International Journal of Natural-Applied Sciences and ENgineering
(IJNASEN)
Vol.4 No.1 (2026) pp. 23–30

Research Article

BAVS-ZK: Hybrid Entropy-Based Credential Derivation for Anonymous Blockchain E-Voting

Hawraa M. Ali[1], Ra'ad A. Muhajjar[2]

[1]Department of Computer Science University of Basrah, Basrah, Iraq
* **Corresponding Author Email:** hawraa.ali@uobasrah.edu.iq- **ORCID:** 0000-0002-5247-7660

[2]Department of Computer Science University of Basrah, Basrah, Iraq
Email: raad.muhajjar@uobasrah.edu.iq- **ORCID:** 0000-0002-5247-0050

Abstract: Blockchain-based electronic voting systems that use zero-knowledge proofs (ZKPs) have been proposed as good candidates to provide both transparency and privacy of ballots. However, a fundamental challenge remains unmet in all existing schemes: the secure generation and protection of the voter's cryptographic secret key.In this paper, HME-KG (Hybrid Multi-Source Entropy Key Generation) is presented, a new credential derivation method which utilizes a cryptographically secure random salt, the national identity number of the voter and a per-device Client Device Secret (CDS) to derive a deterministic, brute-force-resistant secret key. HME-KG is integrated into BAVS-ZK, a complete anonymous blockchain voting framework employing AES-256-GCM encrypted credential storage, a Circom-based Groth16 zk-SNARK voting circuit, and on-chain nullifier verification via Ethereum Sepolia smart contracts. Security analysis demonstrates that HME-KG achieves voter determinism, cross-voter uniqueness, single-source failure resistance, and collision resistance under the security assumptions of SHA-256. Experimental evaluation on a 10,000-voter simulation confirms a 0.9998 scalability coefficient, 1.2-second proof generation, and 306,720 gas per vote—a 38.6% reduction compared to the Open Vote Network baseline. To the extent of current literature, BAVS-ZK is the first blockchain e-voting system to provide a complete, formally specified, and experimentally validated voter credential derivation and protection scheme.

Keywords: Electronic voting; Blockchain; Zero-Knowledge Proofs; zk-SNARK; Groth16; HME-KG; Client Device Secret.

Received: *07 January 2026* **|** ***Revised:*** *15 April 2026* **|** ***Accepted:*** *20 April 2026* **|** ***DOI:*** *10.22399/ijnasen.37*

1. Introduction

Electronic voting (e-voting) systems must satisfy two properties that are fundamentally in tension: every cast ballot must be publicly verifiable to ensure election integrity, yet the content of each individual ballot must remain confidential to prevent coercion and protect voter autonomy [1]. Paper-based elections have physical controls to enforce ballot integrity, but these result in slow tabulation and a lack of accessibility [2]. Blockchain technology provides a structural answer to the verifiability aspect of this tension via an append-only, publicly verifiable ledger [3]. Election laws may be enforced automatically via the Ethereum's smart contracts [4]. However, ballot secrecy is at odds with public transparency, as every transaction is permanently visible on the blockchain [5]. Zero-knowledge proofs (ZKPs) [6] address this tension by enabling a voter to prove their eligibility and the correctness of their ballot without revealing their identity or their vote choice. Using the Groth16 protocol [7], which is supported by EIP-197 [8] and EIP-1108 [9] precompiles on Ethereum, the proof sizes were brought down to under 200 bytes, with a three-pairing verification, in contrast with the earlier method, where every proposition required a linear number of pairings. However, a critical gap remains: the creation and protection of the voter's secret key. This key needs to be unique for each voter, unpredictable, deterministically reproducible, and resilient to database leaks. None of the current systems, including the very recent 2025 proposals [10], [11], specify how such a key is

derived, what entropy sources are used, or how it is securely stored. This paper contributes: (1) HME-KG, a hybrid multi-source entropy key generation scheme combining the voter's national identity number, a cryptographically secure random salt, and a per-device Client Device Secret (CDS), providing determinism, uniqueness, and single-source failure resistance; (2) integration with AES-256-GCM credential encryption under an externally stored master key; (3) BAVS-ZK, a complete voting system integrating HME-KG, a Circom voting circuit, Groth16 proof generation, and on-chain verification; (4) experimental validation through unit testing, a 10,000-voter simulation, and live Sepolia transactions.

2. Related Works

The Open Vote Network (OVN) [12] was the first self-tallying boardroom voting protocol on Ethereum, achieving voter privacy through Schnorr ZKPs at approximately 500,000 gas per voter—the gas efficiency baseline. Neither credential derivation nor storage is addressed.

Panja and Roy [13] combined blockchain with NIZKs to eliminate tallying authorities without specifying credential management. Alam and Joshi [14] employed Paillier homomorphic encryption with partial off-chain verification. Marcellino et al. [15] proposed hybrid ZK-SNARK identity authentication focused exclusively on eligibility verification.

Park et al. [16] introduced zkVoting, a coercion-resistant system using nullifiable commitments with a trusted registrar. Rabia et al. [17] demonstrated Groth16 on-chain integration at proof-of-concept scale without key management. The latest 2025 submissions—Sangraula and Adhikari [10], and ZKHC-Voting [11]—confirm the feasibility of ZKP but do not define the derivation or protection of voter credentials. Table 1 shows that no prior system published through 2025 specifies a voter credential derivation or protection mechanism, the gap directly addressed by HME-KG.

Table 1. *Comparison of Related E-Voting Systems*

System	Year	Cred. Deriv.	Cred. Prot.	On-Chain	Scale
OVN [12]	2017	None	None	Yes	Boardroom
Panja & Roy [13]	2021	None	None	Partial	Small
Alam & Joshi [14]	2022	None	None	Partial	Small
Marcellino [15]	2024	None	External	Hybrid	Prototype
zkVoting [16]	2024	None	Registrar	Yes	Formal
Rabia et al. [17]	2024	None	None	Yes	PoC
Sangraula [10]	2025	None	None	Yes	Prototype
ZKHC-Voting [11]	2025	None	None	Hybrid	Prototype
BAVS-ZK (**proposed**)	2026	HME-KG	AES-256-GCM	Yes (full)	10,000

3. Threat Model

This framework considers a probabilistic polynomial-time (PPT) adversary whose objective is to impersonate legitimate voters, recover voter secret keys, cast fraudulent ballots, or violate voter privacy. BAVS-ZK assumes that SHA-256, AES-256-GCM, Poseidon, and Groth16 remain computationally secure.The attacker may possess one or more of the following capabilities: (1) database compromise—access to a complete copy of the registration database including encrypted credentials and salts; (2) public information disclosure—knowledge of a voter's National Identity Number (NID); (3) blockchain observation—all transactions, commitments, and nullifiers are publicly visible; (4) network eavesdropping—traffic interception without breaking TLS; (5) partial CDS leakage—incomplete knowledge of the Client Device Secret. However, the attacker is not assumed to compromise all independent entropy sources simultaneously. In particular: the master key K_{master} is stored outside the database;

the CSPRNG produces unpredictable 256-bit salts; the CDS cannot be remotely reconstructed; and standard cryptographic assumptions hold.Security Boundary. HME-KG tolerates compromise of any single entropy source. Recovering the secret key requires simultaneous access to all three sources (NID, CDS, R_v) plus the externally protected K_{master}.

Table 2. *Threat Model and Defense Mechanisms*

Threat	Attacker Capability	Defense
DB leakage	Encrypted DB + salts	AES-256-GCM + external K_{master}
Identity disclosure	Knowledge of NID	CDS + random salt
Blockchain analysis	Observe nullifiers	zk-SNARK + Poseidon
Replay attack	Resubmit proof	Nullifier uniqueness
Double voting	Multiple ballots	Smart contract mapping
Brute-force	Guess secret key	256-bit salt + SHA-256
CDS leakage	Partial CDS	Multi-source entropy
Contract tampering	Direct interaction	Groth16 verification

4. Proposed Method

4.1. Design Rationale

Keys derived solely from predictable information like an NID are susceptible to offline dictionary attacks. In contrast, a fully random key cannot be re-derived at voting time unless the raw secret is stored, i.e., the storage-compromise problem is reintroduced. HME-KG prevents both failures by mixing three independent entropy sources. The National Identity Number firmly establishes identity at the core. Alongside this is the Client Device Secret (CDS), a 256-bit value generated once per device with a CSPRNG. Storing this secret only on the client side ensures that it is not predictable, nor stored in a centralized database.In addition, the random salt R_V ensures that even if two different voters use the same device, they never produce the same keys. SHA-256 is specified as the derivation function in HME-KG because it functions as a deterministic pseudorandom function (PRF) , which is different from password-hashing KDFs. Regular password KDFs, like PBKDF2, Argon2, and scrypt, are designed to have latency baked into the computation to slow down dictionary attacks on poor human passwords. In contrast, since the CDS is a 256-bit CSPRNG value, it always has maximum entropy. Therefore, iterative stretching adds extra latency without any incremental security benefits. In the random oracle model, SHA-256 provides the necessary collision resistance and one-wayness required by this construction. The use of AES-256-GCM for credential storage is warranted as it provides authenticated encryption, which offers both confidentiality and data integrity. To reduce the risk of a single system compromise, the master key K_{master} is kept separate from the main database infrastructure. This separation guarantees that a compromise Type equation here.of the database alone is not enough to reveal or compromise voter credentials.

4.2. Formal Specification

Let NID denote the voter's national identity number, CDS a per-device 256-bit Client Device Secret, and R_v a 256-bit per-voter random salt. HME-KG is defined as:

$$H^I_n = \text{SHA-256}(\text{NID} \parallel \text{CDS}) \quad (1)$$

$$X = R_v \oplus H^I_n \quad (2)$$

$$sk = \text{SHA-256}(X) \quad (3)$$

where $\parallel$ denotes concatenation and $\oplus$denotes bitwise XOR.

4.3. HME-KG Algorithm

Algorithm 1: HME-KG Key Derivation

Input: NID, CDS
Output: secretKey, R_v

1. $R_v \leftarrow$ CSPRNG(256)
2. $H^I_n \leftarrow$ SHA-256(NID ‖ CDS)
3. $X \leftarrow R_v \oplus H^I_n$
4. sk $\leftarrow$ SHA-256(X)
5. Store $Enc_{+}\square(R_v)$; Return Hex(sk)

4.4. Credential Storage

The derived secret key is immediately encrypted with AES-256-GCM using a fresh 96-bit IV before database insertion. The CDS is never transmitted to the server and never stored centrally. A database breach without K_{master} cannot recover any voter secret key. To mitigate potential client-side malware risks and platform-specific storage vulnerabilities, a production-grade implementation of BAVS-ZK isolates the CDS within the device's hardware-backed execution environment, such as a Secure Enclave or Trusted Execution Environment (TEE). Cryptographic access to this key material is managed securely via the WebAuthn API or standardized Passkeys, ensuring that the CDS remains computationally inaccessible to malicious user-space processes even if the host operating system environment is partially compromised.

5. The BAVS-ZK System

5.1. System Architecture

BAVS-ZK employs three layers. The Client/UI Layer generates the Groth16 proof in-browser via snarkjs/WASM and maintains the CDS in client-side secure storage. The Off-Chain Layer (Node.js + Express.js + SQLite) handles voter registration, HME-KG derivation, and AES-256-GCM encryption. The On-Chain Layer consists of a Groth16 Verifier contract and a VotingContract that enforces nullifier uniqueness on Ethereum Sepolia. Although the on-chain evaluation is benchmarked on the Sepolia testnet, the entire smart contract architecture of BAVS-ZK is fully compatible with the Ethereum Virtual Machine (EVM) and natively optimized for Layer-2 (L2) Rollups, such as Arbitrum or Optimism. Moving to an L2 paradigm, the execution costs are shifted from volatile Layer-1 mainnet prices to batch-proof execution prices, making the financial impact per vote just a few pennies. In addition, the L2 sequencer and mempool handle extraordinarily high volumes of concurrent transaction traffic in a live election by processing them sequentially, which decouples client-side proof generation from blockchain throughput bottlenecks.

5.2. Voting Circuit

The Circom 2.1.6 circuit compiles to an R1CS with a size of 958 constraints.It takes a secretKey (private) and candidateId (public) as inputs, and calculates the nullifier = Poseidon(secretKey) and commitment = Poseidon(secretKey, candidateId). The Poseidon hash requires approximately 60 circuit constraints versus over 25,000 for SHA-256 in arithmetic circuits.

Algorithm 2: voteWithProof (Solidity)

Input: proof, [candidateId, nullifier, commitment]

1. if !Verifier.verify(proof): REVERT
2. if hasVoted[nullifier]: REVERT
3. hasVoted[nullifier] ← true
4. votes[candidateId]++
5. emit VoteSubmitted(nullifier, commitment)

6. Security Analysis

Proposition 1 (Determinism)
The same NID, CDS, and R_v always produce the same secret key.
Proof Sketch. SHA-256 and XOR are deterministic. Given fixed inputs, each step of Algorithm 1 produces a unique fixed output; thus secretKey is fully determined by (NID, CDS, R_v).

Proposition 2 (Uniqueness)
The probability of two voters sharing the same secret key is computationally negligible.
Proof Sketch. Each R_v is drawn uniformly from $\{0,1\}^{256}$, so $Pr[R_v^i = R_v^j] = 2^{-256}$. Since R_v feeds into sk via XOR before a final SHA-256 application, distinct salts produce distinct keys with overwhelming probability.

Proposition 3 (Single-Source Resistance)
Compromising any single entropy source is insufficient to recover the secret key.
Proof Sketch. If only NID is known, H^I_n = SHA-256(NID‖ CDS) remains unknown. Since X = $R_v \oplus H^I_n$ and R_v is independently random, X is uniformly distributed from the adversary's perspective, making sk = SHA-256(X) computationally indistinguishable from a random value. The same argument holds symmetrically for CDS-only or R_v-only compromise.

Proposition 4 (Collision Resistance)
The security of HME-KG relies on that of SHA-256 and it is collision-resistant under the above security model.
Proof Sketch. A collision in secret key (sk) is either a SHA-256 collision or two equal intermediate values X. Given standard assumptions about SHA-256, it is computationally infeasible for an adversary to find two distinct input tuples (NID, CDS, R_v) that map to the same sk.

Additional Security Properties
Voter Anonymity: The nullifier is the only on-chain value associated with a voter that is publicly observable. Inverting Poseidon(secretKey) is computationally infeasible, and de-anonymization requires access to the database, K_{master} and the CDS at once.
Double-Voting Prevention: The hasVoted[nullifier] mapping is checked and set atomically. No duplicate votes were found in any of the experiments.
Database Privacy: There is no foreign key between the Voters and Votes tables (and, in fact, votes are issued by nullifiers, not voters,so this is structurally consistent). The only mathematical relationship (nullifier = Poseidon(secretKey)) is computed inside the circuit and is never persisted in the database
Trusted Setup: Phase 1 had a17-party multi-party setup (very public and very secure). Phase 2 was performed as a single party ceremony in order to provide the initial compilation/prototype for evaluation and performance benchmarking. For practical production use, this implementation option is intended to be superseded by a distributed Multi-Party Computation (MPC) ceremony implementing the SnarkJS phase2 contribution framework. This change of structure guarantees that the voting circuit is cryptographically sound as long as there exists at least one honest, uncompromised participant in the ceremony.

7. Experimental Results

7.1. Experimental Setup

All experiments were executed on an Intel Core i7 with 16 GB RAM. On-chain experiments were conducted using the Ethereum Sepolia testnet (Chain ID 11155111). The contracts were compiled using Solidity 0.8.19 and Hardhat 2.28.6; the ZK circuit was developed using Circom 2.1.6 and snarkjs 0.7.5.

7.2. Performance Measurements

Table 3. *BAVS-ZK Performance Measurements*

Metric	Value	Notes
Proof generation	1.2 s	Browser (snarkjs + WASM)
Proof size	804 bytes	Groth16, 3 EC elements
On-chain verification	< 10 ms	3 BN254 pairings (EIP-197)
Gas per vote	306,720	Verification + tally update
Gas vs. OVN	-38.6%	OVN baseline: ~500,000
Circuit constraints	958	477 nonlinear + 481 linear
Unit tests	12 / 12 passed	Hardhat + Chai

7.3. Scalability

A 10,000-voter simulation (7,500 votes cast, 75% turnout) had no duplicate votes or nullifier collisions. A linear regression for seven points (1,000–10,000 voters) gives $R^2 = 0.9998$ (scipy.stats.linregress), which confirms that proof generation and verification scale linearly with the number of voters.

7.4. Gas Cost Comparison

Table 4. *Gas Cost Comparison with Related Systems*

System	Gas / Vote	vs. OVN
OVN [12]	~500,000	—
Rabia et al. [17]	~420,000	−16%
BAVS-ZK (proposed)	306,720	−38.6%

7.5. Live On-Chain Evaluation

Nine votes were cast to the deployed VotingContract on Sepolia, all with 12-seconds confirmations. The on-chain nullifier count matched the database has_voted exactly, showing correct end-to-end operation.

7.6. Discussion and Performance Attribution

The experimental results show that BAVS-ZK achieves a significant 38.6% reduction in on-chain gas overhead compared to the baseline of classic Open Vote Network (OVN).This efficiency is mainly due to the use of the Poseidon hash function in the Circom arithmetic circuit in place of SHA-256. Whereas a typical SHA-256 circuit

has over 25,000 non-linear constraints, Poseidon achieves a mere 60 constraints per invocation, which results in a significant proof size compression (804 bytes) and a minimization of cryptographic verification cost on EIP-197 precompiles.Since the Client/UI Layer is responsible for executing the linear R2=0.9998 Groth16 proof generation entirely in the browser with snarkjs/WASM, execution latency remains highly stable (1.2 seconds) on the client side, effectively decoupling computational work from blockchain network congestion or gas price volatility.

8. Limitations and Future Framework Directions

BAVS-ZK efficiently constructs a zero-knowledge framework for voter credential derivation, but its current deployment scope is limited, and its future integration pathways are guided by several parameters:

Evaluation Data Distribution: Performance assessments applied synthetic data to mimic a population of 10,000 voters with strict statistical requirements. Although these datasets exactly mimic the cryptographic workload, testing the framework on diverse, real-world regional electoral registers constitutes a goal for future piloting.

Infrastructure Scaling: The on-chain gas and transaction confirmation were tested in single-node setups over the public Sepolia testnet. While the local proof generation (1.2 seconds) has linear scaling (R2 = 0.9998) and is fully off-chain on the client, it remains necessary to test the peer-to-peer propagation delays under heavy transaction concurrency via full distributed network stress-testing over geographically scattered nodes .

Cryptographic Setup Evolution: The present Groth16 implementation is based upon a local Phase 2 trusted setup for the purpose of demonstration. Avoiding the logistics of organizing global multi-party ceremonies in production by migrating the architecture towards transparent, non-trusted setup constructions — like PLONK or zk-STARKs — is a very reasonable engineering path.

9. Conclusion

This paper proposed HME-KG, which is a hybrid multi-source-entropy-based key generation scheme for bridging the credential management gap in blockchain e-voting. HME-KG derives a per-voter secret key from three independent sources of entropy – the voter's NID, a per-device Client Device Secret, and a cryptographically secure random salt – while still providing determinism, uniqueness, single-source resistance, and collision resistance under standard SHA-256 assumptions. Together with AES-256-GCM storage and external key management, not even a single system breach can reveal the voter's credentials.BAVS-ZK integrates HME-KG with a 958-constraint Circom circuit, Groth16 proof generation, and on-chain verification. Evaluation verified 1.2-second proof generation, 306,720 gas per vote (38.6% under the OVN baseline), and flawless execution in a 10,000-voter simulation and 9 live Sepolia transactions. Future directions include a multi-party Phase 2 trusted setup that is publicly auditable; adding or integrating transparent proof systems (PLONK or zk-STARKs) particularly post-quantum- resistant constructions such as zk-STARKs, such that the trusted setup reliance is completely circumvented; building decentralized identity recovery mechanisms such as zero-knowledge social recovery or registrar-assisted blind credential re-issuance—allowing for secure management of potential client device or CDS data loss without compromising voter anonymity; and use of WebAuthn/Passkeys as a standard-based device-bound credential mechanism.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.

- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] D. Chaum, "Secret-ballot receipts," IEEE Security and Privacy, vol. 2, no. 1, pp. 38–47, 2004.
[2] A. Kiayias and M. Yung, "The vector-ballot e-voting approach," in Proc. ACM CCS, 2004, pp. 72–89.
[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
[4] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Yellow Paper, 2014.
[5] U. Jafar et al., "Blockchain for electronic voting system," Sensors, vol. 21, no. 17, p. 5874, 2021.
[6] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," SIAM J. Computing, vol. 18, no. 1, pp. 186–208, 1989.
[7] J. Groth, "On the size of pairing-based non-interactive arguments," EUROCRYPT 2016, pp. 305–326.
[8] Ethereum Foundation, "EIP-197: Precompiled contracts for optimal Ate pairing," 2017.
[9] Ethereum Foundation, "EIP-1108: Reduce alt_bn128 precompile gas costs," 2019.
[10] P. Sangraula and N. B. Adhikari, "Zero knowledge proof on top of blockchain for anonymous and verifiable e-voting," J. Inst. Engineers (India) Series B, vol. 106, pp. 1861–1872, 2025.
[11] ZKHC Research Group, "ZKHC: A privacy-preserving hybrid blockchain architecture for scalable e-voting," Springer, 2025.
[12] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting," Financial Cryptography, 2017, pp. 357–375.
[13] S. Panja and B. K. Roy, "A secure end-to-end verifiable e-voting system using ZKP and blockchain," Springer, 2021.
[14] J. Alam and S. R. Joshi, "Blockchain based e-voting system with ZKP," IOE Graduate Conf., 2022.
[15] M. Marcellino et al., "Zero-knowledge identity authentication for e-voting," JUCS, 2024.
[16] S. Park et al., "zkVoting: ZKP based coercion-resistant e-voting system," IACR ePrint 2024/1003.
[17] F. Rabia, A. Sara, and G. Taoufiq, "ZkSNARKs and ticket-based e-voting," Data and Metadata, vol. 3, 2024.
[18] NIST, "SP 800-38D: Recommendation for block cipher modes of operation: GCM and GMAC," Nov. 2007.
[19] L. Grassi et al., "Poseidon: A new hash function for zero-knowledge proof systems," USENIX Security, 2021, pp. 519–535.