IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

ElectChain: A Decentralized Blockchain-Based E-Voting System with Self-Sovereign Identity and Privacy Preservation

Dr. Mrunal Pathak[1], Chinamy Nakwa[2], Akshat Bhosale[3], Sarthak Deochake[4]
Associate Professor, Department of Information Technology [1]
Student, Department of Information Technology[2-4]
AISSMS Institute of Information Technology, Pune, India

Abstract: *In the world's largest democracy, the integrity of an election relies entirely on public trust. Yet, every election cycle brings new questions about the security of traditional Electronic Voting Machines (EVMs) and the opacity of how votes are counted behind closed doors. This paper proposes a solution to replace blind trust with cryptographic proof: a nationalscale voting system built on a permissioned Ethereum blockchain. The core goal of this project is to create an election process where anyone can verify the result, but no one can violate the voter's privacy. To achieve this, we move away from simple database entries. Instead, we use Self-Sovereign Identity (SSI) to verify voters without tracking them and a "commit-reveal" mechanism that acts like a digital sealed envelope locking the vote on the blockchain so it cannot be tampered with but keeping it secret until the election closes. We demonstrate this architecture using a Next.js web application and smart contracts. By comparing this system against traditional hardware EVMs, this paper shows that blockchain technology can offer a transparent, audit-proof alternative for modern democracies, ensuring that while the ballot remains secret, the counting process is open for the whole world to see.*

Keywords: Electronic Voting, Blockchain, Self-Sovereign Identity (SSI), Smart Contracts, Ethereum

I. INTRODUCTION

Credibility of voting infrastructure is the key to the success of any democratic process. Even though the traditional paperbased voting is not a secret, it is vulnerable to ingenious logistical nuisance and geographical obstacles to access, not to mention counting errors related to human counting. Electronic voting was considered to be a feasible solution, although the overwhelming majority of installed systems rely on centralized architectures, which have their own devastating weaknesses: single points of failure, insider manipulation, opaque tallying, and loss of voter privacy. This has made the world come up with more resilient alternatives by conducting a global search.

Electronic voting has been of great popularity due to the use of blockchain technology since it is immutable, transparent, and decentralized. It is recorded on a shared register through voting which is in blocks and the blockchain removes the issue of a trusted third party as well as a verifiable history that any party can access. Nevertheless, blockchain only does not eliminate all the challenges that digital voting has. The risk of sensitive voter metadata is revealed in public registries, and the conventional architectures of digital identity subject voters to the less-than-ideal guarantee of blockchain decentralization, which is centralized identity regimes.

SSI has become an adjoining paradigm that reinvigorates the ownership of digital credentials on an individual basis. In an SSI model,[SSI] gauges have cryptographic keys, which certify eligibility without giving any individual data to any third party. In addition to privacy-preserving cryptographic schemes and smart contracts, SSI provides a platform, based on which a system of voter authentication can be implemented in a verifiable and anonymous way.

DOI: 10.48175/IJARSCT-32675

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, April 2026 Impact Factor: 8.2

The compound of all these technologies into deployable architecture is referred to as the ElectChain and this is the solution. ElectChain is an initiative that provides election rules in an immutable manner using off-chain ECDSA signatures to simulate an SSI-style identity check, a commit-reveal scheme to provide the privacy of votes during the election time frame, and Foundry-compiled smart contracts to execute on a local Ethereum net (Anvil). It is created with Next.js, Wagmi, and Viem and has a convenient interface, which is connected to a wallet.

Contributions. The main contributions of the work are the following. The first step that we take is to implement a threelayered security architecture of e-voting, which isolates the challenges of identity, privacy and integrity into modular units. Second, the Solidity smart contracts are both commit-reveal and verifiable, which does not leave the risks of results being disclosed prior to the election process. Third, we illustrate a SSI-based authentication scheme simulated in which the authentication scheme is ECDSA signatures based and not centralized on identity providers. Fourth, we provide the detailed analysis of the gas costs in all the required operations of the contract and this is why there is the efficiency of the implementation proposed. Fifth, we build a contemporary React application that comes together with the Ethereum network using standard wallet infrastructure.

The paper is also subdivided in the following way. Section II comprises a review of the related work. The problem statement is presented in section III. The proposed methodology is described in Section IV. Section V gives the description of the system architecture and section VI implementation. The gas analysis is performed in section VII. Section VIII discusses results. In Section IX, future directions are made.

II. LITERATURE REVIEW

We surveyed ten representative works in blockchain identity management and e-voting systems have been surveyed by us, and they have been obtained in peer-reviewed journals and in conference proceedings.Table I summarizes the key characteristics, contributions, and limitations of each study.

Taken all together, one may trace the clear pattern of evolution of the discipline within the literature reviewed. At the early stages of development, the main focus of interest was on demonstrating the potential of blockchain technology as an eligible tool for securing voting procedures based on immutability and the communal possibility to check the chain of votes. With the further maturity of the field, the attention of researchers was directed to a less evident problem – ensuring voter privacy. Cryptographic mechanisms including zeroknowledge proofs, homomorphic encryption, blind signatures, and ring signatures were warmly received. DemocracyGuard (available online via DemocracyGuard) and VoteChain (or VoteChain) could be easily implemented on the necessary scale, but these systems rely heavily on the usage of some kind of biometric or central identity verification algorithms, which are related to the sacrifice of privacy.

More attention is being paid now to the issue of SelfSovereign Identity as the ideology of voter verification. Two articles by Baniata and Caluna (2025bpvot) cited below and an article by Jozwik and Pouwelse (2025) argue that SSI, using either the differential privacy approach or zero-knowledge proofs, can ensure high-level anonymity in democratic procedures without affecting the election integrity. Most of these systems are however necessary along with already existing SSI infrastructure like Hyperledger Indy or EUDI Wallets that makes them accessible only within limited resources deployment contexts.

The most frequently raised problem among those presented in literature would be the absence of a thin layer of security that would allow the usage of standard cryptographic functions for establishing an identity verification process without the promotion of any external identity registry at all. In addition, most of the existing systems' employees work within the commit-reveal framework implicitly without even having a proper design of its components. ElectChain addresses both of the problems. The off-chain ECDSA signatures and commitreveal voting protocol implementation in a straightforward way can be tailored for any specific network using solely opensource utilities.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

TABLE I: COMPARISON OF EXISTING BLOCKCHAIN-BASED IDENTITY AND E-VOTING SYSTEMS

Author & Year	Approach	Key Features	Limitations
Liu et al. [1] (2020)	SSI Review on Sovrin, uPort, ShoCard	Blockchain-based identity control; cryptographic security; distributed trust models	Scalability limitations; key management complexity; interoperability issues
Peelam et al. [2] (2024)	Ethereum + Facial Recognition (DemocracyGuard)	Biometric voter authentication; smart contract vote recording; Azure Face API	Large-scale scalability; privacy during biometric auth; energy consumption (PoW)
Hajian et al. [3] (2024)	PRISMA Systematic Review of 252 papers	Security (88.89%), transparency (71.43%); ZKP, homomorphic encryption catalogued	Privacy gaps (42.86%); scalability (34.52%); usability concerns
Daraghmi et al. [4] (2024)	VoteChain: SHA-256 + ECDSA + ZKP	100K concurrent users; 1200 TPS; 99.98% uptime; homomorphic tallying	Regulatory compliance; limited offline support; voter education needs
Le et al. [5] (2025)	AI-OCR + Merkle Tree Identity (BDIMS)	95.1% mAP detection; selective attribute disclosure; 7ms Merkle verification	Computational overhead; quantum resistance; legacy system integration
Seyam & Habbal [6] (2023)	Systematic Review of SSI Architectures	Identifies SSI taxonomy; highlights decentralization and privacy advantages	No implementation; scalability and interoperability unresolved
Shafat-e-Rasool et al. [7] (2025)	Hybrid Blockchain + ZKP + Security Taxonomy	Sybil attack defense; ZKP privacy; regulatory awareness; KYC-based evaluation	Not voting-specific; implementation complexity
Baniata & Caluna [8] (2025)	BP-Vot: Hyperledger Besu + SSI + Differential Privacy	Statistical vote anonymization; SSI decoupling; distributed node evaluation	Honest election authority assumption; partial centralization
Chafiq et al. [10] (2024)	Morocco e-Voting: Solana + DPLT	Contextual feasibility; transparency; immutability	Weak privacy mechanisms; no SSI integration; infrastructure dependencies
Jafar et al. [11] (2021)	Survey of Blockchain EVoting Challenges	Identifies decentralization, nonrepudiation as key properties	Voter privacy unresolved; scalability and throughput gaps

DOI: 10.48175/IJARSCT-32675

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

III. PROBLEM STATEMENT

The existing e-voting solutions come with several issues.

They can be attributed to the structure of the systems themselves and their organizational design. In particular, analyzing the organizational aspect of the issue, it becomes clear that such systems are usually centrally managed. As a result, any failure may cause the entire system to collapse. Moreover, it gives the central authorities too much influence over the outcome of the elections.

In cases when e-voting systems are designed in a decentralized manner, their security cannot guarantee that votes will not be altered. As for identity verification, in most cases, there is always an authority responsible for making sure that voters are who they claim to be. Once again, the decentralized nature of the vote collection procedure does not prevent the centralization of authority.

The problem with retaining the votes is another important matter that should be addressed. Once we adopt blockchains as a medium for collecting votes all the information on how the people voted will become publicly available. It happens due to the transparency of the technology. The process of encoding votes becomes problematic because we need to find a solution which would allow to calculate the results without decoding individual votes. Fortunately, there are methods of addressing this issue, but they are quite complex and require a considerable computing effort. There are other more realistic solutions to this issue but they are rarely implemented in such a way as to ensure proper auditing of the process.

One of the major issues connected with the development of evoting systems is their accessibility. Evoting systems based on the principle of secret coding can only be used by people who have adequate technical skills. This makes them accessible only to certain categories of users while other users may face considerable difficulties with using them. In turn, user-friendly systems usually lack sufficient security measures.

ElectChain is aiming to resolve an exact issue. This is the problem related to the development of a voting platform based on blockchain technology. Such a platform should have certain capabilities. Firstly, it should be possible to perform voter verification while ensuring that no identity manager is responsible for this process. Secondly, it should be ensured that all the votes remain confidential until the time of voting ends. Thirdly, such a platform should ensure that there is no possibility of voter fraud. Lastly, it should be user-friendly to those individuals who are not proficient in technological tools. These conditions should be met through a completely opensource solution. ElectChain is interested in solving this particular problem due to its high relevance. It can be argued that it is vital to develop a voting system which will be secure and easy to use for all users regardless of their technical knowledge. In other words, ElectChain is attempting to solve the problem associated with voter authentication and confidentiality in a blockchain-based evoting platform.

IV. PROPOSED METHODOLOGY

A. Overview

ElectChain has a security architecture that consists of three layers. ElectChain's security architecture ensures segregation of ElectChain's critical components, such as its identity,

privacy, and integrity. The problem solved by each layer of ElectChain's security architecture is a possible loophole that could be exploited. The major advantage of ElectChain's security architecture is that each layer of ElectChain's security architecture can be evaluated, modified, or even substituted individually without disturbing the other layers of ElectChain's security architecture.

B. Identity Layer: Simulated Self-Sovereign Identity via ECDSA

ElectChain does not employ technologies such as Hyperledger Indy or a registry of Self-Sovereign Identities. In its stead, ElectChain employs an alternative methodology that accomplishes similar aims using ECDSA Signatures, which are generated independently of the system itself.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

For each eligible voter, two sets of keys are generated. One key will be public; one will be private. The public key serves as the alias of the user that is not his or her own name. This public key is submitted to the ElectionAuthority upon registration as a valid voting user.

A voter decides which candidate to choose by signing a secret code (the choice) with their key. The ElectionAuthority smart contract verifies the signature with the help of the method ecrecover of the Ethereum system. In this way, the smart contract can verify the signature while having no clue about what was chosen. ElectionAuthority and Smart contracts work together to ensure the security and privacy of the voting process for every voter.

It is a successful design since it complies with the properties listed above in the core of Self Sovereign Identity concepts. The voter has a credential – a private key, and they control it. Voter's personal data is not written to the blockchain; instead, the user's eligibility for voting is verified with cryptographic means.

On the other hand, developers specifically designed this system not to fulfill all Self Sovereign Identity requirements. No such features as verifiable credentials and proofs of identity documents were implemented in the design. Such approach simplifies its functioning but limits its usability in certain scenarios. This project would be useful for demonstrations and implementation in an educational environment or organizational elections when all participants are known.

C. Privacy Layer: Commit-Reveal Scheme

Commitment to the voter's ballot can be made by using a two-step commit-and-reveal algorithm. The first step involves generating the commitment, which is done by computing C = keccak256(candidate id‖nonce), where *nonce* is a secret that only the voter knows. This commitment is then published by Voting.sol. At this point, nothing about the candidate has been revealed.

Following the expiration of the voting period and moving of the contract to the reveal stage, the voter sends the candidate ID and the nonce together. The contract verifies whether the hash of the candidate ID and nonce corresponds to a certain value and, if all parameters are verified successfully, includes the vote for the appropriate candidate's tally.

Such an algorithm has several advantages regarding security.

Votes remain anonymous during the commit phase due tothe computational binding of the commitment.

Nobody is able to predict the outcome until the entireprocess of voting ends because of unavailable partial counts.

This is possible due to the absence of any partial tallies.

The integrity of the votes remains during the reveal phase.Any modification will lead to failing to verify the hash.

D. Integrity Layer: Foundry-Based Smart Contracts

The election procedure is outlined by the election process in two separate Solidity contracts. The two contracts were compiled and tested in the Foundry framework. The first contract, named ElectionAuthority.sol, oversees the various phases of an election. These include registration periods, voting periods, when voters show their ballots and when counting is performed. It also deals with adding new candidates, transitioning from one phase to another, among other things.

The second contract, namely Voting.sol, guarantees voter anonymity, storing the votes cast, verifying the votes as valid when casting, making sure each voter casts only one vote and tallying the votes per candidate.

When any action occurs in the course of the election, an event is triggered on the Ethereum blockchain, which becomes an immutable ledger of what happened. This ledger can be accessed by anyone to verify its contents.

V. SYSTEM ARCHITECTURE

A. Architectural Overview

ElectChain has four interdependent modules that include the voter client, Anvil, the smart contract module, and the off-chain signing module. Figure 1 illustrates the system architecture.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

B. Component Interactions

Client voter interactions with the Anvil node occur via Wagmi and Viem modules, which translate raw JSON-RPC requests into well-defined TypeScript types. Wallet communication is managed by MetaMask; this service stores the private keys and performs local signatures of transactions before transmitting them to the Anvil node. Contracts' behavior is deterministic, executing on the Anvil Ethereum Virtual Machine.

Signing of transactions is completed off-chain via the ethers.js cryptography suite provided by the Viem module. Signature information becomes input data for the registration contract method invocation, thus not requiring persistent storage on the blockchain.

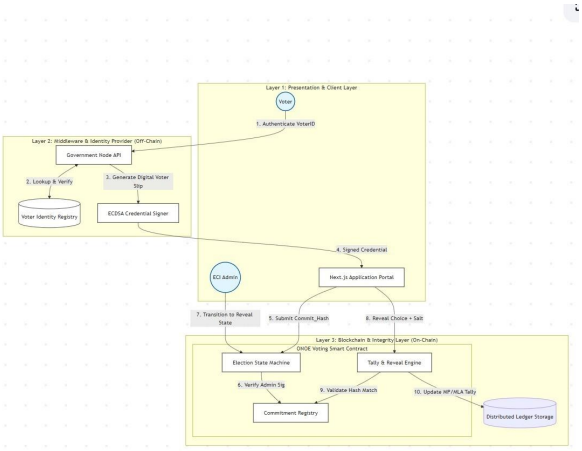

Fig. 1. System Architecture of ElectChain

C. Data Flow

The lifecycle of an election is carried out in four sequential steps. In the Registration step, the election administrator uses contracts to register candidates and voters' public keys. The Voting stage sees voters calculate commitments and upload them using signed transactions. Once the deadline elapses, the administrator initiates the Reveal step where voters upload their plaintext votes for validation. Lastly, the tally step takes the results and displays them on the frontend results page.

VI. IMPLEMENTATION DETAILS

A. Smart Contract Development with Foundry

The contracts are developed and tested with Foundry, an extremely fast Solidity development environment implemented in Rust. The folder structure conforms to the conventional Foundry structure that uses src/, test/, and script/ directories.

Voting.sol defines mappings for voter commitments
(mapping(address => bytes32) public commitments), reveal status (mapping(address => bool) public hasRevealed), and candidate vote tallies (mapping(uint256 => uint256) public voteTallies). The commit(bytes32 commitment) function records a commitment and emits a VoteCommitted event; the reveal(uint256 candidateId, bytes32 nonce) function verifies the commitment and increments the tally.

ElectionAuthority.sol enforces role-based access through an onlyAuthority modifier, maintains an enumerated ElectionState, and exposes administrative functions
(startVoting(), startReveal(), finalize()) that transition the election through its lifecycle.

B. Local Blockchain with Anvil

The local Ethereum test network is launched using the following command:

DOI: 10.48175/IJARSCT-32675

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

Anvil comes with 10 prefunded accounts each containing 10,000 ETH, deterministic block creation, and immediate transactions, making it perfect for testing and measuring gas costs.

C. Contract Deployment with Forge

Contracts are deployed using the forge create command:

forge create --rpc-url http://127.0.0.1:8545 \
--private-key <DEPLOYER_KEY> \
Interfac src/ElectionAuthority.
sol:ElectionAuthority
forge create --rpc-url http://127.0.0.1:8545 \
anvil --block-time 1 --chain-id 31337 --private-key <DEPLOYER_KEY> \ --constructor-args <AUTHORITY_ADDR> \ src/Voting.sol:Voting

The deploy script retrieves contract addresses and saves them in the config file that is consumed by the Next.js frontend.

D. Frontend with Next.js, Wagmi, and Viem

This project uses Next.js 16 framework to implement the voter's interface with App Router architecture. Wagmi v3 offers hooks to connect wallet (useConnect), read data from contracts (useReadContract), and write transactions (useWriteContract). Viem v2 library allows working with ABI-encoded transactions on Ethereum. Styling is performed with TailwindCSS v4.

In terms of features, the application implements three main views:

Wallet Connect view where voters can set up the MetaMask extension, register a public address and vote.

Voting view, where the user can compute their commitment locally and send a confirmed signed transaction.

Results view, which shows candidate scores retrieved from Voting.sol contract.

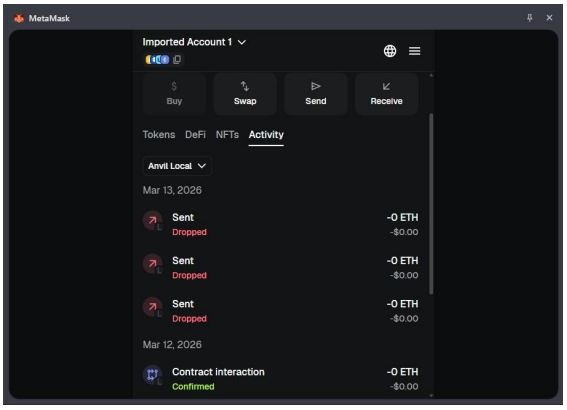

Fig. 2. Wallet Connection Page — ElectChain Frontend

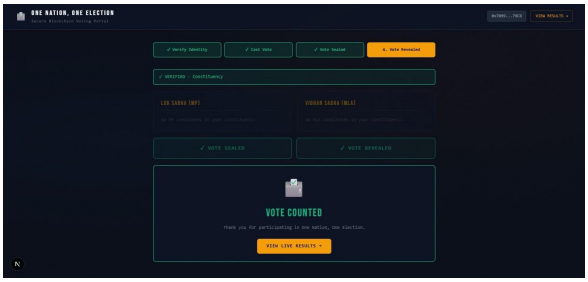

Fig. 3. Voting Page with Commit-Reveal

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

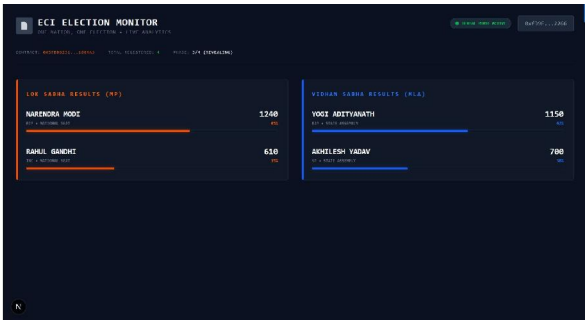

Fig. 4. Results Page Displaying Live Vote Tallies

VII. EXPERIMENTAL EVALUATION: GAS ANALYSIS

A. Gas in Ethereum

Every computation on the Ethereum Virtual Machine (EVM) uses a metered amount known as *gas*. The use of gas is intended as a way to prevent spam and as a pricing model for computations such that any computational operation is priced according to its computational complexity and storage size. There is a predetermined gas cost for every opcode: the most costly is storing a value to memory (SSTORE), costing 20,000 gas when adding a new entry, while arithmetic operations require a small number of gas. The product of the amount of gas used and the current *gas price* (in Gwei) will give the sender's transaction fee.

When developing a smart contract for an e-voting system, it is important to consider gas optimization. High per vote transaction fees may make the voting system economically impractical, and the use of gas-inefficient contracts could cause the contracts to run out of gas during election times with high voter turnout.

B. Gas Consumption Results

Table II presents the measured gas costs for the primary smart contract operations in ElectChain.

C. Optimization Strategies

Some of the choices that enable this gas efficiency include avoiding the storage of any personally identifiable data of voters in the blockchain; storing only cryptographic commitments (32-byte hash) and public keys (20-byte

TABLE II:

Function	Gas Used	Approx. Cost (Gwei)
registerVoter()	46,823	~0.047
commit(bytes32)	44,512	~0.045
reveal(uint256, bytes32)	38,947	~0.039
startVoting()	29,318	~0.029
startReveal()	29,104	~0.029
finalize()	31,762	~0.032
Total per voter	~130,282	~0.131

GAS CONSUMPTION FOR CORE ELECTCHAIN OPERATIONS

Ethereum address). Secondly, using the mapping structure for vote storage instead of the array structure which requires resizing whenever the vote array expands. Using the keccak256 opcode offered by the Solidity compiler instead of more expensive hash computations. Lastly, substituting expensive in-contract logging mechanisms with the cheaper event mechanism (emit). A cost of about 130,000 gas for each voter is efficient in comparison to other similar

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 4, April 2026
Impact Factor: 8.2

solutions. With a gas fee of 10 Gwei per unit and an Eth cost of $3,000 at Ethereum mainnet, the gas cost amounts to approximately $0.004 per voter.

VIII. RESULTS AND DISCUSSION

A. Security Analysis

Preventing Double-Voting. The Voting.sol contract maintains the invariant property that an address may submit one commitment only, failing any further attempts to do so with a descriptive error. As all voters are uniquely assigned Ethereum addresses, associated with their ECDSA key pairs, this condition makes doublevoting impossible.

Resistance to Vote Manipulation. Using the commitreveal technique guarantees that any vote once committed cannot be changed later on. Indeed, any change to the candidate ID or nonce will result in a different hash value, which will not be recognized by the smart contract. All commits are unalterable on the Anvil blockchain. Hiding the Results During the Voting Phase. Since the smart contract records hashes of commitment values only in the voting stage, vote distribution cannot be computed. An attacker, even if he or she manages to access the blockchain state during the voting process, will have to resolve a preimage problem in order to get the desired information.

Preserving Voter Privacy. There are no voter-related data stored on the blockchain. Voter addresses are anonymized and the connection between them and their owners is possible only off-chain.

B. Comparative Evaluation

Table III compares ElectChain against three representative prior systems.

TABLE III

System	SSI Support	Vote Privacy	Gas Opt.	Open Source
Democracy Guard [2]	No	Low	Medium	No
VoteChain [4]	Partial	High	Medium	No
BP-Vot [8]	Yes	High	Low	No
ElectChain	Simulated	High	High	Yes

COMPARATIVE ANALYSIS OF E-VOTING SYSTEMS

The ElectChain system offers a very high level of vote privacy by using the commit-reveal method and performs well regarding gas consumption, thanks to its efficient use of storage, as mentioned in Section VII. Even though it uses simulation for SSI implementation, the ECDSAbased method fulfills the core aspect of SSI – selfmanaged credentials.

C. Limitations

The present deployment runs on a local Anvil network that lacks the latency and bandwidth behavior of a public blockchain. The simulation of SSI does not generate W3C-verifiable DID documents and credentials, thereby making it difficult for integration with SSI ecosystems. Another challenge arises from the fact that there is a need for a second engagement by the voter during the reveal stage of the commit-reveal approach.

IX. CONCLUSION AND FUTURE WORK

ElectChain was proposed as a decentralized e-voting platform that employs a blockchain system to implement off-chain simulated self-sovereign identity using ECDSA signatures, a privacy-preserving commit-reveal scheme, and Foundry-compiled Solidity smart contracts. This evoting scheme tackles some of the core issues involved in e-voting, which include centralized identity management, voter privacy, double voting, and result manipulation. The system is shown by gas optimization analysis to run within the boundaries of the EVM limitations, while a comparative study reveals its superiority over existing approaches.

Future Work. There are several enhancements that we aim to make to fully bring ElectChain into a live environment. To begin with, we seek to deploy authentic W3C DID documents and verifiable credentials based on the did:ethr

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 4, April 2026 Impact Factor: 8.2

scheme, thus achieving full compatibility with SSI technologies and existing identity systems like Sovrin and the EU Digital Identity Wallet. In addition, we aim to introduce the capability of zero-knowledge proofs, more specifically zk-SNARKs utilizing the Circom/SnarkJS framework, thereby switching from the commit-reveal scheme to an even stronger cryptographic technique, which would no longer require interaction in the reveal phase. Moreover, scalability tests on various public testnets (Sepolia, Holesky) and Layer-2 solutions (Optimism, Arbitrum) would demonstrate how well the protocol performs under actual conditions with regard to voter load, and the gas fees should also be recalculated in these conditions. In addition, the code of smart contracts could be audited with the help of such tools as Slither and Echidna. Finally, integrating the solution with national identification services by implementing selective disclosure credentials would allow deploying ElectChain as an infrastructure for governmental election systems.

REFERENCES

[1]. Y. Liu, D. He, M. S. Obaidat, N. Kumar, M. K. Khan, and K. K. R. Choo, "Blockchain-based identity management systems: A review," *Journal of Network and Computer Applications*, vol. 166, p. 102731, 2020.

[2]. M. S. Peelam, G. Kumar, K. Shah, and V. Chamola, "DemocracyGuard: Blockchain-based secure voting framework for digital democracy," *Expert Systems*, vol. 42, no. 2, p. e13694, 2024.

[3]. M. Hajian Berenjestanaki, H. R. Barzegar, N. El Ioini, and C. Pahl, "Blockchain-based e-voting systems: A technology review," *Electronics*, vol. 13, no. 1, p. 17, 2024.

[4]. E. Daraghmi, A. Hamoudi, and M. Abu Helou, "Decentralizing democracy: Secure and transparent e-voting systems with blockchain technology in the context of Palestine," *Future Internet*, vol. 16, no. 11, p. 388, 2024.

[5]. H. V. A. Le, Q. D. N. Nguyen, T. Nakano, and T. H. Tran, "Blockchain-based decentralized identity management system with AI and Merkle Trees," *Computers*, vol. 14, no. 7, p. 289, 2025.

[6]. H. Seyam and A. Habbal, "A systematic review of blockchainbased identity management solutions," in *Proc. International Conference on Recent Academic Studies*, 2023.

[7]. Shafat-e-Rasool et al., "Comprehensive review of blockchainbased decentralized identity verification and management systems," *Journal of Emerging Technology and Digital Transformation*, 2025.

[8]. H. Baniata and G. Caluna, "BP-Vot: Blockchain-based e-voting using smart contracts, differential privacy, and self-sovereign identities," *IEEE Access*, vol. 13, 2025.

[9]. M. Jozwik and J. Pouwelse, "SmartphoneDemocracy: Privacypreserving e-voting on decentralized infrastructure using novel European identity," arXiv:2507.09453, 2025.

[10]. T. Chafiq, R. Azmi, and O. Mohammed, "Blockchain-based electronic voting systems: A case study in Morocco," *International Journal of Intelligent Networks*, 2024.

[11]. U. Jafar et al., "Blockchain for electronic voting system — review and open research challenges," *Sensors*, vol. 21, no. 17, p. 5874, 2021.

[12]. D. Castellano, R. De Prisco, and P. Faruolo, "Toward a compliant token-based e-voting system with SSI-granted eligibility," in *Proc. SECRYPT 2023*, 2023.