BlockVote: Harnessing Blockchain for Transparent E-Voting

Mrs. Kamna Sahu[1], Ms. Asmita Mohite[2], Mr. Manav Khakhi[3], Mr. Arya Bagde[4], Mr. Kedar Karale[5]

[2,3,4,5]*Student, Department of Information Technology, International Institute of Information Technology, Pune, India*

[1]*Professor, Department of Information Technology, International Institute of Information Technology, Pune, India*

---***---

Abstract - *The advent of blockchain technology has shown great potential in various domains, and one of the most promising applications is in the field of electronic voting (e-voting) systems. This paper presents a detailed scientific analysis of a blockchain-based e-voting system and provides a mathematical framework to address its key components, including security, transparency, privacy, and robustness. The proposed framework leverages cryptographic techniques and mathematical formulas to ensure the integrity and trustworthiness of the voting process. Through a rigorous analysis, we demonstrate the effectiveness of the blockchain-based e-voting system in preventing fraud, preserving privacy, and enhancing overall trust in the electoral process.*

Key Words: **E-voting, Blockchain, Security, Encryption, Decentralization, Smart contract, Solidity, Ethereum.**

1. INTRODUCTION

Blockchain technology, characterised by its decentralised and immutable nature, enables the creation of a transparent and tamper-resistant ledger that records all transactions or votes. The decentralised nature of the blockchain eliminates the need for a central authority, reducing the risk of single points of failure or manipulation. Additionally, the immutability of the blockchain ensures that once a vote is recorded, it cannot be altered, enhancing the integrity of the voting process.

Block Structure:

block = (previous_hash, transactions, timestamp, nonce, hash)

This formula represents the structure of a block in a blockchain. Each block contains several components:

- previous_hash: The hash value of the previous block in the chain.
- transactions: A set of transactions included in the block.
- timestamp: The timestamp indicating when the block was created.
- nonce: A random value used in the mining process to find a valid hash.
- hash: The hash value of the entire block, obtained by applying a cryptographic hash function to the block data.

The cryptographic algorithms employed in blockchain-based e-voting systems play a critical role in ensuring the security and privacy of voter information and ballot data. Public key cryptography enables the generation of key pairs, digital signatures, and encryption techniques that authenticate the identity of voters, protect the integrity of ballots, and enable secure communication within the system. Consensus mechanisms, such as Proof of Work (PoW), Proof of Stake (PoS), or Byzantine Fault Tolerance (BFT) algorithms, are fundamental components of blockchain-based e-voting systems. These mechanisms establish agreement among participants on the validity and order of transactions, ensuring a distributed and trustworthy decision-making process. The consensus algorithms provide resilience against adversarial attacks, such as double voting or collusion, through mathematical models and game-theoretical frameworks.

Privacy-preserving techniques, including zero-knowledge proofs and homomorphic encryption, address concerns regarding voter anonymity and the confidentiality of ballot data. Zero-knowledge proofs allow voters to prove the validity of their votes without revealing sensitive information, while homomorphic encryption enables vote aggregation and counting on encrypted data, preserving privacy while ensuring verifiability. The scientific introduction presented in this paper aims to provide a comprehensive understanding of the principles and mechanisms behind blockchain-based e-voting systems. By exploring the mathematical foundations, cryptographic algorithms, consensus mechanisms, and privacy-enhancing techniques, we aim to highlight the potential of this innovative technology to enhance the integrity, transparency, and accessibility of digital elections.

2. LITERATURE SURVEY

2.1 Blockchain Trust Model:

A blockchain trust model refers to the use of blockchain technology to establish trust in various contexts, such as digital transactions, supply chain management, or decentralised systems. The fundamental principle behind blockchain's trust model is to create a transparent, decentralised, and immutable ledger that enables secure interactions and eliminates the need for intermediaries or centralised authorities. Here are the key elements and characteristics of a blockchain trust model:

- **Decentralisation:** Blockchain operates on a decentralised network of nodes, where no single entity has full control. This decentralised nature ensures that trust is not reliant on a single point of failure or authority. Instead, trust is distributed among multiple participants, enhancing security and resilience.

- **Transparency:** Blockchain provides a transparent ledger where all transactions and data are recorded and made publicly accessible. Participants can independently verify and audit the information on the blockchain, promoting trust by eliminating the need to rely on a centralised entity.

- **Immutability:** Once data is recorded on the blockchain, it becomes virtually impossible to alter or tamper with. The immutability of blockchain records ensures the integrity and permanence of transactions, reducing the risk of fraud or manipulation. This feature enhances trust as participants can rely on the historical accuracy of the blockchain data.

- **Consensus Mechanisms:** Blockchain networks employ consensus mechanisms to agree on the state of the blockchain and validate transactions. Consensus ensures that all participants reach agreement on the validity of transactions, preventing malicious activities and establishing trust in the system.

- **Cryptographic Security:** Blockchain uses advanced cryptographic techniques to secure the data stored on the network. Public-key cryptography, digital signatures, and cryptographic hashes are employed to authenticate users, ensure data integrity, and protect against unauthorised access. These security measures contribute to building trust in the blockchain ecosystem.

- **Trustless Interactions:** Blockchain allows for trustless interactions, meaning participants can engage in transactions or collaborations without having to trust each other explicitly. The rules and logic of the blockchain network, often encoded in smart contracts, provide a trusted and automated environment, reducing the need for intermediaries and increasing efficiency.

- **Auditing and Accountability:** With the transparent and immutable nature of blockchain, auditing becomes easier and more reliable. Participants can trace and verify the history of transactions, ensuring accountability and trust among network participants. This transparency also acts as a deterrent to fraudulent or unethical behaviour.

By incorporating these elements, a blockchain trust model establishes a robust and secure environment where participants can confidently engage in transactions, share information, and collaborate without relying on central authorities or intermediaries.

2.2 Security Analysis:

In the context of blockchain-based e-voting, the threat model involves identifying potential threats to the security and integrity of the voting system. Here are some common threats to consider:

- **Malicious Actors:** Adversaries may include hackers, insider attackers, or external entities aiming to manipulate or disrupt the voting process. They may attempt to compromise the blockchain network, tamper with votes, or compromise the confidentiality of voter information.

- **Sybil Attacks:** Sybil attacks involve creating multiple fake identities or nodes in the blockchain network to gain control or influence over the voting process. This can lead to vote manipulation or disruption of consensus.

- **Double Spending:** Double spending occurs when a voter tries to cast multiple votes using the same digital token or identity. Preventing double spending is crucial to maintaining the integrity of the voting system.

- **Denial of Service (DoS) Attacks:** Attackers may launch DoS attacks to overwhelm the blockchain network or the voting application, causing disruption and preventing voters from casting their votes or accessing the system.

- **Insider Threats:** Insider threats involve individuals with authorised access to the system who misuse their privileges for personal gain or to manipulate the voting results. This can include election officials, system administrators, or developers.

- **Compromised Endpoints:** The security of the voting system relies on the integrity of the devices and software used by voters to cast their votes. Compromised endpoints, such as infected devices or malicious software, can lead to unauthorised access or manipulation of votes.

- **Privacy Concerns:** Preserving voter privacy is essential in e-voting systems. Threats may arise from inadequate privacy measures, such as the possibility of linking voter identities to their votes or unauthorised access to voter information.

2.2.1 Cryptographic Techniques:

Cryptographic techniques are essential for ensuring the security and privacy of blockchain-based e-voting systems. Here are some cryptographic techniques commonly employed:

- **Public Key Infrastructure (PKI):** PKI enabled secure communication and verification of identities in the e-voting system. Digital certificates bind public keys to voters, ensuring authenticity and integrity in transactions.

- **Digital Signatures:** Digital signatures provide integrity and non-repudiation, ensuring that votes cannot be tampered with and that voters cannot deny casting their votes. Each vote can be signed with the voter's private key, and the signature can be verified with their public key.

- **Zero-Knowledge Proofs:** Zero-knowledge proofs allow the verification of certain statements without revealing any additional information. They can be used to prove that a vote is valid without disclosing the actual vote itself, preserving voter privacy.

- **Homomorphic Encryption:** Homomorphic encryption allows computations to be performed on encrypted data without decrypting it. It can be utilised to process vote tallies while keeping the votes themselves encrypted, maintaining privacy.

- **Hash Functions:** Hash functions ensure the integrity of data by generating unique fixed-size hash values. Votes can be hashed to detect any modifications or tampering attempts.

- **Key Management:** Effective key management practices, including key generation, storage, and distribution, are essential to maintaining the confidentiality and integrity of cryptographic operations in the e-voting system.

2.2.2 Byzantine Fault Tolerance:

Byzantine Fault Tolerance (BFT) is a concept in distributed computing that refers to the ability of a distributed system to reach a consensus or agreement, even in the presence of faulty or malicious nodes. The term "Byzantine" comes from the Byzantine Generals' Problem, a theoretical problem in computer science that explores the challenges of achieving consensus in a distributed system where some nodes may be unreliable or behave maliciously.

In a Byzantine fault-tolerant system, the goal is to ensure that the system can continue to operate correctly and make progress even if some nodes exhibit arbitrary and potentially conflicting behaviours. These behaviours can include sending incorrect or misleading information, forging messages, or intentionally trying to disrupt the system. To achieve Byzantine fault tolerance, various consensus algorithms have been developed, such as Practical Byzantine Fault Tolerance (PBFT), Byzantine Fault-Tolerant Replication (BFT-R), and ByzCoin. These algorithms use different approaches to enable nodes in a distributed system to agree on a single value or decision, despite the presence of faulty or malicious nodes. One common technique used in BFT algorithms is redundancy, where multiple copies of data or computation are maintained across different nodes. By comparing the results from different nodes and using voting or agreement protocols, the system can identify and mitigate the effects of faulty or malicious nodes. By incorporating BFT mechanisms, blockchain-based e-voting systems can enhance the security, reliability, and resilience of the system against various threats and attacks.

2.3 Implementation Screenshots :

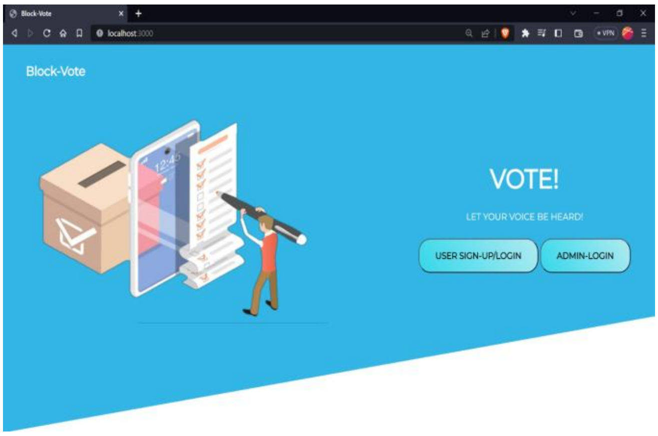


Figure 1 : Homepage

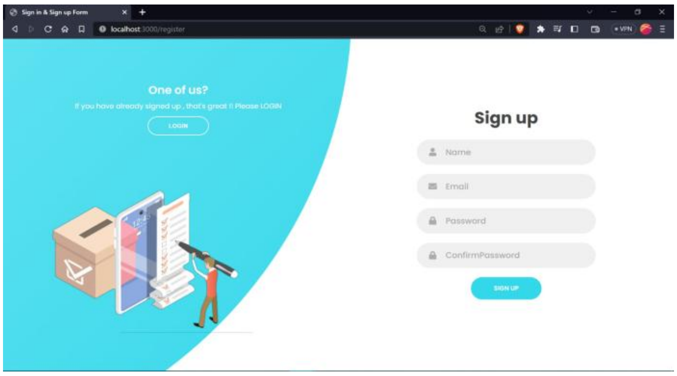


Figure 2 : User Sign - Up Page

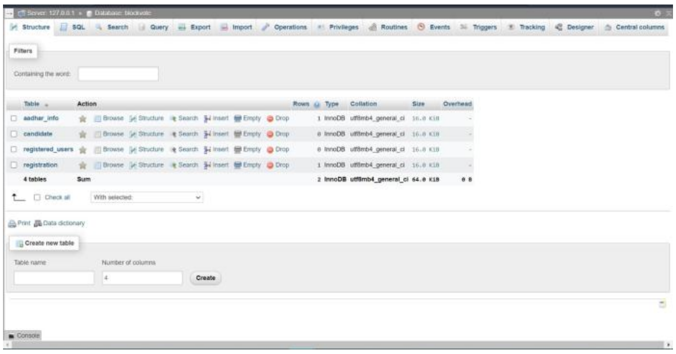
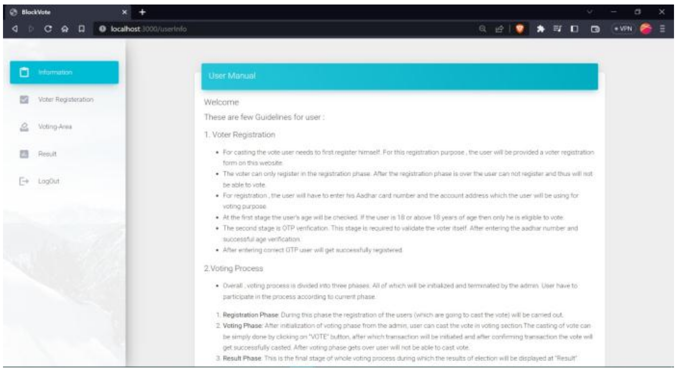

Server: 127.0.0.1 » Database: blockvote
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer Central columns
Filters
Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
☐ aadhar_info	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐ candidate	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐ registered_users	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐ registration	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
4 tables	Sum	2	InnoDB	utf8mb4_general_ci	64.0 KiB	0 B

Figure 3 : Database Implementation



User Manual

Welcome

These are few Guidelines for user :

1. Voter Registration

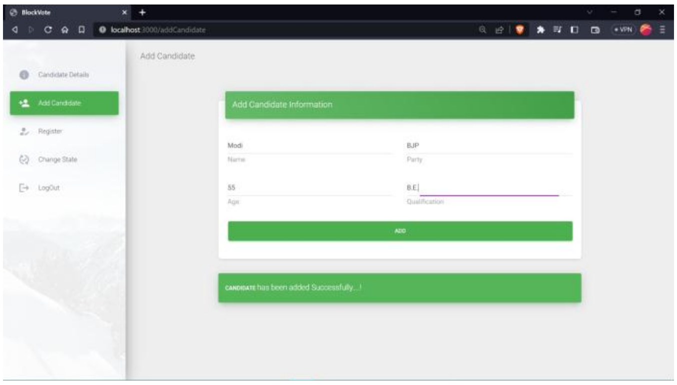
- For casting the vote user needs to first register himself. For this registration purpose , the user will be provided a voter registration form on this website.
- The voter can only register in the registration phase. After the registration phase is over the user can not register and thus will not be able to vote.
- For registration , the user will have to enter his Aadhar card number and the account address which the user will be using for voting purpose.
- At the first stage the user's age will be checked. If the user is 18 or above 18 years of age then only he is eligible to vote.
- The second stage is OTP verification. This stage is required to validate the voter itself. After entering the aadhar number and successful age verification.
- After entering correct OTP user will get successfully registered.

2.Voting Process

- Overall , voting process is divided into three phases. All of which will be initialized and terminated by the admin. User have to participate in the process according to current phase.

1. **Registration Phase**: During this phase the registration of the users (which are going to cast the vote) will be carried out.
2. **Voting Phase**: After initialization of voting phase from the admin, user can cast the vote in voting section.The casting of vote can be simply done by clicking on "VOTE" button, after which transaction will be initiated and after confirming transaction the vote will get successfully casted. After voting phase gets over user will not be able to cast vote.
3. **Result Phase**: This is the final stage of whole voting process during which the results of election will be displayed at "Result"

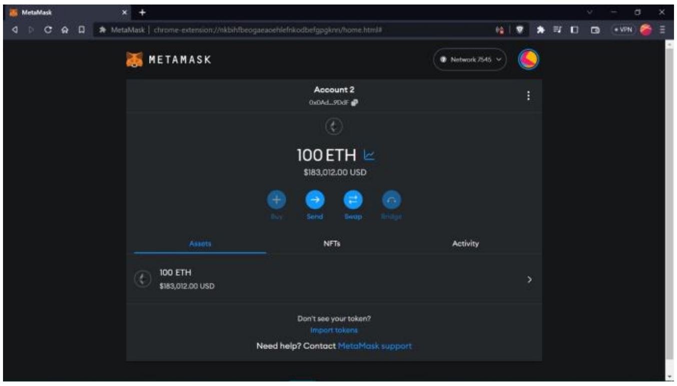
Figure 4 : User Manual



Name: Modi
Party: BJP
Age: 55
Qualification: B.E.

CANDIDATE has been added Successfully...!

Figure 5 : Candidate Dashboard


MetaMask
MetaMask | chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/home.html#
METAMASK
Network 7545
Buy
Send
Swap
Bridge
Assets
NFTs
Activity

Account 2
0x0Ad...9DdF

100 ETH
$183,012.00 USD

100 ETH
$183,012.00 USD

Don't see your token?
Import tokens
Need help? Contact MetaMask support

Figure 6 : Metamask

2.4 Blockchain Network Comparison.

Network	Block Generation Time	Hash rate	Transaction /s	Scalability
Bitcoin	10 mins	330M TH/s	5-7	Very Low
Ethereum	10-19 s	NA	15-30	Low
Ripple	3.5 s	NA	1500	Good
Monero	2 mins	2.5 GH/s	1700	Good
Hyperledger Fabric	10ms User defined	NA	3500	Very Good

Table 1 : Network Comparison

2.5 Performance Metrics :

Performance metrics in blockchain-based e-voting systems focus on measuring the efficiency, speed, and capacity of the system. Here are some performance metrics commonly used in e-voting blockchain-based systems:

- **Transaction Throughput:** This metric measures the number of voting transactions that can be processed per unit of time. It indicates the system's capacity to handle a high volume of votes efficiently. Higher throughput allows for more votes to be processed within a given timeframe.

- **Transaction Latency:** Latency refers to the time delay between a vote being cast and its confirmation on the blockchain. Lower latency is desirable to ensure that votes are recorded and validated quickly, reducing the waiting time for voters.

- **Block Confirmation Time:** Block confirmation time measures the time taken to add a new block to the blockchain. In e-voting systems, it represents the time required for votes to be added to the blockchain and considered as confirmed. A shorter block confirmation time ensures faster validation and inclusion of votes in the blockchain.

- **Scalability:** Scalability assesses the system's ability to handle an increasing number of voters and votes without sacrificing performance. It measures how well the system can scale up to accommodate a growing user base and a larger volume of votes without significant degradation in transaction throughput or latency.

- **Network Congestion:** Network congestion refers to the situation where the network becomes overloaded with voting transactions, leading to delays or bottlenecks. Monitoring network congestion helps identify potential issues that could impact the performance of the e-voting system and allows for the implementation of appropriate measures to alleviate congestion.

- **Consensus Algorithm Efficiency:** The consensus algorithm used in the blockchain impacts the performance of the e-voting system. Metrics such as the time required to reach consensus among nodes and the energy consumption of the consensus mechanism can be measured to assess the efficiency of the consensus algorithm and its impact on the overall system performance.

- **Fault Tolerance:** Fault tolerance measures the system's ability to continue functioning correctly even in the presence of failures or attacks. It includes metrics such as the ability to handle node failures, recover from network partitions, and maintain data integrity in the event of malicious actions or technical glitches.

- **User Experience:** User experience metrics evaluate the usability and convenience of the e-voting system from the perspective of voters. It includes factors such as the ease of accessing and using the system, clarity of instructions, responsiveness of the user interface, and overall satisfaction of voters during the voting process.

Monitoring and optimising these performance metrics help ensure that blockchain-based e-voting systems can efficiently handle a large number of votes, provide timely confirmation and validation, and maintain a high level of reliability and security throughout the voting process.

2.6 Algorithms :

A) **SHA-256 (Secure Hash Algorithm 256-bit)** is a widely used cryptographic hash function. It takes an input message and produces a fixed-size (256-bit) hash value, which is commonly represented as a hexadecimal string.

Properties :

- **Deterministic:** For the same input message, SHA-256 will always produce the same hash value.

- **Preimage resistance:** Given a hash value, it is computationally infeasible to find the original input message.

- **Collision resistance:** It is extremely unlikely for two different input messages to produce the same hash value (collision).

- **Avalanche effect:** A small change in the input message will result in a significantly different hash value.

SHA-256 is commonly used in various applications, including digital signatures, data integrity checks, password storage, and as a building block in blockchain technology for ensuring the immutability and integrity of data within blocks.

B) **Zero-knowledge proofs (ZKPs)** are cryptographic protocols that allow a prover to demonstrate knowledge or possession of certain information to a verifier without revealing any additional details about the information itself.

Working :

- **Setup:** Initially, P and V agree on a common reference, such as a public key or a common assumption that they both trust.

- **Statement:** P wants to prove to V that a certain statement is true, such as possessing a secret value x that satisfies a specific property.

- **Interactive Proof:** P engages in an interactive protocol with V, where they exchange messages and perform computations based on the agreed-upon rules and cryptographic primitives. This protocol is designed in a way that, upon completion, V gains confidence in the validity of the statement without learning anything beyond that.

- **Completeness, Soundness, and Zero-Knowledge:** The zero-knowledge proof should satisfy three fundamental properties:

- **Completeness:** If the statement is true, an honest prover P can convince an honest verifier V of its truth.

- **Soundness:** If the statement is false, no cheating prover can convince an honest verifier with a high probability.

- **Zero-Knowledge:** The proof reveals no additional information beyond the validity of the statement. Even if an observer were to eavesdrop on the entire interaction between P and V, they would not gain any useful knowledge about the underlying secret value or the proof itself.

C) **Homomorphic encryption** relies on mathematical formulas to perform computations on encrypted data. The specific formulas used depend on the type of homomorphic encryption scheme being employed.

The BGV scheme is based on lattice cryptography and utilises the concept of the Learning With Errors (LWE) problem. It allows for both addition and multiplication operations to be performed on encrypted data.

Key Generation: Public Key (pk) and Secret Key (sk) are generated using specific algorithms. The public key is used for encryption, and the secret key is used for decryption.

Encryption: Plaintext (m) is encoded as a lattice point. Random noise (e) is added to the lattice point to make the encryption probabilistic. The encryption algorithm produces the encrypted ciphertext (c) using the public key (pk).

Encryption Formula:

c = Encrypt(pk, m) = (A * s + e + pk * m) mod q

Addition: Addition of two ciphertexts can be performed homomorphically. The addition algorithm takes two ciphertexts (c1 and c2) and adds them to produce a new ciphertext (c').

Addition Formula: c' = c1 + c2 = (c1 + c2) mod q

Multiplication: Multiplication of two ciphertexts can also be performed homomorphically. The multiplication algorithm takes two ciphertexts (c1 and c2) and multiplies them to produce a new ciphertext (c').

Multiplication Formula: c' = c1 * c2 = (c1 * c2) mod q

Decryption: The decryption algorithm takes the ciphertext (c) and uses the secret key (sk) to recover the original plaintext (m).

Decryption Formula:

m = Decrypt(sk, c) = (c - sk * c) mod q

In the above formulas, the variables represent the following:

pk: Public key
sk: Secret key
m: Plaintext message
c: Ciphertext
A: Public parameter
s: Secret parameter
e: Random noise
q: Modulus

These formulas provide a simplified overview of the BGV scheme, which is a powerful fully homomorphic encryption scheme.

2.7 Limitations and Adoptions

The adoption of blockchain technology for e-voting presents several limitations and considerations that need to be addressed before widespread implementation. Here are some key points to consider:

- **Accessibility and Digital Divide:** Blockchain e-voting assumes that all voters have access to the necessary technology and internet connectivity. However, this can exclude certain segments of the population, such as older individuals or those in remote areas with limited infrastructure. Ensuring equal access and bridging the digital divide is crucial.

- **Security and Vulnerabilities:** While blockchain technology provides security benefits, it is not completely immune to vulnerabilities. Threats such as hacking, malware, and phishing attacks can compromise the integrity of the e-voting system. Thorough security measures, including strong encryption and multi-factor authentication, must be in place to mitigate these risks.

- **Privacy and Anonymity:** Maintaining the secrecy of the vote is fundamental in any voting system. Blockchain, as a transparent and immutable ledger, may pose challenges in preserving voter privacy. Striking a balance between transparency and voter anonymity is crucial to ensure a fair and trusted e-voting process.

- **Scalability and Throughput:** Blockchain networks often face scalability issues, particularly when handling a large volume of transactions. As e-voting involves a significant number of participants, the blockchain must be capable of processing votes quickly and efficiently. Solutions like sharding, side chains, or layer-two protocols can enhance scalability.

- **Governance and Consensus:** Consensus mechanisms are central to blockchain networks, but the choice of consensus algorithm can impact the e-voting system's performance and security. The selection process should involve careful consideration of factors such as decentralisation, energy efficiency, and resistance to attacks.

- **Legal and Regulatory Framework:** Implementing blockchain e-voting requires a supportive legal and regulatory framework. This includes addressing issues related to identity verification, jurisdiction, dispute resolution, and the legal status of blockchain-based transactions. Collaboration between policymakers, legal experts, and technologists is crucial to establish comprehensive regulations.

- **Voter Education and Trust:** Introducing a new e-voting system requires significant voter education to build trust and confidence. Clear communication about the benefits, security measures, and transparency of the blockchain e-voting system is essential to address concerns and ensure acceptance from the public.

- **System Auditability and Transparency:** While blockchain provides transparency, it is important to establish mechanisms for auditing and verifying the integrity of the e-voting system. Independent audits, open-source code, and public scrutiny can help detect and address any potential flaws or vulnerabilities.

- **Technical Expertise and Maintenance:** Implementing and maintaining a blockchain e-voting system requires technical expertise and ongoing support. Sufficient resources and skilled professionals must be allocated to ensure the system's smooth operation, timely upgrades, and response to emerging threats.

- **Adoption and Transition:** The adoption of blockchain e-voting requires a gradual transition from traditional voting methods. Pilots and smaller-scale implementations can help identify challenges, fine-tune the system, and gain confidence from stakeholders before expanding to larger elections.

3. CONCLUSIONS

Implementing blockchain technology in electronic voting systems has the potential to drastically alter election operations. By utilising the decentralised, immutable properties of blockchain, e-voting solutions can ensure the correctness and transparency of the voting process. E-voting can also increase voter turnout and improve accessibility for those who might find it difficult to physically visit a polling site. It is important to emphasise, however, that the implementation of an electronic voting system based on blockchain also raises concerns about voter anonymity and the possibility of hacking or system manipulation. It is essential that the proper security measures are implemented in order to assuage these concerns and ensure the validity of the voting process.

REFERENCES

1. What is blockchain technology - IBM Blockchain [Internet]. IBM. [cited 2023Mar14]. Available from: https://www.ibm.com/in-en/topics/what-is-blockchain#

2. Ibrahim MA. What is blockchain technology, an overview. [Internet]. Medium. Medium; 2023 [cited 2023Mar15]. Available from: https://medium.com/@MohammedAminIb2/what-is-blockchain-technology-an-overview-cf1a7067c04

3. Benabdallah A, Audras A, Coudert L, El Madhoun N, Badra M. Analysis of blockchain solutions for e-voting: A systematic literature review. IEEE Access. 2022 Jul 1.

4. Beedham M. Japan is experimenting with a blockchain-powered voting system [Internet]. TNW | Fintech-Ecommerce. 2018 [cited 2023Mar14]. Available from: https://thenextweb.com/news/japan-city-blockchain-voting

5. Ketizmen AM. Smart contracts in EU law [Internet]. ketizmen.av.tr. ketizmen.av.tr.; 2023 [cited 2023Mar15]. Available from: https://www.ketizmen.av.tr/en/post/smart-contracts-in-eu-law

6. Farooq MS, Iftikhar U, Khelifi A. A framework to make voting system transparent using blockchain technology. IEEE Access. 2022 Jun 3;10:59959-69.

7. Blockchain Tutorial [Internet]. Tutorials Point. [cited 2023Mar14]. Available from: https://www.tutorialspoint.com/blockchain/index.htm

8. Kshetri N, Voas J. Blockchain-enabled e-voting. Ieee Software. 2018 Jul 6;35(4):95-9.

9. Al-Madani AM, Gaikwad AT, Mahale V, Ahmed ZA. Decentralized E-voting system based on Smart Contract by using Blockchain Technology. In2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC) 2020 Oct 30 (pp. 176-180). IEEE.

10. Desk P. Votingdao announces upcoming inaugural Blockchain Person of the Year decentralized voting event [Internet]. ZyCrypto. 2022 [cited 2023Mar14]. Available from: https://zycrypto.com/votingdao-announces-upcoming-inaugural-blockchain-person-of-the-year-decentralized-voting-event/

11. (2018). Swiss-Based Agora Records First Government Election on Blockchain as Accblackited Observer in Sierra Leone. (Jan. 15, 2022). [Online]. Available: https://medium.com/agorablockchain/swiss-basedagora-powers-worlds-first-ever-blockchain-elections-in-sierra-leone984dd07a58ee.

12. El Madhoun N, Hatin J, Bertin E. A decision tree for building IT applications: What to choose: blockchain or classical systems?. Annals of Telecommunications. 2021 Apr;76:131-44.

13. Vivek SK, Yashank RS, Prashanth Y, Yashas N, Namratha M. E-voting systems using blockchain: An exploratory literature survey. In2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA) 2020 Jul 15 (pp. 890-895). IEEE.