledger technologies

1 Introduction

In many countries paper ballots are still used for elections. Nevertheless, technologies are developing rapidly and it has already become commonplace not to use cash and not to deal with paper media anywhere. Paper ballots are used to increase the level of reliability of the voting system since the stakes are high. At the same time attempts to implement existing e-voting protocols in some countries (Estonia [1], Australia [2]) showed that not all security problems were solved [3,4] thereby the skepticism of state agencies towards new solutions increased. An ideal system should satisfy the following principles [5]:

– Eligibility,
– Un-reusability,
– Un-traceability,
– Verifiability,
– Receipt-freeness.

S. Misra et al. (Eds.): ICCSA 2019, LNCS 11620, pp. 509–521, 2019.
https://doi.org/10.1007/978-3-030-24296-1_40

In traditional approaches not all of these properties are held. However, the number of covered ones can be increased by transferring voting to a digital space building more convenience for people. It can also enhance the usage by young people. What is more important, it significantly reduces the election costs and decreases inspectors' efforts.

Voting, in particular elections, is a procedure whose members don't trust each other. Thus, the basic postulate is that the system to be attacked not only by an outside entity but also by the system participants themselves – both the voters and the organizers (juggling the results, disrupting the process itself, etc.). In such circumstances approaches based only on cryptographic technology may not always demonstrate an acceptable result. Some additional tools are required. Recently, distributed ledger technologies have been rapidly developing. These technologies were created specifically to make it possible to come to an agreement even in a non-trusted environment. In this paper, we extend one of the existing approaches to conduct electronic voting with the opportunities provided by the blockchain. The basic concepts of distributed ledger technologies are as follows:

- A copy of the ledger is on each node,
- No single point of failure,
- Changing the ledger only if most nodes agree,
- Possibility to reach an agreement even if some nodes are disconnected during network operation.

Blockchain platforms are usually divided into permissionless (anyone can become a member of the network) and permissioned (only known members are allowed to interact with the network). The first group includes, for example, Ethereum and Bitcoin, and the second one—Hyperledger Fabric, Exonum. Blockchains of the second type are suitable for our purposes, since voting participants usually incorporate organizers, voters, and observers who are known in advance and have different rights. Thereby, Hyperledger Fabric platform has been chosen for our project.

The rest of this paper is organized as follows. Section 2 describes existing approaches to e-voting. The review of the used technologies is presented in Sect. 3. The main description of the protocol is presented in Sect. 4. Security and privacy analysis is conducted in Sect. 5. The final Sect. 6 summarizes obtained results, points out directions of future work and concludes the paper.

2 Related Work

The problem of electronic voting is quite complicated due to a large number of requirements. Therefore there are many works devoted to this topic which propose different approaches in varying degrees covering the necessary restrictions. Completely different cryptographic techniques are used in different approaches: blind signature, ring signature, zero-knowledge proof, homomorphic encryption, Paillier cryptosystem etc. In this section for the sake of brevity we designate the body that organizes the voting procedure as Certificate Authority (CA).

The paper [6] proposes using CA's blind signature over the public key and digital commitment (it allows one to commit to a chosen value while keeping it hidden to others, with the ability to reveal the committed value later; at the same time a party cannot change the value after they have committed to it) of a voice to gain the right to vote. After that a ballot (public key, digital commitment, CA signature) is broadcasted to all blockchain nodes. Each node verifies the signature, records the given vote to the ledger and the voter receives the id of his vote. If any changes need to be made, voter forms a new ballot containing the vote id which needs to be replaced and data similar to the previous one. After the voting phase is over the voter sends the data to open a digital commitment of his vote, a choice id, and a CA signature over them. Then votes are counted and the results are recorded to the ledger.

Since the blockchain technology is used, every voter at any step can verify that his vote is taken into account, as it is present in the network. A certain degree of un-traceability is achieved by using blind-signed public keys. A vote will not be counted more than once as there is only one choice not replaced by another one belonging to the same public key at each instant of time in the ledger. Receipt-freeness is missing.

A slightly different method is proposed in paper [7]. Voters must have an ID and PIN to enter the system that creates a "wallet" for each user (using NIZKP is possible) whereby it becomes possible to interact with a smart contract which records all voices to the blockchain network. In order to cast a vote, the voter has to call the function of smart contracts (e.g. corresponding to the electoral district) and pass the candidate's number as an argument. The id of the transaction which contains the choice is returned as a result. Smart contracts throughout the elections keep the current progress of the elections (that is, there is a variable that contains the number of votes for a candidate).

In order to achieve un-traceability, a special transaction structure with no "sender" field is used. A voter cannot change his vote. In order to check that the vote is counted elector has to come to the CA and reveal his identity. Moreover, even in this case there is no guarantee that votes are counted correctly.

In the article [8] the e-voting system is implemented on the Ethereum platform. Voting organizer merely creates the smart contract for election and adds addresses of the wallets which are allowed to vote. After that, voters call the function of this smart contract, passing the number of the candidate as a parameter (you cannot change your decision). At the end of the election process, the organizer calls the function that returns the number of the winning candidate number.

This approach has several disadvantages. For example, the following requirements are not met: un-traceability, receipt-freeness. This is due to the fact that the authors do not assume any cryptographic techniques and the network structure of the Ethereum blockchain is transparent and any participant can see the contents of transactions. In addition, the voter must pay a commission for each call to a smart contract. Also, the proposed approach does not scale well due to the small number of transactions processed per minute.

Liu and Wang [9] propose to introduce a certain number of inspectors who take part in the voting phase. Initially, the voter generates a pair of keys one of which (public) sends to the CA along with the data for authentication. The CA checks the voter and publishes a list of registered public keys whose owners are allowed to vote. Then voters form ballots and send them to the CA to blindly sign using a transaction which contains their public key as a value of the sender field. The CA verifies that the voter has the right to vote and has not yet cast a vote and returns the signed data. The voter sends it to all inspectors for signature. After successfully completing this stage the vote is signed by all the controlling participants. The voter generates a new key pair and sends the signed voice to the CA indicating the new public key as the value of the sender field in the transaction. After the end of the voting, CA publishes the results. This approach generally satisfies all restrictions except receipt-freeness.

The approach proposed in the article [10] uses the Paillier cryptosystem and the short linked ring signature (SLRS) as cryptographic techniques. The voting protocol is the following. The smart contract is initialized, and then parameters of Paillier cryptosystem and ring signatures are loaded to the blockchain. After that, admin data is used to enter the system and download the ring signature parameters and the public key of Paillier cryptosystem. To complete registration in the system user generates keys with SLRS and sends the public key to the smart contract. During the voting phase the voter forms the voice and encrypts it using a cryptosystem. Then he calls a function that provides data to confirm that the encrypted voice contains the number of the existing candidate. This data along with the choice is sent to the smart contract. After validation the smart contract adds an encrypted zero to the vote (to ensure receipt-freeness), signs and returns it to the voter. The voter checks the signature on the received message, imposes his ring signature and sends it back to the smart contract. The smart contract verifies the voter has not yet voted and then reports the selected option to the blockchain. When the counting phase comes, the smart contract signs the amount of the encrypted votes and sends it to the administrator who decrypts the amount using the cryptosystem private key and generates data used to verify that the received amount is correct; this information is sent back to the smart contract which calculates the number of votes scored by each candidate and reports these results to the blockchain.

This approach satisfies all the requirements for a voting system. However, receipt-freeness can be bypassed by collecting the private keys of all users from one pool (a pool of encrypted zeros that are added to the encrypted voter's voice to bring some randomness). Increasing the pool size can become a solution, which makes the attack more complicated.

3 Used Technologies

This section provides brief information about the technologies used in the proposed solution.

Blind Signature. This technology is a digital signature mechanism with the additional property that doesn't allow the one who is providing the signature to check what exactly should be signed. This concept was proposed in [11,12]. In our project we use an implementation based on RSA [13].

Secret Sharing Scheme. This approach allows distributing some secret information among several participants so that it can only be recovered by obtaining separate parts of each secretary. Many schemes are a special case of more general theoretical approach [14].

Identity Mixer (Idemix). This technology was developed by IBM and based on cryptographic techniques the development of which was presented at top conferences and carefully checked by the community. The practical implementation is based on the scheme described in [15–17]. Brief description of this technology is provided below.

Suppose there are three parties: the user, the issuer (producing a certificate and confirming the user's identity) and the verifier who wants to make sure that the user gets all attributes that are necessary to perform an action. In this scheme the main issue solved by Idemix is providing the verifier not all the information about the user but only certain attributes or even just a proof that the requested attribute falls within a certain range. In addition, it is possible to provide unlinkability which makes associating two different requests with the same user impossible. Thus Idemix allows users to ensure anonymity and privacy.

Hyperledger Fabric. Aforementioned Hyperledger Fabric is the permissioned blockchain being developed by Hyperledger consortium, in particular by IBM. A distinctive feature of Fabric compared to public blockchains is the way of processing transactions in the following order:

$$endorse \Rightarrow order \Rightarrow validate$$

The first stage includes checking transactions for the possibility of execution. Once they were executed without changing the state of the ledger they should be ordered based on a consensus algorithm. If the agreement is reached, transactions are sent to all nodes that update their copy of the ledger. Thus there are three types of nodes: Endorsing Peer which executes transactions (logic is executed on them), Ordering Nodes which produce ranking, and Committing Peer to maintain a copy of the registry, write transactions to it; Endorsing Peer is also a committing peer.

4 Approach and System Architecture

The proposed approach is based on the article of He and Su [18] who described an e-voting scheme without using the distributed ledger technology. Our current research aims to implement last-mentioned approach as a way to automate the electoral process using distributed ledger technologies.

According to the proposed scenario there is a possibility of electronic voting while some electors cast a vote using traditional paper-based ballots.

Firstly, we need to consider the high-level description of the voting scheme. A certain organizer wants to carry out a voting which provides the possibility of electronic voting. A deadline for registration is announced. Up to this point any voter who is allowed to vote can indicate one's willingness to do it electronically. After entering the system with credentials, received from the voting organizer (for example, x.509 certificates) a notification about using the system needs to be done (decision can be changed later in case if registration time is not over). After the registration deadline, the lists of those who decide to vote electronically is transferred to the structures responsible for conducting the voting in the traditional way. It is done in order to avoid the situation when a participant votes both in person and using the electronic system. When the voting period begins the participants who are voting electronically log into the system and make their decision before the voting has finished. There is a potential opportunity to change the decision before the voting deadline. At the end of this procedure, the results are published and everyone can verify those using the system. Consider the structure of the process.

Participants:

1. *Org* – the central voting authority (it is an organization in terms of Hyperledger Fabric, has one or more physical nodes).
2. *Dep* – is a subdivision (e.g. a territorial election commission) that is responsible for conducting voting in its district (for example, it starts the voting process at certain times, this can be useful if there are several time zones involved). It is either a separate organization or a subdivision of the central agency. It has one or several physical nodes.
3. *CC* – chaincode (smart contract) in terms of Hyperledger Fabric which is responsible for the logic of the voting.
4. *V* – voter, system user (member of one of the divisions).
5. *Ins* – an observer (maybe more than one) has one or several nodes and monitors compliance with voting rules.

The Fig. 1 contains the architecture of our solution. First of all participants (organizer, department, inspector) are connected to the same channel. They have got at least one ordering node that is required to reach consensus. However, inspector nodes are "committing peers" (don't execute any logic, get transactions after ordering). By comparison, organizer and department nodes are "endorsing peers" (approve transactions before they will be accepted by other organizations). In addition, each participant has got several client applications that can interact with Fabric network. In general admin apps initialize voting, user apps cast vote and inspect the voting process. It is worthy of note that each of *Dep*, *Org*, *Ins* is several nodes, as well as participant peers, may have got more than two ones.

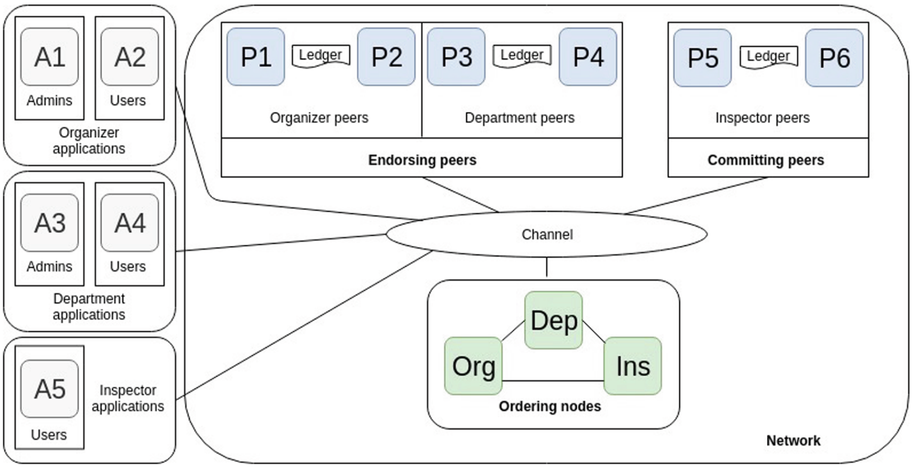

Fig. 1. Architecture of the network.

Now consider the following phases more precisely:

1. Network configuration,
2. Voting configuration,
3. User registration,
4. Voting,
5. Result counting.

4.1 Network Configuration

In the first phase, the main admin determines the rights (read, write) and the number of nodes which every organization has (Fig. 2). Then it extends the network, adds necessary nodes for each organization on which CAs are located, loads logic (chaincode), determines ordering nodes that will be involved in reaching consensus on including transactions into the ledger.

4.2 Voting Configuration

During this phase, the *Org* administrator generates a list of questions, determines the list of *Dep* that can participate in this voting (Fig. 3). This data is loaded into the ledger. Then each of the *Dep* administrators forms an additional data (start/end of voting and registration, list of voters) necessary for holding the voting among voters who belong to this department. This data has to be loaded into the ledger as it is needed for the local voting. Also at this stage, two key pairs are generated $(E_{dep,i}, D_{dep,i})$ for each unit. The public one is recorded to the ledger and the private one is saved in the *private data collection*—a special mechanism in the Hyperledger Fabric, which allows to restrict access to data. After the data is downloaded for each user the list of available polls is updated.

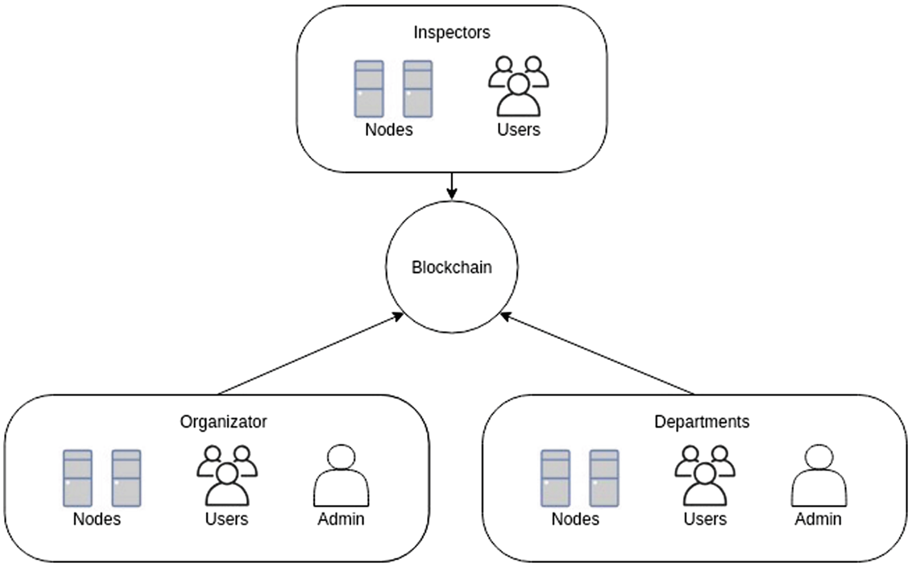

Fig. 2. Participants of the network.

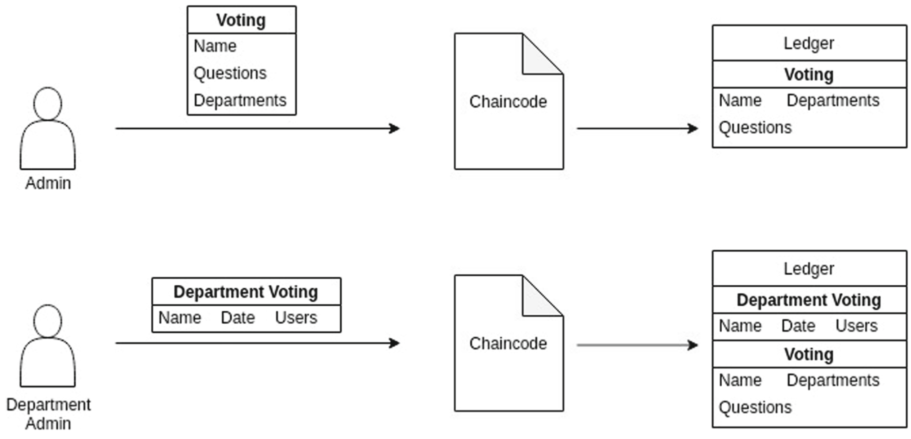

Fig. 3. Initialization voting.

4.3 User Registration

This phase (Fig. 4) is crucial for several purposes:

- To preserve the possibility of a voter to vote using paper-based ballots (if the user has not been registered he is able to vote only in traditional way);
- To ensure the choice privacy and maintain eligibility.

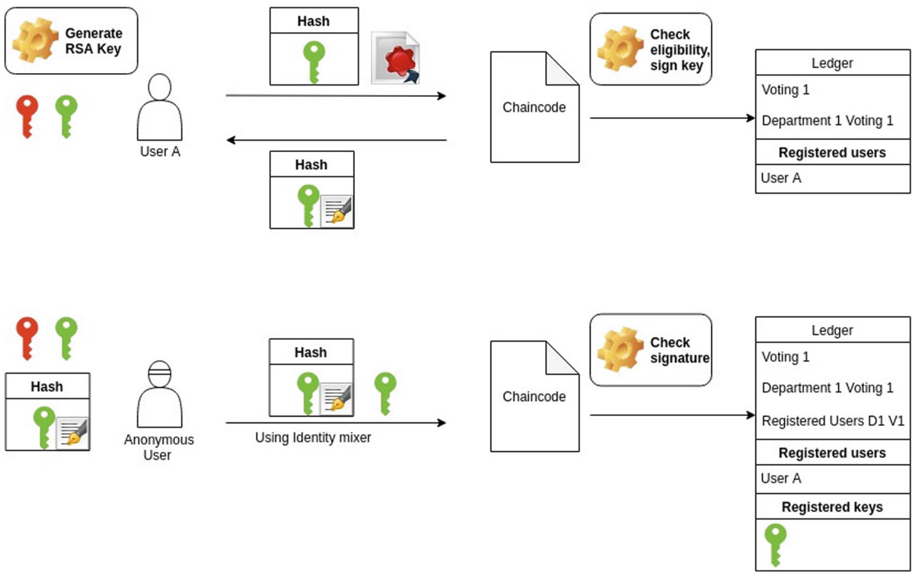

Fig. 4. User registration for electronic voting.

The Sub-step of Obtaining a Blind Signature. During the registration each user generates a pair of keys E_v (public), D_v (private) and a random number R. Then $E_{dep,i}(R) * h(E_v)$ is calculated, here $E_{dep,i}(R)$ is R encrypted with the public key $E_{dep,i}$ of the user department, and $h(E_v)$ is the hash function over the public key of the voter. The result of this expression and the data about elections the voter wants register to are sent to the CC which is in charge of the registration logic. It checks that the current user is allowed to participate in this voting based on data from the CA (user name, department). If it is valid $E_{dep,i}(R) * h(E_v)$ is signed with the private key $D_{dep,i}$ of the department. The received data $D_{dep,i}(E_{dep,i}(R) * h(E_v)) = R * D_{dep,i}(h(E_v))$ (the equality is true since the keys are of the RSA type) is sent back to the user. Then CC records data about that the current user has received a blind signature to the ledger. At the end of the registration phase the list of voters (if it is provided) can be used to avoid revoting. The voter excludes R from $R * D_{dep,i}(h(E_v))$ by multiplying it by R_{inv}, where R_{inv} is the inverse modulo of the department's public key. Then he checks if the hash is signed by its subdivision by comparing $h(E_v)$ and $E_{dep,i}(D_{dep,i}(h(E_v)))$.

The Sub-step of Registering a Public Key that Corresponds to a Person Is Known only by the Voter. At this moment the user has a signed public key $D_{dep,i}(h(E_v))$ which he sends it to the CC along with E_v anonymously using the Identity mixer or idemix (more detail in the next section). There $h(E_v)$ and $E_{dep,i}(D_{dep,i}(h(E_v)))$ are checked. If they are correct, the public key E_v is recorded to the ledger. During the voting phase requests from voters are anonymous (idemix), therefore user's eligibility can be verified using published keys.

Ability to Cancel Registration and Vote Using the Paper-Based Method. In order to provide user an opportunity to vote by the traditional method he needs to be excluded from the list of those who received a blind signature during the first sub-step of registration. This phase is not anonymous. The voter sends to the CC a request to remove the registration key E_v from the list. The CC does not completely delete the E_v key but marks it as revoked. This is necessary as if the user changes his mind and wants to register again he has to register a new key. In addition, he is removed from the list of signed users (again marking him as revoked). If the voter wants to take part in the electronic elections again he moves to the first sub-step and forms a new pair of keys.

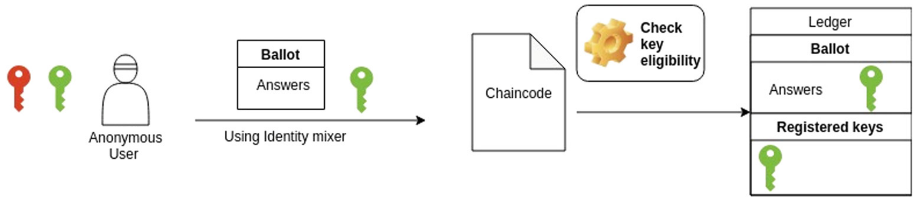

Fig. 5. Voting.

4.4 Voting

During this phase, users who have registered their private keys for voting can send encrypted ballots to chaincode (Fig. 5). Until the end of voting process the voter can change his decision and only the final one is counted. The original article [18] offers to use a symmetric key for encryption. By the end of the voting users have to submit their keys to the system for decrypting ballots and counting votes. However the voter may not provide this data (e.g., network problems). In this case, the bulletin may be considered as spoiled. Therefore it is suggested to form a pair of keys and a private part is divided between organizations and observers. Thus, the results cannot be decrypted until all participants provide their private key parts. And users will encrypt their newsletters using the above public key.

4.5 Result Counting

After the voting is completed participants with private key parts upload them to the CC. Thus, it is possible to decrypt the ballots and to get the results that are published on the ledger.

4.6 Inspectors

In order to increase the voting procedure credibility observers have been introduced into the system which are assumed to be independent and incorruptible. They have physical nodes that store a copy of the ledger. However, since these nodes are committing peers observers are not able to influence on the decision to accept or reject the transaction but they can detect cases of cheating by the other participants. Such situations can take place in the state elections.

5 Security Analysis

Eligibility. This feature assumes that a certificate (e.g., x509), provided to all users, is applied to interaction with the network and registration mechanism that was mentioned above. Note that using the Hyperledger Fabric allows users to avoid creating new certificates and work with those which are used for a particular organization. In the considered scenario it is possible to vote both electronically and on paper. That may cause the problem of defining the user in one of these two groups. In order to do that, we need to exclude those who are in the lists of voters at the first stage of user registration. This is achieved by forming a list of those who need to be deleted from the list of persons allowed to vote at the first stage of user registration. However, this approach has both advantages and disadvantages. The registration stage in the electronic voting includes two steps which should both be completed before the voting has started. Those who apply for registration are excluded from the list of voters allowed for paper voting. There is a possibility that a user does not complete the second phase and therefore he is not able to participate in the voting either through the system or in the traditional way.

Un-Reusability. An elector can vote only if the identifier (public key) is presented in the list of registered keys for the elections. The user cannot register two keys as each key must have a signature of the corresponding department. When the signature on the first key received the user is inserted into the list and no longer able to get a signature for another key.

Un-Traceability. During voting the user's rights are determined based on anonymous identifier (only the voter knows the specific id). And since all requests are sent via idemix there is no linking id (and therefore vote) with a real voter.

Verifiability. Since all voices are recorded in the ledger with reference to id the user can check the results considering his voice (using specific id) thereby making sure that voice was counted.

Receipt-Freeness. This property is not fulfilled in the present implementation since the voter can prove that a certain public key (id that is associated with the choice) belongs to him thereby approving his vote. The problem can be solved by using a ring signature. The core idea is that the ballot is signed by one participant from a certain group and after that, it is impossible to determine who is responsible. It is impossible to prove ownership of a signature even for the person who signed it.

6 Conclusion

The paper is dedicated to a modification of the e-voting protocol proposed in [18] and its implementation using a permissioned blockchain namely the Hyperledger Fabric. That leads to additional benefits:

– Increased transparency and trust,
– Possibility of cancellation of registration,
– Added observers who can detect the presence of fraud or its attempts,
– Immutable ledger history.

As a further improvement of the system, we can suggest applying ring signatures to ensure receipt-freeness as well as finding a solution to a problem when a user is not eligible to vote either electronically or in traditional way because the second stage of registration is not completed.

References

1. Madise, U., Martens, T.: E-voting in Estonia 2005. The first practice of country-wide binding internet voting in the world. In: Krimmer, R. (ed.): Electronic Voting 2006: 2nd International Workshop, Co-organized by Council of Europe, ESF TED, IFIP WG 8.6 and E-Voting.CC, August, 2nd - 4th, 2006 in Castle Hofen, Bregenz, Austria. LNI., GI, vol. 86, pp. 15–26 (2006)
2. Brightwell, I., Cucurull, J., Galindo, D., Guasch, S.: An overview of the ivote 2015 voting system (2015)
3. Springall, D., et al.: Security analysis of the estonian internet voting system. In: Ahn, G., Yung, M., Li, N. (eds.) Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3–7, 2014, pp. 703–715. ACM (2014)
4. Halderman, J.A., Teague, V.: The new south wales ivote system: security failures and verification flaws in a live online election. In: Haenni, R., Koenig, R.E., Wikström, D. (eds.) VOTELID 2015. LNCS, vol. 9269, pp. 35–53. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-22270-7_3
5. Schneier, B.: Applied Cryptography - Protocols, Algorithms, and Source Code in C, 2nd edn. Wiley, New York (1996)
6. Sheer, H., Freya, G., Apostolos, A., Raja, N., Markantonakis, K.: E-Voting with blockchain: an E-voting protocol with decentralisation and voter privacy (2018)

7. Hjálmarsson, F., Hreiðarsson, G., Hamdaqa, M., Hjálmtýsson, G.: Blockchain-based e-voting system. In: 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, 2018, pp. 983–986. IEEE (2018). https://doi.org/10.1109/CLOUD.2018.00151
8. Koç, A.K., Yavuz, E., Çabuk, U.C., Dalkılıç, G.: Towards Secure e-voting using ethereum blockchain. In: 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, 22–25 March 2018. https://doi.org/10.1109/ISDFS.2018.8355340
9. Liu, Y., Wang, Q.: An E-voting Protocol Based on Blockchain. IACR Cryptology ePrint Archive (2017)
10. Yu, B., Liu, J.K., Sakzad, A., Nepal, S., Steinfeld, R., Rimba, P., Au, M.H.: Platform-independent secure blockchain-based voting system. In: Chen, L., Manulis, M., Schneider, S. (eds.) ISC 2018. LNCS, vol. 11060, pp. 369–386. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99136-8_20
11. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology, pp. 199–203. Springer, Boston (1983). https://doi.org/10.1007/978-1-4757-0602-4_18
12. Chaum, D.: Security without identification: transaction systems to make big brother obsolete. Commun. ACM **28**, 1030–1044 (1985)
13. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**, 120–126 (1978). https://doi.org/10.1145/357980.358017
14. Blakley, G.R., Chaum, D. (eds.): CRYPTO 1984. LNCS, vol. 196. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7
15. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28628-8_4
16. Au, M.H., Susilo, W., Mu, Y.: Constant-size dynamic k-TAA. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 111–125. Springer, Heidelberg (2006). https://doi.org/10.1007/11832072_8
17. Camenisch, J., Drijvers, M., Lehmann, A.: Anonymous Attestation Using the Strong Diffie Hellman Assumption Revisited. Cryptology ePrint Archive, IACR (2016)
18. He, Q., Su, Z.: A new practical secure e-voting scheme. In: 14th International Information Security Conference (SEC 1998), IFIP/SEC (1998)