ISSN: 2182-2069 / E-ISSN: 2182-2077

Zero-knowledge Identity Authentication for E-voting System

Matthew Marcellino[1], Arya Wicaksana[2*], and Moeljono Widjaja[3]

[1]Research Scholar, Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia. matthew.marcellino@student.umn.ac.id, https://orcid.org/0009-0006-7213-4237

[2*]Associate Professor, Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia. arya.wicaksana@umn.ac.id, https://orcid.org/0000-0002-0888-036X

[3]Assistant Professor, Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia. moeljono.widjaja@umn.ac.id, https://orcid.org/0000-0003-3002-7426

Received: December 03, 2023; Revised: February 03, 2024; Accepted: March 05, 2024; Published: May 30, 2024

Abstract

The advancement of blockchain technology introduces the new concept of electronic voting systems (e-voting) that are fully anonymous, transparent, trustless, and decentralized. The limitation of blockchain-based e-voting systems is the need for initial setup to verify and validate eligible voters. This initial setup requires human intervention, which curbs the full potential and exploitation of blockchain technology. Identity authentication is crucial in voting systems to ensure the eligibility of the voters and the validity of the results. This paper proposes a hybrid approach using ZK-SNARK for identity authentication systems in blockchain-based e-voting. The proposed hybrid approach aims to maintain the benefit of blockchain technology while guaranteeing the eligibility of voters. Both on-chain and off-chain identity authentication modules are designed and developed to balance the trade-off of centralized and decentralized nature for the blockchain-based e-voting systems. The affordability of the proposed system is essential in justifying the approach's feasibility and usability. Voting systems are expected to host thousands to millions of voters, and the cost is one major consideration. The proposed system is deployed in the Ethereum blockchain network, including its sidechain and Layer 2, i.e., Avalanche, Arbitrum One, and Polygon. The gas fee required for the smart contract deployment in Ethereum is USD12.5, while the lowest gas fee is in the Polygon blockchain network for USD0.02.

Keywords: Blockchain, E-voting, Identity Authentication, Voter Identification, Zero-knowledge.

1 Introduction

Traditional voting systems face numerous challenges and shortcomings that have led to calls for exploration of blockchain technology as a potential solution (Fan et al., 2023; Muralidharan, 2020). These challenges include security concerns like hacking, tampering, fraud, lack of transparency, centralization and control, limited accessibility, cost and efficiency, and trust issues (Kumar et al., 2023; Sreenivasu et al., 2022). Paper-based voting systems are susceptible to ballot stuffing, manipulation of vote counts, and coercion. Traditional voting systems also lack transparency which leads to poor accountability, making it difficult for voters to verify the integrity of the election process. Centralized

Journal of Internet Services and Information Security (JISIS), volume: 14, number: 2 (May), pp. 18-31.
DOI: *10.58346/JISIS.2024.I2.002*

*Corresponding author: Associate Professor, Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia.

voting systems are controlled by government agencies or election authorities which leads to concerns about bias, manipulation, and lack of impartiality in the administration of elections. In addition to the costly and resource-intensive paper-based voting systems (Siddiquee et al., 2017).

As technology advances, E-voting has become very popular as an alternative to traditional voting systems. In general, E-voting offers several benefits, such as saving human resources, speeding up the vote-counting process, improving the accuracy of vote-counting, reducing costs, and increasing public participation in voting (Christyono et al., 2021; Gupta et al., 2023; Mark et al., 2021; Tanwar et al., 2023; Wicaksana et al., 2021). The biggest challenge in building an E-voting system lies in the security aspect. With its decentralized, immutable, and transparent nature, blockchain technology can enhance the security of e-voting systems regarding data integrity and prevent data manipulation. Another security aspect to consider is ensuring the validity and non-falsification of votes cast by voters, which means it is mandatory to do an authentication process for voter identities (Jafar et al., 2021). However, authenticating identities poses a challenge when done on the blockchain because it is public, and voter identity data stored on it can be accessed by anyone (Abuidris et al., 2019; Lee et al., 2021). Therefore, a secure authentication method is needed to ensure voter identity privacy in blockchain-based E-voting systems (Santhosh, M., 2016; Kiruthika et al., 2019).

DVT Chain, an E-voting system based on blockchain developed by (Alvi et al., 2022), designs an identity authentication system for voters using private keys and credentials. In this system, a smart contract on the blockchain called the voter contract is responsible for executing the authentication process. During the authentication process, voters are asked to log into their wallets using their private keys and enter their credentials, including their national identity card number, name, phone number, and security key. After that, the hash of these credentials is formed and compared with the previously stored hash in the smart contract. However, the privacy of the authentication method used has a drawback because the voter's credentials can be publicly exposed in the transaction history of the blockchain as a parameter when calling functions on the smart contract.

ZK-SNARK is suitable for use as an authentication method in blockchain-based E-voting systems. ZK-SNARK is one type of zero-knowledge proof that can solve the problem of voter identity privacy by allowing someone to prove the validity of a statement without revealing the content of the statement itself (Konkin & Zapechnikov, 2023). In the authentication context in blockchain-based E-voting systems, the statement refers to the voter's identity. Previous research by (Murtaza et al., 2019) has implemented ZK-SNARK in a blockchain-based E-voting system (Siddiquee, 2017). However, ZK-SNARK in that system was only used in the vote-casting process as proof that a voter's vote has been counted and not being tampered with. The authentication process in that system still took place physically at the polling station, where voters had to provide credentials such as biometrics and national identification cards and be authenticated by a provided machine before they could cast their votes.

This research focuses on implementing ZK-SNARK for authenticating voter identities in a blockchain-based E-voting system. Implementing ZK-SNARK in the built system leverages the digital signature of the elliptic curve digital signature algorithm (ECDSA), natively supported on the Ethereum-based blockchain, the type of blockchain used during the research (Hui, 2019). The system has a website-based interface, which serves as an entry point for voters to authenticate themselves before the voting process, accessible from anywhere without the need to visit a polling station. The logic of the authentication process is built into a backend service and smart contract on the Ethereum blockchain network.

The rest of this paper is organized as follows. Section 2 describes the preliminaries of the paper. Section 3 explains the research methods. Section 4 presents the experimental results and discussion. Finally, Section 5 concludes this paper with suggestions for future work.

2 Preliminaries

Zero-Knowledge Succinct Non-Interactive Argument of Knowledge

ZK-SNARK stands for Zero-Knowledge Succinct Non-Interactive Argument of Knowledge. ZK-SNARK shares the same properties as zero-knowledge proof in general, namely completeness, soundness, and zero-knowledge. However, ZK-SNARK has two additional properties that make it more efficient, namely succinctness and non-interactivity (Konkin & Zapechnikov, 2023). Succinctness means that the proof generated by the prover can be verified within milliseconds and has a small size, typically just a few hundred bytes. Non-interactivity means that during the proof verification process, the prover and verifier only interact once without any back-and-forth or additional interactions.

Zero-knowledge proof is a method used to prove the validity of a statement without revealing the content of the statement itself (Konkin & Zapechnikov, 2021). In zero-knowledge proof, there are two parties involved which are the prover, who wants to prove a statement, and the verifier, who validates the statement. By using zero-knowledge proof, the prover can prove that the claimed statement is true without providing any information other than the fact that the statement is true. The ZK-SNARK scheme consists of three algorithms, KeyGen, Proof, and Verify, with four main properties (Luong & Park, 2023):

- Completeness: If the statement is true, the prover should be able to convince the verifier.
- Succinctness: The size of the proof is short regardless of the size of the witness and the public input, and the proof is efficiently verified.
- Zero-knowledge: An untrustworthy or malicious verifier should not gain any information about the statement other than its truthfulness.
- Soundness: An untrustworthy or malicious prover should not be able to convince the verifier if the statement is false.

Ethereum Blockchain

Ethereum is a blockchain protocol designed to build decentralized applications. Ethereum provides smart contract functionality on the blockchain, which is code or programs stored on the blockchain network that represent digital versions of traditional contracts (Buterin, 2014). With smart contracts, developers can build applications that are executed in a decentralized manner on the Ethereum blockchain network, such as NFTs, cryptocurrencies, and various financial services. Within the Ethereum protocol, there are two types of accounts, which are externally owned accounts (EOAs) and contract accounts. Both accounts consist of a private key and a public key generated by the elliptic curve digital signature algorithm (ECDSA). Thus, ECDSA is supported natively in Ethereum. In an EOA, the private key is owned and controlled by the user. In contrast, in a contract account, no one owns the private key, and the contract account can only be controlled based on the code written within it (Jiang et al., 2023).

The Ethereum protocol has a mechanism known as gas, which is a unit that measures the computational effort required to perform specific operations on the Ethereum network. Computational resources are needed to execute a transaction on Ethereum, known as the gas fee, which must be paid to

the network maintainers, i.e., the miners or validator. This gas fee is paid using ether (ETH), the native currency of the Ethereum protocol. The exact amount of gas fee to be paid can be calculated using the following formula:

$$\text{gas fee} = \text{gas used} \times (\text{base fee} + \text{priority fee}) \quad (1)$$

In Formula 1, gas used refers to the amount consumed by an operation or transaction on the Ethereum network. The base fee represents the minimum fee (in GWEI) required for each unit of gas used to include the transaction in a block on the blockchain. The priority fee is an additional fee or tip to miners to prioritize the transaction. In addition to compensating miners for the computational resources utilized, the gas fee also serves to maintain the security of the Ethereum network by preventing spamming or malicious activities on the network.

3 Methods

Requirement Analysis

There are three entities in the system: identity provider (IP), voter, and verifier. The Identity Provider (IP) is assumed to be a trusted party that provides the voter identity data for the electoral voting process. In the scope of this research, the IP is a government entity responsible for conducting the voting process. The IP provides an off-chain database containing voter identities and a backend service (API) to handle voter registration and off-chain identity authentication. The system authenticates the identities of voters before they can cast their votes. The verifier is the party responsible for authenticating the identities of voters before they can cast their votes in the election. As the election is conducted using a blockchain-based E-voting system, the verifier in the context of this research is an Ethereum smart contract deployed on the blockchain network.

Design

The implemented system utilizes ZK-SNARK, requiring the generation of a cryptographic representation known as a proof that voters use to prove their identities. The generated proof is zero-knowledge, which means it does not reveal any private and sensitive voter data. There are two types of proofs used in the system: identity certificate and proof of identity. IP issues an identity certificate to voters as evidence that they have registered their identities and meet the requirements to participate in the election. The identity certificate is a digital signature signed by the IP's private key. A proof of identity is a digital signature the voter generates using their private key. The voter uses the proof of identity to prove their identity to the IP and obtain an identity certificate when needed.

The ZK-SNARK scheme consists of three parts: key generation (keygen), proof, and verification. Keygen stands for key generation, which is responsible for generating an Ethereum wallet consisting of a private key and a public key using the elliptic curve digital signature algorithm (ECDSA), which is a standard in Ethereum cryptography. In this ZK-SNARK scheme, the private key serves as the proving key (PK), and the public key serves as the verifying key (VK). The Keygen algorithm is reusable, meaning it can be used for various processes, such as verifying an identity certificate or proof of identity, as shown in Algorithm 1.

Algorithm 1: Key Generation

Input: C

Output: PKc, VKc

1. Import ethers library
2. Wallet = ethers.Wallet.createRandom()
3. Return {wallet.privateKey as PKc, wallet.address as VKc}

Proof is a function used by the prover to generate a proof π to prove a statement to the verifier. In this function, the generated proof is an instance of the ECDSA signature. There are two Proof functions in this ZK-SNARK scheme, namely Generate Identity Certificate and Generate Proof of Identity. The IP uses the Generate Identity Certificate function in Algorithm 2 to generate an identity certificate π_{IC} for a voter. The identity certificate is created by digitally signing the hash of the voter's identity using the IP's proving key and the ECDSA cryptography. The Generate Proof of Identity function in Algorithm 3 is used by the voter to generate a proof of identity π_{PoI}. The proof of identity is created by digitally signing the auth message, which combines the public arguments and the nonce, using the voter's proving key and the ECDSA cryptography.

Algorithm 2: Generate Identity Certificate

Input: identityHash, PKip

Output: π_{IC}

1. Import ethers library
2. π_{IC} = ethers.signMessage(identityHash, PKip)
3. Return π_{IC}

Algorithm 3: Generate Proof of Identity

Input: authMessage, nonce, PKvoter

Output: π_{PoI}

1. Import ethers library
2. Msg = concat(authMessage, nonce)
3. π_{PoI} = ethers.signMessage(msg, PKvoter)
4. Return π_{PoI}

Verify is a function used by the verifier to verify the proof π provided by the prover and ensure the claimed statement is true. There are two Verify functions in this ZK-SNARK scheme, namely Verify Proof of Identity and Verify Identity Certificate. The Verify Proof of Identity function in Algorithm 4 is used by the IP to verify the proof of identity π_{PoI} provided by the voter. The verification process involves recovering the public key that produced the digital signature in the proof of identity π_{PoI}. If the recovered public key matches the voter's verifying key VKvoter, the function returns true, indicating a successful verification.

Algorithm 4: Verify Proof of Identity

Input: π_{PoI}, authMessage, nonce, VKvoter

Output: boolean

1. Import ethers library
2. Msg = concat(authMessage, nonce)
3. recoveredAddr = ethers.verifyMessage(π_{PoI}, msg)
4. if recoveredAddr = VKvoter return true, else return false

The Verify Identity Certificate function in Algorithm 5 is a function that runs automatically on the blockchain when a voter wants to cast a vote. This function is part of the Verifier actor, a smart contract on the Ethereum blockchain. The signature needs to be split into three components to verify the ECDSA signature in the Ethereum smart contract: v, r, and s, using the splitSignature function. After that, the ecrecover function, a native function of the Ethereum smart contract, can recover the public key that produced the digital signature in the identity certificate π_{IC}. If the recovered public key matches the IP's verifying key VKip, the function returns true, indicating a successful verification.

Algorithm 5: Verify Identity Certificate

Input: π_{IC}, identityHash, VKip

Output: boolean

1. prefixedMessage = concat("\x19Ethereum Signed Message:\n32", identityHash)
2. messageHash = keccak256(prefixedMessage)
3. (v, r, s) = splitSignature(π_{IC})
4. recoveredAddr = ecrecover(π_{IC}, messageHash)
5. if recoveredAddr = VKip return true, else return false

Implementation

Registration is the first step that a voter needs to undergo in the blockchain-based e-voting process. The purpose of the registration phase is to verify the voter's identity and provide them with a new digital identity to be used throughout the voting process on the blockchain, particularly for authentication purposes. Before the registration process begins, the Identity Provider (IP) performs initialization steps, as shown in Figure 1.

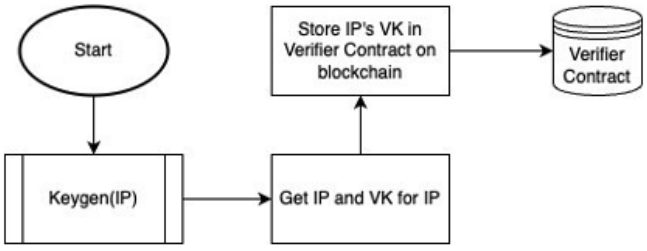

Figure 1: Identity Provider Initialization Process

The IP executes the Keygen function in the ZK-SNARK scheme to generate PKip and VKip. After obtaining this key pair, the IP stores VK_{IP} in the verifier contract on the blockchain. Then, the IP initiates

the registration process following the flow depicted in Figure 2. The registration process is assumed to be conducted offline at a designated registration location provided by the IP. Therefore, voters are required to visit this location with identification documents, such as an ID card, to be verified in person by IP personnel. If the verification process is successful, the IP calls the Keygen function in the ZK-SNARK module to generate PKvoter and VKvoter. PKvoter is sent to the voter as the key to their digital identity. Afterwards, the IP uses the keccak256 algorithm to generate a hash of the voter's identity. This identity hash is digitally signed by the IP using the Generate Identity Certificate function in the ZK-SNARK scheme, generating an identity certificate π_{IC}. VKvoter, the identity hash, and the identity certificate π_{IC} are stored in the IP's off-chain database. In contrast, VKvoter and the identity hash are stored in the verifier contract using a mapping structure.

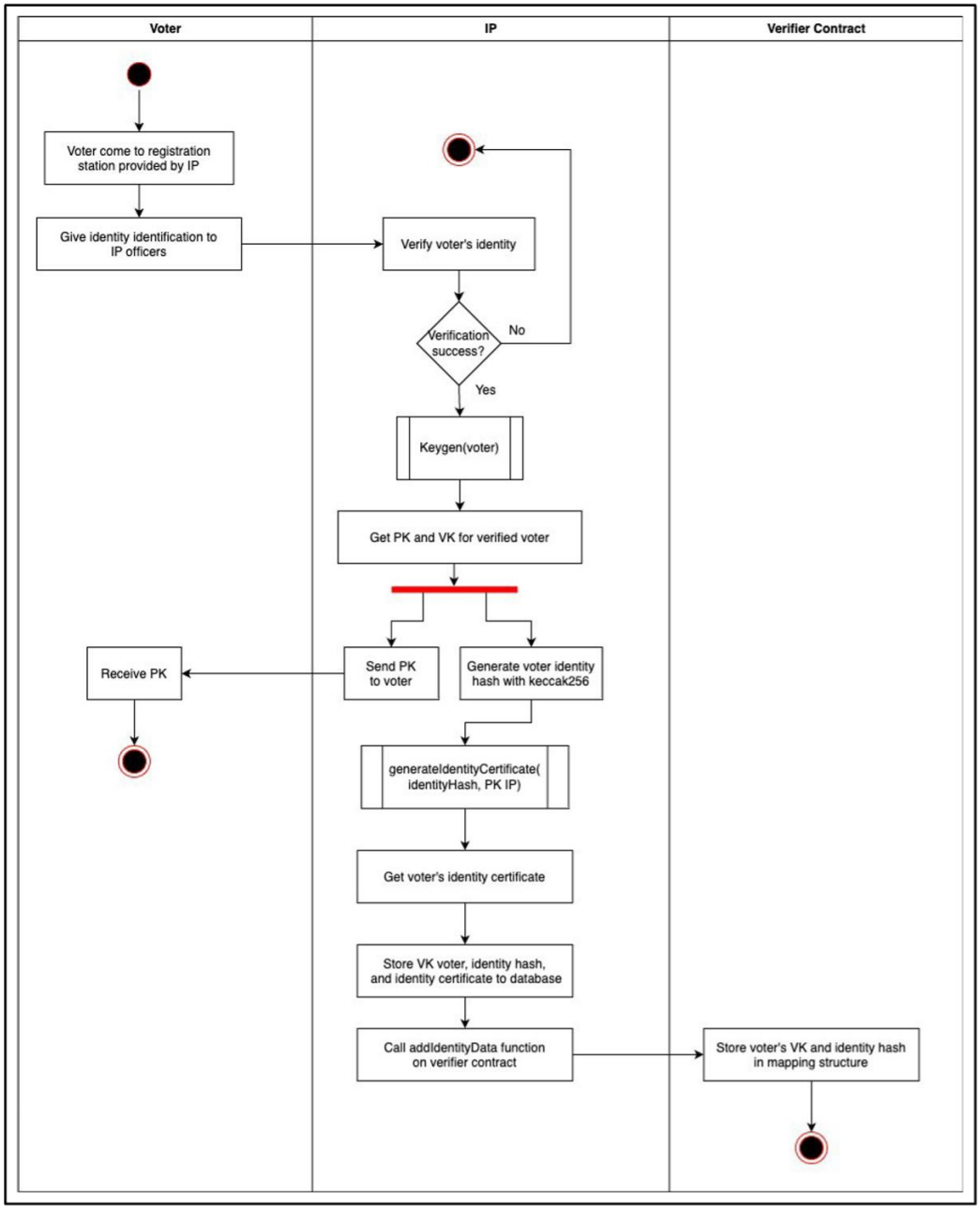

Figure 2: Registration Process

Off-chain authentication is the initial authentication process that voters need to undergo once the voting process has started. The purpose of off-chain authentication is to enable voters to access their identity certificate, which is a requirement for participating in the blockchain-based e-voting system. The flow of off-chain authentication can be seen in Figure 3.

Figure 3: Off-chain Authentication Process

In the off-chain authentication process, the first step for the voter is to log into their Ethereum wallet using the given PK (proving key) from the registration process. After that, the voter can visit the voting website to initiate a request for their identity certificate from the IP's backend system. The IP's backend system responds to this request by sending an *authMessage* and a *nonce* to the voter to be digitally signed. With this response, the voter can execute the Generate Proof of Identity function from the ZK-SNARK scheme to generate the proof of identity π_{PoI} through the voting website. The voter sends another request to the IP's backend system, attaching the proof of identity π_{PoI} to the request. The IP's backend system verifies the proof of identity π_{PoI} using the Verify Proof of Identity function from the ZK-SNARK scheme. If the proof is valid, the backend system sends the voter's identity certificate π_{IC}.

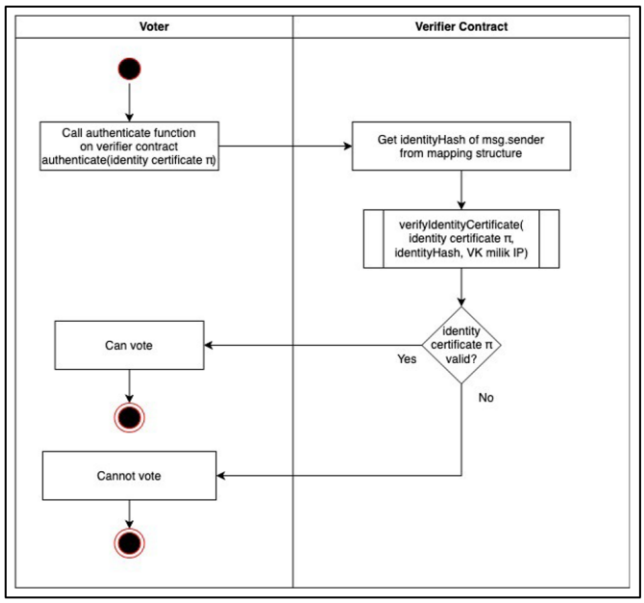

Figure 4: On-chain Authentication Process

In the on-chain authentication process as shown in Figure 4, the voter calls a function on the verifier contract called ”authenticate” using their Ethereum wallet. To call this function, the voter provides the identity certificate π_{IC} obtained during the off-chain authentication process as a parameter. The verifier contract starts the authentication by retrieving the identity hash of the msg.sender, which is the address of the Ethereum wallet, executing the ”authenticate” function. In this case, the msg.sender corresponds to the voter’s public key or VKvoter. Once the identity hash is obtained, the Verify Identity Certificate function in the ZK-SNARK scheme is executed. If the function returns true, then the voter can proceed with voting.

4 Results

The off-chain authentication module in the system can be used by voters when the voting phase has started. The main objective of this module is to ensure that someone who wants to access the identity certificate is a voter with the correct identity and has been registered in the KPU system. Voters can access this module by visiting the voting website. The main display of the voting website can be seen in Figure 5.

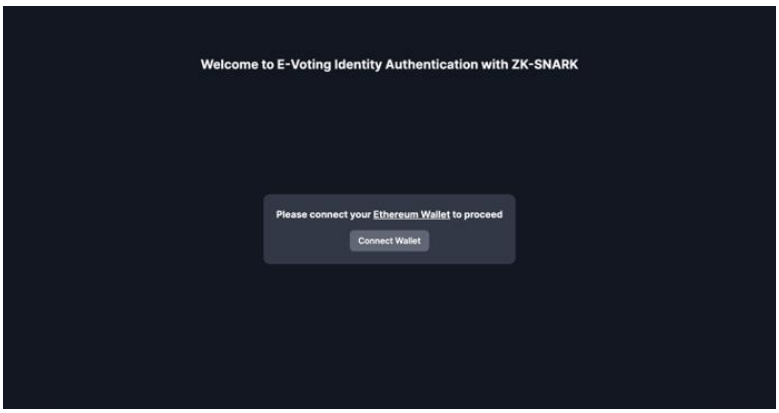

Figure 5: E-voting Website Landing Page

On the main display of the voting website, there is a "Connect Wallet" button to connect the voter's Ethereum wallet to the website. After the voter's wallet has been connected, the website's appearance changes, as displayed in Figure 6.

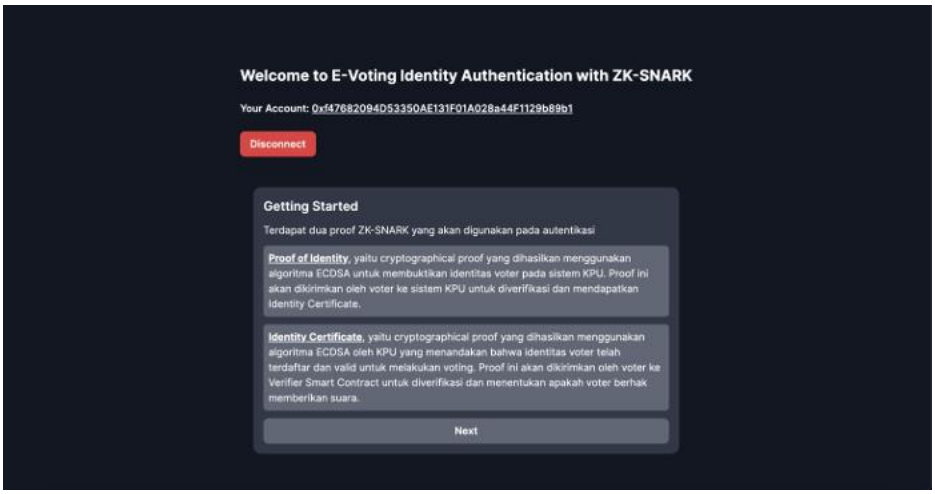

Figure 6: Voter's Wallet Successful Connection

After the voter's wallet is connected, the public key is displayed at the top, and a "Disconnect" button is used to disconnect the wallet from the website. There is also a dialog in the middle that explains the types of ZK-SNARK proof used during the authentication process. After the "Next" button is pressed, the Generate Proof of Identity web page is displayed as in Figure 7.

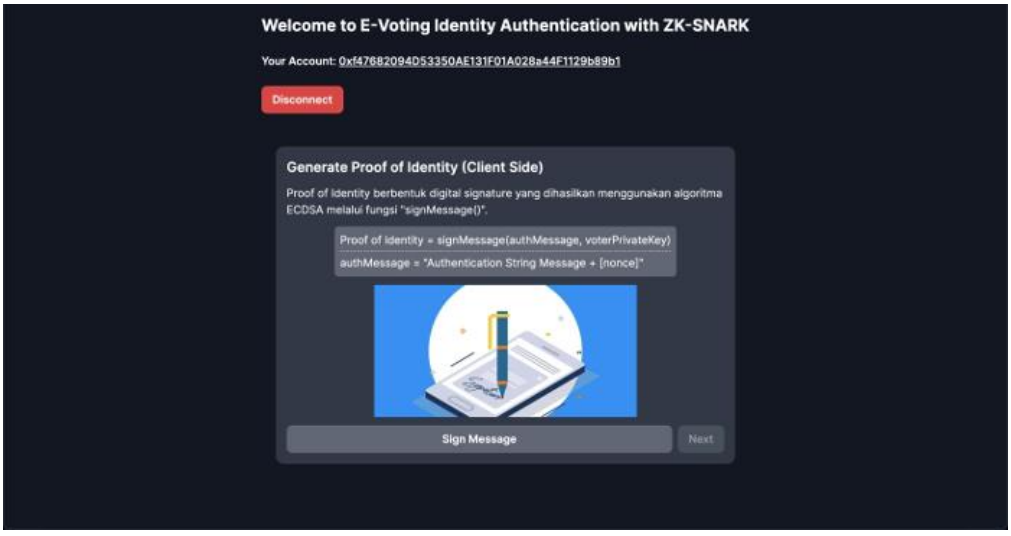

Figure 7: Generate Proof of Identity Web Page

On the web page, there is a brief explanation of how proof of identity is and how it can be produced. When the voter presses the "Sign Message" button, the website executes an API endpoint called Get Nonce. The on-chain authentication module in the system has the main objective of providing access for voting carried out on-chain to voters with a valid identity certificate. Voters can use this module via the voting website after all stages of the off-chain authentication module have been completed. The voting website for carrying out on-chain authentication can be seen in Figure 8.

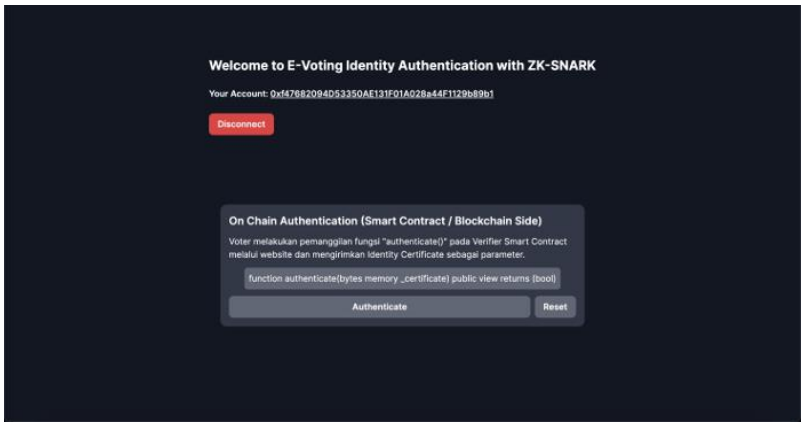

Figure 8: On-chain Authentication Web Page

On the On-Chain Authentication web page, there is a brief explanation of the on-chain authentication process. To authenticate, voters can press the "Authenticate" button. When the button is pressed, the authenticate function in the smart contract is executed using the voter's wallet. The authenticate function implements the Verify Identity Certificate function in the ZK-SNARK module.

The amount of gas used to execute operations on the verifier contract was tested using the Hardhat Gas Reporter tool. Two operations were tested, which are the Add Identity Data function and the deployment of the smart contract. The tool provides the gas used for both operations. Once the gas used is known, the gas fee can be calculated by multiplying the gas used by the gas price and then multiplying it by the price of the native token for each blockchain network. Standard transaction speed is used as the underlying assumption for the comparison. Based on the results, the gas fee on the Ethereum Mainnet is the highest, while the gas fee on the Polygon blockchain network is the lowest. The results of the gas fee testing on the verifier contract can be seen in Table 1.

Table 1: Gas Fee Comparison for Each Blockchain Network (Nov 1st, 2023)

Process	**Gas Used**	**Gas Fee (USD)**			
		Ethereum	**Avalanche**	**Arbitrum One**	**Polygon**
Add identity data	44,788	1.135	0.0125	0.008	0.0019
Deploy contract	491,786	12.464	0.1377	0.089	0.0205

The obtained gas fee comparison for the proposed identity authentication system shows the feasibility and usability of the hybrid approach. This system involves a centralized entity for the initial setup phase at the minimum level possible. Apart from the eligible voter registration at the beginning of the initial setup for first-time users (trusted setup), the system requires no further direct intervention nor interaction with any centralized entity within the e-voting systems. Once the public parameters are created, the system can be used for efficient and private transaction verification without revealing the transaction details.

Authentication systems, tasked with verifying user identities, are susceptible to a range of security vulnerabilities. Password-based attacks exploit weak or reused passwords through brute-force or dictionary attacks, while phishing schemes deceive users into revealing credentials. Man-in-the-Middle attacks intercept communication to eavesdrop or manipulate data, session hijacking allows unauthorized access by stealing valid sessions, and brute-force techniques target various authentication factors. Insufficient multi-factor authentication and insider threats further compound risks. However, implementing a Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (ZK-SNARK) identification system offers significant benefits against these vulnerabilities. ZK-SNARKs provide

cryptographic proofs of identity without revealing sensitive information, protecting against password theft, phishing, and interception attacks. By ensuring privacy and security through cryptographic verification, ZK-SNARK identification systems mitigate the risks associated with traditional authentication methods.

While the study explores the technical intricacies of leveraging ZK-SNARK for identity authentication in blockchain-based e-voting systems, it would benefit from addressing broader considerations to enhance the overall robustness and inclusivity of the proposed solution. Embracing a holistic approach that encompasses factors such as voter accessibility, system trustworthiness, and regulatory compliance will bolster confidence in the integrity and effectiveness of the e-voting framework. Additionally, while the study highlights the affordability of the proposed e-voting system, it would strengthen its credibility by providing concrete evidence or analysis to support this assertion. Affordability entails more than just gas fees for smart contract deployment; it extends to considerations such as infrastructure maintenance, security audits, and user education. Addressing these aspects comprehensively will enhance the feasibility and attractiveness of the proposed solution, ensuring its successful implementation and widespread adoption.

5 Conclusion

This paper presents a hybrid approach using the ZK-SNARK scheme for identity authentication in blockchain-based e-voting systems. The system's objectives are to preserve and maintain the benefits of blockchain technology, like anonymity, transparency, trustless, and decentralization for e-voting systems. The problem of voter registration and identity verification and validation for voting is addressed by the proposed hybrid approach through the combination of on-chain and off-chain authentication processes. The proof generated for the ZK-SNARK uses an elliptic curve digital signature algorithm (ECDSA), which is native to the Ethereum blockchain. Furthermore, the feasibility and usability of the system are evaluated by comparing the gas fee cost for smart contract deployment in Ethereum and its sidechain and Layer 2 blockchain networks. It is noteworthy that besides initial deployment fees for the smart contract, there are no other fees required for the voters. Future works on further testing and evaluating the system's scalability and performance are essential in justifying the feasibility and usability factors.

References

[1] Abuidris, Y., Kumar, R., & Wenyong, W. (2019). A Survey of Blockchain Based on E-voting Systems. *Proceedings of the 2019 2nd International Conference on Blockchain Technology and Applications*, 99–104.

[2] Alvi, S.T., Uddin, M.N., Islam, L., & Ahamed, S. (2022). DVTChain: A blockchain-based decentralized mechanism to ensure the security of digital voting system voting system. *Journal of King Saud University - Computer and Information Sciences*, *34*(9), 6855–6871.

[3] Buterin, V. (2014). Ethereum White Paper. *Etherum*.

[4] Christyono, B.B.A., Widjaja, M., & Wicaksana, A. (2021). Go-Ethereum for electronic voting system using clique as proof-of-authority. *Telkomnika (Telecommunication Computing Electronics and Control)*, *19*(5), 1565-1572.

[5] Fan, Y., Saeidi, M., Lai, K. K., Yang, J., Cai, X., & Chen, Y. (2023). Corruption and Infrastructure Development Based on Stochastic Analysis. *Arhiv za tehničke nauke, 1*(28), 11-28.

[6] Gupta, S., Gupta, A., Pandya, I.Y., Bhatt, A., & Mehta, K. (2023). End to end secure e-voting using blockchain & quantum key distribution. *Materials Today: Proceedings*, *80*, 3363-3370.

https://static.peng37.com/ethereum_whitepaper_laptop_3.pdf

[7] Hui, H., An, X., Wang, H., Ju, W., Yang, H., Gao, H., & Lin, F. (2019). Survey on Blockchain for Internet of Things. *Journal of Internet Services and Information Security, 9*(2), 1-30.

[8] Jafar, U., Aziz, M.J.A., & Shukur, Z. (2021). Blockchain for electronic voting system—review and open research challenges. *Sensors, 21*(17), 1-22.

[9] Jiang, P., Guo, F., Susilo, W., Lin, C., Hu, J., Zhao, Z., Zhu, L., & He, D. (2023). EthereumX: Improving Signature Security With Randomness Preprocessing Module. *IEEE Transactions on Services Computing*, *16*(5), 3318–3331.

[10] Kiruthika, J., Poovizhi, V., Kiruthika, P., & Narmatha, P. (2019). Blockchain based Unforged License. *International Journal of Communication and Computer Technologies (IJCCTS), 7*(2), 4-7.

[11] Konkin, A., & Zapechnikov, S. (2021). Privacy methods and zero-knowledge proof for corporate blockchain. *Procedia Computer Science*, *190*, 471–478.

[12] Konkin, A., & Zapechnikov, S. (2023). Zero knowledge proof and ZK-SNARK for private blockchains. *Journal of Computer Virology and Hacking Techniques*, *19*(3), 443–449.

[13] Kumar, A., Joshi, P., Bala, A., Sudhakar Patil, P., Jang Bahadur Saini, D. K., & Joshi, K. (2023). Smart Transaction through an ATM Machine using Face Recognition. *Indian Journal of Information Sources and Services*, *13*(2), 7–13.

[14] Lee, Y., Son, B., Park, S., Lee, J., & Jang, H. (2021). A Survey on Security and Privacy in Blockchain-based Central Bank Digital Currencies. *Journal of Internet Services and Information Security, 11*(3), 16-29.

[15] Luong, D.A., & Park, J.H. (2023). Privacy-Preserving Identity Management System on Blockchain Using Zk-SNARK. *IEEE Access*, *11*, 1840–1853.

[16] Mark, L., Ponnusamy, V., Wicaksana, A., Christyono, B.B., & Widjaja, M. (2021). A secured online voting system by using blockchain as the medium. *The Smart Cyber Ecosystem for Sustainable Development*, 405-430.

[17] Muralidharan, J. (2020). Wideband Patch Antenna for Military Applications. *National Journal of Antennas and Propagation (NJAP), 2*(1), 25-30.

[18] Murtaza, M.H., Alizai, Z.A., & Iqbal, Z. (2019). Blockchain based anonymous voting system using zkSNARKs. *In IEEE International Conference on Applied and Engineering Mathematics (ICAEM)*, 209-214.

[19] Santhosh, M., Kavitha, S., Keerthana, R., Suganya, L., & Krishnakumar, S. (2016). Electronic voting machine using internet. *International Journal of Communication and Computer Technologies (IJCCTS)*, *4*(2), 72-75.

[20] Siddiquee, K.N.E.A., Andersson, K., Khan, F.F., & Hossain, M.S. (2017). A Scalable and Secure MANET for an i-Voting System. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 8*(3), 1-17.

[21] Sreenivasu, M., Kumar, U.V., & Dhulipudi, R. (2022). Design and Development of Intrusion Detection System for Wireless Sensor Network. *Journal of VLSI Circuits and Systems*, *4*(2), 1-4.

[22] Tanwar, S., Gupta, N., Kumar, P., & Hu, Y. C. (2024). Implementation of blockchain-based e-voting system. *Multimedia Tools and Applications*, *83*(1), 1449-1480.

[23] Wicaksana, A., Widjaja, M., Ponnusamy, V., Talib, M.A., Humayun, M., & Sama, N.U. (2021). Towards Secure and Auditable E-Voting System with Go Ethereum. *Turkish Journal of Computer and Mathematics Education*, *12*(10), 3006–3012.

Authors Biography

Matthew Marcellino received a BSc in Informatics from Universitas Multimedia Nusantara, Indonesia, in 2023. His research interest is blockchain and smart contract applications, and he is currently working as a Web3 Engineer.

Arya Wicaksana is an associate professor at the Department of Informatics at UMN. He received a Master's Degree in research in VLSI Engineering from Universiti Tunku Abdul Rahman (UTAR). He successfully demonstrated the UTAR first-time success ASIC design methodology on a multi-processor system-on-chip project using 0.18µm processing technology 2015. His main research interests are blockchain applications and computational intelligence. He recently worked on blockchain-based decentralized autonomous social media. He has been an invited reviewer and an invited author in various scientific publications.

Moeljono Widjaja is an assistant professor at the Department of Informatics, Universitas Multimedia Nusantara. He received a Doctoral Degree in Electrical Engineering from Monash University (Australia). His main research interests are artificial intelligence, big-data analytics, simulation/modeling, and optimization. He developed a fuzzy controller for an inverted pendulum system and a fuzzy-based bidding strategy for generators in an electricity market. He has been working on intelligent energy management systems. He is a professional member of ACM and IEEE.