Nationwide Unified Voting System Using Blockchain and AI-Driven Face Recognition

Dr.AB.Hajira Be[1], K. Vasumathi [2]

[1] *Associate Professor*
Department of Computer Applications
Karpaga Vinayaga College of Engineering and Technology
Maduranthagam TK
[2]*PG Student*
Department of Computer Applications
Karpaga Vinayaga College of Engineering and Technology
**Corresponding Author: Vasumathi K Email: vasumathikothandaraman@gmail.com*

---***---

Abstract - Online voting for independent elections is generally supported by trusted election providers. Typically these providers do not offer any way in which a voter can verify their vote, and hence the providers are trusted with ballot privacy and in ensuring correctness. Despite the desire to offer online voting for political elections, this lack of transparency and verifiability is often seen as a significant barrier to the large-scale adoption of online elections. Adding verifiability to an online election increases transparency and integrity, as well as allowing voters to verify that the vote they cast has been recorded correctly and included in the tally. However, replacing existing online systems with those that provide verifiable voting requires new algorithms and code to be deployed, and this presents a significant business risk to commercial election providers, as well as the societal risk for official elections selecting for public office. In this paper we present the first step in an incremental approach which minimizes the business risk but demonstrates the advantages of verifiability, by developing an implementation of key elements of a Selene-based verifiability layer and adding it to an operational online voting system. Selene is a verifiable voting protocol that publishes votes in plaintext alongside a voter's tracker. These trackers enable voters to confirm that their votes have been captured correctly by the system, such that the election provider does not know which tracker has been allocated to which voter. This results in a system where even a "dishonest but cautious" election authority running the system cannot be sure of changing the result in an undetectable way, and hence gives stronger guarantees on the integrity of the election than were previously present. We explore the challenges presented by adding a verifiability layer to an operational system. The system was used in two initial trials conducted within real contested elections. We conclude by outlining the further steps in the road-map towards the deployment of a fully trustworthy online voting system.

***Key Words*:** Authentication, data security, distributed ledger, online voting, privacy, verification

1. INTRODUCTION

Verifiability in electronic voting (e-voting) plays an important role in contributing to the trust in electronic voting systems through offering both voters and observers an opportunity to verify independently whether votes have been recorded and tallied correctly. Numerous verifiability schemes have been proposed in the literature both for polling-place electronic voting and for remote voting. However, although some current commercial internet voting systems may contain verifiability mechanisms, they typically do not provide full end-to-end verifiability, as they do not provide proofs or confirming evidence that supports clear individual and universal verifiability. Within the U.K., Civica Election Services (CES; previously Electoral Reform Services) are the leading provider of independent election services, including e-voting–42% of ballots run by CES are electronic, and a further 28% are mixed mode, combining electronic and postal votes. Their system is used for organizational ballots, such as for trades unions, political parties, professional societies and building societies, and is used by 4 million individual voters a year. Currently in the U.K., e-voting is not allowed for statutory (for example, for strike action) or political ballots. Yet e-voting is advocated as a way of improving the engagement in the electorate, but only if it can be demonstrated to be sufficiently secure, something that a provable layer of security with verifiability can add. The CES online voting system provides voters with credentials that they can use to login to submit their ballots, with votes then stored securely within a database. Access to the online system is via a web browser. Once the voters have cast their votes, the votes are used in the tally according to the pre-defined rules for the election. Once a voter has cast their vote, they have no further access to the system and cannot in any way verify their vote or the election. CES are therefore trusted to collect, store and tally the votes, and to maintain the privacy of voters. This can be audited by client organizations, and by Government entities in some cases. Our aim is to move real-world online elections towards full end-to-end security and verifiability to reduce the trust that needs to be placed in providers and improve transparency. We aim to do so in steps, for two reasons. First, we aim to slowly get general voters to expect electronic elections to provide certain features, such as individual and public verifiability. This reflects the observation by Kulyk et al. [35, p. 18] that "the idea of verification, being a fairly alien concept, is problematic. More needs to be done to familiarize voters with the differences between paper and Internet voting." Second, although in principle providing a complete implementation of a new fully end-to-end secure and verifiable system from scratch would be the ideal approach, their use in real elections is not something that can be easily achieved, especially with certification constraints, when a trusted but not end-to-end-verifiable system already exists. For example, CES are a Secretary of State-appointed "scrutineer" for industrial ballots, and systems changes could put that appointment at risk if not done carefully. We therefore take a more pragmatic, layered approach, which we describe below. In this paper, we focus on the first step on that roadmap: the addition of a verifiability layer to support the verification of the ballot's integrity independently of the election provider. As a consequence, the election provider themselves–or malicious insiders or external attackers gaining access to their critical systems– cannot change the result without risking detection.

2. RELATED WORK

There are numerous proposals for end-to-end verifiable voting systems, including Prêt à Voter, Wombat, Scantegrity II, Helios, D-DEMOS, Belenios, Civitas and Selene. These make use of common verifiability mechanisms to underpin the integrity of the election. Some use paper whereas others are purely electronic, and some are intended for use in the polling place, whereas others are intended for remote voting from a voter's own device. A common mechanism for verifying that a vote is cast as intended is the Benaloh Challenge, a cut-and-choose method for confirming that a vote has been constructed correctly. After the vote has been created for submission to the election system, typically by encrypting it, the voter can choose whether to cast the vote or to audit it. An audit involves revealing the vote and providing the evidence that it was indeed encrypted correctly. Audited votes cannot also be cast since their contents have been revealed, contrary to the secret ballot. Therefore a voter can audit several votes before finally deciding to submit an unopened one.

Audits are similar to random sampling of votes: if votes are not constructed correctly then an audit would catch this, so an election with sufficient successful audits gives some level of evidence that all cast votes are also constructed correctly. This approach is taken in Helios, Belenios, Civitas and Wombat. This cut-and-choose approach is also applicable to pre-constructed ballot forms, which can either be used to vote or can be audited (without a vote) to check they have been constructed correctly. This approach is taken in Prêt à Voter and Scantegrity II. Typically a voter will retain a record of the vote that was cast, and will be able to confirm that this matches the published list of all the votes cast, to verify recorded as cast. Most commonly this record will contain the vote in some encrypted form, so it does not reveal the vote. A system that does not provide the voter with a way of revealing how they voted is known as receipt-free. Receipt-freeness is a desirable requirement of voting systems. An alternative approach to obtaining individual verifiability is through the use of Code Voting as provided for example in Pretty Good Democracy. This approach provides (by post or some other private channel) each voter with a code sheet which contains a voting code for each candidate, and a return code. The voter casts a vote by submitting the code for their candidate, and on receiving the return code they obtain confirmation that the vote has been correctly received, since only the election system has knowledge of the voting codes and return codes. This verifiability property assumes that the codes remain secret, at least until the verification step has taken place. The approach taken by Selene is different again: at the end of the election all votes are published alongside a tracker, and each voter is provided privately with their tracker. They are then able to confirm directly that the vote against their tracker is indeed the vote they cast, verifying that it has been cast as intended. This approach enables voters to verify their votes in the clear, rather than in encrypted or code form, so much of the design of Selene is to protect privacy, by ensuring that voters cannot prove their tracker to any other party, and that they do not obtain it until after all the votes have been published. Universal verifiability works with the published list of encrypted votes, which can be processed in a universally verifiable way to obtain the result of the election. There are two main approaches to achieving this. The first is to use an anonymising mix-net to shuffle and re-encrypt the ballots, resulting in a list of encrypted ballots that cannot be matched to the voters' receipts. Zero-knowledge proofs of shuffling or randomised partial checking enable independent verification that this has been carried out correctly. The resulting list is decrypted to reveal the plaintext votes, which can then be tallied publicly in the normal way. All of these steps obtaining the result can be independently verified. This is the approach taken by Prêt à Voter, Wombat, Helios v3, Belenios and Civitas, as well as Selene. An alternative approach is to encrypt the votes in such a way that we can make use of homomorphic encryption, enabling the encrypted total for each candidate to be obtained from the individual encrypted votes. These encrypted totals can then be decrypted to reveal the results, without revealing any individual votes. All of these cryptographic steps can also be carried out in a verifiable way. This is the approach taken in D-DEMOS, Helios and Belenios. Large-Scale Deployments of Cryptographic Online Voting: We are not the first to aim at large scale deployments of online voting schemes in real elections. Other deployments on potentially similar scales are Government-driven. Norway allowed the use of online voting in some elections, but stopped in 2014. The short deployment was used to support further research on the effectiveness of verifiability at detecting tampering. Estonia also uses online voting to complement their citizens' ability to vote in polling stations. Their system provides natural coercion resistance by allowing re- voting (so that a coercer would need to actively keep coercing the voter throughout the election period) and giving precedence to votes cast in-person. Their deployment started from a core private but not end-to-end verifiable system and more recently added verifiability features . It is backed by existing infrastructure for the management of voter credentials (through Government-issued electronic identification), which our system cannot assume exists. A number of Swiss cantons have been experimenting with electronic voting for parts of their monthly votations. A variety of systems have been deployed in practice, some based on Benaloh challenges, and some on code voting. A more recent proposal by Scytl and SwissPost [46] suggests the use of return codes to avoid Benaloh challenges while providing cast as intended verification. This same system is also in use in Australia and in some French elections. There are no studies to date on interactions of code voting alongside pen-and-paper-based voting for the same elections.

3. PROPOSED SYSTEM

The proposed system is designed to overcome the limitations of traditional voting methods. By integrating cutting-edge technologies such as blockchain, facial recognition, and multi-level authentication, this system aims to ensure a more secure, efficient, and transparent electoral process.

4. MODULES

4.1 Voting Web Service The existing CES e-voting system which operates without change except to provide additional information to voters to allow them to verify their vote.

4.2 Vote Database the existing CES relational database holding all details about an election, voters and their plaintext vote (once a ballot has been cast). This is modified to add in the verifiability data per voter and is used as the input and output interface for VMV through the import and export of comma-separated values (CSV) data files.

4.3 CES Network The secure network within which the Voting Web Service and Vote Database are held. Public access is only granted to the Voting Web Service within this network via HTTPS (and to vote only with credentials). Since the Selene Layer accesses voter and vote data, it is also run within the CES Network to ensure that all private data is kept securely within the network.

4.4 Selene Layer Executes the Selene protocol by taking data from the Vote Database as CSV files, communicating with the Verificatum Nodes to perform shuffling and decryption, and with the Verification Web Service to publish verification data, including produced CSV and NIZKPoK proof files. These operations are initiated by an administrator using a computer running within the CES Network.

4.5 Verificatum A series of independently-operated nodes running the Verificatum software. Two or more independent organisations can run a Verificatum Node which is initialised by the Selene Layer. Each Verificatum Node can communicate with each other node within the Mix-net Network. Prior to a mix-net operation, such as shuffling, each node is supplied with identical CSV input and produces identical CSV output together with the corresponding proof files.

4.6 Mix-net Network Each Verificatum Node is run within its own secure network hosted by each independent organisation. Access to each Verificatum Node is only granted to the other Verificatum Nodes and the Selene Layer, which controls the Verificatum operations.

4.7 Verification Web Service A web service with a user interface which allows administrators to publish verification data, auditors to view the published election data and voters to verify their vote. This forms the public face of the VMV demonstrator and allows published files to be served to users. Publication requires privileged access granted to administrators via user accounts. Only administrators have accounts, while anyone can view published data.

4.8 Verification Database Holds the data necessary to run the Verification Web Service, including administrator user accounts and an index of each election’s verification data. This includes the list of the CSV and proof files held in the Data Lake, and their corresponding contract addresses in the Quorum cluster, such that they can be retrieved via the Verification Web Service.

4.9 Data Lake holds the published CSV and NIZKPoK proof files in a repository which is only accessed via the Verification Web Service. Verification Network A secure network in which the Verification Web Service and Data Lake operate. Public access is only granted to the Verification Web Service within this network via HTTPS.

4.10 Quorum Node A series of independently-operated nodes running the Quorum software, a particular Distributed Ledger Technology. Two or more independent organizations can each run one or more Quorum Nodes. Each Quorum Node can communicate with each other node within the DLT Network. When a file is published via the Verification Web Service, it is saved to the Data Lake and a hash of the file is committed to the Quorum cluster. Periodically, the hash is verified against the file held in the Data Lake to ensure its integrity

5. RESULT

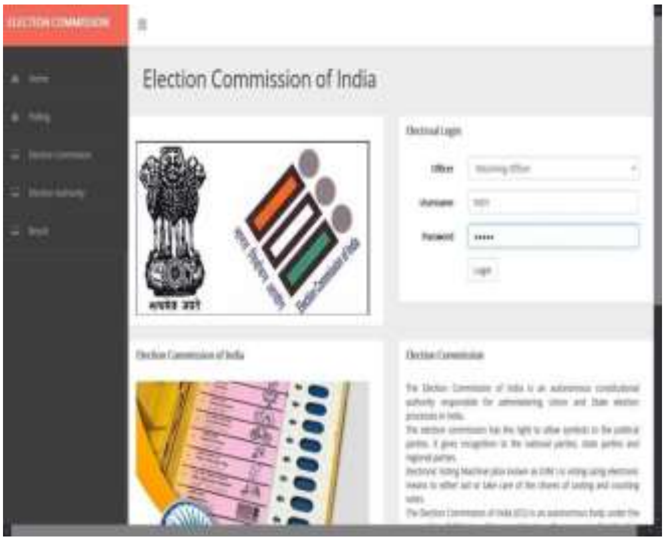

Fig -1: Election Commission of India

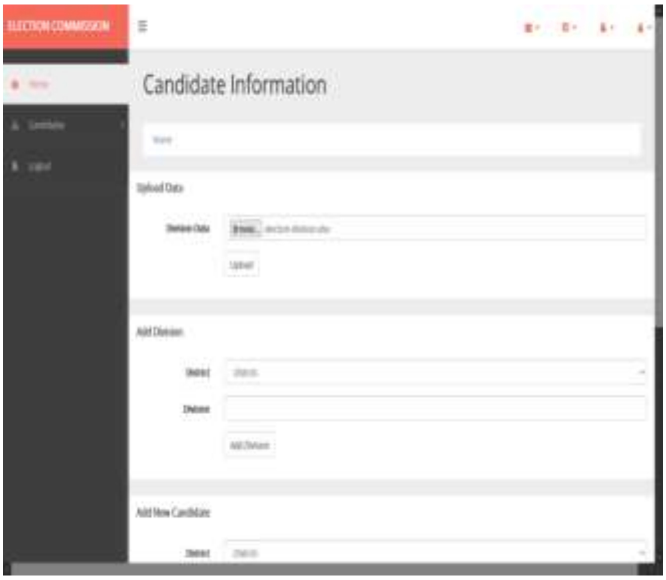

Fig -2: Candidate Information

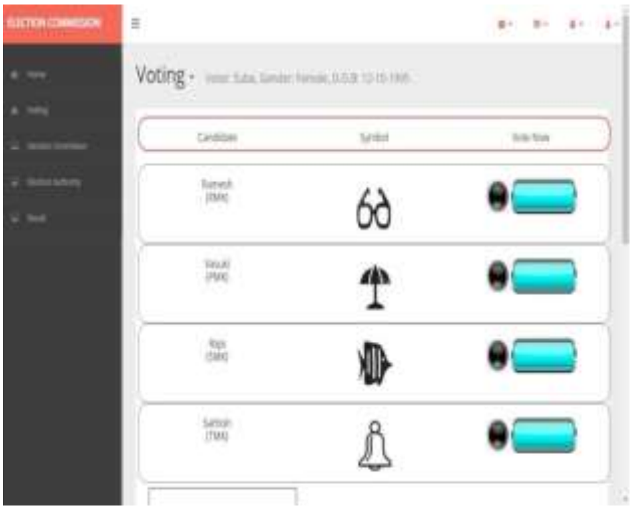

Fig -3: Voting

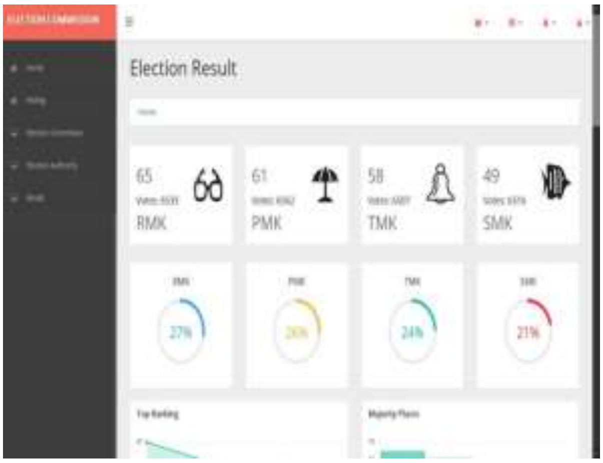

Fig -4: Election Result

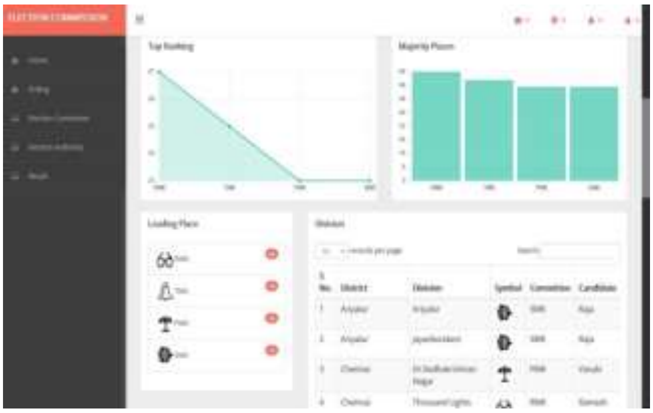

Fig -5: Election Result Analytics

6. CONCLUSIONS

In this paper we proposed VMV ("Verify My Vote"), which adds Selene verifiability mechanisms onto a deployed internet voting system run by our commercial partner CES. Although this is an initial step it has already resulted in a system which provides stronger integrity guarantees for the CES system than it presently has, with VMV used as a simple external auditor for the conduct of specific elections. The system provides individual and universal verifiability provided CES and VMV are not colluding to break the integrity of the election. This initial system also has provided us with a platform for running trials "in the wild" on live elections to explore practical and usability issues, and to investigate open questions around voters'

Understanding and attitude to this approach to verifiability. Our findings from the initial trials[3] are that voters are able to manage the current level of verification provided to confirm that the system has correctly recorded their vote. As further features of Selene are also introduced, these can also be investigated for usability, understanding and attitude. Further work will need to investigate how effectively voters are able to notice mistakes in the evidence they are presented with, and also to compare with other approaches. These questions will need to be studied in a controlled setting, where voters are aware that they are participating in a trial. The VMV system we have developed can be used for this purpose.

REFERENCES

1. C. Z. Acemyan, P. T. Kortum, M. D. Byrne, and D. S. Wallach, “Usability of voter verifiable, end-to-end voting systems: Baseline data for Helios, Prêt à Voter, and Scantegrity II,” in Proc. Electron. Voting Technol. Workshop/Workshop Trustworthy Elections, 2014, pp. 26–56. .
2. B. Adida, “Helios: Web-based open-audit voting,” in Proc. 17th USENIX Secur. Symp., 2008, pp. 335–348.
3. M. Alsadi and S. Schneider, “Verify my vote: Voter experience,” in Proc. Int. Conf. Electron. Voting, 2020, Art. no. 280.
4. Amazon Web Services, Inc., “Welcome to AWS documentation,” 2023. Accessed: Nov. 8, 2023. [Online].
5. Available: https://docs.aws.amazon. com/
6. M. Arnaud, V. Cortier, and C. Wiedling, “Analysis of an electronic boardroom voting system,” in Proc. 4th Int. Conf., Springer, 2013, pp. 109–126.
7. J. Ben-Nun et al., “A new implementation of a dual (paper and cryptographic) voting system,” in Proc. 5th Int. Conf. Electron. Voting, 2012, pp. 315–329.
8. J. Benaloh, “Simple verifiable elections,” in Proc. USENIX/ACCURATE Electron. Voting Technol. Workshop, 2006, pp. 5–5.
9. J. Benaloh, R. L. Rivest, P. Y. A. Ryan, P. B. Stark, V. Teague, and P. L. Vora, “End-to-end verifiability,” 2015, arXiv:1504.03778.
10. Blockchain Solutions Group, “Quorum whitepaper,” 2017. Accessed: Nov 14, 2021. [Online]. Available: https://www.blocksg.com/single-post/ 2017/12/27/Quorum-Whitepaper
11. M. Casey, “VMV: Verify my vote software,” 2020. Accessed: Nov. 8, 2023. doi: 10.5281/zenodo.3695909.
12. P. Chaidos, V. Cortier, G. Fuchsbauer, and D. Galindo, “BeleniosRF: A non-interactive receipt-free electronic voting scheme,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., 2016, pp. 1614–1625.
13. D. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” Commun. ACM, vol. 24, no. 2, pp. 84–90, 1981.
14. D. Chaum et al., “Scantegrity II: End-to-end verifiability for optical scan election systems using invisible ink confirmation codes,” in Proc. USENIX/ACCURATE Electron. Voting Workshop, 2008, Art. no. 13.