

Toward a Compliant Token-Based e-Voting System with SSI-Granted Eligibility

Dario Castellano^{1,2}^a, Roberto De Prisco¹^b and Pompeo Faruolo²

¹University of Salerno, Computer Science Department, Salerno, Italy

²eTuitus, Fisciano (SA), Italy

www.etuitus.it

Keywords: e-Voting, Self-sovereign Identities, Algorand, Dizme.

Abstract: In this paper we present a preliminary design for an e-voting system based on self-sovereign identities and built on the Algorand blockchain. The design keeps into consideration the basic properties of an e-voting system and also the EU recommendations. We use the Dizme framework for the management of the identities of the voters, which allows to keep secret the identity while certifying the right to vote, and we store the encrypted votes on the Algorand blockchain. Votes are decrypted only in the tally phase.

1 INTRODUCTION

Computers and the Internet have revolutionized services, through the possibility of offering them online. One of such services is (online) voting. Extensive research conducted since the early 2000s has focused on establishing secure electronic voting standards and methodologies (Caltech and MIT, 2001; Rivest, 2000), although security concerns persist (Rubin, 2001). The use of flawed Direct-recording electronic voting machines in the US underscored the importance of open-source systems and community auditing for enhanced security (Kohno et al., 2004). While full remote Internet voting has yet to be widely embraced for large-scale elections, the European Parliament appointed a committee to investigate measures and standards (Council of Europe, 2021b), recommending them to member states. Italy, for instance, has explored the implementation of an electronic voting system in compliance with the constitution, conducting tests that enabled overseas citizens to participate (Cortellessa, 2020).

The emergence of blockchain technology has sparked considerable interest in e-voting due to its inherent security features (Kshetri and Voas, 2018; Demuro, 2018). Real-world cases in Moscow, South Korea, Estonia, and Sierra Leone have demonstrated the feasibility of blockchain-based voting for polls

and surveys (Kshetri and Voas, 2018; Soldavini, 2018). Blockchain-enabled e-voting (BEVs) offers several advantages, including participant anonymity, immutable vote storage, and decentralized participation, which can potentially reduce errors and vulnerabilities associated with traditional electronic voting machines. Various models and approaches have been proposed, such as permissioned blockchains and the use of decentralized layers into existing systems (Lee et al., 2016; Hjalmarsson et al., 2018; Perez and Ceesay, 2018).


Our proposed solution starts from European recommendations and leverages the self-sovereign identity approach using the Dizme identity network¹, combined with the secure and permanent storage capabilities of the Algorand blockchain². By utilizing these technologies, we aim at empower an e-voting solution that ensures the integrity and transparency of the voting process while preserving voter privacy and security.


2 BACKGROUND

In this section we first review the basic requirements of an e-voting system and then we describe briefly

¹The Dizme Identity Framework documentation is available at <https://www.dizme.io/>

²Algorand documentation is available at <https://algorand.com>

^a <https://orcid.org/0009-0001-3756-9204>

^b <https://orcid.org/0000-0003-0559-6897>

the Dizme framework and the Algorand blockchain. E-voting research field is rich of contributions that inspired common requirements and criteria in the design of the proposed voting system, e.g. (Anane et al., 2007; Bungale and Sridhar, 2013; Gibson et al., 2016; Rubin, 2001; Wang et al., 2017).

2.1 Basic Requirements and EU Recommendations

Voting systems have fundamental requirements that include:

- *Non-reusability* to prevent multiple votes from a single voter.
- *Non-duplicability* to avoid duplicate votes within the system.
- *Immutability* to ensure that cast votes cannot be altered.
- *Non-traceability* to maintain voter preferences confidential.
- *Eligibility* to authorize individuals to cast votes based on personal information.

In addition to these requirements, auditability and verifiability are crucial for the integrity of the system:

- *Auditability* involves practices to detect and prevent fraud in an election, which is mandatory for e-voting systems (Jamroga et al., 2019; Rubin et al., 2020).
- *Verifiability* enables actors to control the system's inputs and outputs, ensuring correct operation (Rivest and Stark, 2017). Verifiability encompasses the votes being Cast-as-intended, Recorded-as-cast, and Tallied-as-recorded (Benaloh, 2006; Benaloh et al., 2015).

The European Commission's recommendations provide a comprehensive set of requirements for e-voting systems (Council of Europe, 2021b), aiming at addressing various aspects:

- *Universal suffrage* emphasizes user-friendly interfaces and system accessibility to avoid a digital divide.
- *Equal suffrage* focuses on equal access to the system and equitable representation of information for all voters.
- *Free suffrage* promotes transparency, clear feedback to voters, integrity of ballots, and verifiability.
- *Secret suffrage* covers the protection of private data, secure management of eligibility information, receipt freeness, single submission,

anonymity, and prevention of result counting during ongoing elections.

Various e-voting systems, such as Helios (Adida, 2008), have sought to merge auditability and verifiability through risk-limiting audits and providing voters with detailed information at each step. Additionally, cryptographic tools like mix-nets, homomorphic encryption, and blind signatures have been used in digital or hybrid processes to fulfill the requirements (He and Su, 1998).

2.2 Algorand Overview

Algorand is a high-performance blockchain network known for its fast, secure, and decentralized transactions (Algorand, 2023). It utilizes a proof-of-stake consensus mechanism, ensuring scalability and achieving transaction finality within seconds (Chen and Micali, 2016; Chen et al., 2018). Key features of Algorand include:

- **Scalability:** Algorand supports high throughput, with the capacity for up to 1,000 transactions per second (TPS), making it suitable for enterprise applications requiring fast transaction times and high volume.
- **Security:** Algorand employs a unique cryptographic protocol that ensures transaction validation without relying on a centralized authority, guaranteeing the security and integrity of transactions.
- **Decentralization:** Algorand's consensus mechanism based on proof-of-stake promotes decentralization, making the network resistant to censorship and control by any centralized authority.
- **Smart Contracts:** Algorand supports smart contracts using TEAL, a user-friendly programming language. Smart contracts can be stateless or stateful, providing flexibility for transaction validation or complete distributed applications.
- **Atomic Transfers:** Algorand enables atomic transfers, ensuring that multiple transactions are executed as a whole or not at all, eliminating the risk of partial execution.

Algorand finds applications in various use cases, including financial applications requiring fast transactions and high throughput, decentralized applications (dApps) benefiting from strong security and decentralization, and tokenization, where smart contracts enable the creation of digital assets like tokens or non-fungible tokens (NFTs) (Algorand, 2023).

2.3 Self-Sovereign Identity (SSI) and Dizme Framework

Self-Sovereign Identity (SSI) systems transfer control of identity from a centralized authority to a self-governed system, where users have full control over their identities. This model eliminates the need for a central authority and returns identity and related claims to the user. Distributed ledgers and blockchain technologies have paved the way for such models. Various SSI implementations exist, including Sovrin, uPort, Civic, and The Key. In this work, we reference the Sovrin Hyperledger as the SSI implementation. The Sovrin Foundation, an international non-profit organization, governs the Sovrin hyperledger, which serves as the world’s first self-sovereign identity network³. Dizme, a framework, utilizes Sovrin to manage identity credentials. Within this framework, entities known as “issuers” can issue identity credentials, and users, in this case, the voters, can store these identities in private wallets and present them to the system for voting purposes.

2.3.1 Dizme Usage

We employ Dizme for vote credential issuing and validation, where the credential represents the attestation that the user is eligible to vote based on identification criteria. Voters can prove that they possess required identity attributes, such as citizenship or age above 18, using a zero-knowledge proof on the credential containing those attributes. Once verified, the voter receives a credential offer containing an anonymous voting credential.

To establish eligibility, the voting credential is combined with a self-attested attribute representing the voter’s Algorand address, forming a proof. Any verifier in the Dizme framework can verify that the credential is a voting credential and provide the voter with an Algorand asset on the provided address. This ensures eligibility without disclosing any voter personal data and prevents any link between voter identity and Algorand address.

Figure 1 illustrates a diagram of the actors involved in the Dizme framework.

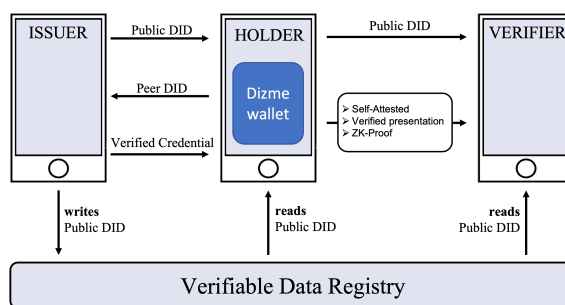


Figure 1: Dizme actors.

proof of concept for further investigation. Next, we define the actors involved in the system’s construction and describe their interactions.

3.1 Actors and Assumptions

The e-voting process involves several actors, each with specific roles, and certain assumptions are made in the proposed protocol.

Actors:

- **Organizer:** Configures election parameters, manages cryptographic keypairs, and oversees the entire election process.
- **Voter:** Interacts with the system to express their preference. They have self-sovereign identity and Algorand blockchain wallets. The voter obtains a valid identity credential, presents it to the Identity Verifier for an election token, and casts their vote by spending the token through a blockchain transaction.
- **Identity Issuer:** Verifies the voter’s identity attributes and issues an election-specific credential certifying their eligibility.
- **Identity Verifier:** Validates the voter’s election credential and token to confirm their eligibility without revealing additional identity information.

Assumptions:

- **Token-based Voting:** Voting eligibility is determined by owning an unrestricted Algorand Asset, which may evolve in the future.
- **Dizme Wallet and SSI:** Voters are assumed to have a Dizme wallet for managing credentials and are familiar with self-sovereign identity (SSI) concepts.
- **Protection of Sensitive Information:** Zero-knowledge proofs are used in SSI interactions to verify possession of information without revealing the actual information, ensuring privacy.

3 PROPOSED FRAMEWORK

This section presents the design of a compliant e-voting system based on the technical recommendations of the European Commission. It serves as a

³Details about Sovrin can be found at <https://sovrin.org/>

3.2 Voting Protocol

The proposed solution allows voters to participate in the election process using their personal devices, making it suitable for medium-scale elections or corporate settings where coercion is not a significant concern. The protocol consists of the following phases:

1. **Initialization:** The Organizer configures the election parameters, deploys the election token on the Algorand blockchain, and sets up the election smart contract. Voters interact with the Identity Issuer to obtain a valid election credential. See Figure 2.
2. **Registration:** Voters interact with the Identity Verifier to obtain a voting token. The Verifier verifies the voter’s election credential and registers an asset transfer transaction on the blockchain. See Figure 3.
3. **Voting:** During the designated voting period, eligible voters interact with the smart contract to encrypt their preference and submit it along with an asset transfer transaction containing the voting token. The encrypted preference is stored in the address-related space provided by Algorand. See Figure 4.
4. **Tally:** After the voting period ends, the Organizer publicly releases the election private key through the smart contract. The tally phase involves retrieving the encrypted preferences, decrypting them, and computing the final result, which can be made publicly available. See Figure 5.

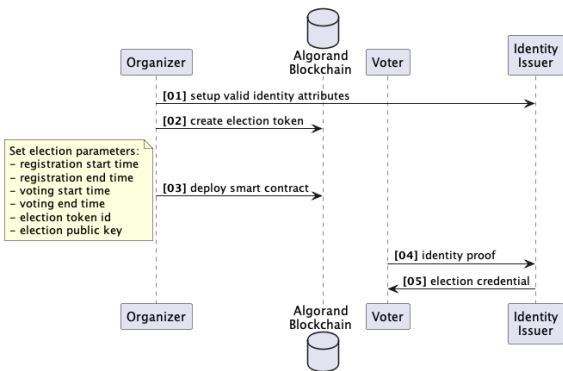


Figure 2: Initialization phase.

3.2.1 Contract Overview

This section provides an overview of the available interactions with the contract and their behavior.

`Opt-in`: This Algorand smart contract call allows

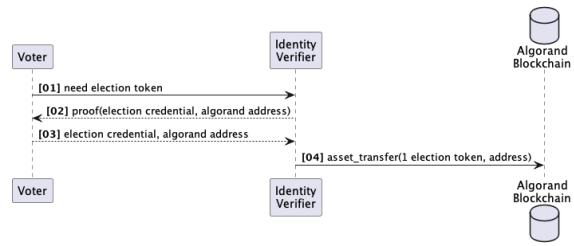


Figure 3: Registration phase.

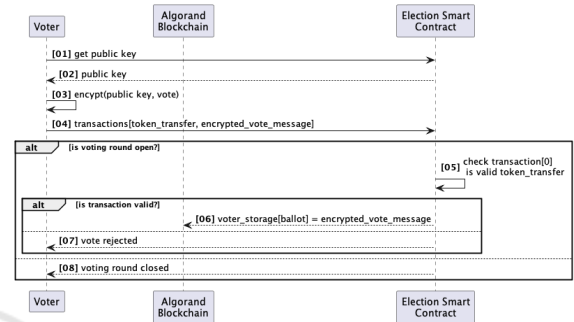


Figure 4: Voting phase.

an address to require storage space for encrypted ballots. It can only be used during the registration phase.

`Create`: This application call creates the contract and sets its global state. Arguments include the Election Token Id, Election Public Key, Registration Start Date, Registration End Date, Voting Start Date, and Voting End Date.

`Vote`: This application call requires two transactions as arguments: an asset transfer from the voter to the contract and a transaction containing the encrypted ballot. It validates the timing of the vote and checks the transaction group. If successful, it stores the encrypted ballot in the on-chain storage for the voter’s address.

`UpdateKey`: This call updates the global state by setting the election private key. It can only be made by the Organizer after the voting period has ended. In addition to the smart contract calls, the following Algorand application calls are used:

`GetGlobalState`: This call provides access to the blockchain’s global storage for the contract.

`GetLocalState`: This call retrieves the encrypted ballot stored in the contract’s local storage for a given voter’s address.

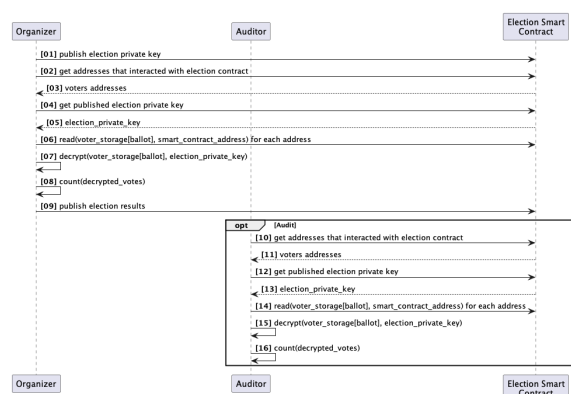


Figure 5: Tallying phase.

4 ANALYSIS

4.1 Basic Requirements

To ensure the correctness of the election and security for any application, the following requirements are considered:

Verifiability: The use of blockchain technology ensures that all operations are recorded permanently and are accessible. Voters can verify their vote submission and the overall vote tally. The three requirements of verifiability (Cast-as-intended, Recorded-as-cast, and Tallied-as-recorded) are met through established cryptographic procedures and transparency of the blockchain.

Eligibility: Verified credentials are used to determine eligibility. Voting tokens are conditional on possessing specific attributes, and the smart contract verifies their use.

Auditability: The voting process is auditable during and after the vote due to the recording of interactions with the contract on the blockchain.

Non-reusability: Tokens can only be acquired during the registration phase, and the system allows the spending of only one token per user to prevent multiple voting.

Non-duplicability and immutability: The blockchain layer prevents duplicability and ensures immutability of the contract and of the vote.

Non-traceability: The usage of a self-sovereign identity (SSI) framework ensures that the voter’s identity credential cannot be linked to the blockchain address used to call the contract.

4.2 EU Recommendations

The recommendations from the European Commission have been considered in the design:

Universal suffrage: The fee mechanism of blockchain usage is supported, and the protocol can be implemented in a public booth to ensure accessibility and address the digital divide.

Equal suffrage: The use of SSI with smart contract transparency enables equal suffrage in the system.

Free suffrage: The immutability, data integrity, and transparency of the blockchain ensure free suffrage.

Secret suffrage: Privacy is achieved by keeping the voter’s identity separate and using anonymous addresses. Vote counting is not possible during the voting phase due to encryption, with decryption occurring only after the voting period ends.

5 IMPLEMENTATION

We are developing a proof of concept implementation to test the feasibility of our proposed solution. The implementation utilizes the Algorand Python library and simulates the actors and their interactions. We use an Algorand testnet node provided by PureStake and generate different Algorand accounts funded with ALGOs for testing.

The current implementation focuses on simulating the identity process, where the voter account is provided with a token and sends an atomic transactions group to the contract. The implementation consists of Python scripts for both the server and the client (actors). We initially use a central server for data access and logging, but we are working on a serverless solution where the client code interacts directly with the Algorand blockchain.

We are also exploring different settings for the e-voting system. In addition to the proof of concept, we are developing a mobile solution using Dart and Flutter, a cross-platform technology, to enable a full remote e-voting experience. We incorporate technologies such as Wallet Connect⁴ for connecting the app with an Algorand account and face or touch recognition for secure access to the application.

6 CONCLUSIONS AND FUTURE WORK

Implementation and User Experience: The development of a complete working implementation is planned, including considerations for user interface

⁴Documentation about Wallet Connect can be found at <https://docs.walletconnect.com/2.0/>

and user experience. It is important to address the accessibility of the system and provide clear interfaces with informative feedback to ensure user understanding and engagement (Council of Europe, 2021b).

Certification Processes: Further exploration of certification processes for e-voting systems is needed. Current EU recommendations allow member states to define their own certification procedures, so the system should be designed with the goal of meeting certification requirements and addressing relevant technological aspects (Council of Europe, 2021a).

Cross-Chain Technologies: Investigating the use of cross-chain technologies, such as light-clients or ad-hoc chains⁵, can contribute to a more versatile and generic protocol. Considerations can also be made for managing fees associated with the system's operations (Yang, 2020).

By addressing these aspects, the proposed design can evolve into a more comprehensive and certified e-voting system.

ACKNOWLEDGEMENTS

This work was partially supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU-NGEU.

REFERENCES

- Adida, B. (2008). Helios: Web-based open-audit voting. In *USENIX security symposium*, volume 17, pages 335–348.
- Algorand (2023). Algorand - <https://algorand.com>.
- Anane, R., Freeland, R., and Theodoropoulos, G. (2007). E-voting requirements and implementation. In *The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (CEC-EEE 2007)*, pages 382–392. IEEE.
- Benaloh, J. (2006). Simple Verifiable Elections. *EVT*, 6:10.
- Benaloh, J., Rivest, R., Ryan, P. Y. A., Stark, P., Teague, V., and Vora, P. (2015). End-to-end verifiability. [arXiv:1504.03778 \[cs\]](https://arxiv.org/abs/1504.03778).
- Bungale, P. P. and Sridhar, S. (2013). Requirements for an Electronic Voting System. *Unpublished Thesis, Department of Computer Science, The Johns Hopkins University*, page 2.
- Caltech and MIT (2001). Voting - What Is, What Could Be. Technical report.
- Chen, J. and Micali, S. (2016). Algorand. *arXiv preprint arXiv:1607.01341*.
- Chen, J. J., Gorbunov, S., Micali, S., and Vlachos, G. (2018). ALGORAND AGREEMENT: Super Fast and Partition Resilient Byzantine Agreement. *IACR Cryptol. ePrint Arch.*, 2018:377.
- Cortellessa, C. M. (2020). Il voto elettronico tra standard europei e principi costituzionali. Prime riflessioni sulle difficoltà di implementazione dell'e-voting nell'ordinamento costituzionale italiano.
- Council of Europe (2021a). Certification of e-voting systems. Technical report, Council of Europe.
- Council of Europe (2021b). List of reference standards of the Council of Europe in the field of elections - Portal - publi.coe.int.
- Demuro, J. (2018). Here are the 10 sectors that blockchain will disrupt forever.
- Gibson, J. P., Krimmer, R., Teague, V., and Pomares, J. (2016). A review of e-voting: the past, present and future. *Annals of Telecommunications*, 71(7):279–286. Publisher: Springer.
- He, O. and Su, Z. (1998). *A new practical secure e-voting scheme*. na.
- Hjalmarsson, F. P., Hreiðarsson, G. K., Hamdaq, M., and Hjalmytsson, G. (2018). Blockchain-Based E-Voting System. *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 983–986.
- Jamroga, W., Roenne, P. B., Ryan, P. Y. A., and Stark, P. B. (2019). Risk-Limiting Tallies. [arXiv:1908.04947 \[cs, stat\]](https://arxiv.org/abs/1908.04947).
- Kohno, T., Stubblefield, A., Rubin, A. D., and Wallach, D. S. (2004). Analysis of an electronic voting system. In *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, pages 27–40. IEEE.
- Kshetri, N. and Voas, J. (2018). Blockchain-enabled e-voting. *IEEE Software*, 35(4):95–99. Publisher: IEEE.
- Lee, K., James, J., Ejeta, T., and Kim, H. (2016). Electronic voting service using block-chain. *Journal of Digital Forensics, Security and Law*, 11(2).
- Perez, A. J. and Ceesay, E. N. (2018). Improving End-to-End Verifiable Voting Systems with Blockchain Technologies. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1108–1115.
- Rivest, R. L. (2000). Electronic Voting.
- Rivest, R. L. and Stark, P. B. (2017). When Is an Election Verifiable? *IEEE Security & Privacy*, 15(3):48–50.
- Rubin, A. (2001). Security Considerations for Remote Electronic Voting over the Internet.
- Rubin, A., Halderman, J. A., Adida, B., and Teague, V. (2020). The 2020 Election: Remote Voting, Disinformation, and Audit. USENIX Association.
- Soldavini, P. (2018). In Sierra Leone le prime elezioni al mondo garantite da blockchain.
- Wang, K.-H., Mondal, S. K., Chan, K., and Xie, X. (2017). A review of contemporary e-voting: Requirements, technology, systems and usability. *Data Science and Pattern Recognition*, 1(1):31–47.
- Yang, J. (2020). Blockchain Light Client.

⁵Algorand Co-Chains properties are described at <https://www.algorand.com/resources/blog/algorand-co-chains>