HYBRID E-VOTING: INTEGRATING HOMOMORPHIC ENCRYPTION AND DLT FOR POLARIZED SCENARIOS

Furio Ruggiero

envLab, Venezia, Italy

ABSTRACT

E-voting in polarized contexts requires a strict balance between public verifiability, ballot secrecy, and coercion resistance. Traditional centralized systems lack transparency, while fully decentralized models face scalability and privacy issues. This paper proposes a hybrid architecture compliant with OSCE/ODIHR standards [1] for low-trust environments. The protocol decouples identity from voting an off-chain Oracle manages authorization via cryptographic tokens, while the Waves DLT acts as an immutable bulletinboard.Utilizinghomomorphicencryption[2],Zero-KnowledgeRangeProofs(ZKRP) [3], and Distributed Key Generation (DKG) [4], the system ensures End-to-End Verifiability (E2E) by delegating tallying to auditable scripts. Finally, the study examines model limitations, specifically regarding endpoint vulnerabilities and physical constraints on coercion resistance.

KEYWORDS

E-Voting, Distributed Ledger Technology, Homomorphic Encryption, End-to-End Verifiability, Zero-Knowledge Proofs

1. INTRODUCTION

The transition from traditional paper-based electoral processes to electronic voting systems represents one of the most complex technological and democratic challenges of the digital age [5]. In ordinary electoral contexts, logistical efficiency is often the primary driver for E-Voting adoption. However, in highly polarized scenarios—where the outcome of the vote can radically alter institutional or territorial frameworks—security requirements become extreme. In such contexts, distrust between parties or toward the central authority demands that the system be mathematically unassailable and publicly verifiable

Council of Europe recommendations [6] and OSCE/ODIHR guidelines establish that an E-Voting system must simultaneously guarantee secrecy, integrity, verifiability, and coercion resistance. Achieving these objectives simultaneously introduces complex cryptographic paradoxes: how can a system prove to a citizen that their vote was counted correctly without allowing them to prove to third parties how they voted (thus preventing vote-buying)?

1.1 The Architectural Tension: Limitations of the "Fully On-Chain" Approach

In an attempt to eliminate the need for trust in a central authority, recent literature has focused on the use of Smart Contracts on blockchain networks (such as Ethereum) to manage the entire lifecycle of an election [7], [8]. Although conceptually elegant, fully on-chain architectures clash with technical and economic realities.

First, executing complex calculations (such as validating thousands of zero-knowledge proofs or iterating the tally) within a Turing-complete Smart Contract incurs unsustainable transaction

DOI:10.5121/ijci.2026.150201

costs (Gas) at scale. Second, and more critically for polarized scenarios, the on-chain registration of identities (voter address whitelists) exposes citizens to severe risks of de-anonymization and social retaliation. Even if the vote payload is encrypted, the metadata (the participation of a specific address in the vote) remains permanently recorded on a public ledger.

1.2 Contribution: A Hybrid and Asymmetric Model

To overcome these limitations, this paper proposes a hybrid and asymmetric architecture. The protocol acknowledges that the native languages of highly scalable blockchains (e.g., RIDE on the Waves network) are intentionally Turing-incomplete and unsuitable for implementing complex algorithms like homomorphic tallying.

Therefore, the system delegates identity management and authorization to a traditional off-chain infrastructure, relegating the Smart Contract to the exclusive and passive role of a Public Bulletin Board. The Smart Contract merely validates authorization tokens and stores the "encrypted envelopes" (ciphertexts). The intensive tally computation is performed off-chain by open-source scripts exploiting the homomorphic property, allowing anyone to verify the count.

2. State Of The Art And Related Work

Research on cryptographic E-Voting protocols has historically sought to balance two seemingly opposing requirements: ballot secrecy and public verifiability.

The turning point in the literature is represented by End-to-End (E2E) Verifiable systems, notably the seminal implementation of Helios [9]. Helios demonstrated the feasibility of a web-based voting system where the outcome is calculated via homomorphic encryption, enabling public auditing without decrypting individual ballots. However, in its original form, Helios does not offer coercion resistance and relies on a central server for ciphertext collection.

To address coercion, Juels, Catalano, and Jakobsson [10] formalized the concepts of Coercion-Resistance and Receipt-Freeness, proposing the use of dummy credentials or allowing the voter to overwrite a coerced vote. The advent of Distributed Ledger Technology (DLT) offered a natural solution for the public bulletin board. The implementation of the Open Vote Network on Ethereum [7] represented the first fully on-chain academic protocol. However, that study highlights critical limitations regarding computational scalability (Gas limits) and voter privacy due to the rigid link between identity and the blockchain transaction.

3. THREAT MODEL AND ARCHITECTURAL REQUIREMENTS

To evaluate robustness in scenarios of severe institutional tension, the protocol assumes an extended Dolev-Yao Threat Model in which attackers possess state-level resources and may collude with the service provisioning infrastructure.

3.1 Hostile Actors

The system must withstand three specific categories of attackers

1. **The Insider Attacker (Compromised Server):** It is assumed that the central server (Identity Oracle) or frontend software may be compromised.

2. **The Coercer (State or Non-State Actor):** An attacker who can physically, economically, or socially threaten the voter, demanding they vote in a certain way under supervision or requesting a receipt afterward.
3. **The Network Attacker (Denial of Service):** An actor capable of flooding the network or selectively censoring regional internet traffic.

3.2 Security Requirements

1. **Universal E2E Verifiability:** Anyone must be able to mathematically recalculate the result using DLT data (Cast-as-intended and Counted-as-cast)..
2. **Computational Privacy:** No actor may derive the plaintext vote from the ciphertext.
3. **Coercion Resistance (Silent Overwrite):** The system must allow a voter to cast a coerced vote that appears valid to the coercer, while providing a secure window to silently overwrite it.
4. **Trust-Minimization:** Cryptographic key generation must never occur at a single point of failure.

4. Proposed Architecture

The protocol is structured into five fundamental phases as illustrated in Figure 1.

4.1 Electoral Setup and Distributed Key Generation (DKG)

To prevent the existence of an Insider Attacker, the election's cryptographic key pair is not generated by a central server. Instead, a Distributed Key Generation (DKG) protocol based on Threshold Cryptography is employed. A set of N independent nodes (Trustees) generates a single Election Public Key **PK_el**. The corresponding Private Key **SK_el** is never assembled but remains fragmented among the Trustees.

4.2 Off-Chain Authorization and Identity Decoupling

1. The voter authenticates on an institutional portal.
2. The ministerial server verifies voting eligibility and generates a temporary Authorization Token containing the hash of the voter's temporary DLT address, digitally signed with the Oracle's private key.
3. The Oracle sends the token to the user's wallet (the APP). The central database knows who is authorized but remains oblivious to the vote's content..

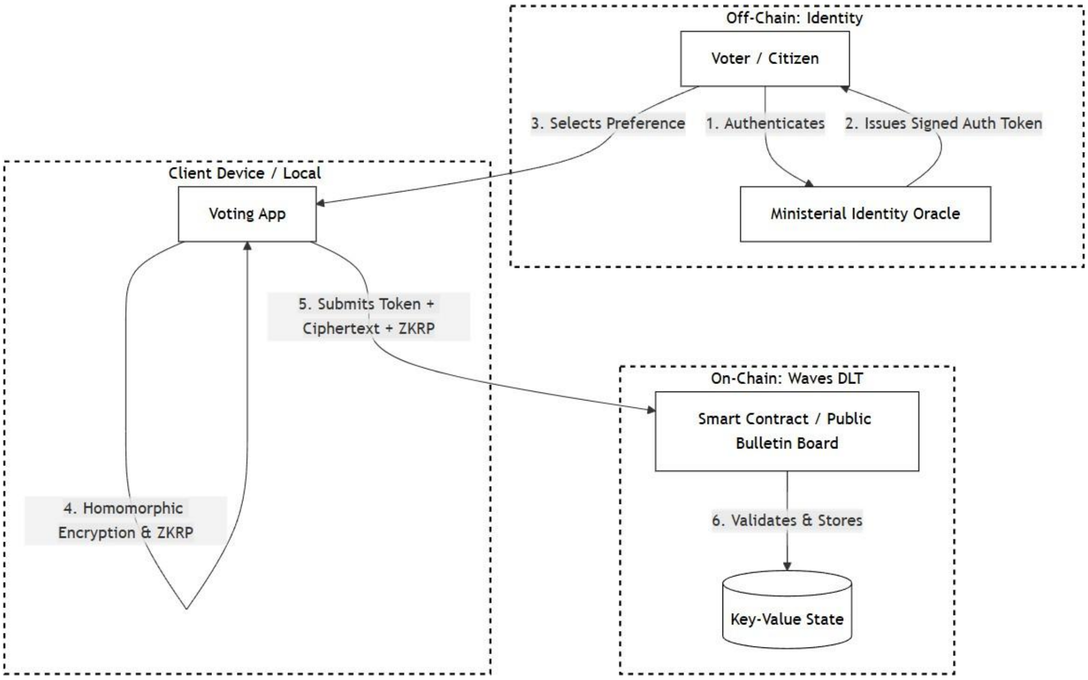
Off-Chain: Identity
Voter / Citizen
3. Selects Preference
1. Authenticates
2. Issues Signed Auth Token
Ministerial Identity Oracle
Client Device / Local
Voting App
5. Submits Token + Ciphertext + ZKRP
4. Homomorphic Encryption & ZKRP
On-Chain: Waves DLT
Smart Contract / Public Bulletin Board
6. Validates & Stores
Key-Value State

Figure 1: High-Level Architecture and Data Flow of the Hybrid E-Voting Protocol

4.3 Ballot Casting (Client-Side Encryption)

All encryption logic occurs locally on the user's device. The voter selects a preference v ∈ {0,1}.
.

1. Homomorphic Encryption: The wallet (APP) encrypts the choice using PK_el, outputting the ciphertext C = Enc(PK_el, v).
2. Zero-Knowledge Range Proof (ZKRP): To prevent malicious payload injection, the wallet generates a non-interactive mathematical proof π attesting that C strictly contains either 0 or 1, without revealing which

.

4.4 On-Chain Validation and State Sharding (Waves DLT)

The wallet submits the transaction to the Waves network, attaching the signed Token, ciphertext C, and proof π. The Smart Contract acts as a passive validator::

1. Verifies the Oracle's signature on the Token.
2. Verifies the mathematical correctness of the ZKRP.
3. Anti-Coercion via Key-Value Overwrite: Saves the vote in the account state using the Token hash as the "Key" and the vote as the "Value". If a coerced voter submits a subsequent transaction, the DLT's native key-value dictionary silently overwrites the previous entry (last-vote-counts).

To bypass the computational complexity limits of the RIDE language at scale, the architecture implements State Sharding [11]. A Coordinator Account routes transactions to multiple Storage Accounts (shards), while the ciphertext and proof are compressed (Key Packing) to minimize state lookup overhead..

4.5 Off-Chain Homomorphic Tally and E2E Verifiability

After the polls close, the tallying is delegated to independent off-chain script s:

1. Observers download the Data Entries from the Storage Accounts.
2. Exploiting homomorphic additivity, ciphertexts are mathematically multiplied among themselves [12]: ∏ Enc(PK_el, v_i) = Enc(PK_el, ∑ v_i). The result is a single Encrypted Tally (C_Tally).
3. Trustees use their key shares to jointly decrypt only C_Tally, revealing the aggregate result without decrypting individual preferences, and emit a Zero-Knowledge Proof of correct decryption.

5. Security And Compliance Analysis

- Privacy: Guaranteed cryptographically before the data leaves the device. A compromised backend only yields authorization metadata, not ballot contents.
- E2E Verifiability: The public DLT bulletin board sterilizes post-vote censorship capabilities.
- Coercion Limits: The key-value dictionary mitigates standard coercion (silent overwrite). However, this degrades if the coercer exercises uninterrupted physical constraint until the voting window closes.
- Endpoint Vulnerability: If the voter's device is infected by state-sponsored malware, it could encrypt an unintended vote. Mitigation requires hardware-isolated wallets.

6. Implementation Considerations And Scalability (Waves DLT)

Empirical experience on Waves networks demonstrates that the primary bottleneck is not raw storage space (~5 MB per account) but computational cost (Ride script complexity).

To resolve this, the architecture implements:

- State Sharding (Coordinator Pattern): Distributing the flow across multiple Storage accounts.
- Off-Chain Data and Minimal Indexing: Using the blockchain strictly for notarization.
- Key Packing: Compressing the ciphertext and ZKRP into a single binary string.

Figure 2 demonstrates how state sharding work

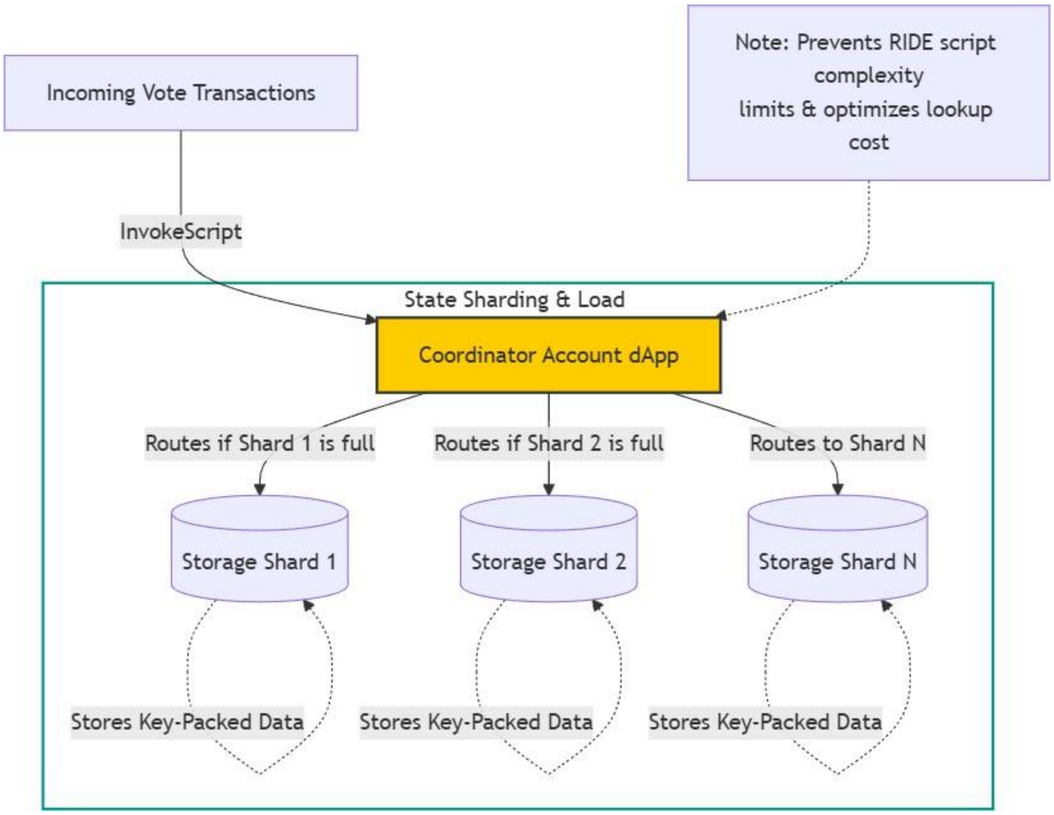

Figure 2: State Sharding Mechanism on Waves DLT via Coordinator and Storage Accounts

7. COMPARATIVE ANALYSIS WITH EXISTING E-VOTING SOLUTIONS

To properly contextualize the proposed hybrid architecture, it is essential to compare it with both traditional centralized e-voting systems [9] and pure on-chain Distributed Ledger Technology (DLT) solutions [7], [8]. Traditional centralized platforms often suffer from the "black-box" paradigm, where voters must blindly trust a central authority for both privacy and tallying integrity. Conversely, pure on-chain voting systems—where smart contracts handle both the voter registry and the tallying logic—face severe scalability bottlenecks [11] and struggle to ensure absolute ballot secrecy, as on-chain data is inherently public and smart contract execution is heavily constrained by network fees.

The proposed hybrid model overcomes these limitations through "Identity Decoupling" and client-side Homomorphic Encryption [2]. By delegating identity verification to an off-chain Oracle and utilizing the DLT strictly as an immutable, sharded public bulletin board for encrypted payloads and Zero-Knowledge Proofs (ZKRPs) [3], the architecture achieves a unique balance. Table 1 summarizes the core differences and highlights the competitive advantages of the proposed approach.

Table1.ComparativeAnalysisofE-VotingArchitectures

Feature	Traditional Centralized Systems	Pure On-Chain DLT Systems	Proposed Hybrid Architecture
Trust Model	Blind trust in central authority	Trustless(Code is Law)	Trustless(E2E Verifiable)
Voter Privacy	Vulnerable to insider attacks	Pseudonymous, risk of deanonymization	Absolute (Homomorphically encrypted)
Scalability	High	Very Low(Smart contract limits)	High(Off-chain identity + DLT Sharding)
Tally Verifiability	Closed/Private Auditing	Public	Public(Zero-Knowledge Proofs)
Coercion Resistance	Poor	Poor	High (Multiple voting with override)

8. CONCLUSION

This paper presents a hybrid E-Voting architecture designed for highly polarized electoral scenarios. By decoupling identity (off-chain) from ballot registration (on-chain), the protocol circumvents the severe economic and privacy limitations of fully on-chain systems. The use of the Waves blockchain as a sharded Public Bulletin Board, combined with homomorphic encryption, delivers total mathematical transparency without sacrificing individual secrecy, offering a robust foundation for modernizing democratic processes in low-trust environments

8.1 Future Research Directions

While the proposed hybrid architecture successfully addresses the complex trilemma of privacy, verifiability, and scalability in highly polarized electoral scenarios, future research will focus on large-scale field testing. A primary operational direction involves integrating the off-chain identity Oracle with emerging sovereign digital identity frameworks, such as the European Digital Identity (eIDAS 2.0) infrastructure, to further streamline voter on boarding. Furthermore, as global computational power advances, future iterations of this architecture will investigate the transition from current elliptic curve cryptography to post-quantum cryptographic primitives. This evolution will be crucial to guarantee the long-term systemic resilience and unconditional secrecy of the cryptographic voting records against future quantum computing threats.

9. REFERENCES

[1] OSCE/ODIHR, "Handbook for the Observation of New Voting Technologies," 2013.

[2] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," European Transactions on Telecommunications, 1997.

[3] Ekbatanifard and F. Ekbatanifard, "Zero-Knowledge Proofs for Secure and Private Voting Systems," International Journal of Academic and Industrial Research Innovations (IJAIRI), 5(6), 212-214, 2025.

[4] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure Distributed Key Generation for Discrete-Log Based Cryptosystems," 1999.

[5] Alqahtani et al., "Blockchain-Based E-Voting Systems: A Technology Review," Electronics (MDPI), 13(1), 17, 2024.

[6] Council of Europe, "Recommendation CM/Rec(2017)5 of the Committee of Ministers to member States on standards for e-voting," 2017.

[7] P. McCorry, S. F. Shahandashti, and F. Hao, "A Smart Contract for Boardroom Voting with Maximum Voter Privacy," Financial Cryptography and Data Security, 2017.

[8] F. Þ. Hjálmarsson, G. K. Hreiðarsson, M. Hamdaqa, and G. Hjálmtýsson, "Blockchain-Based E-Voting System," IEEE 11th International Conference on Cloud Computing, 2018.

[9] B. Adida, "Helios: Web-based Open-Audit Voting," USENIX Security Symposium, 2008.

[10] Juels, D. Catalano, and M. Jakobsson, "Coercion-Resistant Electronic Elections," Workshop on Privacy in the Electronic Society, 2005.

[11] P. Kumar et al., "A Federated AI and DAG-Based Framework for Secure and Scalable E-Voting," Proceedings of the 4th International Conference on Information Technology, Civil Innovation, Science, and Management (ICITSM 2025), EAI, 2025.

[12] W. Zhang et al., "Revolutionizing Voting Security Using Blockchain," International Journal of Innovative Research in Technology (IJIRT), 11(6), 1929-1932, 2024.

10. APPENDIX: TECHNICAL GLOSSARY

- DLT (Distributed Ledger Technology): A decentralized database managed by multiple participants. Preferred in academia over "blockchain" for neutrality.
- E2E Verifiability (End-to-End Verifiability): The mathematical guarantee allowing voters to check their ballot is cast as intended, and anyone to verify all ballots are counted as cast.
- Homomorphic Encryption: A cryptographic algorithm permitting mathematical operations (e.g., addition) on ciphertexts without prior decryption.
- DKG (Distributed Key Generation): A protocol where multiple nodes collaboratively generate a public key, while the private key remains fragmented to prevent single-point compromise.
- ZKRP (Zero-Knowledge Range Proof): A cryptographic proof demonstrating a ciphertext contains a valid value (e.g., strictly 0 or 1) without revealing the value itself.
- State Sharding: A scalability technique that fragments a large database into smaller, manageable parts (shards) to prevent network computational overload.
- WALLET: In this document, wallet and app are synonymous. It is assumed that the user has a single-page web application to interact with.